สำนักหอสมุดกลาง พระจอมเกล้าฉกระบัง

การแสดงกระบวนการผลิตบนเว็บเพจ

PROCESS MONITORING ON WEB PAGE

นายเจษฎา ขันธวรรณา
นายเจษฎา เจริญวัฒนโยธิน
นายพงษ์เอก อุทธโยธา

เลขหมู่.....
เลขทะเบียน 83179
วัน,เดือน,ปี - 6 ส.ค. 2551

b 11962148
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCESS MONITORING ON WEB PAGE

CHEADSADA KANTAVANNA

JETSADA JAREONWATTHANAYOTHIN

PONG-EK AUTHAYOTA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG

2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ การแสดงกระบวนการผลิตบนเว็บเพจ

PROCESS MONITORING ON WEB PAGE

นักศึกษาผู้จัดทำ	นายเจษฎา	ขันธวรรณา	รหัสนักศึกษา 47010121
	นายเจษฎา	เจริญวัฒนโยธิน	รหัสนักศึกษา 47010122
	นายพงษ์เอก	อุทธโยธา	รหัสนักศึกษา 47010478

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2550

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ.เชื้อ นกอยู่	

ภาควิชารับรองแล้ว

(รศ.ประภาษ อุดคกิมาพันธุ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การแสดงกระบวนการผลิตบนเว็บเพจ		
	PROCESS MONITORING ON WEB PAGE		
นักศึกษา	นายเจษฎา	ขันธวรรณา	รหัสนักศึกษา 47010121
	นายเจษฎา	เจริญวัฒนโยธิน	รหัสนักศึกษา 47010122
	นายพงษ์เอก	อุทธโยธา	รหัสนักศึกษา 47010478
อาจารย์ที่ปรึกษา	ผศ.เชื้อ นกอยู่		
ปีการศึกษา	2550		

บทคัดย่อ

โครงงานนี้นำเสนอถึงการใช้ระบบเครือข่าย GPRS ในการใช้งานเชื่อมต่อกับอุปกรณ์ PLC เพื่อที่จะสามารถแสดงผลการทำงานผ่านระบบ GPRS ไปแสดงบนเว็บเพจได้ เพราะฉะนั้นจึงต้องมีการพัฒนาการรับส่งข้อมูลและระบบแสดงผลให้มีความสมบูรณ์มากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Process monitoring on web page	
Authors	Mr. Cheadsada	Kantavanna
	Mr. Jetsada	Jareonwatthanayothin
	Mr. Pong-ek	Authayota
Thesis Advisor	Asst.Prof.Chuae	Nokyoo
Year	2007	

ABSTRACT

This project presents about PLC connect to GPRS network for show the result to web page by GPRS network therefore we will develop and apply receiving-transferring data and show absolutely result.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่จะต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้โครงงานนี้เสร็จลงได้ ก็คือ อาจารย์เชื้อ นกอยู่ อาจารย์ที่ปรึกษาปริญญานิพนธ์ที่ให้ความเอาใจใส่แนะนำและช่วยเหลือเสมอมา นอกจากนี้ยังเอื้อเฟื้อสถานที่ในการทำงานวิจัย ต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดีพร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่และยังให้กำลังใจ เอาใจใส่เสมอมาในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกพระคุณอันสูงสุดและขอกราบขอบพระคุณมา ณ ที่นี้

คุณค่าและประโยชน์อันพึงได้รับจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญรูป.....VII

สารบัญตาราง.....IX

บทที่ 1 บทนำ.....1

1.1 ความสำคัญของปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ขั้นตอนการศึกษา.....2

บทที่ 2 ทฤษฎี.....3

2.1 เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Control).....3

2.1.1 ประวัติความเป็นมาของ PLC.....3

2.1.2 โครงสร้างฮาร์ดแวร์ของ PLC.....4

2.1.3 ส่วนประกอบด้านหน้าของ PLC.....5

2.1.4 การต่อกับแหล่งจ่ายไฟฟ้าภายนอก.....7

2.1.5 การเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล.....7

2.1.6 การติดต่อกับพอร์ตภายนอก.....8

2.1.7 อนาล็อกอินพุตโมดูล.....9

2.1.8 โปรแกรม Direct Soft32.....11

2.2 GPRS.....15

2.2.1 เครือข่าย GSM.....17

2.2.2 การควบคุม GPRS Module.....18

2.3 รูปแบบการเชื่อมต่อ.....18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.1 การติดต่อสื่อสารระหว่าง PLC กับ GPRS Module 18

2.3.1.1 TCP/IP Stack Chip.....18

2.3.1.2 การสื่อสารแบบอนุกรม 19

2.3.1.3 การสื่อสารข้อมูลแบบอะซิงโครนัส.....19

2.3.1.4 มาตรฐานพอร์ทอนุกรมแบบ RS -232.....20

2.3.1.5 คอนเน็กเตอร์สำหรับพอร์ท RS-232 และการเชื่อมต่อ.....21

2.3.2 การ GPRS Module ติดต่อ Server 22

2.3.2.1 Internet.....22

2.3.2.2 TCP/IP Protocol.....23

2.3.2.3 UDP (User Datagram Protocol).....25

2.3.3 การติดต่อ Server/Web browser.....27

2.4 Application Server 29

2.4.1 APACHE.....29

2.4.1.1 ประวัติ.....29

2.4.1.2 ความสามารถ.....30

2.4.2 PHP.....30

2.4.2.1 คุณสมบัติ.....30

2.4.2.2 การรองรับ PHP.....31

2.4.3 MYSQL.....32

2.5 Web Browser.....32

2.5.1 HTML.....32

2.5.2 AJAX.....33

2.5.2.1 ประวัติความเป็นมา.....33

2.5.2.2 หลักการทำงาน.....34

2.5.3 Java Script.....35

2.5.3.1 ประวัติ.....35

2.5.3.2 การใช้งาน.....35

2.5.4 XML.....36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบและทดลองระบบ.....38
3.1 ส่วนของฮารดแวร์ (Hardware).....38
3.1.1 PLC.....38
3.1.2 GPRS Module.....38
3.1.3 TCP/IP Stack Chip.....38
3.2 ส่วนของซอร์ฟแวร์ (Software).....39
3.2.1 Direct Soft 32.....39
3.2.2 Apache.....39
3.2.3 My SQL,PHP.....39
3.2.3 Ajax.....39

บทที่ 4 การทดลองและผลการทดลอง.....40
4.1 การออกแบบการทดลอง.....40
4.2 ขั้นตอนการทดลอง.....40
4.3 บันทึกผลการทดลอง.....41
4.4 สรุปผลการทดลอง.....43

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....44
5.1 ปัญหาทางเทคนิค.....44
5.2 แนวทางการพัฒนา.....44
5.3 สรุป.....44

บรรณานุกรม.....45

ภาคผนวก.....46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1แสดงส่วนประกอบด้านหน้าของ PLC 5
2.2 Wiring ทางด้าน Output.....7
2.3 Wiring ทางด้าน Input.....7
2.4 แสดงการเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล.....7
2.5 แสดงการเชื่อมต่อกับพอรท RS-232.....8
2.6 แสดง Analog Input Module แบบ 4 Channel.....9
2.7 แสดงโครงสร้างภายในของ Analog Input Module แบบ 4 Channel.....9
2.8 แสดงตัวเครื่อง PLC KOYO รุ่น DL06 เมื่อทำการติดตั้งการ์ด Analog และ การ์ดอื่นๆ.....10
2.9 แสดงการสร้าง Project ใหม่ขึ้นมาเพื่อทำการเขียนโปรแกรม.....11
2.10 แสดงการเลือกชนิดของ PLC และรุ่นของ PLC หรือชนิดของ CPU แล้วคลิก OK.....11
2.11 แสดงการต่อ PLC กับ Computer และการทำการ Link กับโปรแกรม DirectSOFT32.....12
2.12 แสดงหน้าจอที่ไว้สำหรับเขียนโปรแกรมให้กับ PLC.....12
2.13 แสดงการเรียกใช้ข้อมูลจากฐานข้อมูลต่างๆ.....12
2.14 แสดงบล็อกสัญลักษณ์การเขียน Ladder.....13
2.15 แสดงไอคอนการติดต่อกับ PLC.....13
2.16 แสดงการเลือก Mode การทำงานของ PLC.....14
2.17 แสดงการทำงานของโปรแกรมที่อยู่ใน Mode RUN.....14
2.18 แสดงเครือข่ายการทำงานของ GPRS.....16
2.19 แสดงเครือข่าย GSM.....17
2.20 แสดงการทำงานของ TCP/IP Stack Chip.....19
2.21 แสดงการแบ่งการทำงานของโปรโตคอล TCP และ UDP.....26
2.22 แสดงความแตกต่าง HTML และ HTTP.....27
2.23 แสดงขั้นตอนการติดต่อ ของโปรโตคอล HTTP.,.....28
2.24 แสดงการทำงานแบบไม่จองสายของ HTTP.....29
2.25 แสดงการทำงาน ของ AJAX.....34
2.26 เปรียบเทียบเวปแอปลิเคชันทั่วไป กับ AJAX.....35
3.1 แสดงการติดต่อของระบบ.....38
4.1 แสดงหน้าเวปเริ่มต้น.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.2 แสดงหน้าเวปเมื่อเริ่ม start และเริ่มแสดงค่าสุดท้ายจากเซิร์ฟเวอร์.....41

4.3 แสดงฟังก์ชันสถานะและการควบคุมอุปกรณ์.....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงสถานการณ์ทำงานของ PLC.....6
2.2 แสดงการทำงานที่ Mode ต่างๆของสวิตซ์.....6
2.3 แสดงการเชื่อมต่อกับพอรท RS-232.....8
2.4 แสดงค่า V-Memory ที่เกี่ยวข้องกับ Analog Input Module.....10
2.5 แสดงค่าที่บันทึกในเซิร์ฟเวอร์.....42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันระบบควบคุมอัตโนมัติเป็นปัจจัยที่ขาดไม่ได้ สำหรับงานทางด้านระบบการผลิตในโรงงานอุตสาหกรรม รวมทั้งระบบที่ให้บริการทางด้านสาธารณูปโภค เช่น โรงผลิตและจ่ายไฟและน้ำ เพราะสามารถลดขั้นตอนการทำงานที่ยุ่งยากช่วยเพิ่มประสิทธิภาพการผลิตและช่วยอำนวยความสะดวกนานัปการแก่ผู้ปฏิบัติงานควบคุมการผลิต การที่สามารถนำเอาค่าสัญญาณหรือข้อมูลจากกระบวนการผลิตมาทำการประมวลผล เพื่อให้ได้ค่าหรือผลลัพธ์ควบคุมสำหรับส่งออกไปยังอุปกรณ์ควบคุม เพื่อให้สามารถควบคุมกระบวนการผลิตได้อย่างถูกต้องแม่นยำ

อีกทั้งปัจจุบันถ้าพูดถึงคอมพิวเตอร์จะต้องคิดถึงเครือข่ายอินเตอร์เน็ต ที่มีโครงข่ายสื่อสารทั่วโลก และอินเตอร์เน็ตยังเป็นสิ่งใกล้ตัวที่ผู้คนเกือบทุกกลุ่มรู้จักและสามารถเข้าถึงอินเตอร์เน็ตได้อย่างสะดวกในสถานที่ที่ต่างกัน

โครงงานนี้เป็นการประยุกต์ใช้เครือข่ายอินเตอร์เน็ต ซึ่งเป็นเครือข่ายสื่อสารระยะไกลในการส่งค่า จาก PLC ซึ่งเป็นตัวควบคุมผ่านจีพีอาร์เอสโมดูล สู่เครือข่ายอินเตอร์เน็ตไปยัง Server และใช้ Web Browser เพื่อเปลี่ยนแปลงฐานข้อมูลใน Server เพื่อการควบคุม รวมทั้งรับค่าจาก Server เพื่อนำไปแสดงผลในหน้า Web Browser

1.2 วัตถุประสงค์ของการทำโครงงาน

1. เพื่อศึกษาและประยุกต์ใช้งาน PLC
2. เพื่อศึกษาการสื่อสารข้อมูลเครือข่าย Internet
3. เพื่อศึกษาและสามารถใช้ GPRS Module ในการรับส่งข้อมูล
4. เพื่อศึกษาและประยุกต์ใช้ Web Page ในการแสดงผลและควบคุม Plant

1.3 ขอบเขตของปริญญานิพนธ์

1. ใช้ PLC ควบคุม Plant
2. สามารถส่งค่าจาก PLC มายัง Web Browser
3. ใช้ Web Browser ส่งค่าควบคุมไปยัง PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของ PLC
2. ศึกษาการสื่อสารข้อมูลทางอินเตอร์เน็ต
3. ศึกษาการทำงานของ Client/Server
4. ศึกษาการเขียน Web Page
5. ศึกษาการระบบการเก็บข้อมูล
6. ใช้ Web Browser เพื่อสั่งงาน Plant ทำงานผ่านทางหน้าจอเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Control)

PLC เป็นอุปกรณ์โซลิดสเตจ (Solid State) ที่ทำงานแบบลอจิก (Logic Function) การออกแบบการทำงานของ PLC ใช้สำหรับควบคุมกระบวนการทำงานของเครื่องจักรและอุปกรณ์

การใช้ (PLC) สำหรับควบคุมเครื่องจักรหรืออุปกรณ์ต่าง ๆ ในโรงงานอุตสาหกรรมจะมีข้อได้เปรียบกว่าการใช้ระบบของรีเลย์ (Relay) ซึ่งจำเป็นจะต้องเดินสายไฟฉะนั้นเมื่อมีความจำเป็นจะต้องเปลี่ยนกระบวนการผลิต หรือลำดับการทำงานใหม่ ก็จะต้องเดินสายใหม่ซึ่งเสียเวลา และค่าใช้จ่ายสูงแต่เมื่อเปลี่ยนมาใช้ PLC แล้ว การเปลี่ยนกระบวนการผลิตก็ทำได้ง่าย โดยการเปลี่ยนโปรแกรมใหม่เท่านั้น นอกจากนี้แล้ว PLC ยังใช้ระบบโซลิด-สเตจ ซึ่งน่าเชื่อถือกว่าเดิมกินกระแสไฟน้อยกว่าและสะดวกกว่าเมื่อต้องขยายงาน

จากหลักการพื้นฐานแล้วอุปกรณ์ตัวนี้ทำงานในลักษณะระบบเลขฐานสองคือ เปิด – ปิด หรือ ON – OFF หรือสัญญาณลอจิกเท่านั้น แต่ปัจจุบันนี้ไม่ได้เป็นเช่นนี้อีกแล้ว คือ สามารถรับส่ง Input แบบต่อเนื่อง หรือ สัญญาณอนาล็อก (Analog) ซึ่งในปริญญานิพนธ์ฉบับนี้ได้นำ PLC มาใช้ในการควบคุมการทำงานของกระบวนการโดยการสั่งการและดูการสั่งการและดูสถาณการณ์ การทำงานจากคอมพิวเตอร์ ซึ่งเป็นการสื่อสารข้อมูลกันระหว่างเครื่อง PLC กับคอมพิวเตอร์เป็นผลให้ระบบควบคุมการงานมีประสิทธิภาพมากขึ้น

2.1.1 ประวัติความเป็นมาของ PLC

PLC ย่อมาจากคำว่า Programmable Logic Controller เป็นอุปกรณ์อิเล็กทรอนิกส์ที่มีหน่วยความจำ ในการเก็บโปรแกรมการทำงานของอุปกรณ์ต่าง ๆ ที่ต่อขั้วเข้าขั้วออกของมัน PLC ยังมีชื่อเรียกอีกอย่างอื่นอีก เช่น PC ย่อมาจาก Programmable Controller และ SC ซึ่งย่อมาจาก Sequence- Controller PLC ขนาดเล็กอาจเรียกว่าเป็นตัวซีเควนเซอร์ก็มี

PLC เริ่มพัฒนาขึ้นครั้งแรกในปี ค.ศ. 1986 โดย Hydromantic Division ของบริษัท General Motors Corporation เพื่อใช้แทนการควบคุมแบบเก่า ๆ ซึ่งใช้ Relay ติดตั้ง ดัดแปลงและแก้ไขได้ยากลำบากมาเป็นการควบคุมแบบใหม่ที่ใช้วงจรอิเล็กทรอนิกส์แทนรีเลย์และใช้การเขียนโปรแกรมทำนองเดียวกันกับการเขียนโปรแกรมคอมพิวเตอร์ กำหนดเงื่อนไขในการควบคุมแทนการเดินสายเชื่อมต่อไฟฟ้าแบบเก่าเพื่อเพิ่มความสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 โครงสร้างฮาร์ดแวร์ของ PLC

1. หน่วยประมวลผลกลาง (CPU: Central Processing Unit) ส่วนที่ทำหน้าที่การควบคุมการทำงานของ PC ซึ่งจะใช้ไมโครโปรเซสเซอร์ชนิด 8 บิต เป็นตัวประมวลผล ปกติหน้าที่ของ CPU คือ รับข้อมูลอินพุตเข้ามาทำการประมวลผลแล้วส่งผลที่ได้ไปยังเอาท์พุตจากนั้นก็จะวนไปรับข้อมูลอินพุตเข้ามาอีกแล้วทำซ้ำๆลักษณะนี้ไปเรื่อยๆ ซึ่งเรียกว่า Scan การทำงานของ CPU จะอยู่ภายใต้การควบคุมของโปรแกรมที่ผู้ใช้ป้อนเข้าไป

2. หน่วยความจำ (Memory Unit) ทำหน้าที่เก็บโปรแกรมข้อมูล ขนาดของหน่วยความจำจะเป็นตัวกำหนดขีดความสามารถของปกติมักจะมีขนาดวัดเป็นจำนวน Stepโดยมีหน่วยความจำชนิดต่างๆ ดังนี้

- ROM (Read Only Memory) เป็นหน่วยความจำที่ไม่อนุญาตให้ผู้ใช้เปลี่ยนแปลงแก้ไขข้อมูลภายใน แต่สามารถรักษาข้อมูลไว้ได้แม้ว่าจะไม่มีกระแสไฟฟ้าเหมาะสำหรับเก็บโปรแกรมบริหารระบบ หรือโปรแกรมที่เสร็จสมบูรณ์ไม่ต้องการแก้ไข

- RAM (Random Access Memory) เป็นหน่วยความจำที่ผู้ใช้สามารถเปลี่ยนแปลงแก้ไขได้เหมาะสำหรับเก็บโปรแกรมที่อยู่ในช่วงกำลังพัฒนาหรือต้องการเปลี่ยนแปลงบ่อยครั้งในการใช้งานจริงๆแล้วจะต้องมีแหล่งจ่ายไปสำรองต่อไว้เพื่อป้องกันไม่ให้ข้อมูลสูญหายเมื่อเกิดไฟดับ

- EPROM (Erasable Access Read Only Memory) เป็นหน่วยความจำสำหรับเก็บโปรแกรมที่มีการพัฒนาจนใช้งานได้ดีให้เป็นการถาวร และมีความสามรถในการอัดโปรแกรม จะทำได้โดยการถ่ายข้อมูล จากหน่วยความจำ RAM ลงสู่หน่วยความจำ EPROM โดยอาศัยเครื่องอัดชนิดพิเศษเพื่อใช้ในการต่อรวมกับชุด PC หน่วยความจำชนิดนี้ หน่วยความจำประเภทนี้โปรแกรมจะไม่มีการสูญหายเมื่อเกิดไฟดับแต่ถ้ามีความจำเป็นที่จะลบโปรแกรมภายในก็สามารถทำได้โดยการล้างโปรแกรมภายใน

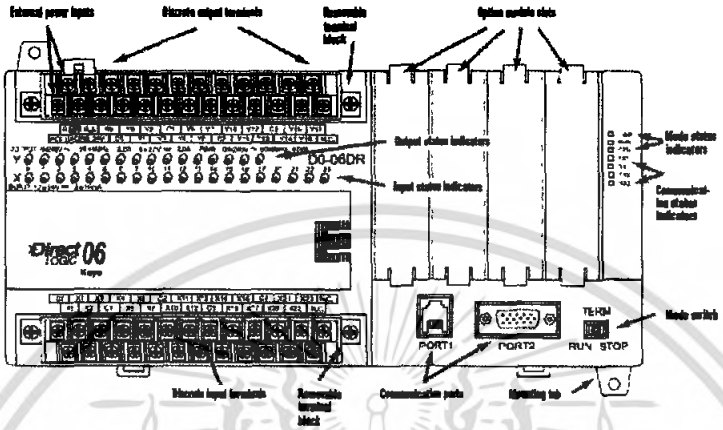
- EEPROM (Electrically Erasable Access Read Only Memory) มีลักษณะคล้ายกับ EPROM คือหน่วยความจำที่เหมาะสำหรับโปรแกรมผู้ใช้ที่เสร็จสมบรูณ์แล้ว การอัดโปรแกรมจะอาศัยเครื่องอัดโปรแกรมชนิดพิเศษเช่นเดียวกัน แต่ตอนลบข้อมูลจะใช้วิธีป้อนสัญญาณพัลส์และการลบข้อมูลเก่าไม่จำเป็นต้องลบทั้งหมด ผู้ใช้สามารถแก้ไขเฉพาะตำแหน่งที่ต้องการ

PLC จะรับสัญญาณมาที่ขั้วเข้าและให้สัญญาณออกทางขั้วออกการให้สัญญาณออกนี้จะเป็นไปตามโปรแกรมที่เก็บไว้ใน PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ส่วนประกอบด้านหน้าของ PLC

ประกอบด้วยส่วนของการติดต่อ ส่วนแสดงผล และสัญลักษณ์คำอธิบายแสดงอยู่บนด้านหน้าของ PLC ขนาดเล็กรุ่น DL06 พอร์ตการติดต่อจะอยู่ด้านหน้าของ PLC ดังรูป



รูปที่ 2.1 แสดงส่วนประกอบด้านหน้าของ PLC

ส่วนการติดต่อที่ Terminals ล่างจะเป็นในส่วนของอินพุตซึ่งมีให้เลือกใช้ตั้งแต่ X0, X1, X2, X3, X4, X5, X6, X7, X10, X11, X12, X13, X14, X15, X16, X17, X20, X21, X22 และ X23 โดยที่ X0 ถึง X3 จะใช้คอมมอนร่วมคือ CO, X20 ถึง X23 จะใช้คอมมอนร่วม คือ C4

ส่วนของการติดต่อที่ Terminals บนจะเป็นส่วนของเอาท์พุทและช่องต่อแหล่งจ่ายจากภายนอกซึ่งในส่วนของเอาท์พุตมีตั้งแต่ Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7, Y10, Y11, Y12, Y13, Y14, Y15, Y16 และ Y17 โดยที่ Y0 ถึง Y3 จะใช้คอมมอนร่วมคือ C0, Y4 ถึง Y7 จะใช้ คอมมอนร่วมคือ C1, Y10 ถึง Y13 จะใช้คอมมอนร่วมคือ C2, Y14, Y17 จะใช้คอมมอนร่วมคือ C3

ในส่วนของ LED สองแถวใต้ Terminals ของเอาท์พุตคือไฟแสดงผลของอินพุทและเอาท์พุตของแต่ละ Terminals ส่วนไฟแสดงผลด้านซ้ายมือสุดของ PLC คือไฟแสดงสถานการณ์ทำงานของ PLC ซึ่งมีรายละเอียดดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงสถานการณ์ทำงานของ PLC

Status Indicators		
Indicator	***Status***	***Meaning***
PWR	ON	Power good
	OFF	Power failure
RUN	ON	CPU is in Run Mode
	OFF	CPU is in Stop or Program Mode
CPU	ON	CPU self diagnostics error
	OFF	CPU self diagnostics good
TX1	ON	Data is being transmitted by the CPU-Port 1
	OFF	No data is being transmitted by the CPU-Port 1
RX1	ON	Data is being received by the CPU-Port 1
	OFF	No data is being received by the CPU-Port 1
TX2	ON	Data is being transmitted by the CPU-Port 2
	OFF	No data is being transmitted by the CPU-Port 2
RX2	ON	Data is being received by the CPU-Port 2
	OFF	No data is being received by the CPU-Port 2

ส่วน Option Slots ทั้งสี่ช่องมีไว้สำหรับนำการ์ดชนิดต่างๆขึ้นอยู่กับการใช้งาน เช่น Analog, I/O เป็นต้น ส่วนด้านล่างของ Option slots จะเป็นพอร์ตที่ใช้ในการสื่อสารของ PLC ซึ่งพอร์ตที่ 1 จะมีลักษณะเป็น RJ12 Phone plug 6 – pin และพอร์ตที่2 เป็นแบบ DB 15 pin VGA (female) ตารางสุดท้ายเป็น Mode switch ซึ่งมีรายระเอียดการทำงานดังนี้

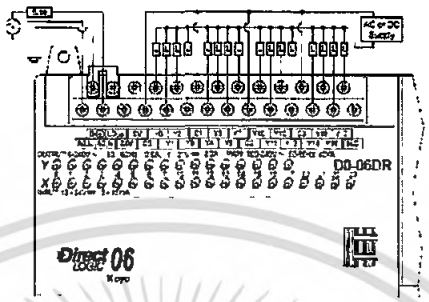
ตารางที่ 2.2 แสดงการทำงานที่ Mode ต่าง ๆ ของสวิตช์

Mode Switch Position	CPU Action
RUN (Run Program)	CPU is forced into the RUN mode if no errors are encountered. No program changes are allowed by the programming/monitoring device.
TERM (Terminal)	RUN PROGRAM and the TEST modes are available. Mode and program changes are allowed by the programming/monitoring device.
STOP	CPU is forced into the STOP mode. No changes are allowed by the programming/monitoring device.

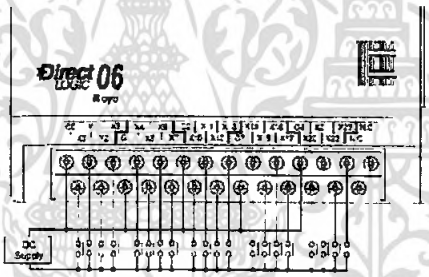
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การต่อกับแหล่งจ่ายไฟฟ้าภายนอก

สำหรับ PLC รุ่น KOYO DL06 มีการต่อแหล่งจ่ายไฟฟ้าดังรูปด้านล่าง เมื่อทำการเชื่อมสายเสร็จแล้วให้เอาฉนวนหุ้มส่วนของการติดต่อ ห้ามเปิดเครื่องขณะทำขั้นตอนดังกล่าวอยู่



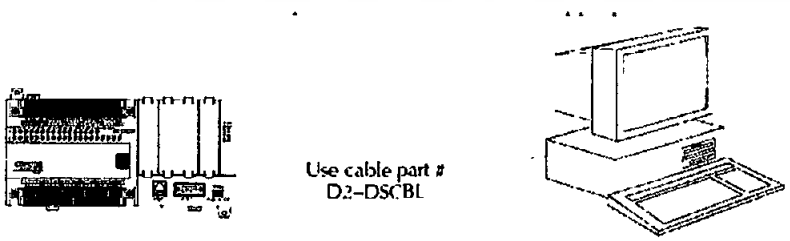
รูปที่ 2.2 Wiring ทางด้าน Output



รูปที่ 2.3 Wiring ทางด้าน Input

2.1.5 การเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล

PLC KOYO DL06 สามารถทำการโปรแกรมได้จาก โปรแกรม Direct SOFT บนเครื่องคอมพิวเตอร์โดยทำการต่อสายเคเบิลดังรูป

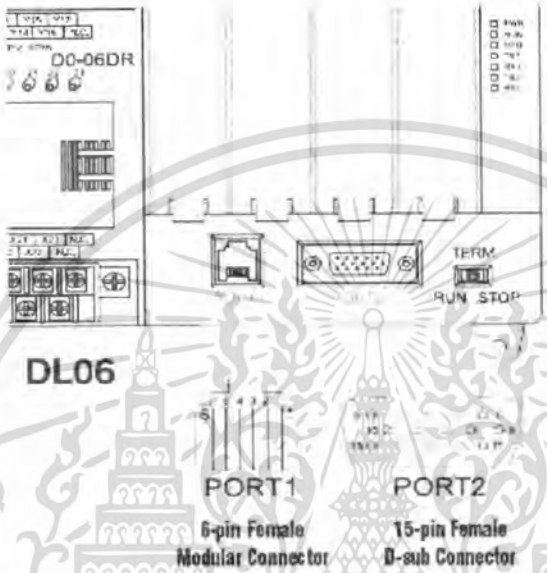


รูปที่ 2.4 แสดงการเชื่อมต่อกับคอมพิวเตอร์ส่วนบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การติดต่อกับพอร์ตภายนอก

การติดต่อกับพอร์ตภายนอก เราจำเป็นที่ต้องใช้สายขาวเพียงพอเพื่อที่จะทำการเชื่อมต่อระหว่าง PLC กับคอมพิวเตอร์ส่วนบุคคลได้อย่างง่ายดายและรวดเร็ว โดยใช้สายสัญญาณมาตรฐาน RS –232C โดยใช้หัวปลั๊กโทรศัพท์ RJ-12 ในการทำหัวแจ็คคู่กับพอร์ตอนุกรม ดังรูป



รูปที่ 2.5 แสดงการเชื่อมต่อกับพอรท RS-232

ตารางที่ 2.3 แสดงการเชื่อมต่อกับพอรท RS-232

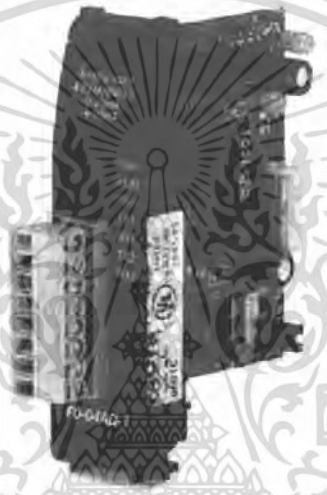
DL05 & DL06 Port 1 Pin Descriptions		
1	0V	Power (-) connection (GND)
2	5V	Power (+) connection
3	RXD	Receive data (RS-232C)
4	TXD	Transmit data (RS-232C)
5	5V	Power (+) connection
6	0V	Power (-) connection (GND)

DL06 Port 2 Pin Descriptions		
1	5V	Power (+) connection
2	TXD	Transmit data (RS-232C)
3	RXD	Receive data (RS-232C)
4	RTS	Ready to send (RS232C)
5	CTS	Clear to send (RS232C)
6	RXD-	Receive data (-) (RS-422/485)
7	0V	Power (-) connection (GND)
8	0V	Power (-) connection (GND)
9	TXD+	Transmit data (+) (RS-422/485
10	TXD-	Transmit data (-) (RS-422/485)
11	RTS+	Ready to send (+) (RS-422/485)
12	RTS-	Ready to send (-) (RS-422/485)
13	RXD+	Receive data (+) (RS-422/485)
14	CTS+	Clear to send (+) (RS-422/485)
15	CTS-	Clear to send (-) (RS-422/485)

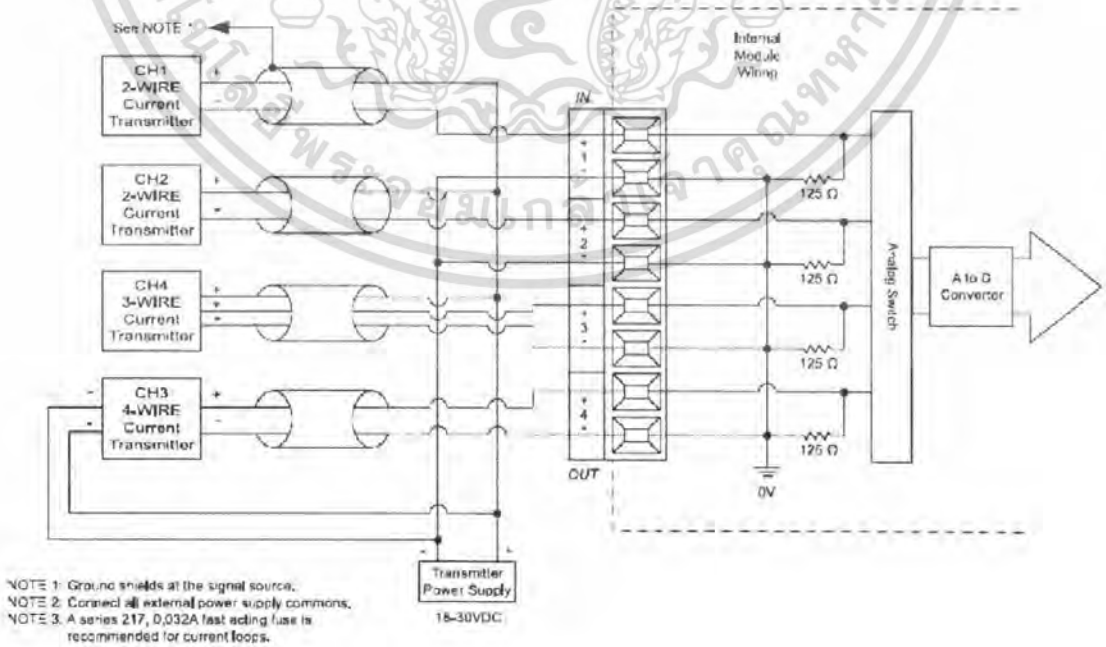
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 อนาล็อกอินพุทโมดูล

เป็นอุปกรณ์ที่ใช้รับสัญญาณอนาล็อกจากภายนอกแล้วทำการแปลงมาเป็นสัญญาณ เพื่อให้ CPU ของเครื่อง PLC สามารถทำการประมวลผลได้ ดังนั้น Analog Input Module จึงมีความจำเป็นเมื่อต้องการที่จะควบคุมขบวนการที่มีการส่งสัญญาณควบคุมออกมาเป็นสัญญาณอนาล็อกเราสามารถเลือกใช้ Analog Input Module ได้ตามความเหมาะสมกับความต้องการได้เนื่องจากในจำนวนตลาดจะมีให้เลือกจำนวน Channel ได้เช่น 2 Channel, 4 Channel หรือ จะมีทั้ง Input และ Output ใน Module ตัวเดียวกันซึ่งในโครงงานนี้จะเลือกใช้ Analog Input Module แบบ 4 Channel ดังรูป



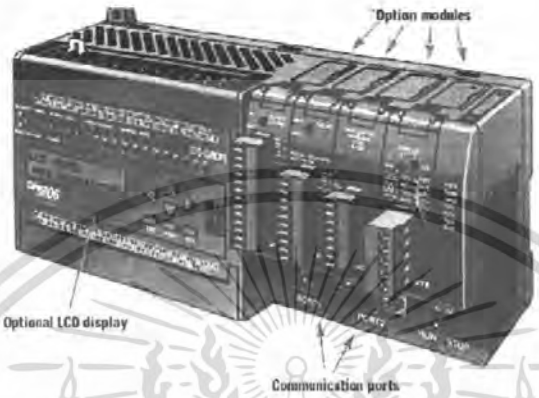
รูปที่ 2.6 แสดง Analog Input Module แบบ 4 Channel



รูปที่ 2.7 แสดงโครงสร้างภายในของ Analog Input Module แบบ 4 Channel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งการ์ด Analog กับ PLC KOYO รุ่น DL06 ในตัวเครื่อง PLC รุ่น DL06 ของ KOYO จะมีช่องที่ใช้สำหรับติดต่อกับ Module พิเศษอยู่แล้ว 1 ช่อง การที่จะติดตั้งสามารถทำได้สะดวกเพียงแค่ทำการเสียบลงในช่องที่กำหนดให้อยู่แล้วกับตัวเครื่อง PLC ก็สามารถใช้งานได้ทันที



รูปที่ 2.8 แสดงตัวเครื่อง PLC KOYO รุ่น DL06 เมื่อทำการติดตั้งการ์ด Analog และ การ์ดอื่นๆ

V-Memory ที่เกี่ยวข้องกับ Analog Input Module ก็จะมี Data Type เลขของ Channel อยู่ที่ V700, V710, V720 และ V730 ขึ้นอยู่กับช่อง Slot ที่ใช้ส่วน Storage Pointer อยู่ที่ V701, V711, V721 และ V731 ขึ้นอยู่กับช่อง Slot ที่ใช้ซึ่งเห็นได้จากตารางดังต่อไปนี้

ตารางที่ 2.4 แสดงค่า V-Memory ที่เกี่ยวข้องกับ Analog Input Module

Analog Input Module DL06 Special V-memory Locations				
Slot No.	1	2	3	4
Data Type and Number of Channels	V700	V710	V720	V730
Storage Pointer	V701	V711	V721	V731

Data Type/Number of channels มีไว้สำหรับเลือกชนิดข้อมูลที่เป็นค่าซึ่งมาจากภายนอก และเลือกว่าจะใช้ Channels ไหน เช่น เราเลือกใช้ Slot 3 V720 = 0400 (BCD) จะเป็น Analog Channel ที่ 4 และข้อมูลเป็นแบบ BCD Storage Pointer (V721) ใช้สำหรับการ Set ให้นำข้อมูลไปเก็บไว้ที่ V-Memory ตำแหน่งต่างๆ

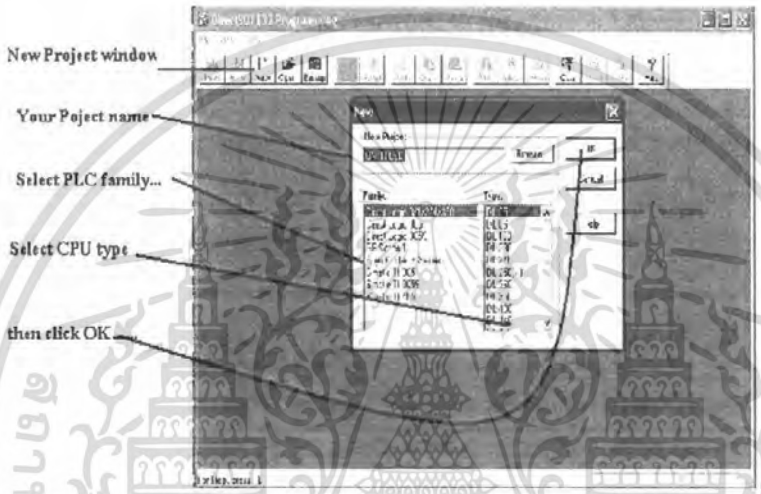
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8 โปรแกรม Direct Soft32

ในการเขียนโปรแกรมลงใน PLC KOYO รุ่น DL06 เราใช้โปรแกรม Direct SOFT32 ในการเขียน Ladder ซึ่งเป็นโปรแกรมการทำงานของ PLC แล้วทำการ Memory ลงใน CPU ของ PLC

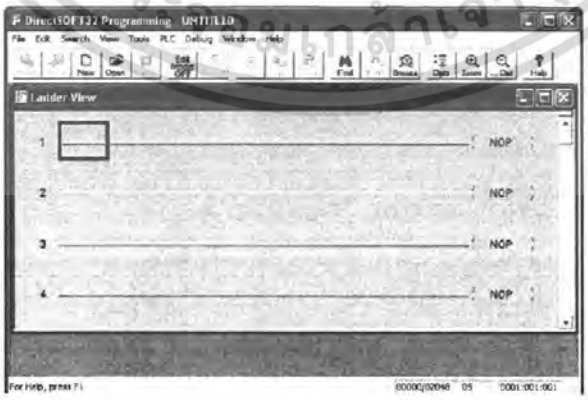
การใช้งานโปรแกรม DirectSOFT32

ขั้นตอนที่ 1 เมื่อทำการคลิกที่โปรแกรม DirectSOFT32 ก็จะปรากฏหน้าต่างดังรูป



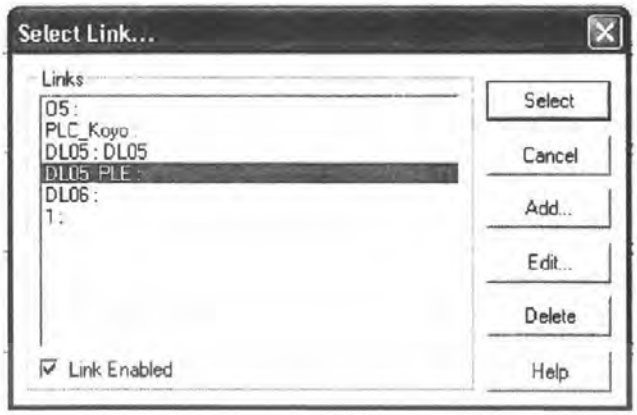
รูปที่ 2.9 แสดงการสร้าง Project ใหม่ขึ้นมาเพื่อทำการเขียนโปรแกรม

ขั้นตอนที่ 2 ทำการต่อ PLC เข้ากับ Computer จากนั้นทำการ Link กับโปรแกรม DirectSOFT32 โดยเลื่อนเมาส์ไปที่ menu bar > PLC > Connect ก็จะปรากฏหน้าต่าง ดังรูป

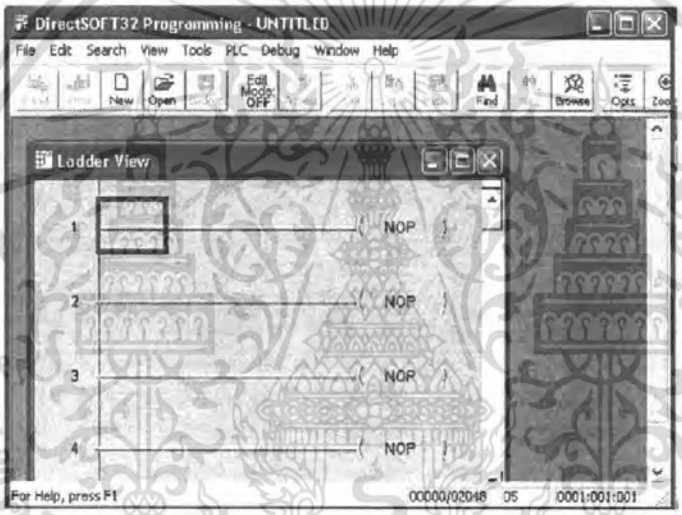


รูปที่ 2.10 แสดงการเลือกชนิดของ PLC และรุ่นของ PLC หรือชนิดของ CPU แล้วคลิก OK

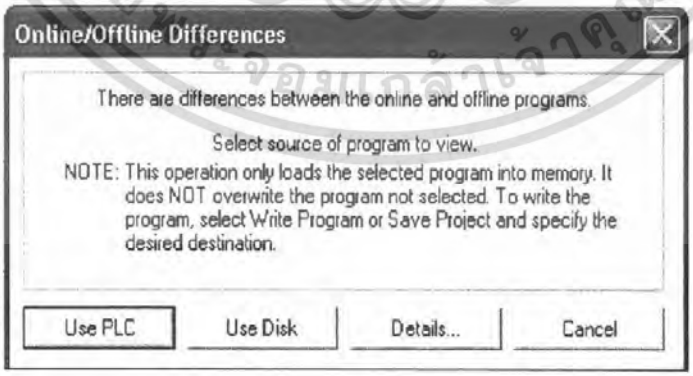
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงการต่อ PLC กับ Computer และการทำการ Link กับโปรแกรม DirectSOFT32



รูปที่ 2.12 แสดงหน้าจอที่ไว้สำหรับเขียนโปรแกรมให้กับ PLC

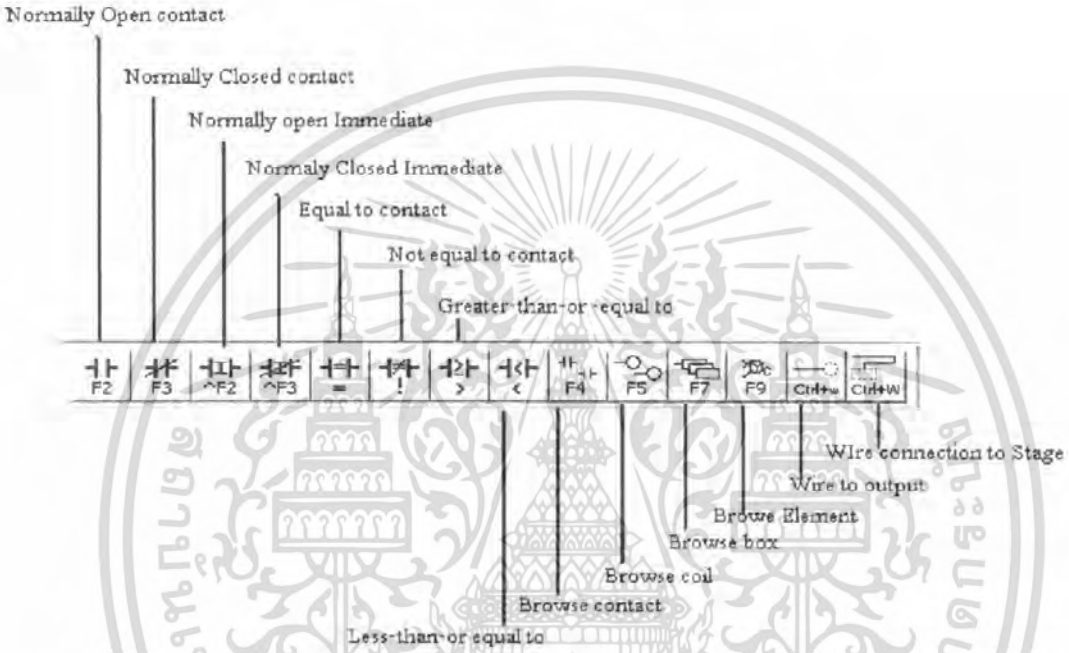


รูปที่ 2.13 แสดงการเรียกใช้ข้อมูลจากฐานข้อมูลต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหน้าต่างข้างบนถ้าเราเลือก Use PLC โปรแกรมก็จะทำการเรียกข้อมูลที่อยู่ใน CPU ของ PLC ออกมาเพื่อทำการ RUN โปรแกรม แต่ถ้าเราเลือก Use Disk โปรแกรมก็จะทำการเตรียมพร้อมที่จะบันทึกโปรแกรมใหม่ลงใน CPU ของ PLC

ขั้นตอนที่3 การเขียนโปรแกรมทำได้โดยการคลิกที่ Edit Mode off (ปุ่มจะมีสีเหลือง) จะปรากฏบล็อกสัญลักษณ์ในการเขียน Ladder ดังรูป

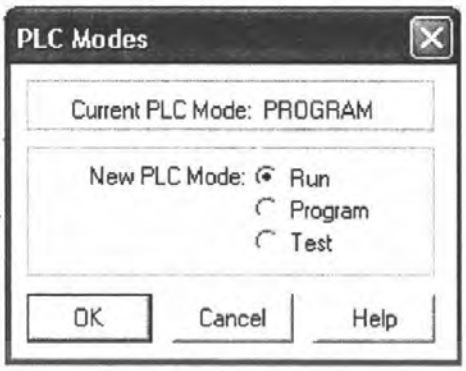


รูปที่ 2.14 แสดงบล็อกสัญลักษณ์การเขียน Ladder

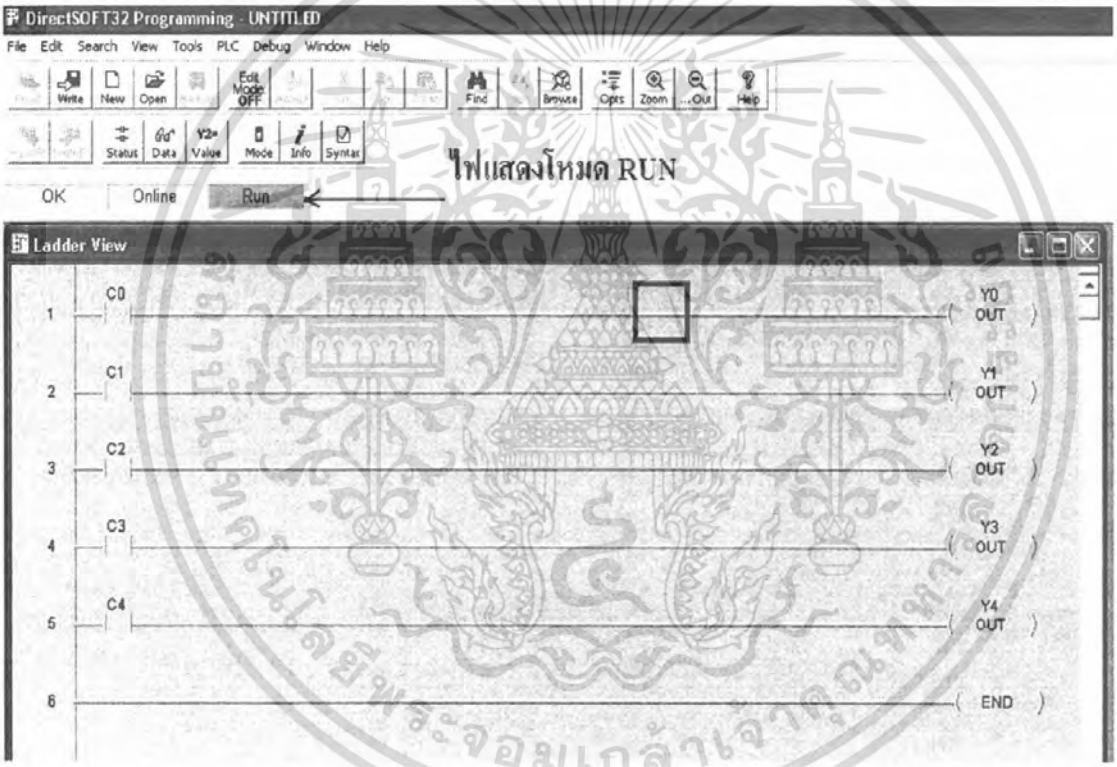
ขั้นตอนที่4 การเขียนโปรแกรมใน PLC เมื่อเครื่อง PC และ PLC ทำการติดต่อกันได้แล้วก็สามารถที่จะเขียนโปรแกรมลงบน PLC ได้ โดยผ่านโปรแกรม DirectSOFT32 ซึ่งทำได้โดยการคลิกที่ไอคอน ที่สองข้างล่างซ้ายซึ่งเป็นไอคอน Write โปรแกรมก็จะถูกเขียนลงใน PLC ดังรูป

รูปที่ 2.15 แสดงไอคอนการติดต่อกับ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงการเลือก Mode การทำงานของ PLC



รูปที่ 2.17 แสดงการทำงานของโปรแกรมที่อยู่ใน Mode RUN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

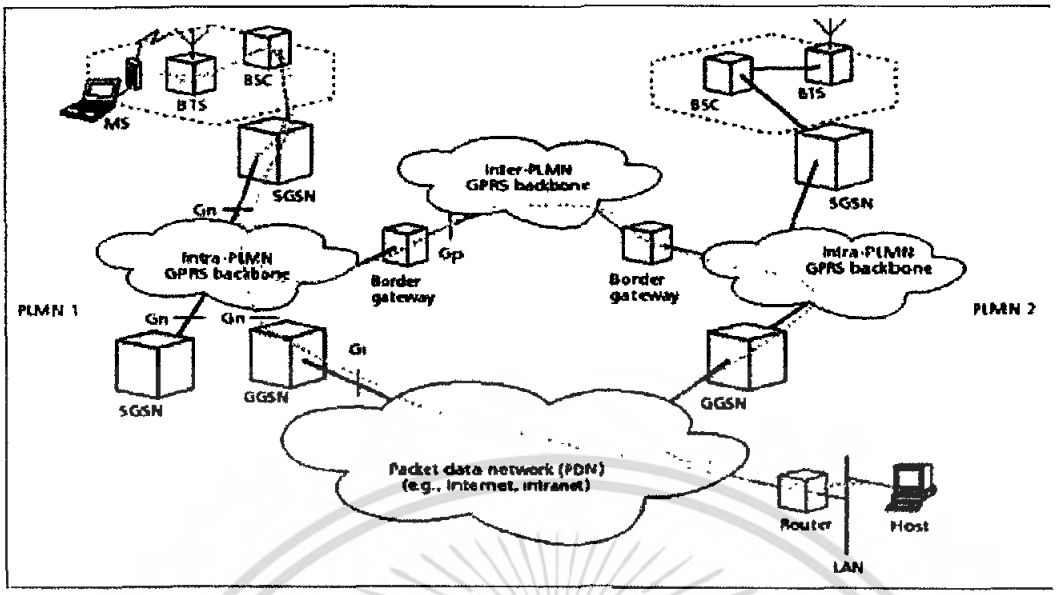
2.2 GPRS

GPRS เป็นตัวย่อจากภาษาอังกฤษ "General Packet Radio Service" ซึ่งเป็นการส่งข้อมูลต่าง ๆ ในรูปแบบแพ็กเก็ตต่าง ๆ การเชื่อมต่อแบบใหม่ที่ใช้ระบบ GPRS เข้ามาก็จะเป็นการเชื่อมต่อและวิธีการส่งข้อมูลที่มีลักษณะเช่นเดียวกับอินเตอร์เน็ตก็คือ เมื่อต้องการข้อมูลหรือส่งข้อมูลอะไรก็แล้วแต่ ก็จะเป็นการส่งข้อมูลลักษณะนั้น เข้าไปในเครือข่ายโทรศัพท์เคลื่อนที่เท่านั้นไม่จำเป็นต้องจองเวลาไว้ตลอดเวลา จึงทำให้วิธีการใช้งานของ GPRS ในแบบใหม่นี้ จะเห็นได้ว่าจะมีการพูดถึงการเก็บเงินที่เป็นจำนวนข้อมูลที่รับและส่งออกมา มากกว่าวิธีการติดต่อสื่อสารจากวิธีเดิมที่คิดจำนวนเวลาในการติดต่อสื่อสารแต่ละครั้ง

การติดต่อด้วยระบบ GPRS ยังสามารถติดต่อสื่อสารด้วยเสียง ในขณะที่เราสามารถติดต่อสื่อสารผ่านโลกอินเตอร์เน็ตในขณะเดียวกัน ซึ่งก็คือ เราสามารถติดต่อสื่อสารทั้ง 2 ระบบภายในช่วงเวลาเดียวกัน แต่ทั้งนี้ต้องขึ้นอยู่กับโทรศัพท์เคลื่อนที่ในแต่ละรุ่นที่ผลิตออกมา แต่เท่าที่ทราบในขณะนี้โทรศัพท์เคลื่อนที่แต่ละรุ่นยังไม่สามารถติดต่อสื่อสารพร้อม ๆ กันได้ GPRS เชื่อมโลกอินเตอร์เน็ต บนโทรศัพท์เคลื่อนที่ GPRS ไม่ได้เป็นลักษณะที่จะสามารถให้บริการได้ด้วยตัวของระบบเอง แต่ตัวมันเองเป็นเพียงแค่ Bearer ให้กับ Application ต่าง ๆ ที่ต้องการใช้ความเร็วที่เพิ่มมากกว่าปกติในระบบ GSM ที่เคยรองรับอยู่เดิมมาก่อน และระบบ GPRS จะต้องต่อไปยัง Packet Data Network ที่เป็น IP Network อีกต่อหนึ่ง

เป็นบริการส่งข้อมูลสำหรับโทรศัพท์มือถือแบบ GSM GPRS นั้นได้รับการพัฒนาโดย 3GPP (Third Generation Partnership Project) และเป็นการเชื่อมต่อเครือข่ายโทรศัพท์เคลื่อนที่โดยใช้ PDN (Public Data Network) ภายในเครือข่าย GSM โดยจะเป็นการรับส่งข้อมูล Data (ปกติเราใช้ข้อมูลแบบ Voice คือโทรเข้า รับสาย และส่งข้อความ SMS - อันนี้เริ่มมาทาง content เล็กน้อย) โดยจะมีการแบ่งสรรข้อมูลที่เรียกกันว่า PCU (Packet Control Unit), SGSN (Serving GPRS Support Node) และ GGSN (Gateway GPRS support Node) เพื่อจัดการการรับส่งข้อมูลระหว่างเครื่องรับโทรศัพท์และเครือข่ายต้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 แสดงเครือข่ายการทำงานของ GPRS

ดังนั้นผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ ที่จะเปิดใช้ในระบบ GPRS ได้นั้นจะต้องทำการติดตั้งระบบเครือข่าย ที่ประกอบด้วยหน่วยหลัก ๆ 2 หน่วยด้วยกันคือ

- SGSN (Serving GPRS Supports Node)
- GGSN (Gateway GPRS Supports Node)

โดยทั้งสองหน่วยหลักขององค์ประกอบนี้จะถูกเชื่อมต่อเข้าด้วยกันโดยมีอุปกรณ์อื่น ๆ เป็นตัวช่วยเพื่อไปร่วมใช้ Radio Interface จาก Base Station โดยผ่านตัวควบคุม ที่เรียกว่า PCU (Packet Control Unit) ที่ติดตั้งไว้ที่ BSC (Base Station Controller) ทั้งนี้อาจมอง Network เป็นอีก Network หนึ่ง ซึ่งเข้ากับ Mobile Phone ผ่านทาง Radio Interface ของระบบ GSM Network เดิม โดยเป็นบริการที่เกี่ยวเนื่องกับการรับส่งข้อมูลเป็น Packet โดยตรง

คุณสมบัติเด่นๆ ของระบบ GPRS คือ

1. การโอนถ่ายข้อมูลที่มีความสามารถในการ รับ- ส่งผ่านเครือข่ายอินเตอร์เน็ตได้สูงถึง 40 kbps (รวดเร็วขึ้นกว่าระบบ CSD ซึ่งเป็นระบบส่งข้อมูลแบบเก่าที่สามารถส่งได้เพียง 9.6 Kbps ถึง 10 เท่า) ซึ่งจะทำให้สามารถรับ- ส่งข้อมูลที่เป็น VDO Mail หรือ ภาพเคลื่อนไหวต่าง ๆได้ พร้อมทั้งเชื่อมต่อกับเครือข่ายอินเตอร์เน็ตได้มีประสิทธิภาพมากกว่าเดิมรวมถึงการ Down lode/ Up lode ได้ง่ายยิ่งขึ้น

2. Always On นั่นหมายความว่าคุณสามารถ Connect เข้าสู่โลกอินเตอร์เน็ตได้ตลอดเวลา โดยไม่มีความจำเป็นต้องทำการ Log On หรือใช้ User Name และ Password เหมือนการ connect เข้าสู่ อินเตอร์เน็ตทั่วไปแต่อย่างใด ทั้งนี้ยังสามารถที่จะเปลี่ยน Mode มาทำการโทรออกหรือรับสายเข้าโดยคุณยังสามารถ Connect กับโลกอินเตอร์เน็ตในระบบ GPRS อยู่ตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

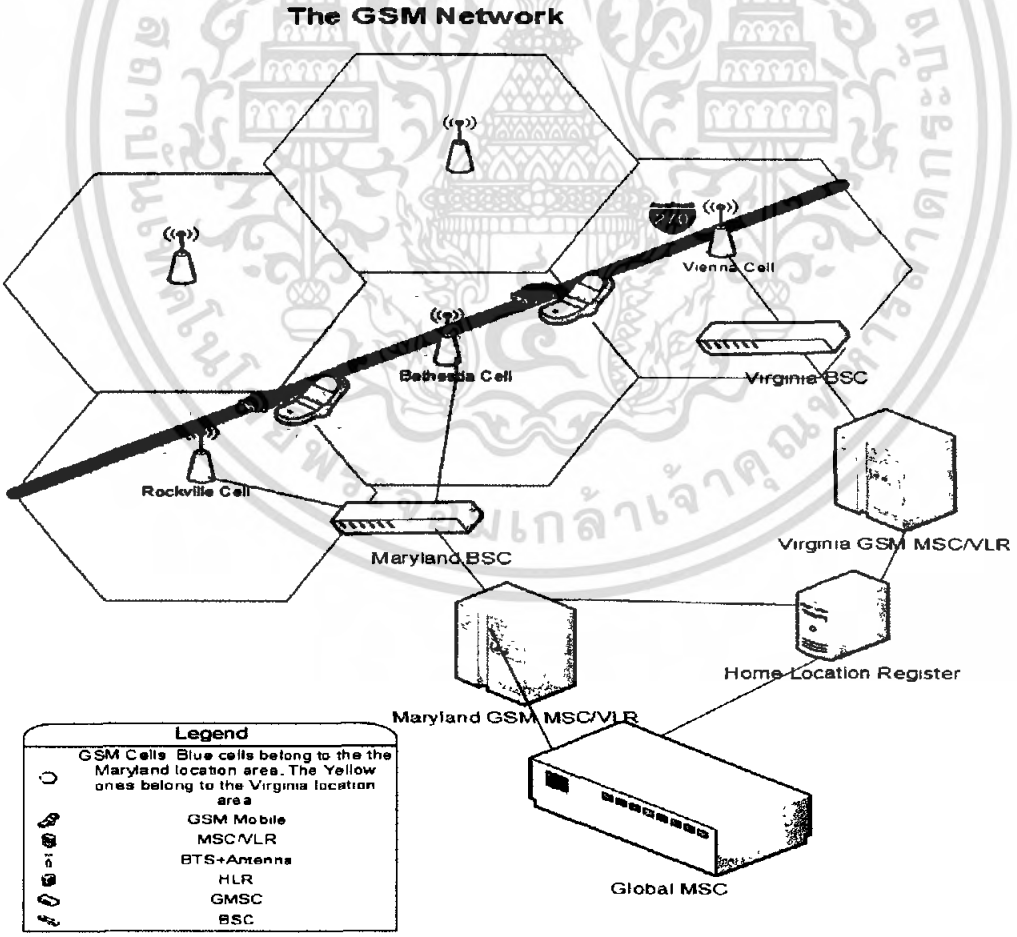
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3. Wireless Internet ที่เชื่อมต่อเข้ากับ Terminal เช่น PDA หรือ Note Book สามารถที่จะโอนถ่ายข้อมูลได้เร็วขึ้นจากที่เคยเป็นอยู่

4. เรื่องค่าบริการของในการต่อเชื่อมกับโลกอินเตอร์เน็ตผ่านเทคโนโลยี GPRS นั้นไม่แพงอย่างที่คิด ทั้งนี้เพราะการคิดค่าบริการในระบบนี้จะเกิดขึ้นก็ต่อเมื่อคุณได้ทำการรับหรือส่งข้อมูลเท่านั้น คุณจะไม่มีค่าใช้จ่ายใดๆในกรณีที่คุณไม่ได้ทำการรับส่งข้อมูล ถึงแม้คุณจะ Connect กับโลกอินเตอร์เน็ตอยู่ก็ตาม

2.2.1 เครือข่าย GSM

GSM ย่อมาจาก Global System for Mobile Communications เป็นมาตรฐานของเทคโนโลยีโทรศัพท์ มือถือที่ได้รับความนิยมมากที่สุดในโลก ปัจจุบันมีผู้ใช้มากกว่า 1.5 พันล้านคนใน 210 ประเทศ GSM เป็นมาตรฐานเปิดภายใต้การดูแลของ 3GPP GSM ใช้เทคโนโลยีดิจิทัลสำหรับช่องสัญญาณควบคุมและสัญญาณเสียงแบบ TDMA ซึ่งแตกต่างจากเทคโนโลยีโทรศัพท์มือถือก่อนหน้านั้น จึงถือว่าเป็นโทรศัพท์มือถือในยุคที่สอง หรือ 2G



รูปที่ 2.19 แสดงเครือข่าย GSM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

83179

2.2.2 การควบคุม GPRS Module

AT Command เป็น Protocol ที่โทรศัพท์เคลื่อนที่ส่วนใหญ่ยึดใช้เป็น Protocol หลัก ในการ Interface กับ คอมพิวเตอร์ซึ่งAT Command เป็นชุดคำสั่งที่มีไว้สำหรับ Modem โดยเฉพาะ โดยคอมพิวเตอร์จะมองโทรศัพท์เคลื่อนที่เป็น Modem เหมือนกัน โดยที่การเชื่อมต่อด้านฮาร์ดแวร์นั้นจะนิยมใช้ 2 ช่องทาง คือ

1. Serial Port
2. IrDA (Infrared Port)

2.3 รูปแบบการเชื่อมต่อ

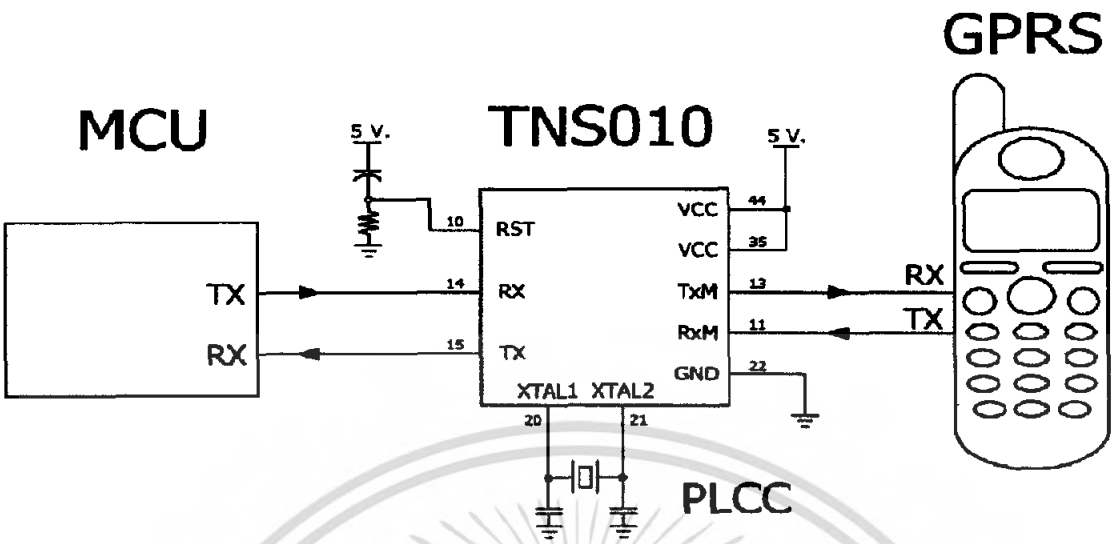
2.3.1 การติดต่อสื่อสารระหว่าง PLC กับ GPRS Module

การติดต่อสื่อสารระหว่าง PLC กับ GPRS Module นั้นใช้การสื่อสารผ่านพอร์ตอนุกรม RS485 ทางด้าน Port 2 ของ PLC และ RS232 ทาง GPRS Module โดยผ่าน Board TCP/IP Stack Chip

2.3.1.1 TCP/IP Stack Chip

หน้าที่ของ Chip ตัวนี้คือ เป็น Core Engine สำหรับการรับ/ส่งข้อมูลบน protocol ต่างๆ ที่ High Level Application ต้องการ ซึ่ง Protocol ที่ Support ก็ได้แก่ TCP/IP, HTTP, DNS, SMTP, PPP แต่จะไม่ทำหน้าที่ในส่วนของการส่งข้อมูลในระดับ Physical Layer ดังนั้นจึงต้องอาศัย Hardware อื่นมาทำหน้าที่นี้ให้ซึ่งก็ดูเหมือนผู้ออกแบบจงใจให้เชื่อมต่อกับ GSM/GPRS Data Module ที่อยู่ในโทรศัพท์มือถือเป็นหลักโดยอาศัย Communication ธรรมดาการเชื่อมต่อกับ CPU หรือ Microcontroller ทำได้ง่าย เพราะเป็น Serial เช่นกัน การใช้งานตัว Chip ก็ง่ายเพราะใช้อุปกรณ์ภายนอกน้อยมาก มีเพียง Xtal 18.432 MHz 1 ตัว กับ C อีก 3 ตัว และ 5V regulator อีก 1 ตัว เท่านั้น ส่วนการสั่งงานตัว Chip ก็ไม่ลำบาก เพราะใช้เป็น AT Command เหมือน Modem

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงการทำงานของ TCP/IP Stack Chip

2.3.1.2 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมแบ่งได้เป็น 2 แบบคือ การสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกามีข้อมูลและ

2.3.1.3 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้การกำหนดค่าอัตราเร็วในการรับข้อมูลและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราเร็วบอดหรือบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที (Bit per second)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 5, 6, 7 หรือ 2 บิต
3. บิตตรวจสอบพาริตี้ (Parity) มีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้ายหรือบิตหยุด (Stop bit) มีขนาด 1, 1.5 หรือ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัสคือเมื่อไม่มีการส่งข้อมูล ขา DATA จะมีสถานะลอจิก "1" เรียกสถานะนี้ว่า สถานะรอหยุด (Waiting Stage) การเริ่มต้นการส่งข้อมูลจะเริ่มจากการ ให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียก 1 บิตนี้ว่าบิตเริ่มต้น (Start Bit) จากนั้นบิตข้อมูลจะถูกส่งออกไปเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อนซึ่งข้อมูลที่ต้องการจะถูกส่งอาจมีจำนวน 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นตามด้วยบิตพาริตี้ (Parity Bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล อัตราความเร็วในการรับส่งข้อมูลแบบอะซิงโครนัสหรืออัตราบอดหรือบอดเรตที่ใช้สำหรับพอร์ทอนุกรม RS-232 มีด้วยกันหลายค่า ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที โดยมีค่าเพิ่มขึ้นตามเทคโนโลยีคอมพิวเตอร์ เนื่องจากบอดเรต คือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมุติว่าข้อมูลอนุกรมมีขนาด 8 บิต ไม่มีการตรวจสอบพาริตี้มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวข้อมูล 1 ไบต์จะมีความยาวเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็สามารถรับส่งข้อมูล ได้ด้วยความเร็ว

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (Odd) แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ พาริตี้คี่หรือพาริตี้คู่แสดงถึงจำนวนลอจิก "1" ทั้งหมดภายในที่ส่งไป 1 ไบต์ รวมบิตพาริตี้ว่าจำนวนเลขคู่หรือเลขคี่ ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ 10011001B จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้น ถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าของพาริตี้จะต้องมีลอจิกเป็น "0" แต่ถ้ากำหนดพาริตี้เป็นคี่ ค่าของพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งพาริตี้เป็นคี่

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ไอซี UART เบอร์16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ไอซี UART เบอร์ 8250 ไอซี UART เหล่านี้มีระดับแรงดันของลอจิกเป็นแบบทีทีแอล (+5V) แต่เพื่อให้มีแรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ที่ระยะทางไกลมากขึ้น ระดับแรงดันทีทีแอลจะถูกดัดแปลงเป็นระดับแรงดันที่สูงขึ้นโดยลอจิก "0" จะมีระดับแรงดัน -3V ถึง -12V และลอจิก "1" มีระดับแรงดัน +3V ถึง +12V

2.3.1.4 มาตรฐานพอร์ทอนุกรมแบบ RS -232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้ส่งผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronics Industries Association: EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3V ถึง -12V แสดงว่ามีข้อมูล(mark) และ +3V ถึง +12V แสดงว่าเป็นช่องว่าง(space)

มาตรฐาน RS-232 ถูกใช้ในการกำหนดรูปแบบการสื่อสารข้อมูลกันระหว่างอุปกรณ์เชื่อมต่อข้อมูล (Data terminal: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่ประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ส่วนอุปกรณ์ DCE ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ทอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานในคอมพิวเตอร์ พอร์ทอนุกรม RS-232 ถูกใช้เพื่อเชื่อมกับโมเด็ม, เมาส์ และเครื่องพิมพ์ที่สามารถติดต่อทางพอร์ทอนุกรมได้

2.3.1.5 คอนเน็กเตอร์สำหรับพอร์ท RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อการใช้งาน เพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยมีการใช้งานมาในอดีตไม่ค่อยมีความสำคัญมากนักจึงถูกยกเลิกไป

1. ขา Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาร์ทจากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

2. ขา Receive Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยจะนำข้อมูลที่อ่านไปเก็บไว้ในรีจิสเตอร์

3. ขา Transmitted: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์ โดยการนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

4. ขา Data Terminal Read DTR เป็นขาเอาท์พุทที่ใช้สำหรับส่งสัญญาณออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อกับอุปกรณ์ปลายทาง โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์และถ้าใช้การเชื่อมต่อแบบ 3 สาย ต้องเชื่อมต่อขา DTR และ DSR ของพอร์ทอนุกรมเข้าด้วยกัน และจะต้องต่อเชื่อมกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาร์

5. ขา Signal Ground: GND เป็นขากราวด์ของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ขา Data Set Read: DSR ขานี้จะใช้ควบคู่ DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับข้อมูลจากภายนอก

7. ขา Request To Send: RST เป็นขาเอาท์พุทสำหรับส่งสัญญาณร้องขอให้กับอุปกรณ์ปลายทางส่งข้อมูลมาให้กับคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ซึ่งในกรณีที่มีการเชื่อมต่อแบบ 3 สายจะต้องเชื่อมต่อขา RTS และ CTS เข้าด้วยกัน เพื่อให้การรับส่งข้อมูลเกิดขึ้นได้ตลอดเวลา

8. ขา Clear To send: CTS เป็นขาอินพุททำหน้าที่รอรับสัญญาณที่ส่งเข้ามาเมื่อมีการส่งสัญญาณเข้ามาที่ขานี้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ขานี้ใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่รับข้อมูลแล้วหรือยัง

9. ขา Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มแล้วยังมีความต้องการตรวจสอบสัญญาณเรียกจากสายโทรศัพท์

2.3.2 การ GPRS Module ติดต่อ Server

2.3.2.1 Internet

อินเตอร์เน็ต (Internet) คือ เครือข่ายของเครือข่ายคอมพิวเตอร์ ระบบต่าง ๆ ที่เชื่อมโยงกัน มาจากคำว่า Inter Connection Network อินเตอร์เน็ต (Internet) เป็นระบบเครือข่ายคอมพิวเตอร์ ที่มีขนาดใหญ่ เครื่องคอมพิวเตอร์ทุกเครื่องทั่วโลก สามารถติดต่อสื่อสารถึงกัน ได้โดยใช้มาตรฐาน ในการรับส่งข้อมูลที่เป็นหนึ่งเดียว หรือที่เรียกว่าโปรโตคอล (Protocol) ซึ่งโปรโตคอล ที่ใช้บนระบบเครือข่ายอินเตอร์เน็ตมีชื่อว่าทีซีพี/ไอพี (TCP/IP :Transmission Control Protocol/Internet Protocol) ลักษณะของระบบอินเตอร์เน็ต เป็นเสมือนใยแมงมุม ที่ครอบคลุมทั่วโลกในแต่ละจุดที่เชื่อมต่ออินเตอร์เน็ตนั้น สามารถสื่อสารกันได้หลายเส้นทางตามความต้องการ โดยไม่กำหนดตายตัว และไม่จำเป็นต้องไปตามเส้นทางโดยตรง อาจจะผ่านจุดอื่น ๆ หรือ เลือกไปเส้นทางอื่นได้หลาย ๆ เส้นทางการติดต่อสื่อสารผ่านระบบเครือข่ายอินเตอร์เน็ตนั้นอาจเรียกว่าการติดต่อสื่อสารแบบไร้มิติ หรือ Cyberspace

อินเตอร์เน็ตมีพัฒนาการมาจาก อาร์พาเน็ต (ARPAnet เรียกสั้นๆ ว่า อาร์พา) ที่ตั้งขึ้นในปี 2512 เป็นเครือข่ายคอมพิวเตอร์ของกระทรวงกลาโหม สหรัฐอเมริกา ที่ใช้ในงานวิจัยด้านทหาร (ARPA : Advanced Research Project Agency)

มาถึงปี 2515 หลังจากที่เครือข่ายทดลองอาร์พาประสบความสำเร็จอย่างสูง และได้มีการปรับปรุงหน่วยงานจากอาร์พาเป็นดาห์พา (Defense Advanced Research Project Agency : DARPA) และในปี 2518 อาร์พาเน็ตก็ขึ้นตรงกับหน่วยการสื่อสารของกองทัพ (Defense Communication Agency)

เอกสารนี้มีลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DARPA) และในปี 2518 อาร์พาเน็ตก็ขึ้นตรงกับหน่วยการสื่อสารของกองทัพ (Defense Communication Agency)

ในปี 2526 อาร์พาเน็ตได้แบ่งออกเป็น 2 เครือข่าย คือเครือข่ายด้านงานวิจัยใช้ชื่ออาร์พาเน็ตเหมือนเดิม ส่วนเครือข่ายของกองทัพใช้ชื่อว่า มิลเน็ต (MILNET : Military Network) มีการเชื่อมต่อโดยใช้ โพรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นครั้งแรก

ในปี 2528 มูลนิธิวิทยาศาสตร์แห่งชาติของอเมริกา (NSF) ได้ให้เงินทุนในการสร้างศูนย์ ซุปเปอร์คอมพิวเตอร์ 6 แห่ง และใช้ชื่อว่า NSFNET และในปี 2533 อาร์พาเน็ตรองรับภาระที่เป็นกระดูกสันหลัง (Backbone) ของระบบไม่ได้ จึงได้ยุติอาร์พาเน็ต และเปลี่ยนไปใช้ NSFNET และเครือข่ายอื่นๆแทน และได้มีกรเชื่อมต่อร่วมกับเครือข่ายอื่นๆ จนมาเป็นเครือข่ายขนาดมหึมา จนถึงทุกวันนี้ และเรียกเครือข่ายนี้ว่า อินเตอร์เน็ต โดยเครือข่ายส่วนใหญ่จะอยู่ในอเมริกา และปัจจุบันนี้มีเครือข่ายย่อยมากถึง 50,000 เครือข่ายทีเดียว และคาดว่า ภายในปี 2543 จะมีผู้ใช้อินเตอร์เน็ตทั้งโลกประมาณ 100 ล้านคน หรือใกล้เคียงกับประชากรโลกทั้งหมด

สำหรับประเทศไทยนั้น อินเตอร์เน็ตเริ่มมีบทบาทอย่างมากในช่วงปี 2530-2535 โดยเริ่มจากการเป็นเครือข่ายในระบบคอมพิวเตอร์ระดับมหาวิทยาลัย (Campus Network) แล้วจึงเชื่อมต่อเข้าสู่อินเตอร์เน็ตอย่างสมบูรณ์เมื่อเดือนสิงหาคม 2538 ก็มีการเปิดให้บริการอินเตอร์เน็ตในเชิงพาณิชย์ (รายแรกคือ อินเตอร์เน็ตเคเอสซี) ซึ่งขณะนั้น เวิลด์ไวด์เว็บกำลังได้รับความนิยมอย่างมากในอเมริกา อย่างไรก็ตาม อินเตอร์เน็ต บางครั้งก็มีการเรียกย่อยเป็น เน็ต (Net) หรือ The Net ด้วยเช่นเดียวกัน อีกคำหนึ่งที่หมายถึงอินเตอร์เน็ตก็คือ เว็บ (Web) และ เวิลด์ไวด์เว็บ (WWW)

2.3.2.2 TCP/IP Protocol

โปรโตคอล TCP/IP หรือ Transmission Control Protocol/Internet Protocol เป็นระเบียบวิธีการ สื่อสารระหว่างคอมพิวเตอร์ ที่ใช้กันมาแต่เดิมในระบบปฏิบัติการ Unix ซึ่งมีการใช้งานอย่าง กว้างขวางมาก จนถือเป็นมาตรฐานได้ จุดกำเนิดของโปรโตคอล TCP/IP นี้เริ่มขึ้นในราว พ.ศ. 2512 ที่กระทรวงกลาโหมของสหรัฐ เมื่อพบปัญหาในการเชื่อมโยงเครือข่ายคอมพิวเตอร์ในหน่วยงานต่างๆ ของตน ซึ่งจะต้องมีการส่งข้อมูลระหว่างกัน และไปยังหน่วยงานภายนอกอื่นๆ เช่น มหาวิทยาลัย ห้องทดลองต่างๆ (ส่วนใหญ่มีเครื่องที่ใช้ระบบ Unix อยู่เป็นจำนวนมาก) เนื่องจากแต่ละแห่งก็จะมีระบบคอมพิวเตอร์ของตนเองที่แตกต่างกันไป การต่อเชื่อมกันก็เป็นไปในลักษณะต่างคนต่างทำไม เหมือนกัน ดังนั้นข่าวสารข้อมูลทั้งหลาย จึงถ่ายเทไปมาได้อย่างยากลำบากมาก กระทรวงกลาโหมสหรัฐได้ จัดตั้งหน่วยงาน Advanced Research Projects Agencies (ARPA) ขึ้นมา เพื่อหาทางแก้ไขปัญหาที่เกิดขึ้นนี้ ผลลัพธ์ที่หน่วยงาน ARPA ได้จัดทำขึ้นคือ การกำหนดมาตรฐานในการสื่อสารข้อมูลและได้จัดตั้งเครือข่าย ARPANET ขึ้นโดยใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล TCP/IP ต่อมาก็กลายมาเป็นมาตรฐานจริงจัง ในราวปี พ.ศ. 2525 ความสัมพันธ์ระหว่าง TCP/IP กับระบบปฏิบัติการ Unix เกิดขึ้น เนื่องจากมหาวิทยาลัยแคลิฟอร์เนีย ที่เบอร์คเลย์ ได้พัฒนาระบบปฏิบัติการ Unix ซึ่งมีการผนวกเข้ากับโปรโตคอล TCP/IP สำหรับใช้ในการสื่อสารระหว่างระบบออกมา และเผยแพร่ต่อไปยังหน่วยงานต่างๆ ทำให้การสื่อสารกันของเครื่องที่ใช้ระบบปฏิบัติการ Unix มักจะต้อง ใช้โปรโตคอล TCP/IP เสมอ และมีบทบาทเป็นสิ่งที่คู่กันต่อมาถึงปัจจุบัน

ในปัจจุบันนี้ ไม่ว่าคอมพิวเตอร์ส่วนบุคคลของผู้ใช้จะเป็นแบบใดก็ตาม เช่น พีซี หรือแมคอินทอช ก็สามารถใช้งานโปรโตคอล TCP/IP เพื่อต่อเชื่อมเข้าสู่อินเตอร์เน็ตได้ วิธีการก็คือเพียงแต่ติดตั้งใช้งานซอฟต์แวร์โปรโตคอล TCP/IP เท่านั้น ส่วนวิธีการและโปรแกรมที่ติดตั้งจะแตกต่างกันขึ้นกับระบบที่ใช้ ซึ่งจะกล่าวต่อไป หมายเลข IP (IP Address) การสื่อสารกันในระบบเครือข่าย อินเตอร์เน็ตที่มีโปรโตคอล TCP/IP เป็นมาตรฐานนี้ เครื่องอมพิวเตอร์ทุกเครื่องที่เชื่อมต่ออยู่ จะต้องมีหมายเลขประจำตัวเอาไว้อ้างอิงให้เครื่องคอมพิวเตอร์อื่นๆ ได้ทราบ เหมือนกับคนทุกคนต้องมีชื่อให้คนอื่นเรียก หมายเลขอ้างอิงดังกล่าวเราเรียกว่า IP Address หรือหมายเลข IP หรือบางทีก็เรียกว่า "แอดเดรส IP" (IP ในที่นี้ก็คือ Internet Protocol ตัวเดียวกับใน TCP/IP นั่นเอง) ซึ่งถูกจัดเป็นตัวเลขชุดหนึ่งขนาด 32 บิต ใน 1 ชุดนี้จะมีตัวเลขถูกแบ่งออกเป็น 4 ส่วน ส่วนละ 8 บิตเท่าๆ กัน เวลาเขียนก็แปลงให้เป็นเลขฐานสิบ ก่อนเพื่อความง่ายแล้วเขียนโดยคั่นแต่ละส่วนด้วยจุด ดังนั้นในตัวเลขแต่ละส่วนนี้จึงมีค่าได้ตั้งแต่ 0 จนถึง 28 -1 = 255 เท่านั้น เช่น 192.10.1.101 เป็นต้น ตัวเลข IP Address ชุดนี้จะเป็นสิ่งที่สำคัญคล้ายเบอร์โทรศัพท์ที่เรามีใช้อยู่และไม่ซ้ำกัน เพราะสามารถกำหนดเป็นตัวเลขได้รวมทั้งสิ้นกว่า 4 พันล้านเลขหมาย แต่การกำหนดให้คอมพิวเตอร์มีเลขหมาย IP Address นี้ไม่ได้เริ่มต้นจากหมายเลข 1 และนับขึ้นไปเรื่อยๆ หากแต่จะมีการจัด แบ่งออกเป็น 2 ส่วน คือ

ส่วนแรกเป็นหมายเลขของเครือข่าย (Network Number)

ส่วนที่สองเรียกว่าหมายเลขของคอมพิวเตอร์ที่อยู่ในเครือข่ายนั้น (Host Number) เพราะในเครือข่ายใดๆ อาจจะมีเครื่องคอมพิวเตอร์เชื่อมต่ออยู่ได้มากมาย ในเครือข่ายที่อยู่คนละระบบอาจมีหมายเลข Host ซ้ำกันก็ได้ แต่เมื่อรวมกับหมายเลข Network แล้ว จะได้เป็น IP Address ที่ไม่ซ้ำกันเลย

ในการจัดตั้งหรือกำหนดหมายเลข IP Address นี้ก็มีวิธีการกำหนดที่ชัดเจน และมีกฎเกณฑ์ที่รัดกุม ผู้ใช้ที่อยากจัดตั้งโฮสต์คอมพิวเตอร์ เพื่อเชื่อมต่อเข้าอินเตอร์เน็ต และให้บริการต่างๆ สามารถขอหมายเลข IP Address ได้ที่หน่วยงาน Internet Network Information Center (InterNIC) ขององค์กร Network Solution Incorporated (NSI) ที่รัฐเวอร์จิเนีย สหรัฐอเมริกา แต่ถ้าผู้ใช้สมัครเข้าเป็นสมาชิกขอ ใช้บริการอินเตอร์เน็ตจากบริษัทผู้ให้บริการ (Internet Service Provider) เรียกย่อๆ ว่าหน่วยงาน ISPรายใดก็แล้วแต่ ก็ไม่ต้องติดต่อขอ IP Address เนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยงาน ISP เหล่านั้นจะกำหนดหมายเลข IP ให้ใช้ หรือส่งค่า IP ชั่วคราวให้ใช้งาน ทั้งนี้ขึ้นอยู่กับแบบการขอใช้บริการที่จะกล่าวต่อไป

โครงสร้างของแอดเดรสที่ใช้ใน classต่างๆของเครือข่าย ซึ่งทั้งหมด ยาว 32 บิต IP Address นี้มีการจัดแบ่งออกเป็นทั้งหมด 5 ระดับ (Class) แต่ที่ใช้งานในทั่วไปจะมีเพียง 3 ระดับ คือ Class A, Class B, Class C ซึ่งก็แบ่งตามขนาดความใหญ่ ของเครือข่ายนั่นเอง ถ้าเครือข่ายใดมีจำนวนเครื่องคอมพิวเตอร เชื่อมต่ออยู่มาก ก็จะมีหมายเลขอยู่ใน Class A ถ้ามีจำนวนเครื่องต่ออยู่ลดหลั่นกันลงมาก็จะอยู่ใน Class B และ Class C ตามลำดับ หมายเลข IP ของ Class A มีตัวแรกเป็น 0 และหมายเลขของเครือข่าย (Network Number) ขนาด 7 บิต และ มีหมายเลขของเครื่องคอมพิวเตอร์ (Host Number) ขนาด 24 บิต ทำให้ในหนึ่งเครือข่ายของ Class A สามารถมีคอมพิวเตอร์เชื่อมต่ออยู่ในเครือข่ายได้ถึง 224= 16 ล้านเครื่อง เหมาะสำหรับองค์กร หรือบริษัทยักษ์ใหญ่ แต่ใน Class A นี้ จะมีหมายเลข เครือข่ายได้ 128 ตัวเท่านั้นทั่วโลก ซึ่งหมายความว่าจะมีเครือข่ายยักษ์ใหญ่แบบนี้ได้เพียง 128 เครือข่ายเท่านั้น สำหรับ Class B จะมีหมายเลขเครือข่ายแบบ 14 บิต และหมายเลขเครื่องคอมพิวเตอร์แบบ 16 บิต (ส่วนอีก 2 บิตที่เหลือบังคับว่าต้องขึ้นต้นด้วย 102) ดังนั้นจึงสามารถมีจำนวนเครือข่ายที่อยู่ใน Class B ได้มากกว่า Class A คือมีได้ถึง 214 = กว่า 16,000 เครือข่าย และก็สามารถมีเครื่องคอมพิวเตอร์เชื่อมต่อกันในเครือข่าย Class B แต่ละเครือข่ายได้ถึง 216 หรือมากกว่า 65,000 เครื่อง สุดท้ายคือ Class C ซึ่งมีหมายเลขเครื่องคอมพิวเตอร์แบบ 8 บิตและมีหมายเลขเครือข่ายแบบ 21 บิต ส่วนสามบิตแรกบังคับว่าต้องเป็น 1102 ดังนั้นใน แต่ละเครือข่าย Class C จะมีจำนวนเครื่องต่อเชื่อมได้เพียงไม่เกิน 254 เครื่องในแต่ละเครือข่าย (28 = 256 แต่หมายเลข 0 และ 255 จะไม่ถูกใช้งาน จึงเหลือเพียง 254) ดังนั้นวิธีการสังเกตได้ง่ายๆ ว่าเราเชื่อมต่ออยู่ที่เครือข่าย Class ใดก็สามารถดูได้จาก IP Address ในส่วนหน้า (ส่วน Network Address) โดย

Class A จะมี Network address ตั้งแต่ 0 ถึง 127 (บิตแรกเป็น 0 เสมอ)

Class B จะมี Network address ตั้งแต่ 128 ถึง 191 (เพราะขึ้นต้นด้วย 102 เท่านั้น)

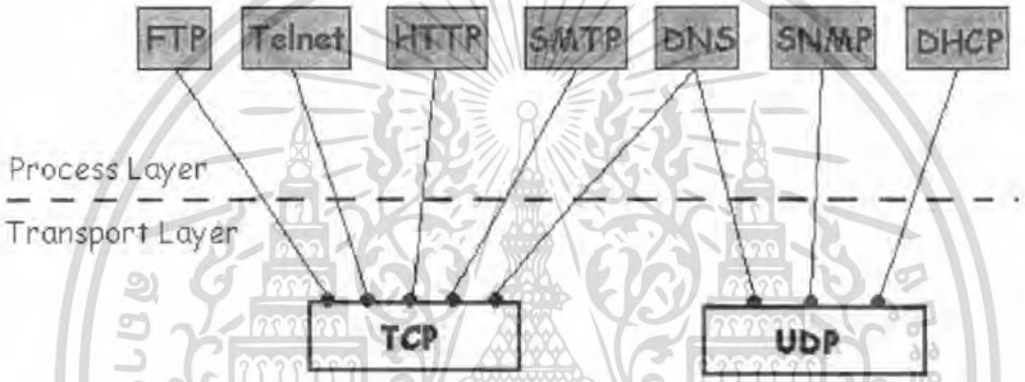
Class C จะมี Network address ตั้งแต่ 192 ถึง 223 (เพราะขึ้นต้นด้วย 1102 เท่านั้น)

เช่น ถ้าเครื่องคอมพิวเตอร์ในอินเตอร์เน็ตมีหมายเลข IP ดังนี้ 181.11.82.22 ตัวเลข 181.11 แสดงว่าเป็นเครือข่ายใน Class B ซึ่งหมายเลขเครือข่ายเต็มๆ จะใช้ 2 ส่วนแรกคือ 181.11 และมีหมายเลขคอมพิวเตอร์คือ 82.22 หรือถ้ามี IP Address เป็น 192.131.10.101 ทำให้ทราบว่าเครื่องคอมพิวเตอร์นั้นเชื่อมต่ออยู่ใน Class C มีหมายเลขเครือข่ายคือ 3 ส่วนแรก ได้แก่ 192.131.10 และหมายเลขประจำ เครื่องคือ 101 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.3 UDP (User Datagram Protocol)

ใน Host-to-Host layer นอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ก็ยังมีโปรโตคอล UDP (User Datagram Protocol) ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลเช่นนี้แอพพลิเคชันหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง



รูปที่ 2.21 แสดงการแบ่งการทำงานของโปรโตคอล TCP และ UDP

ตามรูป จะเห็นว่าโปรโตคอลชั้นบนขึ้นไป ที่ใช้การส่งผ่านข้อมูลโดยโปรโตคอล UDP เช่นโปรโตคอล SNMP (ใช้ควบคุมและจัดการอุปกรณ์ในเครือข่าย) , หรือโปรโตคอล DHCP (ใช้ส่งข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายได้ใช้งาน) การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูล ที่มีการรับส่ง จะไปทำในขั้นตอนของโปรโตคอลชั้นที่สูงกว่าแทน

ตัวอย่างขั้นตอนกลไกการทำงานโดยใช้โปรโตคอล UDP

1. ในขั้นตอนของ Process layer เมื่อ"ปรแกรมควบคุมอุปกรณ์เครือข่ายเช่น โปรแกรม Network management ต้องส่งข้อมูลไปยังอุปกรณ์ที่ต้องการ แอพพลิเคชันนั้นจะติดต่อผ่านโปรโตคอล SNMP ในชั้น Process layer

2. โปรโตคอล SNMP จะติดต่อกับโปรโตคอล UDP ในชั้นถัดไป เพื่อขอติดต่อผ่าน Port ที่กำหนด

3. โปรโตคอล SNMP เตรียมข้อมูลที่จะส่ง รวมทั้งที่อยู่ปลายทาง

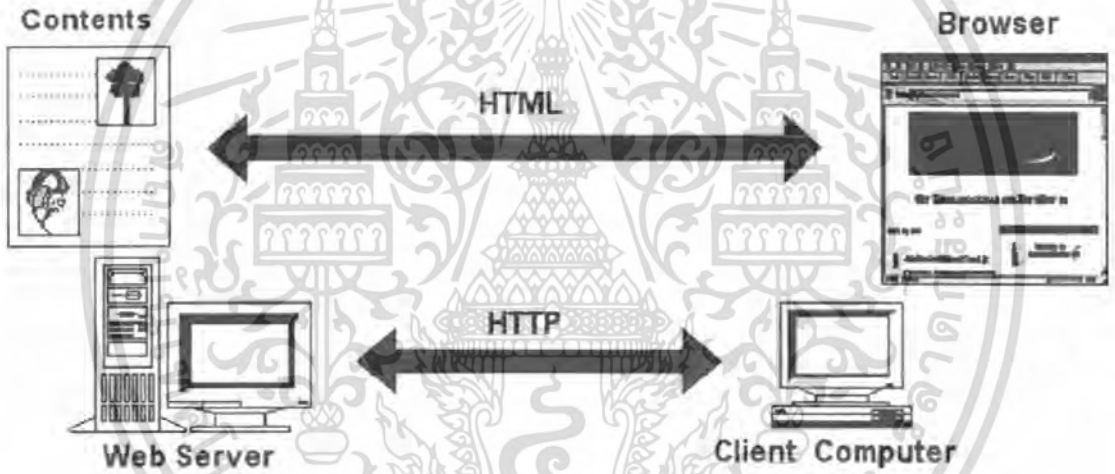
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โปรโตคอล SNMP ส่งผ่านข้อมูลให้โปรโตคอล UDP ที่อยู่ในชั้น Transport

5. โปรโตคอล UDP ทำหน้าที่ผนึกข้อมูลหรือ datagram นั้น ไปกับโปรโตคอล IP ในชั้นถัดไป เพื่อส่งข้อมูลออกจากเครื่อง

2.3.3 การติดต่อ Server/Web browser

Hypertext Transfer Protocol (HTTP) และHypertext Markup Language (HTML) HTTP คือโปรโตคอลที่ใช้สื่อสารระหว่าง client computer กับ server computer ทำให้ทั้งสองเครื่องรู้ว่าจะจัดการส่งข้อมูลไปอย่างไร ส่วน HTML คือสื่อภาษาที่ทำให้เอกสารหรือ contents ที่อยู่บนเครื่อง server computer เมื่อถูกส่งมาที่ client computer แล้วจะนำไปแสดงได้อย่างไร เราเรียกซอฟท์แวร์ที่ใช้แสดงนี้ว่า Browser



รูปที่ 2.22 แสดงความแตกต่าง HTML และ HTTP

ข้อดีของการแยกชั้นการทำงานระหว่าง HTTP กับ HTML

Contents

พัฒนาบนเครื่องแบบใดก็ได้ เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray etc.

มีเครื่องมือช่วยในการพัฒนามากมาย

Web Server

เครื่องที่ใช้เป็น Web Server เป็นเครื่องใดๆ ก็ได้ เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray

ในแต่ละ Platform มี โปรแกรม Web Server ให้เลือกมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

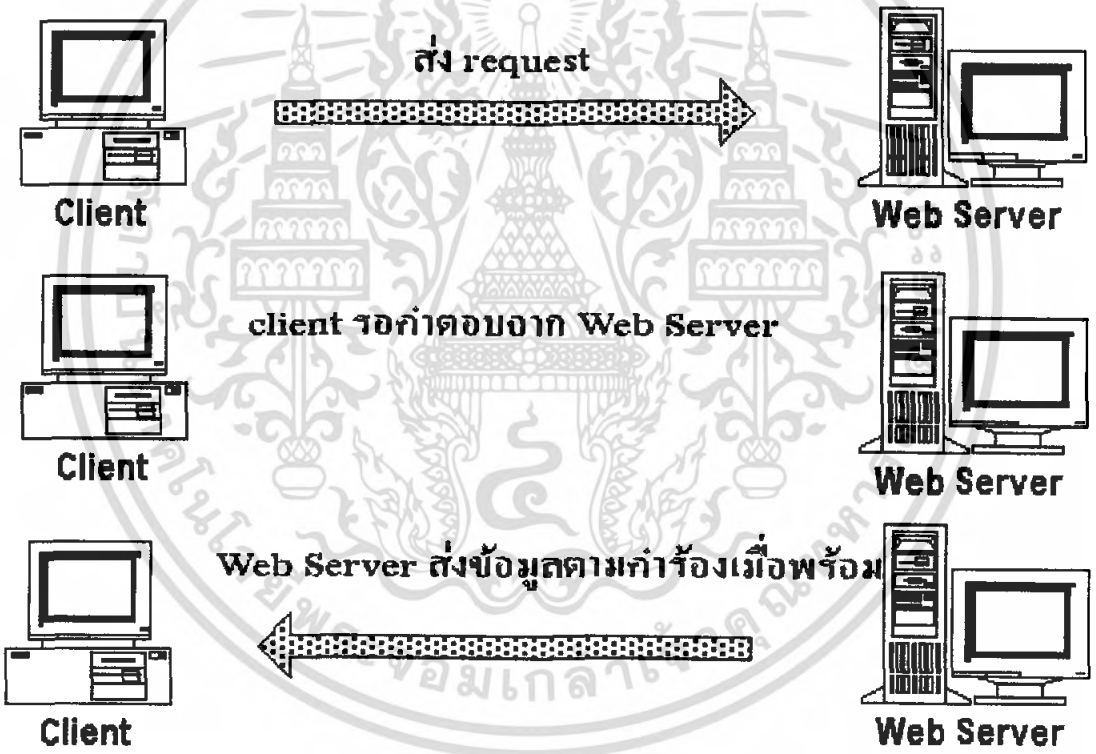
Client Computer

เครื่องที่ใช้เป็น Client Computer เป็นเครื่องใดๆ ก็ได้ เช่น PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray, TV with Set-Top Box, Pen Computer etc.

Browser

โปรแกรม Browser มีให้เลือกใช้มากมายบน PC, Macintosh, IBM, DEC, SUN, HP, SGI, Cray, TV with Set-Top Box, Pen Computer etc.

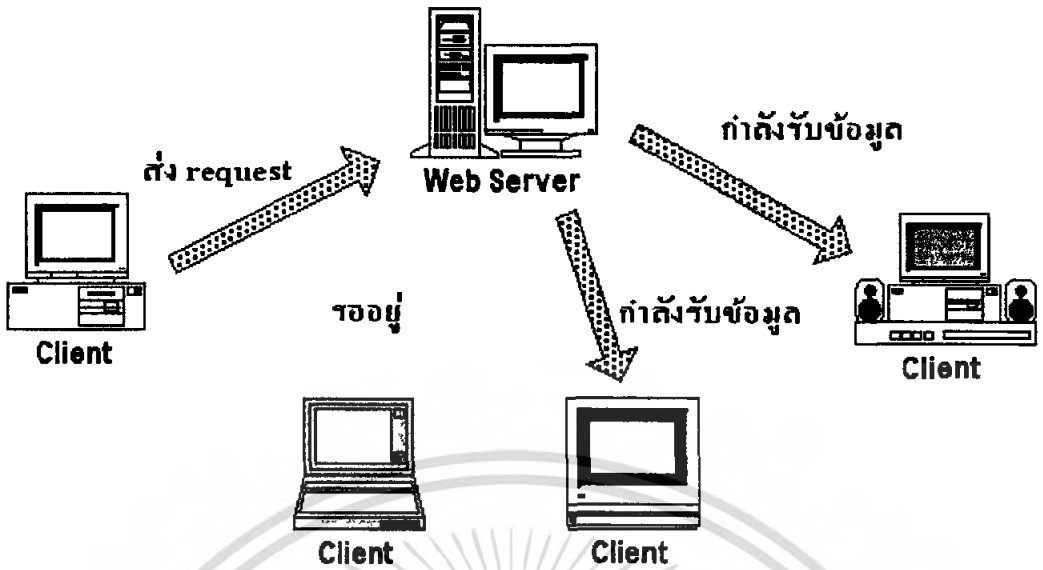
โปรโตคอล HTTP นี้วิ่งอยู่บน TCP/IP อีกชั้นหนึ่ง รูปแบบการทำงานจะไม่มีการจองสาย โดย Client จะเรียกข้อมูลจาก Server โดยการส่ง Request ไปแล้วจะตัดการติดต่อทันที จากนั้นจะรอจนกระทั่ง Server ส่งข้อมูลมาให้



รูปที่ 2.23 แสดงขั้นตอนการติดต่อ ของโปรโตคอล HTTP

ประโยชน์ของการทำงานแบบไม่จองสายของ HTTP ทำให้ WWW Server สามารถให้บริการ Client ได้หลายๆ คนพร้อมๆ กัน การสื่อสารของ WWW จึงมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 แสดงการทำงานแบบไม่จองสายของ HTTP

2.4 Application Server

2.4.1 APACHE

สำหรับเปิดให้บริการเซิร์ฟเวอร์บนโปรโตคอล HTTP โดยสามารถทำงานได้บนหลายระบบปฏิบัติการ ที่มาของชื่อ apache มาจากชื่อชนเผ่าอินเดียแดงพื้นเมืองของทวีปอเมริกา

2.4.1.1 ประวัติ

Apache พัฒนามาจาก HTTPD Web Server ที่มีกลุ่มผู้พัฒนาอยู่ก่อนแล้ว โดย ร็อบ แม็คคูล(Rob McCool) ที่ NCSA (National Center for Supercomputing Applications) มหาวิทยาลัยอิลลินอยส์ เออร์แบนา-แชมเปญจน์ สหรัฐอเมริกา แต่หลังจากที่ แม็คคูล ออกจาก NCS และหันไปให้ความสนใจกับโครงการอื่นๆ มากกว่าทำให้ HTTPD เว็บเซิร์ฟเวอร์ ถูกปล่อยทิ้งไม่มีผู้พัฒนาต่อ แต่เนื่องจากเป็นซอร์ฟแวร์ที่อยู่ภายใต้ลิขสิทธิ์ กนู คือ ทุกคนมีสิทธิ์ที่จะนำเอาซอร์สโค้ดไปพัฒนาต่อได้ ทำให้มีผู้ใช้กลุ่มหนึ่งได้พัฒนาโปรแกรมขึ้นมาเพื่ออุดช่องโหว่ ที่มีอยู่เดิม (หรือ แพช) และยังได้รวบรวมเอาข้อมูลการพัฒนา และการแก้ไขต่างๆ แต่ข้อมูลเหล่านี้อยู่ตามที่ต่างๆ ไม่ได้รวมอยู่ในที่ที่เดียวกัน จนในที่สุด ไบอัน บีเลนดอร์ฟ (Brian Behlendorf) ได้สร้างจดหมายกลุ่ม (mailing list) ขึ้นมาเพื่อนำเอาข้อมูลเหล่านี้เข้าไว้เป็นกลุ่มเดียวกัน เพื่อให้สามารถเข้าถึงข้อมูลเหล่านี้ได้ง่ายยิ่งขึ้น และในที่สุด กลุ่มผู้พัฒนาได้เรียกตัวเองว่า กลุ่มอะแพชี (Apache Group) และได้ปล่อยซอฟต์แวร์ HTTPD เว็บเซิร์ฟเวอร์ ที่พัฒนาโดยการนำเอาแพชหลายๆ ตัวที่ผู้ใช้ได้พัฒนาขึ้นเพื่อปรับปรุงการทำงาน ของซอฟ์ตแวร์ตัวเดิมให้มีประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตั้งแต่ ปี พ.ศ. 2539 Apache ได้รับความนิยมขึ้นเรื่อยๆ จนปัจจุบันได้รับความนิยมเป็นอันดับหนึ่ง มีผู้ใช้งาน อยู่ประมาณ 65% ของเว็บเซิร์ฟเวอร์ที่ให้บริการอยู่ทั้งหมด

2.4.1.2 ความสามารถ

การที่อาปาเช่เป็นซอฟต์แวร์ที่อยู่ในลักษณะของ โอเพ่นซอร์ส ที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาส่วนต่างๆ ของอาปาเช่ได้ ซึ่งทำให้เกิดเป็น โมดูล ที่เกิดประโยชน์มากมาย เช่น mod_perl, mod_python หรือ mod_php ซึ่งเป็นโมดูลที่ทำให้อาปาเช่สามารถใช้ประโยชน์ และทำงานร่วมกับภาษาอื่นได้ แทนที่จะเป็นเพียงเซิร์ฟเวอร์ที่ให้บริการเพียงแค่เอชทีเอ็มแอล อย่างเดียว นอกจากนี้อาปาเช่เองยังมีความสามารถอื่นๆ ด้วย เช่น การยืนยันตัวบุคคล (mod_auth, mod_access, mod_digest) หรือเพิ่มความปลอดภัยในการสื่อสารผ่าน โปรโตคอล https (mod_ssl) นอกจากนี้ ก็ยังมีโมดูลอื่นๆ ที่ได้รับความนิยมใช้ เช่น mod_vhost ทำให้สามารถสร้างโฮสท์เสมือน www.sample.com, wiki.sample.com, mail.sample.com หรือ www.ilovewiki.org ภายในเครื่องเดียวกันได้ หรือ mod_rewrite เป็นเครื่องมือที่จะช่วยให้ url ของเว็บนั้นอ่านง่ายขึ้น ยกตัวอย่างเช่น จากเดิมต้องอ้างถึงเว็บไซต์แห่งหนึ่งด้วยการพิมพ แต่หลังจากใช้ mod_rewrite จะทำให้สั้นลง กลายเป็น http://www.youdomain.com/board/question/2xDffw ซึ่งที่อยู่หลังนี้จะขึ้นอยู่กับว่าผู้ดูแลเว็บไซต์ต้องการให้อยู่ในลักษณะใด

2.4.2 PHP

ภาษาพีเอชพี (PHP) คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีความตอบโต้ได้อย่างรวดเร็ว

2.4.2.1 คุณสมบัติ

การแสดงผลของพีเอชพี จะปรากฏในลักษณะ HTML ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่าน ดูและคัดลอกคำสั่งไปใช้เองได้ นอกจากนี้พีเอชพียังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเทอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพี ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผล การอ่านข้อมูลจากดาต้าเบส ความสามารถจัดการกับคุกกี้ ซึ่งทำงานเช่นเดียวกับโปรแกรมในลักษณะCGI คุณสมบัติอื่นเช่น การประมวลผลตามบรรทัดคำสั่ง (command line)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้ผู้เขียนโปรแกรมสร้างสคริปต์พีเอชพี ทำงานผ่านพีเอชพี พาร์เซอร์ (PHP parser) โดยไม่ต้องผ่านเซิร์ฟเวอร์หรือเบราว์เซอร์ ซึ่งมีลักษณะเหมือนกับ Cron (ใน ยูนิกซ์หรือลีนุกซ์) หรือ Task Scheduler (ในวินโดวส์) สคริปต์เหล่านี้สามารถนำไปใช้ในแบบ Simple text processing tasks ได้

การแสดงผลของพีเอชพี ถึงแม้ว่าจุดประสงค์หลักใช้ในการแสดงผล HTML แต่ยังสามารถสร้าง XHTML หรือ XML ได้ นอกจากนี้สามารถทำงานร่วมกับคำสั่งเสริมต่างๆ ซึ่งสามารถแสดงผลข้อมูลหลัก PDF แฟลช (โดยใช้ libswf และ Ming) พีเอชพีมีความสามารถอย่างมากในการทำงานเป็นประมวลผลข้อความ จาก POSIX Extended หรือ รูปแบบ Perl ทั่วไป เพื่อแปลงเป็นเอกสาร XML ในการแปลงและเข้าสู่เอกสาร XML เรารองรับมาตราฐาน SAX และ DOM สามารถใช้รูปแบบ XSLT ของเราเพื่อแปลงเอกสาร XML

เมื่อใช้พีเอชพีในการทำอีคอมเมิร์ซ สามารถทำงานร่วมกับโปรแกรมอื่น เช่น Cybercash payment, CyberMUT, VeriSign Payflow Pro และ CCVS functions เพื่อใช้ในการสร้างโปรแกรมทำธุรกรรมทางการเงิน

2.4.2.2 การรองรับ PHP

คำสั่งของพีเอชพี สามารถสร้างผ่านทางโปรแกรมแก้ไขข้อความทั่วไป เช่น โน้ตแพด หรือ vi ซึ่งทำให้การทำงานพีเอชพี สามารถทำงานได้ในระบบปฏิบัติการหลักเกือบทั้งหมด โดยเมื่อเขียนคำสั่งแล้วนำมาประมวลผล Apache, Microsoft Internet Information Services (IIS) , Personal Web Server, Netscape และ iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, และอื่นๆ อีกมากมาย. สำหรับส่วนหลักของ PHP ยังมี Module ในการรองรับ CGI มาตรฐาน ซึ่ง PHP สามารถทำงานเป็นตัวประมวลผล CGI ด้วย และด้วย PHP, คุณมีอิสรภาพในการเลือก ระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ นอกจากนี้คุณยังสามารถใช้สร้างโปรแกรมโครงสร้าง สร้างโปรแกรมเชิงวัตถุ (OOP) หรือสร้างโปรแกรมที่รวมทั้งสองอย่างเข้าด้วยกัน แม้ว่าความสามารถของคำสั่ง OOP มาตรฐานในเวอร์ชันนี้ยังไม่สมบูรณ์ แต่ตัวไลบรารีทั้งหลายของโปรแกรม และตัวโปรแกรมประยุกต์ (รวมถึง PEAR library) ได้ถูกเขียนขึ้นโดยใช้รูปแบบการเขียนแบบ OOP เท่านั้น

พีเอชพีสามารถทำงานร่วมกับฐานข้อมูลได้หลายชนิด ซึ่งฐานข้อมูลส่วนหนึ่งที่รองรับได้แก่ ออราเคิล dBase PostgreSQL IBM DB2 MySQL Informix ODBC โครงสร้างของฐานข้อมูลแบบ DBX ซึ่งทำให้พีเอชพีใช้กับฐานข้อมูลอะไรก็ได้ที่รองรับรูปแบบนี้ และ PHP ยังรองรับ ODBC (Open Database Connection) ซึ่งเป็นมาตรฐานการเชื่อมต่อฐานข้อมูลที่ใช้กันแพร่หลายอีกด้วย คุณสามารถเชื่อมต่อกับฐานข้อมูลต่างๆ ที่รองรับมาตรฐานโลกนี้ได้

พีเอชพียังสามารถรองรับการสื่อสารกับการบริการในโปรโตคอลต่างๆ เช่น LDAP IMAP SNMP NNTP POP3 HTTP COM (บนวินโดวส์) และอื่นๆ อีกมากมาย คุณสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปิด Socket บนเครื่อข่ายโดยตรง และ ตอบโต้โดยใช้ โปรโตคอลใดๆ ก็ได้ PHP มีการรองรับสำหรับการแลกเปลี่ยนข้อมูลแบบ WDDX Complex กับ Web Programming อื่นๆ ทั่วไปได้ พูดถึงในส่วน Interconnection, พีเอชพีมีการรองรับสำหรับ Java objects ให้เปลี่ยนมันเป็น PHP Object แล้วใช้งาน คุณยังสามารถใช้รูปแบบ CORBA เพื่อเข้าสู่ Remote Object ได้เช่นกัน

2.4.3 MYSQL

MySQL (มายเอสคิวแอล) เป็นระบบจัดการฐานข้อมูลโดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจMySQL สร้างขึ้นโดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael "Monty" Widenius.

MYSQL เป็นที่นิยมใช้กันมากสำหรับฐานข้อมูลสำหรับเว็บไซต์ เช่น มีเดียวิกิ และ phpBB และนิยมใช้งานร่วมกับภาษาโปรแกรม PHP ซึ่งมักจะได้ชื่อว่าเป็นคู่ จะเห็นได้จากคู่มือคอมพิวเตอร์ต่างๆ ที่จะสอนการใช้งาน MySQL และ PHP ควบคู่กันไป นอกจากนี้ หลายภาษาโปรแกรมที่สามารถทำงานร่วมกับฐานข้อมูล MySQL ซึ่งรวมถึง ภาษาซี ซีพลัสพลัส ปาสคาล ซีชาร์ป จาวา ภาษาเพิร์ล พีเอชพี ไพทอน รูบี และภาษาอื่น ใช้งานผ่าน API สำหรับโปรแกรมที่ติดต่อผ่าน ODBC หรือ ส่วนเชื่อมต่อกับภาษาอื่น (database connector) เช่น เอเอสพี สามารถเรียกใช้ MySQL ผ่านทาง MyODBC,ADO,ADO.NET เป็นต้น

ในการจัดการฐานข้อมูล MySQL คุณสามารถใช้โปรแกรมแบบ command-line เพื่อจัดการฐานข้อมูล (โดยใช้คำสั่ง: mysql และ mysqladmin เป็นต้น). หรือจะดาวน์โหลดโปรแกรมจัดการฐานข้อมูลแบบ GUI จากเว็บไซต์ของ MySQL ซึ่งคือโปรแกรม: MySQL Administrator และ MySQL Query Browser. เป็นต้น

มีส่วนติดต่อ (interface) เพื่อเชื่อมต่อกับภาษาในการพัฒนา อื่นๆ เพื่อให้เข้าถึงฟังก์ชันการทำงานกับฐานข้อมูล MySQL ได้เช่น ODBC (Open Database Connector) อันเป็นมาตรฐานกลางที่กำหนดมาเพื่อให้ใช้เป็นสะพานในการเชื่อมต่อกับโปรแกรมหรือระบบอื่นๆ เช่น MyODBC อันเป็นไดรเวอร์เพื่อใช้สำหรับการเชื่อมต่อในระบบปฏิบัติการวินโดว์, JDBC คลาสส่วนเชื่อมต่อสำหรับ Java เพื่อใช้ในการติดต่อกับ MySQL และมี API (Application Programming Interface) ต่างๆมีให้เลือกใช้มากมายในการที่เข้าถึง MySQL โดยไม่ขึ้นอยู่กับภาษาการพัฒนาใดภาษาหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Web Browser

2.5.1 HTML

HTML (ชื่อเต็มคือ Hypertext Markup Language ภาษามาร์กอัปข้อความหลายมิติ) คือภาษามาร์กอัปออกแบบมาเพื่อใช้ในการสร้างเว็บเพจ หรือข้อมูลอื่นที่เรียกดูผ่านทางเว็บเบราว์เซอร์ เริ่มพัฒนาโดย ทิม เบอร์เนอรส์ ลี (Tim Berners Lee) สำหรับภาษา SGML ในปัจจุบัน HTML เป็นมาตรฐานหนึ่งของ ISO ซึ่งจัดการโดย World Wide Web Consortium (W3C) ในปัจจุบัน ทาง W3C ผลักดัน รูปแบบของ HTML แบบใหม่ ที่เรียกว่า XHTML ซึ่งเป็นลักษณะของโครงสร้าง XML แบบหนึ่งที่มีหลักเกณฑ์ในการกำหนดโครงสร้างของโปรแกรมที่มีรูปแบบที่มาตรฐานกว่า มาทดแทนใช้ HTML รุ่น 4.01 ที่ใช้กันอยู่ในปัจจุบัน HTML ยังคงเป็นรูปแบบไฟล์อย่างหนึ่ง สำหรับ .html และ สำหรับ .htm ที่ใช้ในระบบปฏิบัติการที่รองรับ รูปแบบนามสกุล 3 ตัวอักษร

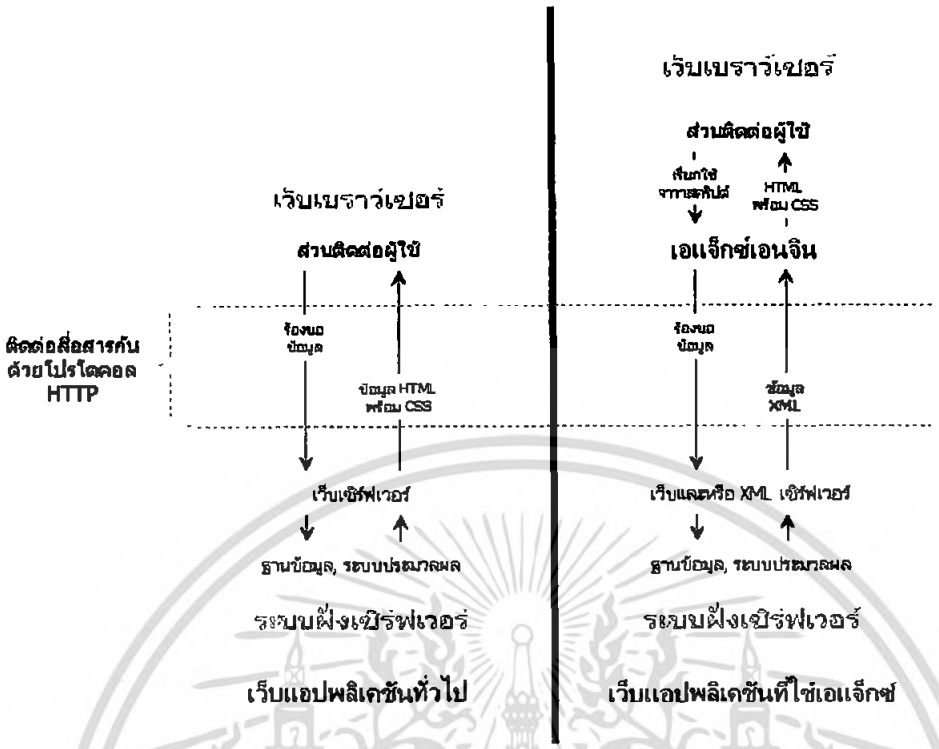
2.5.2 AJAX

เอแจ็กซ์ (AJAX - Asynchronous JavaScript And XML) เป็นเทคนิคในการพัฒนาเว็บแอปพลิเคชันเพื่อให้ความสามารถโต้ตอบกับผู้ใช้ได้ดีขึ้น โดยการรับส่งข้อมูลในฉากหลัง ทำให้ทั้งหน้าไม่ต้องโหลดใหม่ทุกครั้งที่มีการเปลี่ยนแปลง ซึ่งช่วยทำให้เพิ่มการตอบสนอง ความรวดเร็วและการใช้งานโดยรวม

2.5.2.1 ประวัติความเป็นมา

Jeese Jams Garett นั้นเป็นผู้ที่ได้บัญญัติคำเอแจ็กซ์ขึ้นเมื่อปีพ.ศ. 2548 ซึ่งได้นึกขึ้นระหว่างที่เขาอาบน้ำ เพื่อหาคำสั้นๆ สำหรับอธิบายให้ลูกค้าของเขาเกี่ยวกับเทคโนโลยีต่างๆที่ต้องการจะเสนอเอแจ็กซ์โดยตัวมันเองแล้วไม่ได้เป็นเทคโนโลยีหรือภาษาโปรแกรมชนิดใหม่ แต่เป็นการรวมกลุ่มของเทคโนโลยีที่มีใช้อยู่แล้วดังที่กล่าวข้างต้น โดยวิวัฒนาการของเอแจ็กซ์เริ่มต้นเมื่อปีค.ศ. 2002 ไมโครซอฟท์ได้ทำการคิดค้น XMLHttpRequest ขึ้นมาเพื่อเป็นทางเลือกในการเขียนโปรแกรมบนเว็บเพจ เพื่อใช้ติดต่อกับ เว็บเซิร์ฟเวอร์ ในการแลกเปลี่ยนข้อมูล ซึ่งในขณะนั้นมีแต่เพียง อินเทอร์เน็ตเอกซ์พลอเรอร์ เท่านั้นที่มีความสามารถนี้ ต่อมาเว็บเบราว์เซอร์อื่นๆ เช่นจาก มอซิลลา ไฟร์ฟอกซ์ ได้นำแนวคิดของ XMLHttpRequest ไปใส่ในเบราว์เซอร์ของตนด้วย จึงเริ่มทำให้มีการใช้อย่างกว้างขวางขึ้น จนปัจจุบันได้กลายเป็นมาตรฐานที่ทุกเว็บเบราว์เซอร์ต้องมีในตอนแรกนั้นไมโครซอฟท์เป็นผู้ที่ได้นำ XMLHttpRequest โดยใช้ใน Outlook Web Access ที่มาพร้อมกับ Microsoft Exchange Server 2000. ต่อมาเว็บไซต์อย่างกูเกิล ได้เปิดบริการใหม่ชื่อจีเมล ซึ่งใช้ XMLHttpRequest เป็นหัวใจหลักในการดึงข้อมูลจากเว็บเซิร์ฟเวอร์ จึงทำให้แนวคิดและเทคนิคการพัฒนาเว็บแอปพลิเคชันด้วย เอแจ็กซ์เริ่มรู้จักเป็นที่กว้างขวางขึ้น จนปัจจุบันถือว่าเป็นหนึ่งในหัวใจหลักของแนวคิดเรื่อง Web 2.0

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

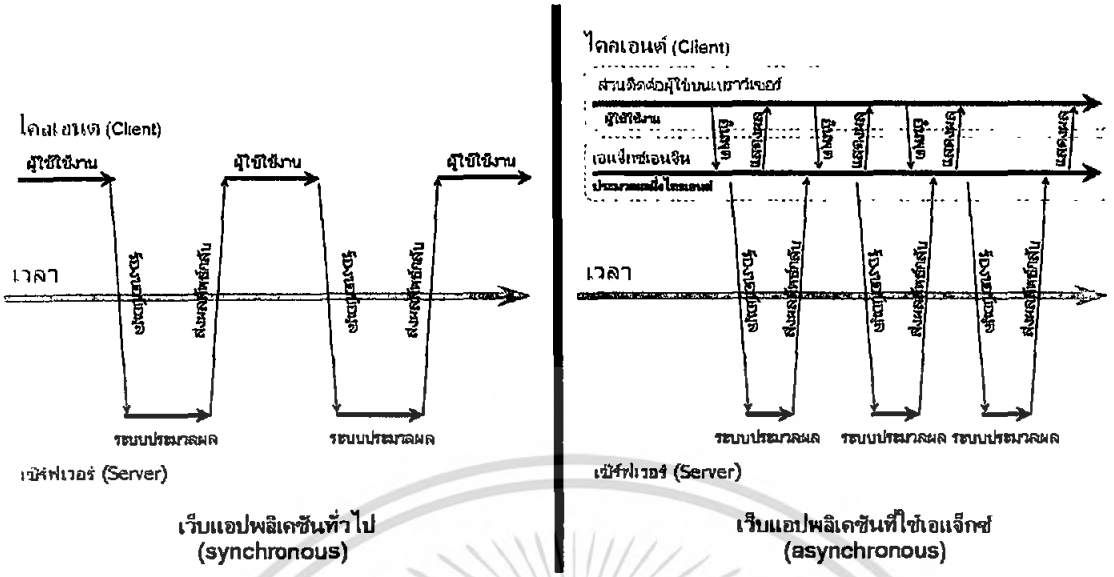


รูป 2.25 แสดงการทำงาน ของ AJAX

2.5.2.2 หลักการทำงาน

วิธีการทำงานของเว็บแอปพลิเคชันแบบดังเดิมนั้น โดยปกติแล้วเมื่อผู้ใช้ทำการร้องขอข้อมูลจากเซิร์ฟเวอร์ ตัวเว็บเบราว์เซอร์จะทำการส่งข้อมูลการร้องขอโดยใช้โปรโตคอล HTTP เพื่อติดต่อกับเว็บเซิร์ฟเวอร์ และที่เว็บเซิร์ฟเวอร์จะทำการประมวลผลจากการร้องขอที่ได้รับและส่งผลลัพธ์เป็นหน้า HTML กลับไปให้ผู้ใช้ วิธีการข้างต้นเป็นวิธีการแบบการร้องขอและการตอบรับ (Request and Response) ซึ่งผู้ใช้จะต้องรอระหว่างที่เซิร์ฟเวอร์ประมวลผลอยู่ ซึ่งเป็นหลักการทำงานแบบ Synchronous แต่การทำงานของเว็บแอปพลิเคชันที่ใช้เทคนิคเอแจ็กซ์จะเป็นการทำงานแบบ Asynchronous หรือการติดต่อสื่อสารแบบไม่ต่อเนื่อง โดยเซิร์ฟเวอร์จะทำการส่งผลลัพธ์เป็นเว็บเพจให้ผู้ใช้ทันทีโดยไม่ต้องรอให้ประมวลผลเสร็จก่อน หลังจากนั้นเว็บเพจที่ผู้ใช้ได้รับจะทำการดึงข้อมูลในส่วนต่างๆทีหลัง หรือจะดึงข้อมูลก็ต่อเมื่อผู้ใช้ต้องการเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.26 เปรียบเทียบเวปแอปลิเคชันทั่วไป กับ AJAX

2.5.3 Java Script

จาวาสคริปต์ (JavaScript) เป็นภาษาสคริปต์ ที่มีลักษณะการเขียนแบบโปรโตไทพ (Prototyped-based Programming) ส่วนมากใช้ในหน้าเว็บเพื่อประมวลผลข้อมูลที่ฝั่งของผู้ใช้งาน แต่ก็ยังมีใช้เพื่อเพิ่มเติมความสามารถในการเขียนสคริปต์โดยฝังอยู่ในโปรแกรมอื่นๆ ชัน ไมโครซิสเต็มส์ เป็นเจ้าของเครื่องหมายการค้า "JavaScript" โดยมันถูกนำไปใช้ภายใต้สัญญาอนุญาตเพื่อการพัฒนาเทคโนโลยีโดย เน็ตสเคป และมูลนิธิมอซิลลา

2.5.3.1 ประวัติ

เริ่มพัฒนาโดย Brendan Eich พนักงานบริษัทเน็ตสเคป โดยขณะนั้นจาวาสคริปต์ใช้ชื่อว่า โมคาและภายหลังได้เปลี่ยนชื่อมาเป็น ไลฟ์สคริปต์ และเป็น จาวาสคริปต์ในปัจจุบัน รูปแบบการเขียนภาษาที่ใช้ คล้ายคลึงกับภาษาซี รุ่นล่าสุดของจาวาสคริปต์คือ 2.0 ซึ่งตรงกับมาตรฐานของ ECMAScriptภาษาจาวาสคริปต์ไม่มีความสัมพันธ์กับ ภาษาจาวา (Java) และเจสคริปต์ (JScript) แต่อย่างใด ยกเว้นแต่โครงสร้างภาษาที่มีลักษณะคล้ายคลึงกัน เนื่องมาจากได้รับการพัฒนาต่อมาจากภาษาซีเหมือนๆ กัน และมีชื่อที่คล้ายคลึงกันเท่านั้นสำหรับเจสคริปต์ (JScript) หลังจากที่จาวาสคริปต์ประสบความสำเร็จ โดยมีเว็บเบราว์เซอร์จากหลายๆ บริษัทนำมาใช้งาน ทางไมโครซอฟท์จึงได้พัฒนาภาษาโปรแกรมที่ทำงานในลักษณะคล้ายคลึงกับจาวาสคริปต์ขึ้น และตั้งชื่อว่าเจสคริปต์ ซึ่งทำงานได้กับเบราว์เซอร์อินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer) เท่านั้น เริ่มใช้ครั้งแรกใน อินเทอร์เน็ตเอกซ์พลอเรอร์ 3.0 เมื่อ สิงหาคม พ.ศ. 2539 โดยสร้างตามมาตรฐาน ECMA 262

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.2 การใช้งาน

จาวาสคริปต์ เป็นภาษาในรูปแบบของภาษาโปรแกรมแบบโปรโตไทพ โดยมีโครงสร้างของภาษาและไวยกรณ์อยู่บนพื้นฐานของภาษาซีปัจจุบันมีการใช้จาวาสคริปต์ที่ฝังอยู่ในเว็บเบราว์เซอร์ในหลายรูปแบบ เช่น ใช้เพื่อสร้างเนื้อหาที่เปลี่ยนแปลงเสมอภายในเว็บเพจ, ใช้เพื่อตรวจสอบความถูกต้องของข้อมูลที่ผู้ใช้กรอกก่อนนำเข้าระบบ, ใช้เพื่อเข้าถึงข้อมูลที่อยู่ภายใต้โครงสร้างแบบ Document Object Model (DOM)เป็นต้น นอกจากนี้จาวาสคริปต์ยังถูกฝังอยู่ในแอปพลิเคชันต่างๆ นอกเหนือจากเว็บเบราว์เซอร์ได้อีกด้วย เช่น widget ของ ยาฮู! เป็นต้น โดยรวมแล้วจาวาสคริปต์ถูกใช้เพื่อให้นักพัฒนาโปรแกรม สามารถเขียนสคริปต์เพื่อสร้างคุณสมบัติพิเศษต่างๆ เพิ่มเติมจากที่มีอยู่บนแอปพลิเคชันดังเดิมโปรแกรมใดๆ ที่สนับสนุนจาวาสคริปต์จะมีตัวขับเคลื่อนจาวาสคริปต์ (JavaScript Engine) ของตัวเอง เพื่อเรียกใช้งานโครงสร้างเชิงวัตถุของโปรแกรมหรือแอปพลิเคชันนั้นๆ

2.5.4 XML

เอกซ์เอ็มแอล (XML) ย่อมาจาก Extensible Markup Language ซึ่งเป็นภาษามาร์กอัปสำหรับการใช้งานทั่วไป พัฒนาโดยW3C โดยมีจุดประสงค์เพื่อเป็น สิ่งที่เอาไว้ติดต่อกันในระบบที่มีความแตกต่างกัน(เช่นใช้คอมพิวเตอร์มี่มีระบบปฏิบัติการคนละตัว หรืออาจจะเป็นคนละโปรแกรมประยุกต์ที่มีความต้องการสื่อสารข้อมูลถึงกัน)นอกจากนี้ยังเพื่อเป็นพื้นฐานในการสร้างภาษามาร์กอัปเฉพาะทางอีกขั้นหนึ่ง XML พัฒนามาจาก SGML โดยดัดแปลงให้มีความซับซ้อนลดน้อยลง XML ใช้ในแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่แตกต่างกัน และเน้นการแลกเปลี่ยนข้อมูลผ่านอินเทอร์เน็ต XML ยังเป็นภาษาพื้นฐานให้กับภาษาอื่นๆ อีกด้วย (ยกตัวอย่าง เช่น Geography Markup Language (GML), RDF/XML, RSS, MathML, Physical Markup Language (PML), XHTML, SVG, MusicXML และ cXML) ซึ่งอนุญาตให้โปรแกรมแก้ไขและทำงานกับเอกสารโดยไม่จำเป็นต้องมีความรู้ในภาษานั้นมาก่อน XML เป็นภาษาที่ใช้เน้น (มาร์กอัป) ส่วนที่เป็นข้อมูล โดยสามารถกำหนดชื่อแท็ก (Element) และชื่อแอตทิบิวต์ ได้ตามความต้องการของผู้สร้างเอกสาร xml โดยเอกสารนั้นจะต้องมีความเป็น Well-formed ส่วน DTD และ Schema จะมีหรือไม่มีก็ได้ ขึ้นอยู่กับว่ามีผู้ใช้เอกสารนั้นมากน้อยแค่ไหน เอกสาร xml จึงเป็นแค่เท็กซ์ไฟล์ชนิดหนึ่ง ที่มีแท็กเปิดและแท็กปิดครอบข้อมูลไว้ตรงกลางเท่านั้น ทำให้เอกสาร xml ถูกใช้ในการติดต่อกับระบบที่ต่างกัน เนื่องจากความง่ายในการสร้างเอกสาร การนำเอกสาร xml ไปใช้งาน จะสนใจแต่ข้อมูลที่ถูกเน้นด้วยแท็กมากกว่าWell-formed เป็นไวยากรณ์พื้นฐานของเอกสาร xml อย่างเช่น เอกสาร xml ต้องเริ่มต้นด้วย <?xml version="1.0" ?> เอกสาร xml 1 เอกสาร จะต้องมีแท็กรูทเพียงแท็กเดียว หมายความว่า แท็กและข้อมูลต่างๆ จะต้องอยู่ภายในแท็กแรกสุดเพียงแท็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

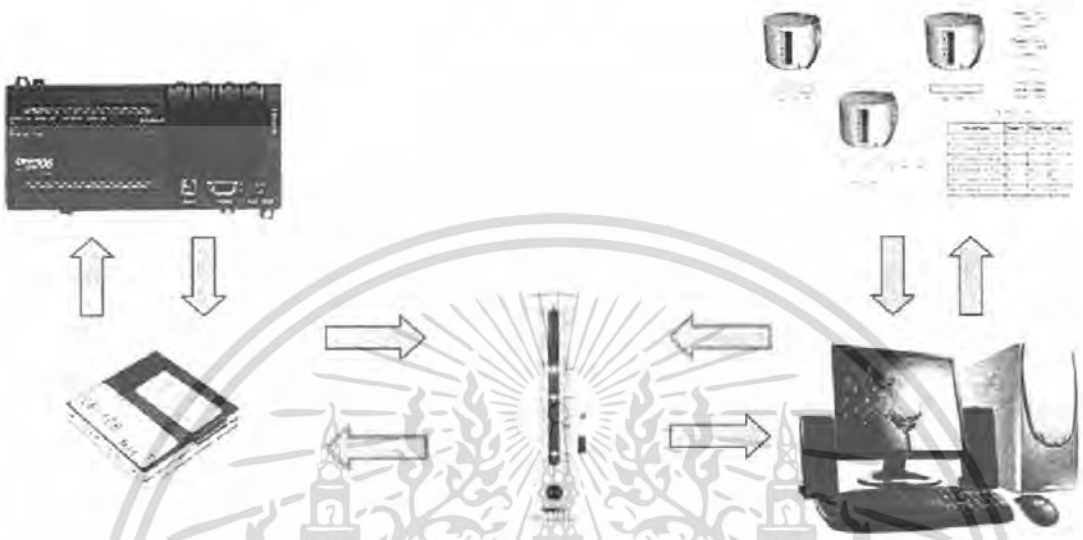
เดียว การเปิดและปิดแท็กจะต้องไม่มีการครอมกัน เช่น ตัวหนาและเอียง จะไม่ Well-formed เนื่องจากเอกสาร xml สามารถกำหนดชื่อแท็ก และชื่อแอตทิบิวต์ได้ตามความต้องการของผู้สร้างเอกสาร ทำให้ในการเน้นข้อมูลใดข้อมูลหนึ่ง สามารถมีเอกสาร xml หลายรูปแบบ (ผู้เขียนอาจใช้ชื่อแท็กต่างกัน ทั้งที่สื่อความหมายไปที่สิ่งเดียวกัน) หากว่าเอกสาร xml นั้น ถูกนำไปใช้ติดต่อกับระบบอื่นๆ อาจทำให้สื่อความหมายไม่ตรงกัน ดังนั้นจึงต้องมีการกำหนดรูปแบบที่เป็นมาตรฐานขึ้น (ตกลงรูปแบบระหว่างกัน) โดย DTD และ Schema จะเป็นตัวกำหนดว่าเอกสาร xml นั้น จะต้องมีแท็กอะไรบ้าง กายในแท็กนั้นจะมีแท็ก แอตทิบิวต์ หรือข้อมูลอะไรได้บ้าง โดย DTD จะต่างกับ Schema ตรงที่ Schema เป็นเอกสาร xml ด้วย

XMLHttpRequest เป็น Application Programming Interface (API) ซึ่งสามารถเรียกใช้ได้จาก จาวาสคริปต์ เจสคริปต์ วีบีสคริปต์ และภาษาสคริปต์อื่นๆ ในการแลกเปลี่ยน และปรับรูปแบบ XML จากเว็บเซิร์ฟเวอร์ โดยใช้ HTTP ซึ่งสร้างการเชื่อมต่อระหว่างเว็บเบราว์เซอร์ (Client-Side) กับ เว็บเซิร์ฟเวอร์ (Server-Side)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและทดลองระบบ



รูปที่ 3.1 แสดงการติดต่อของระบบ

3.1 ส่วนของฮาร์ดแวร์ (Hardware)

3.1.1 PLC

PLC เป็นตัวควบคุมกระบวนการจัดค่าพารามิเตอร์ต่าง ๆ ทั้งตัวรับจากทรานดิวเซอร์และส่งสัญญาณควบคุมให้กับ Solid State Relay โดย PLC จะมีการเชื่อมต่อไปยังคอมพิวเตอร์ส่งค่าระดับ (ในรูปของข้อมูล) และรับสัญญาณ Operate

การจัดอุปกรณ์อินพุต และเอาต์พุต ต่อเข้ากับขั้วอินพุตและเอาต์พุตของ PLC โดยลักษณะการต่อเป็นแบบ Souse ต่อไฟเลี้ยง บวก เข้ากับ Common ของ PLC กับขาด้านหนึ่งของอุปกรณ์อินพุต และเอาต์พุต

3.1.2 TCP/IP Stack Chip

TNS010i TCP/IP Stack Chip จะรับคำสั่งซึ่งเป็น AT Command จาก PLC ในรูปของ ASCII มาประมวลผลรูปแบบคำสั่ง และทำการส่งต่อไปยัง GPRS Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 GPRS Module

Sony Ericsson gm29 เป็นอุปกรณ์ในการเชื่อมต่อเครือข่าย GPRS เพื่อทำการส่งค่าที่ได้จาก PLC ไปยังเครื่อง Server ตามที่อยู่ที่กำหนด และรับค่าจาก Server มายัง PLC เพื่อทำการควบคุม Plant

3.2 ส่วนของซอฟท์แวร์(Software)

3.2.1 Direct Soft 32

เป็นซอร์ฟแวร์ที่ใช้ในการเขียน Ladder ซึ่งเป็นโปรแกรมการทำงานของ PLC แล้วทำการ Memory ลงใน CPU ของ PLC สามารถเพิ่มเติม แก้ไขได้สะดวกกว่าการแก้ใน PLC โดยตรง

3.2.2 Apache

เป็นซอฟท์แวร์ สำหรับเปิดให้บริการเซิร์ฟเวอร์บนโปรโตคอล HTTP โดยสามารถทำงานได้บนหลายระบบปฏิบัติการ

3.2.3 My SQL, PHP

MY SQL เป็นโปรแกรมที่ใช้ในการจัดเก็บข้อมูลที่ส่งมาจาก GPRS Module โดยมี PHP เป็น API ในการใช้งาน MY SQL

3.2.3 Ajax

เป็นส่วนช่วยในการแสดงผลบนหน้าเว็บเพจ โดยใช้ Javascript กับ Object ของ Xml (Xmlhttprequest) ทำงานร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การออกแบบการทดลอง

1. ระบบที่ใช้ในการทดลองนี้ เป็นการจำลองการแสดงผล ของการวัดระดับ ซึ่งมีสัญญาณอนาล็อกจากอุปกรณ์ทรานสดิวเซอร์ มาตรฐาน 4 – 20 mA เข้าสู่ PLC

2. ทำการโปรแกรม PLC ให้สามารถส่งสัญญาณเข้าสู่ GPRS Module โดยพอร์ท RS-232 ทำให้ GPRS Module ส่งแพคเกจข้อมูล เข้าสู่ SQL Server โดยเครือข่าย GPRS

3. เซิร์ฟเวอร์ รับและบันทึกค่าที่ได้ จาก GPRS Module เพื่อที่จะสามารถเรียกดูจากเครือข่าย อินเทอร์เน็ตได้จากทุกที่ผ่านทาง Web Browser

4. ทำให้ Web Page สามารถควบคุมการทำงานอุปกรณ์บางส่วนของระบบในกระบวนการ

4.2 ขั้นตอนการทดลอง

1. ทำการเชื่อมต่อ อุปกรณ์ ตามที่ออกแบบไว้

2. เปิดเซิร์ฟเวอร์ และรัน โปรแกรม Apache , MySQL

3. เมื่อทำการเปิด TCP/IP Stack และสั่งงาน TCP/IP Stack ด้วย Hyper Terminal ใช้คำสั่งเพื่อเปิดพอรทไฟสีเขียวจะติด จะเป็นการเปิดพอรท GPRS ถ้าขั้นตอนนี้เรียบร้อยดี เมื่อมีการส่งข้อมูลจะมีสัญญาณไฟ LED สีเหลืองกระพริบ เป็นการบอกสถานะ

4. เมื่อมีสัญญาณ อนาล็อกเข้ามา PLC จะทำงานและเริ่มส่งค่าออกทาง GPRS Module ไปยัง เซิร์ฟเวอร์ เซิร์ฟเวอร์จะรับค่าพร้อมเวลาที่รับ และเริ่มบันทึกลงตาราง

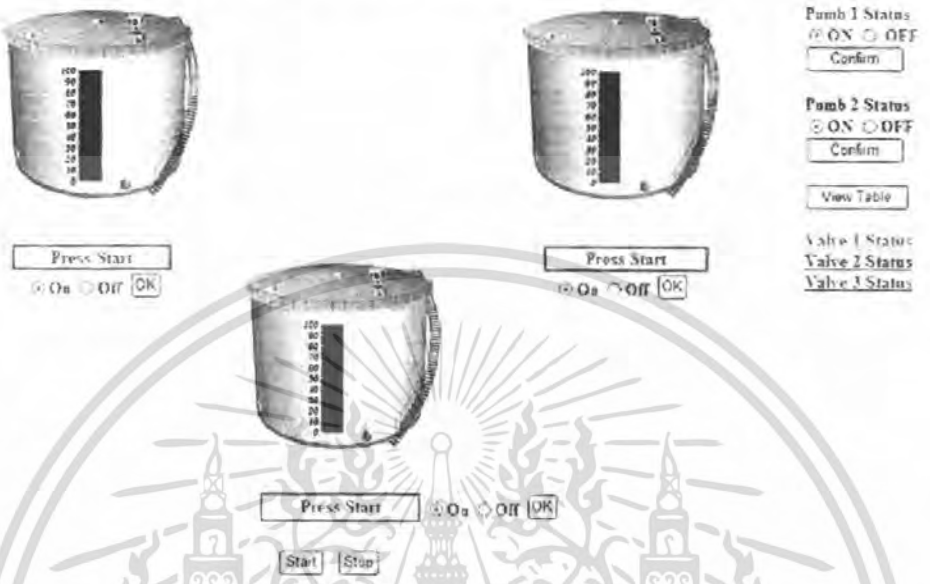
5. การแสดงผลสามารถเรียกดูผ่านทาง Web Browser ได้จากเครือข่าย อินเทอร์เน็ต

6. ทำการทดลองควบคุมอุปกรณ์ผ่านทาง Web Browser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

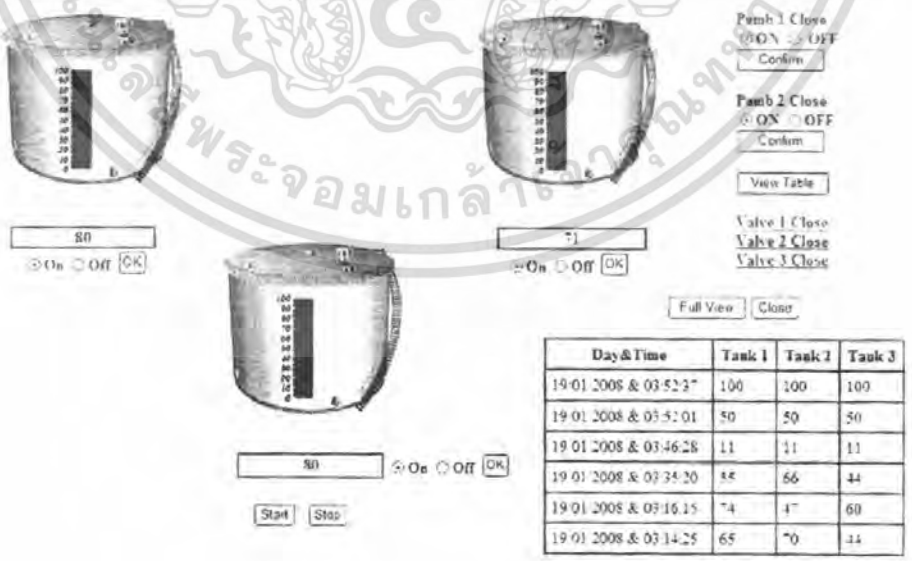
4.3 บันทึกผลการทดลอง

1. เมื่อเปิด Web Browser ขึ้นมา จะเป็นโปรแกรมแสดงผล ดังรูป



รูปที่ 4.1 แสดงหน้าเวปเริ่มต้นโดย Web Browser จะแสดงค่าเป็น 0 ทุกแท่งค์

2. เมื่อกดปุ่ม Start Web Browser จะแสดงค่าสุดท้ายที่อยู่ในตารางของ เซิร์ฟเวอร์ พร้อมทั้ง สถานะของปั๊ม และ วาลวควบคุม ในตอนนั้นด้วย



Day&Time	Tank 1	Tank 2	Tank 3
19/01/2008 & 03:52:37	100	100	100
19/01/2008 & 03:52:01	50	50	50
19/01/2008 & 03:46:28	11	11	11
19/01/2008 & 03:35:20	55	66	44
19/01/2008 & 03:16:15	74	47	60
19/01/2008 & 03:14:25	65	70	44

รูปที่ 4.2 แสดงหน้าเวปเมื่อเริ่ม Start และเริ่มแสดงค่าสุดท้ายจากเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

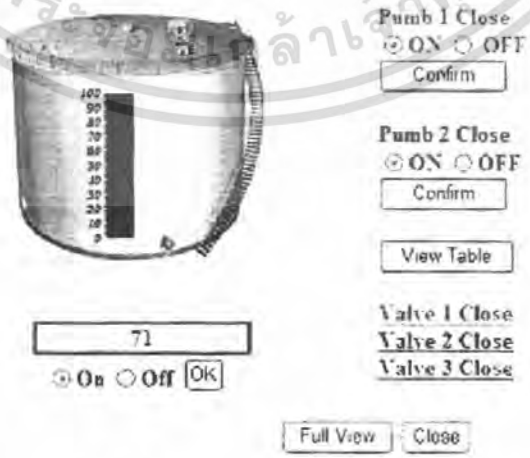
3. จากตารางแสดงผล สามารถดูข้อมูลที่บันทึกในตาราง ย้อนหลังได้โดยกดปุ่ม Full View

ตาราง 2.5 แสดงค่าที่บันทึกในเซิร์ฟเวอร์

Day & Time	Tank1	Tank2	Tank3
19/01/2008 & 03:52:37	100	100	100
19/01/2008 & 03:52:30	100	100	100
19/01/2008 & 03:52:23	0	40	20
19/01/2008 & 03:52:16	50	40	20
19/01/2008 & 03:52:09	50	40	20
19/01/2008 & 03:52:01	50	50	50
19/01/2008 & 03:51:55	50	50	50
19/01/2008 & 03:47:00	22	22	22
19/01/2008 & 03:46:49	22	22	22
19/01/2008 & 03:46:38	11	11	11
19/01/2008 & 03:46:28	11	11	11
19/01/2008 & 03:46:17	11	11	11
19/01/2008 & 03:45:39	40	50	60
19/01/2008 & 03:45:30	40	50	60
19/01/2008 & 03:35:29	55	66	77
19/01/2008 & 03:35:20	55	66	44
19/01/2008 & 03:35:09	55	44	44
19/01/2008 & 03:34:59	55	44	44
19/01/2008 & 03:34:50	44	44	44
19/01/2008 & 03:34:38	44	44	44

145 Recored in 8 pages : 1 [2][3][4][5][6][7] Next>> Close

4. สำหรับการควบคุมผ่านทาง Web Browser จะมีฟังก์ชันให้จัดการอุปกรณ์บางจากรูป เป็น การแสดงสถานะ และปุ่มควบคุม ปั๊ม และ วาล์ว



รูปที่ 4.3 แสดงฟังก์ชันสถานะและการควบคุมอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 สรุปผลการทดลอง

1. จากการทดลอง GPRS Module สามารถรับข้อมูลจาก PLC แล้วส่งค่า ไปบันทึกลงฐานข้อมูลที่ติดตั้ง บนเซิร์ฟเวอร์

2. Web Browser สามารถแสดงผลการเปลี่ยนแปลงจาก PLC ได้ และสามารถส่งค่าควบคุม ลงในฐานข้อมูล เพื่อส่งต่อไปยัง PLC ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 ปัญหาทางเทคนิค

1. เรื่องของสัญญาณ GPRS ที่บางครั้งไม่สามารถเชื่อมต่อได้เนื่องจากความหนาแน่นของเครือข่าย GPRS ณ เวลานั้น

2. สายสัญญาณต่าง ๆ อาจเกิดจากมีการนำสายไปใช้แล้วทำให้สายเกิดความเสียหายภายใน หรือ การเข้าหัวสายที่ไม่มีความหนาแน่นทนทาน ทำให้เกิดปัญหาการเชื่อมต่อสื่อสาร

3. อุปกรณ์บางชนิดที่ใช้นั้นบางชนิดหาซื้อได้ยากหรือบริษัทอาจเลิกผลิตไปแล้วทำให้ต้องมีการใช้อุปกรณ์ที่นำมาทดแทนซึ่งไม่สามารถให้ผลตามที่คาดไว้

4. อุปกรณ์ที่บางรุ่นบางยี่ห้อเป็นของที่ยังมีการรับประกันอยู่ ถ้าเสียก็ยังสามารถนำไปซ่อมได้แต่ยังมีปัจจัยทางด้านระยะเวลาที่ศูนย์แต่ละศูนย์นั้นมีระยะเวลาการซ่อมแซมที่มากน้อยไม่เท่ากันถ้าระยะเวลาซ่อมนานก็จะก่อให้เกิดปัญหามากกระทบทั้งระบบการทำงานของโครงงาน

5.2 แนะทางการพัฒนาต่อ

สามารถนำไปติดตั้งในต่างจังหวัด ตัวอย่างเช่น การประปานครหลวง หรือการไฟฟ้าต้องการที่จะวัดระดับน้ำที่อยู่ ณ ที่ต่าง ๆ ตามจังหวัดต่าง ๆ ก็สามารถนำระบบนี้ไปติดตั้ง ณ จุดต่างๆ ค่าระดับที่วัดได้ ณ แต่ละจุดจะถูกส่งเข้า PLC Ladder ภายใน PLC ก็จะทำตามกระบวนการที่เขียนส่งเป็น AT Command ต่างๆไปที่ตัว GPRS แล้ว GPRS ก็จะส่งค่าไปที่ Server เพื่อแสดงค่าระดับน้ำที่ Server ก็สามารถเปิดดูระดับน้ำจากที่ไหนก็ได้ที่มีการเชื่อมต่อ Internet แต่ก็จะมีข้อจำกัดอยู่ที่สัญญาณของแต่ละพื้นที่จะมีความแรง-ต่ำมากเพียงใด และก็ Free Server ที่จะเปิดให้บริการ ผู้มีความรู้มีประสบการณ์ที่จะสามารถทำการดูแลรักษาซ่อมบำรุงตัวอุปกรณ์ได้ เมื่ออุปกรณ์มีปัญหา

5.3 สรุป

จากการทดลองสามารถนำ PLC ซึ่งมีลักษณะการทำงานแบบ Sequential มาประยุกต์ในการสำรวจและควบคุมโดยการส่งค่าผ่านทาง Internet โดยในขั้นนี้ เครื่องควบคุม PLC สามารถส่งค่าออกทาง GPRS Module ไปยังเครื่อง Server ได้แต่ยังไม่สามารถบันทึกข้อมูลลงในฐานข้อมูลได้และรับผลการควบคุมจากหน้าเวปเพจได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ประสิทธิ์ จุลเสรีวงศ์ เอกสารประกอบการสอน วิชาวิศวกรรมการวัดคุม ภาควิชา วิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

PLC Direct Intercorporated ,DL 06 User Manual, 1994 www.Indusoft.com

http://www.wisco.co.th

http://4uengineer.com

http://php.net

http://thaidev.com

http://th.wikipedia.org/wiki

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCII Instructions

The DL06 CPU supports several instructions and methods that allow ASCII strings to be read into and written from the PLC communications ports. Specifically, port 2 on the DL06 can be used for either reading or writing raw ASCII strings, but cannot be used for both on the same CPU. The DL06 can also decipher ASCII embedded within a supported protocol (K–Sequence, DirectNet, Modbus) via the CPU port.

Reading ASCII Input Strings

There are several methods that the DL06 can use to read ASCII input strings.

1) ASCII IN (AIN) – This instruction configures port 2 for raw ASCII input strings with parameters such as fixed and variable length ASCII strings, termination characters, byte swapping options, and instruction control bits. Use barcode scanners, weight scales, etc. to write raw ASCII input strings into port 2 based on the (AIN) instruction's parameters.

2) Write embedded ASCII strings directly to V–memory from an external HMI or similar master device via a supported communications protocol using the CPU ports. The AIN instrucrion is not used in this case. 3) If a DL06 PLC is a master on a network, the Network Read instruction (RX) can be used to read embedded ASCII data from a slave device via a supported communications protocol using port 2. The RX instruction places the data directly into V–memory.

Writing ASCII Output Strings

The following instructions can be used to write ASCII output strings:

1) Print from V–memory (PRINTV) – Use this instruction to write raw ASCII strings out of port 2 to a display panel or a serial printer, etc. The instruction features the starting V–memory address, string length, byte swapping options, etc. When the instruction's permissive bit is enabled, the string is written to port 2.

2) Print to V–memory (VPRINT) – Use this instruction to create pre–coded ASCII strings in the PLC (i.e. alarm messages). When the instruction's permissive bit is enabled, the message is loaded into a pre–defined V–memory address location. Then the (PRINTV) instruction may be used to write the pre–coded ASCII string out of port 2. American, European and Asian Time/Date stamps are supported.

Additionally, if a DL06 PLC is a master on a network, the Network Write instruction (WX) can be used to write embedded ASCII data to an HMI or slave device directly from V–memory via a supported communications protocol using port 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Managing the ASCII Strings

The following instructions can be helpful in managing the ASCII strings within the CPUs V–memory:

- ASCII Find (AFIND) – Finds where a specific portion of the ASCII string is located in continuous V–memory addresses. Forward and reverse searches are supported.
- ASCII Extract (AEX) – Extracts a specific portion (usually some data value) from the ASCII find location or other known ASCII data location.
- Compare V–memory (CMPV) – This instruction is used to compare two blocks of V–memory addresses and is usually used to detect a change in an ASCII string. Compared data types must be of the same format (i.e. BCD, ASCII, etc.).
- Swap Bytes (SWAPB) – usually used to swap V–memory bytes on ASCII data that was written directly to V–memory from an external HMI or similar master device via a communications protocol. The AIN and AEX instructions have a built–in byte swap feature.

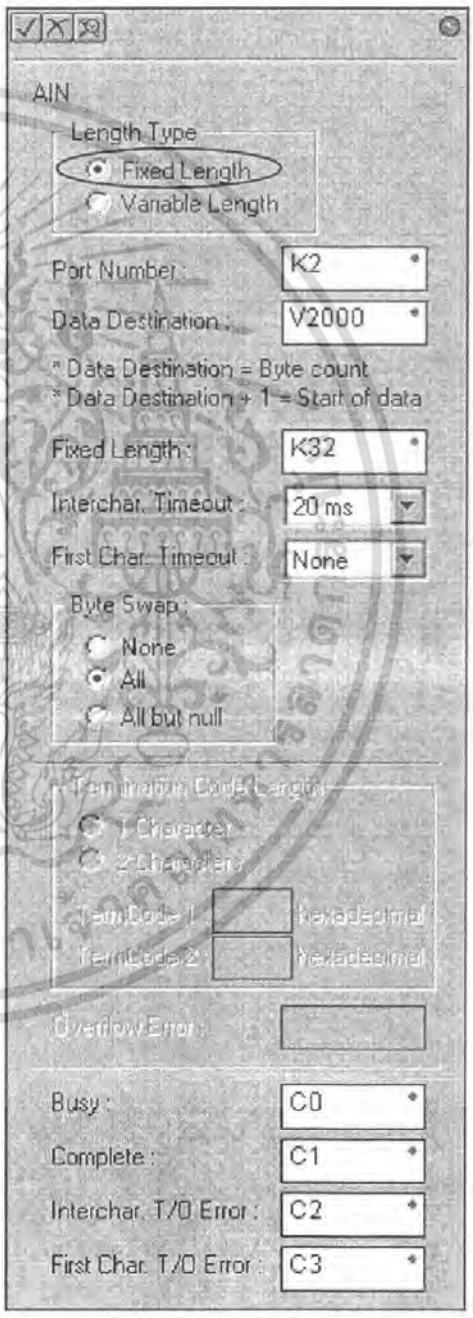
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCII Input (AIN)

The ASCII Input instruction allows the CPU to receive ASCII strings through the specified communications port and places the string into a series of specified V–memory registers. The ASCII data can be received as a fixed number of bytes or as a variable length string with a specified termination character(s). Other features include, Byte Swap preferences, Character Timeout, and user defined flag bits for Busy, Complete and Timeout Error.

AIN Fixed Length Configuration

- Length Type: select fixed length based on the length of the ASCII string that will be sent to the CPU port
- Port Number: must be DL06 port 2 (K2)
- Data Destination: specifies where the ASCII string will be placed in V–memory
- Fixed Length: specifies the length, in bytes, of the fixed length ASCII string the port will receive
- Inter–character Timeout: if the amount of time between incoming ASCII characters exceeds the set time, the specified Timeout Error bit will be set. No data will be stored at the Data Destination V–memory location. The bit will reset when the AIN instruction permissive bits are disabled. 0ms selection disables this feature.
- First Character Timeout: if the amount of time from when the AIN is enabled to the time the first character is received exceeds the set time, the specified First Character Timeout bit will be set. The bit will reset when the AIN instruction permissive bits are disabled. 0ms selection disables this feature.
- Byte Swap: swaps the high–byte and low–byte within each V–memory register of the Fixed Length ASCII string. See the SWAPB instruction for details.
- Busy Bit: is ON while the AIN instruction is receiving ASCII data
- Complete Bit: is set once the ASCII data has been received for the specified fixed length and reset when the AIN instruction permissive bits are disabled.
- Inter–character Timeout Error Bit: is set when the Character Timeout is exceeded. See Character Timeout explanation above.

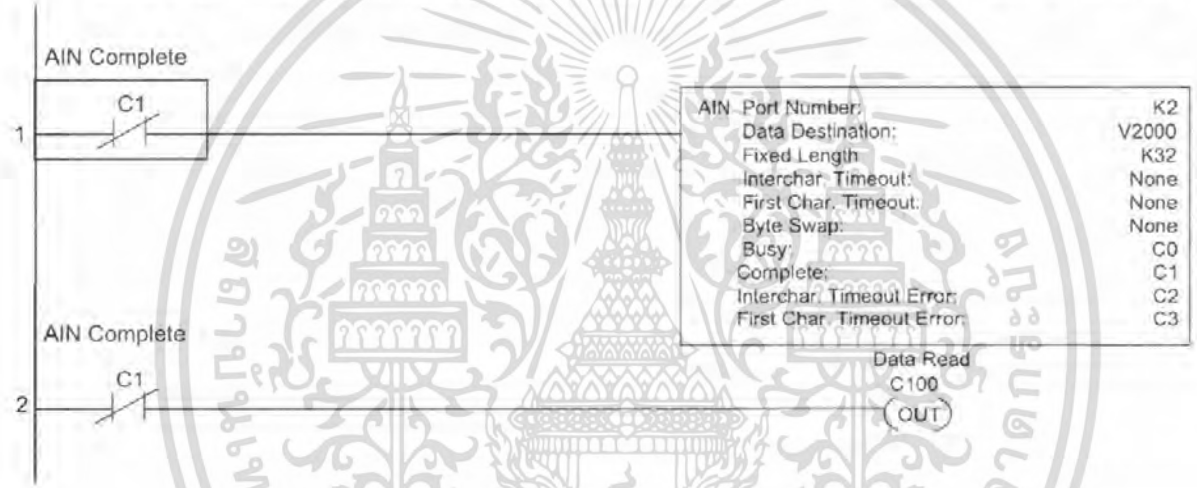


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

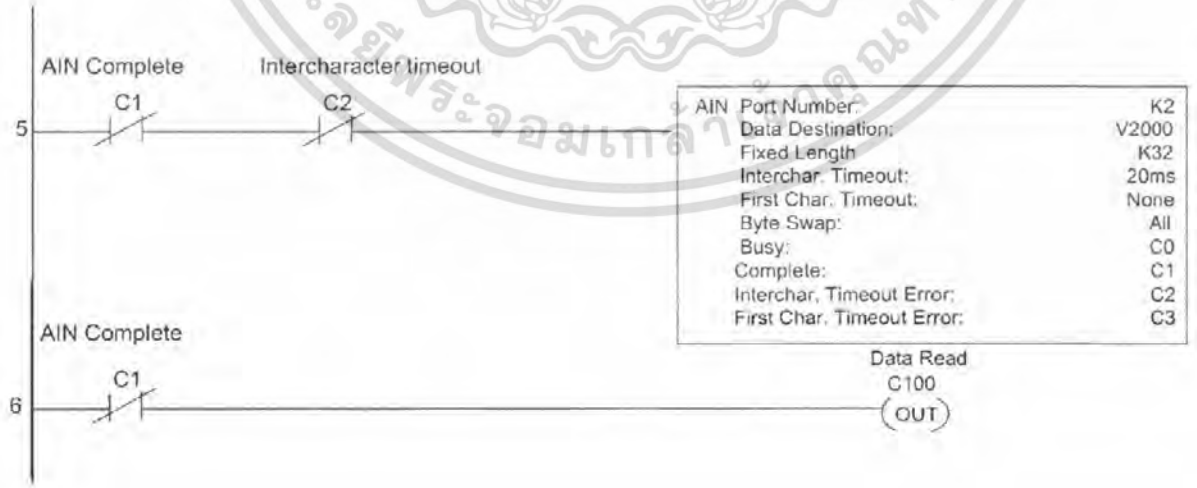
Parameter	
Data Destination	All V–memory
Fixed Length	K1–128
Bits: Busy, Complete, Timeout Error, Overflow	C0–3777

AIN Fixed Length Examples

Fixed Length example when the PLC is reading the port continuously and timing is not critical



Fixed Length example when character to character timing is critical



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AIN Variable Length Configuration:

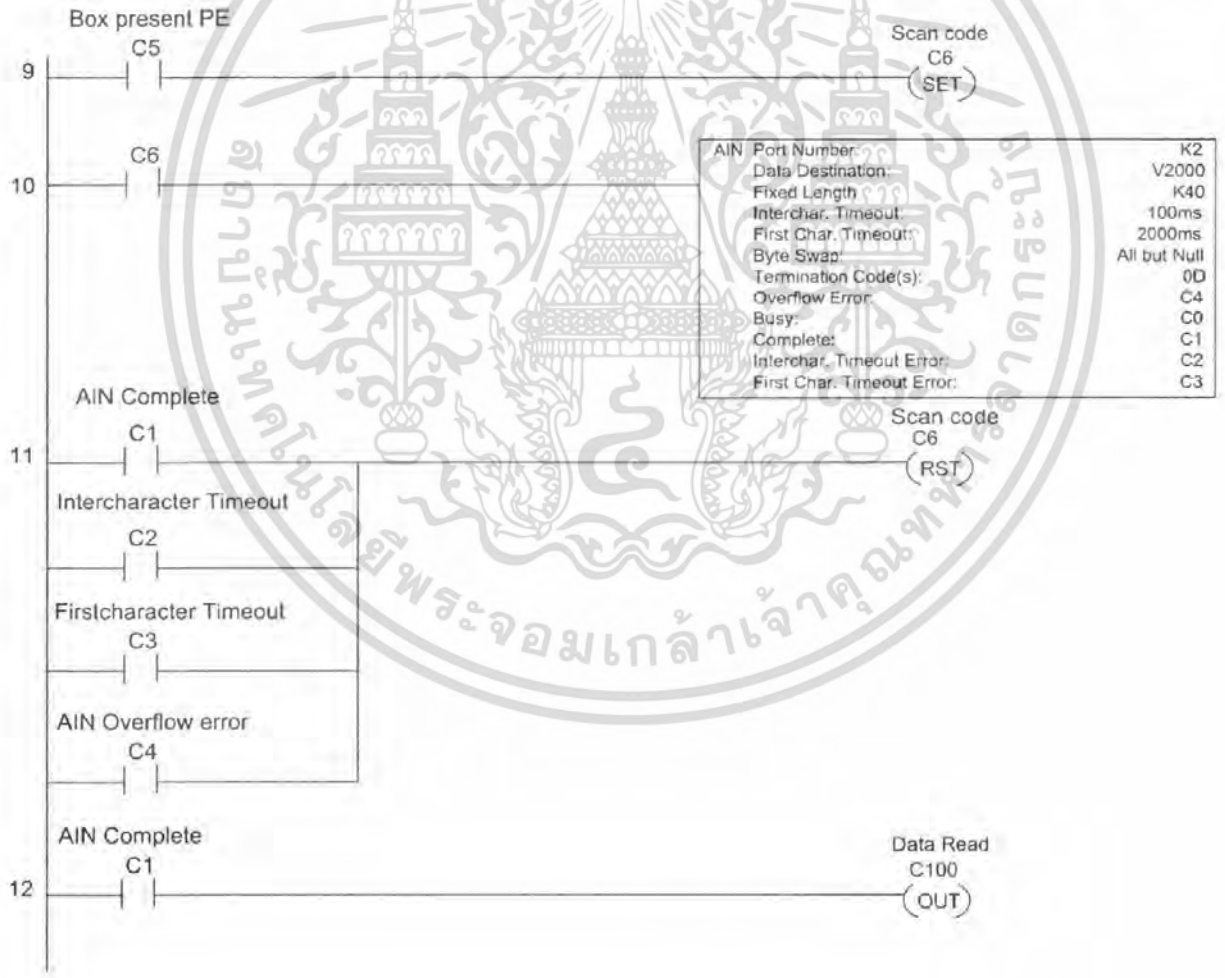
- Length Type: select Variable Length if the ASCII string length followed by termination characters will vary in length
- Port Number: must be DL06 port 2 (K2)
- Data Destination: specifies where the ASCII string will be placed in V–memory Maximum Variable Length: specifies, in bytes, the maximum length of a Variable Length ASCII string the port will receive
- Inter–character Timeout: if the amount of time between incoming ASCII characters exceeds the set time, the Timeout Error bit will be set. No data will be stored at the Data Destination V–memory location. The Timeout Error bit will reset when the AIN instruction permissive bits are disabled. 0ms selection disables this feature.
- First Character Timeout: if the amount of time from when the AIN is enabled to the time the first character is received exceeds the set time, the specified First Character Timeout bit will be set. The bit will reset when the AIN instruction permissive bits are disabled. 0ms selection disables this feature.
- Byte Swap: swaps the high–byte and low–byte within each V–memory register of the Variable Length ASCII string. See the SWAPB instruction for details.
- Termination Code Length: consists of either 1 or 2 characters. Refer to the ASCII table on the following page.
- Busy Bit: is ON while the AIN instruction is receiving ASCII data
- Complete Bit: is set once the ASCII data has been received up to the termination code characters. It will be reset when the AIN instruction permissive bits are disabled.
- Inter–character Timeout Error Bit: is set when the Character Timeout is exceeded. See Character Timeout explanation above.
- First Character Timeout Error Bit: is set when the First Character Timeout is exceeded. See First Character Timeout explanation above.
- Overflow Error Bit: is set when the ASCII data received exceeds the Maximum Variable Length specified.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parameter	
Data Destination	All V–memory
Fixed Length	K1–128
Bits: Busy, Complete, Timeout Error, Overflow	C0–3777

AIN Variable Length Example

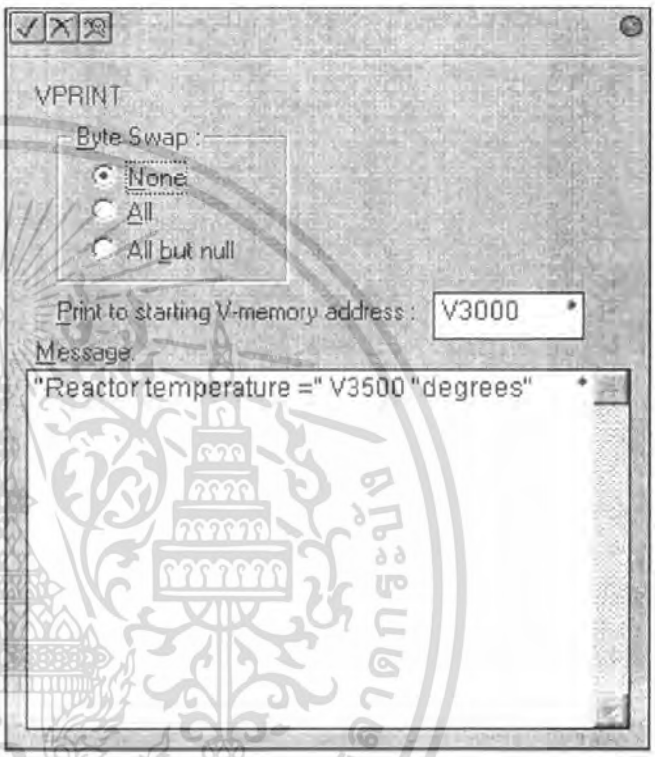
AIN variable length example used to read barcodes on boxes (PE = photoelectric sensor)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCII Print to V–memory (VPRINT)

The ASCII Print to V–memory instruction will write a specified ASCII string into a series of V–memory registers. Other features include Byte Swap, options to suppress or convert leading zeros or spaces, and _Date and _Time options for U.S., European, and Asian date formats and 12 or 24 hour time formats.



- Byte Swap: swaps the high–byte and low–byte within each V–memory register the ASCII string is printed to. See the SWAPB instruction for details.
- Print to Starting V–memory Address: specifies the beginning of a series of V–memory addresses where the ASCII string will be placed by the VPRINT instruction.
- Starting V–memory Address: the first V–memory register of the series of registers specified will contain the ASCII string's length in bytes.
- Starting V–memory Address +1: the 2nd and subsequent registers will contain the ASCII string printed to V–memory.

Parameter	DL06 Range
Print to Starting V–memory Address	All V–memory

VPRINT Time / Date Stamping– the codes in the table below can be used in the VPRINT ASCII string message to "print to V–memory" the current time and/or date.

#	Character code	Date / Time Stamp Options
1	_Date:us	American standard (month/day/2 digit year)
2	_Date:e European standard	(day/month/2 digit year)
3	_Date:a Asian standard	(2 digit year/month/day)
4	_Time:12	standard 12 hour clock (0–12 hour:min am/pm)
5	_Time:24	standard 24 hour clock (0–12 hour:min am/pm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VPRINT V-memory element – the following modifiers can be used in the VPRINT ASCII string message to "print to V–memory" register contents in integer format or real format. Use V-memory number or V-memory number with ":" and data type. The data types are shown in the table below. The Character code must be capital letters.

NOTE: There must be a space entered before and after the V-memory address to separate it from the text string. Failure to do this will result in an error code 499.

#	Character code	Description
1	none	16-bit binary (decimal number)
2	: B	4 digit BCD
3	: D	32-bit binary (decimal number)
4	: D B	8 digit BCD
5	: R	Floating point number (real number)
6	: E	Floating point number (real number with exponent)

Examples:

V2000 Print binary data in V2000 for decimal number

V2000 : B Print BCD data in V2000

V2000 : D Print binary number in V2000 and V2001 for decimal number

V2000 : D B Print BCD data in V2000 and V2001

V2000 : R Print floating point number in V2000/V2001 as real number

V2000 : E Print floating point number in V2000/V2001 as real number with exponent

The following modifiers can be added to any of the modifies above to suppress or convert leading zeros or spaces. The character code must be capital letters.

#	Character code	Description
1	S	Suppresses leading spaces
2	C0	Converts leading spaces to zeros
3	0	Suppresses leading zeros

Example with V2000 = 0018 (binary format)

V-memory Register with Modifier	Number of Characters			
	1	2	3	4
V2000	0	0	1	8
V2000:B	0	0	1	2
V2000:B0	1	2		

Example with V2000 = sp sp18 (binary format) where sp = space

V-memory Register with Modifier	Number of Characters			
	1	2	3	4
V2000	sp	sp	1	8
V2000:B	sp	sp	1	2
V2000:BS	1	2		
V2000:BC0	0	0	1	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VPRINT V-memory text element – the following is used for "printing to V–memory" text stored in registers. Use the % followed by the number of characters after V-memory number for representing the text. If you assign "0" as the number of characters, the function will read the character count from the first location. Then it will start at the next V-memory location and read that number of ASCII codes for the text from memory.

Example:

V2000 % 16 16 characters in V2000 to V2007 are printed.

V2000 % 0 The characters in V2001 to Vxxxx (determined by the number in V2000) will be printed.

VPRINT Bit element – the following is used for "printing to V–memory" the state of the designated bit in V-memory or a control relay bit. The bit element can be assigned by the designating point (.) and bit number preceded by the V-memory number or relay number. The output type is described as shown in the table below.

#	Data format	Description
1	none	Print 1 for an ON state, and 0 for an OFF state
2	: BOOL	Print "TRUE" for an ON state, and "FALSE" for an OFF state
3	: ONOFF	Print "ON" for an ON state, and "OFF" for an OFF state

Example:

V2000 . 15 Prints the status of bit 15 in V2000, in 1/0 format

C100 Prints the status of C100 in 1/0 format

C100 : BOOL Prints the status of C100 in TRUE/FALSE format

C100 : ON/OFF Prints the status of C00 in ON/OFF format

V2000.15 : BOOL Prints the status of bit 15 in V2000 in TRUE/FALSE format

The maximum numbers of characters you can VPRINT is 128. The number of characters required for each element, regardless of whether the :S, :C0 or :0 modifiers are used, is listed in the table below.

Element type	Maximum Characters
Text, 1 character	1
16 bit binary	6
32 bit binary	11
4 digit BCD	4
8 digit BCD	8
Floating point (real number)	3
Floating point (real with exponent)	13
V-memory/text	2
Bit (1/0 format)	1
Bit (TRUE/FALSE format)	5
Bit (ON/OFF format)	3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text element – the following is used for "printing to V–memory" character strings. The character strings are defined as the character (more than 0) ranged by the double quotation marks. Two hex numbers preceded by the dollar sign means an 8-bit ASCII character code. Also, two characters preceded by the dollar sign is interpreted according to the following table:

#	Character code	Description
1	$$	Dollar sign ($)
2	$"	Double quotation (")
3	$Lor $l	Line feed (LF)
4	$N or $n	Carriage return line feed (CRLF)
5	$P or $p	Form feed
6	$R or $r	Carriage return (CR)
7	$T or $t	Tab

The following examples show various syntax conventions and the length of the output to the printer.

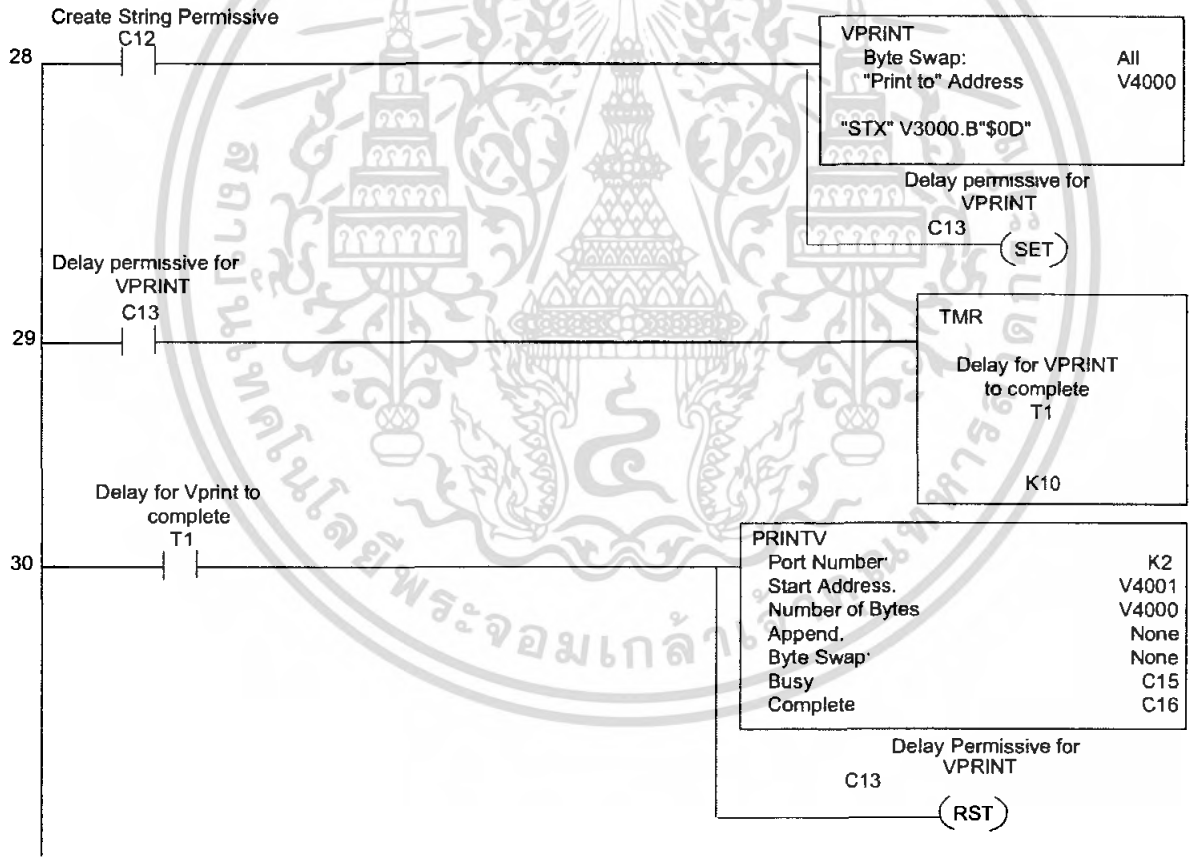
" "	Length 0 without character
"A"	Length 1 with character A
" "	Length 1 with blank
"$" "	Length 1 with double quotation mark
"RL"	Length 2 with one CR and one LF
"$0D$0A"	Length 2 with one CR and one LF
"$$"	Length 1 with one $ mark

In printing an ordinary line of text, you will need to include double quotation marks before and after the text string. Error code 499 will occur in the CPU when the print instruction contains invalid text or no quotations. It is important to test your VPRINT instruction data during the application development.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VPRINT Example Combined with PRINTV Instruction

The VPRINT instruction is used to create a string in V-memory. The PRINTV is used to print the string out of port 2.

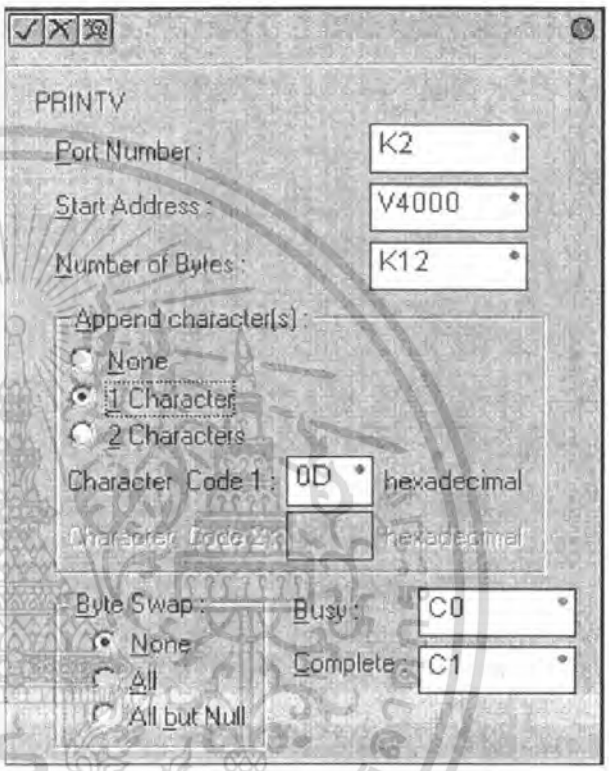


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCII Print from V–memory (PRINTV)

The ASCII Print from V–memory instruction will send an ASCII string out of the designated communications port from a specified series of V–memory registers for a specified length in number of bytes. Other features include user specified Append Characters to be placed after the desired data string for devices that require specific termination character(s), Byte Swap options, and user specified flags for Busy and Complete.

- Port Number: must be DL06 port 2 (K2)
- Start Address: specifies the beginning of series of V–memory registers that contain the ASCII string to print
- Number of Bytes: specifies the length of the string to print
- Append Characters: specifies ASCII characters to be added to the end of the string for devices that require specific termination characters
- Byte Swap: swaps the high–byte and low–byte within each V–memory register of the string while printing. See the SWAPB instruction for details.
- Busy Bit: will be ON while the instruction is printing ASCII data
- Complete Bit: will be set once the ASCII data has been printed and reset when the PRINTV instruction permissive bits are disabled.



See the previous page for an example using the PRINTV instruction.

Parameter	DL06 Range
Port Number	port 2 (K2)
Start Address	All V–memory
Number of Bytes	All V–memory or k1–128
Bits: Busy, Complete	C0–3777

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Module Specifications

The F0-04AD-1 Analog Input module offers the following features:

- The DL05 and DL06 will read all four channels in one scan.
- The removable terminal block makes it possible to remove the module without disconnecting the field wiring.
- Analog inputs can be used as process variables for the four (4) PID loops in the DL05 and the eight (8) PID loops in the DL06 CPUs.
- Field device burn–out is detected on all four channels when 4–20mA range is selected.
- On-board active analog filtering and RISC-like microcontroller provide digital signal processing to maintain precise analog measurements in noisy environments.

NOTE: *The DL05 CPU's analog feature for this module requires* ***Direct****SOFT32 Version 3.0c (or later) and firmware version 2.10 (or later). The DL06 requires* ***Direct****SOFT32 version V4.0, build 16 (or later) and firmware version 1.00 (or later). See our website for more information:* ***www.automationdirect.com****.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The following tables provide the specifications for the F0–04AD–1 Analog Input Module. Review these specifications to make sure the module meets your application requirements.

Input Specifications	
Number of Channels	4, single ended (one common)
Input Range	0 to 20 mA or 4 to 20 mA current (jumper selectable)
Resolution	12 bit (1 in 4096) for 0-20mA, scaled for 4-20mA
Step Response	25.0 mS (typ) to 95% of full step change
Crosstalk	-80 dB, 1/2 count maximum *
Active Low-pass Filtering	-3 dB at 40Hz (-12 dB per octave)
Input Impedance	125 Ohm ± 0.1%, 1/8 W current input
Absolute Maximum Ratings	-30 mA to +30 mA current input
Converter type	Successive approximation
Linearity Error (End to End)	± 2 counts maximum *
Input Stability	± 1 count *
Full Scale Calibration Error (Offset error not included)	± 10 counts maximum, @ 20mA current input*
Offset Calibration Error	± 5 counts maximum @ 4mA current input *
Maximum Inaccuracy	±.4% @ 25°C (77°F) ±.85% 0 to 60°C (32 to 140°F)
Accuracy vs. Temperature	±100 ppm/ °C maximum full scale calibration (including maximum offset change)
Recommended Fuse (external)	0.032 A Series 217 fast-acting current inputs

* One count in the specification table is equal to one least significant bit of the analog data value (1 in 4096).

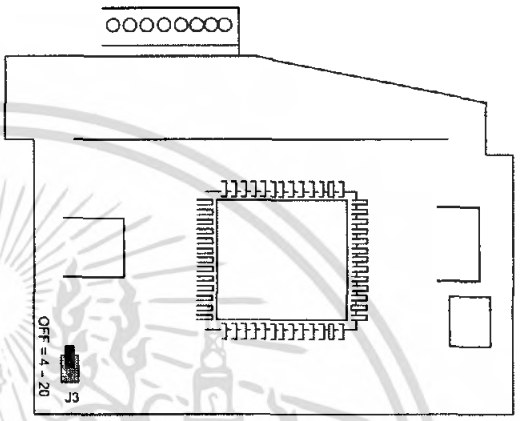
General Specifications	
PLC Update Rate	4 channels per scan
16-bit Data Word	12 binary data bits 2 channel ID bits 2 diagnostic bits
Operating Temperature	0 to 60° C (32 to 140°F)
Storage Temperature	-20 to 70°C (-4 to 158°F)
Relative Humidity	5 to 95% (non-condensing)
Environmental air	No corrosive gases permitted
Vibration	MIL STD 810C 514.2
Shock	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304
Power Budget Requirement	50 mA @ 5VDC (supplied by base)
Connector	Phoenix Mecano, Inc. Part No. AK1550/8-3.5 - green
Connector Wire Size	28 - 16 AWG
Connector Screw Torque	0.4 Nm
Connector Screwdriver Size	DN-SS1 (recommended)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Setting the Module Jumper

The position of jumper J3 determines the input signal level. You can choose between 4–20mA and 0–20mA. The module ships with the jumper not connecting the two pins. In this position, the expected input signal is 4–20mA. To select 0–20mA signals, use the jumper to cover both pins.

The default jumper setting selects a 4–20mA signal source. The default jumper setting does not connect the two pins.



WARNING: Before removing the analog module or the terminal block on the face of the module, disconnect power to the PLC and all field devices. Failure to disconnect power can result in damage to the PLC and/or field devices.

Connecting and Disconnecting the Field Wiring

Wiring Guidelines

Your company may have guidelines for wiring and cable installation. If so, you should check those before you begin the installation. Here are some general things to consider:

- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the transmitter source. *Do not* ground the shield at both the module and the source.
- Do not run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.
- Route the wiring through an approved cable housing to minimize the risk of accidental damage. Check local and national codes to choose the correct method for your application.

The F0–04AD–1 does not supply power to field devices. You will need to power transmitters separately from the PLC.

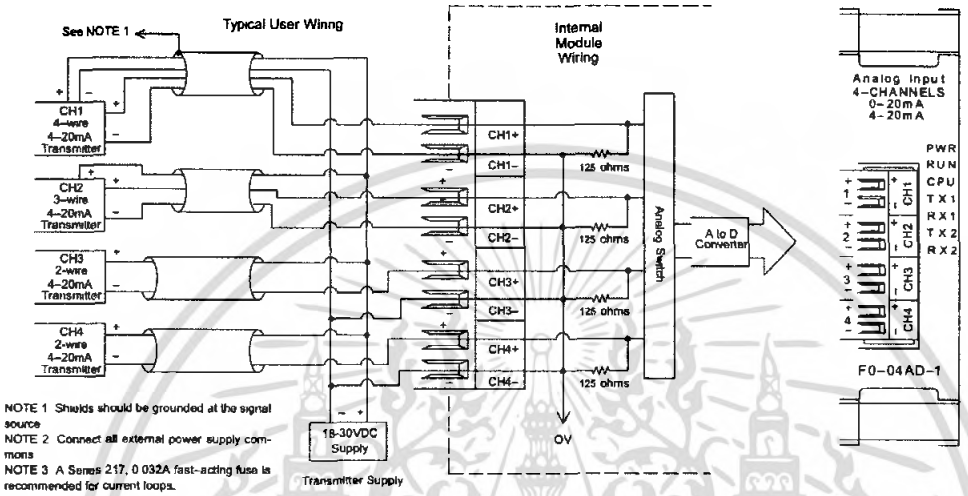
To remove the terminal block, disconnect power to the PLC and the field devices. Pull the terminal block firmly until the connector separates from the module.

You can remove the analog module from the PLC by folding out the retaining tabs at the top and bottom of the module. As the retaining tabs pivot upward and outward, the module's connector is lifted out of the PLC socket. Once the connector is free, you can lift the module out of its slot.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wiring Diagram

Use the following diagram to connect the field wiring. If necessary, the F0–04AD–1 terminal block can be removed to make removal of the module possible without disturbing field wiring.



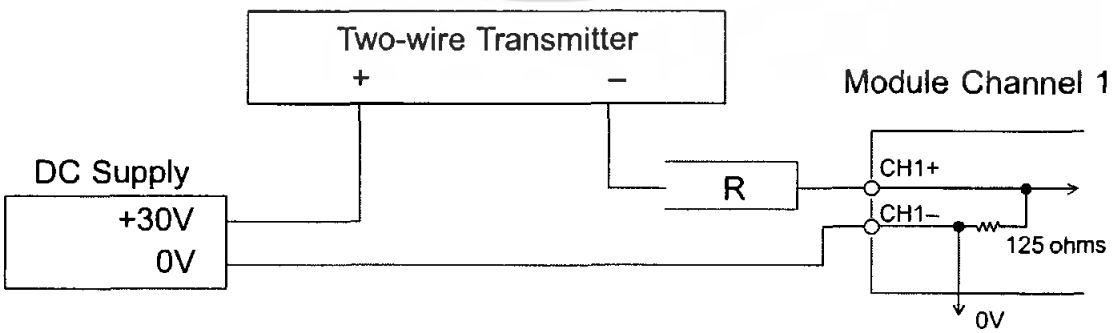
Current Loop Transmitter Impedance

Manufacturers of transmitters and transducers specify a wide variety of power sources for their products. Follow the manufacturer's recommendations.

In some cases, manufacturers specify a minimum loop or load resistance that must be used with the transmitter. The F0-04AD-1 provides 125 ohm resistance for each channel. If your transmitter requires a load resistance below 125 ohms, you do not have to make any changes. However, if your transmitter requires a load resistance higher than 125 ohms, you need to add a resistor in series with the module.

Consider the following example for a transmitter being operated from a 30 VDC supply with a recommended load resistance of 750 ohms. Since the module has a 125 ohm resistor, you need to add an additional resistor.

R = Tr – Mr	R = resistor to add
R = 750 – 125	Tr = Transmitter Requirement
R ≥ 625	Mr = Module resistance (internal 125 ohms)

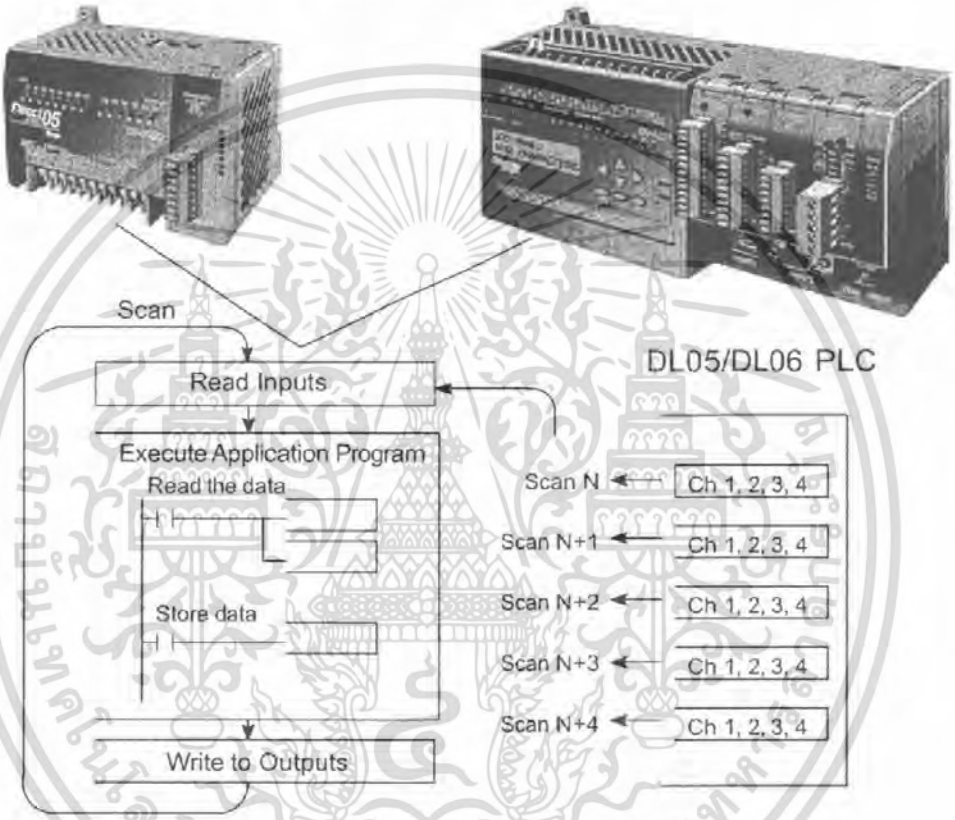


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Module Operation

Channel Scanning Sequence

The DL05 and DL06 will read all four channels of input data during each scan. Each CPU supports special V-memory locations that are used to manage the data transfer. This is discussed in more detail beginning in the section on "Special V–memory Locations".



Analog Module Updates

Even though the channel updates to the CPUs are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signals and converts each signal into a 12-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete control logic in the RLL program.

The module takes approximately 25 milliseconds to sense 95% of the change in the analog signal. For the vast majority of applications, the process changes are much slower than these updates.

NOTE: *If you are comparing other manufacturers' update times (step responses) with ours, please be aware that some manufacturers refer to the time it takes to convert the analog signal to a digital value. Our analog to digital conversion takes only a few microseconds. It is the settling time of the filter that is critical in determining the full update time. Our update time specification includes the filter settling time.*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special V-memory Locations

Formatting the Module Data

The DL05 and DL06 PLCs have special V-memory locations assigned to their respective option slots. These V-memory locations allow you to:

- specify the data format (binary or BCD)
- specify the number of channels to scan (4 channels for the F0–04AD–1)
- specify the V-memory locations to store the input data

DL05 Data Formatting

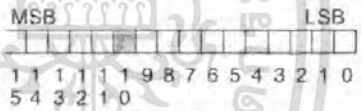
The table below shows the special V-memory locations used by the DL05 PLC for the F0–04AD–1.

Analog Input Module DL05 Special V-memory Locations	
Data Type and Number of Channels	V7700
Storage Pointer	V7701

Structure of V7700

Special V–memory location 7700 identifies that a F0–04AD–1 module is installed in the DL05 option slot and the data type to be either binary or BCD.

Loading a constant of 400 into V7700 identifies a 4 channel analog input module is installed in the DL05 option slot, and reads the input data values as BCD numbers.



Loading a constant of 8400 into V7700 identifies a 4 channel analog input module is installed in the DL05 option slot, and reads the input data values as binary numbers.

Structure of V7701

V7701 is a system V–memory location used as a pointer to a user V-memory location where the analog input data is stored. The V–memory location loaded into V7701 is an octal number identifying the first user V-memory location for reading the analog input data. This V–memory location is user selectable. For example, loading O2000 causes the pointer to write Ch 1's data value to V2000, Ch 2's data value to V2001, Ch 3's data value to V2002, and Ch 4's data value to V2003.

You will find an example program that loads appropriate values to V7700 and V7701 on page 3–9.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DL06 Data Formatting

Special V–memory locations are assigned to the four option slots of the DL06 PLC. The table below shows these V-memory locations which can be used to setup the F0–04AD–1.

Analog Input Module DL06 Special V-memory Locations				
Slot No.	1	2	3	4
Data Type and Number of Channels	V700	V710	V720	V730
Storage Pointer	V701	V711	V721	V731

Setup Data Type and Number of Channels

V–memory locations 700, 710, 720 and 730 are used to set the data format to be read in either binary or BCD, and to set the number of channels that will be active.

For example, the F0–04AD–1 is installed in slot 1. Loading a constant of 400 into V700 sets 4 channels active, and the input data value is read as a BCD number.

With the F0–4AD–1 in slot 1, loading a constant of 8400 into V700 sets 4 channels active, and the input data value is read as a binary number.

Storage Pointer Setup

V–memory locations 701, 711, 721 and 731 are special locations used as storage pointers. A V–memory address is loaded into this location as an octal number identifying the first user V-memory location for the analog input data. This V–memory location is user selectable. For example, loading O2000 causes the pointer to write Ch 1's data value to V2000, Ch 2's data value to V2001, Ch 3's data value to V2002, and Ch 4's data value to V2003.

You will find an example program that loads appropriate values to V700 and V701 beginning on page 3–10.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 *AT+CGDATA Enter Data State*

Description	*Command*	*Possible Responses*
Establish GPRS/packet domain connection	+CGDATA=[<L2p>,[cid[,cid>[,...]]]]	• CONNECT • ERROR • OK • ERROR
Show if the command is supported	+CGDATA=?	• +CGDATA: (list of supported <L2P>s) • OK • ERROR

Description:

Causes the MS to perform whatever actions are necessary to establish communication between the TE and the network using one or more GPRS/packet domain PDP types. This may include performing a GPRS/packet domain attach and one or more PDP context activations.

<L2P>	*Description*
PPP	Point-to-point protocol for a PDP such as IP

6.4 *AT+CGDCONT Define PDP Context*

Description	*Command*	*Possible Responses*
Select PDP context parameters	+CGDCONT=[<cid>[,<PDP_type>[,<APN>[,<PDP_addr>[,<pd1>[,...[,pdN]]]]]]]	• +CME ERROR: <err> • OK • ERROR
Show the current setting	+CGDCONT?	• +CGDCONT: <cid>,<PDP_type>,<APN>,<PDP_addr>,[,<pd1>[,...[,pdN]]] [<CR><LF>+CGDCONT: <cid>,<PDP_type>,<APN>,<PDP_addr>,[,<pd1>[,...[,pdN]]] [...]] • OK • ERROR
Show if the command is supported	+CGDCONT=?	• +CGDCONT: (range of supported <cid>s),<PDP_type>,,,[,(list of supported <pd1>s)[,...[,(list of supported <pdN>s)]]] [<CR><LF>+CGDCONT: (range of supported <cid>s),<PDP_type>,,,[,(list of supported <pd1>s)[,...[,(list of supported <pdN>s)]]] [...]]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	• OK • ERROR

Description:

Specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter, <cid>. When entered in the data is automatically saved to flash.

<cid>	*Description*
Integer type	(PDP Context Identifier) a numeric parameter which specifies a particular PDP context definition. The parameter is local to the TE-MS interface and is used in other PDP context-related commands. The range of permitted values (minimum value = 1) is returned by the test form of the command
1-10	Supported values. Ericsson specific

<PDP_type>	*Description*
IP	Internet Protocol (IETF STD 5)

<APN>	*Description*
String type	(Access Point Name) a string parameter which is a logical name that is used to select the GGSN or the external packet data network. If the value is null or omitted, then the subscription value will be requested

<PDP_address>	*Description*
String type	A string parameter that identifies the MS in the address space applicable to the PDP. If the value is null or omitted, then a value may be provided by the TE during the PDP startup procedure or, failing that, a dynamic address will be requested

<pdN>	*Description*
String type	Zero to N string parameters whose meanings are specific to the <PDP_type>

6.5 *AT+CGEREP GPRS Event Reporting*

Description	*Command*	*Possible Responses*
Set command	+CGEREP=[<mode> [,<bfr>]]	• +CME ERROR: <err> • OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 Data - TCP/IP

For more information on using these commands please refer to the application note 'Using AT commands to control on SEM modules'

7.1 *AT*E2IPA M2M IP Activate*

Description	*Command*	*Possible Responses*
Request Activation/deactivation of IP	AT*E2IPA=<Activate>,<cid>	• ERROR OK • E2IPA: <ErrNum> • ERROR
Displays current IP activation Status	AT*E2IPA?	• *E2IPA: <Active> OK • ERROR
Shows if the command is supported	AT*E2IPA=?	• *E2IPA: (list of supported <Activate>s), (list of supported < cid >s) OK • ERROR

Description:

This command allows the Activation of the IP Session on the module. Once activated with a previously stored context (see AT+CGDCONT & AT*ENAD) IP connections can be made, and the module will negotiat itself an IP Address.

Abortability:

This command may not be aborted

Defined values:

<Activate>	*Description*
0	Deactivate IP Session
1	Activate IP Session

<cid>	*Description*
Integer	Value range 0-10 for a reference to a PDP context identifier previously defined with AT+CGDCONT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<ErrNum>	*Description*
1	PDP Invalid Context
2	PDP Account Invalid
3	PDP Shutdown Failure
8	PDP Setup Cancelled
9	PDP Too Many Active Accounts
10	PDP Conflict with Higher Priority Account
11	PDP Too Many Active Users
12	PDP Non Existant Account
13	PDP Stop at User Request
14	PDP Authentication failed
15	PDP Bearer Failed Connect
16	PDP Remote Server Busy
17	PDP Remote Server Refused
18	PDP Bearer Busy
19	PDP Line Busy
20	PDP Unknown Error
255	PDP Invalid Parameter

<Active>	*Description*
0	IP Session Inactive
1	IP Session Active

Unsolicited Result codes:

None

Execution Time:

Executed immediately, not time critical.

Example! AT*E2IPA=?
*E2IPA= (0-1),(0-10)

OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT*E2IPA?
*E2IPA: 0 (IP Session Inactive)

OK

AT*E2IPA=1,1

OK (Activate PDP Context with id == 1)

AT*E2IPA?
*E2IPA: 1

OK (IP Session Active)

AT*E2IPA=1,2 (Try activating an invalid context)
*E2IPA: 002 (Error code denotes problem)

ERROR

7.2 AT*E2IPO M2M IP cOnnect/Open

Description	*Command*	*Possible Responses*
Request an IP Connection	AT*E2IPO=<IPType>, <IPAddr>,<IPPort>	• ERROR • CONNECT • *E2IPO: <ErrNum> ERROR
Displays current IP Connection Status	AT*E2IPO?	• *E2IPO: <Open> OK • ERROR
Shows if the command is supported	AT*E2IPO=?	• *E2IPO: (list of supported <IPType>s), (list of supported <IPAddr>s), (list of supported <IPPort>s) OK • ERROR

Description:

This command allows the unit to connect to another IP address and enter on line data mode thereby allowing data to be transferred to between the two IP addresses transparently. The command allows either TCP or UDP connections.

If an error occurs during the negotiation with the IP address the unit will return an ERROR with an associated error number, otherwise it will return CONNECT. If at any point the unit deems that the connection has been lost (see AT*E2IPS command)or it receives a disconnect message then the NO CARRIER response will be returned.

On line command mode can be entered by either dropping DTR or the +++at<cr> escape sequence, this does however need to be turned on (see AT*E2IPS command for more details).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abortability:

This command may not be aborted. (A connection can take upto 75 seconds to time out due to the unit having to wait for the packets that it has sent to time out, this is hard coded and cannot be adjusted at present)

Defined values:

<IPType>	*Description*
0	UDP (Connectionless)
1	TCP (Connection oriented)

<IPAddr>	*Description*
string	String type of the form "nnn.nnn.nnn.nnn" where n is a digit from 0-9. Standard IPv4 format for IP Address to connect to.

<IPPort>	*Description*
Integer	Range 0- 65535. Decimal representation of Port number to connect to (e.g. HTTP port is 80).

<ErrNum>	*Description*
1	No Connection
2	Connecting
3	Connected
4	Connection Busy
255	Other Error

<Open>	*Description*
0	IP connection Closed
1	IP connection Open

Unsolicited Result codes:

NO_CARRIER, OK

Execution Time:

Executed immediately, not time critical.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example!

AT*E2IPO?
*E2IPA: 0 (IP Connection closed)

OK

AT*E2IPO=1,"122.123.201.211",80

CONNECT (TCP Connection successful with specified host – now in online Data Mode (DCDasserted))

<<Data Transfer>>

NO_CARRIER (Connection closed remotely – back into off-line command mode (DCD Deasserted))

7.3 *AT*E2IPC M2M IP Close Connection*

Description	*Command*	*Possible Responses*
Request an IP Connection Close	AT*E2IPC	• ERROR • OK • *E2IPC: <ErrNum> ERROR
Shows if the command is supported	AT*E2IPC=?	• OK • ERROR

Description:

This command allows the closure of a currently open IP Connection. This command can only be called in on-line command mode during a currently active IP connection. If successful the connection will be closed, and it will not be possible to re-enter online data mode until another connection is established.

Abortability:

This command may not be aborted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Defined values:

<ErrNum>	*Description*
1	No Connection
2	Connecting
3	Connected
4	Connection Busy
255	Other Error

Unsolicited Result codes:

None

Execution Time:

Executed immediately, not time critical.

Note! It is recommended that if there is a problem at any time with the TCP/IP connection and the application is trying to re establish the link that this command be used to ensure that all of the internal module buffers have been cleared down.

7.4 AT*E2IPI M2M IP Info

Description	*Command*	*Possible Responses*
Request IP Information	AT*E2IPI=<InfoType>	• ERROR • *E2IPI: <IPAddr> OK
Shows if the command is supported	AT*E2IPI=?	• *E2IPI: list of supported <InfoType>s OK • ERROR

Description:

This command allows information about the current IP Status of the module to be reported. The command will only succeed when an IP Session is Active (see AT*E2IPA).

Abortability:

This command may not be aborted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Defined values:

<InfoType>	*Description*
0	IP Address of Module
1	IP Address of Primary DNS Server
2	IP Address of Secondary DNS Server

Unsolicited Result codes:

None

Execution Time:

Executed immediately, not time critical.

Example!
```
AT*E2IPI=0
*E2IPI: 120.211.22.102

OK
```

7.5 *AT*E2IPRH M2M IP Resolve Host*

Description	*Command*	*Possible Responses*
Request an IP Resolve Host	AT*E2IPRH=<URL>	• ERROR • *E2IPRH: <IPAddr> OK
Shows if the command is supported	AT*E2IPRH=?	• OK • ERROR

Description:

This command allows an IP lookup on a given URL string denoting an IP Host. The command will only succeed when an IP Session is Active (see AT*E2IPA) and if the string is 32 characters or less.

Abortability:

This command may not be aborted. Be aware this function can take up to 125-140 seconds to return if the URL does not exist.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Defined values:

<URL>	*Description*
String	URL of required server for which an IP lookup is required. The URL is encapsulated in quotes ("") and cannot exceed 32 characters.

Unsolicited Result codes:

None

Execution Time:

Executed immediately, not time critical.

Example! AT*E2IPRH="www.google.co.uk"
*E2IPRH: 123.123.123.123

OK

7.6 *AT*E2IPE M2M IP Last Error*

Description	*Command*	*Possible Responses*
Request a Error on Last IP Connection	AT*E2IPE or AT*E2IPE?	• ERROR • *E2IPE: <ErrType> OK
Shows if the command is supported	AT*E2IPE=?	• OK • ERROR

Description:

This command responds with the Error code for the last IP Connection. This is useful to determine the cause of the last IP Connection release – whether it was meant or due to an error.

Abortability:

This command may not be aborted.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TNS010i *TCP/IP* stack Chip

Description

TNS010i TCP/IP stack chip is a complete single turnkey product that enables the Internet connection. Not only does it reduce development time to market, the TNS010i allows several real-time efficient transmissions of data and commands between an 8/16/32-bit micro controller (MCU) and the core TCP/IP software engine. With no priori information about the Internet protocol or networking, this device allows the MCU to send and receive web pages, data, and commands in a form of scripting phrases through its built in TCP/IP protocol engine.

Executing only a few simple commands from an MCU through serial pins of TNS010i and you will be able to transmit and receive data to/from web application program. Each command sent across the TNS010i will invoke TNS010i to response back to an MCU with a message. This debugging information helps users keep track of every step during the connection attempt.

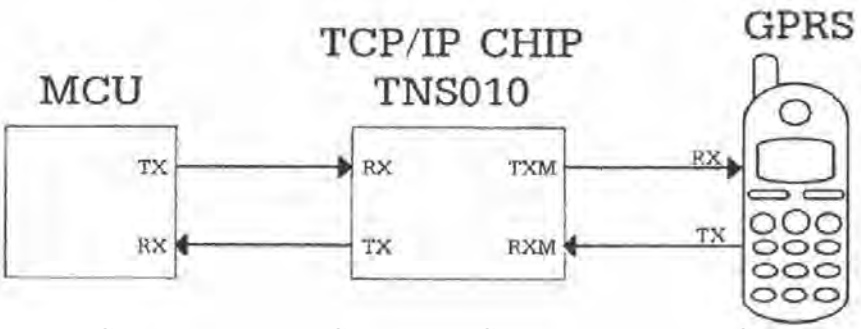
This device can connect to the GPRS phone directly with no additional components.
Even the MCU needs to set-up or read out the GPRS parameters, this device automatically bypasses those AT commands to the phone thus eliminating

- The need for two serial-port hardware on the MCU side. and
- A complex circuit for switching among the three modules: MCU, TNS010i, and GPRS.

Features

- Support basic protocols including TCP/IP, PPP, DNS ,HTTP,SMTP,POP3 and TCP/IP socket.
- Easy interface to GPRS phone or Modem.
- Input commands length as long as 1,000 bytes.
- No need for external RAM or ROM.
- Minimal components. Requires only one 18.432 MHz Xtal, and 3 Capacitors .
- 19,200 BPS Command communicates with micro controller (MCU)
- Information exchanges rate with GPRS is variable from 1,200 BPS to 57,600 BPS
- low power consumption
- 44-pin PLCC and LQFP package

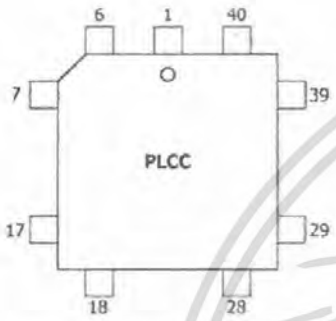
The Connection Diagram

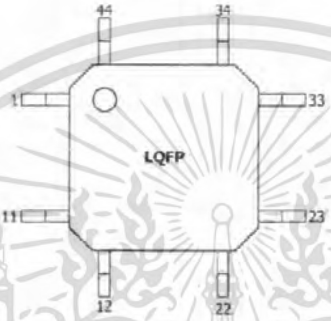


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

Symbol	Description	Min	Type	Max	Unit
VIL	Input Low Voltage	-0.5		0.9	V.
VIH	Input High Voltage	1.9		5.5	V.
VIH_	Input High Voltage XTAL1,RST	3.5		5.5	V.
Icc	Power supply current			60	mA
Vcc	Power supply voltage	4.75		5.25	V.





Pin Description

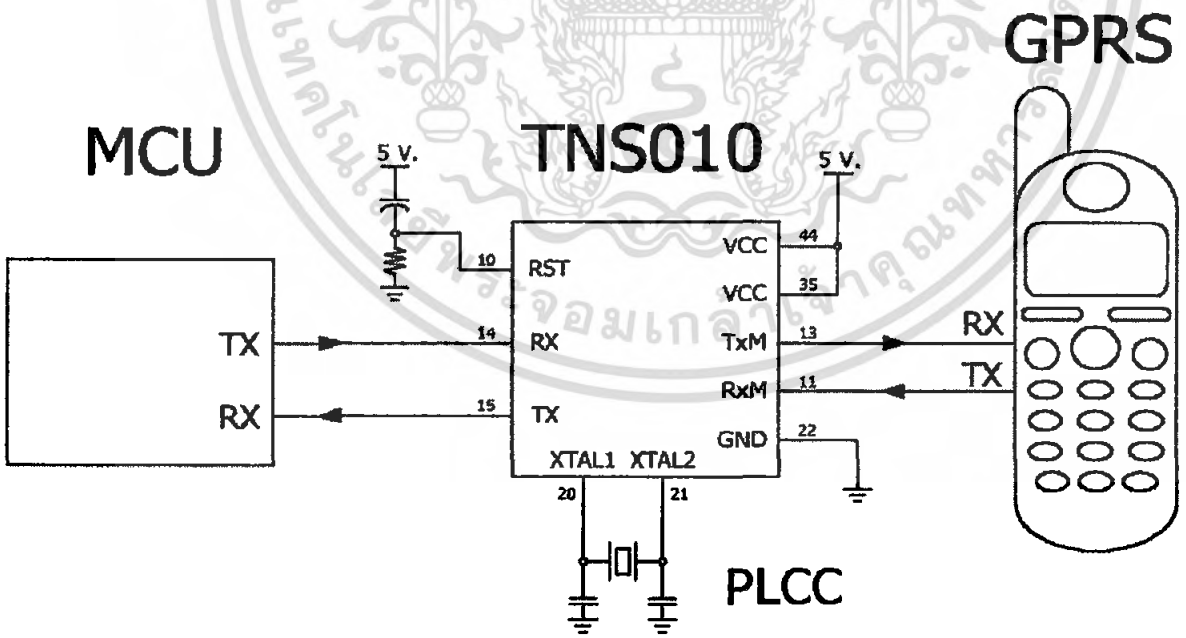
Symbol	PLCC	LQFP	In/Out	Name and Function
Vcc	44,35	38,29	I	Power Supply : power supply voltage for operation
Gnd	22	16	I	Ground : 0 V. reference
XTAL1	21	15	I	Crystal 1: 18.432 MHz. Input oscillator amplifier.
XTAL2	20	14	O	Crystal 2: 18.432 MHz. Output oscillator amplifier.
GPRS/modem	18	12	I	GPRS / Land line modem time out select
CD	2	42	I	Carrier detect
DTR	3	43	O	Data terminal ready
DSR	4	44	I	Data set ready
RTS	5	43	O	Request to send
CTS	6	44	I	Clear to send
RST	10	4	I	Reset : hold high at least 0.22 mS
TxM	13	7	O	Transmit to modem/GPRS phone
RxM	11	5	I	Receive from modem/GPRS phone
Tx	15	9	O	Transmit to controller
Rx	14	8	I	Receive from controller
LED-Connect	9	3	O	LED status shows connected to net
LED-Data trans	19	13	O	LED status shows data transferring

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP/IP Software Specifications

Maximum at+i command length (include at+i itself)	1000 bytes
Maximum TCP segment size	512 bytes
Serial RX buffer length (Ring buffer)	2,500 bytes
Serial TX buffer length	1,000 bytes
PAP	Support
CHAP	Not support
DNS	Support
HTTP	Support
FTP	Not support
SMTP	Support
POP3	Support
TCP/IP socket	Support

Typical testing circuit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Summary of Commands

- 19,200 BPS command baud rate (MCU (controller) ←→ TCP/IP Chip) N-8-1
- 1,200-57,600 BPS GPRS/modem baud rate. (TCP/IP Chip ←→ GPRS/Modem) N-8-1
- Dark printing characters are main commands and the gray are option commands

Chip test

```
Command:  at+i<CR>
Response:  +I_OK
```

Open a PPP connection

```
Command:  at+iopen<CR>
Response:
- Success  +I_OK

- Not success
          +I_ERROR 1 PPP_NOT_ESTABLISH(can't connect to PPP server)
          +I_ERROR 3 DIAL_NOT_SET      (no dial number set up)
          +I_ERROR 4 MODEM_NOT_RESPONSE (no modem response)
          +I_ERROR 7 PASSWORD_NOT_GOOD(invalid password)
```

Close a PPP connection

```
Command:   at+iclose<CR>
Response:
- Success  +I_OK

- Not success
          +I_ERROR 4 MODEM_NOT_RESPONSE(can't disconnect modem)
```

Display a current assigned IP address

```
Command:   at+iip<CR>
Response:  CLIENT_IP=x.x.x.x
           SERVER_IP=x.x.x.x
           +I_OK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Set up a dialing number

Command:	at+isetd=< ISP dialing number><CR>
Response:	**+I_OK**
Example:	at+isetd=*99***1#<CR>
	+I_OK

Display a dialing number

Command:	at+isetd=?<CR>
Response:	**<DIAL NUMBER>**
	+I_OK
Example:	at+isetd=? <CR>
	***99**1#**
	+I_OK

Set up DNS server's IP address (in case of ISP not automatic DNS assigned)

Command:	at+isetdns=<ip address><CR>
Response:	**+I_OK**
Example:	at+isetdns=203.155.33.1<CR>
	+I_OK

Display DNS server's IP address

Command:	at+isetdns=?<CR>
Response:	**<DNS SERVER IP ADDRESS>**
	+I_OK
Example:	at+isetdns=? <CR>
	203.155.33.1
	+I_OK

Set up PPP user name

Command:	at+isetuser=<username><CR>
Response:	**+I_OK**
Example:	at+isetuser=David<CR>
	+I_OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Display PPP user name

Command:	at+isetuser=?<CR>
Response:	**<USER NAME>**
	+I_OK
Example:	at+isetuser=? <CR>
	David
	+I_OK

Set up PPP password

Command:	at+isetpass=<password><CR>
Response:	**+I_OK**
Example:	at+isetpass=David password<CR>
	+I_OK

Display PPP password

Command:	at+isetpass=?<CR>
Response:	**<PASSWORD>**
	+I_OK
Example:	at+isetpass=?<CR>
	David password
	+I_OK

HTTP Request

Command:
at+ihttp://<url>[<path>][:<port>][<space></h></l></d>]<CR>

Response:

- Success

+I_OK
<WEB PAGE CONTENT.....
.....
.....
.....>

- Not success

+I_ERROR 2 DNS_IP_NOT_SET
(If specify domain name, must set a DNS IP address first : at+isetdns)
+I_ERROR 5 TCP_TIMEOUT *(server is busy)*
+I_ERROR 0 PPP_CLOSED *(PPP is not established yet, must open with at+iopen)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Note: 1. /h = display all **header** field.
2. /l = display only **length** field
3. /d = display only **date** field
4. default port number is 80

Example:
```
at+ihttp://www.google.com /l<CR>
+I_OK
Content-Length:2998
<html><head><meta http-equiv="content-type"
content="text/html; charset=windows-
874"><title>Google</title><style><!--
body,td,a,p,.h{font-family:;}
.h{font-size: 20px;}
.q{text-decoration:none; color:#0000cc;}
.....................
.....................
</html>
```

Example:
```
at+ihttp://www.myserver.com:123/doc.htm/d/l<CR>
+I_OK
Date: Mon, 29 Dec 2003 12:29:09 GMT
Content-Length: 3203
<HTML>.....................
.....................
.....................
</HTML>
```

How to request the next HTTP ?

After successfully retrieving the first web page, the successive request needs no further initial set-up commands.
Just send

```
Command:
at+ihttp://<url>[<path>][:<port>][<space></h></l></d>]<CR>
```

In case a connection is broken

1. Close the existing connection first with

```
Command:    at+iclose<CR>
```

2. Then re-open PPP connection with

```
Command:    at+iopen<CR>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Followed by requesting a web page

```
Command:
at+ihttp://<url>[<path>][:<port>][<space></h></l></d>]<CR>
```

TCP/IP SOCKET

1.OPEN TCP/IP SOCKET

```
Command:    at+isockopen=<url>:<port><CR>
Response:   +I_OK
            <RESPONSE/NONE>
            Note: <port> is require , no default value

  Example:  at+sockopen=mail.server.com:25<CR>
            +I_OK
            220 SMTP Service ready
```

2.SEND DATA ON TCP/IP SOCKET

```
Command:    at+isock=<data><CR>
Response:   +I_OK
            <RESPONSE/NONE>

  Example:  at+sock= HELO\r\n<CR>
            +I_OK
            250 HELLO <CR>
```

3.CLOSE TCP/IP SOCKET

```
Command:    at+isockclose<CR>
Response:   +I_OK
```

EMAIL SEND/RECEIVE (SMTP & POP3)

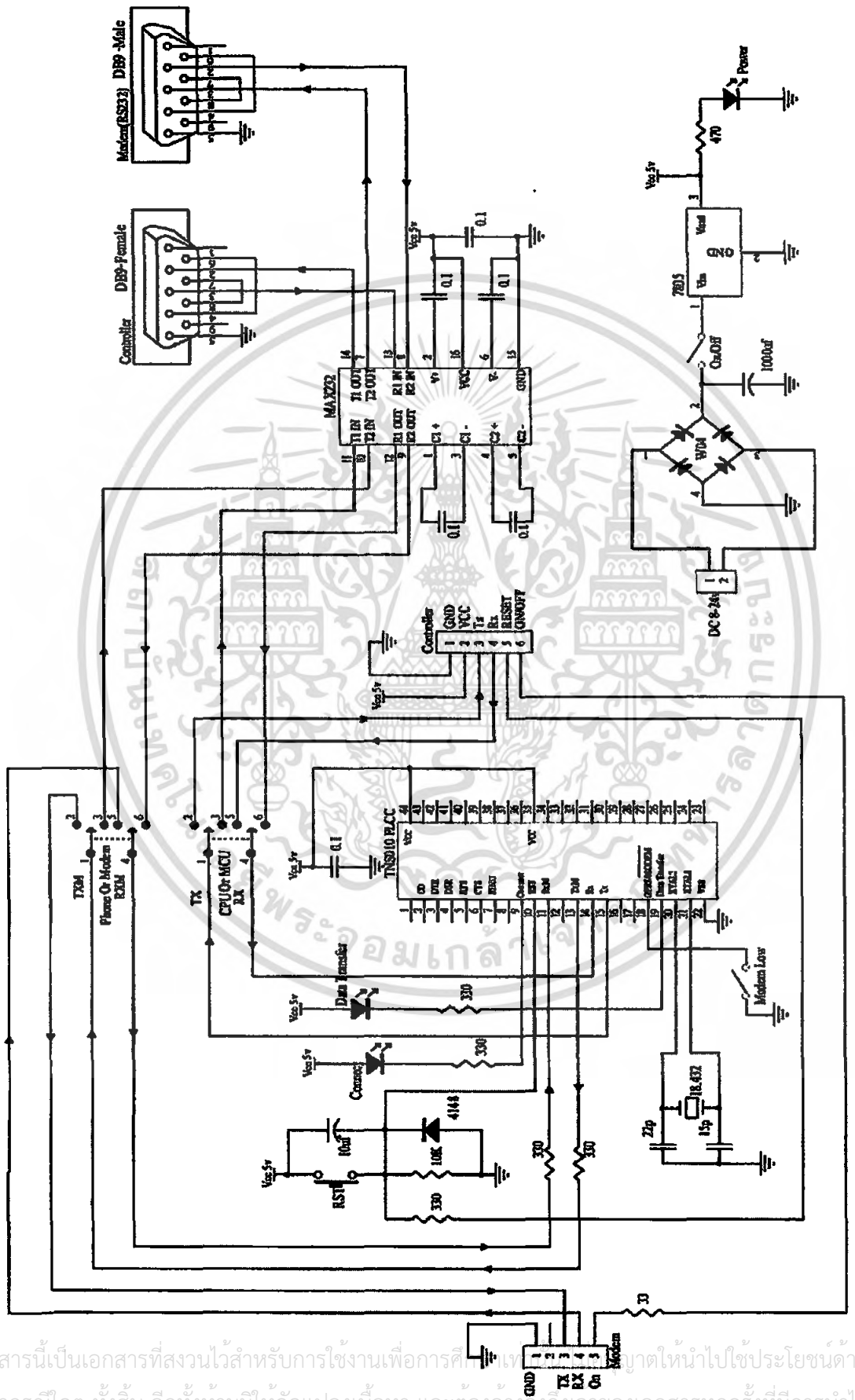
1.OPEN -> SEND -> CLOSE Mail

```
Command:
at+ismtpsend=<mailserver>[:<port>],<from>,<to>,<subject>,<data><CR>
Response:   +I_OK
             or
            +I+ERROR 10 MAIL_ERROR
Note:
     - <port> default value is 25.

Example:
             at+ismtpsend=mail.server.com,sender@aaa.com,
             recv@bbb.com,My subject,Hello how are you?<CR>
             +I_OK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEV-010i-B schematic



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้