สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต

LOCKER CONTROL SYSTEM VIA INTERNET

นายภูวดล ธงชัยพันธ์
นายวรพล ไหวดี
นางสาววัลยา ชุ่มบริบูรณ์
นางสาวสิริวิมล ดีรีวัน

เลขหมู่.....
เลขทะเบียน 119563
วัน,เดือน,ปี - 8 ธ.ค. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOCKER CONTROL SYSTEM VIA INTERNET

POOHWADON THONGCHAIPANT

WORAPON VAIDEE

WALLAYA CHUMBORIBOON

SIRIWIMOL KEEREEWAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต

LOCKER CONTROL SYSTEM VIA INTERNET

นักศึกษาผู้จัดทำ

นายภูวดล	ธงชัยพันธ์	รหัสนักศึกษา	50011196
นายวรพล	ไหวดี	รหัสนักศึกษา	50011368
นางสาววัลยา	ชุ่มบริบูรณ์	รหัสนักศึกษา	50011446
นางสาวสิริวิมล	คีรีวัน	รหัสนักศึกษา	50011690

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2553

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ อาจินต์ น่วมสำราญ	
รองศาสตราจารย์ ดร.วิทยา ทิพย์สุวรรณพร	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต			
	LOCKER CONTROL SYSTEM VIA INTERNET			
นักศึกษาผู้จัดทำ	นายภูวดล	ธงชัยพันธ์	รหัสนักศึกษา	50011196
	นายวรพล	ไหวดี	รหัสนักศึกษา	50011368
	นางสาววัลยา	ชุ่มบริบูรณ์	รหัสนักศึกษา	50011446
	นางสาวสิริวิมล	คีรีวัน	รหัสนักศึกษา	50011690
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ อาจินต์		น่วมสำราญ	
	รองศาสตราจารย์ ดร. วิทยา		ทิพย์สุวรรณพร	
ปีการศึกษา	2553			

บทคัดย่อ

ในบทความนี้ได้นำเสนอระบบควบคุมการเปิดและปิดตู้เก็บเอกสาร ผ่านทางอินเตอร์เน็ต โดยติดตั้งโซลินอยด์ที่ประตูตู้เก็บเอกสารด้านใน เพื่อทำหน้าที่เป็นกลอนล็อคแบบอิเล็กทรอนิกส์ ติดตั้งลิมิตสวิตซ์ 2 ตัว เพื่อทำหน้าที่ในการตรวจสอบสถานะการเปิดและปิดของตู้เก็บเอกสาร บอร์ดรีเลย์ที่ทำหน้าที่ในการสื่อสารข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 1280 กับกลอนโซลินอยด์และรับค่าสถานะจากลิมิตสวิตซ์ และยังเชื่อมต่อกับ switch hub และ server ที่ระบุไอพีแอดเดรสอยู่ เพียงแค่มีอุปกรณ์ที่สามารถเชื่อมต่อกับอินเตอร์เน็ต ก็จะสามารถควบคุมการเปิด-ปิดตู้เก็บเอกสารผ่านอินเตอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Locker Control System via Internet	
Authors	Mr. Poohwadon	Thongchaipant
	Mr. Worapon	Vaidee
	Miss Wallaya	Chumboriboon
	Miss Siriwimol	Keereewan
Thesis Advisor	Assoc.Prof. Arjin	Numsomran
	Assoc.Prof.Dr.Vittaya	Tipsuwanporn
Year	2010	

ABSTRACT

This article present a control system for opening and closing the locker via the internet. By installing solenoid latch inside the locker's doors which is electronic locking system. Each door is consisted of two limit switches in order to indicate the status of the locker. The relay board is used for the communication between the Arduino Mega 1280 microcontroller and solenoid for receiving the statate of the locker's door. The Arduino also connected to the server via switch hub. Therefore,if we have only an equipment which can be accessed to the internet. Then the locker can be closed or opened in any remote place.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับความเมตตาจาก รองศาสตราจารย์ อาจินต์ น่วมสำราญ และรองศาสตราจารย์ ดร.วิทยา ทิพย์สุวรรณพร ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านและเพื่อนๆนักศึกษา ทุกคนที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณพ่อและคุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญภาพ VIII

บทที่ 1 บทนำ 1
1.1 ความสำคัญของปริญญานิพนธ์ 1
1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1
1.3 ขอบเขตของปริญญานิพนธ์ 1
1.4 ขั้นตอนการศึกษา 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ระบบควบคุมตู้เก็บเอกสารผ่านทางอินเตอร์เน็ต 3
2.1 ระบบควบคุมตู้เก็บเอกสารผ่านทางอินเตอร์เน็ต 3
2.1.1 ตู้เอกสาร 4
2.1.2 โซลินอยด์ 4
2.1.3 ลิมิตสวิทซ์ 5
2.2 ชุดควบคุมการเปิด-ปิดตู้เก็บเอกสาร 5
2.2.1 ไมโครคอนโทรลเลอร์ ATMEGA 1280 5
2.2.2 โปรแกรมควบคุมการทำงานและแสดงผล 7
2.2.3 ภาษาที่ใช้ในการพัฒนา 26
2.2.4 ชุดบอร์ดรีเลย์ 27
2.3 ชุดติดต่อกับผู้ใช้งาน (Graphic User Interface : GUI) 28
2.3.1 ติดต่อผ่านเว็บบราวเซอร์ 28
2.4 เซอร์ฟเวอร์และฐานข้อมูล 29
2.4.1 อุปกรณ์ Server 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4.2 เว็บเซอร์ฟเวอร์อาปาเช (Apache) 30
2.4.3 ระบบฐานข้อมูล (MySQL)....31
2.4.4 พีเอชพี(PHP) 32
2.5 ระบบจ่ายไฟ....33
2.5.1 ระบบ Switching Power Supply 33

บทที่ 3 โปรโตคอลและการสื่อสารข้อมูล....36
3.1 กล่าวนำ....36
3.2 การสื่อสารข้อมูล....36
3.3 Ethernet และโปรโตคอล (TCP/IP)....37
3.4 TCP/IP และอินเทอร์เน็ต....37
3.5 สถาปัตยกรรมชุดโปรโตคอล TCP/IP....39
3.6 โปรโตคอล IP (Internetwork Protocal)....43
3.7 Time To Live (TTL) 44
3.8 โปรโตคอลในลำดับชั้นทรานสปอร์ต 45
3.9 UDP Protocal 47
3.10 การต่อ Ethernet LAN....47

บทที่ 4 การทำงานและการทดลองตู้เอกสาร....49
4.1 กล่าวนำ....49
4.2 ขั้นตอนการทดลอง....49
4.2.1 การทำงานของระบบ 49
4.2.2 การทดสอบระบบ....53
4.3 การใช้งานโปแกรม 55
4.3.1 การสั่งงานผ่านอินเทอร์เน็ต....55
4.4 ผลการทำงาน....56
4.5 สรุปผล 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 58

5.1 สรุปผลการดำเนินการ 58

5.2 ข้อเสนอแนะ 58

5.2.1 ระบบสำรองไฟ 58

5.2.2 การขยาย port I/O 64

5.3 แนวทางในการพัฒนา 64

บรรณานุกรม 65

ภาคผนวก 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 ตัวอย่างลักษณะของแต่ละหมายเลขพอร์ต 46

5.1 โอกาสความผิดพลาดของระบบไฟฟ้า 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต 3

2.2 ตู้เก็บเอกสารหรือสิ่งของ 4

2.3 โซลินอยด์ 4

2.4 ลิมิตสวิทซ์ 5

2.5 ไมโครคอนโทรลเลอร์ ATMEGA 1280 5

2.6 Arduino environment 8

2.7 Flow Chat ของโปรแกรม 23

2.8 โปรแกรม Adobe Flash Professional 24

2.9 โปรแกรม Adobe Dreamweaver CS5 25

2.10 โปรแกรม SSH Secure Shell Client 25

2.11 ชุดบอร์ดรีเลย์ 28

2.12 อุปกรณ์ Server 29

2.13 Web Server Diagram 31

2.14 Database Server Diagram 32

2.15 Switching Power Supply 33

2.16 องค์ประกอบพื้นฐานของสวิตชิ่งเพาเวอร์ซัพพลาย 35

3.1 RS-232 36

3.2 เครือข่ายอินเทอร์เน็ตในมุมมองทางกายภาพ (Physical Networks) ประกอบไปด้วยเครือข่ายมากมาย 38

3.3 แสดงเครือข่ายอินเทอร์เน็ตในมุมมองของ TCP/IP 39

3.4 เปรียบเทียบระหว่างแบบจำลอง OSI และสถาปัตยกรรมชุดโปรโตคอล TCP/IP 40

3.5 แสดงการจัดเรียงลำดับหมายเลขของแต่ละดาต้าแกรม 40

3.6 โปรโตคอล TCP ซึ่งเป็นโปรโตคอลแบบคอนเน็กชั่นโอเรียนเต็ด จะมีการสร้างคอนเน็กชั่นเพื่อเชื่อมต่อกับโฮสต์ปลายทาง ก่อนที่จะดำเนินการส่งข้อมูล 42

3.7 ไอพี ดาต้าแกรม (IP datagram)44

3.8 การติดต่อด้วยหมายเลขพอร์ตระหว่างเครื่องไคลเอนต์และเซิร์ฟเวอร์ 46

3.9 ตัวอย่างโปรโตคอลในลำดับชั้นแอปพลิเคชั่น 47

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.10 ลักษณะของการเข้าหัว RJ-45 แบบ cross....48
3.11 ลักษณะของการเข้าหัว RJ-45 แบบ Direct....48
4.1 ติดตั้งระบบล็อคแบบอิเล็กทรอนิกส์....50
4.2 การติดตั้งระบบตรวจสอบสถานะของตู้....50
4.3 โฟลชาร์ตการทำงานของโปรแกรม....51
4.4 การติดตั้งระบบควบคุมการเปิด ล็อกตู้เก็บเอกสาร....52
4.5 แสดงโฟลชาร์ตแสดงการทำงานของโปรแกรม....53
4.6 การต่อวงจรควบคุมการเปิดและล็อคตู้เก็บเอกสาร....54
4.7 การสั่งเปิดและล็อคตู้ผ่านคอมพิวเตอร์....54
4.8 การสั่งการเปิดและล็อคตู้เก็บเอกสารผ่านโปรแกรม Hyper terminal....55
4.9 การเปิดและล็อคตู้เก็บเอกสารผ่านทางอินเตอร์เน็ต....55
4.10 แสดงเว็บเพจที่ใช้ในการควบคุมการเปิดและปิดตู้เก็บเอกสาร....56
5.1 Standby Power Supply....61
5.2 On - line Protection UPS หรือ Line Interactive UPS....62
5.3 True on - line UPS, Double conversion type....63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

เราจะพบเห็นระบบตู้เก็บเอกสารที่วางอยู่ตามห้องสมุดหรือตู้รับฝากรองเท้า ที่วางอยู่ตามวัดหรือสถานที่สำคัญๆทางศาสนา ซึ่งจะมีเจ้าหน้าที่ประจำอยู่ในแต่ละที่เพื่ออำนวยความสะดวกในการรับฝากสิ่งของหรือรองเท้าจากนักท่องเที่ยว บางสถานที่อาจมีการเก็บค่าบริการในการรับฝากด้วย ระบบดังกล่าวถือว่าเป็นระบบที่ดีที่ป้องกันทรัพย์สินของนักท่องเที่ยวสูญหาย แต่ปัญหาคือแต่ละสถานที่จำเป็นจะต้องมีเจ้าหน้าที่คอยให้บริการอย่างน้อย 2-3 คน หรือผู้ที่นำของมาฝากอาจทำกุญแจสูญหายทำให้ไม่สามารถเปิดตู้รับสิ่งของที่นำมาฝากได้ จากปัญหาดังกล่าวข้างต้น จึงเป็นที่มาให้กลุ่มผู้วิจัยมีแนวคิดจัดทำระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต (Locker Control System via Internet) เพื่อลดปัญหาดังกล่าว และทำให้ระบบการจัดการตู้เก็บเอกสารหรือตู้รับฝากสิ่งของนั้นมีประสิทธิภาพมากยิ่งขึ้น โดยใช้เทคโนโลยีสารสนเทศที่มีอยู่ในปัจจุบัน เช่น ระบบอินเตอร์เน็ต ระบบโทรศัพท์เคลื่อนที่ ระบบเครือข่ายไร้สาย เป็นต้น เข้ามาช่วยในการบริหารจัดการ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ให้ศึกษาวิธีการ Lock แบบอิเล็กทรอนิกส์ของตู้เก็บเอกสาร
2. ให้ศึกษาเกี่ยวกับ Microcontroller Arduino Mega 1280
3. ให้ศึกษาเกี่ยวกับเครือข่ายไร้สาย
4. ให้ศึกษาเกี่ยวกับการ Interface และการเชื่อมต่อระยะไกล
5. ให้ศึกษาและออกแบบระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต

1.3 ขอบเขตของปริญญานิพนธ์

1. จัดทำตู้เก็บเอกสารโดยใช้ตัวล็อคแบบอิเล็กทรอนิกส์
2. เขียนโปรแกรมด้วย Microcontroller Arduino Mega 1280 เพื่อเชื่อมต่อกับระบบล็อค
3. เขียนโปรแกรม Web Base เพื่อแสดงผลระยะไกล
4. ออกแบบเทคนิคการเปิดและปิดตู้เก็บเอกสารอัตโนมัติ
5. จัดทำคู่มือและเอกสารประกอบการใช้งานระบบตู้เก็บเอกสารผ่านทางอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. วางแผนงานการทำ Project โดยกำหนดระยะเวลาในการดำเนินงาน
2. ออกแบบระบบการทำงานของตู้ล็อคเกอร์ซึ่งเป็นฮาร์ดแวร์ และศึกษาการเขียนโปรแกรมเพื่อควบคุมการทำงานของตู้ล็อคเกอร์ โดยใช้โปรแกรม Arduino ซึ่งเป็นโปรแกรมเฉพาะของไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุมการทำงานของฮาร์ดแวร์
3. จัดหาอุปกรณ์เพื่อจัดทำฮาร์ดแวร์
4. เขียนโปรแกรมควบคุมการทำงานของฮาร์ดแวร์
5. ทำ web server และจัดทำ database
6. รันโปรแกรมทดสอบระบบ
7. วิเคราะห์การทำงานของระบบที่ใช้ควบคุมฮาร์ดแวร์ เพื่อหาข้อผิดพลาดและทำการแก้ไขปรับปรุง
8. สรุปผลการทำงาน

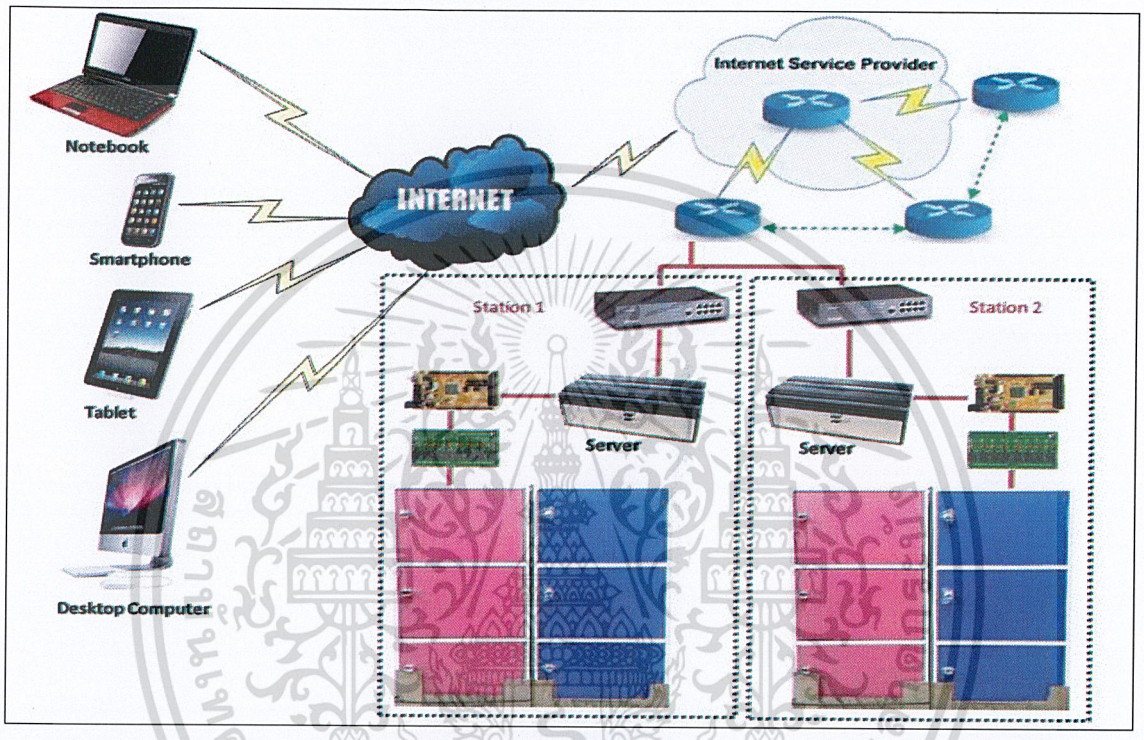
1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเขียนโปรแกรมด้วย Microcontroller เพื่อเชื่อมต่อกับระบบล็อค
2. สามารถจัดทำตู้เก็บเอกสารโดยใช้ตัวล็อคแบบอิเล็กทรอนิกส์
3. สามารถเขียนโปรแกรม Web Base เพื่อแสดงผลระยะไกล
4. สามารถออกแบบและจัดทำระบบการเปิดและปิดของตู้เก็บเอกสารอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต

2.1 ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต



ภาพที่ 2.1 ระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต

เราจะพบเห็นระบบตู้เก็บเอกสารที่วางอยู่ตามห้องสมุดหรือตู้รับฝากรองเท้าที่ วางอยู่ตามวัดหรือสถานที่สำคัญๆทางศาสนา ซึ่งจะมีเจ้าหน้าที่ประจำอยู่ในแต่ละที่เพื่ออำนวยความสะดวกในการรับฝากสิ่งของหรือรองเท้าจากนักท่องเที่ยว บางสถานที่อาจมีการเก็บค่าบริการในการรับฝากด้วย ระบบดังกล่าวถือว่าเป็นระบบที่ดีที่ป้องกันทรัพย์สินของนักท่องเที่ยวสูญหาย แต่ปัญหาคือแต่ละสถานที่จำเป็นจะต้องมีเจ้าหน้าที่คอยให้บริการอย่างน้อย 2-3 คน หรือผู้ที่นำของมาฝากอาจจะทำกุญแจสูญหายทำให้ไม่สามารถเปิดตู้รับสิ่งของที่นำมาฝากได้ จากปัญหาดังกล่าวข้างต้นจึงเป็นที่มาให้กลุ่มผู้วิจัยมีแนวคิดจัดทำระบบควบคุมตู้เอกสารผ่านทางอินเตอร์เน็ต เพื่อลดปัญหาดังกล่าว และทำให้ระบบการจัดการตู้เก็บเอกสาร หรือตู้รับฝากสิ่งของมีประสิทธิภาพมากยิ่งขึ้น โดยใช้เทคโนโลยีสารสนเทศที่มีอยู่ในปัจจุบัน เช่น ระบบอินเตอร์เน็ต ระบบโทรศัพท์เคลื่อนที่ ระบบเครือข่ายไร้สาย เป็นต้น เข้ามาช่วยในการบริหารจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

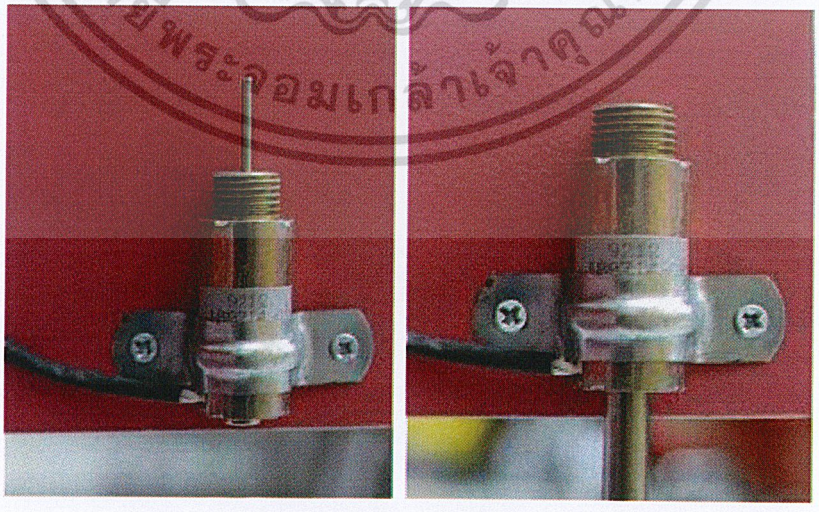
2.1.1 ตู้เก็บเอกสาร

ตู้หรือ Locker ในความหมายของปริญญานิพนธ์นี้หมายถึงตู้ในลักษณะตามรูปที่ 2.2 ซึ่งตู้หนึ่งใบจะมีช่องสำหรับเก็บเอกสารหรือสิ่งของจำนวนหลายช่อง

ภาพที่ 2.2 ตู้เก็บเอกสารหรือสิ่งของ

2.1.2 โซลินอยด์ (Solenoid)

โซลินอยด์เป็นกลอนอิเล็กทรอนิกส์ทำหน้าที่เปิดและปิดตู้เอกสาร ในการประยุกต์ใช้งานโซลินอยด์โดยการแปลงพลังงานไฟฟ้าเป็นพลังงานกลโดยตรง ซึ่งสัญญาณไฟฟ้าที่ป้อนเข้ามาทางขดลวดโซลินอยด์ ทำให้เกิดอำนาจแม่เหล็กดึงดูดแกนแม่เหล็กถาวรของโซลินอยด์ทำให้เกิดการเคลื่อนที่ขึ้น ดังนั้นจึงใช้หลักการเคลื่อนที่ของแกนแม่เหล็กถาวรในโซลินอยด์ในการใช้งาน ตัวอย่างเช่น เป็นขัดกลอนประตูของรถยนต์หรือประตูบ้าน การขับเคลื่อนกระเดื่องทำให้กลไกทำงานหรือหยุดทำงาน เป็นต้น

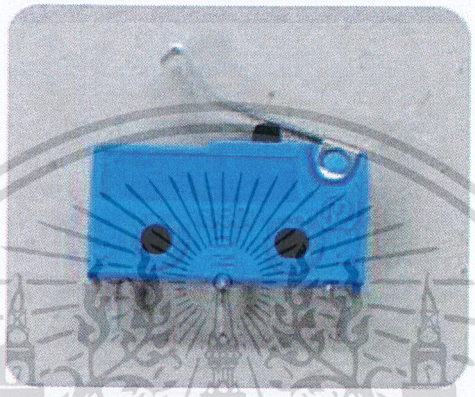


ภาพที่ 2.3 โซลินอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 ลิมิตสวิทซ์

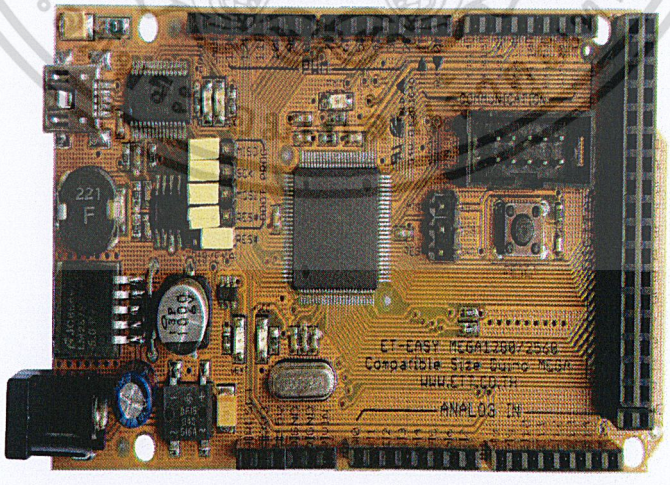
ลิมิตสวิตช์ (Limit Switch) หรือสวิตช์จำกัดระยะ เป็นสวิตช์ที่ทำงานโดยอาศัยการชนของวัตถุกับลูกล้อ (Roller) และเป็นผลให้หน้าสัมผัสที่ต่ออยู่กับก้านชนเปิดและปิด ตามจังหวะของการชน ซึ่งในโครงงานนี้ได้นำลิมิตสวิตซ์เข้ามาใช้เพื่อเป็นอุปกรณ์ตรวจสอบสถานะการเปิดและปิดของตู้เอกสาร แล้วส่งสัญญาณไปให้ไมโครคอนโทรลเลอร์ประมวลผลและส่งสถานะของตู้ไปเก็บยัง server



ภาพที่ 2.4 ลิมิตสวิทซ์

2.2 ชุดควบคุมการเปิด- ปิดตู้เก็บเอกสาร

2.2.1 ไมโครคอนโทรลเลอร์ ATMEGA 1280



ภาพที่ 2.5 ไมโครคอนโทรลเลอร์ ATMEGA 1280

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microcontroller ATMEGA 1280 เป็นชิพตระกูล AVR ของบริษัท Atmel รองรับการเขียนโปรแกรมภาษาซีของ Arduino โดยชิพมีหน่วยความจำแฟลชสำหรับเก็บการเขียนโปรแกรม 128 กิโลไบต์ มีหน่วยความจำแรม 8 กิโลไบต์ มี EPROM 4 กิโลไบต์ สำหรับใช้เป็นที่เก็บข้อมูลถาวรได้เมื่อไม่มีไฟเลี้ยง มีขานำเข้า/ส่งออก หรือ I/O สำหรับต่อใช้งานทั่วไป 86 ขา มี PWM (สำหรับควบคุมอัตราการหมุนของมอเตอร์) ที่สามารถกำหนดความละเอียดได้ระดับ 16 บิต จำนวน 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณแอนาล็อกเป็นสัญญาณดิจิตอล (ADC) 16 ช่องสัญญาณ

คุณสมบัติของไมโครคอนโทรลเลอร์ ATMEGA 1280

1. เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ประสิทธิภาพสูงแต่ใช้พลังงานต่ำ ในตระกูล AVR

2. สถาปัตยกรรมแบบ RISC

- มีชุดคำสั่ง 135 คำสั่ง และส่วนใหญ่คำสั่งเหล่านี้จะใช้เพียง 1 สัญญาณนาฬิกาในการประมวลผลคำสั่ง
- มีรีจิสเตอร์สำหรับใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว
- ทำงานได้สูงสุดที่ 16 ล้านคำสั่งต่อวินาที (MIPS) เมื่อใช้สัญญาณนาฬิกา 16 เมกกะเฮิร์ซ (MHz)

3. หน่วยความจำ

- หน่วยความจำแฟลชสำหรับโปรแกรมขนาด 128 กิโลไบต์ เขียน/ลบได้ 10,000 ครั้ง
- หน่วยความจำแบบ EEPROM ขนาด 4 กิโลไบต์ เขียน/ลบได้ 100,000 ครั้ง
- หน่วยความจำแรมชนิดเอสแรม (SRAM) ขนาด 8 กิโลไบต์
- เก็บข้อมูลได้กว่า 20 ปีที่อุณหภูมิ 85°C และกว่า 100 ปีที่อุณหภูมิ 25°C

4. มีระบบโปรแกรมตัวเองอยู่ในตัวชิพ

5. สามารถทำการอ่านขณะเขียนได้จริง โดยสามารถล็อกการทำงานได้เพื่อความปลอดภัยของซอฟต์แวร์

6. มีการเชื่อมประสานกับ JTAG (IEEE std. 1149.1 compliant)

7. คุณสมบัติการเชื่อมต่อกับอุปกรณ์ภายนอก

- มีตัวตั้งเวลาและตัวนับขนาด 8 บิต จำนวน 2 ตัว ที่สามารถแยกโหมดการทำงานจากกันได้ 2 โหมด คือ Prescalar และ Compare
- มีตัวตั้งเวลาและตัวนับขนาด 16 บิต จำนวน 4 ตัว ที่แยกโหมดการทำงานได้ 3 โหมดคือ Prescaler, Compare และ Capture
- มีตัวนับแบบเวลาจริง (Real Time Counter) ที่แยกวงจรกำหนดความถี่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มี PWM จำนวน 12 ช่องสัญญาณที่สามารถกำหนดความละเอียดได้ 16 บิต
- มีตัวแปลงสัญญาณแอนาล็อกให้เป็นดิจิตอลขนาด 10 บิต จำนวน 16 ช่องสัญญาณ
- มีพอร์ตสื่อสารอนุกรมที่สามารถกำหนดอัตราการรับ/ส่งได้จำนวน 4 พอร์ต
- เชื่อมประสานอนุกรมแบบ SPI ได้ทั้งการเป็นมาสเตอร์และสเลฟ (Master/Slave)
- มีการเชื่อมประสานแบบอนุกรมด้วยสายสัญญาณ 2 เส้น แบบส่งข้อมูลแบบเรียงไบต์ (Byte Oriented)
- มีตัวตั้งเวลาแบบวอตช์ด็อกที่สามารถกำหนดการทำงานได้ โดยสามารถแยกสัญญาณนาฬิกาได้ จากตัวชิพ
- มีตัวเปรียบเทียบสัญญาณแบบแอนาล็อกอยู่ในตัว
- มีการรองรับการขัดจังหวะและการเวก-อัพ (Wake-up) เมื่อมีการเปลี่ยนแปลงเกิดขึ้นกับขาของชิพ

8. คุณสมบัติพิเศษ

- มีระบบเริ่มระบบเมื่อมีการรีเซ็ตและมีระบบตรวจจับการเกิดบราวน์เอาท์ (Brown-out) ที่สามารถกำหนดการทำงานได้
- มีตัวตรวจหาความเที่ยงตรงของออสซิเลเตอร์อยู่ในตัว (Internal Calibrated Oscillator)
- มีแหล่งการขัดจังหวะทั้งภายในและภายนอก (External and Internal Interrupt Sources)
- มีโหมดการทำงานสลีป 6 แบบ คือ : Idle, ADC Noise Reduction, Power-save, Power-down, Standby, และ Extended Standby

9. I/O และตัวถัง

- มีขาของ I/O ที่สามารถกำหนดการทำงานได้ 86 ขา
- ตัวถังแบบ TQFP ชนิด 100 ขา

10. ช่วงอุณหภูมิที่ชิพทำงานได้ -40°C ถึง 85°C

11. การใช้พลังงาน

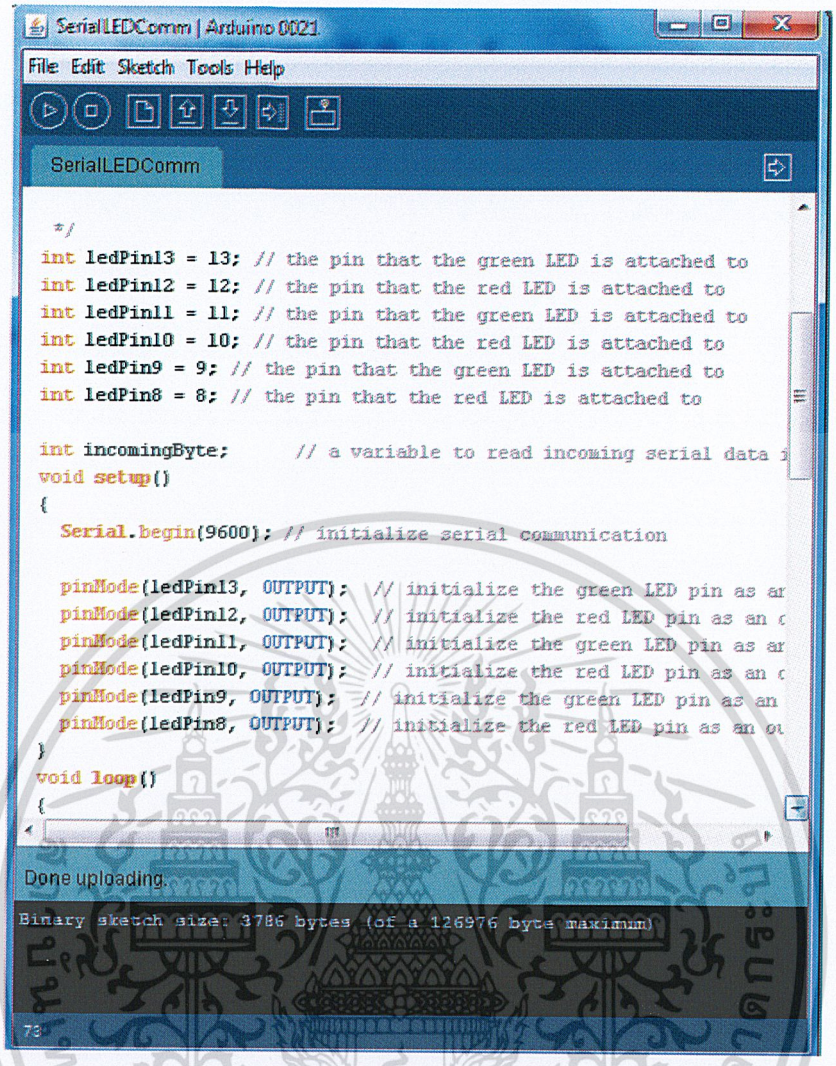
- โหมดการทำงาน: ที่ 1 MHz ต้องการแรงดัน 1.8 V กระแส 500 μA
- โหมดเพาเวอร์ดาวน์ (Power-down) ต้องการกระแสเพียง 0.1 μA ที่แรงดัน 1.8 V

2.2.2 โปรแกรมควบคุมการทำงานและแสดงผล

2.2.2.1 โปรแกรม Arduino

สำหรับภาษาที่ใช้ในการพัฒนาเป็นโปรแกรมภาษาซี โดยมีโปรแกรม Arduino Environment เป็นคอมไพเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 Arduino environment

การเขียนโปรแกรม Arduino นั้นประกอบด้วยส่วนหลักๆ 3 ส่วน คือ structure, values (ตัวแปรและค่าคงที่), และ functions. โดยที่ ภาษา Arduino นั้นพัฒนามาจากภาษา C/C++

- **STRUCTURE**

การเขียนโปรแกรม Arduino ประกอบด้วยส่วนย่อย 2 ส่วน คือ

1. void setup()

ฟังก์ชัน setup() จะถูกเรียกใช้เมื่อโปรแกรมเริ่มทำงาน ในส่วนนี้จะมีการกำหนดค่าเริ่มต้นของโปรแกรม กำหนดโหมดของให้กับขาต่างๆ เรียกใช้งานไลบรารี อื่นๆ ซึ่งจะถูกเรียกใช้งานเพียงครั้งเดียวเมื่อต่ออุปกรณ์กับแหล่งจ่ายไฟหรือมีการกดปุ่มรีเซต

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int buttonPin = 3;
void setup()
{
        beginSerial(9600);          //เปิดการใช้งานserial port, อัตราการส่งข้อมูลเท่ากับ 9600 bps
        pinMode(buttonPin, INPUT);      //กำหนดขาdigital เป็นอินพุต
}
void loop()
{
// ...
}
```

2. void loop()

หลังจากเรียกฟังก์ชัน setup() แล้ว ฟังก์ชัน loop() จะทำงานต่อและจะทำงานเป็นวนลูป ควบคุมการทำงานของบอร์ด Arduino

```
int buttonPin = 3;          // กำหนดค่าให้กับตัวแปร buttonPin (ขา Digital 3)
void setup() {
beginSerial(9600);
pinMode(buttonPin, INPUT);      // ภายใน loop จะตรวจสอบการกดปุ่มและส่งข้อมูลไปยัง serial
}
void loop() {
if (digitalRead(buttonPin) == HIGH)
serialWrite('H');
else
serialWrite('L');
delay(1000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- if

คำสั่ง if จะใช้ตรวจหาสถานะที่ต้องการว่าเกิดขึ้นหรือยัง เช่นค่า analog ที่อ่านได้มีค่าเกินจำนวนที่ตั้งไว้หรือยัง และทำงานตามคำสั่งในวงเล็บปีกกาถ้าสถานะเป็นจริง (TRUE) แต่ถ้าสถานะเป็นเท็จ (FALSE)โปรแกรมจะข้ามกลุ่มคำสั่งนั้นไป ดังตัวอย่าง

```
if (someVariable ?? value)
{
doSomething;
}
```

หมายเหตุ: ระมัดระวังการใช้ =' ในรูปคำสั่ง if (x=10) คำสั่งนี้ถูกต้องตามรูปแบบคือนำค่า 10 ไปใส่ในตัวแปร x และจะทำให้ผลลัพท์เป็นจริงตลอด อาจทำให้เกิดผลที่ไม่คาดคิด ดังนั้นควรใช้ '==' เช่น if (x==10) ซึ่งเพียงแต่ตรวจค่า x ว่ามีค่าเท่ากับ 10 หรือไม่ เท่านั้น จำไว้ว่า '=' หมายถึง 'เท่ากับ' ตรงกันข้ามกับ '==' ที่หมายถึง 'เท่ากับใช่หรือไม่'

- if... else

if เป็นการเพิ่มทางเลือก 'ถ้าไม่ใช่สถานะที่ต้องการ' ให้มีทางเลือกอีกทาง ตัวอย่างเช่น ถ้าคุณต้องการตรวจสอบ digital input และทำงานอย่างหนึ่งถ้าอินพุทเป็น HIGH หรือทำงานอีกอย่างถ้าอินพุทเป็น LOW คุณจะเขียนคำสั่งในลักษณะนี้

```
if (inputPin == HIGH)
{
doThingA;
}
else
{
doThingB;
}
```

else สามารถขยายเงื่อนไขต่อเนื่องออกไปได้อีก ทำให้เราสามารถทดสอบเงื่อนไขได้หลายเงื่อนไขในเวลาเดียวกัน เราสามารถตั้งเงื่อนไขสำหรับการทดสอบได้อย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่จำกัด แต่จำไว้ว่าเงื่อนไขทั้งหมดจะมีกลุ่มคำสั่งเดียวที่ทำงานเนื่องจากสถานะที่ทดสอบตรงตามเงื่อนไข

```
if (inputPin < 500)
{
doThingA;
}
else if (inputPin >= 1000)
{
doThingB;
}
else
{
doThingC;
}
```

หมายเหตุ: คำสั่ง if จะทดสอบเงื่อนไขในวงเล็บว่าจริงหรือเท็จ ซึ่งเงื่อนไขจะเป็นคำสั่งในภาษา C ดังตัวอย่างแรก if (inputPin == HIGH) ในตัวอย่างนี้ คำสั่ง if จะตรวจสอบว่า inputPin เป็น HIGH หรือ +5V

- for

คำสั่ง for จะใช้เพื่อทำกลุ่มคำสั่งในวงเล็บปีกกาหลายๆครั้งเป็นจำนวนที่ต้องการ ปกติจะใช้ตัวนับจำนวนเพิ่มจำนวนครั้งที่ทำงาน จนถึงจำนวนที่ต้องการแล้วจึงออกจาก loop เงื่อนไขของการทำงานใน loop จะกำหนดโดยคำสั่งที่ส่วนหัวของ loop ซึ่งมีสามส่วนแยกกันด้วยเซมิโคลอน

```
for (initialization; condition; expression)
{
doSomething;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของการเริ่มต้นค่าตัวแปรหรือที่เรียกว่า initialization จะทำงานก่อนและทำงานเพียงครั้งเดียว และเมื่อทำงานครบหนึ่งรอบ ส่วนทดสอบรอบการทำงานหรือส่วน condition จะถูกทดสอบถ้าเงื่อนไขเป็นจริง กลุ่มคำสั่ง expression จะถูกทำงานและเงื่อนไขจะถูกทดสอบอีกครั้ง ถ้าเงื่อนไขเป็นเท็จจึงจะออกจาก loop ตัวอย่างต่อไปนี้จะตั้งค่าเริ่มต้นให้ ตัวแปร i เป็นศูนย์ ทดสอบว่ามันยังมีค่าต่ำกว่า 20 และถ้าเป็นจริง เพิ่มค่า i ด้วย 1 และทำคำสั่งในวงเล็บปีกกา

```
for (int i=0; i < 20; i++) // declares i, test if less than 20, increment i by 1
{
degitalWrite(13, HIGH); // turn pin 13 on
delay(250); // pauses for 1/4 second
degitalWrite(13, LOW); //turn pin 13 off
delay(250); // pauses for 1/4 second
}
```

หมายเหตุ : คำสั่ง for loop ในภาษา C มีความซับซ้อนกว่า for loop ในภาษาคอมพิวเตอร์อื่นบางภาษารวมทั้งภาษาเบสิก บางส่วนของส่วนหัวอาจไม่มีกลุ่มคำสั่งแต่ก็ต้องใส่เครื่องหมายเซมิโคลอนไว้ คำสั่งของกลุ่มคำสั่งในส่วนหัวอันได้แก่ส่วน initialization, condition,และ expression จะเป็นคำสั่งภาษา C ที่มีส่วนตัวแปรที่ไม่เกี่ยวข้องกันก็ได้ ขึ้นอยู่กับการออกแบบของโปรแกรมเมอร์

- while

จะทำงานอย่างต่อเนื่องตลอดไป จนกระทั่งเงื่อนไขในวงเล็บจะเป็นเท็จ ดังนั้นจะต้องมีเหตุการณ์บางอย่างมาเปลี่ยนแปลงค่าตัวแปรที่ใช้ทดสอบ หรือการทำงานใน while loop อาจทำงานไปตลอดโดยไม่มีการออกจาก loop ขึ้นอยู่กับคำสั่งที่คุณเขียน เช่นเพิ่มค่าตัวแปรหรือทดสอบสถานะภายนอก เช่น ตรวจสอบ sensor เป็นต้น

```
while (someVariable ?? value) // test if less than 200
{
doSomething; // executes enclosed statements
someVariable++; // increments variable by 1
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- do... while

do loop จะทำงานและทดสอบสถานะการทำงานที่ด้านท้ายของ loop คล้ายกับการทำงานของ while loop ยกเว้นการตรวจสอบสถานะเพื่อออกจาก loop จะทำที่ส่วนท้ายของ loop ดังนั้น do loop จะทำงานอย่างน้อยหนึ่งครั้งเสมอ

```
do
{
doSomething;
} while (someVariable ?? value);
```

ตัวอย่างต่อไปนี้อ่านค่า readSensor() แล้วเก็บไว้ที่ตัวแปร 'x' หยุดเป็นเวลา 50 มิลลิวินาที และทำงานวนลูปจนกระทั่ง 'x' มีค่าน้อยกว่า 100

```
do
{
x = readSensor(); // assigns the value of readSensors() to x
delay(50); // pauses 50 milliseconds
} while (x < 100); // loop if x is less than100
```

- Further Syntax

1. วงเล็บปีกกาเปิด{ จะต้องตามด้วยวงเล็บปีกกาปิด }เสมอ ในโปรแกรมอาจมีวงเล็บเปิดและปิดหลายๆอันแต่จะต้องมีจำนวนเท่ากัน ถ้าวงเล็บเปิดปิดไม่เท่ากันจะทำให้เกิด compiler error และบางครั้งทำให้หาข้อผิดพลาดได้ยาก ในโปรแกรมที่มีความซับซ้อนสำหรับโปรแกรม Arduino environment จะมีความสามารถในการตรวจความสมดุลย์ของวงเล็บปีกกา โดยคลิกที่วงเล็บปีกกา หรือคลิกด้านข้างขวาของวงเล็บปีกกาที่ต้องการตรวจสอบ โปรแกรมจะแสดงวงเล็บปีกกาที่เป็นคู่ของมันทันที

2. SEMICOLON เครื่องหมายเซมิโคลอน ; จะต้องใช้ที่ส่วนท้ายของคำสั่งและใช้แยกส่วนของโปรแกรมออกจากกัน นอกจากนี้เซมิโคลอนยังใช้แยกส่วนของ for loop ออกเป็นส่วนๆด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- /*....*/ BLOCK COMMENTS

คำอธิบายโปรแกรม สามารถใส่ไว้ในเครื่องหมาย /*.....คำอธิบาย....*/ คำอธิบายที่อยู่ในเครื่องหมาย /*...*/ นี้ตัว compiler จะไม่นำมาประมวลผล ซึ่งจะมีประโยชน์ในการอธิบายความมุ่งหมายของคำสั่ง เพื่อให้เข้าใจการทำงานแต่ละส่วนของโปรแกรม ซึ่งจะเริ่มด้วยเครื่องหมาย /* และจบด้วยเครื่องหมาย */ และสามารถเขียน comment ได้หลายๆบรรทัดถ้าต้องการ

```
/* this is an enclosed block comment
don't forget the closing comment -
they have to be balanced!
*/
```

เนื่องจากการเขียนคำอธิบายไม่ได้เอาไปใช้ในโปรแกรม ทำให้ไม่สิ้นเปลืองหน่วยความจำโปรแกรม ดังนั้นควรเขียนคำอธิบายให้ชัดเจนและทำให้เป็นนิสัย อีกอย่างที่เครื่องหมาย comment มีประโยชน์คือใช้ตัดส่วนของโปรแกรมที่ไม่ต้องการออกไปสำหรับการตรวจสอบความผิดพลาดของโปรแกรม

- // LINE COMMENTS

การเขียนคำอธิบายโปรแกรมบรรทัดเดียวใช้เครื่องหมาย // เป็นเครื่องหมายเริ่มต้นและสิ้นสุดด้วยการขึ้นบรรทัดใหม่ และเช่นเดียวกับ block comment คือมันจะไม่เปลืองเนื้อที่โปรแกรมใช้งาน เนื่องจาก compiler จะไม่นำไปใช้ในโปรแกรม

```
// this is a single line comment
```

คำอธิบายโปรแกรมบรรทัดเดียวใช้บ่อยเพื่ออธิบายผลของกลุ่มคำสั่ง หรืออาจเป็นจุดเตือนในการเขียนคำสั่งต่อๆไป

- Arithmetic Operators

ตัวกระทำการทางคณิตศาสตร์หมายรวมถึง การบวก, ลบ, คูณ, และหาร ซึ่งมันจะให้ค่า ผลรวม, ผลต่าง, ผลคูณ, หรือผลการหาร ของสองตัวดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
y = y + 3;
x = x - 7;
i = j * 6;
r = r / 5
```

การทำงานของโปรแกรมจะเป็นการนำชนิดข้อมูล ตัวแปรของตัวดำเนินการมาใช้ ดังนั้น จากตัวอย่าง 9/4 ผลลัพธ์จะเท่ากับ 2 แทนที่จะเท่ากับ 2.25 เนื่องจาก 9 และ 4 เป็นตัวแปรจำนวนเต็มหรือ integer ซึ่งไม่สามารถเก็บค่าทศนิยมได้ และถ้าผลลัพธ์ที่ได้มีค่ามากกว่าที่ตัวแปรจะรับได้มันจะเกิดการ overflow ถ้ามีการคำนวณกับชนิดตัวแปรที่ต่างกัน ตัวแปรที่มีขนาดใหญ่กว่าจะถูกนำมาใช้ เช่น การบวก ลบเลขจำนวนเต็มกับเลขทศนิยม การบวกลบทศนิยมจะถูกนำมาใช้ควรเลือกตัวแปรที่มีขนาดที่ใหญ่พอที่จะเก็บ ผลลัพธ์การคำนวณ ตรวจสอบจุดที่ตัวแปรจะมีค่าย้อนกลับหรือ roll over เช่น 0-1 หรือ 0-32768 สำหรับการคำนวณที่เป็นเลขเศษส่วน ใช้ตัวแปรที่มีจุดทศนิยม แต่ควรระวังผลข้างเคียงถ้าตัวเลขมาก จะทำให้ความเร็วในการคำนวณช้าลง

หมายเหตุ : การ cast ตัวแปรคือ การเปลี่ยนชนิดตัวแปรจากชนิดหนึ่งไปยังอีกชนิดหนึ่ง คือ (int)myFloat เช่น i = (int)3.6 จะเป็นการแปลงค่าให้ i มีค่าเป็น 3

- Comparison Operators

```
== (equal to)
!= (not equal to)
< (less than)
> (greater than)
<= (less than or equal to)
>= (greater than or equal to)
```

- Boolean Operators

การตรวจสอบทาง logic จะใช้เปรียบเทียบตัวดำเนินการสองตัวและให้ผลลัพธ์ จริง หรือ เท็จ(TRUE OR FALSE) ขึ้นอยู่กับตัวดำเนินการ ตัวดำเนินการทางลอจิกมี 3 ตัว คือ AND, OR และ NOT ซึ่งใช้บ่อยในกลุ่มคำสั่ง if

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Compound Operators

การเขียนคำสั่งโปรแกรมแบบย่อหมายถึงการ ทำงานกับตัวแปรแล้วนำผลลัพธ์ที่ได้ไปเก็บในตัวแปรนั้น จะพบมากใน for loop การเขียนคำสั่งแบบย่อหรือ compound assignment ที่ใช้ทั่วไปคือ

```
x ++ // same as x = x + 1, or increments x by +1
x -- // same as x = x - 1, or decrements x by -1
x += y // same as x = x + y, or increments x by +y
x -= y // same as x = x - y, or decrements x by –y
x *= y // same as x = x * y, or multiply x by y
x /= y // same as x = x / y, or devides x by y
```

หมายเหตุ: ตัวอย่างเช่น x *= 3 จะคูณค่า x เป็นสามเท่าแล้วนำผลลัพธ์เก็บใน x

- Comparison operator

การเปรียบเทียบตัวแปรหรือค่าคงที่กับค่า อื่นๆ จะใช้บ่อยในกลุ่มคำสั่ง if เพื่อตรวจค่าว่ามีสถานะตรงตามที่ต้องการหรือไม่ ตัวอย่างต่อไปนี้เป็นคำสั่งเปรียบเทียบตัวแปรสองตัว

```
x == y // x is equal to y ??
x != y // x is not equal to y ??
x < y // x is less than y ??
x > y // x is greater than y ??
x <= y // x is less than or equal to y ??
x >= y // x is greater than or equal to y ??
```

- **VARIABLES**

Variable หรือ ตัวแปรเป็นชื่อและที่เก็บค่าของตัวเลขสำหรับใช้งานในโปรแกรม ตัวแปรมีค่าที่เปลี่ยนแปลงได้ซึ่งตรงกันข้ามกับ constant หรือค่าคงที่ ที่ไม่สามารถเปลี่ยนแปลงค่าได้ การใช้งานตัวแปรจะต้องมีการกำหนดชนิดและอาจกำหนดค่าเริ่มต้นให้กับตัวแปรนั้น ตัวอย่างด้านล่างนี้เป็นการกำหนดตัวแปรชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าพระนครเหนือ


```
inputVariable หลังจากนั้นนำค่าที่ได้จาก analog input pin 2 มาเก็บไว้
int inputVariable = 0; // declares a variable and assigns value of 0
inputVariable = analogRead(2); // set variable to value of anlog pin 2
```

ค่าของตัวแปร inputVariable จะเปลี่ยนแปลงได้ บรรทัดแรกจะกำหนดให้มันเป็นชนิด int หรือตัวเลขจำนวนเต็มโดยมีค่าเริ่มต้นเป็น 0 บรรทัดที่สองจะนำค่าจาก analog pin 2 มาเก็บไว้ ซึ่งจะทำให้ค่าของ pin 2 จะถูกเรียกใช้ในโปรแกรมได้เมื่อตัวแปรถูกสร้างขึ้น หรือใส่ค่าใหม่ให้มัน คุณสามารถตรวจสอบค่าของมันว่ามันมีค่าตามที่ต้องการ หรือคุณจะนำค่านั้นไปใช้เลยก็ได้ ตัวอย่างด้านล่างนี้เป็นสามตัวอย่างที่ใช้งานตัวแปร คือตรวจสอบค่าตัวแปร inputVariable ว่ามีค่าต่ำกว่า 100 หรือไม่ ถ้ามีค่าต่ำกว่า 100 ให้ใส่ค่า 100 ให้กับตัวแปร หลังจากนั้นตั้งค่าหน่วงเวลาตามค่าตัวแปร inputVariable ซึ่งก็คือค่า 100 เป็นอย่างต่ำ

```
if (inputVariable < 100) // test variable if less than 100
    {
    inputVariable = 100; // if true assigns value of 100
    }
delay(inputVariable); // uses variables as delay
```

หมายเหตุ: ตัวแปรควรจะมีชื่อที่สื่อถึงหน้าที่ใช้งาน เพื่อให้โปรแกรมสามารถอ่านได้ง่าย ชื่อตัวแปรเช่น tiltSensor หรือ pushButton จะช่วยให้โปรแกรมเมอร์หรือคนอื่นที่อ่านโปรแกรมสามารถเข้าใจได้ว่าตัวแปรนั้นเอาไว้ทำอะไร คุณสามารถตั้งชื่ออะไรก็ได้ที่ไม่ตรงกับชื่อเฉพาะที่ใช้ใน Arduino language

1. Constants

ในภาษา Arduino มีค่าที่กำหนดไว้แล้วไม่มาก เราเรียกค่าพวกนี้ว่า constant หรือค่าคงที่ซึ่งมีไว้เพื่อให้โปรแกรมอ่านง่ายค่าคงที่เหล่านี้จะแบ่งออกได้หลายกลุ่ม

- true/false

ค่าเหล่านี้เป็นค่าคงที่ boolean ซึ่งบอกสถานะระดับลอจิก FALSE หมายถึง 0(ศูนย์) ในขณะที่ TRUE จะหมายถึง 1 หรืออะไรก็ได้ที่ไม่ใช่ศูนย์ ดังนั้นในทางลอจิกแล้ว -1, 2, -200 จะหมายถึง TRUE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119563

```
if (b == TRUE);
{
doSomething;
}
```

- high/low

ค่าคงที่เหล่านี้แสดงถึงระดับลอจิกที่ขาไอซีว่าเป็น HIGH หรือ LOW และใช้เมื่อมีการอ่านหรือเขียนไปที่ขาไอซี HIGH จะแทนระดับลอจิก 1, ON, หรือ 5 volts ในขณะที่ LOW คือระดับลอจิก 0, OFF, หรือ 0 volts

2 variable declaration

ตัวแปรที่จะนำไปใช้งานได้จะต้องถูกตั้งค่า ตัวแปรก่อนนำไปใช้ การตั้งค่าตัวแปรหมายถึงการระบุชนิดตัวแปร ให้เป็น int, long, float เป็นต้น การกำหนดชื่อให้ตัวแปรมีชื่อเรียก และอาจมีการกำหนดค่าเริ่มต้นให้กับตัวแปร ซึ่งการตั้งค่านี้จะทำเพียงครั้งเดียวในโปรแกรม แต่ค่าของตัวแปรจะเปลี่ยนแปลงไปตามการทำงานของโปรแกรม ตัวอย่างต่อไปนี้ตั้งค่าตัวแปร input Variable ให้เป็นตัวแปรจำนวนเต็ม int, หรือ integer, และใส่ค่าเริ่มต้นให้เป็นศูนย์ ซึ่งเป็นการตั้งค่าตัวแปรแบบง่าย

```
int inputVariable = 0;
```

ตัวแปรสามารถตั้งค่าในแบบระบุตำแหน่งในหน่วยความจำ

3 variable scope

ตัวแปรสามารถตั้งค่าตอนเริ่มต้นโปรแกรมก่อน void setup() หรือตั้งค่าตัวแปรภายในฟังชั่น และบางครั้งก็ตั้งค่าตัวแปรภายในกลุ่มคำสั่ง for loop ซึ่งการตั้งค่าตัวแปรในแบบต่างๆ มีผลถึงขอบเขตการใช้ตัวแปร หรืออีกนัยหนึ่งการที่โปรแกรมจะสามารถใช้ตัวแปรนั้น ตัวแปรแบบ global เป็นตัวแปรที่โปรแกรมมองเห็นและใช้งานได้จากทุกฟังชั่นและทุกกลุ่มคำสั่งในโปรแกรม ตัวแปรนี้จะตั้งค่าที่ตอนเริ่มต้นโปรแกรมก่อน setup() function ตัวแปรแบบ local เป็นตัวแปรที่ตั้งค่าภายในฟังชั่นหรือภายในกลุ่มคำสั่ง for loop ตัวแปรนี้จะมองเห็นและใช้งาน ได้เฉพาะภายในฟังชั่นที่มันตั้งค่า ชื่อตัวแปรแบบนี้อาจมีชื่อซ้ำกันในแต่ละฟังชั่น แต่ในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังชั่นแต่ละตัวจะใช้ตัวแปรตัวนั้นเฉพาะที่ตั้งค่าภายในตัวมัน เท่านั้น ตัวอย่างด้านล่างนี้แสดงการตั้งค่าตัวแปรและขอบเขตการใช้งานของตัวแปร

```
int value; // 'value' is visible to any function
void setup();
{
// no setup neededd
}
void loop()
{
for (int i=0; i<20;) // 'i is only visible inside the for - loop
{
i++;
}
float f; // 'f' is only visible inside loop
}
```

4 Data Types

- Byte

ตัวแปร byte เก็บตัวเลข 8 bit ไม่มีทศนิยม มีค่า 0 – 255

- int

integer เป็นตัวแปรพื้นฐานที่เก็บตัวเลขโดยไม่มีจุดทศนิยม และเก็บค่า 16 bit มีค่าระหว่าง 32,767 ถึง -32,768

- long

เป็นตัวแปรจำนวนเต็มแบบขยายโดยไม่มีจุดทศนิยม เก็บค่าแบบ 32 bit มีค่าระหว่าง 2,147,483,647 ถึง -2,147,483,648

- float

ตัวแปรชนิด floating-point หรือตัวแปรที่มีจุดทศนิยม ตัวแปรนี้มีค่ามากกว่าค่าของตัวแปรจำนวนเต็ม โดยใช้เนื้อที่เก็บ 32 bit มีค่าระหว่าง 3.4028235E+38 ถึง -3.4028235E+38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- arrays

ตัวแปร arrays หรือตัวแปรหลายมิติเป็นตัวแปรที่สามารถเข้าถึงได้ด้วยค่าตัวชี้หรือ index ค่าตัวแปรใน array อาจเรียกใช้โดยระบุชื่อ array และระบุตัวชี้ index number ตัวแปร array จะมี index เริ่มต้นจาก 0 ตัวแปร array จะต้องตั้งค่า ก่อนจะนำไปใช้งาน และอาจกำหนดค่าเริ่มต้นหรือไม่ก็ได้

ตัวแปร arrays ใช้บ่อยใน for loops, ซึ่งจะมีการเพิ่มค่าตัวนับ และใช้ตัวนับเป็นตัวชี้ตำแหน่งของ arrays แต่ละตัว ตัวอย่างต่อไปนี้เป็นการใช้ array ในการสั่ง LED ติดดับ โดยใช้ for loop ตัวนับจะเริ่มจาก 0 เขียนค่าที่ได้จาก array ตำแหน่งที่ 0 ของ array flicker[] ในที่นี้คือค่า 180 ไปยัง PWM pin 10 หยุดเป็นเวลา 200 ms แล้วจึงย้ายไปเอาค่าในตำแหน่งถัดไปของ array

```
int ledPin = 10; // LED on pin 10
byte flicker[] = {180, 30, 255, 200, 10, 90, 150, 60}; // array of 8 different values
viod setup()
{
pinMode(ledPin, OUTPUT); // sets OUTPUT pin
}
viod loop()
{
for(int i=0; i<7; i++) // loop equals number of values in array
{
analogWrite(ledPin, flicker[i]); // write index value
delay(200);
}
}
```

- **FUNCTION**

ฟังชั่นเป็นกลุ่มคำสั่งที่มีชื่อเรียก และกลุ่มคำสั่งซึ่งจะทำงานเมื่อฟังชั่นถูกเรียกใช้ นอกจากฟังชั่น void setup() และ void loop() แล้วยังมีฟังชั่นอื่นๆที่พร้อมใช้งาน(build-in) ซึ่งจะได้อธิบายต่อไป ฟังชั่นที่ทำงานเฉพาะสามารถเขียนขึ้น เพื่อให้ทำงานที่ซ้ำๆกันเพื่อลดความยาวของคำสั่ง ในการสร้างฟังชั่นจะเริ่มต้นด้วยการกำหนดชนิดค่าที่ฟังชั่นจะคืนกลับให้โปรแกรมที่เรียกใช้ เช่น int สำหรับค่า integer ถ้าฟังชั่นไม่มีการคืนค่า จะใช้คำว่า void ต่อจากชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่ฟังชั่นคืนกลับจะเป็นชื่อฟังชั่นและตามด้วยวงเล็บซึ่งจะ เป็นค่าพารามิเตอร์ที่จะส่งค่าให้กับฟังชั่น

```
type functionName(parameters)
{
statements;
}
```

ตัวอย่างด้านล่างนี้เป็นค่า integer type function delayVal() ที่ใช้ตั้งค่าหน่วงเวลาในโปรแกรมโดยอ่านค่าของ potentiometer การทำงานของฟังชั่นจะกำหนดตัวแปร v และนำค่าของ potentiometer ซึ่งมีค่า 0-1023 มาเก็บไว้ใน v จากนั้นหารค่านั้นด้วย 4 เพื่อให้ได้ค่า 0-255 สุดท้ายจึงส่งค่ากลับไปยัง main program

1. Digital I/O

- pinMode(pin, mode)

pinMode(pin, mode) ใช้ในกลุ่ม void setup() เพื่อกำหนดหน้าที่ขาของไมโครคอนโทรลเลอร์ให้เป็น ขารับสัญญาน INPUT หรือขาส่งสัญญาณ OUTPUT

ขาของ Arduino digital ถูกตั้งให้เป็นขารับอินพุทโดยปริยาย ดังนั้นจึงไม่จำเป็นต้องสั่งให้มันทำหน้าที่เป็นอินพุทด้วยคำสั่ง pinMode() ขาที่ถูกตั้งให้เป็นอินพุทจะมีคุณสมบัติเป็นขาที่มีอิมพีแดนซ์สูง นอกจากนี้ยังมีความต้านทานพูลอัพ 20K ที่พร้อมใช้งานในตัว Atmega chip ซึ่งสามารถสั่งให้ทำงานได้ด้วยซอฟท์แวร์ โดยคำสั่งดังนี้

ความต้านทานพูลอัพปกติจะใช้ต่อกับอุปกรณ์ อินพุทเช่นสวิทซ์ ตัวอย่างข้างบนไม่ได้ทำให้ขาเปลี่ยนเป็นขา output แต่เป็นวิธีตั้งให้ความต้านทางพูลอัพทำงานเท่านั้น ขาที่กำหนดให้เป็น OUTPUT เรียกได้ว่าเป็นขาที่ทำให้มีอิมพีแดนซ์ต่ำและสามารถจ่ายกระแสได้ 40 มิลลิแอมป์ ให้อุปกรณ์อื่น ซึ่งเพียงพอที่จะขับ LED ให้ติด(อย่าลืมใส่ความต้านทานจำกัดกระแสด้วย) อย่างไรก็ตามมันไม่สามารถขับรีเลย์ โซลีนอยด์ หรือมอเตอร์ได้โดยตรง

การลัดวงจรบนขาของ Arduino และการจ่ายกระแสมากเกินไปอาจทำความเสียหายให้กับขา output หรืออาจทำให้ตัวไอซีเสียหาย ดังนั้นควรต่อความต้านทานค่า 470 โอห์มหรือ 1k โอห์มอนุกรมกับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- digitalWrite(pin, value)

digitalWrite(pin, value) จะทำให้ขาที่กำหนดมีค่าเป็น HIGH หรือ LOW pin: กำหนดขา value : HIGH or LOW

- digitalRead(pin)

digitalRead(pin) จะอ่านค่าจาก pin ที่กำหนดซึ่งค่าที่อ่านได้จะมีค่าเป็น HIGH หรือ LOW pin: ตัวเลขที่กำหนดขาที่ต้องการอ่านค่า

2. Analog I/O

- int analogRead(pin)

คำสั่งนี้อ่านค่าจากขา Analog จะได้ค่า 10 bit คำสั่งนี้จะทำงานกับขา analog input (0-5) เท่านั้น และได้ผลลัพธ์เป็นเลขจำนวนเต็มค่า 0 – 1023

- AnalogWrite(pin, value)

เป็นคำสั่งเขียนค่า analog เทียมโดยใช้ hardware enabled pulse width mdulation(PWM) ไปยังขา output ที่สามารถทำ PWM ได้ ใน Arduino รุ่นใหม่ที่ใช้ชิพ Atmega168 คำสั่งนี้จะทำงานกับขา 3, 5, 6, 9, 10, และ 11 ส่วน Arduino รุ่นเก่าที่ใช้ Atmega8 จะรองรับเพียงขา 9, 10 และ 11 ค่าที่เขียนสามารถใช้เป็นตัวแปรหรือค่าคงที่จาก 0 – 255

ค่า 0 ที่เขียนไปยังขา analog จะทำให้เกิด 0 โวลต์คงที่ที่ขานั้น ส่วนค่า 255 จะทำให้เกิด 5 โวลต์คงที่ที่ขานั้นเช่นกัน สำหรับค่าระหว่าง 0 - 255 ขาจะเปลี่ยนไปเป็นค่าที่อยู่ระหว่าง 0 และ 5 โวลต์ ค่ายิ่งมากแรงดันที่ปรากฏก็ยิ่งเข้าใกล้ HIGH(5 โวลต์) ตัวอย่างเช่น ค่า 64 จะทำให้เกิด 0 โวลต์เป็นสามในสี่ของเวลาทั้งหมด และเป็น 5 โวลต์หนึ่งในสี่ที่เหลือ ถ้าค่าที่เขียนคือ 128 จะทำให้เกิด 0 ครึ่งหนึ่งของคาบเวลาและ 255 อีกครึ่งคาบเวลา ถ้าเป็นค่า 192 จะทำให้เกิด 0 โวลต์หนึ่งในสี่และ 5 โวลต์ สามในสี่ของคาบเวลา

เนื่องจากคำสั่งนี้เป็นคำสั่งให้ hardware ในตัวไอซีทำงาน ขาไอซีจะผลิตคลื่นที่มีคาบเวลาตามค่าที่เขียนอย่างคงที่หลังจากทำคำสั่ง analogWrite ไปเรื่อยๆ จนกระทั่งมีการเรียกใช้คำสั่ง analogWrite ครั้งใหม่ (หรือการเรียกใช้คำสั่ง digitalRead หรือ digitalWrite บนขาไอซีขาเดียวกัน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

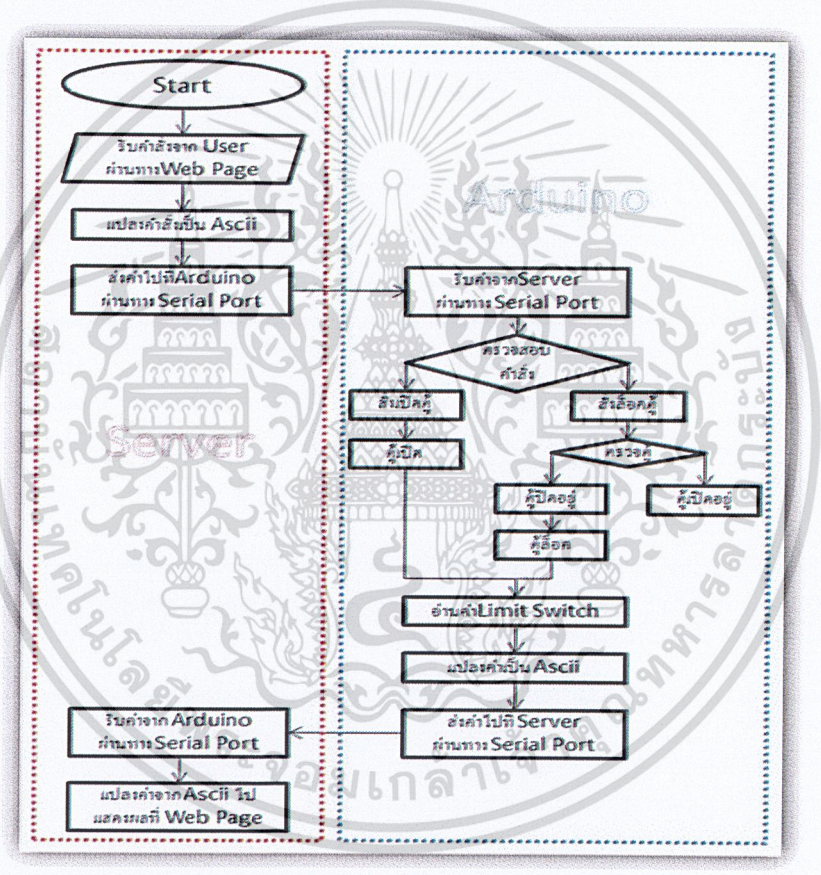
3. Time

- delay(ms)

คำสั่งนี้จะหยุดการทำงานของโปรแกรมเป็นเวลาตามที่กำหนดเป็นมิลลิวินาที ซึ่ง 1000 วินาทีเท่ากับ 1 วินาที

- millis()

คำสั่งนี้จะได้ผลลัพธ์ค่าเวลาเป็นมิลลิวินาทีแสดงค่าที่ Arduino board เริ่มต้นทำโปรแกรมปัจจุบัน ค่าที่ได้เป็นค่า unsigned long ขนาด 32 bit



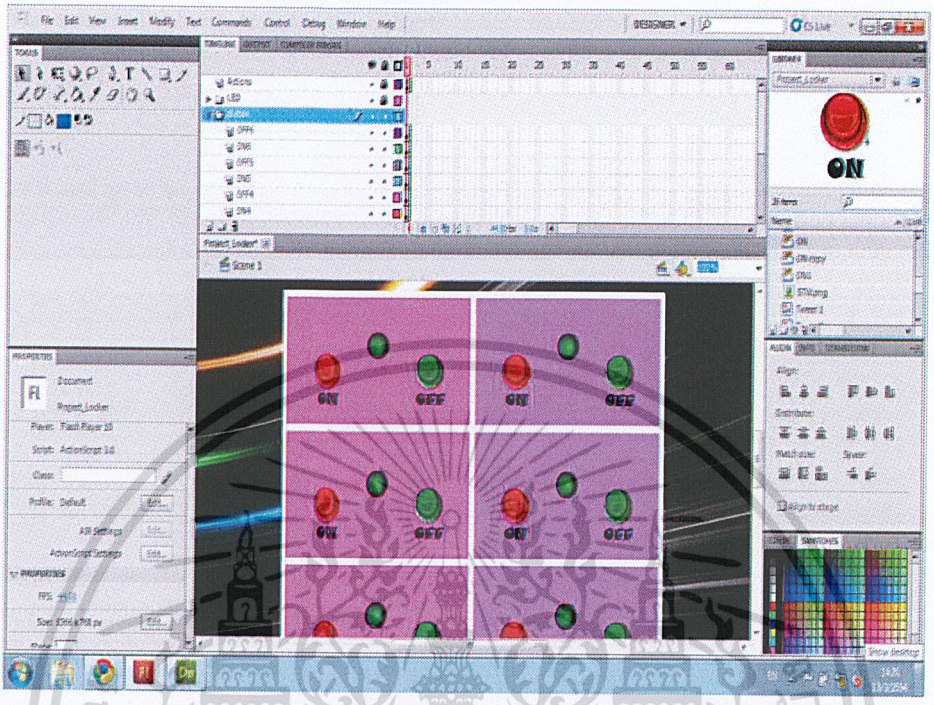
ภาพที่ 2.7 Flow Chat ของโปรแกรม

2.2.2.2 โปรแกรม Adobe Flash Professional CS5

เป็นโปรแกรมที่ใช้สร้างผลงานภาพกราฟฟิก ภาพเคลื่อนไหว Animation สร้างสื่อภาพนิ่งเพื่อนำเสนองาน Presentation ออกแบบงานเว็บไซด์ Web Interactive เกมและโปรแกรมสำหรับออกแบบงานมัลติมีเดีย ซึ่งในโปรเจคนี้ได้ใช้โปรแกรม Adobe Flash Professional CS5 (*Evaluation Version) ในการช่วยออกแบบลูกเล่นหน้า Web Page เพราะโปรแกรม Adobe Flash

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Professional CS5 (Evaluation Version) สามารถสร้างภาพเคลื่อนไหว Animation ได้ ทำให้หน้า Web Page ที่ใช้ในการควบคุมการเปิด- ปิดตู้เก็บเอกสารมีความน่าสนใจมากขึ้น

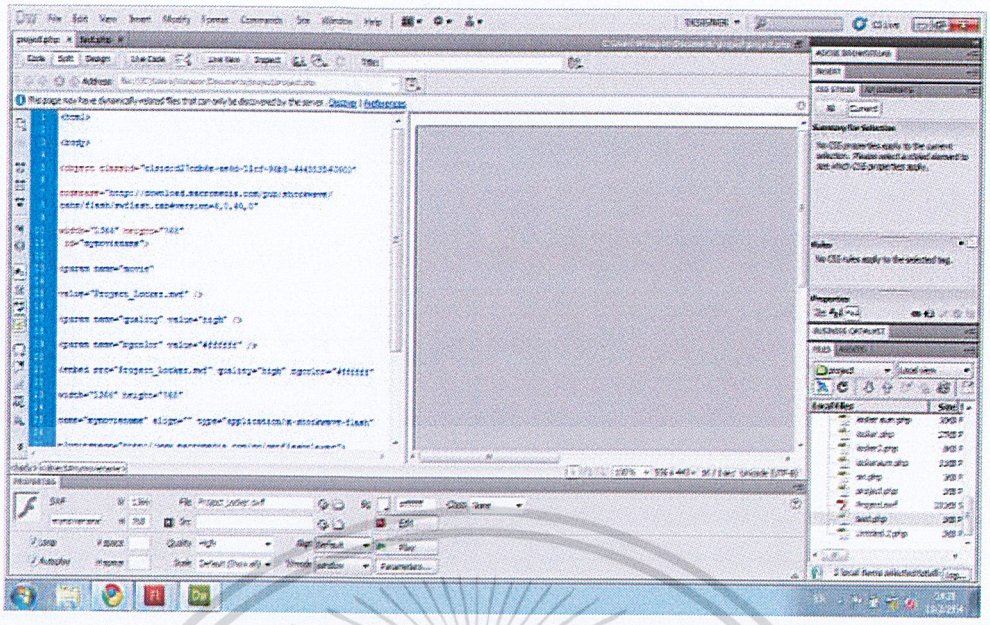


ภาพที่ 2.8 โปรแกรม Adobe Flash Professional

2.2.2.3 โปรแกรม Adobe Dreamweaver CS5

เป็นโปรแกรมที่ได้รับความนิยมในการออกแบบเว็บไซต์เป็นอย่างมาก ซึ่งในโปรเจคนี้ได้ใช้โปรแกรม Adobe Dreamweaver CS5(*Evaluation Version)ในส่วนของการออกแบบหน้า web pageที่ใช้ควบคุมการเปิด-ปิดตู้เก็บเอกสาร

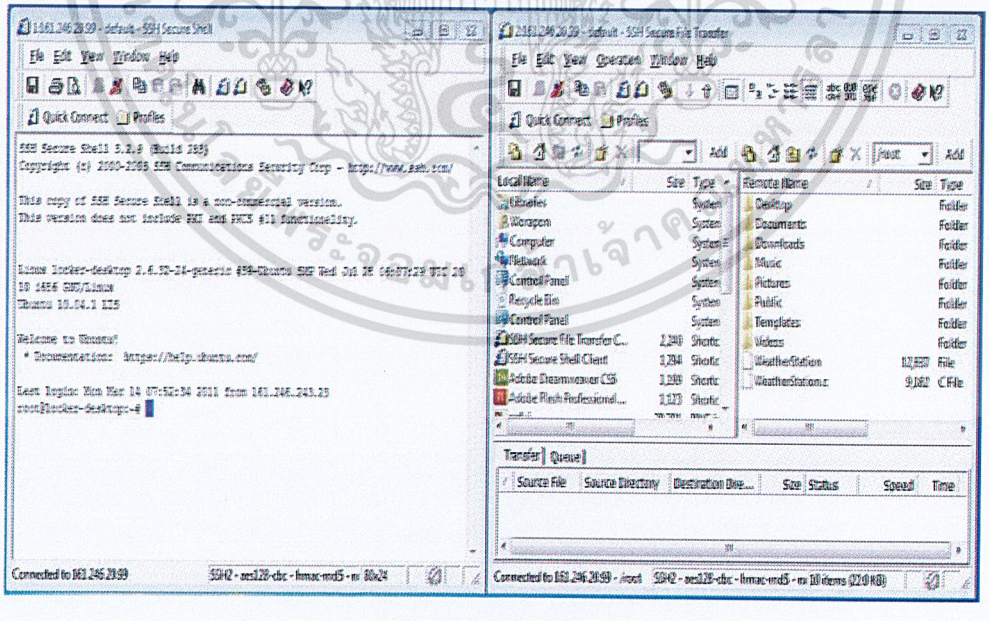
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.9 โปรแกรม Adobe Dreamweaver CS5

2.2.2.4 โปรแกรม SSH Secure Shell Client

secure Shell หรือ SSH คือโพรโทคอลสำหรับเครือข่ายคอมพิวเตอร์ ที่ออกแบบมาเพื่อให้ล็อกเข้าไปยังคอมพิวเตอร์เครื่องอื่น และทำงานต่างๆบนเครื่องนั้น



ภาพที่ 2.10 โปรแกรม SSH Secure Shell Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ภาษาที่ใช้ในการพัฒนา

2.2.3.1 Action script Language

ActionScript เป็นภาษาด้าน Programming ที่ใช้ในโปรแกรม Macromedia Flash โดยเป็นการเขียนสคริปต์ เพื่อควบคุมการทำงานด้านต่างๆ ของชิ้นงานที่สร้างขึ้น ActionScript เป็นเหมือนตัวเชื่อมระหว่างสิ่งที่ผู้ใช้เข้าใจกับสิ่งที่ Flash เข้าใจ การทำงานจะขึ้นอยู่กับเหตุการณ์ (Event) ที่เกิดขึ้น หากเหตุการณ์นั้นไม่เกิดขึ้นก็ไม่มีการทำงาน จึงทำให้งานที่สร้างจากโปรแกรม Flashและมีการเขียน ActionScript ควบคุมจะไม่ใช่ Animation ธรรมดา แต่จะเป็นงานที่สามารถโต้ตอบกับผู้ใช้กับได้อย่างเต็มรูปแบบ เช่น

```
on(release)
{
        picture1._visible = false;
        picture2._visible = true;
}
```

2.2.3.2 HTML Language

Html หรือ HTML ย่อมาจากคำว่า HyperText Markup Language เป็นภาษาที่ใช้ในการแสดงผลของเอกสารบน website หรือที่เราเรียกกันว่าเว็บเพจ เป็นภาษาที่พัฒนาโดย World Wide Web Consortium(W3C) HTML เป็นภาษาที่สำคัญมากกับเทคโนโลยีบนเว็บไซต์ ไม่ว่าคุณจะเขียนโปรแกรมบนเว็บไซต์ด้วยภาษาใด ๆเช่น PHP, ASP, Perl หรืออื่นๆ คุณก็ต้องมีความจำเป็นในการแสดงผลข้อมูลออกมายัง Web Browser ด้วยภาษา HTML เป็นหลักใหญ่หรือมองว่า HTML คือ Output ในการแสดงผลสู่จอภาพของ Web Browser

```
<HTML>
<HEAD>
<TITLE> ชื่อโปรแกรมหรือข้อมูลที่ต้องการแสดงในส่วนหัว </TITLE>
</HEAD>
<BODY>
คำสั่งหรือข้อความที่ต้องการให้แสดง
</BODY>
</HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.3 PHP Language

ภาษาพีเอชพี (PHP Language) คือ ภาษาคอมพิวเตอร์ประเภทโอเพนท์ซอร์ท (Open Source Computer Language) สำหรับพัฒนาเว็บเพจแบบไดนามิก เมื่อเครื่องบริการได้รับคำร้องจากผู้ใช้ก็จะส่งให้กับตัวแปลภาษา ทำหน้าที่ประมวลผลและส่งข้อมูลกลับไปยังเครื่องของผู้ใช้ที่ร้องขอ ในรูปเอชทีเอ็มแอล ภาพ หรือแฟ้มดิจิตอลอื่นใด ลักษณะของภาษามีรากฐานคำสั่งมาจากภาษาซี เป็นภาษาที่สามารถพัฒนาให้ใช้งานแบบโต้ตอบกับผู้ใช้ได้ภาษาพีเอชพี (PHP Language) มีการทำงานแบบเซอร์ฟเวอร์ไซต์สคริปต์ (Server-Side Script) จึงต้องมีเครื่องบริการ (Server) ที่ทำหน้าที่บริการการแปลภาษา และส่งผลให้กับเครื่องผู้ใช้ (Client) ที่ร้องขอด้วยการส่งคำร้องเข้ามายังเครื่องบริการ คำว่า PHP ย่อมาจาก Personal Home Page แต่พัฒนาให้มีประสิทธิภาพยิ่งขึ้น จึงเปลี่ยนเป็น Professional Home Page

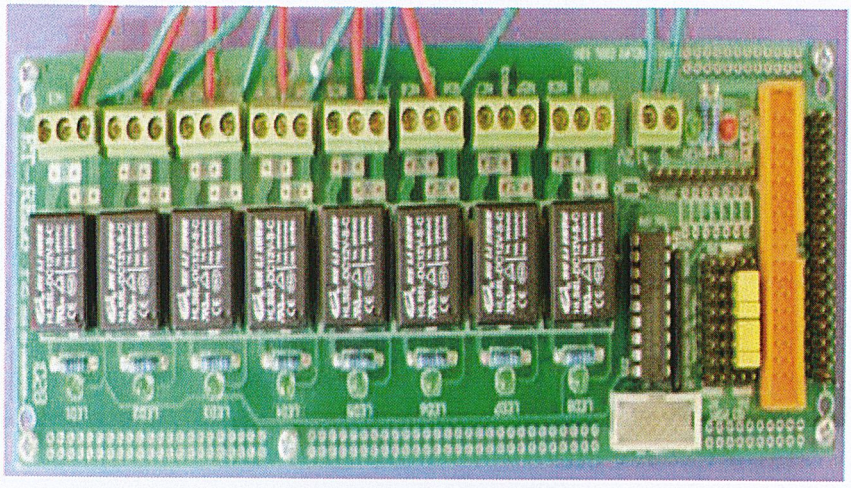
```
<?php
        echo "Hello, World!";
?>
```

2.2.4 ชุดบอร์ดรีเลย์

ET - REL8 เป็นบอร์ดรีเลย์ OUTPUT ขนาด 8 ช่อง มีหน้าสัมผัสตัวละ 2 ชุด (NO และ NC) โดยตัวบอร์ดถูกออกแบบให้เชื่อมต่อกับ I/O PORT 34 PIN (ET-BUS)มาตรฐาน ETT สามารถตั้งการต่อเชื่อมอิสระได้ทั้ง PORT A, PORT B หรือ PORT C

- OUTPUT RELAY 8 CH (NO,NC)
- CONTACT CAPACITY 10A 24VDC
- COIL VOLTAGE 12 VDC,30 mA
- สามารถ SET เลือก POWER 12 VDC ของ COIL RELAY ได้จากขั้วภายนอกหรือจากขั้ว 34 PIN (ในกรณีต่อใช้กับ ET-PC8255)
- ขั้วแบบ 34 PIN ต่อกับบอร์ดต่างๆ ของทาง ETT ได้
- สามารถ SET ตั้ง PORT ได้ A, B หรือ C สามารถต่อ ET-REL8 ได้ 3 บอร์ดต่อขั้ว 34 PIN 1 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.11 ชุดบอร์ดรีเลย์

2.3 ชุดติดต่อกับผู้ใช้งาน (Graphic User Interface: GUI)

2.3.1 ติดต่อผ่านเว็บบราวเซอร์

เว็บบราวเซอร์ (Web Browser) หรือโปรแกรมค้นดูเว็บคือโปรแกรมคอมพิวเตอร์ที่ผู้ใช้สามารถดูข้อมูลและโต้ตอบกับข้อมูลสารสนเทศที่จัดเก็บในหน้าเว็บที่สร้างด้วยภาษาเฉพาะ เช่น ภาษาเอชทีเอ็มแอล (html) ที่จัดเก็บไว้ที่ระบบบริการเว็บหรือเว็บเซอร์ฟเวอร์หรือระบบคลังข้อมูลอื่น ๆ โดยโปรแกรมค้นดูเว็บเปรียบเสมือนเครื่องมือในการติดต่อกับเครือข่ายคอมพิวเตอร์ ขนาดใหญ่ที่เรียกว่าเวิลด์ไวด์เว็บ (www)

โปรแกรมค้นดูเว็บเชื่อมโยงกับเว็บเซิร์ฟเวอร์ผ่านมาตรฐาน หรือ โปรโทคอลการรับส่งข้อมูลแบบเอชทีทีพี (http) ในการส่งหน้าเว็บหรือเว็บเพจปัจจุบันเอชทีทีพี (http) รุ่นล่าสุดคือ 1.1 ซึ่งสนับสนุนโดยโปรแกรมค้นดูเว็บทั่วไป ยกเว้น อินเทอร์เน็ตเอกซ์พลอเรอร์ (Internet Explorer) ที่ยังสนับสนุนไม่เต็มที่

ที่อยู่ของเว็บเพจเรียกว่ายูอาร์แอลหรือยูอาร์ไอ ซึ่งรูปแบบมักจะเริ่มต้นด้วยคำว่า http:// สำหรับการติดต่อแบบเอชทีทีพี โปรแกรมค้นดูเว็บส่วนมากสนับสนุนการเชื่อมต่อรูปแบบอื่นนอกจากนี้ เช่น ftp:// สำหรับเอฟทีพี (FTP) https:// สำหรับเอชทีทีพีแบบสนับสนุนการเข้ารหัสข้อมูลเพื่อความปลอดภัย เป็นต้น

รูปแบบของไฟล์สำหรับเว็บเรียกว่าเอชทีเอ็มแอล (html) และสนับสนุนไฟล์รูปแบบอื่นๆ เช่น รูปภาพ (JPG, GIF, PNG) หรือเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 เซอร์ฟเวอร์และฐานข้อมูล

2.4.1 อุปกรณ์ Server

ภาพที่ 2.12 อุปกรณ์ Server

Server คือเครื่องคอมพิวเตอร์หรือระบบปฏิบัติการหรือโปรแกรมคอมพิวเตอร์ ที่ทำหน้าที่ให้บริการอย่างใดอย่างหนึ่งหรือหลายอย่าง แก่เครื่องคอมพิวเตอร์หรือโปรแกรมคอมพิวเตอร์ที่เป็นลูกข่ายในระบบเครือข่าย Server ในทาง computer มี 3 ความหมายคือ

- เครื่องคอมพิวเตอร์ที่ทำหน้าที่ให้บริการอะไรบางอย่างแก่คอมพิวเตอร์หรือโปรแกรมคอมพิวเตอร์อื่น
- ระบบปฏิบัติการคอมพิวเตอร์ที่ทำหน้าที่ให้บริการอะไรบางอย่างแก่คอมพิวเตอร์ หรือโปรแกรมคอมพิวเตอร์อื่น
- โปรแกรมคอมพิวเตอร์ที่ทำหน้าที่ให้บริการอะไรบางอย่างแก่คอมพิวเตอร์หรือ โปรแกรมคอมพิวเตอร์อื่น

โดยปกติแล้วโปรแกรมคอมพิวเตอร์ที่เป็น Server จะทำงานบนระบบปฏิบัติการ อาจจะเป็น Linux หรือ Windows หรือ Unix ก็ได้ ดังนั้นคำว่า server จึงมิได้หมายถึงคอมพิวเตอร์เพียงอย่างเดียว แต่ยังหมายถึงระบบปฏิบัติการคอมพิวเตอร์ หรือโปรแกรมคอมพิวเตอร์อีกด้วย ยกตัวอย่างเป็นกลุ่มๆ ดังต่อไปนี้

- Web server คือโปรแกรมที่ทำหน้าที่ให้บริการเว็บ อาทิเช่น Apache web server
- Mail server คือโปรแกรมที่ทำหน้าที่ให้บริการ E-mail อาทิเช่น Postfix, qmail, courier
- DNS server คือโปรแกรมที่ทำหน้าที่ให้บริการโดเมนเนม อาทิเช่น bind9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Database server คือโปรแกรมที่ทำหน้าที่ให้บริการ Database อาทิเช่น mysql, postgresql, DB2 สำหรับระบบปฏิบัติการที่นิยมใช้เป็น server ได้แก่

- Linux สำหรับ Linux Distribution ที่ได้รับความนิยมได้แก่ Debian Ubuntu Readhat Fedora etc.

- Windows สำหรับ Windows ที่นิยมใช้เป็น server ได้แก่ Windows Server 2003

- Unix สำหรับ Unix ถือเป็นระบบปฏิบัติการที่เก่าแก่ระบบหนึ่ง ที่ยังใช้งานอยู่จนถึงทุกวันนี้ ได้แก่ BSD

2.4.2 เว็บเซอร์ฟเวอร์ อาปาเช (Apache)

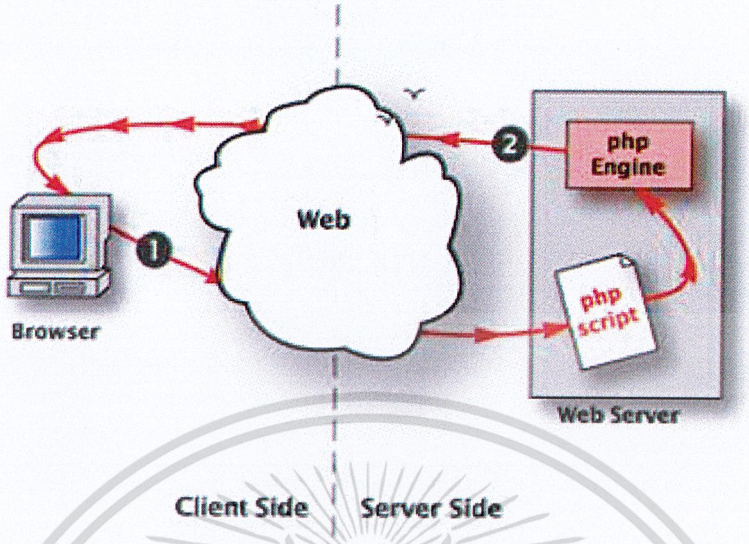
Web Server คือ บริการ HTTP (Hyper Text Transfer Protocol) เพื่อให้ผู้ใช้สามารถอ่านข้อมูล ทั้งภาพ และเสียง จากเครื่องบริการผ่าน Browser เช่น บริการ http://www.thaiall.com หรือ http://localhost เป็นต้นเครื่องบริการที่รอรับคำร้องขอจาก web browser ข้อมูลที่จะส่งไปอาจเป็นเว็บเพจ ภาพ หรือเสียง เป็นต้น สำหรับโปรแกรมที่ได้รับความนิยมให้นำมาเปิดบริการ web คือ Apache web server หรือ Microsoft web server

โปรแกรม Apache เป็นเว็บเซอร์เวอร์ ที่ได้รับความนิยมมากที่สุดบนอินเทอร์เน็ตตั้งแต่เดือนเมษายน 1996 เหตุสำคัญที่ทำให้มันได้รับความนิยมมี 3 ประการคือ ทำงานได้หลาย OS (แพลตฟอร์ม) ฟีเจอร์ที่มีมากและต้นทุนต่ำ Apache ทำงานได้เกือบทุกแพลตฟอร์มในปัจจุบัน และมีฟีเจอร์ที่หลากหลาย เพื่อให้นักพัฒนาเว็บสามารถสร้างและเพิ่มขยายขีดความสามารถในการออกแบบเว็บได้ และสามารถใช้งานได้ฟรี

Apache พัฒนามาจาก httpd Web Server ที่มีกลุ่มผู้พัฒนาอยู่ก่อนแล้ว โดยร็อบ แม็คคูล (Rob McCool) การที่อาปาเชเป็นซอฟต์แวร์ที่อยู่ในลักษณะของโอเพ่นซอร์สที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาส่วนต่างๆของอาปาเชได้ ซึ่งทำให้เกิดเป็นโมดูลที่เกิดประโยชน์มากมาย เช่น mod_perl, mod_python หรือ mod_php ซึ่งเป็นโมดูลที่ทำให้อาปาเชสามารถใช้ประโยชน์ และทำงานร่วมกับภาษาอื่นได้แทนที่จะเป็นเพียงเซอร์ฟเวอร์ที่ให้บริการเพียงแค่เอชทีเอ็มแอลอย่างเดียว นอกจากนี้อาปาเชเองยังมีความสามารถอื่นๆ ด้วย เช่น การยืนยันตัวบุคคล (mod_auth, mod_access, mod_digest) หรือเพิ่มความปลอดภัยในการสื่อสารผ่านโปรโตคอล https (mod_ssl) นอกจากนี้ก็ยังมีโมดูลอื่นๆที่ได้รับความนิยมใช้ เช่น mod_vhost ทำให้สามารถสร้างโฮสต์เสมือน www.sample.com, wiki.sample.com, mail.sample.com หรือ www.ilovewiki.org ภายในเครื่องเดียวกันได้ หรือ mod_rewrite เป็นเครื่องมือที่จะช่วยให้ url ของเว็บนั้นอ่านง่ายขึ้น ยกตัวอย่างเช่น จากเดิมต้องอ้างถึงเว็บไซต์แห่งหนึ่ง ด้วยการพิมพ์ http://www.yourdomain.com/board/quiestion.php? action=viewtopic&qid=2xDffw แต่หลังจากใช้ mod_rewrite จะทำให้สั้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลายเป็น http://www.yourdomain.com/board/question/2xDffw ซึ่งที่อยู่หลังนี้จะขึ้นอยู่กับว่าผู้ดูแลเว็บไซต์ต้องการให้อยู่ในลักษณะใด



ภาพที่ 2.13 Web Server Diagram

2.4.3 ระบบฐานข้อมูล มายเอสคิวแอล (MySQL)

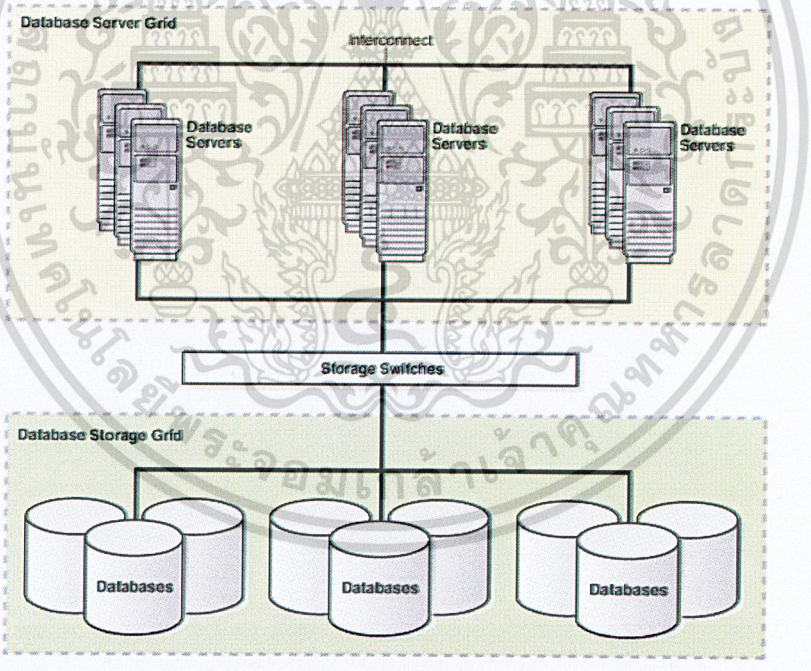
Database Server คือ เครื่องบริการข้อมูลที่เปิดให้ผู้ใช้เพิ่มข้อมูล ลบ หรือแก้ไข สำหรับโปรแกรมบริการระบบฐานข้อมูลที่นิยมใช้ ได้แก่ MYSQL หรือ Microsoft Access เป็นต้น โดยผู้ใช้ต้องเขียนโปรแกรมสั่งประมวลผล ปรับปรุงข้อมูล หรือนำข้อมูลในส่วนที่ตนเองมีสิทธิ์ ไปใช้ตามต้องการ

MySQL เป็นระบบจัดการฐานข้อมูล โดยใช้ภาษา SQL. เป็นซอฟต์แวร์โอเพนซอร์สสามารถทำงานบนระบบปฏิบัติการ Linux โดยใช้ภาษา SQL MySQL แตกต่างจากซอร์ฟแวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดนโดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรีและแบบที่ใช้ในเชิงธุรกิจ MySQL สร้างขึ้น โดยชาวสวีเดน 2 คน และชาวฟินแลนด์ ชื่อ David Axmark, Allan Larsson และ Michael “Monty” Widenius

รุ่นของผลิตภัณฑ์นั้นแบ่งออกมาได้สามสายการผลิต ได้แก่ เวอร์ชั่นที่เป็นของฟรี (คอมมิวนิตี้เวอร์ชั่น) เวอร์ชั่นการค้า (คอมเมอร์เชียลเวอร์ชั่น) และเวอร์ชั่นที่สนับสนุนกับผลิตภัณฑ์ SAP (MAX DB) ความแตกต่าง คือ เวอร์ชั่นคอมมิวนิตี้นั้นสามารถนำไปใช้งานได้ฟรี แต่ขาดการสนับสนุนการช่วยเหลือเมื่อมีปัญหาเกิดขึ้น เวอร์ชั่นที่เป็นคอมเมอร์เชียลนั้นให้บริการด้านการสนับสนุนเมื่อมีปัญหา (ซื้อบริการ) เวอร์ชั่นสุดท้ายสนับสนุนและใช้งานกับโปรแกรม SAP มีส่วนติดต่อ (interface) เป็นตัวเชื่อมต่อกับภาษาในการพัฒนาอื่นๆ และเพื่อให้เข้าถึงฟังก์ชั่นการทำงานกับฐานข้อมูลมายเอสคิวแอลได้ เช่น ODBC (Open Database Connector) อันเป็นมาตรฐานกลางที่กำหนดมาเพื่อให้ใช้เป็นสะพานในการเชื่อมต่อกับโปรแกรมหรือระบบอื่นๆ เช่น MyODBC อัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น Driver เพื่อใช้สำหรับการเชื่อมต่อในระบบปฏิบัติการวินโดว์, JDBC คลาสส่วนเชื่อมต่อสำหรับ Java เพื่อใช้ในการติดต่อกับ MySQL และมี API (Application Programming Interface) ต่างๆมีให้เลือกใช้มากมาย ในการที่เข้าถึง MySQL โดยไม่ขึ้นอยู่กับภาษาการพัฒนาใดภาษาหนึ่ง มายเอสคิวแอลเป็นมาตรฐานสำหรับระบบข้อมูลเชิงสัมพันธ์ (Relational Database Management system ย่อเป็น RDBMS) คำสั่งของเอสคิวแอล สามารถจัดการกับฐานข้อมูลได้หลากหลาย เช่น การปรับปรุงข้อมูล การดึงข้อมูลเพื่อแสดงผลที่ต้องการ ระบบฐานข้อมูลในปัจจุบันเกือบทั้งหมดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDMBS) เช่น ออราเคิล (Oracle) , ไซเบส (Sybase) , ไมโครซอฟ (Microsoft) ,มายเอสคิวแอลเซอร์ฟเวอร์ (MySQL Server) ,ไมโครซอฟแอกเซจ (Microsoft Access) ,ฐานข้อมูลของไอบีเอ็ม (IBM) เป็นต้น ถึงแม้ว่าฐานข้อมูลของแบบระบบจัดการ ฐานข้อมูลเชิงสัมพันธ์ทุกระบบ จะมาสารถใช้คำสั้งพื้นฐานของมายเอสคิวแอลเหมือนกัน แต่ทุกระบบอาจจะมีคำสั่งพิเศษของตัวเองที่ระบบอื่นอาจจะใช้ไม่ได้ก็ได้ ดังนั้นถ้าต้องการทราบว่าระบบฐานข้อมูลที่ใช้อยู่นั้นมีความสามารถพิเศษใดเพิ่มเติมนอกเหนือจากคำสั่งมายเอสคิวแอลแล้ว เราก็สามารถดูได้จากคู่มือของระบบฐานข้อมูลนั้นๆ



ภาพที่ 2.14 Database Server Diagram

2.4.4 พีเอชพี (PHP)

PHP Programming เกิดในปี 1994 โดย Rasmus Lerdorf โปรแกรมเมอร์ชาวสหรัฐอเมริกาได้คิดค้นสร้างเครื่องมือที่ใช้ในการพัฒนาเว็บส่วนตัวของเขา โดยใช้ข้อดีของภาษา C และ Perl เรียกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

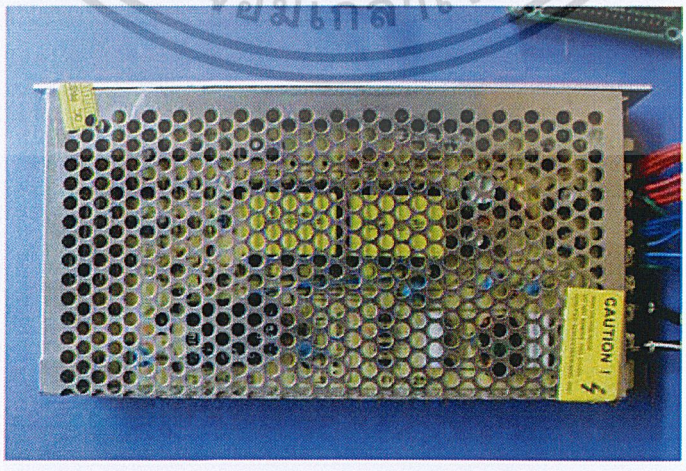
Personal Home Page และได้สร้างส่วนติดต่อกับฐานข้อมูลชื่อว่า Form Interpreter (FI) รวมทั้งสองส่วน เรียกว่า PHP/FI ซึ่งก็เป็นจุดเริ่มต้นของ PHP มีคนที่เข้ามาเยี่ยมชมเว็บไซต์ของเขาแล้วเกิดชอบจึงติดต่อขอเอาโค้ดไปใช้บ้าง และนำไปพัฒนาต่อ ในลักษณะของ Open Source ภายหลังมีความนิยมขึ้นเป็นอย่างมาก ภายใน 3 ปีมีเว็บไซต์ที่ใช้ PHP/FI ในติดต่อฐานข้อมูลและแสดงผลแบบ ไดนามิกและอื่นๆ มากกว่า 50000 ไซต์

PHP เป็นภาษาสคริปต์ที่ประมวลผลที่ฝั่งเซิร์ฟเวอร์ แล้วส่งผลลัพธ์ไปแสดงผลที่ฝั่งไคลเอ็นต์ผ่านบราวเซอร์เช่นเดียวกับ CGI และ ASP การแสดงผลของพีเอชพี (PHP) จะปรากฏในลักษณะของ HTML ซึ่งจะไม่แสดงคำสั่งที่ผู้ใช้เขียน ซึ่งเป็นลักษณะเด่นที่พีเอชพีแตกต่างจากภาษาในลักษณะไคลเอนต์-ไซด์ สคริปต์ เช่น ภาษาจาวาสคริปต์ ที่ผู้ชมเว็บไซต์สามารถอ่านและคัดลอกคำสั่งไปใช้ได้เอง นอกจากนี้พีเอชพี (PHP) ยังเป็นภาษาที่เรียนรู้และเริ่มต้นได้ไม่ยาก โดยมีเครื่องมือช่วยเหลือและคู่มือที่สามารถหาอ่านได้ฟรีบนอินเตอร์เน็ต ความสามารถการประมวลผลหลักของพีเอชพี (PHP) ได้แก่ การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง การอ่านข้อมูลจากผู้ใช้และประมวลผลการอ่านข้อมูลจากดาต้าเบส (Data Base)

2.5 ระบบจ่ายไฟ

2.5.1 ระบบ Switching Power Supply

สวิตชิ่งเพาเวอร์ซัพพลาย (Switching Power Supply) เป็นแหล่งจ่ายไฟตรงคงค่าแรงดันแบบหนึ่ง และสามารถเปลี่ยนแรงดันไฟจากไปสลับโวลต์สูง ให้เป็นแรงดันไฟตรงค่าต่ำ เพื่อใช้ในงานอิเลคทรอนิกส์ได้เช่นเดียวกันแหล่งจ่ายไฟเชิงเส้น (Linear Power Supply) ถึงแม้เพาเวอร์ซัพพลายทั้งสองแบบ จะต้องมีการใช้หม้อแปลงในการลดทอนแรงดันสูงให้เป็นแรงดันต่ำเช่นเดียวกัน แต่สวิตชิ่งเพาเวอร์ซัพพลายจะต้องการใช้หม้อแปลงที่มีขนาดเล็กและน้ำหนักน้อย เมื่อเทียบกับแหล่งจ่ายไฟเชิงเส้น อีกทั้งสวิตชิ่งเพาเวอร์ซัพพลายยังมีประสิทธิภาพสูงกว่าอีกด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.15 Switching Power Supply

ในปัจจุบันสวิตชิ่งเพาเวอร์ซัพพลาย ได้เข้ามามีบทบาทกับชีวิตเราอย่างมาก เครื่องใช้อิเล็กทรอนิกส์ขนาดเล็ก ซึ่งต้องการแหล่งจ่ายไฟที่มีกำลังสูงแต่มีขนาดเล็ก เช่น เครื่องคอมพิวเตอร์ เครื่องโทรสาร และ โทรทัศน์ จำเป็นจะต้องใช้สวิตชิ่งเพาเวอร์ซัพพลาย แนวโน้มการนำสวิตชิ่งเพาเวอร์ซัพพลายมาใช้ในเครื่องใช้อิเลคทรอนิกส์ทุกประเภทจึงเป็นไปได้สูง การศึกษาหลักการทำงานและการออกแบบสวิตชิ่งเพาเวอร์ซัพพลายจึงเป็นสิ่งจำเป็นที่ไม่อาจหลีกเลี่ยงได้ สำหรับผู้ที่เกี่ยวข้องกับงานอิเล็กทรอนิกส์ทุกประเภท

2.5.1.1 Switching Power Supply กับแหล่งจ่ายไฟเชิงเส้น

ข้อได้เปรียบของสวิตชิ่งเพาเวอร์ซัพพลาย เมื่อเปรียบเทียบกับแหล่งจ่ายไฟเชิงเส้น คือ ประสิทธิภาพที่สูง ขนาดเล็ก และน้ำหนักเบากว่าแหล่งจ่ายไฟเชิงเส้น เนื่องจากแหล่งจ่ายไฟเชิงเส้นใช้หม้อแปลงความถี่ต่ำจึงมีขนาดใหญ่ และน้ำหนักมาก ขณะใช้งานจะมีแรงดันและกระแสผ่านตัวหม้อแปลงตลอดเวลา กำลังงานสูญเสียที่เกิดจากหม้อแปลงจึงมีค่าสูง การคงค่าแรงดันแหล่งจ่ายไฟเชิงเส้นส่วนมากจะใช้เพาเวอร์ทรานซิสเตอร์ต่ออนุกรมที่เอาต์พุต เพื่อจ่ายกระแสและคงค่าแรงดัน กำลังงานสูญเสียในรูปความร้อนจะมีค่าสูงและต้องใช้แผ่นระบายความร้อนขนาดใหญ่ ซึ่งกินเนื้อที่เมื่อเพาเวอร์ซัพพลายต้องการกำลังงานสูงๆ จะทำให้มีขนาดใหญ่และมีน้ำหนักมาก ปกติแหล่งจ่ายไฟเชิงเส้นจะมีประสิทธิภาพประมาณ 30% หรืออาจทำได้สูงถึง 50% ในบางกรณี ซึ่งนับได้ว่าค่อนข้างต่ำเมื่อเปรียบเทียบกับสวิตชิ่งเพาเวอร์ซัพพลายซึ่งมีประสิทธิภาพในช่วง 65%-80%

สวิตชิ่งเพาเวอร์ซัพพลายมีช่วงเวลาโคลสต์อัพประมาณ $20x\ 10^{-3}$ ถึง $50x10^{-3}$ วินาที ในขณะที่แหล่งจ่ายไฟเชิงเส้นจะทำได้เพียงประมาณ $2x10^{-3}$ วินาที ซึ่งมีผลต่อการจัดหาแหล่งจ่ายไฟสำรองเพื่อป้องกันการหยุดทำงานของอุปกรณ์ที่ใช้กับเพาเวอร์ซัพพลาย เมื่อเกิดการหยุดจ่ายแรงดันไฟสลับ รวมทั้งสวิตชิ่งเพาเวอร์ซัพพลายสามารถทำงานได้ในช่วงแรงดันอินพุตค่อนข้างกว้างจึงยังคงสามารถทำงานได้เมื่อเกิดกรณีแรงดันไฟตกอีกด้วย

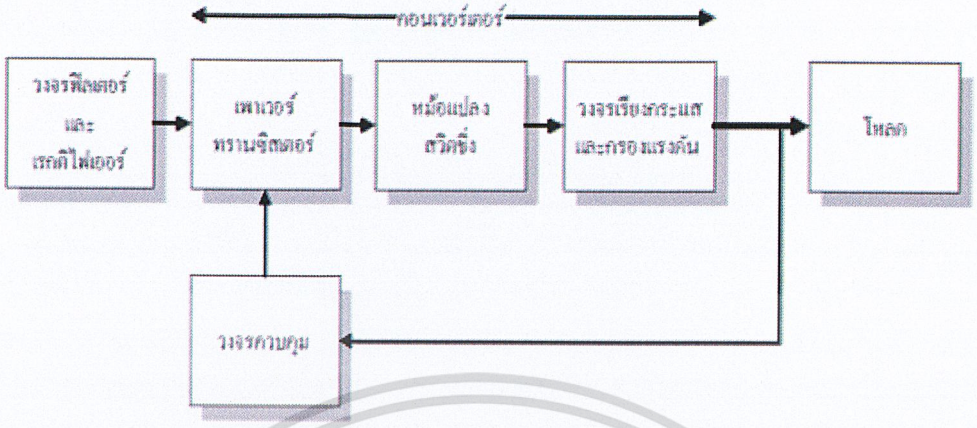
อย่างไรก็ตาม สวิตชิ่งเพาเวอร์ซัพพลายจะมีเสถียรภาพในการทำงานที่ต่ำกว่า และก่อให้เกิดสัญญาณรบกวนได้สูงเมื่อเปรียบเทียบกับแหล่งจ่ายไฟเชิงเส้น รวมทั้งสวิตชิ่งเพาเวอร์ซัพพลายยังมีความซับซ้อนของวงจรมากกว่าและมีราคาสูงที่กำลังงานต่ำๆ แหล่งจ่ายไฟเชิงเส้นจะประหยัดกว่าและให้ผลดีเท่าเทียมกัน ดังนั้นสวิตชิ่งเพาเวอร์ซัพพลายจึงมักนิยมใช้กันในงานที่ต้องการกำลังงานตั้งแต่ 20 วัตต์ขึ้นไปเท่านั้น

2.5.1.2 หลักการทำงานของสวิตชิ่งเพาเวอร์ซัพพลาย

สวิตชิ่งเพาเวอร์ซัพพลายโดยทั่วไปมีองค์ประกอบพื้นฐานที่คล้ายคลึงกัน และไม่ซับซ้อนมากนัก ดังแสดงในรูปที่ 2.16 ซึ่งหัวใจสำคัญของสวิตชิ่งเพาเวอร์ซัพพลายจะอยู่ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเวอร์เตอร์ เนื่องจากทำหน้าที่ทั้งลดทอนแรงดันและคงค่าแรงดันเอาต์พุตด้วย องค์ประกอบต่างๆ ทำงานตามลำดับดังนี้



ภาพที่ 2.16 องค์ประกอบพื้นฐานของสวิตชิ่งเพาเวอร์ซัพพลาย

แรงดันไฟสลับค่าสูงจะผ่านเข้ามาทางวงจร RFI ฟิลเตอร์ เพื่อกรองสัญญาณรบกวนและแปลงเป็นไฟตรงค่าสูงด้วยวงจรเรกติไฟเออร์ เพาเวอร์ทรานซิสเตอร์จะทำงานเป็นเพาเวอร์คอนเวอร์เตอร์โดยการตัดต่อแรงดันเป็นช่วงๆ ที่ความถี่ประมาณ 20-200 KHz จากนั้นจะผ่านไปยังหม้อแปลงสวิตชิ่งเพื่อลดแรงดันลง เอาต์พุตของหม้อแปลงจะต่อกับวงจรเรียงกระแส และกรองแรงดันให้เรียบ การคงค่าแรงดันจะทำได้โดยการป้อนกลับค่าแรงดันที่เอาต์พุตกลับมายังวงจรควบคุม เพื่อควบคุมให้เพาเวอร์ทรานซิสเตอร์นำกระแสมากขึ้น หรือน้อยลงตามการเปลี่ยนแปลงของแรงดันที่เอาต์พุต ซึ่งจะมีผลทำให้แรงดันเอาต์พุตคงที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โปรโตคอลและการสื่อสารข้อมูล

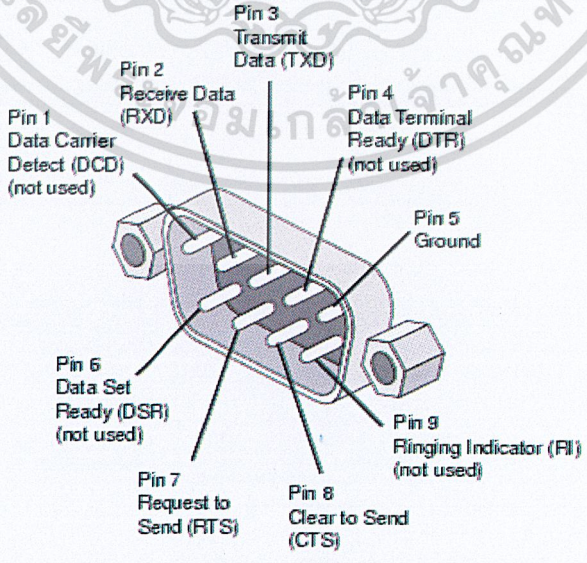
3.1 กล่าวนำ

ปริญญานิพนธ์เล่มนี้ได้นำเสนอ การออกแบบและสร้างระบบควบคุมการเปิด–ปิดตู้เอกสารผ่านทางอินเตอร์เน็ต ซึ่งมีความจำเป็นจะต้องทำการศึกษาหลักการและทฤษฎีต่างๆ เกี่ยวกับโปรโตคอลและการสื่อสารข้อมูล อย่างเช่น การสื่อสารข้อมูล Ethernet และโปรโตคอล (TCP/IP) Time To Live (TTL) UDP Protocal เป็นต้น

3.2 การสื่อสารข้อมูล

ในปัจจุบันการสื่อสารข้อมูลได้ถูกพัฒนาไปอย่างก้าวหน้าทั้งในชีวิตประจำวัน และในทางอุตสาหกรรม อุปกรณ์รับส่งข้อมูลระยะไกล ในการสื่อสารระหว่างอุปกรณ์รับส่งข้อมูลระยะไกลและหน่วยควบคุมส่วนกลางจะต้องมีโปรโตคอลและรูปแบบการสื่อสาร

ในการติดต่อสื่อสารในปัจจุบัน ถ้าเราต้องการจะติดต่อสื่อสารโดยที่ไม่ใช้แผ่นดิสก์ หรืออุปกรณ์พกพาประเภทต่างๆ เราก็จะนึกถึงการเชื่อมต่อแบบอนุกรมโดยใช้สาย Serial Link โดยต่อเข้าที่ช่องเชื่อมต่ออนุกรมซึ่ง ถ้าใช้ Serial Port ก็ติดต่อสื่อสารได้แค่สองเครื่องเท่านั้น แต่ในการสื่อสารแบบอนุกรม (serial) นี้ ก็มีข้อเสียหลายประการ เช่น ถ้าเราจำเป็นต้องติดต่อหลายๆ เครื่องจะไม่สามารถติดต่อกันได้โดยใช้ Serial Port



ภาพที่ 3.1 RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 Ethernet และโปรโตคอล TCP/IP (Transmission Control Protocol/Internet Protocol)

เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหาโปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้ ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบันTCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน 3 ประการคือ

- เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
- ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่น ในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่น เพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
- มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

TCP/IP นั้น จะแบ่งหน้าที่ความรับผิดชอบออกเป็นชั้นซ้อนกัน ที่เรียกว่า โปรโตคอลสแตก (Protocol Stack) ถึงแม้วัตถุประสงค์หลักของโปรโตคอล TCP/IP ที่ออกแบบมา เพื่อใช้งานบนเครือข่ายระยะไกลเป็นสำคัญ แต่โปรโตคอลดังกล่าวก็ยังสามารถใช้งานได้ดีบนเครือข่ายภายในอย่างเครือข่ายท้องถิ่น ด้วยการเชื่อมต่อเครือข่ายอินเทอร์เน็ตได้ และด้วยเหตุนี้เองในปัจจุบันโปรโตคอล TCP/IP จึงเป็นโปรโตคอลที่ได้รับความนิยมเป็นอย่างสูง

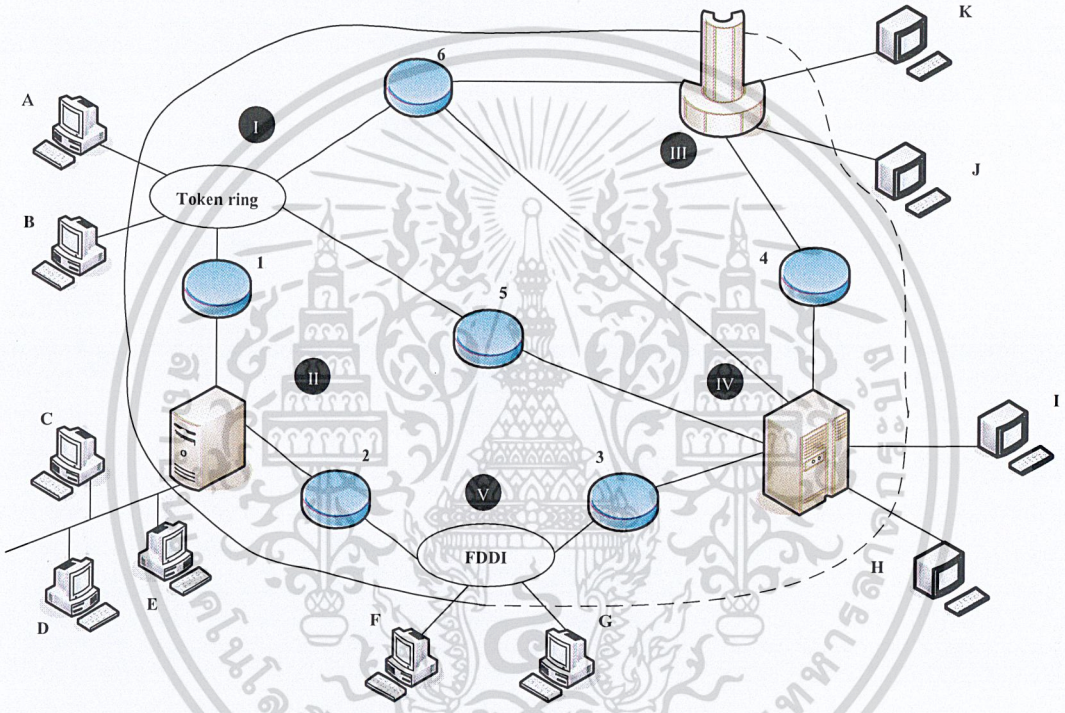
3.4 TCP/IP และอินเทอร์เน็ต

TCP/IP กับแนวคิดของเครือข่ายอินเทอร์เน็ตนี้ได้พัฒนามาร่วมกัน โดยก่อนที่จะกล่าวรายละเอียดถึงโปรโตคอล TCP/IP จำเป็นต้องเข้าใจถึงความสัมพันธ์ระหว่าง TCP/IP กับเครือข่ายอินเตอร์เน็ตเสียก่อน

พื้นฐานทางเทคโนโลยีเครือข่ายต่างๆ ไม่ว่าจะเป็น เครือข่ายท้องถิ่น เครือข่ายระดับเมือง หรือเครือข่ายระดับประเทศ ซึ่งจะทำให้ทราบถึงเทคโนโลยีพื้นฐาน รวมถึงอุปกรณ์ฮาร์ดแวร์สำคัญๆ ที่ใช้งานเหมาะสมและข้อจำกัด ตัวอย่างเช่น เทคโนโลยีเครือข่ายท้องถิ่นออกแบบมาด้วยการสื่อสารความเร็วสูงบนระยะทางสั้นๆ ในขณะที่เครือข่ายระดับประเทศออกแบบมาเพื่อการ

เอกสารนี้ถูกสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารให้สามารถครอบคลุม เนื้อที่ที่กว้างกว่าและมีระยะทางไกล แต่ละองค์กรจะใช้เทคโนโลยีเครือข่าย ที่ออกแบบมาให้มีความเหมาะสมที่สุดกับความต้องการและประเภทของงาน และหลายๆ องค์กรในปัจจุบันได้มีการใช้บริการเครือข่ายหลายชนิดด้วยกัน ตัวอย่างเช่น เทคโนโลยีเครือข่ายท้องถิ่น หรืออีเทอร์เนตจะเป็นแนวทางที่ดีที่สุดสำหรับการเชื่อมต่อเครือข่ายขนาดเล็ก เพื่อใช้งานภายในสำนักงาน นอกจากนี้ยังสามารถนำวงแลนหลายๆ วงเชื่อมโยงเข้ากับเครือข่ายระดับประเทศหรือแวน แต่อาจจำเป็นต้องใช้เทคโนโลยีเฟรมรีเลย์ (Frame Relay) ซึ่งเทคโนโลยีดังกล่าว จะช่วยให้คอมพิวเตอร์ที่อยู่ต่างพื้นที่สามารถสื่อสารร่วมกันได้

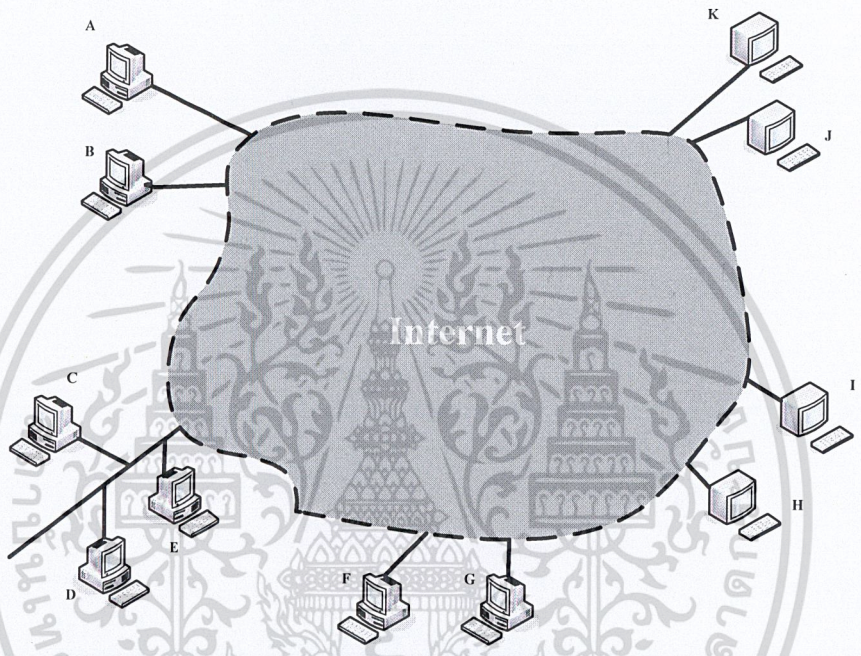


ภาพที่ 3.2 เครือข่ายอินเทอร์เน็ตในมุมมองทางกายภาพ (Physical Networks) ประกอบไปด้วยเครือข่ายมากมาย

เครือข่ายอินเตอร์เน็ตเป็นเครือข่ายที่ประกอบไปด้วยหลายๆเครือข่าย ที่เชื่อมโยงเข้าด้วยกันทั่วโลก ซึ่งมีทั้งเครือข่ายที่เป็นของหน่วยงานของรัฐ รวมทั้งเครือข่ายขององค์กรที่มีทั้งแบบแสวงหากำไรและไม่แสวงหากำไร เครือข่ายอินเตอร์เน็ตเป็นเครือข่ายสาธารณะที่ผู้คนทั่วไปสามารถ เชื่อมต่อเข้าใช้งานได้ แต่หากเครือข่ายที่มีเจ้าของอย่างเครือข่ายท้องถิ่น อาจจำต้องใช้เทคโนโลยีป้องกัน มิให้ผู้อื่นเข้ามาใช้งานหรือขึ้นฐานข้อมูลแบบเครือข่ายส่วนบุคคลของตนได้โดยทุกๆ เครือข่ายที่อยู่บนเครือข่ายอินเทอร์เน็ตจะต้องใช้มาตรฐานโปรโตคอล (TCP/IP) อินเตอร์เน็ตที่ทำงานภายใต้ TCP/IP นั้น หากพิจารณาแบบง่ายๆ ก็มีความคล้ายคลึงกับเครือข่ายหนึ่งที่มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เชื่อมต่อกับคอมพิวเตอร์ชนิดต่างๆ แต่แท้จริงแล้วอินเตอร์เน็ตเป็นเครือข่ายขนาดมหึมา ที่มีจำนวนมากกว่าสิบล้านโฮสต์ และมีมากกว่าพันเครือข่ายที่เชื่อมต่อเข้าด้วยกัน และด้วยเครือข่ายที่มีขนาดใหญ่มากมายมหาศาลขนาดนี้ จึงทำให้คอมพิวเตอร์ที่เชื่อมต่อเข้ากับเครือข่ายอินเตอร์เน็ตนั้น มีหลายระดับหลายประเภทด้วยกัน ไม่ว่าจะเป็นคอมพิวเตอร์ในระดับพีซี แลปท็อป เมนเฟรมคอมพิวเตอร์ และรวมถึงโทโพโลยีรูปแบบการเชื่อมต่อเครือข่าย ซึ่งอาจเป็นทั้งแบบอินเตอร์เน็ต โทเค็นริง หรือ FDDI เป็นต้น



ภาพที่ 3.3 แสดงเครือข่ายอินเทอร์เน็ตในมุมมองของ TCP/IP

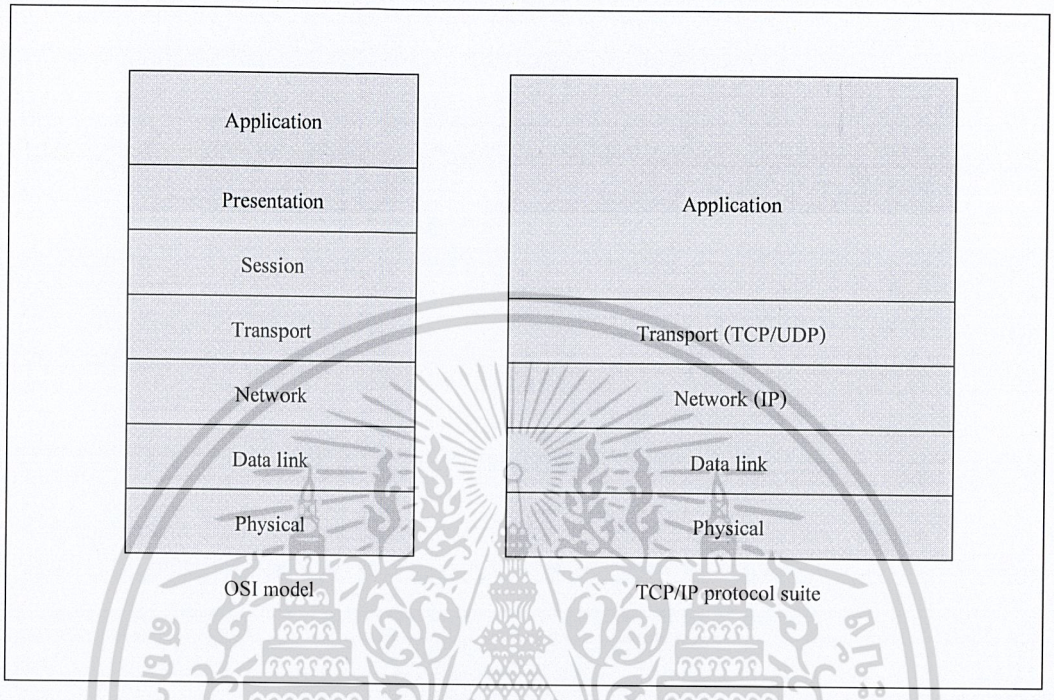
สำหรับมุมมองของ TCP/IP นั้น TCP/IP จะมองเครือข่ายที่ประกอบไปด้วย ฟิสิคัลเน็ตเวิร์กเหล่านั้น เป็นเหมือนหนึ่งเครือข่ายขนาดใหญ่ที่แสดงได้ดังรูปที่ 3.3 กล่าวคือ แต่ละโฮสต์ที่เชื่อมต่อเข้ากับเครือข่าย TCP/IP จะมองในรูปแบบลอจิคัลเน็ตเวิร์ก (Logical Network) ที่เสมือนเป็นหนึ่งเครือข่ายมากกว่าจะมองเครือข่ายเหล่านั้นเป็นฟิสิคัลเน็ตเวิร์กที่ประกอบไปด้วยกลุ่มเครือข่าย

3.5 สถาปัตยกรรมชุดโปรโตคอล TCP/IP

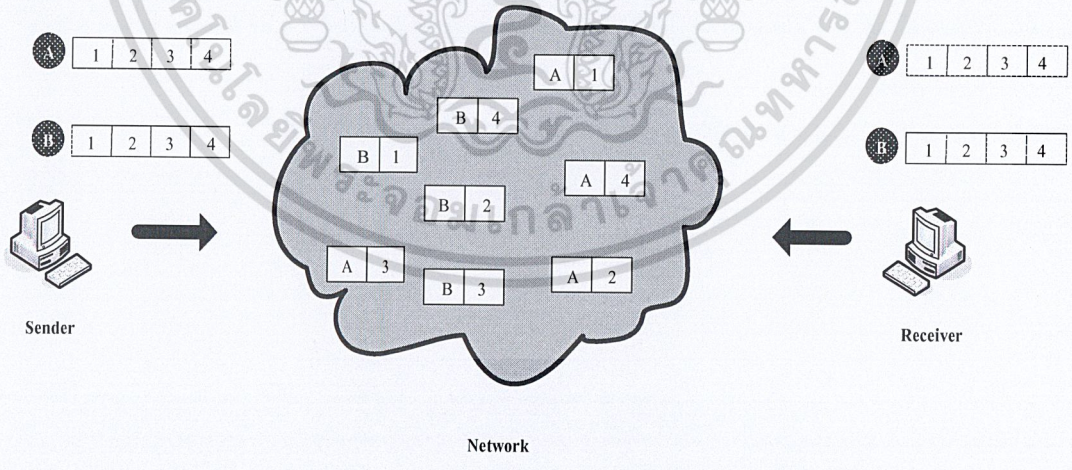
จากรายละเอียดข้างต้น จะเห็นได้ว่าโปรโตคอลมาตรฐานที่ใช้งานบนเครือข่ายระดับสากลหรือเครือข่ายอินเทอร์เน็ตนั้นคือ โปรโตคอล TCP/IP ซึ่งสถาปัตยกรรมชุดโปรโตคอล TCP/IP ได้มีการพัฒนาขึ้นมาก่อนแบบจำลอง OSI ดังนั้น ลำดับชั้นต่างๆ ในโปรโตคอล TCP/IP จึงไม่ตรงกับแบบจำลอง OSI แต่ก็นับได้ว่าเป็นความโชคดี ที่แบบจำลองทั้งสองมีหลักการทำงานที่คล้ายคลึงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากโดย TCP/IP จะมีเพียง 5 ลำดับชั้น ซึ่งประกอบด้วยลำดับชั้นฟิสิคัล ดาต้าลิงก์เน็ตเวิร์ก และแอพพลิเคชัน โดยลำดับชั้นแอพพลิเคชันใน TCP/IP ก็คือการรวมกับของลำดับชั้นเซสชัน พรีเซ็นเตชัน และแอพพลิเคชันของแบบจำลอง OSI



ภาพที่ 3.4 เปรียบเทียบระหว่างแบบจำลอง OSI และสถาปัตยกรรมชุดโปรโตคอล TCP/IP



ภาพที่ 3.5 แสดงการจัดเรียงลำดับหมายเลขของแต่ละดาต้าแกรม

จากรูปที่ 3.5 ทำให้สามารถตรวจสอบได้ว่า มีแพ็กเกจใดที่หายไปหรือยังมาไม่ถึงรวมถึงข้อมูลใดที่มีการส่งมาซ้ำ หากมีข้อมูลการส่งมาซ้ำก็จะสามารถดำเนินการกำจัดแพ็กเกจซ้ำนี้ออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ TCP ยังต้องมีกลไกในการควบคุมการไหลของข้อมูล เพื่อใช้ควบคุมจังหวะการรับส่งระหว่างฝ่ายส่งกับฝ่ายรับไม่ให้มีการส่งข้อมูลจนท่วมล้น ทำให้ปลายทางรับข้อมูลไม่ทัน สำหรับหน้าที่ต่างๆ ของแต่ละลำดับชั้นในสถาปัตยกรรมชุดโปรโตคอล TCP/IP สามารถกล่าวในรายละเอียดได้ดังต่อไปนี้

3.4.1 ลำดับชั้นฟิสิคัลและดาต้าลิงก์ (Physical and Data Link Layer)

ในบางตำราได้มีการรวบรวมลำดับชั้นทั้งสองอยู่ในลำดับชั้นเดียวกัน โดยลำดับชั้นทั้งสองมีหน้าที่ในการควบคุมฮาร์ดแวร์ และการรับส่งข้อมูลผ่านเครือข่าย ลำดับชั้นฟิสิคัลและดาต้าลิงค์นั้น TCP/IP มิได้มีการระบุโปรโตคอลเฉพาะเจาะจงลงไป กล่าวคือจะสนับสนุนมาตรฐานโปรโตคอลทั้งหมดบนระดับดาต้าลิงก์ ไม่ว่าจะเป็นอีเตอร์เน็ตหรือโทเค็นริง และด้วยเหตุดังกล่าว จึงมีเครือข่ายหลายประเภทที่สามารถสื่อสารกับโปรโตคอล TCP/IP ได้ อย่างไรก็ตาม เครือข่าย TCP/IP ที่ใช้งานบนเครือข่ายระดับสากลนี้ยังสามารถนำมาใช้งานเพื่อเชื่อมต่อเป็นเครือข่ายท้องถิ่น (LAN) เครือข่ายระดับเมือง (MAN) หรือเครือข่ายระดับประเทศ (WAN) ได้

3.4.2 ลำดับชั้นเน็ตเวิร์ก (Network Layer)

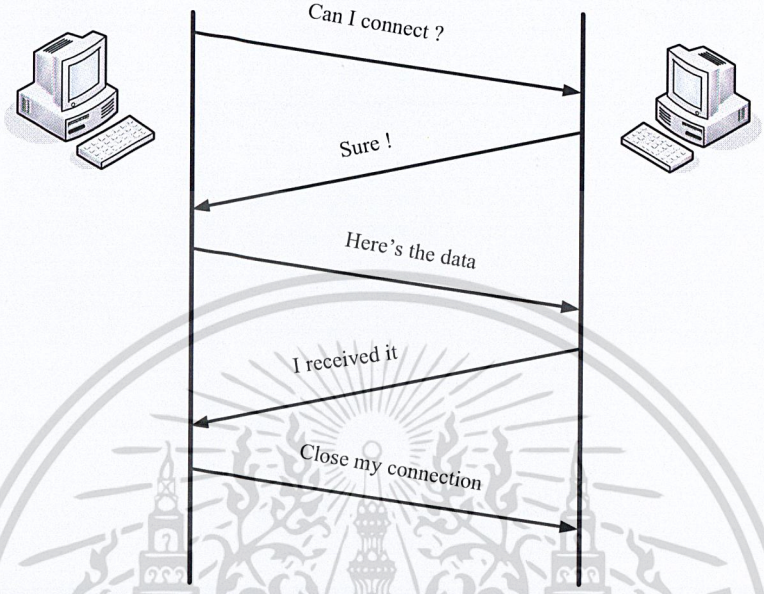
ทำหน้าที่ในการเลือกเส้นทาง เพื่อจัดส่งข้อมูลในรูปแบบของแพ็กเกจ โดยจะมีการใช้อัลกอริทึมในการกำหนดเส้นทาง (Routing Algorithms) เพื่อให้ข้อมูลเดินทางไปถึงปลายทาง ซึ่งโปรโตคอลที่รับผิดชอบในลำดับชั้นเน็ตเวิร์กนี้เรียกว่า IP (Internetworking Protocol) แต่การทำงานของลำดับชั้นนี้จะเป็นเพียงการตัดสินใจว่าจะส่งข้อมูลไปยังเส้นทางใด เพื่อไปถึงปลายทางเท่านั้น ไม่ได้รับประกันว่าข้อมูลที่ส่งไปจะถึงปลายทางหรือไม่ ซึ่งการรับประกันการส่งข้อมูลจะเป็นหน้าที่ของลำดับชั้นทรานสปอร์ต หรือโปรโตคอล TCP

3.4.3 ลำดับชั้นทรานสปอร์ต (Transport Layer)

ในลำดับชั้นทรานสปอร์ต จะทำหน้าที่จัดเตรียมข้อมูลเพื่อส่งจากต้นทางไปยังปลายทาง หรือรับส่งข้อมูลระหว่างโฮสต์ที่อยู่ห่างไกลกันในลักษณะแบบEnd-to-End โดยที่ลำดับชั้นทรานสปอร์ต จะประกอบด้วย โปรโตคอลสองชุดไว้คอยบริการ คือ โปรโตคอล TCP (Transmission Control Protocol) ซึ่งเป็นโปรโตคอลแบบคอนเน็กชันโอเรียนเต็ด (Connection-Oriented) โดยมีการรับรองว่าข้อมูลที่ส่งไปจะส่งถึงมือผู้รับอย่างแน่นอน นอกจากนี้ยังมีโปรโตคอลแบบ UDP (User Datagram Protocol) ซึ่งเป็นโปรโตคอลแบบคอนเน็กชันเลส (Connectionless) ที่ทำงานตรงกันข้ามกับโปรโตคอล TCP กล่าวคือจะไม่สร้างคอนเน็กชันเพื่อการเชื่อมต่อระหว่างโฮสต์ โดยจะส่งข้อมูลทันทีที่ต้องการโปรโตคอล TCP ที่มีการทำงานแบบคอนเน็กชันโอเรียนเต็ด โดยจะมีการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเน็กชัน เพื่อการเชื่อมต่อระหว่างโฮสต์ต้นทางกับปลายทางก่อนทีจะมีการส่งข้อมูลจริง อีกทั้งมีการรับประกันถึงข้อมูลที่ส่งไปด้วยว่าถึงมือผู้รับอย่างแน่นอน



ภาพที่ 3.6 โปรโตคอล TCP ซึ่งเป็นโปรโตคอลแบบคอนเน็กชั่นโอเรียนเต็ด จะมีการสร้างคอนเน็กชั่นเพื่อเชื่อมต่อกับโฮสต์ปลายทางก่อนที่จะดำเนินการส่งข้อมูล

3.4.4 ลำดับชั้นแอปพลิเคชัน (Application Layer)

เป็นลำดับชั้นประยุกต์ ซึ่งเป็นส่วนของผู้ใช้งานที่ใช้ติดต่อกับระบบ อนุญาตให้ยูสเซอร์ที่ใช้งานซอฟต์แวร์แอพพลิเคชันต่างๆ ที่อาจมีหลายรูปแบบด้วยกัน โดยมุ่งเน้นการ interface กับผู้ใช้งานเป็นสำคัญ กล่าวคือ ในลำดับชั้นแอพพลิเคชันนี้จะมีโปรแกรมประยุกต์ต่างๆ มากมายที่จัดเตรียมไว้ เพื่อความสะดวกในการสื่อสารระหว่างยูสเซอร์กับคอมพิวเตอร์ และสนับสนุนการบริการต่างๆ โดยตัวอย่างโปรโตคอลในลำดับชั้นนี้ได้แก่ Telnet, FTP, SMTP, HTTP เป็นต้น

3.4.5 โปรโตคอลในลำดับชั้นเน็ตเวิร์ก

สำหรับในลำดับชั้นเน็ตเวิร์ก นอกจาก TCP/IP ที่สนับสนุนโปรโตคอลหลักอย่าง IP แล้วยังบรรจุโปรโตคอลสนับสนุนอื่นๆ อีก 4 โปรโตคอลด้วยกัน ซึ่งได้แก่ โปรโตคอล ARP, RARP, ICMP และ IGMP

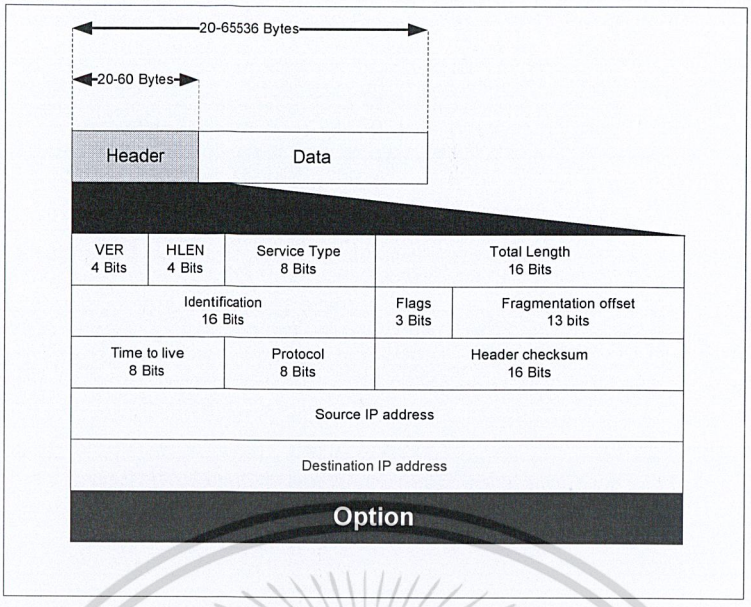
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 โปรโตคอล IP (Internetwork Protocol)

IP เป็นกลไกการส่งข้อมูลที่ใช้โปรโตคอล TCP/IP ในลักษณะคอนเน็กชันเลส โดยจะไม่รับประกันการส่งข้อมูลว่าจะไปถึงผู้รับหรือไม่ ไม่มีการตรวจสอบข้อผิดพลาด และด้วยการปราศจากกลไกการรับประกันข้อมูลที่ส่งไปถึงปลายทาง การไม่มีการตรวจสอบข้อผิดพลาดและไม่ต้องสร้างคอนเน็กชันกับโฮสต์ปลายทาง จึงทำให้หลักการทำงานของโปรโตคอล IP นี้ไม่มีความซับซ้อน โดยมีหน้าที่เพียงนำส่งข้อมูลไปถึงปลายทางได้ด้วยหมายเลข IP ซึ่งเป็นหมายเลขที่ใช้ระบุตำแหน่งเครื่องและเป็นหมายเลขที่ไม่ซ้ำกัน

อย่างไรก็ตาม หากความน่าเชื่อถือในการส่งข้อมูลไปยังปลายทางเป็นสิ่งจำเป็น โปรโตคอล IP ก็จะทำงานควบคู่ไปกับโปรโตคอลที่มีเครื่องมือใน การตรวจสอบข้อมูลว่าส่งไปถึงปลายทางหรือไม่นั่นก็คือโปรโตคอล TCP ซึ่งสามารถอธิบายเพื่อให้เห็นภาพได้ชัดเจนด้วยการเปรียบเทียบกับการส่ง จดหมายไปรษณีย์ โดยการส่งจดหมายแบบปกติ ผู้ส่งจะนำจดหมายมาใส่ซองติดแสตมป์และนำไปหยอดลงในตู้ส่งจดหมาย จากนั้นเมื่อถึงเวลาบุรุษไปรษณีย์ก็จะเปิดตู้จดหมาย เพื่อนำจดหมายนี้ส่งไปถึงผู้รับปลายทางตามที่อยู่ที่ได้จ่าหน้าซองไว้ ซึ่งการส่งจดหมายในลักษณะนี้ จะไม่มีการรับประกันการส่งว่าจดหมายนี้จะถึงผู้รับปลายทางหรือไม่ จดหมายอาจมีการตกหล่น หรือสูญหายระหว่างทางก็เป็นได้ ดังนั้น หากผู้ส่งต้องการความน่าเชื่อถือด้วยวิธีรับประกันว่า จดหมายฉบับนี้จะส่งถึงมือผู้รับอย่างแน่นอน หรือหากไม่ถึงผู้รับ ก็จะต้องได้รับแจ้งข่างสารกลับมาให้ทราบ การส่งจดหมายในลักษณะนี้จึงจำเป็นต้องมีการลงทะเบียน โดยผู้รับจะต้องมีการเซ็นรับจดหมายเพื่อยืนยันว่าได้รับจดหมายฉบับนี้จริง จึงถือเป็นกระบวนการส่งจดหมายถึงผู้รับเสร็จสมบูรณ์ ดังนั้น IPก็เปรียบเสมือนกับการส่งจดหมายธรรมดา ในขณะที่ TCP ก็คือการส่งจดหมายแบบลงทะเบียนที่มีการรับประกันการส่งถึงมือผู้รับนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.7 ไอพี ดาต้าแกรม (IP datagram)

3.7 Time To Live (TTL)

เป็นฟิลด์ที่ใช้สำหรับกำหนดอายุขัยของดาต้าแกรม โดยโฮสต์ฝ่ายส่งจะมีการกำหนดค่าเริ่มต้นของอายุขัยให้กับดาต้าแกรม (โดยทั่วไปค่าที่กำหนดไว้คือ 32 หรือ 64) และเมื่อดาต้าแกรมนี้ได้เดินทางผ่านเครือข่ายอินเทอร์เน็ต จากเร้าเตอร์ไปยังเร้าเตอร์ เร้าเตอร์แต่ละตัวก็จะทำการลดค่านี้ลงทีละหน่วย และหากค่าดังกล่าวได้ถูกลดค่าลง จนกระทั่งมีค่าเป็นศูนย์ก่อนที่ดาต้าแกรมเหล่านั้นจะเดินทางไปถึงปลายทางสุดท้ายดาต้าแกรมที่หมดอายุขัยนี้ก็จะถูกละทิ้งไป ซึ่งกระบวนการดังกล่าวจะเป็นการป้องกันมิให้ดาต้าแกรมที่มีปัญหาส่งข้อมูลวนเวียนอยู่ภายในเครือข่ายนั่นเอง

3.7.1 Protocol

เป็นฟิลด์ที่ใช้ระบุชนิดของโปรโตคอลในลำดับชั้นส่วนบน (Upper-Layer) เพื่อจะได้เป็นตัวกำหนดว่าจะส่งให้กับ TCP หรือ UDP

3.7.2 Header Checksum

เป็นฟิลด์ขนาด 16 บิต ที่ใช้สำหรับตรวจสอบความถูกต้องสมบูรณ์ของเฮดเดอร์โดยการตรวจสอบในที่นี้จะทำเฉพาะในส่วนของเฮดเดอร์เท่านั้นไม่ได้รวมแพ็กเก็ตข้อมูล

3.7.3 Source Address

คือหมายเลข IP ของโฮสต์ต้นทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.4 Destination Address

คือหมายเลข IP ของโฮสต์ปลายทาง

3.7.5 Options

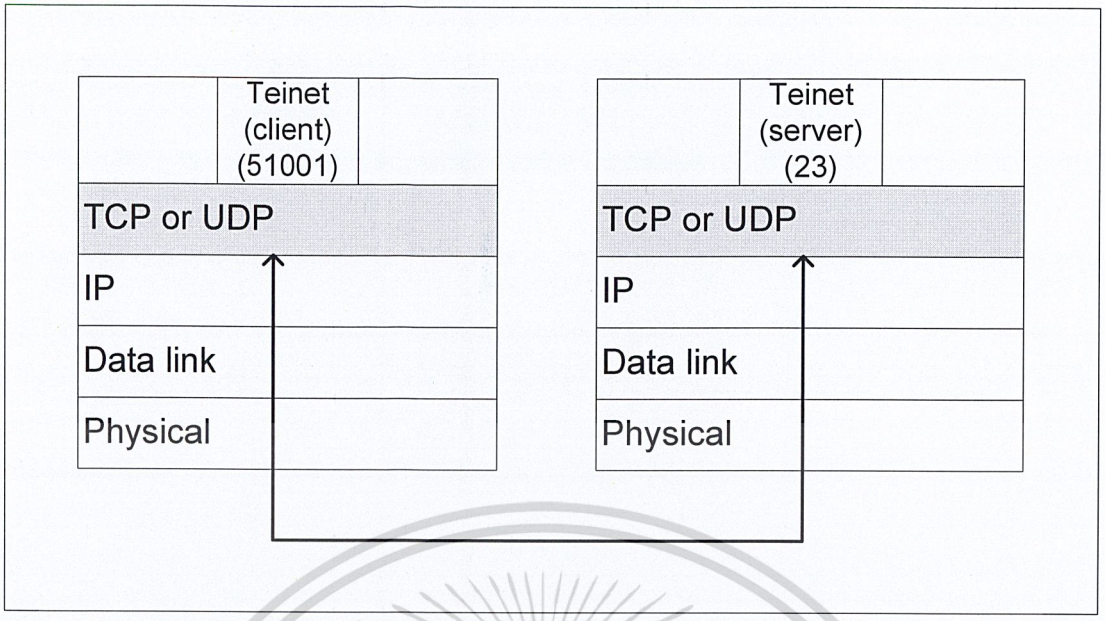
เป็นส่วนเพิ่มเติม ในกรณีที่ต้องการให้กำหนดหน้าที่เพิ่มเติมให้กับไอพีดาต้าแกรมโดยอาจมีหน้าที่เกี่ยวกับการควบคุมเส้นทาง เวลา การจัดการ และวางแนวทาง เป็นต้น

3.8 โปรโตคอลในลำดับชั้นทรานสปอร์ต

ในลำดับชั้นทรานสปอร์ตจะประกอบด้วย 2 โปรโตคอลหลักๆ ด้วยกันคือโปรโตคอล TCP และ UDP โดยรับผิดชอบด้านการส่งข้อมูลในลักษณะ End-to-End Delivery ที่รับประกันการส่งด้วยโปรโตคอล TCP สำหรับไอพีจะทำการส่งมอบข้อมูล หรือดาต้าแกรมจากโฮสต์ต้นทางไปยังโฮสต์ปลายทาง โดยเป็นไปในรูปแบบของ Host-to-Host Protocol ซึ่งเป็นเพียงการส่งข้อมูลจากเครื่องหนึ่งไปยังอีกเครื่องหนึ่งเท่านั้น แต่ในปัจจุบันระบบปฏิบัติการหรือโอเอสมักสนับสนุนการทำงานในลักษณะมัลติยูสเซอร์ หรือการประมวลผลหลายๆงาน โดยโปรแกรมที่ได้รับการประมวลผลเราจะเรียกว่ากระบวนการ (process) และโฮสต์อาจจำเป็นต้องมีการประมวลผลกระบวนการต่างๆ หลายกระบวนการในช่วงเวลาเดียวกัน

โปรโตคอลในลำดับชั้นทรานสปอร์ต ที่เป็นกลุ่มชุดของ TCP/IP ได้มีการกำหนดการเชื่อมต่อในแต่ละกระบวนการที่เรียกว่า พอร์ต (Port) โดยพอร์ตจะเป็นหลายเลขกำกับช่องสื่อสารที่ใช้เป็นจุดเชื่อมต่อเพื่อการส่งข้อมูลระหว่างโปรแกรม ดังนั้น พอร์ตก็เปรียบเสมือนกับแอดเดรสบนโปรโตคอลระดับแอปพลิเคชันนั่นเอง โดย interface ระหว่างกระบวนการและพอร์ต จะดำเนินการโดยโปรแกรมระบบปฏิบัติการของโฮสต์ ไอพีเป็นโปรโตคอลในรูปแบบ Host-to-Hostซึ่งหมายถึงการส่งมอบแพ็กเก็ตจากอุปกรณ์หนึ่งไปยังอุปกรณ์อื่นๆ ในขณะที่ลำดับชั้นทรานสปอร์ตจะเป็นโปรโตคอลในรูปแบบ Port-to-Port ที่ทำงานอยู่บนโปรโตคอลไอพี โดยจะส่งแพ็กเก็ตจากพอร์ตต้นทางให้กับไอพีเพื่อ ขอใช้บริการโปรโตคอลไอพีเพื่อดำเนินการส่งข้อมูล และจากการบริการของไอพีนี้เองก็จะส่งข้อมูลไปยังพอร์ตปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 การติดต่อด้วยหมายเลขพอร์ตระหว่างเครื่องไคลเอนต์และเซิร์ฟเวอร์

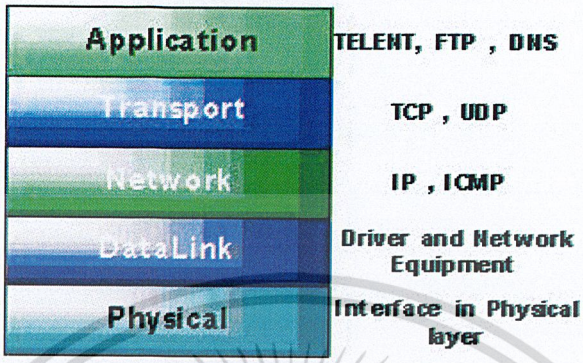
พอร์ตแต่ละหมายเลขจะถูกกำหนดด้วยหมายเลขแอดเดรสที่เป็นเลขจำนวนเต็มที่มี ขนาด 16 บิต จำนวนบิตดังกล่าว ถือว่ามีเพียงพอต่อการนำไปใช้งาน เนื่องจากขนาด 16 บิตนี้สามารถสนับสนุนพอร์ตได้มากถึง 65,536 พอร์ตด้วยกัน (0 – 65535) โดยหมายเลขพอร์ตตั้งแต่ 0 – 1023 นี้จะถูกสงวนเพื่อการบริการมาตรฐานของโฮสต์ที่เป็นเซิร์ฟเวอร์ ที่เรียกว่า Well-Know Ports ในขณะที่ไคลเอนต์ที่ต้องการขอติดต่อกับเซิร์ฟเวอร์นั้น จะใช้หมายเลขพอร์ตที่มีค่าตั้งแต่ 1024 เป็นต้นไป โดยตัวซอฟต์แวร์ TCP จะเป็นตัวกำหนดหมายเลขพอร์ตที่ยังว่างอยู่ให้ซึ่งจะต้องไม่ซ้ำกันกับที่มีการใช้งานอยู่ของเครื่องไคลเอนต์นั้นๆ สำหรับการมัลติเพล็กซ์ในลำดับชั้นนี้จะใช้แนวคิดที่เรียกว่าซ็อกเก็ต

ตารางที่ 3.1 ตัวอย่างลักษณะของแต่ละหมายเลขพอร์ต

Port Number	Service	Description
20	FTP(Data)	File Transfer Protocol and Data Used for transfering files
21	FTP(Control)	File Transfer Protocol and Control Used for transfering files
23	TELNET	telnet, used to gain "remote control" over another Machine on the network
25	SMTP	Simple Mail Transfer Protocol, used for transfering e-mail between e-mail service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นตัวที่ใช้เพื่อแยกแยะกระบวนการของฝ่ายต้นทางกับฝ่ายปลายทางที่ติดต่อกัน และทำให้สามารถเปิดเซสชันหลายเซสชันพร้อมกันได้ โดยซ็อกเก็ตจะประกอบด้วย 3 สิ่งด้วยกัน คือ หมายเลข IP โปรโตคอลทรานสปอร์ต หมายเลขพอร์ต



ภาพที่ 3.9 ตัวอย่างโปรโตคอลในลำดับชั้นแอปพลิเคชั่น

3.9 UDP Protocol

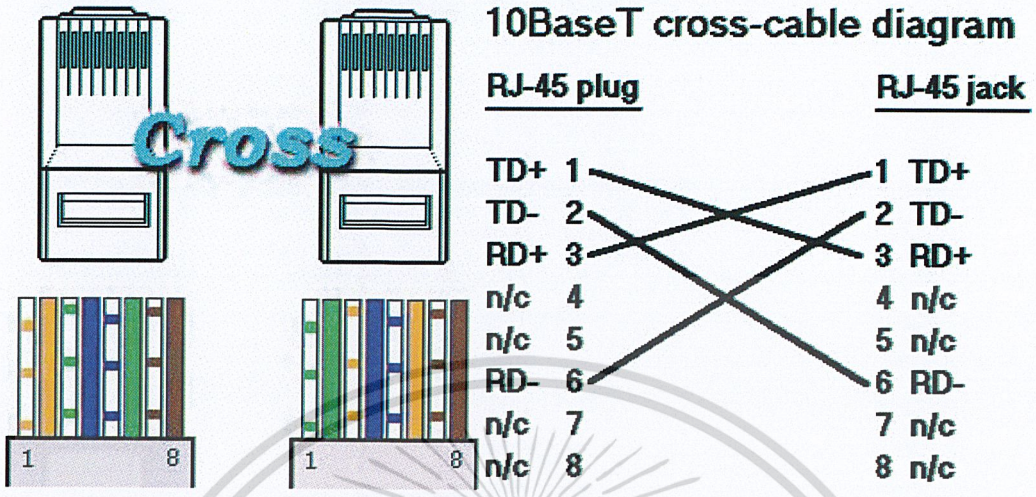
เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล กลไกการตรวจสอบโดย checksum ของ UDP นั้น เพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไข หรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เองจะไม่มีการยืนยันหรือแจ้งให้ผู้ส่งทราบ

3.9 การต่อ Ethernet LAN

สำหรับการเชื่อมต่อกับเครือข่าย Network ระหว่างบอร์ด CP-JR ARM7 LPC2368 นั้น จะใช้ขั้วต่อมาตรฐาน Ethernet แบบ RJ45 โดยวงจรส่วนนี้จะใช้ขาสัญญาณ P1[0,1,4,8,9,10,14..17] ในการเชื่อมต่อโดยใช้ Chips Physical Ethernet เบอร์ DP83848 เป็น Driver ในการเชื่อมต่อสำหรับวิธีการเชื่อมต่อสายสัญญาณ Ethernet LAN ของบอร์ดเข้ากับระบบเครือข่ายจะทำได้ 2 แบบด้วยกัน คือ การต่อแบบ Direct Line และต่อผ่าน Hub

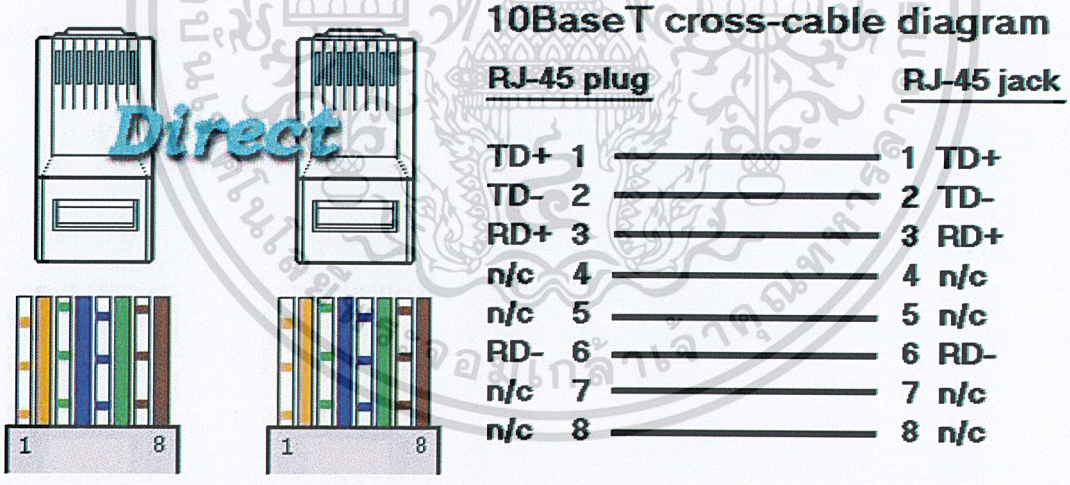
เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 1 คือ การเชื่อมต่อเข้ากับคอมพิวเตอร์โดยตรง สาย LAN จะต้องเข้าสายแบบ Cross



ภาพที่ 3.10 ลักษณะของการเข้าหัว RJ-45 แบบ cross

กรณีที่ 2 คือ การเชื่อมต่อผ่าน Hub ของเครื่องคอมพิวเตอร์ Server จะต้องเข้าสายแบบ Direct



ภาพที่ 3.11 ลักษณะของการเข้าหัว RJ-45 แบบ Direct

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทำงานและการทดลองตู้เอกสารควบคุมผ่านทางอินเตอร์เน็ต

4.1 กล่าวนำ

จากที่ได้กล่าวมาในบทก่อนหน้านี้ ได้ทำการออกแบบและจัดเตรียมอุปกรณ์ต่างๆ เพื่อสร้างระบบควบคุมตู้เก็บเอกสารผ่านทางอินเตอร์เน็ต และได้ทำการสร้างระบบควบคุมตู้ให้เปิดและปิดตามที่ต้องการได้โดยสามารถทำการควบคุมการเปิดและปิดตู้เก็บเอกสารผ่านทางเครือข่ายอินเตอร์เน็ตได้ ซึ่งในบทนี้เราก็จะทำการทดลองให้ได้ตามวัตถุประสงค์

4.2 ขั้นตอนการทดลอง

4.2.1 การทำงานของระบบ

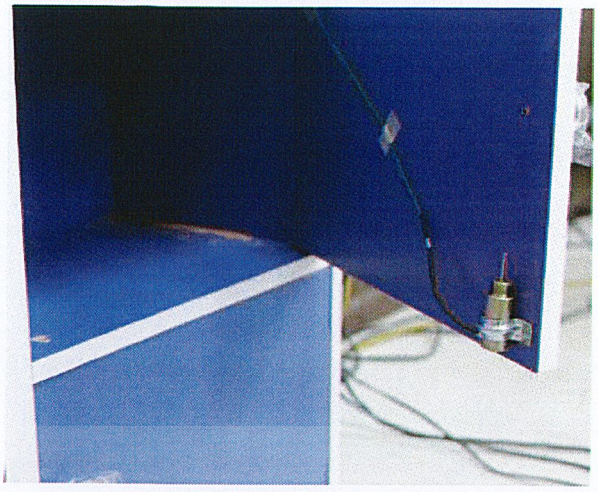
สำหรับระบบการทำงานของโปรแกรมของระบบควบคุมตู้เก็บเอกสารนั้น จะประกอบไปด้วย ระบบล็อคแบบอิเล็คทรอนิกส์ ระบบตรวจสอบสถานะของตู้ ระบบควบคุมการเปิดตู้-ล็อคตู้ และระบบการสั่งงานผ่านเครือข่ายอินเตอร์เน็ต

1. ระบบล็อคแบบอิเล็คทรอนิกส์
2. ระบบตรวจสอบสถานะของตู้
3. ระบบควบคุมการเปิดตู้-ล็อคตู้
4. ระบบการสั่งงานผ่านเครือข่ายอินเตอร์เน็ต

4.2.1.1 ติดตั้งระบบล็อคแบบอิเล็คทรอนิกส์

ติดตั้งโซลินอยด์เข้าที่ประตูด้านในและเจาะรูที่ตัวตู้เก็บเอกสารด้านใน เพื่อทำหน้าที่เป็นกลอนล็อคแบบอิเล็กทรอนิกส์เมื่อป้อนกระแสไฟฟ้า โดยแสดงดังภาพที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.1 ติดตั้งระบบล็อคแบบอิเล็กทรอนิกส์

4.2.1.2 ติดตั้งระบบตรวจสอบสถานะของตู้

ระบบตรวจสอบสถานะของตู้เก็บเอกสารนั้นสามารถทำได้ โดยการติดตั้งลิมิตสวิตซ์ โดยติดตั้งเข้ากับทั้งสองด้านของตู้เก็บเอกสาร เพื่อทำหน้าที่ในการตรวจสอบสถานะการเปิดและปิดของตู้เก็บเอกสารดังภาพที่ 4.2

ภาพที่ 4.2 การติดตั้งระบบตรวจสอบสถานะของตู้

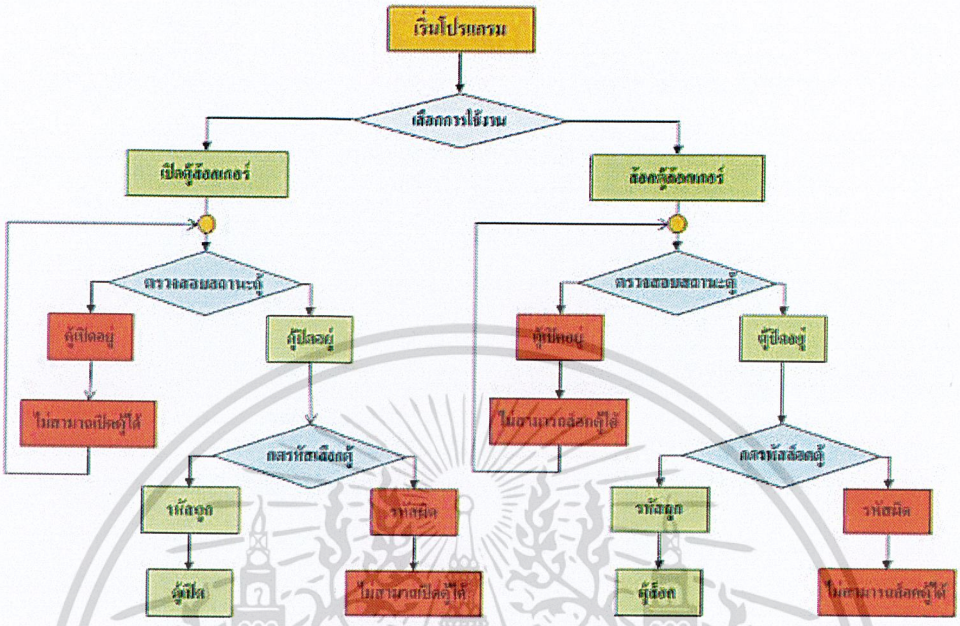
4.2.1.3 ระบบควบคุมการเปิดตู้-ล็อคตู้

- **เขียนโปรแกรมที่ทำงานบนไมโครคอนโทลเลอร์ Arduino**

สำหรับโปรแกรมที่ทำงานบน Arduino นั้น ทำหน้าที่อยู่ 2 อย่างคือ นำสัญญาณที่ออกทางขาเอาท์พุทของบอร์ด (ในปริญญานิพนธ์นี้ใช้ขา Digital 22-27) เพื่อนำไปเป็นอินพุทให้กับ Relay สำหรับสั่งเปิดตู้-ล็อคตู้ หน้าที่อีกประการหนึ่งก็คือ อ่านค่าจากอินพุตของบอร์ด (ในปริญญานิพนธ์นี้ใช้ขา Digital 30-41) ซึ่งได้มาจากเอาท์พุตของลิมิตสวิตซ์ เพื่อบอกสถานะของตู้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปิดหรือปิด โดยโปรแกรมจะเขียนด้วยภาษาซี โดยใช้ Arduino environment เป็นคอมไพเลอร์ ซึ่งโครงสร้างการทำงานของโปรแกรมแสดงดังโฟลชาร์ตในภาพที่ 4.3

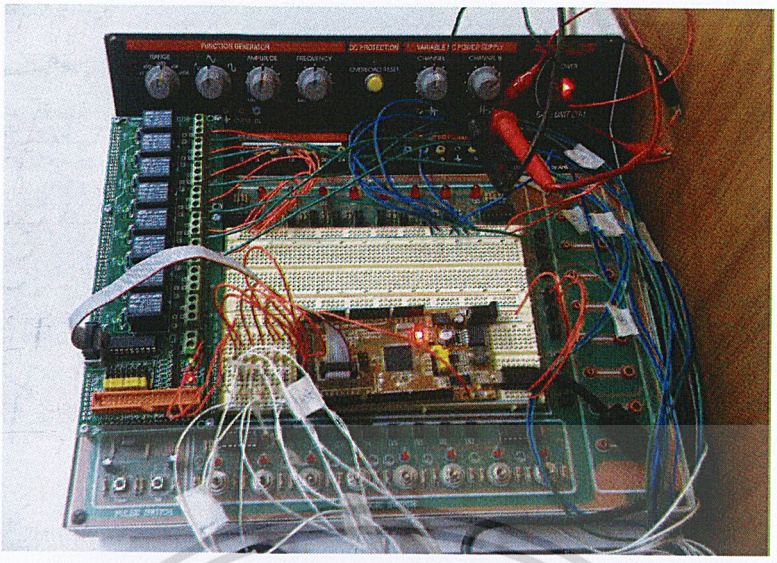


ภาพที่ 4.3 โฟลชาร์ตการทำงานของโปรแกรม

- **ติดตั้งระบบควบคุมการเปิดตู้-ล็อคตู้**

การติดตั้งระบบควบคุมการเปิดและปิดสามารถทำได้ โดยการติดตั้งบอร์ดไมโครคอนโทรลเลอร์และบอร์ดรีเลย์เข้ากับตู้เก็บเอกสาร โดยที่เมื่อมีการเปิดและปิดประตูตู้เก็บเอกสารลิมิตสวิตซ์ที่เป็นตัวตรวจสอบสถานะการเปิดและปิดนั้น จะส่งสัญญาณกลับมาประมวลผลที่บอร์ดไมโครคอนโทรลเลอร์ ซึ่งในปริญญานิพนธ์นี้ได้ใช้บอร์ดไมโครคอนโทลเลอร์รุ่น ATMEGA 1280 และสามารถแสดงผลการเปิดและปิดของตู้เก็บเอกสารโดย LED ที่บอร์ดรีเลย์ ซึ่งสามารถแสดงได้ดังภาพที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.4 การติดตั้งระบบควบคุมการเปิด ล็อกตู้เก็บเอกสาร

4.2.1.4 ระบบการสั่งงานผ่านเครือข่ายอินเตอร์เน็ต

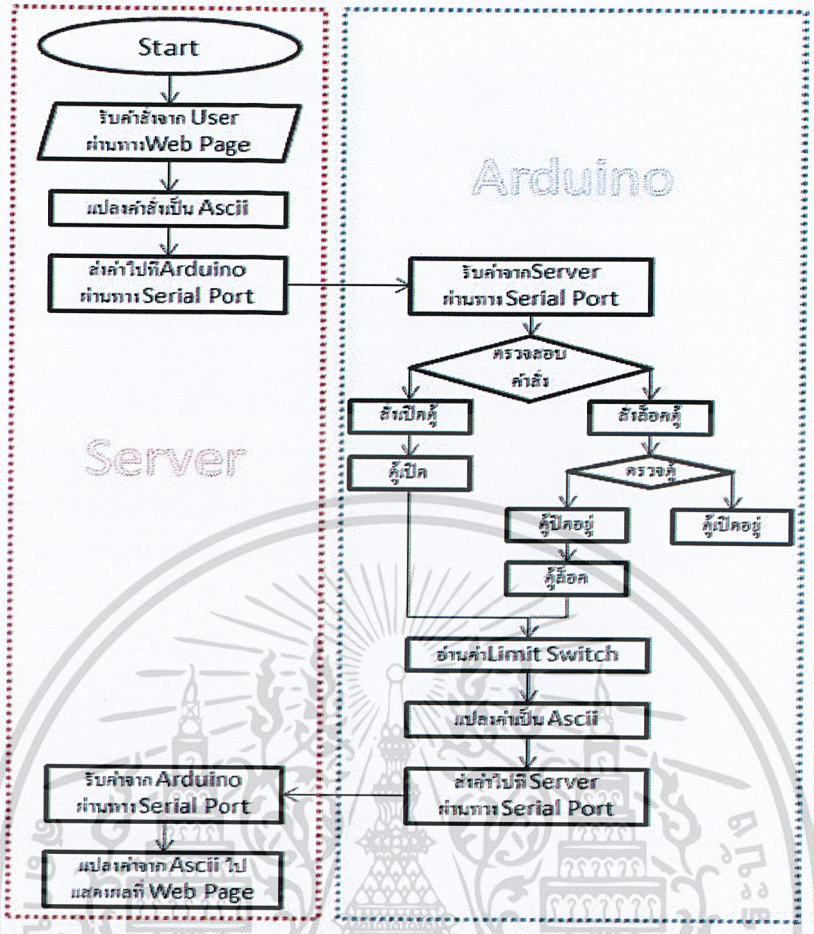
- **โปรแกรมที่ทำงานบนไมโครคอนโทลเลอร์ Arduino**

สำหรับโปรแกรมที่ทำงานบน Arduino นั้น ทำหน้าที่อยู่ 2 อย่างคือ หนึ่งทำหน้าที่รอรับค่าจากเครื่อง server ผ่านทางพอร์ตอนุกรม (serial port) แล้วทำการนำสัญญาณที่ออกทางขาเอาท์พุทของบอร์ด (ในปริญญานิพนธ์นี้ใช้ขา Digital 22-27) เพื่อนำไปเป็นอินพุทให้กับ Relay สำหรับสั่งเปิด-ปิดตู้ หน้าที่อีกประการหนึ่งก็คือ อ่านค่าจากอินพุตของบอร์ด (ในปริญญานิพนธ์นี้ใช้ขา Digital 30-41) ซึ่งได้มาจากเอาท์พุตของลิมิตสวิตซ์ จากนั้นส่งสัญญาณไปยัง server ผ่านทางพอร์ตอนุกรม (serial port) เพื่อบอกสถานะของตู้ว่าเปิดหรือปิด โดยโปรแกรมจะเขียนด้วยภาษาซี โดยใช้ Arduino environment เป็นคอมไพเลอร์

- **โปรแกรมที่ทำงานบน Server**

สำหรับโปรแกรมที่ทำงานบนอุปกรณ์ Server เป็นโปรแกรมที่เขียนด้วยภาษา php และ html และทำงานอยู่บนอุปกรณ์ Server ซึ่งใช้โปรแกรม Flash ในการแสดงผลเมื่อเข้าใช้งานโปรแกรมผ่านทางเว็บบราวเซอร์ โดยโปรแกรมจะประกอบไปด้วยระบบการสั่งงานควบคุมการเปิดตู้-ล็อคตู้ และระบบตรวจสอบสถานะการทำงานของตู้ ซึ่งโครงสร้างการทำงานของโปรแกรมที่ทำงานบนไมโครคอนโทรลเลอร์และโปรแกรมที่ทำงานบน Server แสดงดังโฟลชาร์ตในภาพที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.5 แสดงโฟลชาร์ตแสดงการทำงานของโปรแกรม

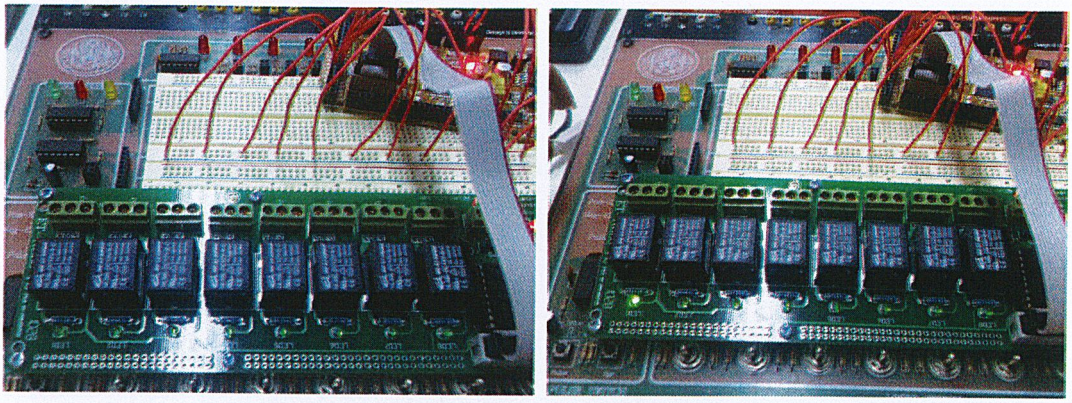
4.2.2 การทดสอบระบบ

เพื่อให้แน่ใจว่าระบบที่ออกแบบ สามารถทำงานได้จริง จึงได้ทำการทดสอบการทำงานของโปรแกรม เพื่อทดสอบว่าโปรแกรมสามารถรับค่าจาก Server มาที่ Arduino แล้วส่งสัญญาณออกทางเอาท์พุทเพื่อไปเปิดตู้และล็อคตู้แล้วส่งค่าสถานะของตู้กลับมาที่ Server ได้จริง

โดยขั้นตอนการทดลองมีดังนี้

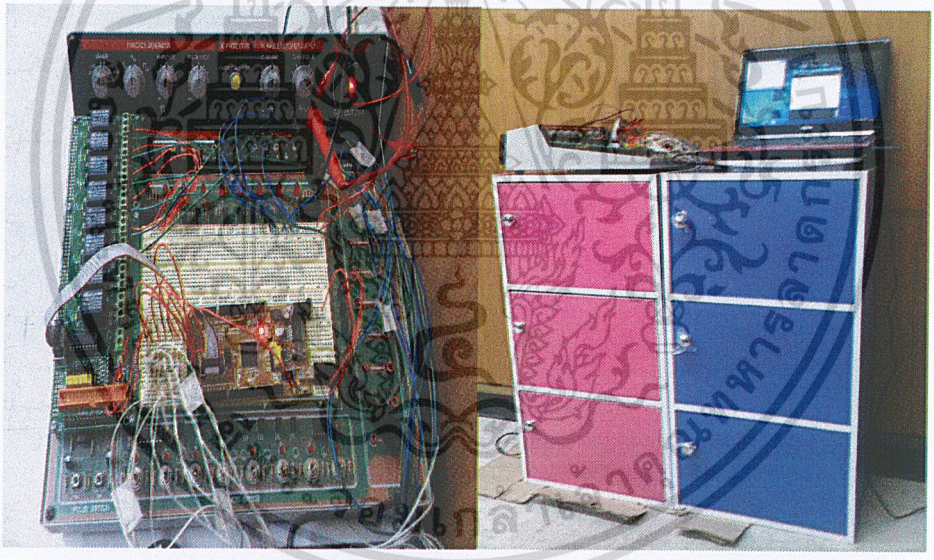
1. ต่อวงจรตามภาพที่ 4.6 โดยมีบอร์ดรีเลย์ซึ่งต่ออยู่กับขาเอาท์พุทของบอร์ด Arduino และป้อนคำสั่งจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.6 การต่อวงจรควบคุมการเปิดและล็อคตู้เก็บเอกสาร

2. เชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับบอร์ดรีเลย์แล้วเชื่อมต่อ บอร์ดรีเลย์กับระบบล็อคอิเล็กทรอนิกส์ของตู้ แล้วป้อนคำสั่งเพื่อทำการเปิดตู้-ล็อคตู้ผ่านทางคอมพิวเตอร์ แสดงได้ดังภาพที่ 4.7



ภาพที่ 4.7 การสั่งเปิดและล็อคตู้ผ่านคอมพิวเตอร์

3. ทดสอบการรับส่งค่าผ่านทางพอร์ตอนุกรม (Serial Port) โดยใช้โปรแกรม Hyper terminal โดยป้อนรหัสเข้าไปที่หน้าจอโปรแกรม Hyper terminal ก็สามารถควบคุมการเปิดและล็อคตู้เก็บเอกสารซึ่งสามารถแสดงได้ดังภาพที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 4.8 การสั่งการเปิดและล็อคตู้เก็บเอกสารผ่านโปรแกรม Hyper terminal

4. ทดสอบการสั่งงานเปิดตู้-ล็อคตู้ผ่านทางอินเตอร์เน็ต สามารถทำได้โดยการเข้าไปที่หน้าเว็บเพจที่สร้างขึ้น เพื่อใช้ในการควบคุมการเปิดและล็อคตู้เก็บเอกสาร โดยในปริญญานิพนธ์นี้ คือ http://161.246.20.99/project.php และทำการกดปุ่ม ON ที่หน้าเว็บเพจ ณ ตำแหน่งของตู้เก็บเอกสาร ตู้ก็จะเปิด และเมื่อกดปุ่ม OFF ตู้ก็จะปิด แสดงดังภาพที่ 4.9

ภาพที่ 4.9 การเปิดและล็อคตู้เก็บเอกสารผ่านทางอินเตอร์เน็ต

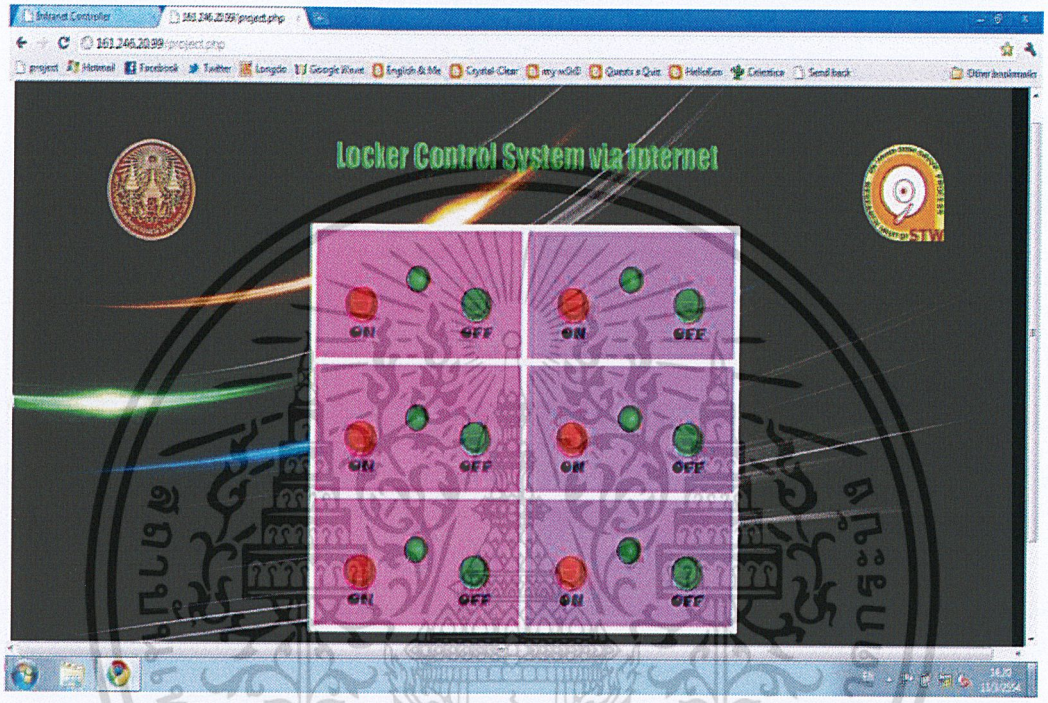
4.3 การใช้งานโปรแกรม

4.3.1 การสั่งงานผ่านอินเตอร์เน็ต

สามารถสั่งการผ่านอินเตอร์เน็ตโดยใช้เว็บบราวเซอร์ เช่น Google Chrome , Internet Explorer โดยเปิดหน้าเว็บเพจที่จะใช้ในการสั่งงาน ซึ่งหน้าเว็บเพจในปริญญานิพนธ์นี้คือ http://161.246.20.99/project.php เมื่อเปิดหน้าเว็บเพจจะแสดงตู้เก็บเอกสารและปุ่มกดในการเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และปิดของตู้เก็บเอกสาร เมื่อทำการกดปุ่ม ON ณ ตำแหน่งของตู้ จะมีสัญญาณส่งกลับมาที่เซอร์ฟเวอร์ เซอร์ฟเวอร์จะส่งคำสั่งมาที่บอร์ดไมโครคอนโทรลเลอร์ ATMEGA 1280 เพื่อทำการประมวลผลและควบคุมการทำงานของกลอนโซลินอยด์ และส่งสัญญาณเอาท์พุตออกไปเพื่อเปิดตู้เก็บเอกสาร จากนั้น LED ที่แสดงสถานะหน้าเว็บเพจจะแสดงสถานะ ON และเมื่อกดปุ่ม OFF ก็จะสามารถสั่งล็อกตู้เก็บเอกสาร LED หน้าเว็บเพจจะแสดงสถานะ OFF



รูปที่ 4.10 แสดงเว็บเพจที่ใช้ในการควบคุมการเปิดและปิดตู้เก็บเอกสาร

4.4 ผลการทำงาน

เมื่อเปิดหน้าเว็บเพจคือ http://161.246.20.99/project.php ผ่านทางเครือข่ายอินเทอร์เน็ตแล้วกดปุ่มเปิดหรือปิดในหน้าเว็บเพจนั้น จะเป็นการส่งคำสั่งในการเปิดหรือปิดตู้เก็บเอกสารตาม code ที่ได้กำหนดไว้เพื่อระบุว่าเปิดตู้เก็บเอกสารช่องใด จากนั้นเซิร์ฟเวอร์จะส่ง response กลับมาที่บอร์ด Arduino ผ่านทาง serial port โดยจะแสดงผลผ่านหน้าเว็บเพจ โดยจะให้ LED ตามตำแหน่งตู้เก็บเอกสารที่ได้ระบุไว้ตอนที่ใส่รหัสนั้นติดหรือดับ เมื่อ LED นั้นติดแสดงว่า ตู้เก็บเอกสารนั้นสามารถเปิดได้ และเมื่อ LED นั้นดับคือการแสดงสถานะว่าตู้เก็บเอกสารนั้นปิด และผลการทำงานสามารถควบคุมการทำงานของตู้เก็บเอกสาร โดยการเปิดหรือปิดตู้เก็บเอกสารผ่านทางเครือข่ายอินเทอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 สรุปผล

จากการทดลองดังกล่าว เมื่อทำการเปิดหน้าเว็บเพจด้วยเว็บเบราว์เซอร์ที่สร้างไว้ในที่นี้ คือ http://161.246.20.99/project.php และทำการทดลองดังที่กล่าวมาข้างต้น สามารถควบคุมการทำงานของตู้เก็บเอกสารให้เปิดและปิดตามคำสั่งผ่านทางเครือข่ายอินเตอร์เน็ตได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการดำเนินการ

เมื่อได้ออกแบบระบบควบคุมการเปิดและล็อคตู้เก็บเอกสาร โดยทำการติดตั้งระบบล็อคแบบอิเล็กทรอนิกส์ ติดตั้งระบบตรวจสอบสถานะการเปิดและปิด โดยการติดตั้งลิมิตสวิตซ์เข้าไปทั้งสองตัวเพื่อความแน่ใจว่าลิมิตสวิตซ์ตัวใดตัวหนึ่งจะไม่ชำรุดเสียหาย รวมทั้งการสั่งเปิดและปิดตู้เก็บเอกสารผ่านทางคอมพิวเตอร์ และสั่งเปิดและปิดผ่านทางอินเตอร์เน็ต ผลการทดลองที่ได้ คือ สามารถออกแบบตู้เก็บเอกสารและทำระบบเปิด-ล็อคตู้เก็บเอกสาร โดยใช้โซลินอยด์เป็นตัวล็อคแบบอิเล็กทรอนิกส์ และเขียนโปรแกรมด้วยไมโครคอนโทรลเลอร์ Arduino เพื่อควบคุมระบบเปิด-ล็อคของตู้เก็บเอกสาร รวมไปถึงการเขียนโปรแกรม Web Base เพื่อให้สามารถควบคุมตู้เก็บเอกสารจากระยะไกลได้โดยผ่านทางอินเตอร์เน็ต

5.2 ข้อเสนอแนะ

เราอาจจะใช้ Uninterruptible Power Supply มาเป็นอุปกรณ์ในการสำรองไฟฟ้าในกรณีที่ไฟฟ้าเกิดการขัดข้อง อย่างเช่นเวลาไฟดับได้

5.2.1 ระบบสำรองไฟ

เนื่องจากระบบไฟฟ้าที่จ่ายให้กับผู้ใช้นั้น ความผิดปกติสามารถเกิดขึ้นได้ในทุกๆส่วนที่มีระบบจ่ายไปถึง ความผิดปกติและปัญหาในระบบไฟฟ้าเกิดขึ้นเนื่องจากหลายๆสาเหตุ และมีผลกระทบกับผู้ใช้ระบบไฟฟ้ามากน้อยต่างกัน ตามระดับของความผิดปกติที่เกิดขึ้น ซึ่งความเสียหายที่เกิดขึ้นมีทั้งความเสียหายเล็กน้อยไปถึงในวงกว้าง ซึ่งผู้ใช้ระบบไฟฟ้าสามารถป้องกันความผิดปกติและปัญหาในระบบไฟฟ้าได้หลายวิธีขึ้นอยู่กับ ลักษณะการใช้งานในระบบไฟฟ้านั้น วิธีหนึ่งซึ่งให้ประสิทธิภาพในการป้องกันตามความผิดปกติในระบบไฟฟ้าได้ดี และเป็นที่นิยมมาก คือ ระบบสำรองไฟฟ้าแบบต่อเนื่อง

Uninterruptible Power Supply หรือเรียกว่า UPS

UPS มีหลักการทำงาน คือ ในสภาวะที่ระบบไฟฟ้าปกติเครื่อง UPS จะทำการเก็บกำลังไฟฟ้าสำรองไว้ในชุดเก็บกระแสไฟฟ้าสำรอง ซึ่งแบตเตอรี่จะนำมาใช้ในการเก็บกระแสไฟฟ้าสำรองไว้ และเมื่อระบบไฟฟ้ามีความผิดปกติ เครื่อง UPS จะทำการ จ่ายไฟฟ้าที่เก็บสำรองไว้ส่งผ่านไปที่ผู้ใช้ไฟฟ้า ซึ่งความเร็วของการส่งผ่านพลังงานไฟฟ้านี้ขึ้นอยู่กับชนิดของชนิด UPSกำลังไฟฟ้าสำรองที่ถูกจ่ายออกมาจาก UPS นี้ จะถูกจ่ายออกมาเป็นเวลาชั่วคราว เพียงพอที่จะสำรองไฟฟ้าให้กับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไม่ทำให้เกิดปัญหาหรือความเสียหายต่อผู้ใช้ในกรณีที่ระบบไฟฟ้าถูกตัดตอนโดยทันที ซึ่งในบางส่วนของผู้ใช้ระบบเครื่องกำเนิดไฟฟ้าแบบเครื่องยนต์ ได้ถูกติดตั้งไว้แล้วซึ่งกำลังไฟฟ้าสำรองจาก UPS สามารถสำรองได้เพียงพอก่อนที่เครื่องกำเนิดไฟฟ้าจะทำการจ่ายไฟฟ้าเข้ามาแทนที่ นอกจาก UPS จะสามารถสำรองไฟฟ้าได้แล้ว UPS บางชนิดยังสามารถป้องกันความผิดปกติในระบบไฟฟ้าได้หลายประการ ซึ่งทั้งนี้ขึ้นอยู่กับคุณภาพ ราคา และการเลือกใช้ UPS ขึ้นอยู่กับผู้ใช้ที่จะเลือกใช้ UPS ตามความเหมาะสม และจุดประสงค์ ของการใช้งาน

5.2.1.1 ความผิดปกติในระบบไฟฟ้า

1. ระดับแรงดันไฟฟ้าต่ำกว่าปกติ; Sags.

เป็นความผิดปกติของระดับแรงดันต่ำกว่า 80% ของแรงดันไฟฟ้าปกติเป็นเวลา 20 มิลลิวินาที ถึง 2 วินาที สาเหตุจากปิดสวิตซ์เครือข่าย, ฟ้าผ่า, การลัดวงจรลงดิน การสตาร์ทมอเตอร์หรือเครื่องจักรกลต่างๆ

2. ระดับแรงดันไฟฟ้าเกินกว่าปกติ; Surges (Swells)

เป็นความผิดปกติของระดับแรงดันไฟฟ้าสูงกว่าปกติ คือ สูงกว่า 110% ของแรงดันไฟฟ้าปกติเป็นเวลา 20 มิลลิวินาที ถึง 2 วินาที สาเหตุเกิดจากเปิดสวิตซ์เครือข่ายม การลดขนาดของโหลดอย่างรวดเร็ว

3. ทรานเซี้ยนท์; Transient

เป็นความผิดปกติของแรงดันไฟฟ้า โดยแบ่งเป็น 2 ชนิดคือ

- 3.1 Impulse Transient มีขนาดของแรงดันไฟฟ้าเป็น 2 เท่าของแรงดันไฟฟ้าปกติช่วงเวลา 0.5 ถึง 20 มิลลิวินาที
- 3.2 Oscillatory Transient มีขนาดของแรงดันไฟฟ้า 50 โวลต์ ถึง 60 กิโลโวลต์ ในช่วงเวลา 0.5 ไมโครวินาที ถึง 2 มิลลิวินาที มีสาเหตุมาจากคอมพิวเตอร์ การเปิดหรือปิดสวิชต์เครือข่าย การลัดวงจร และฟ้าผ่า

4. Waveform distortion

เป็นความผิดปกติที่เกิดจากการผิดเพี้ยนสัญญาณของแรงดันไฟฟ้า อันเนื่องมาจากฮาร์มอนิก ซึ่งเกิดจากอุปกรณ์ไฟฟ้าบางชนิดสร้างฮาร์มอนิกเข้ามารบกวนในระบบไฟฟ้า ซึ่งฮาร์-มอนิกทำให้เกิดปัญหาหลายประการต่อระบบไฟฟ้า ดังนี้

- แรงดันไฟฟ้าผิดเพี้ยน
- กระแสไฟฟ้าเพี้ยน
- ความสูญเสียกำลังไฟฟ้าเพิ่มขึ้น
- เกิดการใช้งานเกินสายนิวทรัลและคาปาซิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เกิดการรบกวนโทรศัพท์และระบบสื่อสาร
- แรงดันไฟฟ้าระหว่างนิวทรัลกับดินสูงจน ไฟฟ้าทำงานผิดพลาดโทรศัพท์และระบบสื่อสาร
- แรงดันไฟฟ้าระหว่างนิวทรัลกับดินสูง จนไฟฟ้าทำงานผิดพลาด
- เกิดความร้อนสูงในหม้อแปลงและมอเตอร์ ทำให้อายุการใช้งานลดลง
- ก่อกวนให้เซอร์กิตเบรกเกอร์ทริป ทำให้หยุดชะงัก

5. Single Phase Outage

เป็นความผิดปกติของแรงดันไฟฟ้า ซึ่งลดลงต่ำกว่า 80% ของแรงดันไฟฟ้าปกติ เป็นเวลามากกว่า 2 วินาที มีสาเหตุมาจากฟ้าผ่า ไฟฟ้าลัดวงจร การอาร์คและฟิวส์ขาด

6. ไฟฟ้าดับสนิท Outage

เป็นการดับสนิทของแรงดันไฟฟ้าจนเป็นศูนย์โวลท์ เป็นเวลามากกว่า 10 มิลลิวินาทีมีสาเหตุมาจากฟ้าผ่า ไฟฟ้าลัดวงจร การอาร์ก

7. การรบกวนจาก RFI/EMI Noise

เป็นการรบกวนจากการแทรกสอดของคลื่นวิทยุ RFI ; Radio; Frequency Interference และการรบกวนจากสนามแม่เหล็กไฟฟ้า EMI;Electro-Magnetic Interference

5.2.1.2 โอกาสของความผิดปกติของระบบไฟฟ้า

โอกาสความผิดปกติสามารถแบ่งออกตามปัญหาที่พบได้ดังตาราง

ตารางที่ 5.1 โอกาสความผิดพลาดของระบบไฟฟ้า

ลำดับที่	ปัญหาที่พบ	โอกาสที่เกิด
1	คลื่นรบกวน,สไปก์	80%
2	แรงดันไฟฟ้าตก, แรงดันไฟฟ้าเกิน	12%
3	ความถี่ไฟฟ้าปกติ, ไฟฟ้าขัดข้องช่วงสั้นมาก, ไฟฟ้าขัดข้องช่วงนาน	1%
รวม		100%

หมายเหตุ : อุปกรณ์ป้องกันในแต่ละกลุ่มจะแตกต่างกัน

1. วิธีการควบคุมคุณภาพไฟฟ้าได้แก่ กรองคลื่นรบกวน ปรับแรงดันไฟฟ้า ปรับความถี่ไฟฟ้าสร้างรูปคลื่นไฟฟ้าใหม่ จ่ายกำลังไฟฟ้าต่อเนื่อง

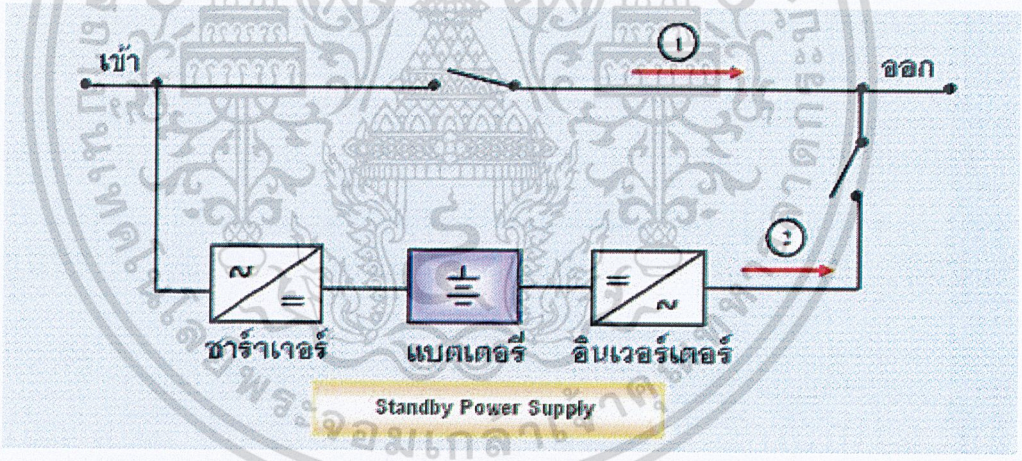
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วงจรกรองไฟฟ้า (Filters)
3. หม้อแปลงแยกขดลวด (Isolation Transformer)
4. ชุดกับดักเสิร์จ (Surge Arrester)
5. ชุดควบคุมฮาร์มอนิกแบบแอกทีฟ (Active Harmonic)
6. เครื่องควบคุมค่าแรงดันไฟฟ้า (Voltage Regulators)
7. สายป้อนทดแทน (Redundant Power Feeder)
8. ชุดมอเตอร์-เครื่องกำเนิดไฟฟ้า (Motor-Generator)
9. เครื่องยูพีเอส (Uninterruptible Power Supply)

5.2.1.3 การควบคุมคุณภาพไฟฟ้าโดยใช้ UPS

UPS สามารถควบคุมคุณภาพและแก้ปัญหาการผิดปกติในระบบไฟฟ้าได้หลายประการ ซึ่งทั้งนี้ขึ้นอยู่กับชนิดและคุณภาพของ UPS ซึ่งสามารถแบ่งชนิดของ UPS ได้ 3 ชนิดดังนี้

1. Standby Power Supply



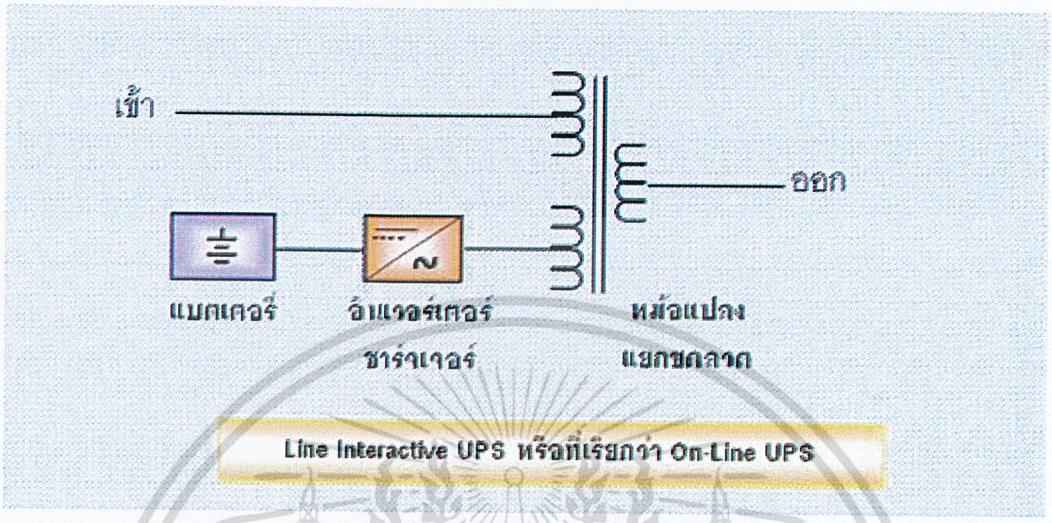
ภาพที่ 5.1 Standby Power Supply

เป็นเครื่องจ่ายกำลังไฟฟ้าสำรองหรืออาจเรียกว่า Off-Line UPS ซึ่งเราเรียกว่า Standby power supply เพราะ UPS นี้จะไม่จ่ายกำลังไฟฟ้าให้กับโหลดอย่างทันที หลังจากไฟดับมีการขาดตอนของกำลังไฟฟ้าอยู่ช่วงเวลาหนึ่ง ดังรูป ไฟฟ้าปกติ กำลังไฟฟ้าจะถูกส่งผ่านไปที่โหลดโดยตรง ไม่มีการป้องกันความปกติของไฟฟ้า เช่น สัญญาณรบกวน EMI ไฟตกไฟเกิน การลดทอนฮาร์มอนิกและในกรณีที่ไฟฟ้าดับ เครื่องจะทำการสร้างกำลังไฟฟ้าสำรองขึ้นมาใหม่ จากภาคอินเวอเตอร์ซึ่งรับแรงดันไฟฟ้าจากแบตเตอรี่สำรอง เนื่องจาก UPS ชนิดนี้มีส่วนประกอบไม่มากจึงทำให้มีราคาถูกกว่าชนิดอื่น ซึ่ง UPS ชนิดนี้จะไม่เหมาะกับอุปกรณ์ไฟฟ้าที่ต้องการความรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และต่อเนื่องโดยไม่สะดุดของกำลังไฟฟ้า และไม่ต้องการประสิทธิภาพในการความผิดปกติของระบบไฟฟ้ามากนัก

2. On - line Protection UPS หรือ Line Interactive UPS

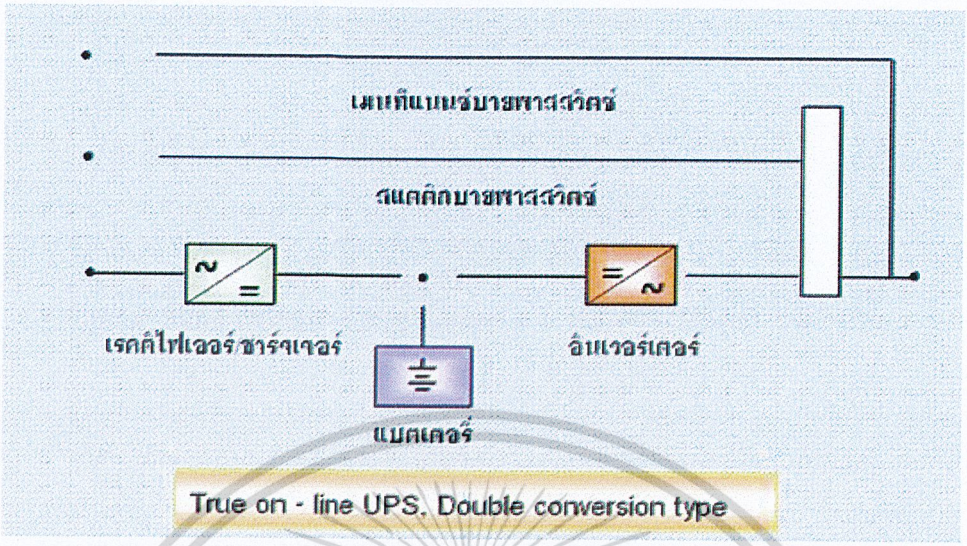


ภาพที่ 5.2 One line Protection UPS หรือ Line Interactive UPS

เป็นเครื่อง UPS แบบ Static ชนิดหนึ่ง บางครั้งอาจเรียกว่า single conversion UPS ดังภาพ ในสภาวะปกติกำลังไฟฟ้าจะถูกส่ง ถูกผ่านเข้าหม้อแปลงและต่อไปยังโหลดโดยตรง บางรุ่นอาจติดตั้งชุดรักษาระดับแรงดันไฟฟ้า Line conditioner เพื่อรักษาระดับแรงดันไฟฟ้าให้คงที่ตลอด อันเนื่องจากปัญหาไฟตกไฟเกิน แต่ยังไม่สามารถป้องกันปัญหาความผิดปกติในระบบไฟฟ้า อาทิ สัญญาณรบกวน (noise) การเกิดฮาร์มอนิก และไฟกระชากได้ในสภาวะปกติกำลังไฟฟ้าจากหม้อแปลงจะทำการแปลงแรงดันไฟกระแสสลับเป็นกระแสตรงที่ภาคคอนเวอร์เตอร์ แล้วผ่านชาร์จเจอร์เพื่อชาร์จประจุเก็บไว้ในแบตเตอรี่ในสภาวะที่ไฟดับ ที่ภาคคอนเวอร์เตอร์จะทำหน้าที่เป็นอินเวอร์เตอร์ เพื่อแปลงแรงดันไฟฟ้ากระแสตรงเป็นกระแสสลับให้กับหม้อแปลงจ่ายกำลังไฟฟ้าให้โหลดต่อไป UPS ชนิดนี้สามารถสำรองกำลังไฟฟ้าและรักษาระดับแรงดันไฟที่ได้ แต่ยังไม่เหมาะกับอุปกรณ์เครื่องมือที่มีความไวต่อการผิดปกติของระบบไฟฟ้ากำลังสูง เช่น เครื่อง PC Network server อุปกรณ์ เครื่องมือแพทย์ เครื่องมือสื่อสารโทรคมนาคม เครื่องจักรในโรงงานอุตสาหกรรมการผลิตหรือระบบที่ต้องการความมั่นใจในการสำรองไฟสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. True on - line UPS, Double conversion type



ภาพที่ 5.3 True on - line UPS, Double conversion type

เครื่อง UPS ชนิด Double conversion นี้เป็นยูพีเอสที่มีประสิทธิภาพในการสำรองกำลังไฟฟ้าและป้องกันความผิดปกติของระบบไฟฟ้ามากที่สุด ดังภาพที่ 5.3 กำลังไฟฟ้าที่ป้อนให้กับ UPS จะถูกแปลงแรงดันไฟฟ้ากระสลับเป็นแรงดันไฟฟ้ากระแสตรงที่ภาคคอนเวอร์เตอร์ ส่วนหนึ่งจะถูกชาร์จประจุเก็บเข้าไปเก็บไว้ที่แบตเตอรี่ อีกส่วนจะถูกส่งผ่านเข้าภาคอินเวอร์เตอร์เพื่อแปลงแรงดันไฟกระแสตรง กลับไปเป็นกระแสสลับอีกครั้งสู่โหลดผู้ใช้ต่อไป ซึ่งการแปลงแรงดันไฟทั้งสองครั้งนี้จึงเรียกว่า Double conversion ซึ่งการแปลงแรงดันครั้งนี้ ทำให้แรงดันที่ถูกสร้างมีรูปคลื่น (wave form) ของแรงดันไฟฟ้าเป็นคลื่น sin ที่เที่ยงตรง (pure sine wave) ป้องกันการปิดเพี้ยนของแรงดันไฟฟ้า อันเนื่องมาจากฮาร์มอนิก รวมทั้งความถี่ของแรงดันไฟฟ้าจะเที่ยงตรงที่ 50 เฮิร์ซ และที่ภาคกรองและป้องกันสัญญาณรบกวนก็จะป้องกันสัญญาณรบกวนทางสนามแม่เหล็ก (EMI) ป้องกันไฟกระชาก (spike) หรือจากฟ้าฝ่า (lightning) ได้อย่างสมบูรณ์ โดยรวมแล้ว True On-Line UPS สามารถป้องกันความผิดปกติของระบบไฟฟ้าได้หลายสาเหตุ อาทิ ไฟตก (Dip) ไฟเกิน (surge) ไฟกระชาก (transient) ฮาร์มอนิก และไฟฟ้าหยุดชะงัก (outage) True on-Line UPS สามารถสำรองไฟฟ้าทันทีเมื่อไฟดับโดยไม่มีการขาดตอน 0 มิลลิวินาที (Zero transfer time) จึงเหมาะสำหรับ เครื่องหรืออุปกรณ์ไฟฟ้าที่ต้องการความเที่ยงตรงและมีความไว (Sensitivity) ต่อการเปลี่ยนแปลงของระบบไฟฟ้า อาทิ network server PC อุปกรณ์เครื่องมือสื่อสาร โทรคมนาคม เครื่องจักรในโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 การขยาย port I/O

ในกรณีที่ต้องการควบคุมตู้ล็อกเกอร์นับจำนวนมากปริมาน I/O ของคอนโทรลเลอร์ไม่พอต่อความต้องการและจะเป็นการสิ้นเปลืองหาก ต้องซื้อคอนโทรลเลอร์เพิ่มอีกหนึ่งตัว แนวทางแก้ไข คือ ใช้ IC เบอร์ PCF 8574 ทำให้เราสามารถเพิ่ม I/O ได้เพิ่มมากขึ้น

5.3 แนวทางในการพัฒนา

1. ในกรณีที่ต้องการตรวจสอบว่า ในล็อคเกอร์มีของอยู่หรือไม่ สามารถนำเซนเซอร์มาติดตั้งเพื่อตรวจสอบสถานะแล้วแสดงสถานะทางอินเตอร์เน็ตไปที่ Web Baseได้

2. ในกรณีที่ต้องการความปลอดภัยในการใช้งานตู้ล็อกเกอร์ที่มากขึ้น โดยเราสามารถเขียนโปรแกรมตรวจสอบและจำกัดผู้ที่ผู้สามารถใช้ล็อคเกอร์ได้ รวมไปถึงยังสามารถนำไปใช้กับระบบการชำระเงิน เพื่อให้สิทธิกับผู้ที่เป็นสมาชิกสามารถใช้ล็อกเกอร์ได้อย่างปลอดภัย

3. ในกรณีที่ต้องการไมโครคอนโทรลเลอร์ที่ประมวลที่รวดเร็วขึ้น รวมถึงพื้นที่ในการจัดเก็บที่มากขึ้น และแอพพลิเคชั่นที่มากขึ้น สามารถเปลี่ยนไปใช้ไมโครคอนโทรลเลอร์ตัวอื่นได้ เช่น ARM AVR PIC

4. ในกรณีที่ต้องการความสะดวกสบายในการเข้าใช้งานตู้ล็อกเกอร์ของอุปกรณ์แต่ละชนิด เช่น iPad, Samsung Galaxy Tab หรืออุปกรณ์อื่นๆ สามารถเขียนแอพพลิเคชั่นเฉพาะ ที่สามารถใช้งานบนอุปกรณ์เหล่านั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] นายเอกชัย มะการ. เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino. กรุงเทพมหานคร : 2552.

[2] ประภาพร ช่างไม้. มือใหม่ สร้างเว็บไซต์ Photoshop + Dreamweaver. พิมพ์ครั้งที่ 1. กุมภาพันธ์ : 2553.

[3] บัญชา ปะสีละเตสัง. พัฒนาเว็บแอพพลิเคชั่นด้วย PHP ร่วมกับ MySQL และ Dreamweaver . พิมพ์ครั้งที่ 1 . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น . 2553.

[4] ประภาพร ช่างไม้. คู่มือการเขียนโปรแกรมภาษา C ฉบับผู้เริ่มต้น . พิมพ์ครั้งที่ 1 กรุงเทพมหานคร : ดีอีวีบุ๊ค . 2545.

[5] อนรรฆนงค์ คุณมณี. basic of PHP . กรุงเทพมหานคร : ดีอีวีบุ๊ค . 2550 .

[6] มณีโชติ สมานไทย. ภาษา HTML ฉบับผู้เริ่มต้น . พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : ดีอีวีบุ๊ค . 2545 .

[7] บัณฑิต จามรภูติ . คัมภีร์ Ubuntu Linux Server เล่ม 1 . กรุงเทพมหานคร : 2543

[8] บัณฑิต จามรภูติ . คัมภีร์ Ubuntu Linux Server เล่ม 2 . กรุงเทพมหานคร : 2543

[9] อรพิน ประวัติบริสุทธิ์ . คู่มือเรียนภาษาซี .กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น จำกัด . 2545 .

[10] นิรุธ อำนวยศิลป์ . คู่มือเขียนโปรแกรมด้วยภาษาซี . กรุงเทพมหานคร : ดวงกมลสมัย . 2544

[11] พร้อมเลิศ หล่อวิจิตร . คู่มือเรียน PHP และ MySQL สำหรับผู้เริ่มต้น . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น จำกัด . 2545 .

[12] จีระสิทธิ์ อึ้งรัตนวงศ์ . Q&A 108 สูตรสำเร็จ PHP . พิมพ์ครั้งที่ 2 . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น . 2545 .

[13] นวอร แจ่มขำ . สร้างเว็บไซต์ด้วย Dreamweaver . พิมพ์ครั้งที่ 1 . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น . 2541 .

[14] กำพล ลีลาภรณ์ . Flash Actionscript . พิมพ์ครั้งที่ 1 . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น . 2546 .

[15] สมศักดิ์ โชคชัยชุติกุล . Insight PHP ฉบับสมบูรณ์ . พิมพ์ครั้งที่ 7 . กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น . 2540 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

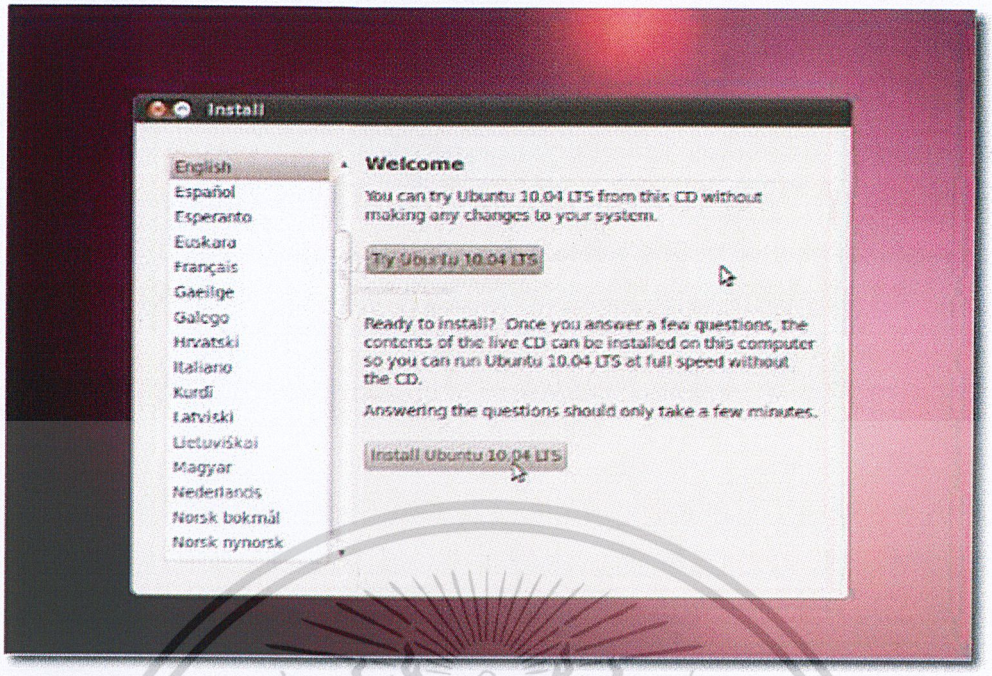
1. การติดตั้ง Ubuntu Server

Ubuntu 10.04, หรือ Lucid Lynx, ระบบปฏิบัติการอีกตัวของ open source อย่างลีนุก เมื่อเป็นรุ่นที่ 12 ของ Ubuntu OS การติดตั้งสำหรับผู้ใช้งานคอมพิวเตอร์ ที่สนใจระบบปฏิบัติการฟรีอย่าง Ubuntu 10.04 LTS พิเศษสำหรับมือใหม่ที่ไม่เคยติดตั้งและใช้งานความต้องการของระบบ เราจะต้องมีแผ่น โปรแกรม Ubuntu 10.04 LTS Desktop ISO image สามารถดาวโหลดได้จากที่นี้ครับ โหลด Ubuntu 10.04 LTS หลังดาวโหลดมาแล้ว จะต้องใช้โปรแกรมเขียนแผ่น CD/DVD burning เช่น (Nero, CDBurnerXP, Roxio) ด้วยแผ่น CD เปล่าที่ความเร็ว 8x จาก นั้นใส่แผ่นเข้าไปในเครื่องคอมพิวเตอร์ที่ต้องการติดตั้งและ reboot เครื่องคอมพิวเตอร์ โดยกำหนดให้ boot จาก CD กดแป้น F8, F11 or F12 (ขึ้นอยู่กับ BIOS) เพื่อเลือก CD/DVD-ROM ให้เป็น boot device.คอย การโหลดของ CD เพื่อ

ภาพที่ 1 หน้าจอของ Ubuntu

เราจะพบหน้าจอติดตั้งเป็นคำแนะนำแบบ wizard. เลือกภาษาที่เราต้องการ จากนั้นคลิก "Install Ubuntu 10.04 LTS" ตามภาพที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



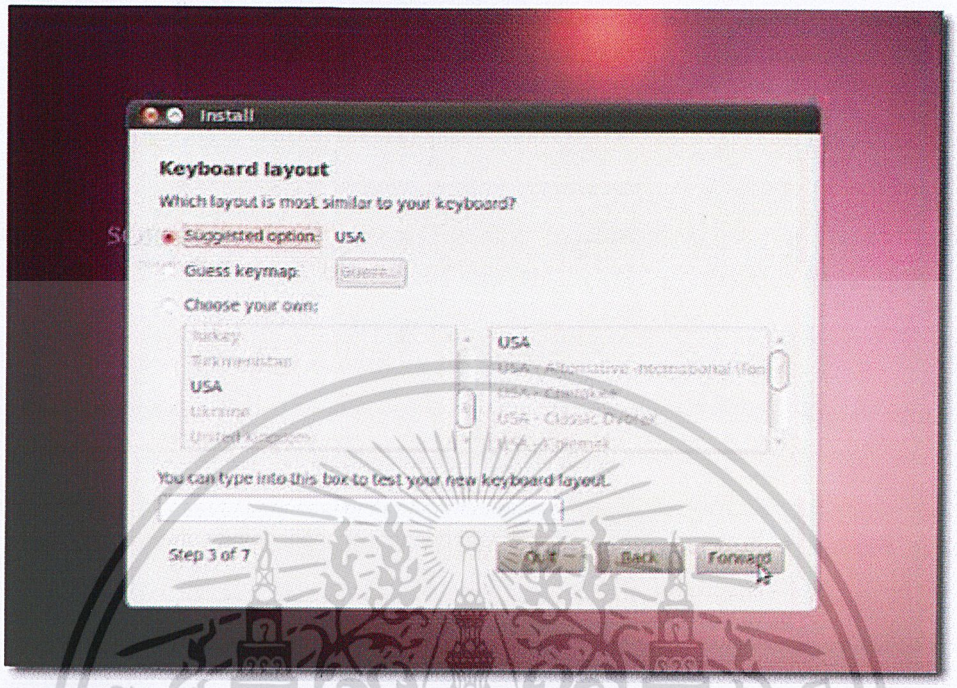
ภาพที่ 2 หน้าจอสำหรับการติดตั้ง

เราอยู่ที่ไหน? หน้าต่างที่สองจะแสดงออกมาเป็นแผนที่โลก ขึ้นอยู่กับการเลือกสถานที่อยู่ของเรา เขตเวลาจะปรับให้อัตโนมัติ โดยสามารถเลือกจากรายการ drop-down list ด้านล่างของหน้าต่าง คลิกปุ่ม Forword" ตามภาพที่ 3

ภาพที่ 3 การเลือกสถานที่อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบ keyboard ในหน้าต่างที่สาม ,เราจะสามารถเลือกรูปแบบของ keyboard layout. ปกติค่า default จะสามารถใช้งานได้อยู่แล้ว คลิกปุ่ม “Forward” เมื่อกำหนดเรียบร้อย



ภาพที่ 4 เลือกรูปแบบ keyboard layout

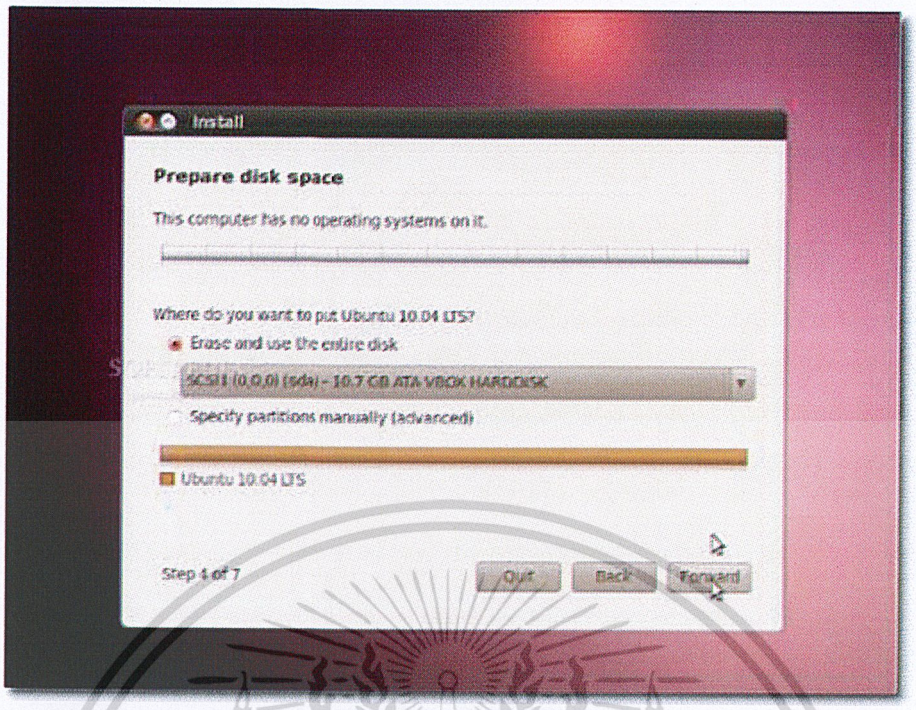
การแบ่งพื้นที่ของ Hard diskเรามี 4 ทางเลือกในการทำ :

1. ถ้าเรามีอีกหนึ่ง OS เช่น Windows XP และคุณต้องการติดตั้งแบบสองระบบ เลือก option แรก : “ลงคู่กัน และเราจะเลือกว่าจะใช้ตัวไหน ตอนเปิดเครื่อง

Note: สำหรับ option นี้จะแสดงก็ต่อเมื่อเรามี OS อย่างเช่น Windows ติดตั้งอยู่แล้วเท่านั้น จำไว้ว่าหากติดตั้งแล้ว ระบบ Windows boot loader จะถูกเขียนทับด้วย Ubuntu boot loader

2. ถ้าเราลบหรือ Format ระบบ OS เดิม ออกไป หรือ hard drive ของเราว่าง ระบบก็จะให้เราปรับ automatically partition ให้เราเลือก “Use the entire disk.”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5 การแบ่งพื้นที่ hard disk

Note: สำหรับ option นี้แนะนำสำหรับผู้ใช้งานส่วนใหญ่ ซึ่งไม่มี OS อื่นๆ ติดตั้งอยู่ หรือ ลบระบบเดิมออก เช่น Windows OS.

3. สำหรับทางเลือก คือการติดตั้ง บน "Use the largest continuous free space" ซึ่งจะติดตั้ง Ubuntu 9.10 เบนพาร์ทิชั่นที่ไม่มีการ กำหนด ของ hard drive.

4. ทางเลือกที่ 4 คือ "ระบุ partitions manually" และแนะนำสำหรับผู้ใช้งานขั้น advanced users, ในการสร้าง special partitions หรือ format hard drive ด้วยระบบอื่นๆ แต่มีประโยชน์ในการสร้าง /home partition, ซึ่งมีประโยชน์ในกรณีเราต้องการ reinstall ทั้งระบบ ต่อไปนี้เป็นขั้นตอนการทำ manual partitioning with /home:

- เลือก "Specify partitions manually (advanced) จากนั้นคลิก "Forward" button;

- แน่ใจว่าเลือก hard drive ที่ถูกต้อง /dev/sda จะเป็นตัวแรกของ hard drive. /dev/sdb จะเป็นตัวที่สอง แน่ใจว่าเราต้องรู้ว่าต้องการ format! อันไหน มิฉะนั้นข้อมูลจะหายไปหมด ;

- สมมุติว่า ตัวที่สองว่าง ไม่มีข้อมูล หรือระบบ OS แต่มีการแบ่งพาร์ทิชั่น ให้คลิกเลือกแต่ละตัวของพาร์ทิชั่น และคลิกปุ่ม "Delete" button. หลังจากนั้นจะแสดง "free space." ทำต่อไปกับพาร์ทิชั่นที่เหลือ จนกระทั่งเราได้ "free space" line;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ด้วย “free space” line ถูกเลือก, คลิกปุ่ม “Add” button. ในหน้าต่างใหม่ พิมพ์ 2000 ใน “New partition size in megabytes” เลือก “swap area” จากช่อง “Use as:” drop down list. คลิก OK button and, เราจะเห็น “swap” line พร้อมขนาดที่เราระบุ;

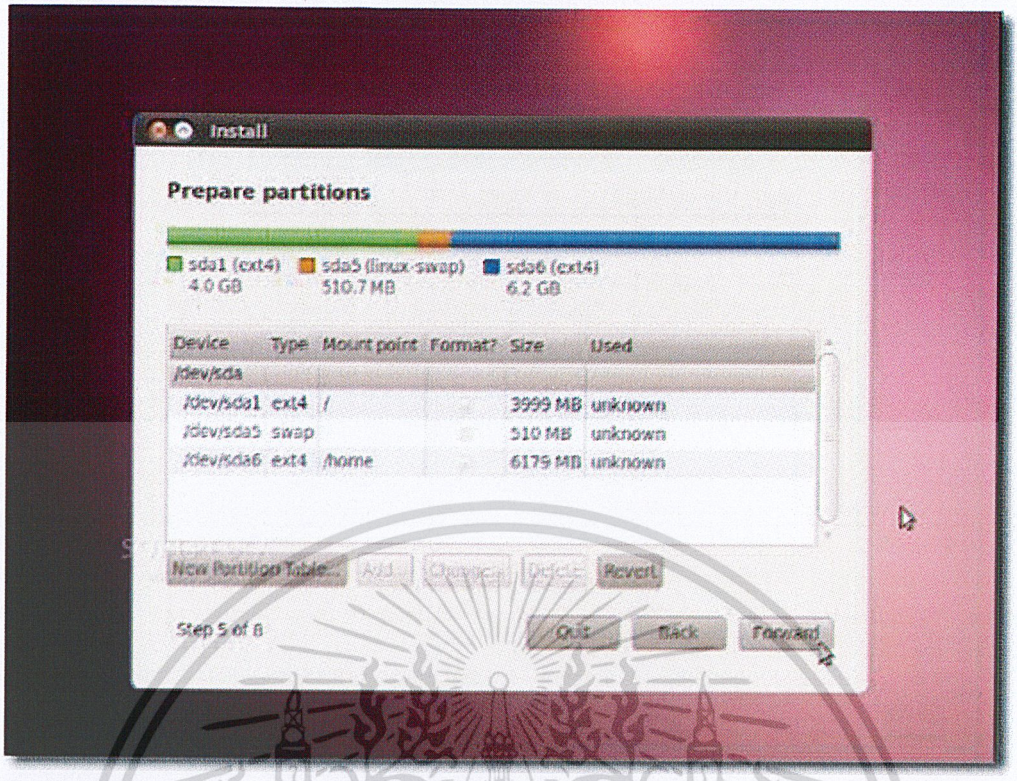
- ด้วย “free space” line ถูกเลือก, คลิก “Add” button. ในหน้าต่างใหม่ , เลือก “Primary” option, พิพม์ค่าระหว่าง 10,000 และ 50,000 ในช่อง “New partition size in megabytes” เลือก / as the “Mount point.” คลิก OK buttonเราจะเห็น “ext4 /” line พร้อมขนาด;

- ด้วย “free space” line ถูกเลือก , คลิก “Add” button. ในหน้าต่างใหม่ , เลือก “Primary” option, ใส่ค่าระหว่าง 30,000 และ 50,000 (หรือเท่าขนาดที่เราเหลืออยู่) ในช่อง “New partition size in megabytes” field และเลือก /home เป็น “Mount point.” คลิก OK button และเราจะเห็น “ext4 /home” line พร้อมขนาดที่เราระบุ.

ภาพที่ 6 การสร้าง Partitions

เราควรจะได้ตารางพาร์ทิชั่นหน้าตาประมาณนี้ ครับ คลิก “Forward” button เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

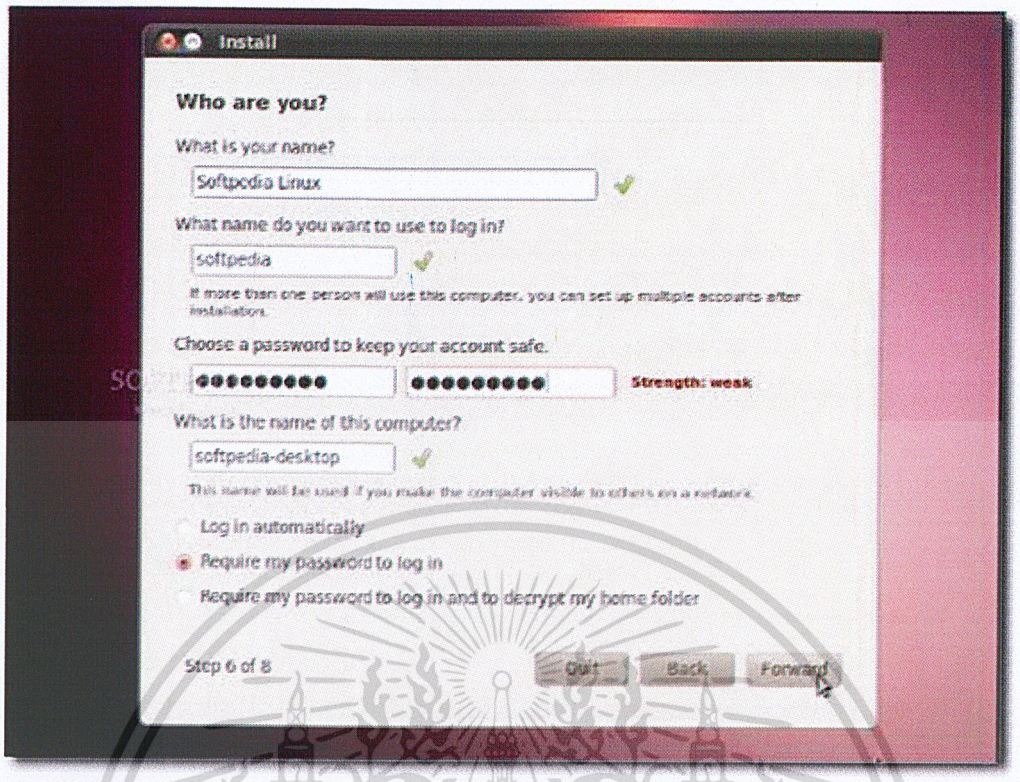


ภาพที่ 7 ตาราง Partitions

WARNING: แจ้งเตือนว่าข้อมูลบน drive หรือ partition จะถูกลบหรือหายไป คลิก “Forward” button เพื่อเข้าสู่ขั้นตอนต่อไป …

คุณคือใคร ? ในหน้าต่างนี้ , เราต้องทำตามที่การติดตั้งบอก . กรอกข้อมูลจริงของเราลงไป ชื่อที่เราต้องการใช้งานในการ log in บน Ubuntu OS (หรือ “username,” ซึ่งจำเป็นต้องใช้ในการ log in เข้าระบบ , กำหนด password และชื่อของ computer (จะตั้งให้เราโดยออโต้ แต่เราสามารถเปลี่ยนใหม่ได้).ใน ขั้นตอนนี้, จะมี option called “Log in automatically.” ถ้าเราเลือก option นี้, เราจะ automatically logged in เข้าสู่ระบบ Ubuntu desktop. คลิก “Forward” buttonเพื่อไปขั้นต่อไป

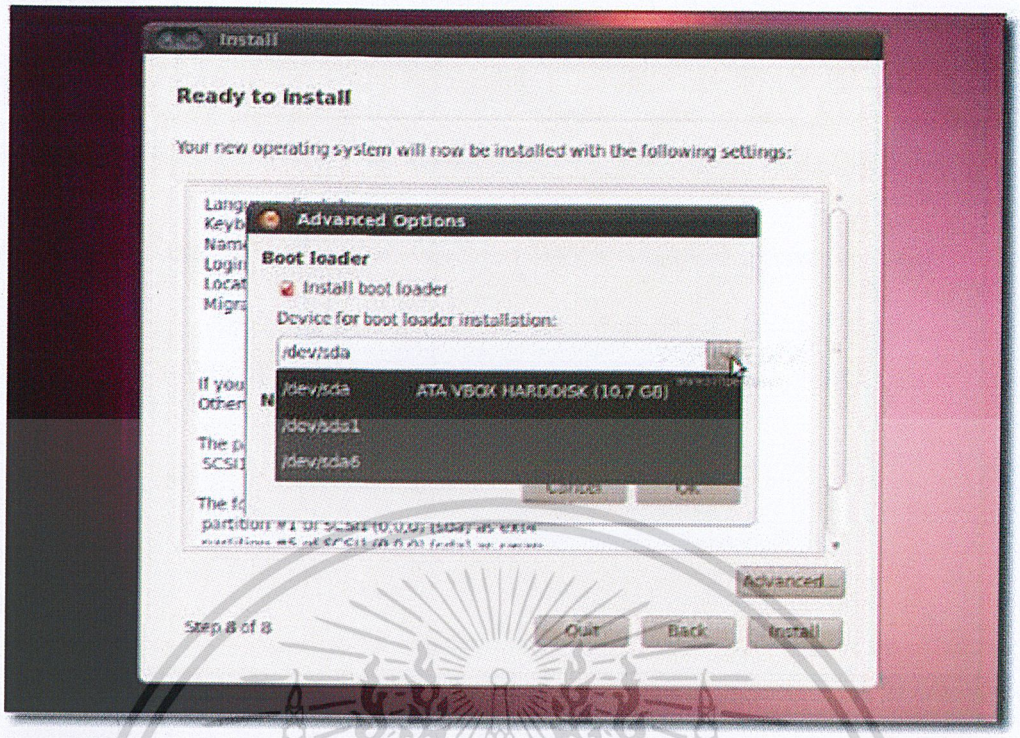
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 8 Log In เพื่อยืนยันตัวบุคคล

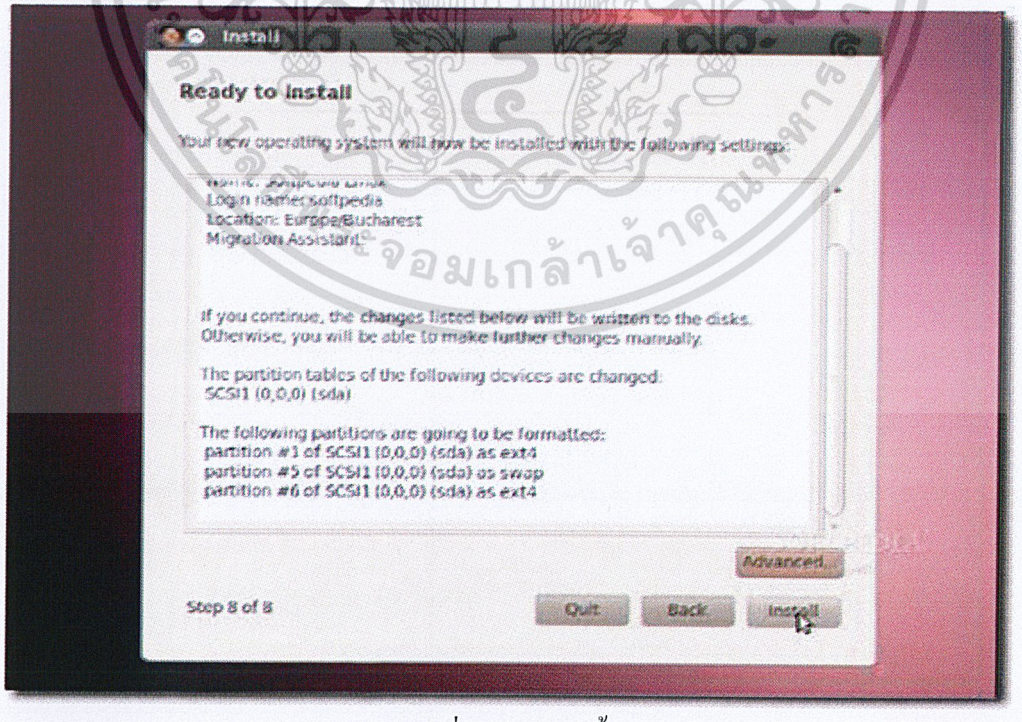
คุณพร้อมจริงๆ สำหรับ Ubuntu เป็น ขั้นตอนสุดท้ายของ ของการติดตั้ง installation. เราสามารถเลือกที่จะติดตั้ง boot loader บน partition อื่นๆ หรือบน hard drive นอกเหนือจากค่า default one, แต่แนะนำสำหรับผู้ใช้งานขั้น advanced users. ถ้าใครติดตั้ง ลง USB memory stick, ราวกับว่ามันเป็น USB hard drive, จะทำให้ตัว installer สับสนกับ computer's hard disk drive MBR. ดังนั้น, คลิก "Advanced" button และเลือกตัวที่ถูก (the USB stick in this case)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 9 การเลือกการติดตั้ง

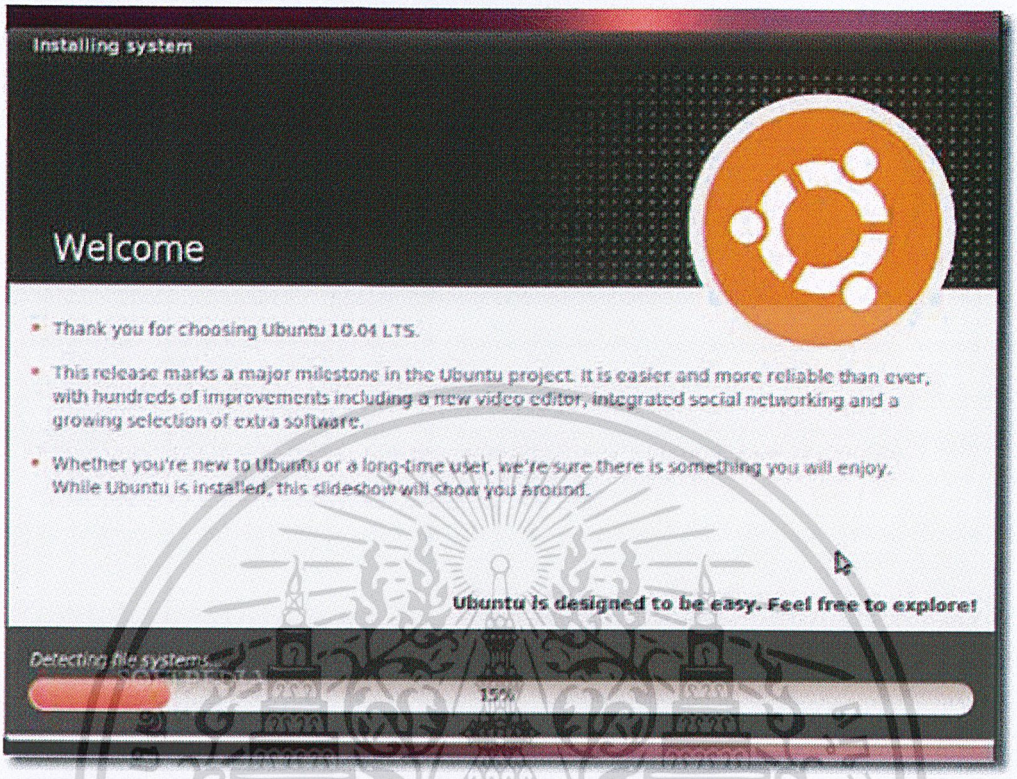
คลิก “Install” button เพื่อเริ่มการติดตั้ง



ภาพที่ 10 การติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

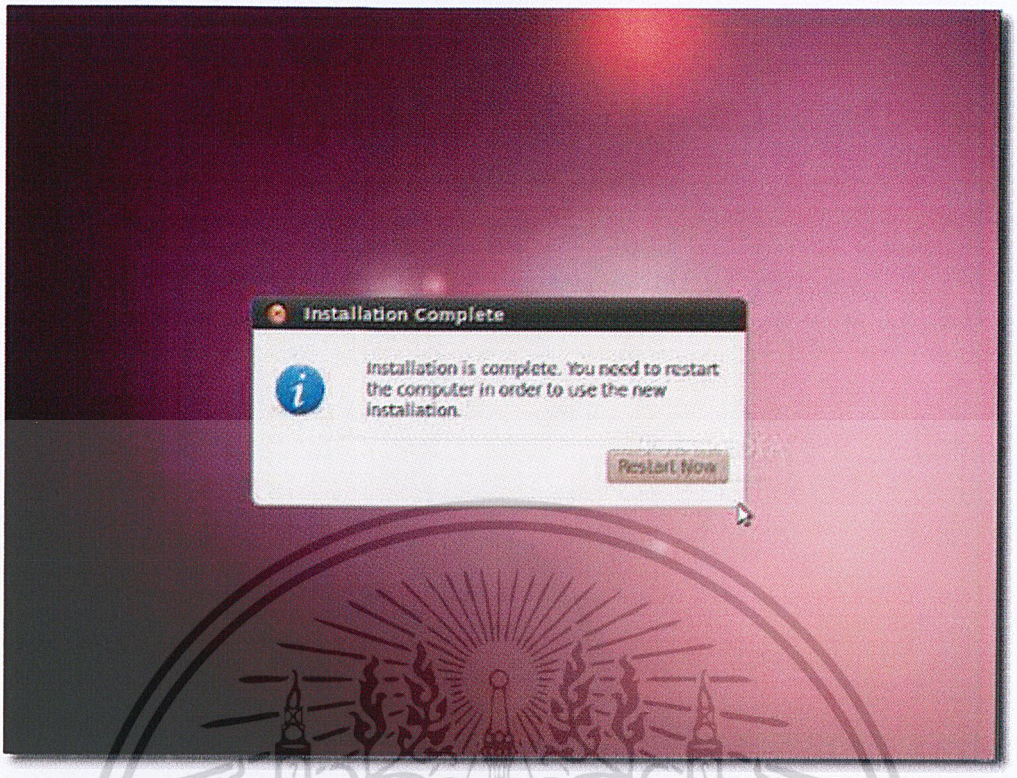
โปรแกรม Ubuntu 10.04 LTS (Lucid Lynx) operating system จะได้รับการติดตั้ง



ภาพที่ 11 โปรแกรมกำลังทำการติดตั้ง

ใช้เวลาประมาณ 10 – 18 นาที (ขึ้นกับ computer's specs), จะมี pop-up window ขึ้นมา, แจ้งว่าการติดตั้งเสร็จสิ้น และให้เรา restart เครื่อง computer เพื่อใช้งาน Ubuntu operating system.คลิก "Restart Now" button

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



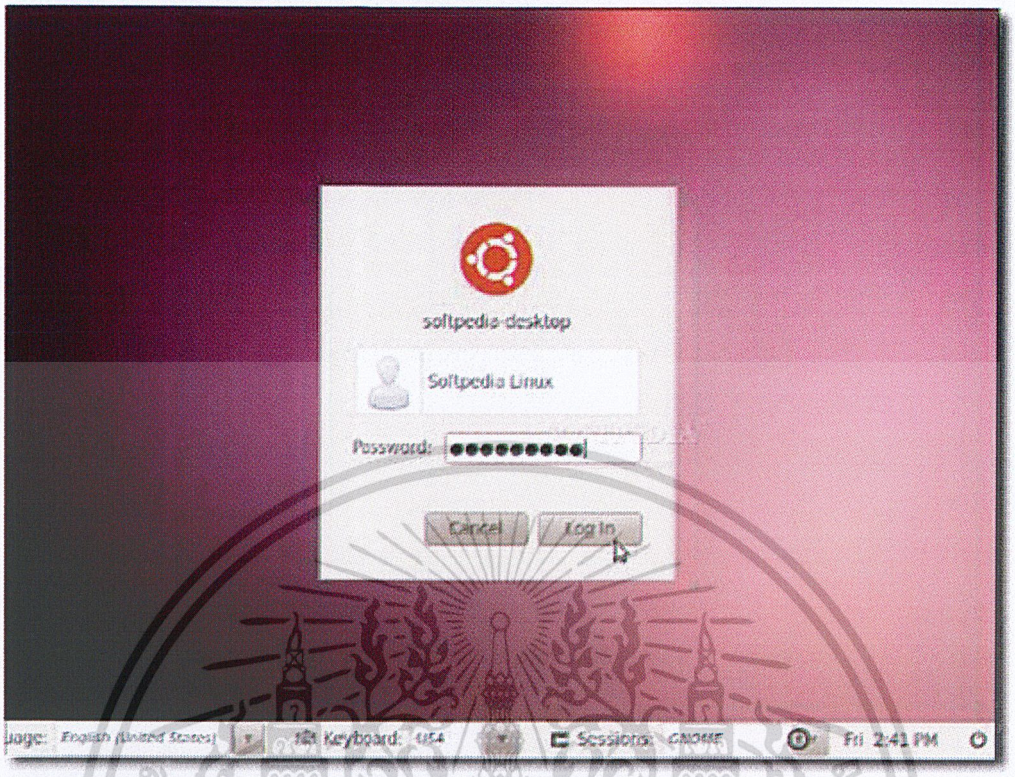
ภาพที่ 12 POP UP เตือนว่าติดตั้งโปรแกรมเรียบร้อย

แผ่น CD จะถูกขับออก จากนั้นกดแป้น “Enter” key เพื่อ reboot. เครื่อง computer จะถูก restarted and, และสักพักเราจะเห็นการ boot เข้าสู่ Ubuntu boot splash

ภาพที่ 13 การ Boot เข้าสู่ Ubuntu

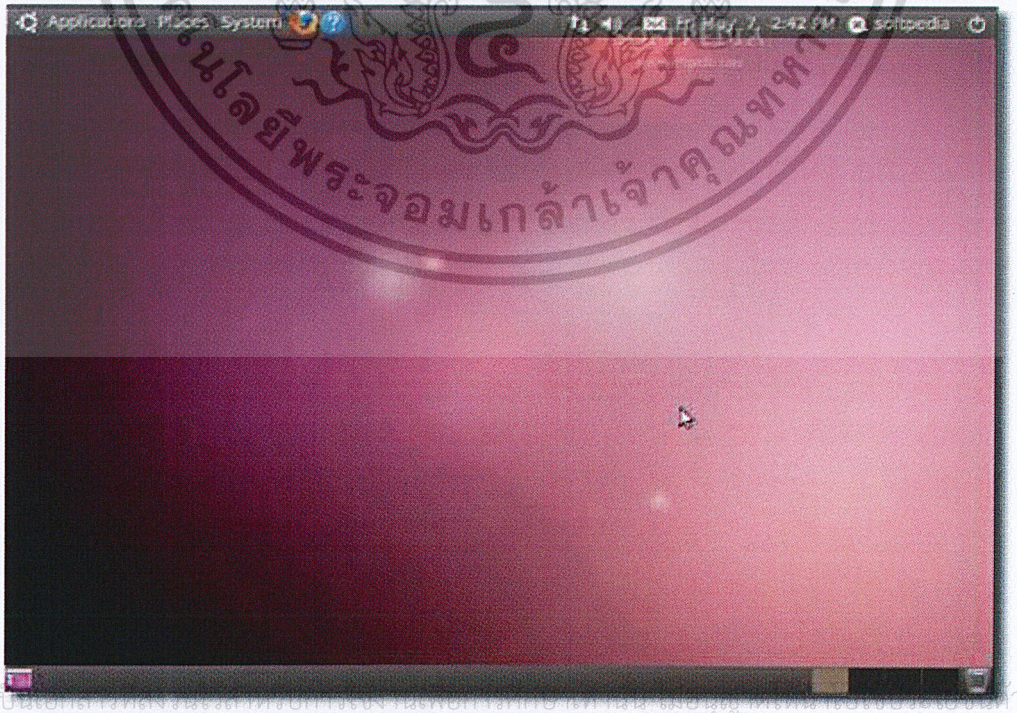
เอกสารนี้เ[illegible]านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัด[illegible]องเอกสารทุกครั้งที่มีการนำไปใช้

ที่หน้าต่าง login screen, คลิก username และใส่ password. คลิก “Log In” button or hit Enter



ภาพที่ 14 หน้า Log In เข้าสู่ Ubuntu

Have fun using Ubuntu 10.04 LTS! ขอให้ มีความสุขและสนุกกับระบบ Ubuntu 10.04 LTS



ภาพที่ 15 เข้าสู่การใช้งาน Ubuntu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรม Arduino

```
int opPin[] = {22,23,24,25,26,27,28,29};
int inPin1[] = {30,32,34,36,38,40};
int inPin2[] = {31,33,35,37,39,41};
int incomingByte;

void setup()
{
 Serial.begin(9600);
 pinMode(opPin[0],OUTPUT);
 pinMode(opPin[1],OUTPUT);
 pinMode(opPin[2],OUTPUT);
 pinMode(opPin[3],OUTPUT);
 pinMode(opPin[4],OUTPUT);
 pinMode(opPin[5],OUTPUT);
 pinMode(opPin[6],OUTPUT);
 pinMode(opPin[7],OUTPUT);
 pinMode(inPin1[0],INPUT);
 pinMode(inPin1[1],INPUT);
 pinMode(inPin1[2],INPUT);
 pinMode(inPin1[3],INPUT);
 pinMode(inPin1[4],INPUT);
 pinMode(inPin1[5],INPUT);
 pinMode(inPin2[0],INPUT);
 pinMode(inPin2[1],INPUT);
 pinMode(inPin2[2],INPUT);
 pinMode(inPin2[3],INPUT);
 pinMode(inPin2[4],INPUT);
 pinMode(inPin2[5],INPUT);
 digitalWrite(opPin[6],LOW);
 digitalWrite(opPin[7],LOW);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void loop()
{
 if(Serial.available() > 0)
 {
  incomingByte = Serial.read();
  if(incomingByte == '0')
  {
   digitalWrite(opPin[0], HIGH);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
    int lSwitch2 = lock2<<4;
    int lSwitch3 = lock3<<3;
    int lSwitch4 = lock4<<2;
    int lSwitch5 = lock5<<1;
    int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
   Serial.print(rCheck,BYTE);
  }
  else if((incomingByte ==
'1')&&((digitalRead(inPin1[0])==LOW)||(digitalRead(inPin2[0])==LOW)))
  {
   digitalWrite(opPin[0], LOW);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
    int lSwitch2 = lock2<<4;
    int lSwitch3 = lock3<<3;
    int lSwitch4 = lock4<<2;
    int lSwitch5 = lock5<<1;
    int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
    Serial.print(rCheck,BYTE);
}
else if(incomingByte == '2')
{
  digitalWrite(opPin[1], HIGH);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
    int lSwitch2 = lock2<<4;
    int lSwitch3 = lock3<<3;
    int lSwitch4 = lock4<<2;
    int lSwitch5 = lock5<<1;
    int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
  Serial.print(rCheck,BYTE);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else if((incomingByte ==
'3')&&((digitalRead(inPin1[1])==LOW)||(digitalRead(inPin2[1])==LOW)))
  {
   digitalWrite(opPin[1], LOW);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
    int lSwitch2 = lock2<<4;
    int lSwitch3 = lock3<<3;
    int lSwitch4 = lock4<<2;
    int lSwitch5 = lock5<<1;
    int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
    Serial.print(rCheck,BYTE);
  }
  else if(incomingByte == '4')
  {
   digitalWrite(opPin[2], HIGH);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      int lSwitch2 = lock2<<4;
      int lSwitch3 = lock3<<3;
      int lSwitch4 = lock4<<2;
      int lSwitch5 = lock5<<1;
      int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
     Serial.print(rCheck,BYTE);
   }
   else if((incomingByte ==
'5')&&((digitalRead(inPin1[2])==LOW)||(digitalRead(inPin2[2])==LOW)))
   {
    digitalWrite(opPin[2], LOW);
      int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
      int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
      int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
      int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
      int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
      int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
      int lock0 = 1;
      int lSwitch0 = lock0<<6;
      int lSwitch1 = lock1<<5;
      int lSwitch2 = lock2<<4;
      int lSwitch3 = lock3<<3;
      int lSwitch4 = lock4<<2;
      int lSwitch5 = lock5<<1;
      int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
      Serial.print(rCheck,BYTE);
   }
   else if(incomingByte == '6')
   {
    digitalWrite(opPin[3], HIGH);
      int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
      int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
```

เอกสารนี้เป็นกรรมสิทธิ์ของมหาวิทยาลัยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ทำซ้ำดัดแปลง ก่อนได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
      int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
      int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
      int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
      int lock0 = 1;
      int lSwitch0 = lock0<<6;
      int lSwitch1 = lock1<<5;
      int lSwitch2 = lock2<<4;
      int lSwitch3 = lock3<<3;
      int lSwitch4 = lock4<<2;
      int lSwitch5 = lock5<<1;
      int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
     Serial.print(rCheck,BYTE);
   }
   else if((incomingByte ==
'7')&&((digitalRead(inPin1[3])==LOW)||(digitalRead(inPin2[3])==LOW)))
   {
    digitalWrite(opPin[3], LOW);
      int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
      int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
      int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
      int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
      int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
      int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
      int lock0 = 1;
      int lSwitch0 = lock0<<6;
      int lSwitch1 = lock1<<5;
      int lSwitch2 = lock2<<4;
      int lSwitch3 = lock3<<3;
      int lSwitch4 = lock4<<2;
      int lSwitch5 = lock5<<1;
      int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Serial.print(rCheck,BYTE);
  }
  else if(incomingByte == '8')
  {
   digitalWrite(opPin[4], HIGH);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
    int lSwitch2 = lock2<<4;
    int lSwitch3 = lock3<<3;
    int lSwitch4 = lock4<<2;
    int lSwitch5 = lock5<<1;
    int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
   Serial.print(rCheck,BYTE);
  }
  else if((incomingByte ==
'9')&&((digitalRead(inPin1[4])==LOW)||(digitalRead(inPin2[4])==LOW)))
  {
   digitalWrite(opPin[4], LOW);
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
```

เอกสารนี้เป็นกรรมสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int lSwitch0 = lock0<<6;
        int lSwitch1 = lock1<<5;
        int lSwitch2 = lock2<<4;
        int lSwitch3 = lock3<<3;
        int lSwitch4 = lock4<<2;
        int lSwitch5 = lock5<<1;
        int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
        Serial.print(rCheck,BYTE);
      }
      else if(incomingByte == 'a')
      {
       digitalWrite(opPin[5], HIGH);
        int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
        int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
        int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
        int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
        int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
        int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
        int lock0 = 1;
        int lSwitch0 = lock0<<6;
        int lSwitch1 = lock1<<5;
        int lSwitch2 = lock2<<4;
        int lSwitch3 = lock3<<3;
        int lSwitch4 = lock4<<2;
        int lSwitch5 = lock5<<1;
        int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
       Serial.print(rCheck,BYTE);
      }
      else if((incomingByte ==
    'b')&&((digitalRead(inPin1[5])==LOW)||(digitalRead(inPin2[5])==LOW)))
      {
       digitalWrite(opPin[5], LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
    int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
    int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
    int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
    int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
    int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
    int lock0 = 1;
    int lSwitch0 = lock0<<6;
    int lSwitch1 = lock1<<5;
    int lSwitch2 = lock2<<4;
    int lSwitch3 = lock3<<3;
    int lSwitch4 = lock4<<2;
    int lSwitch5 = lock5<<1;
    int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
    Serial.print(rCheck,BYTE);
}
else if(incomingByte == 'r')
{
  int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
  int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
  int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
  int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
  int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
  int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
  int lock0 = 1;
  int lSwitch0 = lock0<<6;
  int lSwitch1 = lock1<<5;
  int lSwitch2 = lock2<<4;
  int lSwitch3 = lock3<<3;
  int lSwitch4 = lock4<<2;
  int lSwitch5 = lock5<<1;
  int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
      Serial.print(rCheck,BYTE);
    }
    else
    {
     int lock1 = digitalRead(inPin1[0])||digitalRead(inPin2[0]);
     int lock2 = digitalRead(inPin1[1])||digitalRead(inPin2[1]);
     int lock3 = digitalRead(inPin1[2])||digitalRead(inPin2[2]);
     int lock4 = digitalRead(inPin1[3])||digitalRead(inPin2[3]);
     int lock5 = digitalRead(inPin1[4])||digitalRead(inPin2[4]);
     int lock6 = digitalRead(inPin1[5])||digitalRead(inPin2[5]);
     int lock0 = 1;
     int lSwitch0 = lock0<<6;
     int lSwitch1 = lock1<<5;
     int lSwitch2 = lock2<<4;
     int lSwitch3 = lock3<<3;
     int lSwitch4 = lock4<<2;
     int lSwitch5 = lock5<<1;
     int rCheck = lSwitch1+lSwitch2+lSwitch3+lSwitch4+lSwitch5+lock6+lSwitch0;
     Serial.print(rCheck,BYTE);
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรม PHP Flash

```
<?php
$toPHP= $_POST['candidate'];
$FileName = "/dev/ttyUSB0";
$FilePointer = fopen($FileName, "wr+");
fwrite($FilePointer, "$toPHP");
$arduino = fread($FilePointer , 1);
fclose($FilePointer);
?>
```

4. โปรแกรม PHP on

```
<?php
$toPHP= $_POST['candidate'];
$FileName = "/dev/ttyUSB0";
$FilePointer = fopen($FileName, "wr+");
fwrite($FilePointer, "$toPHP");
$arduino = fread($FilePointer , 1);
fclose($FilePointer);
?>
```

5. โปรแกรม Flash Actionscript

```
ON1.addEventListener(MouseEvent.CLICK, fl_ClickToHide_5);

function fl_ClickToHide_5(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "0";

        var myRequest:URLRequest = new URLRequest("test.php");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON1.visible = false;
        LED_OFF1.visible = true;

}

OFF1.addEventListener(MouseEvent.CLICK, fl_ClickToHide_6);

function fl_ClickToHide_6(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "1";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON1.visible = true;
        LED_OFF1.visible = false;
}

ON2.addEventListener(MouseEvent.CLICK, fl_ClickToHide_7);

function fl_ClickToHide_7(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "2";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON2.visible = false;
        LED_OFF2.visible = true;
}

OFF2.addEventListener(MouseEvent.CLICK, fl_ClickToHide_8);

function fl_ClickToHide_8(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "3";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้นำไปดัดแปลง เผยแพร่ หรือนำออกเผยแพร่ต่อ ในรูปแบบอื่นใด ก่อนได้รับอนุญาตเป็นลายลักษณ์อักษรจากผู้เขียนเอกสารทุกครั้งที่มีการนำไปใช้

```
        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON2.visible = true;
        LED_OFF2.visible = false;
}

ON3.addEventListener(MouseEvent.CLICK, fl_ClickToHide_9);

function fl_ClickToHide_9(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "4";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON3.visible = false;
        LED_OFF3.visible = true;
}

OFF3.addEventListener(MouseEvent.CLICK, fl_ClickToHide_10);

function fl_ClickToHide_10(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "5";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
        }
        LED_ON3.visible = true;
        LED_OFF3.visible = false;
}

ON4.addEventListener(MouseEvent.CLICK, fl_ClickToHide_11);

function fl_ClickToHide_11(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "6";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON4.visible = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LED_OFF4.visible = true;
}

OFF4.addEventListener(MouseEvent.CLICK, fl_ClickToHide_14);

function fl_ClickToHide_14(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "7";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON4.visible = true;
        LED_OFF4.visible = false;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ON5.addEventListener(MouseEvent.CLICK, fl_ClickToHide_15);

function fl_ClickToHide_15(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "8";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON5.visible = false;
        LED_OFF5.visible = true;
}

OFF5.addEventListener(MouseEvent.CLICK, fl_ClickToHide_16);

function fl_ClickToHide_16(event:MouseEvent):void
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้นำไปดัดแปลง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "9";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON5.visible = true;
        LED_OFF5.visible = false;
}

ON6.addEventListener(MouseEvent.CLICK, fl_ClickToHide_17);

function fl_ClickToHide_17(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        myVariables.candidate = "a";

        var myRequest:URLRequest = new URLRequest("test.php");

        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON6.visible = false;
        LED_OFF6.visible = true;
}

OFF6.addEventListener(MouseEvent.CLICK, fl_ClickToHide_18);

function fl_ClickToHide_18(event:MouseEvent):void
{
        var myVariables:URLVariables = new URLVariables();

        myVariables.candidate = "b";

        var myRequest:URLRequest = new URLRequest("test.php");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
        myRequest.method = URLRequestMethod.POST;

        myRequest.data = myVariables;

        var myLoader:URLLoader = new URLLoader;

        myLoader.dataFormat = URLLoaderDataFormat.VARIABLES;

        myLoader.load(myRequest);

        myLoader.addEventListener(Event.COMPLETE, loadComplete);

        function loadComplete(evt:Event):void
        {
        }
        LED_ON6.visible = true;
        LED_OFF6.visible = false;
}
```

6. โปรแกรม HTML flash

```
<html>
<title>Locker Control System via Internet</title>

<body>

<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"

codebase="http://download.macromedia.com/pub/shockwave/
cabs/flash/swflash.cab#version=6,0,40,0"
width="1366" height="768"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ทำซ้ำ ดัดแปลง หรือมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 id="mymoviename">

 <param name="movie"

 value="Project_Locker.swf" />

 <param name="quality" value="high" />

 <param name="bgcolor" value="#ffffff" />

 <embed src="Project_Locker.swf" quality="high" bgcolor="#ffffff"

 width="1366" height="768"

 name="mymoviename" align="" type="application/x-shockwave-flash"

 pluginspage="http://www.macromedia.com/go/getflashplayer">


 </embed>

 </object>

 </body>

 </html>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. โปรแกรม HTML on

```
<HTML>
<TITLE>Microcontroller Data Input Form </TITLE>
</HEAD>
<BODY>
<B>Microcontroller Data Input Form</B>
<BR><BR>
<FORM METHOD="GET" ACTION="handleform.php">
   Operate LED  <INPUT TYPE=INTEGER NAME="switch_led1" SIZE=3>
   Put another control here
<BR><BR>
<INPUT TYPE=SUBMIT NAME="SUBMIT" VALUE="Submit!">
</FORM>
</BODY>
</HTML>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้