สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมเก็บข้อมูลชนิดไร้สาย

Wireless Data Logging Program

T119216

นายธนภัทร บุณยศรีสวัสดิ์

นายธีรวัชร สีหานาท

นายนนท์ธวัช จันทมิตร

เลขหมู่.....
เลขทะเบียน 119216
วัน,เดือน,ปี - 6 ส.ค. 2554

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wireless Data Logging Program

Mr. Tanapat Bunyasrisawat

Mr. Thirawat Shihanat

Mr. Nontavat Chantamit

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGNEERING IN CONTROL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ACADEMIC YEAR 2010

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมเก็บข้อมูลชนิดไร้สาย

Wireless Data Logging Program

ผู้จัดทำ นายธนภัทร บุณยศรีสวัสดิ์

นายธีรวัชร สีหานาท

นายนนท์ธวัช จันทมิตร

.....สังวาล บกสุวรรณ.....อาจารย์ที่ปรึกษา

(อาจารย์สังวาล บกสุวรรณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเก็บข้อมูลชนิดไร้สาย

โดย

นายธนภัทร บุณยศรีสวัสดิ์ 50010618

นายธีรวัชร สีหานาท 50010708

นายนนท์ธวัช จันทมิตร 50010732

อาจารย์ที่ปรึกษา

อาจารย์สังวาล บกสุวรรณ

ปีการศึกษา 2553

บทคัดย่อ

การเก็บข้อมูล (Data Logging) เป็นส่วนของการออกแบบระบบควบคุม ซึ่งง่ายที่จะให้ผู้ออกแบบสามารถเข้าถึงข้อมูลในลักษณะเวลาจริง เนื่องด้วยความสำคัญนี้เองโปรเจคนี้จึงออกแบบมาเพื่อความสะดวกกับผู้ใช้ สำหรับการสื่อสารควบคุมระบบแบบวายฟาย (WI-FI) ซึ่งจะป้องกันเหตุการณ์ที่ไม่คาดคิด ในขั้นตอนของการพัฒนาระบบควบคุมจียูไอ (GUI) ที่เรานำเสนอนั้นสามารถที่จะแสดงข้อมูลในระบบที่กำลังปฏิบัติการอยู่ ยิ่งกว่านั้นข้อมูลที่น่าสนใจสามารถที่จะถูกเก็บในลักษณะของไฟล์เพื่อที่จะนำไปวิเคราะห์ต่อไป คุณสมบัติที่สำคัญอีกอย่างหนึ่งของ GUI คือความสามารถในการแสดงผลกราฟและจัดการข้อมูลอย่างอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wireless Data Logging Program

By

Mr. Tanapat Bunyasrisawat 50010618

Mr. Thirawat Shihanat 50010708

Mr. Nontavat Chantamit 50010732

Advisor

Mr. Sungwan Boksuwan

Academic Year 2010

Abstract

A data logging is the vital part in a control system design, which allows designers to access some information in a real-time manner. Because of its importance, the project aims at designing a friendly Graphics User Interface for wireless control systems through WI-FI communication, which protects a computer from undesired situations in developing stage of designing the control systems. The proposed GUI is able to monitor the information of the operating system in the real-time fashion. Moreover, the interesting information can be logged and saved as a file in order to analyze in other computers. The ultimate feature of the proposed GUI is to graph and to manipulate the data automatically.

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก อาจารย์ สังวาล บกสุวรรณ ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆ ที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ ผศ.ดร. ถาวร เบญจนราสุทธิ์ ที่คอยถามถึงความคืบหน้าอยู่ตลอดเวลา และเจ้าหน้าที่ห้องปฏิบัติการ นายอมรเทพ กาพย์แก้ว ที่ได้เอื้อเฟื้ออุปกรณ์ที่จำเป็น ในการทำการทดลองของโครงงานนี้มาโดยตลอด

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงงานอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำโครงงานนี้สำเร็จสมบูรณ์ลงได้

คณะผู้จัดทำ

นายธนภัทร บุณยศรีสวัสดิ์

นายธีรวัชร สีหานาท

นายนนท์ธวัช จันทมิตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	**I**
บทคัดย่อภาษาอังกฤษ	**II**
กิตติกรรมประกาศ	**III**
สารบัญ	**IV**
สารบัญภาพ	**VII**
สารบัญตาราง	**IX**
บทที่ 1 บทนำ	**1**
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน	1
1.4 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	**3**
2.1 TCP/IP Protocol	3
2.1.1 ชั้นโฮสต์-เครือข่าย	5
2.1.2 ชั้นสื่อสารอินเทอร์เน็ต	5
2.1.3 IP	6
2.1.4 ICMP	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.1.5 ชั้นสื่อสารนำส่งข้อมูล 8

2.1.6 UDP 8

2.1.7 TCP 9

2.1.7.1 การสื่อสารของ TCP 11

2.1.7.2 การบันทึกเวลาแบบ Three-way handshake 12

2.1.7.3 ชั้นสื่อสารการประยุกต์ 13

2.2 พื้นฐานการใช้งาน GDI+ 13

2.2.1 พื้นฐานระบบกราฟิกใน GDI+ 14

2.2.1.1 พื้นที่แสดงผล 14

2.2.1.2 พื้นที่จำลอง 14

2.2.2 ขั้นตอนการสร้างพื้นที่จำลองใน GDI+ 14

2.2.3 ระบบสีใน GDI+ 14

2.2.4 การใช้สีจากออบเจ็กต์ Color 14

2.2.5 การใช้งานเหตุการณ์ Paint() ของฟอร์ม 14

2.2.6 การใช้งานออบเจ็กต์ Pen 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 3 หลักการออกแบบ	**16**
3.1 หลักการออกแบบโปรแกรม	16
3.2 ขั้นตอนการสร้างโปรแกรม	16
3.2.1 การเชื่อมต่อโปรแกรม	16
3.2.2 การบันทึกข้อมูล	17
3.2.3 การแสดงผลทางด้านกราฟิก	17
3.2.4 การควบคุมการทำงานของโปรแกรม	18
บทที่ 4 การทดลอง	**19**
4.1 การทดลองโปรแกรมรับส่งข้อมูลผ่านสัญญาณไร้สาย	19
บทที่ 5 บทวิจารณ์และสรุป	**26**
5.1 สรุปผลการทดลอง	26
เอกสารอ้างอิง	**27**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
2.1 ขั้นตอนการ Encapsulation และ Demultiplexing	4
2.2 โครงสร้าง TCP/IP	5
2.3 IP Header	6
2.4 ICMP Header	8
2.5 UDP Header	9
2.6 TCP Header	9
2.7 การสื่อสารของ TCP	11
2.8 ขั้นตอนการทำงานจากโฮสต์ 1 ไปยังโฮสต์ 2	12
3.1 การเชื่อมต่อ	16
3.2 การบันทึก	17
3.3 การแสดงกราฟิก	17
3.4 การควบคุมการทำงาน	18
4.1 โปรแกรมวายฟายพล็อต(WIFI Plot)	19
4.2 โปรแกรมวายฟายดาต้าล็อคกิ้ง (WIFI Data Logging)	20
4.3 โปรแกรมวายฟายพล็อตเมื่อทำการกดปุ่ม Connect แล้ว	20
4.4 โปรแกรมวายฟายดาต้าล็อคกิ้งเมื่อทำการกดปุ่ม Connect แล้ว	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่ **หน้า**

4.5 ข้อมูลกราฟเส้นที่ส่งมาจากโปรแกรมวายฟายพล็อต 21

4.6 ข้อมูลกราฟเส้นที่ทำการส่งไปยังโปรแกรมวายฟายดาต้าล็อคกิ้ง 22

4.7 ภาพขยายกราฟเส้นในโปรแกรมวายฟายดาต้าล็อคกิ้ง 22

4.8 หน้าต่าง save dialog 23

4.9 ตัวอย่างไฟล์ที่ทำการบันทึกเป็นรูปภาพ 23

4.10 ตัวอย่างไฟล์ที่บันทึกเป็นข้อมูล 24

4.11 ตัวอย่างรูปกราฟซายน์ 24

4.12 ตัวอย่างรูปกราฟพาราโบลา 25

4.13 ตัวอย่างรูปกราฟแบบสุ่ม 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 รายละเอียดของ Flag 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

โครงงานนี้มีจุดประสงค์เพื่อแสดงผลการเขียนโปรแกรมที่ใช้ในการติดต่อสื่อสารระหว่างคอมพิวเตอร์(Computer) สองเครื่องผ่านเครือข่ายแลน(LAN) หรือ สัญญาณไวเลส และโปรแกรมที่เป็นส่วนประกอบของโปรแกรม โดยใช้โปรแกรมวิชชวลสตูดิโอ 2008 (Visual Studio 2008) และได้ศึกษาองค์ประกอบพื้นฐานของโปรแกรมในส่วนของวินโดส์แอปพลิเคชั่น(Windows application) และคอนโทรลพื้นฐาน เช่น การคอนโทรลบัททอล การคอนโทรลเลเบิล และลิ้งเลเบิล การคอนโทรลเท็กบ๊อก และการโปรแกรมในส่วนของเน็ทเวิร์คสตรีม คลาสไบนารีรีดเดอร์ และไบนารีไรเตอร์ เพื่อเป็นตัวติดต่อรับส่งข้อมูลไร้สายระหว่างอุปกรณ์ในส่วนของตัวแสดงผลกราฟผลตอบสนองของระบบจะใช้การโปรแกรมกราฟิก ด้วยจีดีไอพลัส ซึ่งก็คือการโปรแกรมคลาสพื้นฐานทางด้านกราฟิกเพื่อทำการพล็อตกราฟที่จะแสดงผลการตอบสนองของสัญญาณ

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. ใช้โปรแกรม Visual C# 2008 เขียนโปรแกรมที่ใช้ในการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์เพื่อนำมาแสดงผลการควบคุมระบบแบบไร้สายผ่านหน้าจอคอมพิวเตอร์
2. ศึกษาหลักการทำงานของตัวส่งสัญญาณไวเลส แบบวายฟาย

1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

ในการศึกษาที่จะเขียนโปรแกรมแอปพลิเคชั่น เพื่อใช้ในการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์(Microcontroller) โดยผ่านทางสัญญาณไวเลส แบบวายฟาย โดยใช้โปรแกรม Visual C# 2008 ซึ่งเป็นโปรแกรมที่พัฒนามาจากโปรแกรมวิชชวลซีพลัสพลัส (Visual C++) ทำให้มีการใช้งานที่ง่ายยิ่งขึ้น เพื่อเป็นพื้นฐานที่จำนำไปสู่การเขียนโปรแกรมในระดับที่สูงขึ้นไปในอนาคต การสร้างแอปพลิเคชั่น ในการติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ นั้น ต้องทำการศึกษาทฤษฎี TCP/IP Protocol และ ทฤษฎี GDI+

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์นี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวนำถึงวัตถุประสงค์ ขั้นตอนการศึกษา และการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้อง ในสร้างวินโดว์แอปพลิเคชั่น เพื่อติดต่อกับไมโครคอนโทรลเลอร์ การใช้งานชุดโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเตอร์เน็ต และชั้นของการสื่อสารในชุดของโปรโตคอล

บทที่ 3 หลักการออกแบบ การนำเสนอหลักการออกแบบวินโดว์แอปพลิเคชั่น ที่จะติดต่อสื่อสารกับไมโครคอนโทรลเลอร์แบบไร้สาย

บทที่ 4 การทดลอง เป็นส่วนการทดสอบองค์ประกอบต่างๆ ในวินโดว์แอปพลิเคชั่น ตลอดจนการทดลองติดต่อสื่อสารแบบไร้สายกับไมโครคอนโทรลเลอร์

บทที่ 5 บทวิจารณ์และสรุป จะสรุปผลการดำเนินงาน ปัญหาที่จะเกิดขึ้น และแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 TCP/IP Protocol

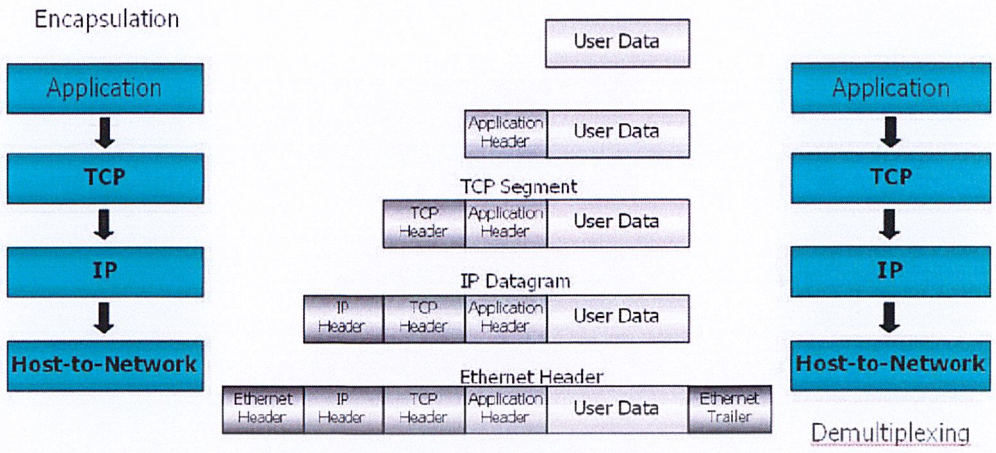
TCP/IP (Transmitsion Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอล (Protocol) ที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ตโดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอล(Protocol)ก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้TCP/IPเป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน สามประการคือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนด(node)กลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิด ทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบเรียลไทม์(Realtime) และทั้งการสื่อสารแบบเสียง(Voice) และข้อมูล(data)

การส่งข้อมูลผ่านในแต่ละเลเยอร์(Layer) แต่ละเลเยอร์(Layer)จะทำการประกอบข้อมูลที่ได้รับมากับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่าเฮดเดอร์(Header) ภายในเฮดเดอร์(Header) จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการเอนแคปซูเลท(Encapsulate) เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือโปรโตคอล(Protocol) เดียวกันทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็นเฮดเดอร์(Header) ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า ดีมัลติเพลกซิ่ง(Demultiplexing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



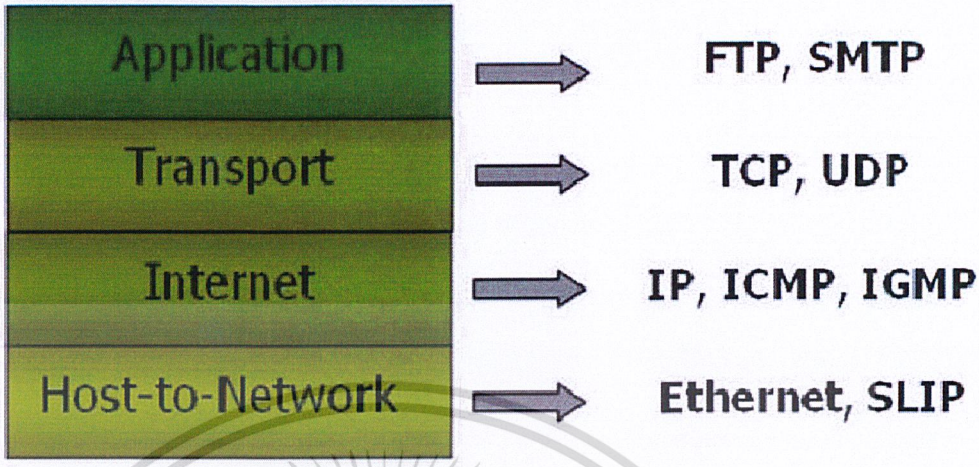
รูปที่ 2.1 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการเอนแคปซูเลท(Encapsulate) ในแต่ละเลเยอร์(Layer) มีชื่อเรียกแตกต่างกัน ดังนี้

- ข้อมูลที่มาจากผู้ใช้(User) หรือก็คือข้อมูลที่ผู้ใช้(User) เป็นผู้ป้อนให้กับแอปพลิเคชั่น (Application) เรียกว่า ยูสเซอร์ดาต้า(User Data)
- เมื่อแอพพลิเคชั่น(Applicaion) ได้รับข้อมูลจากผู้ใช้(User) ก็จะนำมาประกอบกับส่วนหัวของแอพพลิเคชั่นเรียกว่า แอปพลิเคชั่นดาต้า(Application Data) และส่งต่อไปยังโปรโตคอลทีซีพี(Protocol TCP)
- เมื่อโปรโตคอลทีซีพีได้รับแอปพลิเคชั่นดาต้า(Application Data) ก็จะนำมารวมกับเฮดเดอร์ของโปรโตคอลทีซีพีเรียกว่า ทีซีพีเซกเมนต์(TCP Segment) และส่งต่อไปยังโปรโตคอลไอพี(Protocol IP)
- เมื่อโปรโตคอล(Protocol) IP ได้รับทีซีพีเซกเมนต์ก็จะนำมารวมกับเฮดเดอร์ของโปรโตคอลไอพีเรียกว่า ไอพีดาต้าแกรม(IP Datagram) และส่งจากโฮสต์ไปยังเน็ทเวิร์ค เลเยอร์ (Host-to-Network Layer)
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้



รูปที่ 2.2 โครงสร้าง TCP/IP

2.1.1 ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โพรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการหน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูลทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

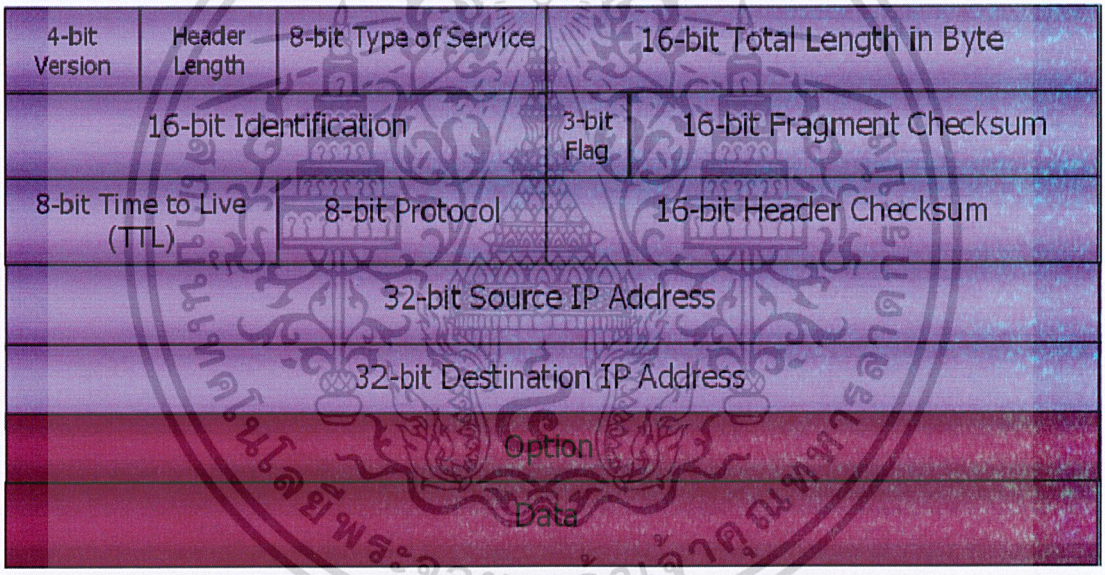
2.1.2 ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (Packet Switching Network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็กเก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งแพ็กเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย แพ็กเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกัน ดังนั้น แพ็กเก็ตที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 IP (Internet Protocol)

IP เป็นโปรโตคอลในระดับเน็ตเวิร์คเลเยอร์ ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเก็ต ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และมีระบบการแยกและประกอบดาต้าแกรม (datagram) เพื่อรองรับการส่งข้อมูลระดับ data link ที่มีขนาด MTU (Maximum Transmission Unit) ที่แตกต่างกันทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลายเช่น Ethernet ,Token Ring หรือ Apple Talk การเชื่อมต่อของ IP เพื่อทำการส่งข้อมูลจะเป็นแบบ Connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆ ครั้งของการส่งข้อมูล 1 ดาต้าแกรม โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (Fragmentation) และถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง



รูปที่ 2.3 IP Header

เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม Option บางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

Version : หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือเวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)

Header Length : ความยาวของเฮดเดอร์ โดยทั่วไปถ้าไม่มีส่วน optionจะมีค่าเป็น 5 (5*32 bit)

Type of Service (TOS) : ใช้เป็นข้อมูลสำหรับเราเตอร์(Router) ในการตัดสินใจเลือกการเราต์ข้อมูลในแต่ละดาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Length : ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรม ซึ่งด้วยขนาด16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของดาต้าแกรม คือ 65535byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอพพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 byte

Identification : เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี Identification เดียวกันมารวมกัน

Flag : ใช้ในกรณีที่มีการแยกดาต้าแกรม

Fragment offset : ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

Time to live (TTL) : กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้ง กลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

Protocol : ระบุโปรโตคอลที่ส่งในดาต้าแกรม เช่น TCP ,UDP หรือ ICMP

Header checksum : ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์

Source IP address : หมายเลข IP ของผู้ส่งข้อมูล

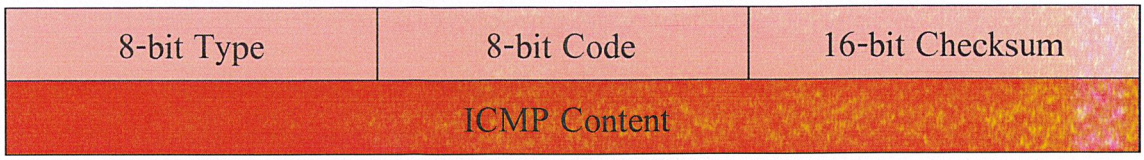
Destination IP address : หมายเลข IP ของผู้รับข้อมูล

Data : ข้อมูลจากโปรโตคอลระดับบน

2.1.4 ICMP (Internet Control Message Protocol)

ICMP เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม (Datagram) ในกรณีที่เกิดปัญหากับดาต้าแกรม เช่น เราเตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต้นทางเพื่อรายงานข้อผิดพลาด ที่เกิดขึ้น อย่างไรก็ดี ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริงหรือไม่หากมีการส่งดาต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมา ก็แปลความหมายได้สองกรณีคือข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสารทั้งการส่งดาต้าแกรม และ ICMP Message ที่ส่งกลับมาก็มีปัญหาระว่างทางก็ได้ ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (Unreliable) ซึ่งจะเป็นหน้าที่ของ โปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารนั้นๆ มีความน่าเชื่อถือในส่วนของ ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิตและส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ICMP Header

2.1.5 ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

แบ่งเป็นโพรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (Connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า Message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ Message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วยโปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client/server system) ซึ่งมีการสื่อสารแบบ ถาม/ตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเทอร์เน็ต

2.1.6 UDP (User Datagram Protocol)

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูลกลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไขหรือมีความผิดพลาดระหว่างการส่ง 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 เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16-bit Source Port	16-bit Destination Port
Lenght	Checksum
Data	

รูปที่ 2.5 UDP Header

Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

UDP Length : ความยาวของดาต้าแกรม ทั้งส่วน Header และ data นั่นหมายความว่าค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header

Checksum : เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

2.1.7 TCP (Transmission Control Protocol)

อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดยดาต้าแกรมของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (Reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (Connection - oriented)

<table>
<tr><td colspan="8">16-bit Source Port Number</td><td colspan="2">16-bit Source Destination Port</td></tr>
<tr><td colspan="10">32-bit Sequence Number</td></tr>
<tr><td colspan="10">32-bit Acknowledge Number</td></tr>
<tr><td>Header Length</td><td>6-Bit Reserved</td><td>URG</td><td>ACK</td><td>PUSH</td><td>RESET</td><td>SYN</td><td>FIN</td><td colspan="2">16-bit Windows Size</td></tr>
<tr><td colspan="8">16-bit TCP Checksum</td><td colspan="2">16-bit Urgent Pointer</td></tr>
<tr><td colspan="10">TCP Option</td></tr>
<tr><td colspan="10">Data</td></tr>
</table>

รูปที่ 2.6 TCP Header

Source Port Number : หมายเลขพอร์ต(Port) ต้นทางที่ส่งดาต้าแกรมนี้

Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sequence Number : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง

Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ

Header Length : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ Option แต่ต้องไม่เกิน 60 ไบต์

Flag : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

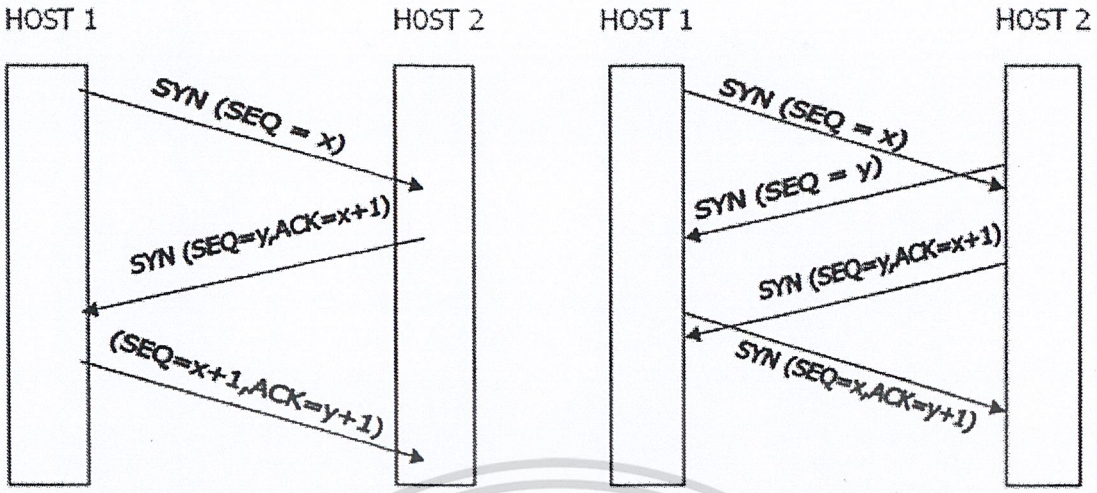
Type	Description
URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent Pointer)
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่กำลังรออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ

ตารางที่ 2.1 รายละเอียดของ Flag

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน Flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่น ฟิลด์ Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มี flag เป็นตัวกำหนดก็อาจจะมีการนำข้อมูลมาใช้และก่อให้เกิดความผิดพลาดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7.1 การสื่อสารของ TCP



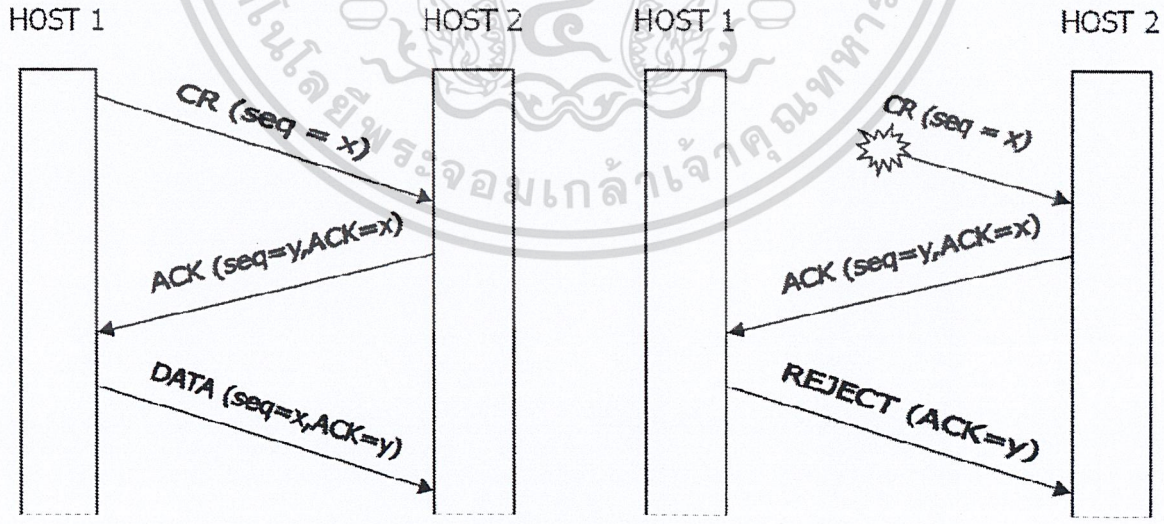
รูปที่ 2.7 การสื่อสารของ TCP

เมื่อเซกเมนต์ CONNECT (SYN = "1" และ ACK = "0") เดินทางมาถึง Entity TCP ที่โฮสต์ปลายทางจะค้นหาโพรเซส(Process) ตามหมายเลขพอร์ตที่กำหนดในเขตข้อมูล Destination port ซึ่งถ้าหากไม่พบก็จะตอบปฏิเสธด้วยเซกเมนต์ที่มี RST = "1" กลับไปยังผู้ส่งเซกเมนต์ CONNECT ของผู้ส่งจะถูกส่งต่อไปยังโพรเซส ตามพอร์ตที่ระบุซึ่งอาจจะตอบรับหรือตอบปฏิเสธก็ได้ถ้าโพรเซสนั้นต้องการสื่อสารด้วยก็จะส่งเซกเมนต์ตอบรับกลับไป รูปที่ 2.7 แสดงลำดับขั้นตอนการส่ง TCP เซกเมนต์ในการสร้างการเชื่อมต่อในสภาวะปกติระหว่างผู้ส่งและผู้รับในกรณีที่โฮสต์สองแห่งพยายามสร้างการเชื่อมต่อระหว่างซ็อคเก็ต(Socket) คู่เดียวกัน ผลสุดท้ายจะมีการเชื่อมต่อเกิดขึ้นเพียงหนึ่งช่องทางเท่านั้นเนื่องจากการเชื่อมต่อในแต่ละช่องทางจะถูกกำหนดขึ้นโดยใช้หมายเลขซ็อคเก็ตผู้ส่งและผู้รับ ถ้าการเชื่อมต่อลำดับแรกสำเร็จก็จะถูกบันทึกไว้ในตารางการสื่อสาร เช่น (x, y) ถ้าการเชื่อมต่อลำดับที่สองสำเร็จในเวลาต่อมา ข้อมูลนี้ก็จะถูกบันทึกไว้ที่เดียวกันคือ (x, y)ขั้นตอนในการสร้างการเชื่อมต่อและการยกเลิกสามารถเขียนอธิบายด้วยไฟไนท์สเตทแมชชีน(Finite State Machine)ที่มีการทำงาน 11 สถานะ ในแต่ละสถานะจะมีเหตุการณ์บางอย่างที่เป็นไปได้ซึ่งจะได้รับการตอบสนองด้วยการกระทำที่เหมาะสม ในทางตรงกันข้ามเหตุการณ์ที่เป็นไปไม่ได้จะกลายเป็นข้อผิดพลาดที่จะต้องรายงานให้ทราบการเชื่อมต่อเริ่มต้นจากสถานะ CLOSED เมื่อเรียกใช้บริการ LISTEN หรือ CONNECT ก็จะมีการเปลี่ยนสถานะไปจากเดิม และถ้าอีกฝ่ายตองการเชื่อมต่อด้วยการเชื่อมต่อก็จะเกิดขึ้นและย้ายไปอยู่ในสถานะ ESTABLISHED คือการเชื่อต่อสมบูรณ์ และเมื่อยกเลิกการติดต่อก็จะกลับไปสู่สถานะ CLOSED อย่างเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7.2 การบันทึกเวลาแบบ Three-way handshake

Three-way Handshake เป็นวิธีการส่งแพ็กเก็ตที่สามารถช่วยแก้ปัญหาในเรื่องแพ็กเก็ตซ้ำซ้อนได้ดีแต่วิธีนี้จำเป็นจะต้องสร้างช่องสื่อสารให้ได้ก่อนที่จะเริ่มรับ-ส่งข้อมูล อย่างไรก็ตามแพ็กเก็ตควบคุมที่ใช้ในการต่อรองค่าตัวแปรสำหรับการสื่อสารต่างๆ อาจเกิดการตกค้างอยู่ในระบบได้ทำให้การกำหนดค่าหมายเลขลำดับมีปัญหาไปด้วย เช่นการสร้างช่องสื่อสารระหว่างโฮสต์ 1 และ โฮสต์2 เริ่มจาก โฮสต์1 ขอเริ่มการเชื่อต่อด้วยการส่งแพ็กเก็ต CR (Connection Request) ไปยังโฮสต์2 ซึ่งจะมีค่าตัวแปรต่างๆสำหรับการสื่อสารรวมทั้งหมายเลขลำดับและหมายเลขช่องสื่อสารไปด้วย ผู้รับคือโฮสต์2 ก็จะส่ง ACK (Acknowledge)กลับมายังโฮสต์1 แต่ถ้าแพ็กเก็ตจากผู้ส่งเกิดสูญหายระหว่างทางและสำเนาแพ็กเก็ตที่ยังตกค้างอยู่ระบบเกิดเดินทางไปถึงผู้รับในภายหลังก็จะทำให้การสร้างช่องสื่อสารใช้การไม่ได้เนื่องจากมีค่าตัวแปรต่างๆไม่ตรงกันการใช้ Three-way handshake เป็นการไม่บังคับให้ผู้ส่งและผู้รับข้อมูลจะต้องกำหนดค่าเริ่มต้นของหมายเลขลำดับเป็นเลขเดียวกัน ทำให้สามารถนำวิธีนี้มาใช้ร่วมกับวิธีการจัดจังหวะการทำงานให้พร้อมกัน(Synchronization) แบบต่างๆได้แทนที่จะเป็นการใช้วิธีการบันทึกเวลา ดังรูปที่ 2.8 แสดงขั้นตอนการเริ่มต้นการทำงานจากโฮสต์ 1 ไปยังโฮสต์ 2 สมมุติให้โฮสต์ 1 เลือกหมายเลขลำดับเป็น “x” และส่งแพ็กเก็ต CONNECTION REQUEST ไปยังโฮสต์ 2 โฮสต์ 2 ตอบรับด้วยแพ็กเก็ต CONNECTION ACCEPTED ซึ่งจะยอมรับหมายเลขลำดับ “x” พร้อมกับประกาศหมายเลขลำดับ “y” ที่เป็นของตนเอง จากนั้น โฮสต์ 1 ก็จะตอบรับค่าตัวเลือกของโฮสต์ 2 ผ่านทางเขตข้อมูลสำหรับการควบคุมในแพ็กเก็ตข้อมูลแรกที่ส่งมา



รูปที่ 2.8 ขั้นตอนการทำงานจากโฮสต์ 1 ไปยังโฮสต์ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมติว่าได้เกิดปัญหาการสูญหายของแพ็กเก็ตในขณะที่สำเนาแพ็กเก็ตที่ค้างในระบบเดินทางไปถึงผู้รับแทน รูปที่2.8 แสดงเหตุการณ์ที่แพ็กเก็ตTPDU (ตัวแรกในรูป) เป็นสำเนาแพ็กเก็ตเก่าที่พึ่งจะเดินทางไปถึงโฮสต์ 2 โดยที่โฮสต์ 1 ไม่ทราบ โฮสต์ 2 ก็จะทำงานตามปกติคือจะตอบรับด้วยการส่งแพ็กเก็ตCONNECTION ACCEPTED TPDU กลับมา ที่โฮสต์ 1 ซึ่งโฮสต์1 จะสามารถตรวจสอบได้ว่าหมายเลขลำดับโฮสต์ 2 ตอบกลับมานั้นเป็นหมายเลขลำดับที่ได้เลิกใช้ไปแล้วจึงมีการส่งแพ็กเก็ต REJECT กลับมายังโฮสต์ 2 เพื่อบอกยกเลิกการทำงาน จะเห็นว่าวิธีการนี้อาศัยการสื่อสารผ่านแพ็กเก็ต 3 ตัวซึ่งเป็นที่มาของคำว่า “การจับมือร่วมสามขั้นตอน” ผลสุดท้ายทั้งโฮสต์ 1 และโฮสต์ 2 ก็จะไม่มีการสร้างช่องสื่อสารขึ้นมาจากข้อมูลในสำเนาแพ็กเก็ตเก่าแต่อย่างใด

2.1.7.3 ชั้นสื่อสารการประยุกต์ (Application Layer)

มีโพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โพรโตคอลสำหรับการจัดการแฟ้มข้อมูลเรียกว่า FTP และโพรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์เรียกว่า SMTP โดยโพรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โพรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้โพรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

2.2 พื้นฐานการใช้งาน GDI+

คำว่า GDI+ (อ่านว่า จีดีไอพลัส) ย่อมาจากคำว่า Graphics Device Interface เป็นคำที่ใช้เรียกกลุ่มออบเจ็กต์(Object)ที่รับผิดชอบด้านการแสดงผลในสถาปัตยกรรม .NET

GDI+ แยกได้ 5 ลักษณะ คือ

- ส่วนแสดงผลของคอนโทรลต่าง ๆ
- การจัดการเกี่ยวกับระบบไฟล์รูปภาพ
- ระบบการพิมพ์เอกสารไปยังเครื่องพิมพ์
- การทำภาพเคลื่อนไหวแอนิเมชั่น (Animation)
- ระบบฟอนต์และสี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 พื้นฐานระบบกราฟิกใน GDI+

พื้นฐานการแสดงกราฟิกหน่วยที่เล็กที่สุดคือ “จุด” เมื่อนำจุดมาเรียงต่อกันจะได้เส้นตรง และเมื่อจุดหลายจุด หลายสีมาเรียงต่อกันก็จะได้ภาพขึ้นมา ออบเจ็กต์ที่ทำหน้าที่รับผิดชอบหน่วยกราฟิกที่เล็กที่สุดมีอยู่ 2 ตัว คือ

- **ออบเจ็กต์ Point** ทำหน้าที่ระบุพิกัดคู่ลำดับ (Co-ordinate) x,y โดยเทียบจากมุมซ้ายบนของพื้นที่แสดงผล
- **ออบเจ็กต์ Size** ทำหน้าที่กำหนดพื้นที่จำลองขึ้นมา เพื่อสร้างงานด้านกราฟิกโดยทำงานร่วมกับออบเจ็กต์ Rectangle อาจจะใช้พื้นที่แสดงผลทั้งหมดของฟอร์ม หรืออยู่ในขอบเขตของพื้นที่จำลองก็ได้

2.2.1.1 พื้นที่แสดงผล หมายถึง พื้นที่แสดงผลทั้งหมดของฟอร์ม เรียกว่า ClientRectangle

2.2.1.2 พื้นที่จำลอง หมายถึง พื้นที่ที่คุณสร้างขึ้นมาด้วยออบเจ็กต์ Size เพื่อใช้เป็นเนื้อที่สำหรับแสดงงานด้านกราฟิก สามารถกำหนดขนาดพื้นที่จำลองได้ที่เราต้องการ ซึ่งอาจจะใช้พื้นที่บางส่วน หรือทั้งหมดของพื้นที่แสดงผลในฟอร์มก็ได้

2.2.2 ขั้นตอนการสร้างพื้นที่จำลองใน GDI+

- ใช้ออบเจ็กต์ Point ระบุตำแหน่ง โดยวัดจากมุมซ้ายบนของฟอร์มกับมุมซ้ายบนของพื้นที่จำลอง
- ใช้ออบเจ็กต์ Size กำหนดขนาดพื้นที่จำลองที่ต้องการสร้างขึ้นมา

2.2.3 ระบบสีใน GDI+

ก่อนที่เราจะเริ่มสร้างงานด้านกราฟิก VC# พื้นฐานแรกที่เราควรจะทราบก็คือ ระบบสี ในสถาปัตยกรรม GDI+ จะกำหนดให้ออบเจ็กต์ Color รับผิดชอบส่วนของการแสดงสีทั้งหมด โดยอ้างอิงระบบแม่สีแบบ RGB (แดง เขียว น้ำเงิน)

2.2.4 การใช้สีจากออบเจ็กต์ Color

สามารถระบุได้ 2 ลักษณะ

- อาศัยเมธอด FromArgb() ทำหน้าที่ผสมแม่สี 3 สี คือ Red, Green, Blue
- ระบุชื่อสีที่ออบเจ็กต์ Color สนับสนุน เช่น Color.White, Color.Red

2.2.5 การใช้งานเหตุการณ์ Paint() ของฟอร์ม

สำหรับวิธีการที่สามารถสั่งให้วาดงานด้านกราฟิกลงในพื้นที่แสดงผลของฟอร์ม สามารถทำได้ 2 วิธีคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- อาศัยเหตุการณ์ Paint() ของฟอร์มเป็นวิธีพื้นฐาน โดยกำหนดให้เมื่อเกิดเหตุการณ์ Form_Paint() ขึ้นแล้ว สั่งให้วาดงานด้านกราฟิกลงในพื้นที่แสดงผล โดยเหตุการณ์ Form_Paint() เป็นเหตุการณ์ที่เกิดต่อจากเหตุการณ์ Form_Load() อัตโนมัติ
- การทำ override เหตุการณ์ Form_Paint() ของฟอร์ม เป็นวิธีการที่เราสามารถจำลองหรือเลียนแบบเหตุการณ์ Form_Paint() โดยการใช้คำสั่ง override

2.2.6 การใช้งานออบเจ็กต์ Pen

ออบเจ็กต์ Pen ใช้สำหรับวาดเส้น โดยการกำหนดจุดเริ่มต้นและจุดสิ้นสุดด้วยออบเจ็กต์ Point ซึ่งเป็นออบเจ็กต์ที่มีลักษณะใกล้เคียงกับปากกาของจริง เพราะว่าในการเลือกปากกา สิ่งที่ต้องคำนึงถึงมี 2 อย่างคือ

- สีน้ำหมึกของปากกา เช่น หมึกสีแดง หมึกสีดำ ฯลฯ
- ขนาดหัวปากกาที่ใช้ เช่น ขนาด 0.5, 0.3 ฯลฯ

นั่นคือ สิ่งที่คิดเมื่อใช้ปากกาในโลกของความเป็นจริง แต่ในโลกของ GDI+ แล้วปากกาสามารถกำหนดได้ 3 อย่างคือ

- สีน้ำหมึก
- ขนาดหัวปากกา
- ลวดลายเส้นที่ออกจากหัวปากกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการออกแบบ

3.1 หลักการออกแบบโปรแกรม

หลักการออกแบบนั้นพิจารณาจากผลที่ต้อง ซึ่งในที่นี้สิ่งที่ต้องการแบ่งออกเป็นส่วนย่อยอันประกอบไปด้วย ส่วนของการติดต่อสื่อสาร ส่วนของการรับส่งข้อมูล ส่วนของการแสดงผลทางด้านกราฟิก และ ส่วนของการบันทึกข้อมูล

ซึ่งในที่นี้ได้ทำการออกแบบโปรแกรมเพื่อรับส่งข้อมูลไร้สายแบบวายฟาย ภายในโปรแกรมสามารถทำการเชื่อมต่อกันและรับส่งข้อมูลระหว่างส่วนของเครื่องแม่ข่าย กับ ส่วนของเครื่องลูกข่าย สามารถแสดงผลข้อมูลในรูปแบบของกราฟเส้น และ สามารถบันทึกข้อมูลในรูปแบบของรูปภาพ และบันทึกในรูปแบบของข้อมูลเพื่อนำไปวิเคราะห์ในขั้นตอนต่อไป

3.2 ขั้นตอนการสร้างโปรแกรม

3.2.1 การเชื่อมต่อโปรแกรม

รูปที่ 3.1 การเชื่อมต่อ

ส่วนนี้เป็นการเชื่อมต่อกันระหว่างเครื่องแม่ข่ายกับเครื่องลูกข่าย

1. กำหนด Namespace Socket	using System.Net.Sockets
2. สร้าง Class TcpClient	TcpClient client = new TcpClient()
3. กำหนด Method Connect	client.Connect(ipAddress, port)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

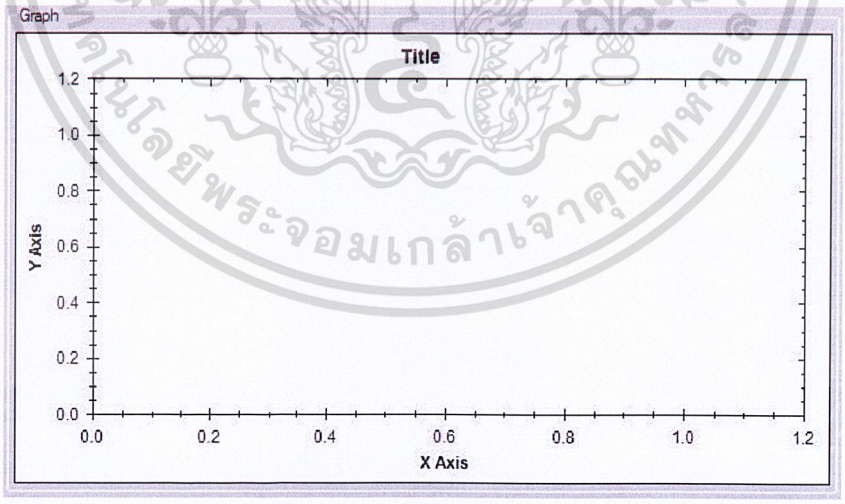
3.2.2 การบันทึกข้อมูล

รูปที่ 3.2 การบันทึก

ส่วนนี้เป็นการบันทึกข้อมูลในรูปแบบรูปภาพ และ รูปแบบข้อมูล

1. นำ SaveFileDialog ภายใน toolbox มาวาง
2. กำหนด Filter ใน Properties ของ SaveFileDialog เพื่อกำหนดชนิดข้อมูล
3. ใช้ Method ShowDialog SaveFileDialog.ShowDialog()

3.2.3 การแสดงผลทางด้านกราฟิก



รูปที่ 3.3 การแสดงกราฟิก

ส่วนนี้เป็นการแสดงภาพข้อมูลที่ได้รับออกมาในรูปแบบของกราฟเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

119216

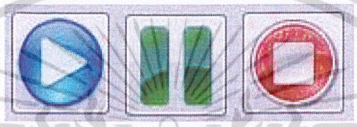
1. กำหนด Namespace ZedGraph using ZedGraph

2. นำ ZedGraphControl ภายใน toolbox มาวาง

3. กำหนด Class GraphPane ZedGraphControl.GraphPane

4. กำหนด Method AddCurve AddCurve(String,IPointList,Color,SymbolType)

3.2.4 การควบคุมการทำงานของโปรแกรม



รูปที่ 3.4 การควบคุมการทำงาน

ส่วนนี้เป็นการสั่งการทำงานของโปรแกรม ซึ่งประกอบด้วยปุ่มเริ่มทำงาน (Play), ปุ่มหยุดชั่วคราว (Pause), ปุ่มหยุด (Stop) ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

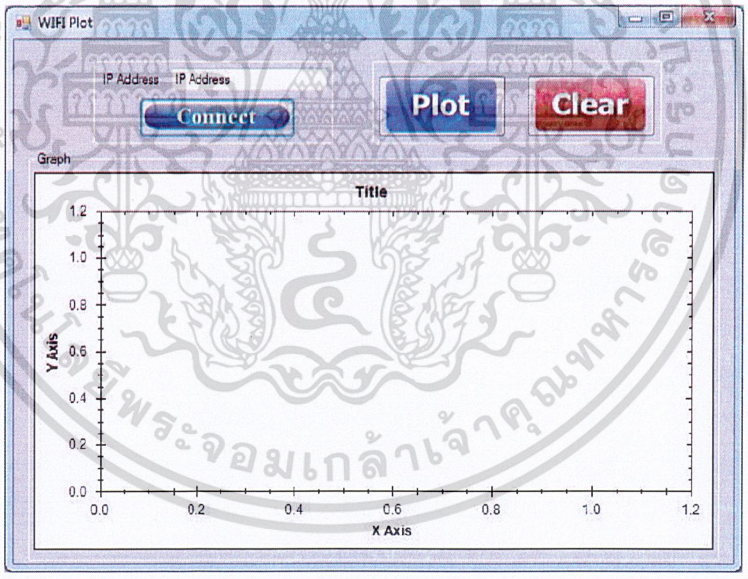
บทที่ 4

การทดลอง

ในบทนี้จะกล่าวถึงการทดลองและผลการทดลองในการเขียนโปรแกรมเพื่อใช้ในการรับส่งข้อมูลผ่านสัญญาณไร้สายในระบบวายฟาย โดยมีรายละเอียดในการทดลองดังนี้

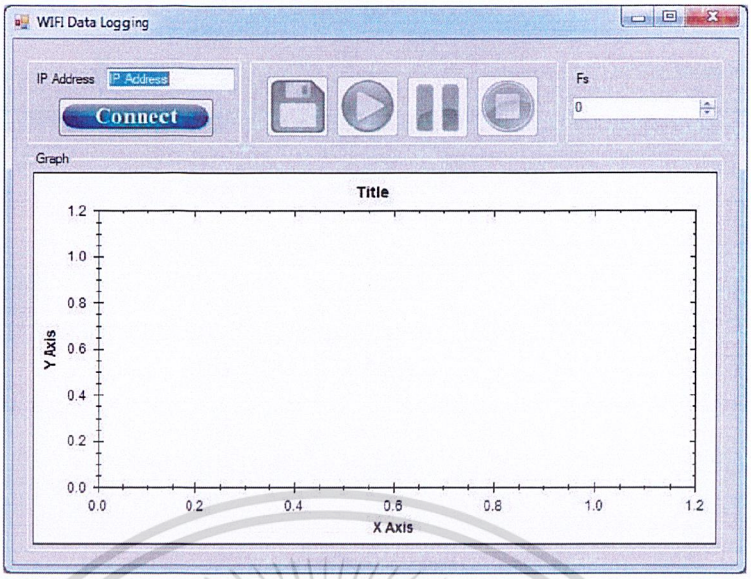
4.1 การทดลองโปรแกรมรับส่งข้อมูลผ่านสัญญาณไร้สาย

ในส่วนนี้จะเป็นการทดการส่งข้อมูลผ่านสัญญาณไร้สายในระบบวายฟายโดยส่งจากโปรแกรมวายฟายพล็อต (WIFI Plot) ไปยังโปรแกรมวายฟายดาต้าล็อคกิ้ง (WIFI Data Logging) และทำการบันทึกข้อมูลที่แสดงผลจากโปรแกรมวายฟายดาต้าล็อคกิ้ง (WIFI Data Logging)



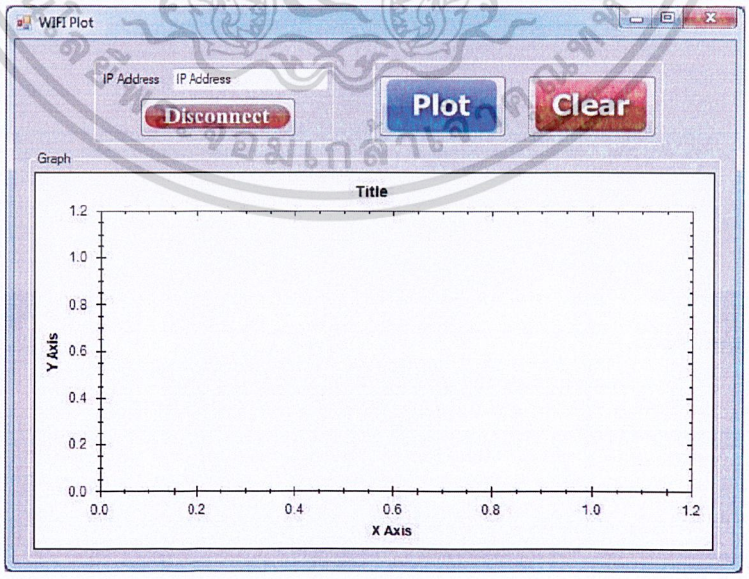
รูปที่ 4.1 โปรแกรมวายฟายพล็อต(WIFI Plot)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



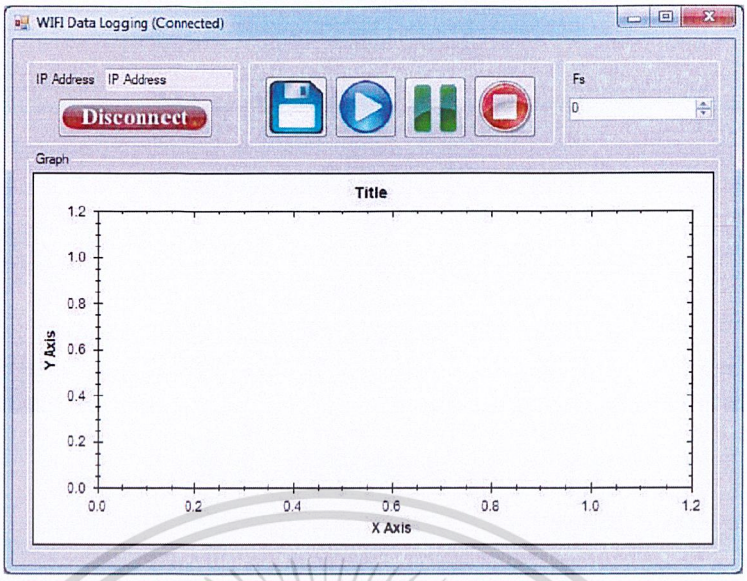
รูปที่ 4.2 โปรแกรมวายฟายดาต้าล็อคกิ้ง (WIFI Data Logging)

ในการทดลองจะเริ่มโดยการให้โปรแกรมวายฟายพล็อตอยู่ที่เครื่องแม่ข่ายส่วนโปรแกรมวายฟายดาต้าล็อคกิ้งให้อยู่ที่เครื่องลูกข่ายจากนั้นทำการเชื่อมต่อทั้งสองโปรแกรมเข้าด้วยกันโดยการใส่หมายเลข IP Address ของเครื่องแม่ข่ายที่ช่องใส่หมายเลข IP Address ของทั้งสองโปรแกรม หลังจากนั้นทำการกดปุ่ม Connect ของทั้งสองโปรแกรมดังรูปที่ 4.3 และรูปที่ 4.4



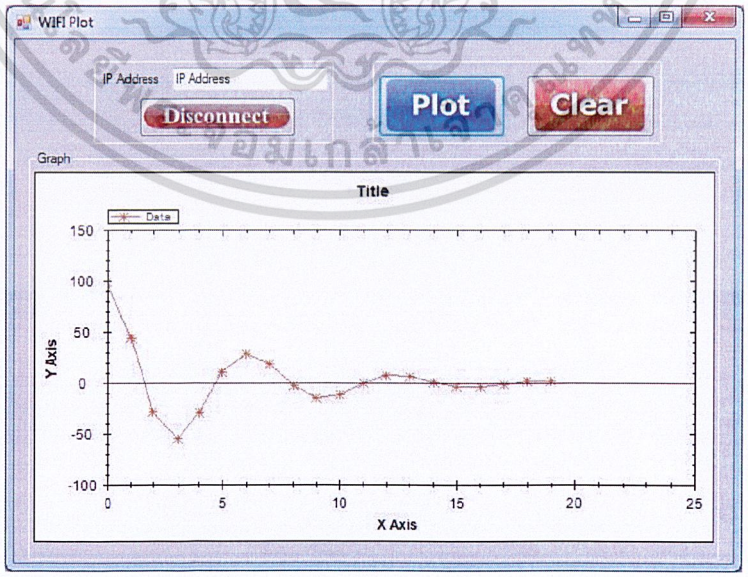
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 โปรแกรมวายฟายพล็อตเมื่อทำการกดปุ่ม Connect แล้ว



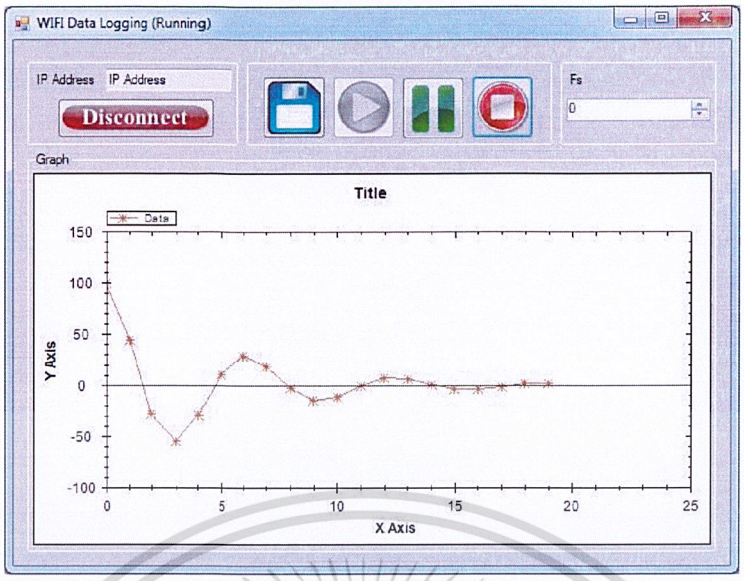
รูปที่ 4.4 โปรแกรมวายฟายดาต้าล็อคกิ้งเมื่อทำการกดปุ่ม Connect แล้ว

ทำการกดปุ่ม ที่โปรแกรมวายฟายดาต้าล็อคกิ้งสั่งให้โปรแกรมทำการเริ่มต้นการรับข้อมูลที่จะส่งมาจากเครื่องแม่ข่าย หลังจากนั้นกดปุ่ม โปรแกรมวายฟายพล็อตเพื่อเริ่มต้นการพล็อตกราฟและเป็นเริ่มต้นส่งข้อมูลมายังเครื่องลูกข่ายเพื่อแสดงผลที่หน้าต่างของโปรแกรมวายฟายดาต้าล็อคกิ้งดังรูปที่ 4.5 และรูปที่ 4.6



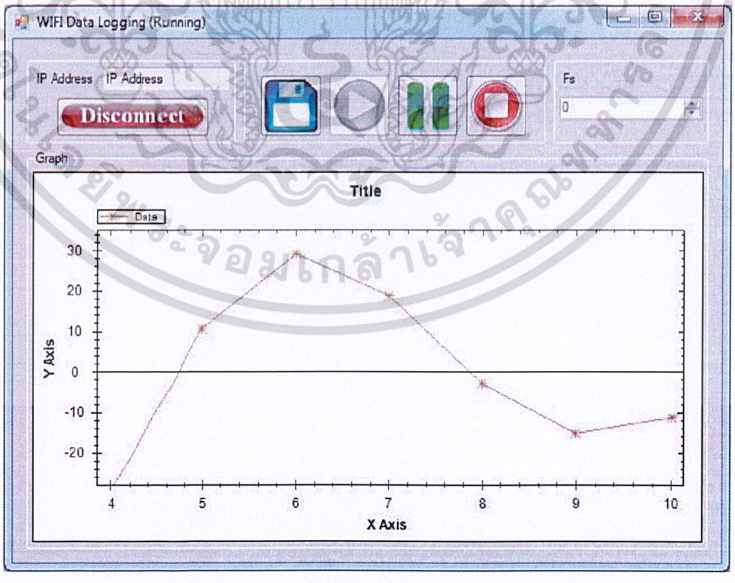
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 ข้อมูลกราฟเส้นที่ส่งมาจากโปรแกรมวายฟายพล็อต



รูปที่ 4.6 ข้อมูลกราฟเส้นที่ทำการส่งไปยังโปรแกรมวายฟายดาต้าล็อคกิ้ง

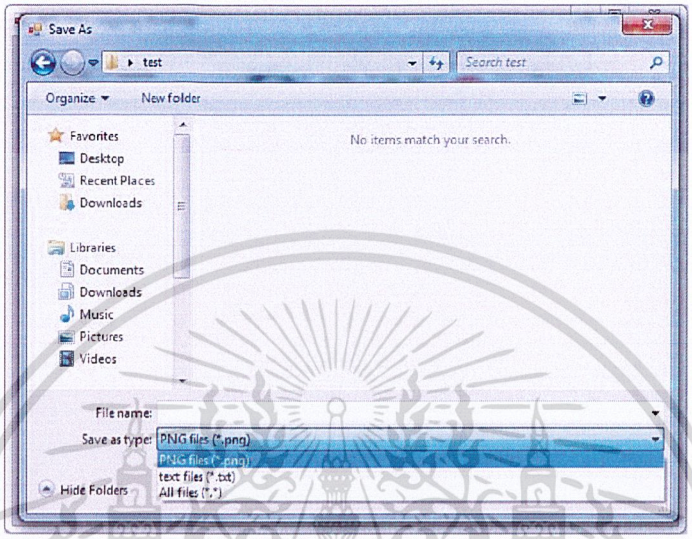
เมื่อโปรแกรมวายฟายดาต้าล็อคกิ้งแสดงข้อมูลแล้วสามารถทำการขยายกราฟไปยังจุดที่สนใจเพื่อดูค่าที่ต้องการ



รูปที่ 4.7 ภาพขยายกราฟเส้นในโปรแกรมวายฟายดาต้าล็อคกิ้ง

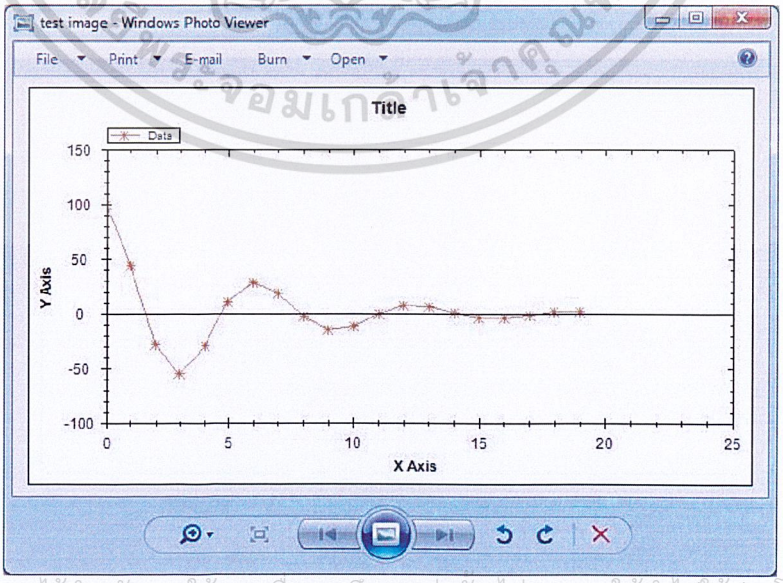
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นทำการทดลองบันทึกข้อมูลที่ถูกแสดงผลจากโปรแกรมวายฟายดาต้าล็อคกิ้งโดยกดปุ่ม เพื่อให้โปรแกรมทำการบันทึกข้อมูลในขณะที่กำลังแสดงผลอยู่โดยสามารถเลือกชนิดของไฟล์ที่จะทำการบันทึกได้สองแบบคือแบบ.PNG และแบบ.TXT ดังรูปที่ 4.8



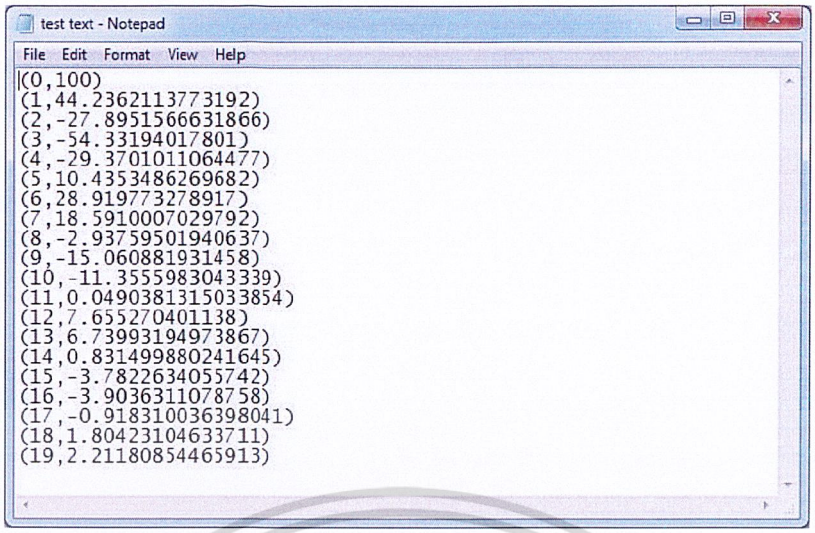
รูปที่ 4.8 หน้าต่าง save dialog

เมื่อทำการบันทึกเรียบร้อยแล้วจะได้ไฟล์ตามที่เลือกชนิดไว้ดังรูปที่ 4.9 และรูปที่ 4.10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

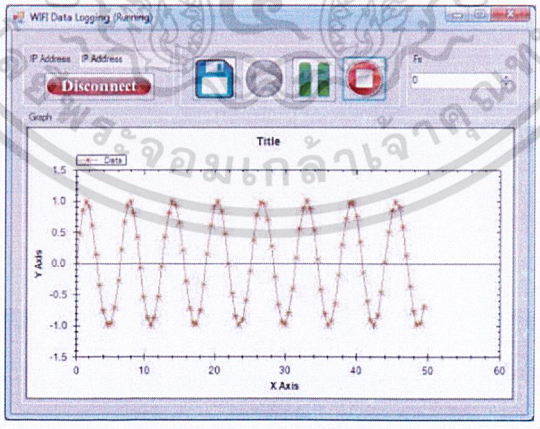
รูปที่ 4.9 ตัวอย่างไฟล์ที่ทำการบันทึกเป็นรูปภาพ



รูปที่ 4.10 ตัวอย่างไฟล์ที่บันทึกเป็นข้อมูล

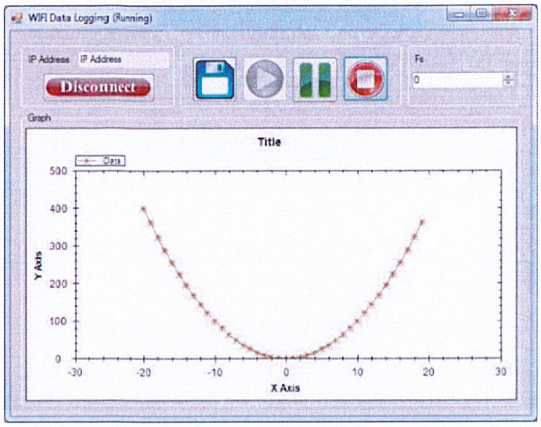
จากการการทดลองดังกล่าวแสดงให้เห็นว่าโปรแกรมวายฟายดาต้าล็อคกิ้งสามารถที่จะแสดงผลข้อมูลที่ถูกส่งมาจากโปรแกรมวายฟายพล็อต โดยผ่านสัญญาณไร้สายในระบบวายฟายได้

กราฟที่โปรแกรมวายฟายดาต้าล็อคกิ้งสามารถเป็นรูปแบบต่างๆได้ ตามข้อมูลที่ได้รับมาจากโปรแกรมวายฟายพล็อต

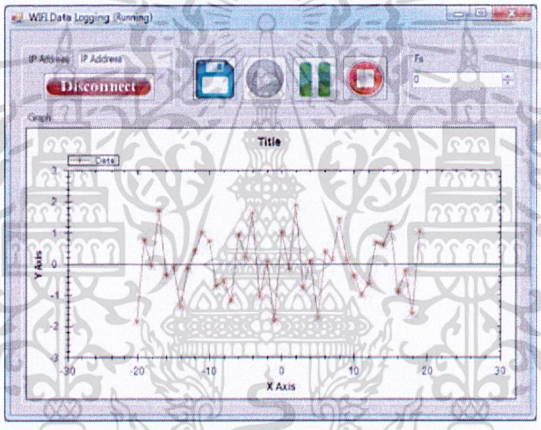


รูปที่ 4.11 ตัวอย่างรูปกราฟซายน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 ตัวอย่างรูปกราฟพาราโบลา



รูปที่ 4.13 ตัวอย่างรูปกราฟแบบสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

การทดลองในโครงงานนี้จะใช้โปรแกรมทั้งหมดสองโปรแกรมด้วยกันโดยโปรแกรมแรกจะเป็นโปรแกรมที่ใช้ในการรับข้อมูลผ่านสัญญาณไร้สายแบบวายฟายคือโปรแกรมชื่อ “วายฟายดาต้าล็อคกิ้ง(WIFI Data Logging)” และในส่วนโปรแกรมที่สองจะเป็นโปรแกรมที่ใช้สำหรับจำลองข้อมูลที่ใช้ในการแสดงผลที่หน้าจอของโปรแกรมวายฟายดาต้าล็อคกิ้งซึ่งก็คือโปรแกรม “วายฟายพล็อต(WIFI Plot)” ผลปรากฏว่าข้อมูลที่ทำการป้อนให้โปรแกรมวายฟายพล็อตเมื่อผ่านการส่งสัญญาณผ่านระบบวายฟายไปที่โปรแกรมวายฟายดาต้าล็อคกิ้งจะถูกแสดงขึ้นมาที่หน้าจอของโปรแกรมโดยการแสดงผลจะเป็นแบบ “Real Time” ซึ่งเราสามารถที่จะสั่งให้หยุดการแสดงผลชั่วคราวเพื่อสะดวกในการอ่านค่า นอกจากนี้ยังสามารถที่จะทำการบันทึกข้อมูลในรูปแบบของไฟล์ .PNG และรูปแบบไฟล์ .TXT เพื่อสะดวกต่อการนำไปใช้ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] บัญชา ปะสีละเตสัง. การพัฒนาแอปพลิเคชั่น Visual C#. กรุงเทพฯ: ซีเอ็ดยูเคชั่น. 2552.

[2] Harvey M. Deitel and Paul J. Deitel. Visual C# 2005 : How to Program, Second Edition. Prentice Hall. 2005

[3] Microsoft. "MSDN Library." [Online]. Available : http://msdn.microsoft.com/en-us/library/ms123401.aspx. 2010.

[4] JChampion. "A flexible charting library for .NET." [Online]. Available : http://www.codeproject.com/KB/graphics/zedgraph.aspx. 2007.

[5] Sourceforge. "ZedGraph Class Library Documentation." [Online]. Available : http://zedgraph.sourceforge.net/documentation/default.html. 2010.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้