

หุ่นยนต์สำรวจไร้สาย

WIRELESS CONTROL SURVEY ROBOT



T119409



เลขหมู่.....**119409**
เลขทะเบียน.....
วัน,เดือน,ปี.....- 7 S.A. 2554

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WIRELESS CONTROL SURVEY ROBOT



**THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONIC ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2010**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม หลักสูตรวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์สำรวจไร้สาย
WIRELESS CONTROL SURVEY ROBOT

ผู้จัดทำ	นายวรุณชัย	โมงใส	50011404
	นายสันต์	ชี้ทางดี	50011419
	นายวิชาธร	รังสิยางกูล	50011455


.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์สำรวจไร้สาย

โดย

นายวรุณชัย	โมงใส	50011404
นายสันต์	ชี้ทางดี	50011419
นายวิชาธร	รังสียางกุล	50011455

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์

ปีการศึกษา 2553

บทคัดย่อ

โครงงานนี้นำเสนอ ทฤษฎีระบบควบคุมแบบไร้สาย สำหรับควบคุมหุ่นยนต์กู้ภัย โดย โครงสร้างประกอบด้วย Computer , Access Point , dsPIC , กล้อง webcam , การควบคุม Servo Motor จุดมุ่งหมายของโครงงานนี้เพื่อสร้างเป็นหุ่นยนต์กู้ภัยเพื่อช่วยผู้ประสบภัยพิบัติต่าง ๆ

ขั้นตอนการดำเนินการเริ่มจากการศึกษาการเชื่อมต่อไร้สาย ศึกษาและออกแบบการเขียน โปรแกรมเพื่อรับและส่งสัญญาณการควบคุมไปยังหุ่นยนต์กู้ภัย ซึ่งโปรแกรมคอมพิวเตอร์ที่ใช้ใน ที่นี้คือ Visual Basic 6 และ C Language เพื่อรับ - ส่งข้อมูลจากคอมพิวเตอร์ผ่าน UART ไปยัง ไมโครคอนโทรลเลอร์และไปควบคุมการทำงานของหุ่นยนต์ให้เป็นไปตามต้องการ และมีการ ออกแบบโครงสร้างตัวหุ่นยนต์ ศึกษาและออกแบบวงจรขับ Servo Motor

WIRELESS CONTROL SURVEY ROBOT

By

Mr. Warunchai Mongsai

Mr. Wasan Cheetangdee

Mr. Wichatorn Rungsiyangkul

Advisor

Asst. Prof. Sumit

Panaudomsub

Academic Year 2010

Abstract

This project presents theory and implementation procedures of wireless control system for control the rescue robot. Structures are including computer, access point, dsPIC, webcam and controlling Servo motor. Purpose if for create rescue robot for help a patient in danger situation.

Starting with we study a wireless connection. We study about writing program for send signal control a robot. In this term we study Visual Basic 6 and C Language for transfer data from computer to UART and control robot what we want.

The result wireless control can send control signal to a robot for absolutely work.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำโครงการนี้มีจุดมุ่งหมายคือสร้างเป็นหุ่นยนต์สำรวจไร้สาย โดยมีการส่งหุ่นยนต์เข้าไปสำรวจพื้นที่แล้วมีการส่งภาพกลับมาเพื่อให้ผู้ที่คอนโทรลทราบว่าในบริเวณที่สำรวจมีวัตถุอะไรอยู่ หรือ ลักษณะของพื้นที่เป็นเช่นไร เพื่อความปลอดภัย โดยที่ไม่ต้องส่งคนเข้าไปสำรวจพื้นที่โดยตรง

ทางผู้จัดทำขอขอบคุณ อาจารย์สุมิตร พนาอุดมทรัพย์ ที่ได้กรุณาให้คำปรึกษาอย่างดีโดยตลอดตั้งแต่ต้น

การทำโครงการ เรื่อง หุ่นยนต์สำรวจไร้สาย (Wireless Control Survey Robot)นี้ ผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์แก่ผู้ที่สนใจ ซึ่งจะได้พบกับเทคโนโลยีใหม่ๆ มากมาย ซึ่งหุ่นยนต์ที่สร้างนี้อาจจะไม่สามารถใช้งานได้จริง แต่จะเป็นประโยชน์ในด้านการพัฒนาต่อไป

ขอบคุณห้องวิจัยปัญญาโท ของ ดร.นพดล มณีรัตน์ ที่เอื้อเฟื้อสถานที่ และให้ยืมอุปกรณ์ที่ขาดเหลืออยู่เสมอๆ ตลอดจนพี่ๆ เพื่อนๆ ที่คอยให้กำลังใจ และ คำปรึกษาเป็นอย่างดี

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

วรุณชัย

วสันต์

วิชาธร

โม่งใส

จี๋ทางดี

รังสิยางกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 ความเป็นมาของโปรเจ็ค	1
1.3 วัตถุประสงค์ในการทำปริยานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการพื้นฐาน	3
2.1 การเชื่อมต่อไร้สาย	3
2.2 Port และ IP Address	8
2.3 โปรแกรม Visual Basic 6.0	12
2.3.1 โปรแกรม Visual Basic ในส่วนการส่งภาพ	14
2.3.2 โปรแกรม Visual Basic ในส่วนการควบคุม	21
2.4 กล้องเว็บแคม(Webcam)	22
2.5 หน่วยประมวลผล dsPIC30F4011	23
2.5.1 การใช้งานโมดูล UART กับพอร์ตอนุกรม	23
2.5.1.1 คุณสมบัติที่สำคัญของโมดูล UART	23
2.5.1.2 รีจิสเตอร์ควบคุมโมดูล UART	25
2.5.1.3 การคำนวณอัตราบอด (Baud Rate)	25
2.5.2 การใช้งานโมดูลเปรียบเทียบข้อมูล	26
2.5.2.1 รีจิสเตอร์ควบคุมโมดูลเปรียบเทียบข้อมูลเพื่อสร้างเอาต์พุต	27
2.5.2.2 โหมดการใช้งานเปรียบเทียบข้อมูลสร้างสัญญาณ PWM	27
2.5.2.1 รีจิสเตอร์ควบคุมโมดูลเปรียบเทียบข้อมูลเพื่อสร้างเอาต์พุต	27
2.5.2.2 โหมดการใช้งานเปรียบเทียบข้อมูลสร้างสัญญาณ PWM	27
2.5.2.3 การคำนวณหาค่า PWM สำหรับเวลา 1 มิลลิวินาที	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5.2.1 รีจิสเตอร์ควบคุมโมดูลเปรียบเทียบข้อมูลเพื่อสร้างเอาต์พุต.	27
2.5.2.2 โหมดการใช้งานเปรียบเทียบข้อมูลสร้างสัญญาณ PWM	27
2.5.2.3 การคำนวณหาค่า PWM สำหรับเวลา 1 มิลิวินาที	28
2.6 การเชื่อมต่อทางแสง	29
2.7 ดีซีมอเตอร์ (DC Motor)	29
2.8 เซอร์โวมอเตอร์ (SERVO MOTOR)	30
บทที่ 3 การออกแบบและการทำงาน	33
3.1 ภาพรวมโครงการ	33
3.1.1 ส่วนการออกแบบ	33
3.1.2 ส่วนควบคุมและสั่งงาน	34
3.1.3 ส่วนปฏิบัติงาน	34
3.2 การเชื่อมต่อไร้สาย	34
3.3 ส่วนของการมองของหุ่นยนต์	34
3.4 ส่วนโปรแกรมควบคุม	35
3.4.1 การออกแบบโปรแกรมในส่วน dsPIC30F4011	35
3.4.2 การออกแบบโปรแกรมในส่วน Visual Basic 6	36
บทที่ 4 การทดลอง	37
4.1 การทดลองในส่วนของโปรแกรม Visual Basic	37
4.1.1 โปรแกรม Visual Basic ในส่วนของการรับ ส่งภาพ	37
4.1.2 โปรแกรม Visual Basic ในส่วนการควบคุม	41
4.2 การทดลองในส่วนการควบคุมการทำงานของหุ่นยนต์ทั้งหมด	44
บทที่ 5 การทดลอง	48
5.1 ปัญหาที่พบ	48
5.2 วิธีแก้ไขปัญหา	48
ภาคผนวก	49
ภาคผนวก ก Code Program Visual Basic และ Code dsPIC30F4011	50
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	69
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ รูป

	หน้า
รูปที่ 2.1 Wireless Access Point	5
รูปที่ 2.2 แสดงระบบ Sever-Client	6
รูปที่ 2.3 หน้าต่าง cmd	7
รูปที่ 2.4 มีการตอบสนองคำสั่งกลับมา	7
รูปที่ 2.5 หน้าต่าง Local Area Connection Status กติ Properties	10
รูปที่ 2.6 คับเบิลคลิกที่ Internet Protocol Version4 (TCP/IPv4)	10
รูปที่ 2.7 กำหนด IP Address.	11
รูปที่ 2.8 เลือก Change	11
รูปที่ 2.9 ทำการเปลี่ยนชื่อ Workgroup	12
รูปที่ 2.10 ลักษณะของโปรแกรม Visual Basic	13
รูปที่ 2.11 แสดงโปรแกรมStream Video ในส่วนของ Server	14
รูปที่ 2.12 แสดงโปรแกรมStream Video ในส่วนของ Client	16
รูปที่ 2.13 การกำหนดให้เป็น Dynamic IP Address	18
รูปที่ 2.14 หน้าต่างของโปรแกรม Control	21
รูปที่ 2.15 กล้องเว็บแคม	22
รูปที่ 2.16 บล็อกไดอะแกรมการส่งข้อมูลของโมดูล UART	24
รูปที่ 2.17 บล็อกไดอะแกรมการรับข้อมูลของโมดูล UART	24
รูปที่ 2.18 ไดอะแกรม โมดูลเปรียบเทียบข้อมูลเอาต์พุต	26
รูปที่ 2.19 แสดงลักษณะรูปร่างของเซอร์โวมอเตอร์	30
รูปที่ 2.20 แสดงการต่อสายกับเซอร์โวมอเตอร์	31
รูปที่ 2.21 แสดงความสามารถในการหมุนของเซอร์โวมอเตอร์	31
รูปที่ 2.22 ไดอะแกรมเวลาของสัญญาณพัลส์ที่ใช้สำหรับขับเซอร์โวมอเตอร์	32
รูปที่ 2.23 แสดงการป้อนพัลส์ช่วงเวลาต่างกัน	32
รูปที่ 3.1 ส่วนของตัวรถและกล้อง webcam	33
รูปที่ 3.2 ส่วนมุมมองของหุ่นยนต์	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา VI และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ รูป (ต่อ)

	หน้า
รูปที่ 4.1 ภาพแสดงขั้นตอนที่ 1	37
รูปที่ 4.2 ภาพแสดงขั้นตอนที่ 2	38
รูปที่ 4.3 ภาพแสดงขั้นตอนที่ 3	38
รูปที่ 4.4 ภาพแสดงขั้นตอนที่ 4	38
รูปที่ 4.5 ภาพแสดงขั้นตอนที่ 5	39
รูปที่ 4.6 ภาพแสดงขั้นตอนที่ 7	39
รูปที่ 4.7 ภาพแสดงขั้นตอนที่ 8	40
รูปที่ 4.8 ภาพแสดงขั้นตอนที่ 9	40
รูปที่ 4.9 แสดงขั้นตอนที่ 1	41
รูปที่ 4.10 แสดงขั้นตอนที่ 2	41
รูปที่ 4.11 แสดงขั้นตอนที่ 3	42
รูปที่ 4.12 แสดงขั้นตอนที่ 4	42
รูปที่ 4.13 แสดงขั้นตอนที่ 5	43
รูปที่ 4.14 แสดงขั้นตอนที่ 6	43
รูปที่ 4.15 แสดงขั้นตอนที่ 7	44
รูปที่ 4.16 แสดงการเชื่อมต่อระหว่างบอร์ด dsPIC และวงจรขับมอเตอร์กระแสตรง	44
รูปที่ 4.17 แสดงการเชื่อมระหว่างบอร์ด dsPIC และ Computer Notebook ผ่านทาง RS-232	45
รูปที่ 4.18 แสดงการเชื่อมระหว่างบอร์ด dsPIC และ Computer Notebook เมื่อทำการประกอบเข้ากับตัวรถ	45
รูปที่ 4.19 แสดงการเชื่อมระหว่างกล้อง Webcam และ Computer Notebook	46
รูปที่ 4.20 หุ่นยนต์รอรับการสั่งงานจากเครื่อง Client ผ่านทาง Access Point	46
รูปที่ 4.21 แสดงการสั่งงานให้ตัวรถเลี้ยวซ้าย และหมุนกล้องซ้าย	47
รูปที่ 4.22 แสดงการสั่งงานให้ตัวรถเลี้ยวขวา และหมุนกล้องขวา	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ ตาราง

	หน้า
ตารางที่ 2.1 เปรียบเทียบความแตกต่างระหว่างเครือข่ายมีสายกับเครือข่ายไร้สาย	4
ตารางที่ 2.2 Class ของแต่ละ IP Address	9
ตารางที่ 2.3 ชนิดของแอปพลิเคชัน	13



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

การศึกษาในสาขาวิชาวิศวกรรมศาสตร์แมคคาทรอนิกส์เป็นการศึกษาและประยุกต์ทฤษฎีต่างๆ เพื่อการออกแบบและควบคุมระบบให้มีเสถียรภาพและมีสมรรถนะตามความต้องการ หรือให้เป็นไปตามข้อกำหนด ดังนั้น เพื่อให้สอดคล้องกับวัตถุประสงค์นี้จึงจำเป็นต้องมีการศึกษาการควบคุมระยะไกลขึ้นมา เพื่อควบคุมระบบทางไกล อีกทั้งการศึกษาวจรอิเล็กทรอนิกส์ รวมถึงการเขียน โปรแกรมคอมพิวเตอร์ การศึกษาและการเลือกอุปกรณ์วัดและแปลงสัญญาณตลอดจนบูรณาการเรื่องที่ศึกษาเหล่านี้ในการประยุกต์ใช้กับระบบควบคุมทางกายภาพจริง ซึ่งนับเป็นสิ่งจำเป็นในการศึกษาสาขาวิชานี้

ในโครงการนี้จึงเลือกที่จะศึกษาระบบควบคุมไร้สาย ซึ่งเป็นตัวอย่างหนึ่งของระบบควบคุมและน่าสนใจในการศึกษา โดยจะทำการประยุกต์การควบคุมไร้สายเข้ากับ การจำลองการทำงานของหุ่นยนต์กู้ภัย เพราะเป็นการผสมผสานความรู้ในหลายๆด้าน นอกจากจะได้รับความรู้เรื่อง Software และ Hardware แล้วยังได้ความรู้ทางด้านแมคคาทรอนิกส์ด้วย

1.2 ความเป็นมาของโปรเจค

เนื่องจากในปัจจุบันเทคโนโลยีต่าง ๆ มีบทบาทต่อการพัฒนาสังคมหรือแม้แต่การพัฒนาประเทศ ด้วยเหตุผลที่ว่าเทคโนโลยีเป็นพื้นฐานสำคัญอย่างหนึ่งในการพัฒนาทางด้านต่าง ๆ ไม่ว่าจะเป็น ด้านการแพทย์ , ด้านการศึกษา , ด้านการสื่อสาร หรือ แม้กระทั่งระบบรักษาความปลอดภัย เป็นต้น ซึ่งทางผู้จัดทำมองว่าเทคโนโลยีที่กำลังได้รับความสนใจในเวลานี้คือ หุ่นยนต์ ซึ่งจะเห็นได้ว่า หุ่นยนต์เป็นสิ่งที่ได้รับความสนใจจากคนทุกเพศทุกวัย และปัจจุบันได้มีการพัฒนา , ออกแบบหุ่นยนต์มาหลากหลายประเภท ซึ่งประโยชน์ของหุ่นยนต์จะขึ้นอยู่กับประเภทของหุ่นยนต์และลักษณะการใช้งาน โดยในที่นี้ผู้จัดทำได้เล็งเห็นประโยชน์และความสำคัญของหุ่นยนต์กู้ภัย จึงได้มีการศึกษาและจัดทำ project เรื่อง Wireless Control Survey Robot ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจุดประสงค์หลักของการทำ project เรื่อง Wireless Control Survey Robot นี้ ก็คือ มีการส่งหุ่นยนต์เข้าไปสำรวจพื้นที่ แล้วมีการส่งภาพกลับมาเพื่อให้ผู้ที่คอนโทรลทราบว่าในบริเวณที่สำรวจมีวัตถุอะไรอยู่ หรือ ลักษณะของพื้นที่เป็นเช่นไร เพื่อความปลอดภัย โดยที่ไม่ต้องส่งคนเข้าไปสำรวจพื้นที่โดยตรง ในการทำ project เรื่อง Wireless Control Survey Robot นี้ ผู้จัดทำหวังว่า project นี้จะเป็นประโยชน์แก่ผู้ที่สนใจ ซึ่งจะได้พบกับเทคโนโลยีใหม่ ๆ มากมาย ซึ่งหุ่นยนต์ที่สร้างนี้อาจจะไม่สามารถใช้งานได้จริง แต่จะเป็นประโยชน์ในด้านการพัฒนาต่อไป

1.3 วัตถุประสงค์ในการทำปริญานิพนธ์

1. ทำการศึกษาระบบควบคุมไร้สาย (Wireless LAN Control) โดยประยุกต์เข้ากับการจำลองการทำงานของหุ่นยนต์กู้ภัย ซึ่งประกอบด้วยคอมพิวเตอร์ โนตบุ๊ก จำนวนสองเครื่อง วงจรส่วนประมวลผล dsPIC เซอร์โวมอเตอร์ (Servo Motor) และอุปกรณ์จับภาพด้วยกล้องเว็บแคม (Webcam) โดยใช้โปรแกรม Visual Basic 6 และ ภาษาซี
2. ทำการศึกษาและออกแบบตัวหุ่น เพื่อนำไปใช้ในสถานการณ์จริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการพื้นฐาน

2.1 การเชื่อมต่อไร้สาย

ระบบเครือข่ายไร้สาย (WLAN = Wireless Local Area Network) คือ ระบบการสื่อสารข้อมูลที่นำมาใช้ทดแทนหรือเพิ่มต่อกับระบบเครือข่ายแลนไร้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และ คลื่นอินฟราเรด ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่องผ่านอากาศ , ทะลุกำแพง , เพดาน หรือ สิ่งก่อสร้างอื่น ๆ โดยปราศจากความต้องการของการเดินสาย นอกจากนี้ระบบเครือข่ายไร้สายก็มีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบ LAN แบบไร้สาย ที่สำคัญก็คือ การที่ต้องใช้สายทำให้การเคลื่อนย้าย การใช้งานทำได้โดยสะดวก ไม่เหมือนระบบ LAN แบบใช้สาย ที่ต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่งการใช้งานเครื่องคอมพิวเตอร์

การรับส่งข้อมูลของ Wireless มีลักษณะเป็น การรับ – ส่งข้อมูลสองทางแบบสมบูรณ์ (Full Duplex) ในส่วนของการรับส่งข้อมูลแบบฮาร์ฟดูเพล็กซ์จะสามารถทำให้สามารถส่งข้อมูลกลับไปกลับมาได้ แต่มีปัญหา คือ การพูดคุยวิธีนี้ไม่เป็นธรรมชาติ ซึ่งจะต้องคอยจนกว่าจะมีผู้หนึ่งพูดจบก่อนจึงจะส่งสัญญาณได้ แต่การสื่อสารทั่ว ๆ ไปจะมีการพูดแทรกขึ้น ซึ่งวิธีเดิมไม่สามารถทำได้ จึงมีการปรับปรุงระบบสื่อสารให้สามารถรับและส่งได้พร้อม ๆ กัน เรียกวิธีนี้ว่า “ฟูลดูเพล็กซ์ (Full Duplex)” วิธีนี้จะต้องใช้คลื่นความถี่สองช่องด้วยกัน โดยแต่ละช่องใช้สำหรับการส่งข้อมูลไปยังอีกฝั่งหนึ่ง ข้อดีของวิธีนี้ทำให้การสื่อสารเป็นธรรมชาติมากขึ้น สามารถตอบโต้กันได้ทันที แต่มีข้อเสีย คือ เปลืองช่องสัญญาณ ซึ่งต้องใช้ถึง 2 ช่องสัญญาณด้วยกัน

ตัวอย่างที่เห็นได้ชัดเจนของการสื่อสารวิธีนี้คือ โทรศัพท์บ้าน และ โทรศัพท์มือถือ ที่ผู้พูดสามารถสนทนาสวนกันไปมาได้ทันที ไม่ต้องรอให้อีกฝ่ายหนึ่งพูดจบก่อน

ข้อแตกต่างระหว่างเครือข่ายมีสายกับเครือข่ายไร้สาย

ในอดีตการพัฒนาาระบบสื่อสารโทรคมนาคมนั้นมีการพัฒนาทั้งทางด้านการสื่อสารที่ใช้สายนำสัญญาณ เป็นสื่อกลาง วิธีนี้จะควบคุมการกระจายสัญญาณได้ดีกว่า ซึ่งสามารถทราบได้ว่าสัญญาณจะเดินทางไปได้ไกลเพียงใด และจะควบคุมไม่ให้เกิดการรบกวนกันระหว่างสายที่ง่ายกว่า การเพิ่มประสิทธิภาพระบบสื่อสารก็ทำได้ง่ายเพียงเพิ่มสายไปเท่านั้นเอง ต่างจากระบบไร้สายที่ใช้คลื่นความถี่วิทยุกว่าจะเดินทางมาถึงปลายทางนั้นก็อาจจะผ่านอากาศหรือสะท้อนผ่านช่องเพดาน นั่นก็เป็นตัวกลางหลายชนิดที่มีผลกระทบต่อการใช้สัญญาณนี้ จึงคาดเดาได้ยากว่า สัญญาณจะเดินทางได้ไกลเพียงใดและจะถูกรบกวนหรือไม่ และนอกจากนี้การใช้ตัวกลางร่วมกันจะเกิดปัญหารบกวนกันระหว่างคลื่นความถี่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายมีสาย (Wired)	เครือข่ายไร้สาย (Wireless)
ไม่ต้องแชร์ความถี่กันใช้ สายนำสัญญาณแต่ละเส้นสามารถใช้ช่องสัญญาณความถี่เดียวกันได้	ต้องแชร์กันใช้ช่องสัญญาณคลื่นความถี่เดียวกัน
มีอัตราการลดทอนสัญญาณต่ำ นั่นหมายความว่าส่งสัญญาณได้ไกล	มีอัตราการลดทอนสัญญาณสูง นั่นหมายความว่าส่งสัญญาณได้ระยะสั้นๆ
มีสัญญาณรบกวนต่ำ	มีสัญญาณรบกวนสูง

ตารางที่ 2.1 เปรียบเทียบความแตกต่างระหว่างเครือข่ายมีสายกับเครือข่ายไร้สาย

สาเหตุที่ต้องใช้เครือข่ายไร้สายคือ เครือข่ายแบบนี้มีความคล่องตัวในการทำงานค่อนข้างสูง เนื่องจากไม่ต้องเสียเวลาเดินสาย และร้อยสาย ทำให้สิ้นเปลืองค่าใช้จ่าย แต่ถ้านำเอาเครือข่ายไร้สายมาใช้ การวางก็เพียงติดตั้งรับ-ส่งสัญญาณทุกจุดที่ต้องการใช้งาน จากนั้นก็เริ่มใช้งานได้ทันที นั่นก็คือ สาเหตุต่างๆว่าทำไมเครือข่ายไร้สายในปัจจุบันถึงได้รับความนิยม

เครือข่ายไร้สายตามมาตรฐาน 802.11

เครือข่าย Wireless LAN ที่ใช้งานกันส่วนใหญ่ถูกพัฒนาขึ้นมาตามมาตรฐาน 802.11 สาเหตุที่มาตรฐานนี้ถูกใช้กันมากแล้วกลายเป็นอุปกรณ์ที่ใช้กันมาก ก็เนื่องจากการใช้คลื่นวิทยุที่มีกำลังส่งสูง พอที่จะทะลุทะลวงสิ่งกีดขวางได้ดี สามารถใช้งานในร่มได้ไกลถึง 100 เมตร และกลางแจ้งถึง 400 เมตร (นั่นเป็นระยะมาตรฐานแต่การใช้งานในทางปฏิบัติจะน้อยกว่านี้เกือบ 3 เท่า) และมีความเร็วในการทำงานสูงถึง 11-54 เมกะบิตต่อวินาที ซึ่งเร็วพอที่จะนำมาใช้แทนระบบเครือข่ายแลนตัวเดิมได้ แต่เครือข่าย Wireless LAN นี้มีด้วยกัน 4 มาตรฐาน คือ a, b, g และ n ซึ่งแต่ละตัวมีวิธีการ โมดูลสัญญาณต่างกัน และใช้ความถี่ต่างกัน จึงมีประสิทธิภาพที่ต่างกันด้วย

ในการทำโครงการเรื่อง Wireless Control นี้ ผู้จัดทำได้เลือกใช้เครือข่ายไร้สายตามมาตรฐาน 802.11g (Access Point g) ซึ่งเป็นมาตรฐานที่มีข้อดี คือ มีความเร็วในการทำงานสูงถึง 54 เมกะบิต และมีระยะทำงานไกล นอกจากนี้ยังใช้ความถี่ 2.4 กิกะเฮิรตซ์ ซึ่งเป็นช่วงความถี่ที่ใช้งานได้เสรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ใช้งานเครือข่าย (Wireless Access Point)

Wireless Access Point เป็นอุปกรณ์สำคัญอีกชิ้นหนึ่งบนเครือข่าย Wireless LAN ทำหน้าที่เสมือนฮับ เชื่อมเครื่องคอมพิวเตอร์ไร้สายและอุปกรณ์ Wireless LAN ต่างๆเข้าด้วยกันเป็นเครือข่าย อีกทั้งเป็นสะพาน เชื่อมต่อเครือข่าย Wireless LAN เข้ากับเครือข่ายอีเทอร์เน็ต ทำให้อุปกรณ์บนระบบทั้งสองสามารถสื่อสาร ข้อมูลถึงกันได้

ลักษณะทางกายภาพของ Access Point ประกอบด้วยสายอากาศแบบซ่อนอยู่ภายในหรือบางยี่ห้อ ก็จะเป็นแบบติดตั้งไว้ภายนอก สามารถถอดเปลี่ยนเป็นสายอากาศเกนสูงเพื่อเพิ่มกำลังรับ-ส่งคลื่นวิทยุให้ครอบคลุม พื้นที่ที่กว้างไกลมากขึ้น ส่วนด้านหลังของ Access Point จะมีพอร์ต RJ-45 หรือที่เรียกว่า “แลนพอร์ต” ซึ่งใช้เป็นจุดต่อสายสัญญาณ UTP ไปเข้าฮับหรือสวิตช์ของเครือข่ายอีเทอร์เน็ต



รูปที่ 2.1 Wireless Access Point

Access Point ทำหน้าที่เสมือนฮับใช้เป็นศูนย์กลางการเชื่อมโยงอุปกรณ์ Wireless LAN ต่างๆเข้าด้วยกันจำนวนของอุปกรณ์ที่ Access Point เครื่องหนึ่งๆสามารถรองรับได้ก็จะประมาณ 16, 32, 64 หรือ 128 อุปกรณ์ (แล้วแต่รุ่นและยี่ห้อของ Access Point) ดังนั้นในการเลือกซื้อ Access Point ควรตรวจสอบจำนวนในการรองรับการเชื่อมโยงด้วย

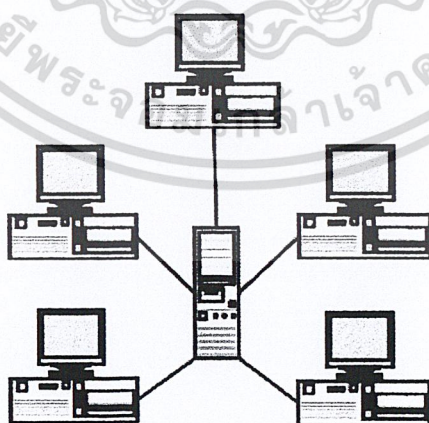
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนการเชื่อมต่อไร้สายนี้ทางผู้จัดทำได้ทำการทดลองเชื่อมต่อ, ถ่ายโอนข้อมูลระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง ผ่านทาง Access Point ซึ่งจุดประสงค์ในการใช้การเชื่อมต่อไร้สาย (Wireless Local Area Network) ก็คือให้เครื่องคอมพิวเตอร์ 2 เครื่อง (เครื่อง Server และ เครื่อง Client) สามารถเชื่อมต่อกัน และสามารถส่งผ่านภาพที่เป็น VDO จากเครื่อง Server มายังเครื่อง Client ผู้จัดทำได้นำกล้องเว็บแคมเชื่อมต่อทาง port USB กับเครื่อง Server โดยที่กล้อง Webcam ทำหน้าที่รับภาพ แล้วแสดงผลของภาพบนเครื่อง Server ก่อนที่จะมีการส่งภาพจากเครื่อง Server มายังเครื่อง Client โดยมี Access Point เป็นอุปกรณ์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์ทั้ง 2 เครื่อง ซึ่งระยะในการเชื่อมต่อผ่าน Access Point (G) มีระยะในการส่งข้อมูลประมาณ 150 เมตร ไม่ว่าจะอยู่ที่จุดใด บริเวณไหน ภาพที่ปรากฏบนเครื่อง Client ก็จะเหมือนกับภาพที่กล้อง Webcam รับได้ทุกประการ

การเขียนโปรแกรมผ่านระบบ LAN

โดยทั่วไปการทำงานบนระบบเครือข่าย จะประกอบไปด้วย 2 ส่วนหลักคือ แม่ข่าย (Server) และ ลูกข่าย (Client) โดย Server จะทำหน้าที่เหมือนกองอำนวยการ เป็นผู้จัดให้กับระบบ Client ที่จะเป็นส่วนร้องขอข้อมูลจาก Server โดยส่วนของ Server ในที่นี้จะทำหน้าที่เป็นส่วนเก็บข้อมูล, จัดการบริหารข้อมูลหรือทรัพยากร เพื่อให้ Client สามารถใช้บริการได้

Server และ Client ต่างก็ต้องมีตำแหน่งที่อยู่ (IP Address) และช่องทางการติดต่อ (Port) โดยทั้งสองฝ่ายจะสามารถเชื่อมต่อถึงกันได้จะต้องอยู่ในช่องทางเดียวกัน โดยการกำหนดหมายเลข Port ให้ตรงกัน ทั้งนี้โปรแกรมที่ติดต่อนั้นจะต้องอ้างอิงหมายเลขของ Port ทุกครั้ง เนื่องจากในระบบมีลูกข่ายมากมายที่กำลังติดต่อกับ Server อยู่



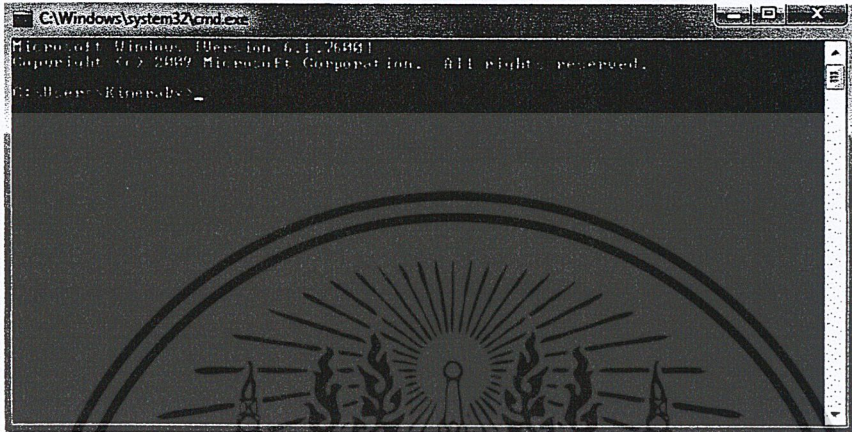
รูปที่ 2.2 แสดงระบบ Sever-Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบการเชื่อมต่อของคอมพิวเตอร์ทั้งสอง

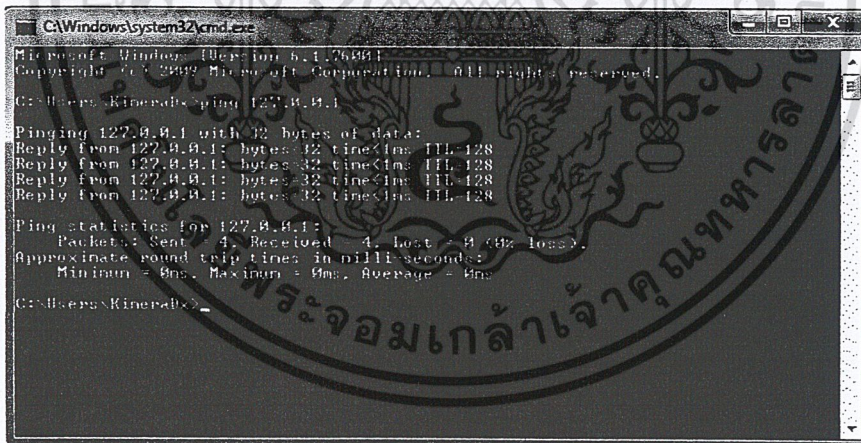
การตรวจสอบคอนเนกชันของคอมพิวเตอร์ใน WorkGroup โดยการใช้คำสั่ง ping เพื่อตรวจสอบการตอบสนองของคอมพิวเตอร์ปลายทางโดยทำตามขั้นตอนดังต่อไปนี้

1. กดปุ่ม Menu Start ในช่อง search พิมพ์ว่า cmd แล้วกด Enter จะได้หน้าต่าง cmd ขึ้นมา



รูปที่ 2.3 หน้าต่าง cmd

2. พิมพ์คำสั่ง ping ตามด้วยหมายเลข IP Address ของคอมพิวเตอร์ปลายทาง ถ้ามีการเชื่อมต่อจะเกิดการตอบสนองเกิดขึ้น



รูปที่ 2.4 มีการตอบสนองคำสั่งกลับมา

เมื่อทำตามขั้นตอนดังกล่าว ก็ถือว่าได้สร้างระบบ Server-Client แบบง่ายด้วยคอมพิวเตอร์ 2 เครื่องซึ่งสามารถนำไปดัดแปลงเขียนโปรแกรมในระบบเครือข่ายได้ต่อไป แต่อย่างไรก็ตามการเขียนโปรแกรมสำหรับทดสอบการทำงานบนระบบเครือข่ายใน Visual Basic เราสามารถใช้คอมพิวเตอร์เครื่องเดียวได้ ซึ่งจะช่วยให้ประหยัดเวลาและลดความยุ่งยากก่อนที่จะนำโปรแกรมไปใช้งานในระบบจริงได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Port และ IP Address

Port

สำหรับ Application ในชั้น layer สูงๆ ที่ใช้ TCP (Transmission Control Protocol) หรือ UDP (User Datagram Protocol) จะมีหมายเลข Port หมายเลขของ Port จะเป็นเลข 16 bit เริ่มตั้งแต่ 0 ถึง 65535 หมายเลข Port ใช้สำหรับตัดสินว่า service ใดที่ต้องการเรียกใช้ ในทางทฤษฎี หมายเลข Port แต่ละหมายเลขถูกเลือกสำหรับ service ใดๆ ขึ้นอยู่กับ OS (operating system) ที่ใช้ ไม่จำเป็นต้องเหมือนกัน แต่ได้มีการกำหนดขึ้นให้ใช้กันอย่างเป็นมาตรฐานเพื่อให้มีการติดต่อการส่งข้อมูลที่ดีขึ้น ทาง Internet Assigned Numbers Authority (IANA) เป็นหน่วยงานกลางในการประสานการเลือกใช้ Port ว่า Port หมายเลขใดควรเหมาะสำหรับ Service ใด และได้กำหนดใน Request For Comments (RFC) 1700 ตัวอย่างเช่น เลือกใช้ TCP Port หมายเลข 23 กับ Service Telnet และเลือกใช้ UDP Port หมายเลข 69 สำหรับ Service Trivial File transfer Protocol (TFTP) ตัวอย่างต่อไปนี้เป็นบางส่วนของ File/etc/services แสดงให้เห็นว่า หมายเลข Port แต่ละหมายเลขได้ถูกจับคู่กับ Transport Protocol หนึ่งหรือสอง Protocol ซึ่งหมายความว่า UPP หรือ TCP อาจจะใช้ หมายเลข Port เดียวกันก็ได้ เนื่องจากเป็น Protocol ที่ต่างกัน

หมายเลข Port ถูกจัดแบ่งเป็น 2 ประเภทดังนี้

1. Well Known Ports คือ จะเป็น Port ที่ระบบส่วนใหญ่ กำหนดให้ใช้โดย Privileged User (ผู้ใช้ที่มีสิทธิพิเศษ) โดย port เหล่านี้ ใช้สำหรับการติดต่อระหว่างเครื่องที่มีระบบเวลาที่ยาวนาน วัตถุประสงค์เพื่อให้ service แก่ผู้ใช้ (ที่ไม่รู้จักหรือคุ้นเคย) แยกหน้า จึงจำเป็นต้องกำหนด Port ติดต่อกันสำหรับ Service นั้นๆ
2. Registered Ports จะเป็น Port หมายเลข 1024 ขึ้นไป ซึ่ง IANA ไม่ได้กำหนดไว้

IP Address

IP Address คือหมายเลขประจำเครื่องคอมพิวเตอร์ ซึ่งประกอบด้วยตัวเลข 4 ชุด มีเครื่องหมายจุดขึ้นระหว่างชุด เช่น 192.168.100.1 หรือ 172.16.10.1 เป็นต้น

มาตรฐานของ IP Address ปัจจุบันเป็นมาตรฐาน version 4 หรือที่เรียกกันสั้น ๆ ว่า IPv4 ซึ่งกำหนดให้ ip address มีทั้งหมด 32 bit หรือ 4 byte แต่ละ byte จะถูกกั้นด้วยจุด (.) ภายในหมายเลขที่เราเห็นยังถูกแบ่งออกเป็น 2 ส่วนดังนี้

1. Network Address หรือ Subnet Address
2. Host Address

บนเครื่อง computer ที่ใช้ TCP/IP Protocol จะมีหมายเลข IP Address กำกับอยู่ address นี้ เป็นอยู่ใน Layer 3 ของ OSI model ซึ่งสามารถเปลี่ยนแปลงได้ตลอดเวลา (Logical address) และบนเครื่อง computer ไม่ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะใช้ Protocol ใด ๆ ก็ตามจะต้องมีหมายเลข ที่เรียกว่า MAC Address ประจำอยู่ที่ Network card เสมอ MAC Address นี้เป็น Hardware Address ที่เปลี่ยนแปลงไม่ได้ เว้นแต่จะเปลี่ยน Network card

Class ของแต่ละ IP Address

A	w.x.y.z	w	x.y.z
B	w.x.y.z	w.x	y.z
C	w.x.y.z	w.x.y	z

ตารางที่ 2.2 Class ของแต่ละ IP Address

สาเหตุที่ต้องแบ่ง IP Address ออกเป็น Class

เพื่อความเป็นระเบียบ ทางองค์กรกลางที่ดูแลเรื่องของ IP Address จึงได้มีการจัด Class หรือ หมวดหมู่ของ IP Address ไว้ทั้งหมด 5 Class โดย Class ของ Address จะเป็นตัวกำหนดว่า Bit ใดบ้างใน หมายเลข IP Address ที่ต้องถูกใช้เพื่อเป็น Network Address และ Bit ใดบ้าง ที่ต้องถูกใช้เป็น Host Address นอกจากนั้น Class ยังเป็นตัวกำหนดด้วยว่าจำนวนของ Network Segment ที่มีได้ใน Class นั้น ๆ มีเท่าไร และจำนวนของเครื่องคอมพิวเตอร์ที่สามารถมีได้ ภายใน Network Segment นั้น ๆ มีเท่าไร

Class D

Class นี้จะไม่ถูกนำมาใช้กำหนดให้กับเครื่องคอมพิวเตอร์ทั่วไป แต่จะถูกใช้สำหรับการส่งข้อมูลแบบ

Multicast ของบาง Application

Multicast

เป็นการส่งจากเครื่องต้นทางหนึ่งไปยังกลุ่มของเครื่องปลายทางอีกกลุ่มหนึ่ง แต่ไม่ใช่ ทุกเครื่องใน Network Segment นั้น ๆ

Class E

Class นี้เป็น Address ที่ถูกสงวนไว้ก่อน ยังไม่ถูกใช้งานจริง ๆ

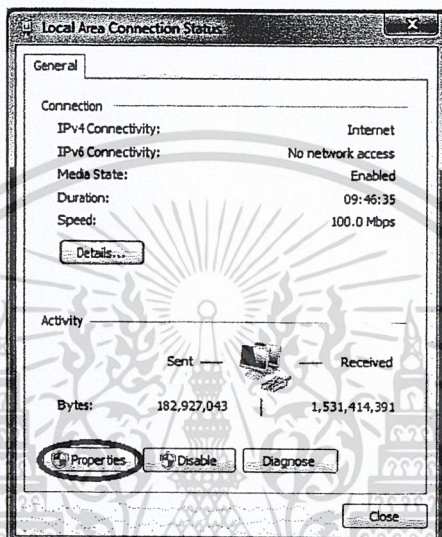
- ถ้า Byte แรก ซ้ายสุดเป็น ตัวเลข 1-126 แสดงว่าเป็นหมายเลข IP Address ที่อยู่ใน Class A (IP address 127 นั้น จะเป็น Loopback Address ของ Class นี้ นะครับ หรือ ของคอมพิวเตอร์ตัวเอง)
- ถ้า Byte แรก ซ้ายสุดเป็น ตัวเลข 128-191 แสดงว่าเป็นหมายเลข IP Address ที่อยู่ใน Class B
- ถ้า Byte แรก ซ้ายสุดเป็น ตัวเลข 192-223 แสดงว่าเป็นหมายเลข IP Address ที่อยู่ใน Class C
- ส่วน 224 ขึ้นไปจะเป็น Multicast Address ที่กล่าวไว้ข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดค่า IP Address

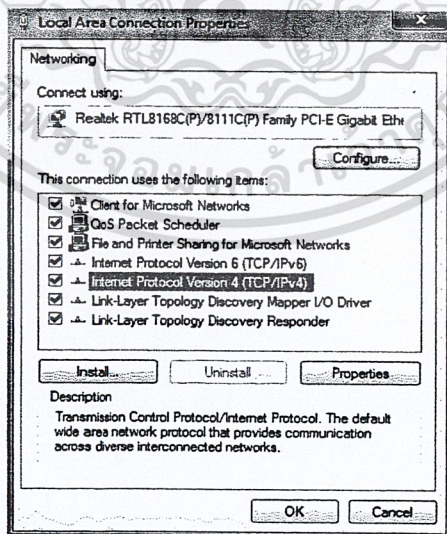
เนื่องจากจะใช้ IP Address สำหรับการติดต่อระหว่างคอมพิวเตอร์ 2 เครื่อง ดังนั้นจำเป็นต้องกำหนด IP Address ของคอมพิวเตอร์ทั้ง 2 เครื่องเสียก่อน โดยทำตามขั้นตอนดังนี้

1. ไปที่ Control Panel แล้วเลือกหัวข้อ Network and Internet หลังจากนั้นเลือก Networking and Sharing Center แล้วกดที่ Connection แล้วเลือก Properties



รูปที่ 2.5 หน้าต่าง Local Area Connection Status กด Properties

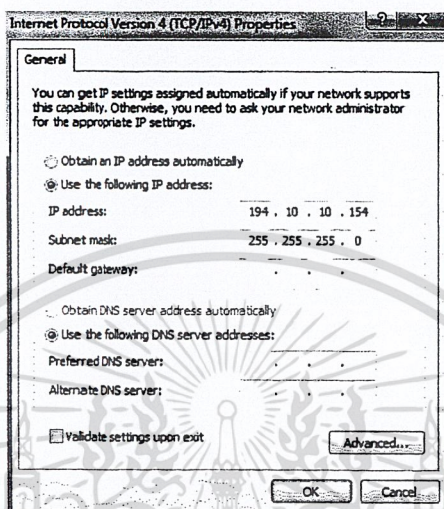
2. ดับเบิลคลิกที่ Internet Protocol Version4 (TCP/IPv4)



รูปที่ 2.6 ดับเบิลคลิกที่ Internet Protocol Version4 (TCP/IPv4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. กำหนด IP Address ให้ถูกต้อง โดยกำหนดรูปแบบมาตรฐานในช่อง IP Address และต้องกำหนดให้ เลข 3 หลักสุดท้ายอยู่ระหว่าง 0 ถึง 255 นอกจากนี้หมายเลขในกลุ่มควรสอดคล้องกันด้วย เช่น กำหนด IP ของ Server ให้เป็น 194.10.10.154 ดังนั้นควรกำหนดให้เครื่อง Client เป็น 194.10.10.155 เป็นต้น

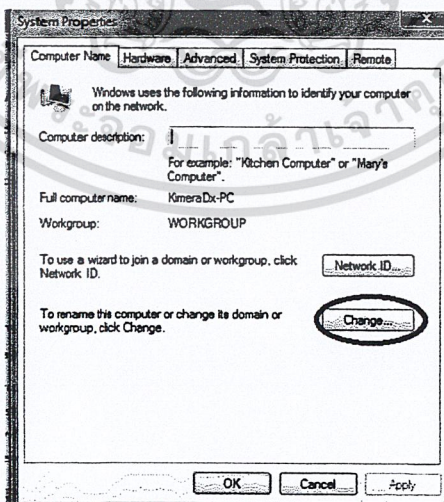


รูปที่ 2.7 กำหนด IP Address

การกำหนด WorkGroup ของคอมพิวเตอร์

จะต้องกำหนด WorkGroup ให้กับกลุ่มของคอมพิวเตอร์ที่เราเชื่อมต่อเข้าด้วยกัน โดยทำตามขั้นตอนดังต่อไปนี้

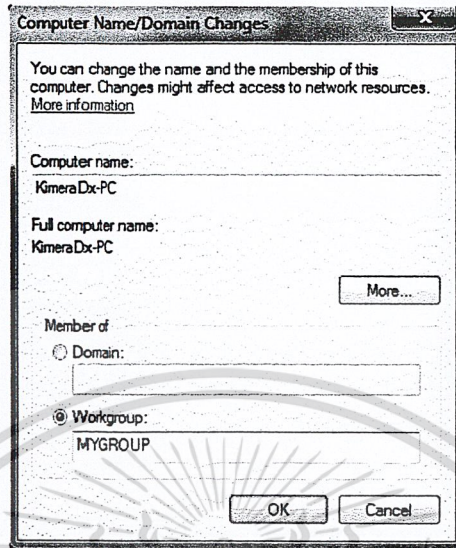
1. คลิกขวาที่ My Computer เลือก Properties แล้วเลือก Advance system settings ไปที่ Tab ที่เขียนว่า Computer Name แล้วกดที่ปุ่ม Change



รูปที่ 2.8 เลือก Change

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เปลี่ยนชื่อ WorkGroup ตามที่ต้องการ โดยจะต้องตั้งชื่อของ WorkGroup นี้ ให้ตรงกันทั้ง 2 เครื่อง



รูปที่ 2.9 ทำการเปลี่ยนชื่อ Workgroup

3. กดปุ่ม OK แล้วทำการรีสตาร์ทคอมพิวเตอร์ใหม่เมื่อเราเปลี่ยน WorkGroup และ Restart คอมพิวเตอร์ทั้ง 2 เครื่องแล้ว เราจะได้ WorkGroup ใหม่ที่มีคอมพิวเตอร์ทั้ง 2 เครื่อง อยู่ในกลุ่ม

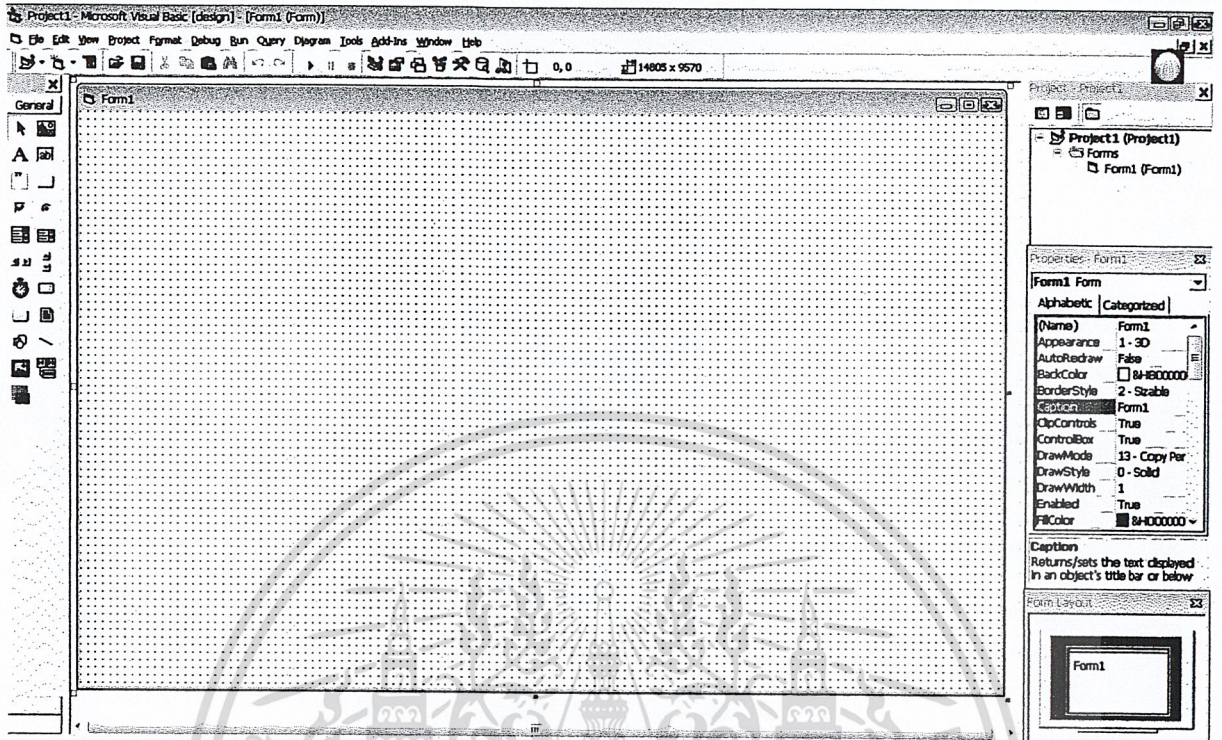
2.3 โปรแกรม Visual Basic 6.0

ในโครงการนี้ ทางผู้จัดทำได้เลือกใช้โปรแกรม Visual Basic 6.0 เนื่องจากเป็นเครื่องมือ ที่ใช้ในการเขียนโปรแกรม บน Windows ที่ได้รับความนิยมสูง และนอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic 6.0 ยังมีเครื่องมือที่ช่วยในการเขียนโปรแกรมซึ่งทำให้เป็นเรื่องที่ไม่ยุ่งยาก เพราะมีเครื่องมือช่วยที่ไม่ต้องจดจำ ไวยากรณ์ภาษาที่ยุ่งยาก สามารถตรวจสอบได้อัตโนมัติว่าโปรแกรมที่เขียนถูกต้องตามหลักภาษาหรือไม่ มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์ลดลงได้มาก

สำหรับการทำ project เรื่อง Wireless Control Server Robot นี้ ได้มีการนำโปรแกรม Visual Basic มาใช้ประโยชน์ในเรื่องต่าง ๆ เช่น การรับส่งภาพของเครื่อง Server และเครื่อง Client , การคอนโทรล การเคลื่อนที่ของหุ่นยนต์ เป็นต้น



สำหรับโปรแกรม Visual Basic 6 ทั้งหมดจะประกอบไปด้วยเมนูบาร์ , ทูลบาร์ , หน้าต่าง Properties , หน้าต่าง Project Explorer , Form Layout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 2.10 ลักษณะของโปรแกรม Visual Basic

ในแต่ละส่วนของโปรแกรม Visual Basic จะมีหน้าที่แตกต่างกันไป ซึ่งเราจะใช้เครื่องมือเหล่านี้ร่วมกันในการพัฒนาแอปพลิเคชัน เช่น แถบเมนูบาร์ (Menu bar) จะมีคำสั่งต่างๆ ที่ครอบคลุมการทำงานทั้งหมด , แถบทูลบาร์ (Tool bar) จะประกอบไปด้วยปุ่มต่างๆ ที่ใช้เรียกคำสั่งในเมนูใช้งานบ่อยๆ เช่น การเปิดโปรเจกต์ , บันทึกโปรเจกต์ เป็นต้น

ชนิดแอปพลิเคชัน	การใช้งาน
	สำหรับการพัฒนาแอปพลิเคชันที่สามารถรันบนระบบปฏิบัติการ Windows ทั่วไป
	สำหรับพัฒนาแอปพลิเคชันที่สามารถใช้งาน และเชื่อมโยงกับแอปพลิเคชันอื่นๆ ที่สนับสนุนเทคโนโลยี ActiveX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

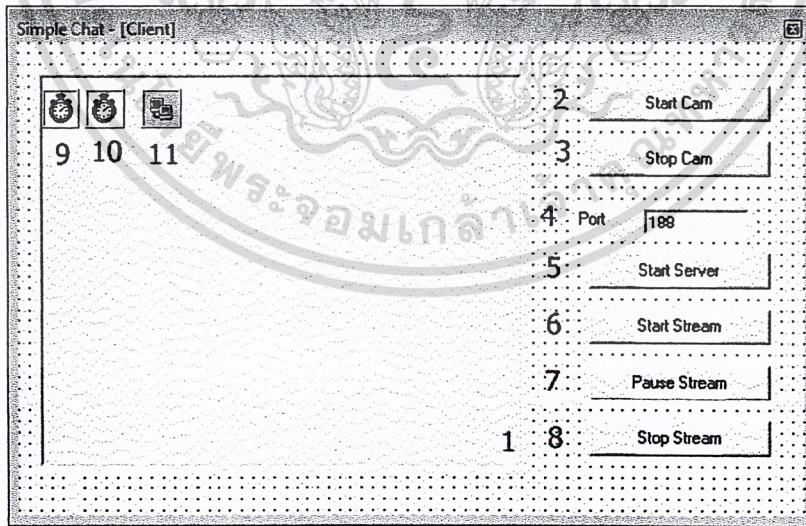
ชนิดแอฟพลิเคชัน	การใช้งาน
 ActiveX DLL	สำหรับการพัฒนาแอฟพลิเคชันเช่นเดียวกับ ActiveX EXE แต่อยู่ในรูปไฟล์ไลบรารี (.dll)
 ActiveX Control	สำหรับพัฒนาคอนโทรลแบบ ActiveX

ตารางที่ 2.3 ชนิดของแอฟพลิเคชัน

2.3.1 โปรแกรม Visual Basic ในส่วนการส่งภาพ

ผู้จัดทำได้ทำการเขียนโปรแกรมสำหรับการ Stream Video โดยใช้โปรแกรม Visual Basic 6 ทำการทำการออกแบบ และ เขียนโปรแกรมออกมา 2 ตัวเพื่อสื่อสารและส่งภาพถ่าย Video ถึงกันคือ Stream Sever และ Stream Client

โปรแกรมตัวที่หนึ่ง Stream Sever โปรแกรมตัวนี้จะทำหน้าที่เป็นเหมือนเครื่องเซิร์ฟเวอร์ในการ รับส่ง Video ถึงกัน ลักษณะและส่วนประกอบของโปรแกรมที่ผู้จัดทำได้ออกแบบไว้จะเป็นลักษณะดังรูปต่อไปนี้



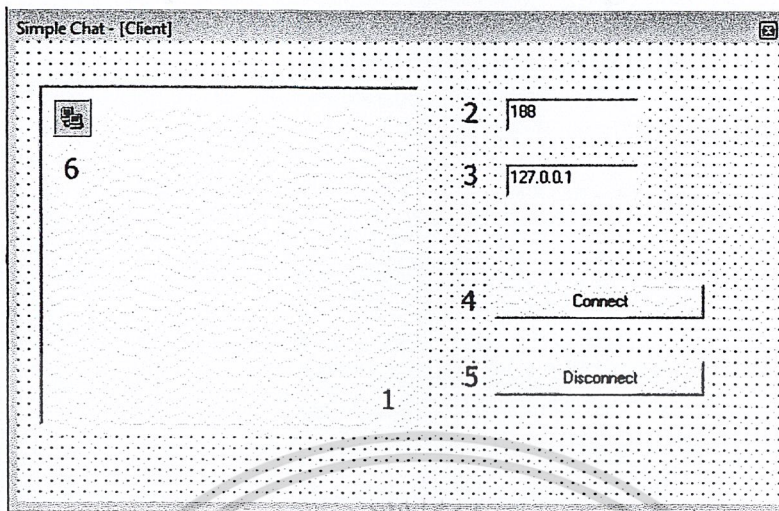
รูปที่ 2.11 แสดงโปรแกรมStream Video ในส่วนของ Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบและหน้าที่ของส่วนต่าง ๆ ของโปรแกรม

1. Picture Box ทำหน้าที่แสดงภาพที่รับมาจากเว็บแคม
2. Command Button: Start Cam เมื่อกดปุ่มนี้ โปรแกรมส่วน Server จะเริ่มรับภาพจากเว็บแคม
3. Command Button : Stop Cam เมื่อกดปุ่มนี้ โปรแกรมส่วน Server จะหยุดรับภาพจากเว็บแคม
4. Port เป็นส่วนที่กำหนดเลข Port ในการเชื่อมต่อระหว่างส่วนของ Server และ Client
5. Command Button: Start Server เมื่อกดปุ่มนี้ โปรแกรมจะทำการเปิด Server เพื่อให้ Client สามารถเชื่อมต่อเข้ามาได้
6. Command Button: Start Stream เมื่อกดปุ่มนี้ โปรแกรมจะเริ่มทำการ Stream Video ที่ได้รับจากเว็บแคมไปยังโปรแกรมของตัว Client ที่มาเชื่อมต่อด้วย
7. Command Button: Pause Stream เมื่อกดปุ่มนี้จะทำการหยุดการ Stream Video ไปยังโปรแกรมส่วน Client ชั่วคราว ซึ่งสามารถกดปุ่มนี้อีกครั้งเพื่อทำการ Stream Video ต่อไปได้เลย
8. Command Button :Stop Stream เมื่อกดปุ่มนี้จะเป็นการหยุดการ Stream Video ไปยังโปรแกรมส่วน Client ทันที หากจะเริ่มการ Stream ต้อง ไปกดที่ Start Stream ใหม่
9. Timer1 เป็น Timer ที่ทำหน้าที่เกี่ยวข้องกับการเชื่อมต่อระหว่างโปรแกรมส่วน Server กับกล้องเว็บแคม
10. Timer2 เป็น Timer ที่ทำหน้าที่เกี่ยวข้องกับการเชื่อมต่อต่างๆระหว่าง โปรแกรม Server และ Client
11. Winsock เป็นคอนโทรลเลอร์ตัวหนึ่งของ Visual Basic ที่ช่วยในการรับ-ส่ง ผ่านข้อมูลระหว่างเครือข่าย

โปรแกรมตัวที่สอง Client ทำหน้าที่ในการรับ Video ที่Stream มาจากตัว Server โดยการติดต่อสื่อสารส่งข้อมูลนี้ ทำโดยใช้ Winsock ลักษณะและส่วนประกอบของ โปรแกรมที่ผู้จัดทำได้ออกแบบไว้จะเป็นลักษณะดังรูปต่อไปนี้



รูปที่ 2.12 แสดงโปรแกรม Stream Video ในส่วนของ Client

ส่วนประกอบและหน้าที่ของส่วนต่าง ๆ ของ โปรแกรม

1. Picture Box ทำหน้าที่แสดงภาพ Video ที่การ Stream มาจากโปรแกรมในส่วน Server
2. Port เป็นส่วนที่กำหนดเลข Port ในการเชื่อมต่อระหว่างส่วนของ Server และ Client
3. IP เป็นช่องที่ใส่ IP ของเครื่องที่มีโปรแกรมในส่วน Server อยู่เพื่อทำการเชื่อมต่อ
4. Command Button: Connect เป็นปุ่มที่กดเพื่อเริ่มการเชื่อมต่อกับโปรแกรมในส่วน Server
5. Command Button: Disconnect เป็นปุ่มที่กดเพื่อหยุดการเชื่อมต่อกับโปรแกรมในส่วน Server
6. Winsock เป็นคอนโทรลเลอร์ตัวหนึ่งของ Visual Basic ที่ช่วยในการรับ-ส่ง ผ่านข้อมูลระหว่าง

เครือข่าย

วิธีการใช้งาน

1. ติดตั้งเวปแคม เข้ากับคอมพิวเตอร์ที่จะใช้เปิดโปรแกรมตัว Server
2. รันโปรแกรม Server แล้วกดปุ่ม Start Cam เพื่อให้โปรแกรมแสดงภาพที่ได้รับมาจากเวปแคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3. รันโปรแกรมส่วน Client ขึ้นมาระบุ Port และ IP ให้ตรงกับโปรแกรม Server (กรณีรันทั้งสองโปรแกรมบนคอมพิวเตอร์เครื่องเดียวให้ระบุ IP เป็น 127.0.0.1 ตามที่ได้กำหนดไว้ในส่วน Module ของโปรแกรมส่วน Server)
4. กดปุ่ม Start Server ของโปรแกรมส่วน Server เพื่อเปิด Port รอรับการเชื่อมต่อจาก Client เมื่อกดแล้วบนหัวโปรแกรมจะมีข้อความว่า “Listening on Port: ตามด้วยเลขPort ที่ตั้งค่าไว้”
5. กดปุ่ม Connect บนโปรแกรมส่วน Client เมื่อติดต่อสำเร็จบนหัวโปรแกรมจะแสดงขึ้นมาว่า “Connection Active”
6. กดปุ่ม Start Stream บนโปรแกรมส่วน Server ภาพที่แสดงบนโปรแกรมส่วน Server ก็จะไปปรากฏบนโปรแกรมในส่วน Client ด้วยและบนหัวโปรแกรมส่วน Server จะมีข้อความแสดงว่า “Stream Active”
7. กดปุ่ม Pause Stream เพื่อหยุดการส่งภาพจากส่วน Server ไปยัง ส่วน Client ชั่วคราว
8. กดปุ่ม Stop Stream เพื่อหยุดการส่งภาพจากส่วน Server ไปยังส่วน Client

คอนโทรล Winsock

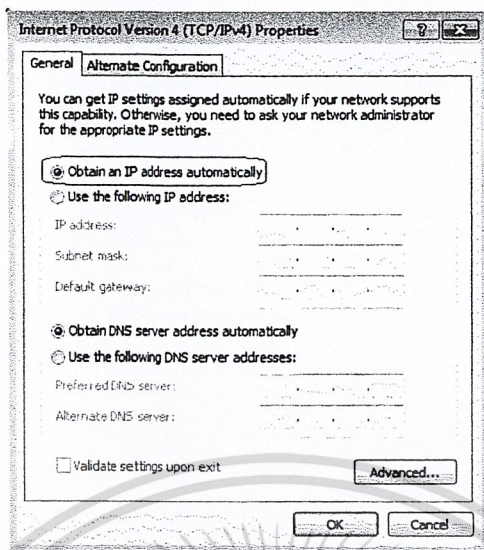
Winsock หรือ Winsock Socket API คือคอนโทรลของ Visual Basic ที่ช่วยในการเขียนโปรแกรมรับ-ส่งข้อมูลผ่านระบบเครือข่าย โดยการใช้โปรโตคอล TCP หรือ UDP เป็นตัวช่วยในการสื่อสารระหว่างคอมพิวเตอร์

การเขียนโปรแกรมด้วยคอนโทรล Winsock มี 2 ขั้นตอน คือ

1. Server รอรับการติดต่อ (Listen) จากไคลเอนต์ใดๆ โดยการเปิดพอร์ตของไคลเอนต์นั้นๆรอไว้
2. Client ติดต่อกับ Server ผ่านพอร์ตที่กำหนด เมื่อการติดต่อสำเร็จคอมพิวเตอร์ทั้งคู่จะสามารถรับ-ส่งข้อมูลถึงกันได้

ผู้ทดลองสามารถรันโปรแกรมบนคอมพิวเตอร์เครื่องเดียวได้ แต่ในกรณีการรันโปรแกรมบนคอมพิวเตอร์เครื่องเดียวและคอมพิวเตอร์ถูกใช้เชื่อมต่อกับอินเทอร์เน็ตผ่านโมเด็มด้วย ผู้ทดลองจะไม่สามารถกำหนด IP Address ของคอมพิวเตอร์ที่เรียกว่า Static IP Address ได้เนื่องจาก IP Address จะถูกกำหนดให้เป็นชนิด Dynamic IP Address โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การกำหนดให้เป็น Dynamic IP Address

เพื่อหลีกเลี่ยงปัญหานี้ผู้ทดลองสามารถใช้ IP Address เป็น 127.0.0.1 ซึ่งเป็น IP Address แทนตัวเองของคอมพิวเตอร์ทุกเครื่อง เราสามารถทดสอบโดยการใช้คำสั่ง Ping ตามที่ได้อธิบายตามหัวข้อที่ผ่านมาแล้วได้

มาตรฐาน Video for Windows (VFW)

VFW สำหรับ Win16

VFW เป็นมาตรฐานของเทคโนโลยีการจับภาพของบริษัท ไมโครซอฟท์ ซึ่งถูกพัฒนาครั้งแรกในรูปแบบของ add-on Win16 ของระบบปฏิบัติการ Windows 3.1 ใช้เป็นเทคโนโลยีการจับภาพจากอุปกรณ์บันทึกภาพแบบดิจิทัล เช่นกล้องวิดีโอ แต่คุณภาพของ VFW Win16 ไม่ดีมากนัก เพราะภาพที่ได้มีขนาดสูงสุดเพียง 320x240 pixel เท่านั้น

VFW สำหรับ Win32

ตั้งแต่ Windows 95 เป็นต้นมา ฟังก์ชันต่างๆของ VFW ได้ถูกนำมารวบรวมให้อยู่ในคอมโพเนนต์ msvfw32.dll และ avicap32.dll ซึ่งกลายมาเป็นคอมโพเนนต์หลักในระบบ Windows

อย่างไรก็ตาม ไมโครซอฟท์ก็ยังมีการพัฒนาเทคโนโลยีการจับภาพอย่างต่อเนื่องและใน ค.ศ. 1996 ได้พัฒนาจากมาตรฐาน VFW ให้เป็นมาตรฐาน Active Movie และมาตรฐานล่าสุดอย่าง DirectShow ที่สามารถจับภาพคุณภาพสูงได้ แต่อย่างไรก็ตามก็ยังไม่มีการใช้งานกันอย่างกว้างขวาง เนื่องจาก Hardware จับภาพส่วนใหญ่ยังไม่สนับสนุนเทคโนโลยีนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้จะมีเทคโนโลยีการจับภาพใหม่ๆ เกิดขึ้นมาอย่างต่อเนื่อง แต่มาตรฐาน VFW ก็ยังเป็นที่ยอมรับกันกระทั่งปัจจุบัน เนื่องจาก VFW สนับสนุนโคร์เวอร์ของอุปกรณ์จับภาพแทบทุกชนิด ประกอบกับเป็นมาตรฐานของไมโครซอฟท์ ซึ่งมีลูกค้าอยู่ทั่วโลก ทำให้ผู้ผลิตอุปกรณ์จับภาพยังคงจำเป็นต้องรักษามาตรฐาน VFW ไว้

ฟังก์ชัน VFW API capCreateCaptureWindow

capCreateCaptureWindow เป็นฟังก์ชันสำหรับสร้างพื้นที่ Capturewindow เป็นการจองพื้นที่หน่วยความจำสำหรับแสดงภาพ เมื่อมีการติดต่อกับโคร์เวอร์ของอุปกรณ์รับภาพก็ได้ รูปแบบฟังก์ชันเป็นดังนี้

Private Declare Function capCreateCaptureWindow Lib "avicap32.dll"

Alias "capCreateCaptureWindowA"

(ByVal lpszWindowName As String,

ByVal X As Long,

ByVal Y As Long,

ByVal nWidth As Long,

ByVal nHeight As Long,

ByVal hwndParent As Long,

ByVal nID As Long) As Long

อธิบายพารามิเตอร์

lpszWindowName คือ ชื่อของพื้นที่ Capture Window เช่น "My Capture Window"

dwStyle คือ รูปแบบการแสดงผล เช่น Window child, Window Visible แต่ถ้าไม่ต้องการกำหนดรูปแบบให้กำหนด dwStyle เป็น 0

X คือ พิกัด X ของมุมบนซ้ายของพื้นที่ Capture Window โดยปกติจะกำหนดให้ X = 0

Y คือ พิกัด Y ของมุมบนซ้ายของพื้นที่ Capture Window โดยปกติจะกำหนดให้ Y = 0

nWidth คือ ขนาดความกว้างของพื้นที่ Capture Window ปกติจะกำหนดให้ nWidth = 640

nHeight คือ ขนาดความสูงของพื้นที่ Capture Window ปกติจะกำหนดให้ nHeight = 480

hwndParent คือ Parent Handle

nID คือ Window Identifier ปกติจะกำหนดให้ nID = 0

เมื่อสร้าง Capture window แล้ว ขั้นตอนต่อไปคือการส่งเมสเสจจาก Windows ไปยัง Capture window ให้ทำงานตามเมสเสจร้องขอ ซึ่งประกอบไปด้วยเมสเสจหลัก 4 ตัวคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONNECT คือ เมสเสจสำหรับติดต่อกับ Capture window (เว็บแคม)

DISCONNECT คือ เมสเสจสำหรับยกเลิกการติดต่อกับเว็บแคม

GET_FRAME คือ เมสเสจสำหรับการดึงภาพจากเว็บแคม

COPY คือ เมสเสจสำหรับการก๊อปปี้ภาพที่ดึงมาไปเก็บที่คลิปบอร์ด

ฟังก์ชัน SendMessage

SendMessage เป็นฟังก์ชันสำหรับส่งเมสเสจ จาก Windows ไปสู่ออบเจกต์ หรืออินสแตนส์ออบเจกต์ เป้าหมายใดๆ เพื่อให้ทำการอย่างใดอย่างหนึ่งเช่น เมื่อมีการเรียกใช้เมธอดเคลียร์ของคอนโทรล List box ได้รับเมสเสจก็จะทำการวิเคราะห์และลบรายการทั้งหมดทิ้ง

เนื่องจาก Visual Basic ไม่สามารถส่งเมสเสจด้วยตัวเองได้ ดังนั้นจำเป็นจะต้องเรียกใช้ฟังก์ชัน SendMessage สำหรับเป็นตัวกลางติดต่อกับ Windows

รูปแบบการเรียกใช้ฟังก์ชัน

```
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any) As Long
```

อธิบายพารามิเตอร์

hwnd คือ ออบเจกต์ปลายทางที่ต้องการส่งเมสเสจไป

wMsg คือ เมสเมสจที่ต้องการส่ง

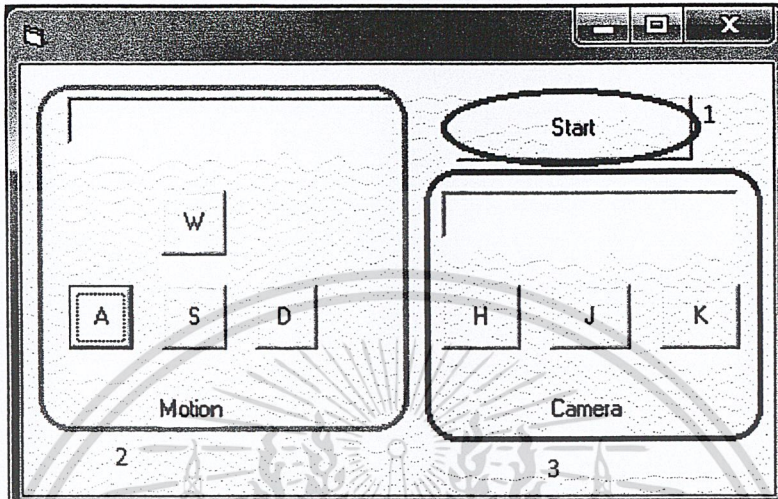
wParam และ lParam คือ พารามิเตอร์เสริมซึ่งมีหน้าที่และความหมายแตกต่างกันไปในเมสเสจแต่ละ

ชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โปรแกรม Visual Basic ในส่วนการควบคุม

หน้าต่างและส่วนประกอบของโปรแกรม



รูปที่ 2.14 หน้าต่างของโปรแกรม Control

ส่วนประกอบของโปรแกรม

1. ปุ่ม Start ซึ่งทำหน้าที่เริ่มการรับข้อมูลจากการกดคีย์บอร์ด
2. ส่วนของการควบคุมการเคลื่อนที่ของตัวหุ่นยนต์โดยแบ่งเป็น
 - 2.1 เมื่อกด A หุ่นยนต์จะเคลื่อนที่ไปทางซ้าย ในกล่องข้อความจะปรากฏข้อความว่า "Turn Left"
 - 2.2 เมื่อกด S หุ่นยนต์จะเคลื่อนที่ไปทางซ้าย ในกล่องข้อความจะปรากฏข้อความว่า "Reward"
 - 2.3 เมื่อกด W หุ่นยนต์จะเคลื่อนที่ไปทางซ้าย ในกล่องข้อความจะปรากฏข้อความว่า "Forward"
 - 2.4 เมื่อกด D หุ่นยนต์จะเคลื่อนที่ไปทางซ้าย ในกล่องข้อความจะปรากฏข้อความว่า "Turn Right"
3. ส่วนของการควบคุมกล้อง โดยแบ่งเป็น
 - 3.1 เมื่อกด H กล้องจะหมุนไปทางซ้าย ในกล่องข้อความจะปรากฏข้อความว่า "Camera Left"
 - 3.2 เมื่อกด J กล้องจะหมุนกลับมาตรงกลาง ในกล่องข้อความจะปรากฏข้อความว่า "Camera Center"
 - 3.3 เมื่อกด K กล้องจะหมุนไปทางขวา ในกล่องข้อความจะปรากฏข้อความว่า "Camera Right"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 กล้องเว็บแคม (Webcam)

กล้องเว็บแคม (Web Camera) หรืออาจจะเรียก Video Camera ก็ได้ เว็บแคมเป็นอุปกรณ์จับภาพที่สามารถจับภาพเคลื่อนไหวของผู้ใช้และสามารถส่งผ่านระบบเครือข่ายหรืออินเทอร์เน็ต ทำให้ผู้ใช้สามารถติดต่อกันได้ทั้งภาพและเสียง ในปัจจุบันมีการนำเว็บแคมมาใช้ประโยชน์อย่างแพร่หลาย เช่น ดูภาพผ่านระบบอินเทอร์เน็ต , การประชุมทางไกล ซึ่งมักจะใช้งานร่วมกับ โปรแกรมดูภาพผ่านอินเทอร์เน็ตเช่น โปรแกรม MSN, Camfrog และ Skype เป็นต้น



รูปที่ 2.15 กล้องเว็บแคม

ชนิดของเว็บแคม

1. แบ่งตามชนิดของเซ็นเซอร์

เซ็นเซอร์สำหรับกล้องเว็บแคม โดยทั่วไปมี 2 ชนิด คือ CCD และ CMOS แต่ที่นิยมใช้กันมากที่สุดคือ CMOS สำหรับเซ็นเซอร์แบบ CCD จะเป็นเซ็นเซอร์ที่นิยมใช้ในกล้องดิจิทัล เนื่องจากให้ความละเอียดที่สูงกว่า และก็มีสัญญาณรบกวนน้อยกว่าเซ็นเซอร์ชนิด CMOS

2. แบ่งตามรูปแบบการเชื่อมต่อ

การเชื่อมต่อของกล้องเว็บแคม ในปัจจุบันส่วนใหญ่จะเชื่อมต่อผ่านสายสัญญาณแบบ USB แทบทั้งสิ้น โดยทั่วไปมี 2 เวอร์ชันคือ USB 1.1 และ USB 2.0 นอกจากนี้ยังมีกล้องเว็บแคมชนิดไร้สายที่ใช้การเชื่อมต่อผ่านระบบ Wi-Fi หรือ Wireless มีข้อดีในเรื่องความสะดวกสบายในการจัดวาง แต่ยังมีราคาค่อนข้างแพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 หน่วยประมวลผล dsPIC30F4011

2.5.1 การใช้งานโมดูล UART กับพอร์ตอนุกรม

โมดูล Universal Asynchronous Receiver Transmitter (UART) หรือการสื่อสารข้อมูลในรูปแบบอะซิงโครนัส เป็นหนึ่งในโมดูลที่เกี่ยวข้องกับการอินพุต / เอาต์พุตแบบอนุกรมที่มีอยู่ภายใน dsPIC30F โดยโมดูล UART จะเป็นอะซิงโครนัสแบบฟูลดูเพล็กซ์(full-duplex) รับส่งข้อมูลได้พร้อมกัน(สองทิศทาง) สามารถมาประยุกต์ใช้งานกับการเชื่อมต่อ กับ อุปกรณ์ ภายนอก เช่น คอมพิวเตอร์, RS-232 เป็นต้น

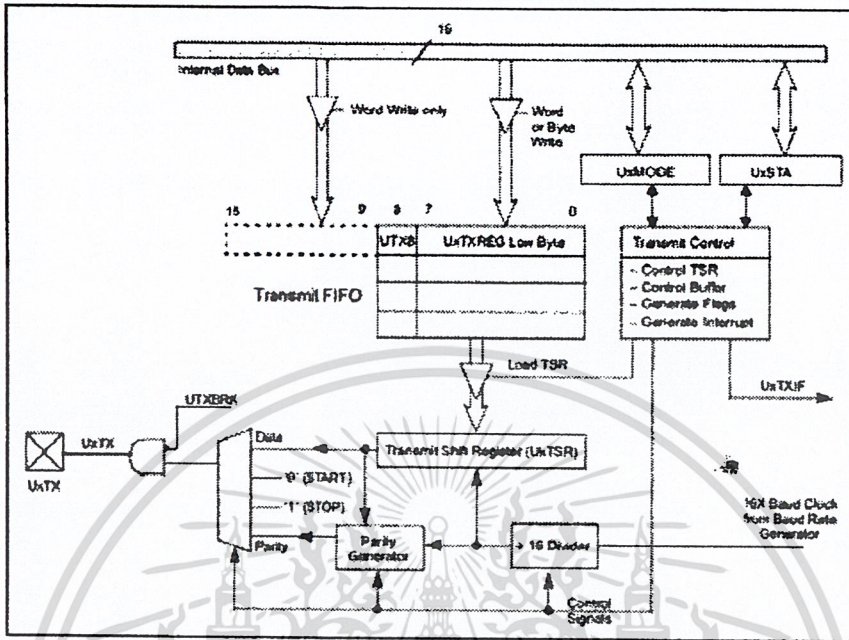
ไมโครคอนโทรลเลอร์ dsPIC30F4011 มีโมดูล UART ให้ใช้งานได้ 1 UART โดยกำหนดขาพอร์ตสำหรับโมดูล UART ไว้ที่ขาพอร์ต U1RX/RF2 สำหรับรับข้อมูลอนุกรม และ U1TX/RF3 สำหรับส่งข้อมูลอนุกรม นอกจากนี้ยังได้เพิ่มเติมพอร์ตอนุกรมเสริม (UART Alternate I/O) ที่ตำแหน่งขาพอร์ต U1ATX/RC13 สำหรับการส่งข้อมูลและ U1ATX/RC14 สำหรับการรับข้อมูลอนุกรมการใช้งานต้องเลือกอย่างใดอย่างหนึ่งเท่านั้น โดยสามารถเลือกการใช้งานผ่านการเซตบิต ALTIO (บิตที่ 10) ในรีจิสเตอร์ UxMODE (UARTx Mode Register)

2.5.1.1 คุณสมบัติที่สำคัญของโมดูล UART

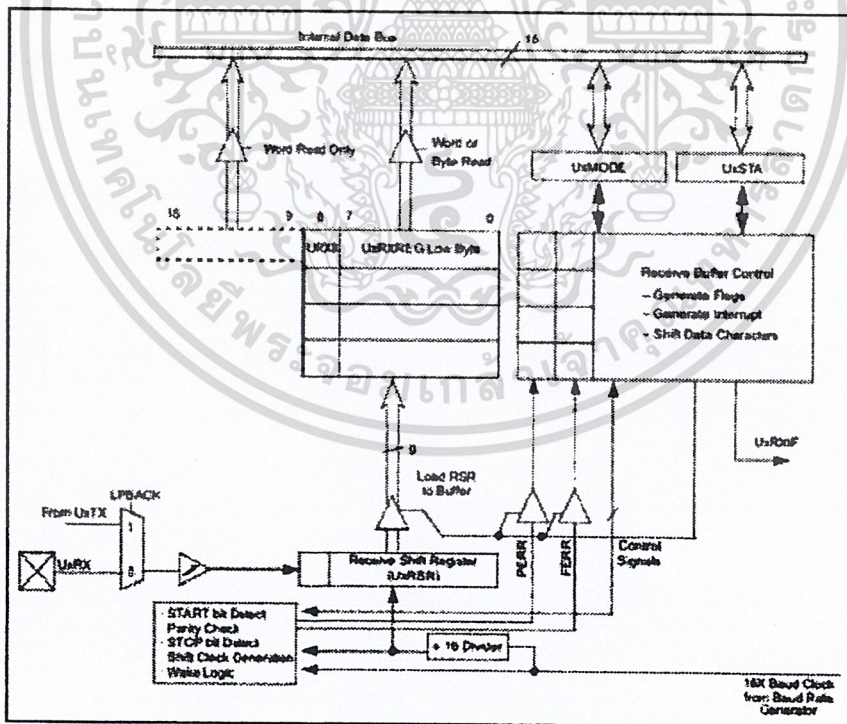
- รับส่งข้อมูลแบบฟูลดูเพล็กซ์
- มีคุณสมบัติการกำหนดข้อมูลการรับส่ง ไม่ว่าจะเป็นจำนวนบิตข้อมูล (8 หรือ 9 บิต), บิตพาริตี (คู่, คี่, ไม่มี), บิตหยุดข้อมูล (1 หรือ 2 บิต)
- กำหนดอัตราบอด (Baud Rate) พร้อมกับปริสเกลเลอร์ 16 บิต
- กำหนดอัตรารับส่งข้อมูลได้ตั้งแต่ 29 bps ถึง 1.875 Mbps ที่ความถี่ 30MHz
- มีบัฟเฟอร์รับส่งข้อมูล 4 ระดับ
- มีส่วนตรวจสอบข้อผิดพลาด พาริตี, เฟรมข้อมูล และการซ้อนทับของบัฟเฟอร์ข้อมูล
- สนับสนุนการทำงานในโหมด 9 บิตข้อมูลพร้อมกับการตรวจจับแอดเดรส (ถ้าบิตที่ 9 เป็น 1 จะเป็นแอดเดรส บิตที่ 9 เป็น 0 จะเป็นข้อมูล)
- อินเทอร์รัปต์รับส่งข้อมูล
- โหมดลูปแบ็ก (Loop back) เพื่อการตรวจสอบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บล็อกไดอะแกรมการรับส่งข้อมูลโมดูล UART แสดงดังรูป 2.17 และ 2.18



รูปที่ 2.16 บล็อกไดอะแกรมการส่งข้อมูลของโมดูล UART



รูปที่ 2.17 บล็อกไดอะแกรมการรับข้อมูลของโมดูล UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.2 รีจิสเตอร์ควบคุมโมดูล UART

- รีจิสเตอร์ UxMODE (UARTx Mode Register)

กำหนดโหมดการทำงานของโมดูล UART เช่น ขาพอร์ตรับส่งข้อมูลที่ต้องการ อัตราการรับส่งข้อมูล, จำนวนบิตข้อมูล พาริตีบิต และบิตหยุด เป็นต้น

- รีจิสเตอร์ UxSTA (UARTx Status and Control Register)

รีจิสเตอร์แสดงสถานะการทำงานของโมดูล UART และบิตควบคุมการทำงานของ UART เพิ่มเติม

- รีจิสเตอร์ UxRXREG (UARTx Receive Register)

รีจิสเตอร์รับข้อมูล ใช้งานเพียง 9 บิต สำหรับรับข้อมูลที่เข้ามาทางการเชื่อมต่อของ พอร์ต

อนุกรม

- รีจิสเตอร์ UxTXREG (UARTx Transmit Register)

รีจิสเตอร์ส่งข้อมูล (เขียนได้อย่างเดียว) ใช้งานเพียง 9 บิต สำหรับส่งข้อมูลออกทาง การเชื่อมต่อ พอร์ตอนุกรม

- UxBRG (UARTx Baud Rate Register)

รีจิสเตอร์กำหนดอัตราบอดหรือความเร็วในการรับส่งข้อมูลของโมดูล UART

2.5.1.3 การคำนวณอัตราบอด (Baud Rate)

ก่อนการใช้งานการสื่อสารข้อมูลอนุกรมจะต้องกำหนดอัตราบอดสำหรับการสื่อสารข้อมูล โดยคำนวณอัตราบอดได้ด้วยสมการ ดังนี้

$$Baud\ Rate = \frac{F_{CY}}{16 \cdot (UxBRG + 1)}$$

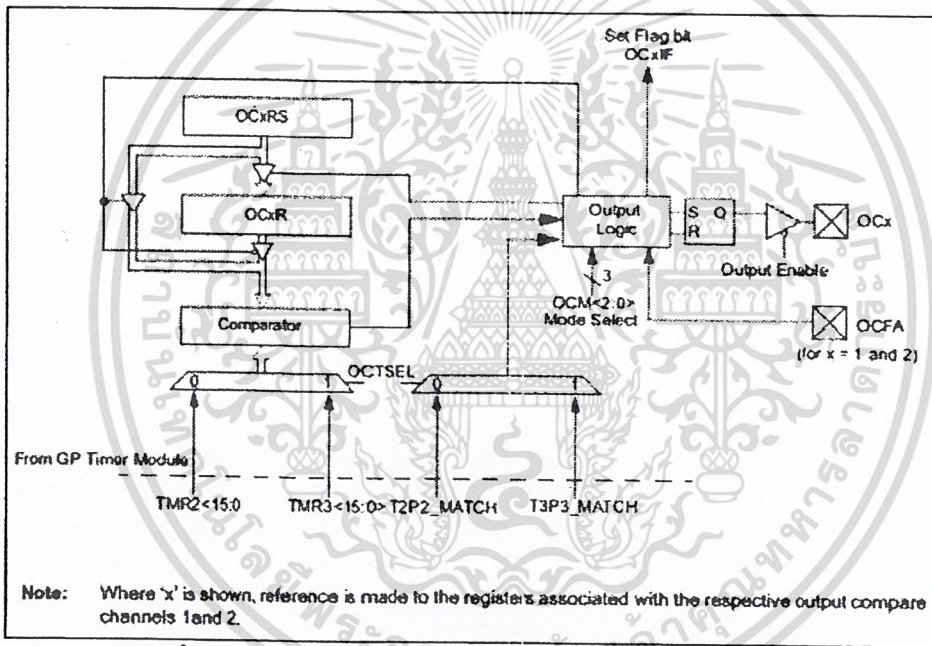
$$UxBRG = \frac{F_{CY}}{16 \cdot Baud\ Rate} - 1$$

โดย F_{CY} = ความถี่สัญญาณนาฬิกาหารด้วย 4 ($F_{osc}/4$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การใช้งานโมดูลเปรียบเทียบข้อมูล

โมดูลเปรียบเทียบข้อมูลเพื่อสร้างสัญญาณเอาต์พุต (Output Compare) เป็นการเปรียบเทียบข้อมูลในรีจิสเตอร์ OCxR (Output Compare x Register) และรีจิสเตอร์ OCxRS (Secondary Output Compare x Register) กับข้อมูลค่าฐานเวลาในรีจิสเตอร์ TMRx (Timer Register x) ตามเงื่อนไขของโหมดการทำงาน เมื่อมีค่าเท่ากัน จะสร้างสัญญาณออกมาตามที่กำหนดไว้ แสดงผลลัพธ์ผ่านทางขาพอร์ต OCx (Output Compare x) แสดงดังรูปที่ 2.18 ตำแหน่งขาพอร์ตจะขึ้นอยู่กับเบอร์ไมโครคอนโทรลเลอร์ที่เลือกใช้งานสำหรับ dsPIC30F2010 จะมีขา OC คือขาพอร์ต OC1/RD0 และ OC2/RD1 บล็อกโคอะแกรมการทำงานของโมดูลแสดงดังรูปที่ 2.18



รูปที่ 2.18 โคอะแกรม โมดูลเปรียบเทียบข้อมูลเอาต์พุต

โมดูลเปรียบเทียบข้อมูลเพื่อสร้างสัญญาณเอาต์พุต สามารถกำหนดให้สร้างสัญญาณพัลส์ (Pulse) ได้ทั้งแบบพัลส์เดี่ยว สัญญาณพัลส์ต่อเนื่องและสร้างสัญญาณ PWM (Pulse Width Modulation) โดยมีโหมดการทำงานดังนี้

- โหมดเปรียบเทียบข้อมูลสร้างสัญญาณเอาต์พุตเดี่ยว (Single Compare Match mode)

การทำงานของโหมดนี้จะเป็นการสร้างสัญญาณเอาต์พุตผ่านทางขา OCx และสามารถกำหนดได้ว่าจะให้สร้างสัญญาณ High (ลอจิก '1'), Low (ลอจิก '0') หรือสัญญาณ Toggle (กลับสถานะลอจิก) โดยใช้รีจิสเตอร์ OCxR ในการเปรียบเทียบข้อมูล

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โหมดเปรียบเทียบข้อมูลคู่สร้างสัญญาณเอาต์พุต (Dual Compare Match mode)

การทำงานของโหมดนี้จะเป็นการสร้างสัญญาณพัลส์เอาต์พุตผ่านทางขา OCx โดยรีจิสเตอร์นับค่าเวลา (TMRx) จะเปรียบเทียบค่ากับรีจิสเตอร์ OCxR และ OCxRS ในการเปรียบเทียบข้อมูลสามารถกำหนดโหมดการทำงานได้ 2 โหมดคือ

1. สัญญาณพัลส์เดี่ยว (Single Output Pulse)
2. สัญญาณพัลส์ต่อเนื่อง (Continuous Output Pulse)

- โหมดสร้างสัญญาณ PWM (Simple Pulse Width Modulation mode)

การทำงานของโหมดนี้จะเป็นการสร้างสัญญาณทางความกว้างของพัลส์หรือ PWM โดยแบ่งลักษณะการทำงานในการตรวจสอบข้อผิดพลาดได้ 2 แบบ คือ

1. มีอินพุตการตรวจสอบและป้องกันการผิดพลาด (with FAULT Protection Input)
2. ไม่มีอินพุตการตรวจสอบและป้องกันการผิดพลาด (without FAULT Protection Input)

2.5.2.1 รีจิสเตอร์ควบคุมโมดูลเปรียบเทียบข้อมูลเพื่อสร้างเอาต์พุต

รีจิสเตอร์ที่เกี่ยวข้องกับโมดูลเปรียบเทียบข้อมูล มีดังนี้

- รีจิสเตอร์ OCxCON (Output Compare Control Register)
เป็นรีจิสเตอร์กำหนดคุณสมบัติการทำงานของโมดูลเปรียบเทียบข้อมูล
- รีจิสเตอร์ OCxR (Output Compare Data Register)
เป็นรีจิสเตอร์เก็บค่าข้อมูลที่ต้องการเปรียบเทียบตัวที่ 1 มีขนาด 16 บิต
- รีจิสเตอร์ OCxRS (Output Compare Secondary Data Register)
เป็นรีจิสเตอร์เก็บค่าข้อมูลที่ต้องการเปรียบเทียบตัวที่ 1 มีขนาด 16 บิต

2.5.2.2 โหมดการใช้งานเปรียบเทียบข้อมูลสร้างสัญญาณ PWM

การควบคุมการหมุนของเซอร์โวมอเตอร์ จะเป็นการควบคุมสัญญาณเพียง 1 เส้น สำหรับเซอร์โวมอเตอร์ในแต่ละตัว และต้องควบคุมสัญญาณในรูปแบบ PWM ซึ่งเราสามารถนำโมดูลเปรียบเทียบข้อมูล มาสร้างสัญญาณในรูปแบบ PWM โดยมีสูตรในการคำนวณหาค่า PWM ดังนี้

$$\text{PWM Period} = [(PRy) + 1] * T_{cy} * (\text{TMRy Prescale Value})$$

$$\text{PWM Frequency} = 1 / [\text{PWM Period}]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.3 การคำนวณหาค่า PWM สำหรับเวลา 1 มิลลิวินาที

การคำนวณหาค่าของสัญญาณพัลส์ 1 มิลลิวินาที เพื่อกำหนดให้กับรีจิสเตอร์ OCxRS สำหรับควบคุมสัญญาณ PWM สามารถคำนวณได้ดังนี้

สัญญาณพัลส์คาบวง 1 มิลลิวินาที สัญญาณพัลส์คาบรอบ 20 มิลลิวินาที เพราะฉะนั้นสัญญาณ PWM เท่ากับ 21 มิลลิวินาที (21000 ไมโครวินาที) กำหนดปริสเกลเลอร์ 256 เฟสล็อกกลูบ 4 ความถี่ออสซิลเลเตอร์ 4.0 MHz จากสูตร

$$\text{PWM Period} = [(PRy) + 1] * T_{cy} * (\text{TMRy Prescale Value})$$

$$21000 \text{ us} = (PR + 1) \times (4 \times (1/4 \text{ MHz})) \times 256$$

$$PR + 1 = 0.021 / 0.000256$$

$$PR = 82.031 - 1$$

$$PR = 81$$

ค่า PR จะเป็นคาบเวลาของสัญญาณ PWM ที่ต้องการใช้งาน

$$\text{ที่เวลา } 21 \text{ ms} = 100\%$$

$$1 \text{ ms} = 100/21$$

$$= 4.76\%$$

$$D_{\text{pwm}} = (D(\%) \times T_{\text{pwm}}) / 100$$

$$= (4.76 \times 0.021) / 100$$

$$= 0.00099 \approx 0.001$$

$$\text{OCxRS} = D_{\text{pwm}} / (T_{\text{osc}} \times N)$$

$$= 0.001 / ((1/4.0\text{M}) \times 256)$$

$$= 15.625 \approx 16$$

เพราะฉะนั้นค่าที่กำหนดให้กับรีจิสเตอร์ OCxRS เท่ากับ 16 หรือค่าดิวตี้ไซเคิล (Duty Cycle) เพื่อสร้างสัญญาณ PWM ที่มีคาบเวลา 21 มิลลิวินาทีและสัญญาณพัลส์ด้านบวกรเท่ากับ 1 มิลลิวินาที การคำนวณสัญญาณ PWM ที่เวลา 1.5 มิลลิวินาทีและ 2 มิลลิวินาที จะมีรูปแบบเดียวกันกับตัวอย่างข้างต้น โดยที่คาบเท่ากับ 1.5 มิลลิวินาทีจะได้ค่า OCxRS เท่ากับ 21 และ คาบเวลาเท่ากับ 2 มิลลิวินาที จะได้ค่า OCxRS เท่ากับ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การเชื่อมต่อทางแสง

ตัวเชื่อมโยงทางแสง (Opto coupler) หรือตัวแยกโดยใช้แสง (Opto Isolator) เป็นอุปกรณ์ที่นำมาเชื่อมโยงสัญญาณต่างๆของวงจรที่มีกราวด์ต่างกัน สามารถป้องกันการรบกวนซึ่งกันและกันระหว่างอินพุตกับเอาต์พุตได้อย่างเด็ดขาด ตัวเชื่อมโยงทางแสง เป็นอุปกรณ์ที่ประกอบด้วยแหล่งกำเนิดแสงและตัวจับแสงอยู่ภายในตัวเดียวกัน แหล่งกำเนิดแสงส่วนใหญ่จะใช้ไดโอดเปล่งแสงอินฟราเรดที่ทำมาจาก แกลเลียมอาร์เซไนด์ (GaAs) ส่วนตรวจจับแสง (Photo Detector) สามารถเป็นได้ทั้ง โฟโตทรานซิสเตอร์, SCR, โฟโตไดโอด และ TRIAC ในตัวเชื่อมโยงทางแสง สัญญาณอินพุตจะถูกเปลี่ยนเป็นสัญญาณแสงแล้วจึงส่งสัญญาณแสงนี้ไปที่โฟโตดีเทคเตอร์ ตัวเชื่อมโยงทางแสงถูกออกแบบมาเพื่อป้องกันอุปกรณ์อิเล็กทรอนิกส์ไม่ได้รับแรงกระชากสูงๆ ซึ่งเป็นต้นเหตุที่ทำให้เกิดสัญญาณเอาต์พุตที่ไม่ถูกต้อง

2.7 ดีซีมอเตอร์ (DC Motor)

หลักการทํางานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานสดิวเซอร์ที่มีการออกแบบให้คุณลักษณะพิเศษคือ แรงบิดของเพลลาของมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแสคือ

$$T = K \Phi I$$

เมื่อ	T	คือ	แรงบิดของเพลลา มีหน่วยเป็น นิวตัน- เมตร
	Φ	คือ	เส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์
	I	คือ	กระแส มีหน่วยเป็น แอมป์
	K	คือ	ค่าคงที่

จะเห็นได้ว่าแรงบิดของเพลลาจะเป็นสัดส่วนโดยตรงกับ ผลคูณของเส้นแรงแม่เหล็กและกระแส การที่ขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กจะทำให้เกิดแรงดันตกคร่อมตัวมันเอง โดยแรงดันจะเป็นสัดส่วนกับความเร็วของเพลลาของมอเตอร์และเส้นแรงแม่เหล็ก โดยมีความสัมพันธ์ ดังนี้

$$E = K \Phi \omega$$

เมื่อ	E	คือ	แรงดันย้อนกลับ มีหน่วยเป็น โวลท์
	Φ	คือ	เส้นแรงแม่เหล็ก มีหน่วยเป็น เวเบอร์
	ω	คือ	ความเร็วของมอเตอร์ มีหน่วยเป็น เรเดียน /วินาที
	K	คือ	ค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 เซอร์โวมอเตอร์ (SERVO MOTOR)

โครงสร้างและส่วนประกอบของเซอร์โวมอเตอร์

ภายในตัวเซอร์โวมอเตอร์จะเป็นมอเตอร์ความเร็วสูง มีเฟืองทศรอบให้หมุนช้าลงเพื่อจะได้มีกำลังแรงบิดที่สูงขึ้น แขนหรือกระเดื่องของเซอร์โวมอเตอร์จะยึดติดกับแกนเฟืองที่ทศรอบ การที่รูปพัลส์สี่เหลี่ยมขนาดกว้าง 1-2 ms สามารถควบคุมทิศทางของมอเตอร์ในตัวเซอร์โวมอเตอร์ให้เคลื่อนไปตำแหน่งใดๆ ได้ก็เพราะภายในเซอร์โวมอเตอร์มีวงจรควบคุมมอเตอร์อีกชุดหนึ่ง วงจรควบคุมมอเตอร์จะนำค่าแรงดันเฉลี่ยรูปสี่เหลี่ยม (ค่าแรงดันเฉลี่ยนี้ขึ้นอยู่กับขนาดความกว้างของพัลส์) เข้าไปเปรียบเทียบกับแรงดันค่าหนึ่งที่มีอยู่ในวงจร ถ้าค่าต่างกันวงจรควบคุมจะสั่งให้มอเตอร์หมุนไปตามทิศทางที่ขึ้นกับขนาดความกว้างของพัลส์ แกนเฟืองทศรอบจะถูกพ่วงไปขับแกนของ VR (ตัวต้านทานปรับค่าได้) ในวงจรควบคุมมอเตอร์ให้หมุนไปด้วยค่าของ VR ตัวนี้จะเป็นตัวกำหนดค่าแรงดันเปรียบเทียบของวงจรมอเตอร์จะเปลี่ยนไปด้วยค่าแรงดันเปรียบเทียบที่เปลี่ยนไป จนเท่ากับค่าเฉลี่ยของสัญญาณพัลส์รูปสี่เหลี่ยมที่เข้ามา มอเตอร์จะหยุดการหมุนทันที ซึ่งหมายความว่าแขนหรือกระเดื่องของเซอร์โวมอเตอร์ได้เคลื่อนไปอยู่ในตำแหน่งที่ต้องการ ตามการบังคับควบคุมจากทางด้านเครื่องส่งสัญญาณแล้ว

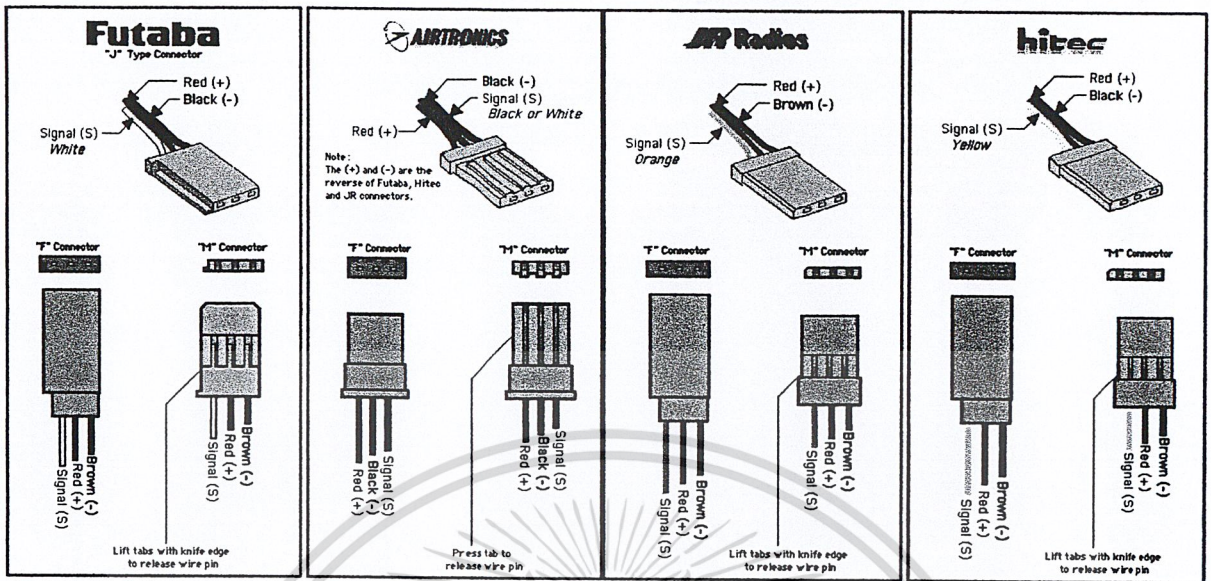
หลักการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ (SERVO MOTOR) เป็นมอเตอร์ไฟตรง (DC) แบบหนึ่งทำงานด้วยสัญญาณพัลส์ มีสายต่อ 3 เส้นคือ เส้นสีแดงคือ ไฟเลี้ยง 4.8-6 Volt เส้นสีดำคือสายกราวด์ และเส้นสีขาวคือสายรับสัญญาณแบบ PWM ภายในเซอร์โวมอเตอร์จะมีวงจรอิเล็กทรอนิกส์บรรจุอยู่ที่กล่าวมาเพื่อทำการควบคุมการทำงาน การควบคุมความเร็วจะมาจากลักษณะของสัญญาณพัลส์ที่ป้อนเข้ามาไปยังสายรับสัญญาณ



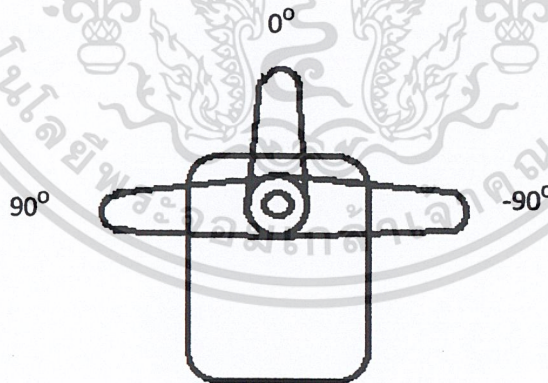
รูปที่ 2.19 แสดงลักษณะรูปร่างของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 แสดงการต่อสายกับเซอร์โวมอเตอร์

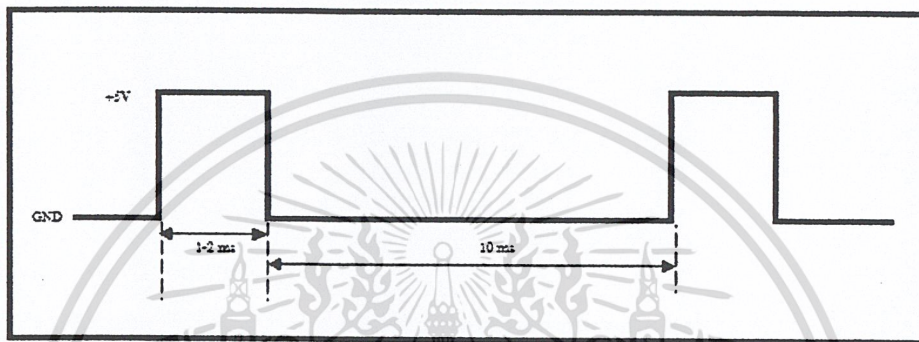
แกนหมุนของเซอร์โวมอเตอร์จะไม่สามารถหมุนได้รอบตัวครบ 360 องศาเหมือนกับมอเตอร์ไฟตรง เนื่องจากข้อจำกัดของกลไกภายใน (แต่สามารถทำให้หมุน 360 องศาได้โดยต้องมีการดัดแปลง ซึ่งจะทำให้เซอร์โวมอเตอร์ไม่สามารถทำให้เหมือนเดิมก่อนดัดแปลงได้) เซอร์โวมอเตอร์สามารถหมุนได้ประมาณ 180 องศา



รูปที่ 2.21 แสดงความสามารถในการหมุนของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขับให้เซอร์โวมอเตอร์หมุนจะต้องใช้ขบวนการของสัญญาณพัลส์ที่มีคาบเวลารวมประมาณ 12 มิลลิวินาที โดยต้องมีเวลาที่พัลส์บวก 1-2 มิลลิวินาที ประเป็นช่วงพัลส์ลบประมาณ 10 มิลลิวินาที โดยถ้าหากพัลส์บวกมีความกว้าง 1 มิลลิวินาที แกนหมุนของเซอร์โวมอเตอร์จะอยู่ตำแหน่งกึ่งกลาง และ สุดท้าย หากพัลส์บวกมีความกว้าง 2 มิลลิวินาที มอเตอร์จะอยู่ในตำแหน่งขวา อย่างไรก็ตามการกำหนดตำแหน่งที่หมุนไปของเซอร์โวมอเตอร์สามารถควบคุมได้ โดยการกำหนดความกว้างของพัลส์บวกให้ละเอียดขึ้นเช่น เป็น 1.1, 1.2, 1.3 มิลลิวินาที เป็นต้น

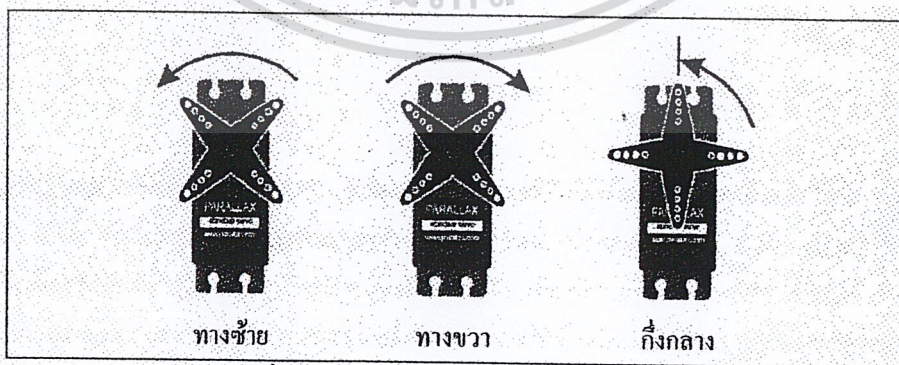


รูปที่ 2.22 โค้ดแกรมเวลาของสัญญาณพัลส์ที่ใช้สำหรับขับเซอร์โวมอเตอร์

ในกรณีนี้เราสร้างสัญญาณพัลส์ที่มีช่วงเวลาไม่เท่ากันอย่างเช่น เรากำหนดให้สร้างพัลส์ที่มีค่าเวลาหรือความกว้างขนาดเท่ากับ 1.5 มิลลิวินาที จะสามารถควบคุมแกนของเซอร์โวมอเตอร์ให้อยู่ตำแหน่งกลางที่ 0 องศา

เมื่อให้มีการป้อนสัญญาณพัลส์ที่มีความกว้างของเวลาขนาดเท่ากับ 1 มิลลิวินาที จะให้บังคับแกนของเซอร์โวมอเตอร์หมุนไปทางซ้าย 45 องศา

เมื่อให้มีการป้อนสัญญาณพัลส์ ซึ่งมีค่าความกว้างของเวลาขนาดเท่ากับ 2 มิลลิวินาที เซอร์โวมอเตอร์จะมีการบังคับแกนของเซอร์โวมอเตอร์หมุนไปทางขวาที่ตำแหน่ง 45 องศาทันที



รูปที่ 2.23 แสดงการป้อนพัลส์ช่วงเวลาต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการทำงาน

3.1 ภาพรวมของโครงการ

ในโครงการนี้เป็นการรวบรวม สิ่งงานการควบคุมความเร็วของมอเตอร์ มารวมกันบนหน้าจอกอมพิวเตอร์ เพื่อความสะดวก และรวดเร็วในการประมวลผล การทำงาน การสั่งงาน ในการออกแบบนั้น จึงแบ่งได้ 2 ส่วน คือ ส่วนการควบคุมการสั่งงาน และส่วนปฏิบัติการ

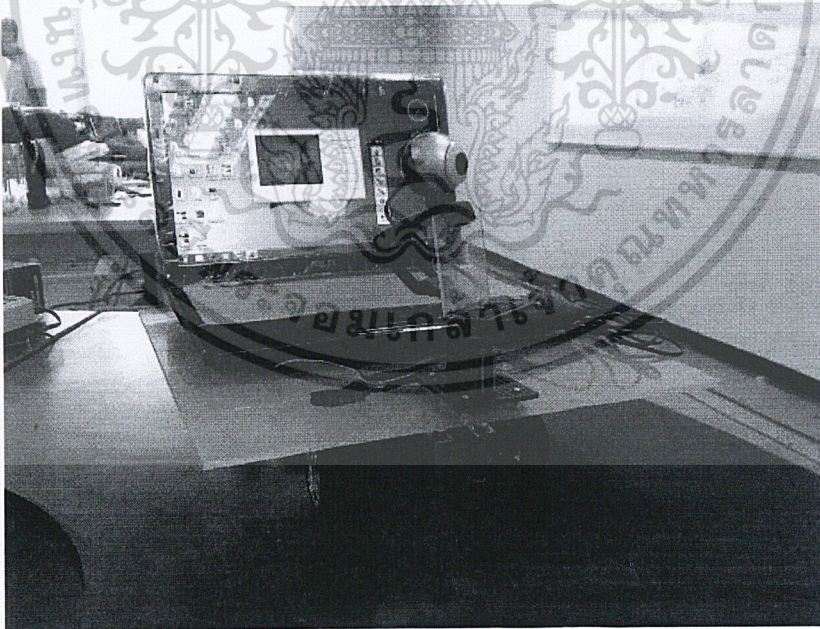
3.1.1 ส่วนการออกแบบ

ส่วนของตัวรถ

ส่วนของโครงสร้าง ใช้โครงสร้างของรถบังคับ โดยใช้ DC มอเตอร์ มีการออกตัวรถใหม่เพื่อใช้ในการวาง Computer Notebook และติดตั้งเซอร์ไว้มอเตอร์เพื่อทำหน้าที่ในการหมุนกล้อง Webcam

กล้อง

กล้องจะหมุนได้ 180 องศา ทำให้สามารถ มองเห็นเหยื่อผู้ประสพภัยในสนามได้



รูปที่ 3.1 ส่วนของตัวรถและกล้อง webcam

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ส่วนควบคุมและสั่งงาน

ในส่วนนี้จะอยู่บนหน้าจคอมพิวเตอร์ เพื่อความสะดวกในการสั่งงาน และการดูผลการทำงาน จึงเลือกใช้โปรแกรม Visual Basic 6

3.1.3 ส่วนปฏิบัติงาน

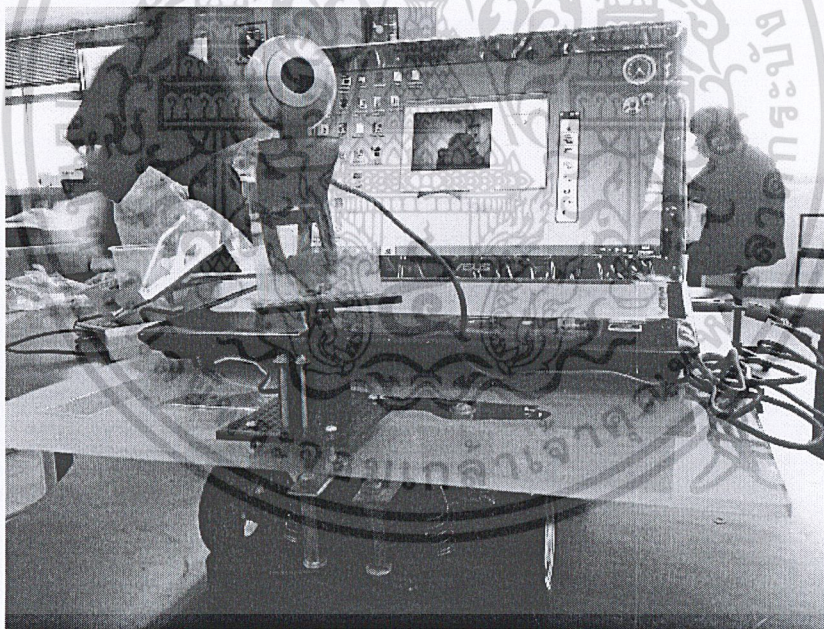
ส่วนนี้เป็นตัวโครงสร้างของหุ่นยนต์ ซึ่งประกอบด้วยส่วนต่าง ๆ เช่น มอเตอร์ ซึ่งได้รับสัญญาณควบคุมจาก dsPIC

3.2 การเชื่อมต่อไร้สาย

โมดูลที่ใช้ในการรับ ส่งสัญญาณควบคุมในการทดลองนี้ ใช้ Access point 2.4 GHz ยี่ห้อ Linksys

3.3 ส่วนของการมองของหุ่นยนต์

ส่วนของกล้องนั้นได้เลือกใช้ กล้องเว็บแคม (Webcam Camera) ที่มีความละเอียดของตัวกล้อง 1.3 ล้านพิกเซล



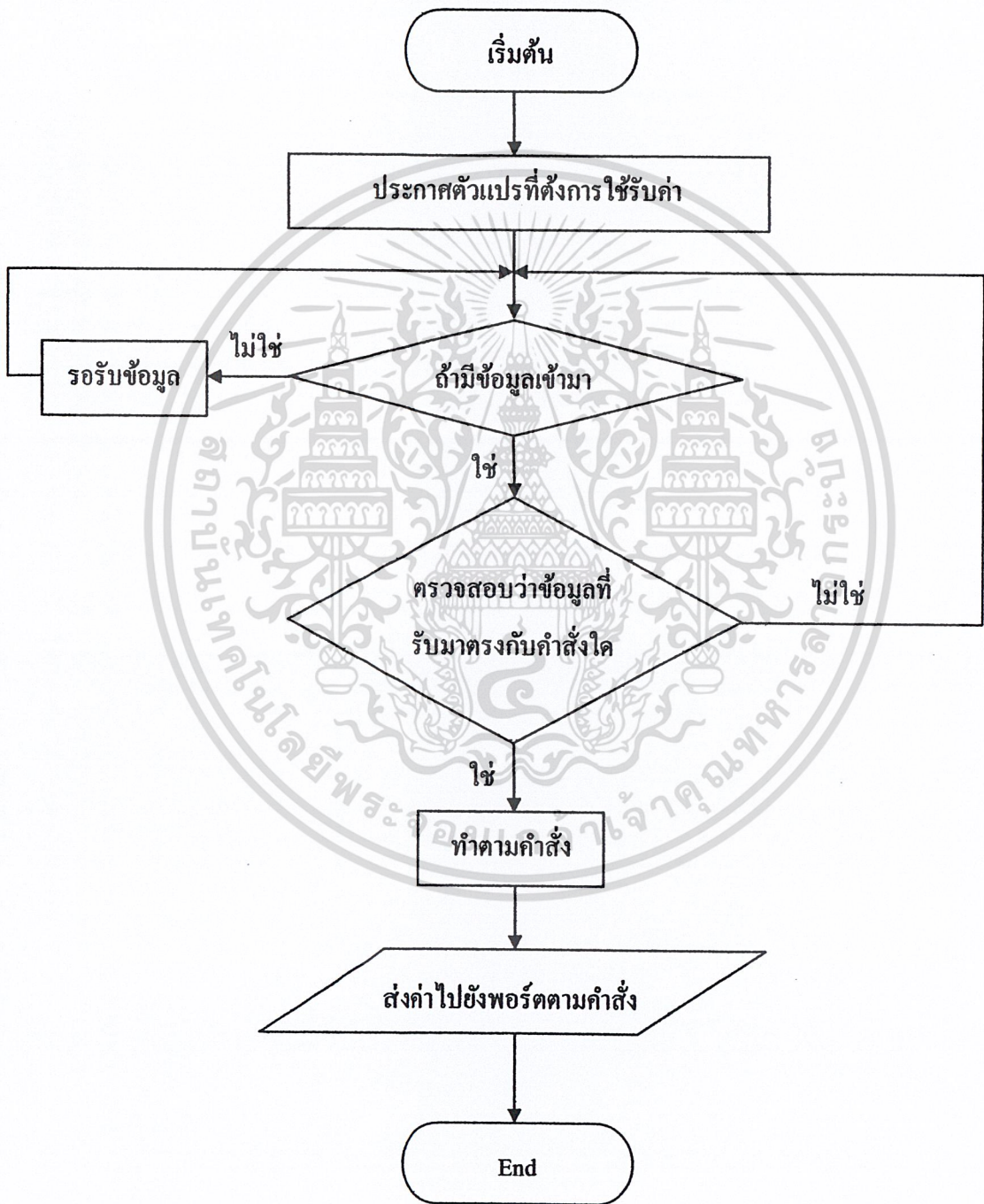
รูปที่ 3.2 ส่วนมุมมองของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนโปรแกรมควบคุม

3.4.1 การออกแบบโปรแกรมในส่วน dsPIC30F4011

ในการออกแบบนี้สามารถอธิบายได้ด้วยโฟลชาร์ตข้างล่างนี้

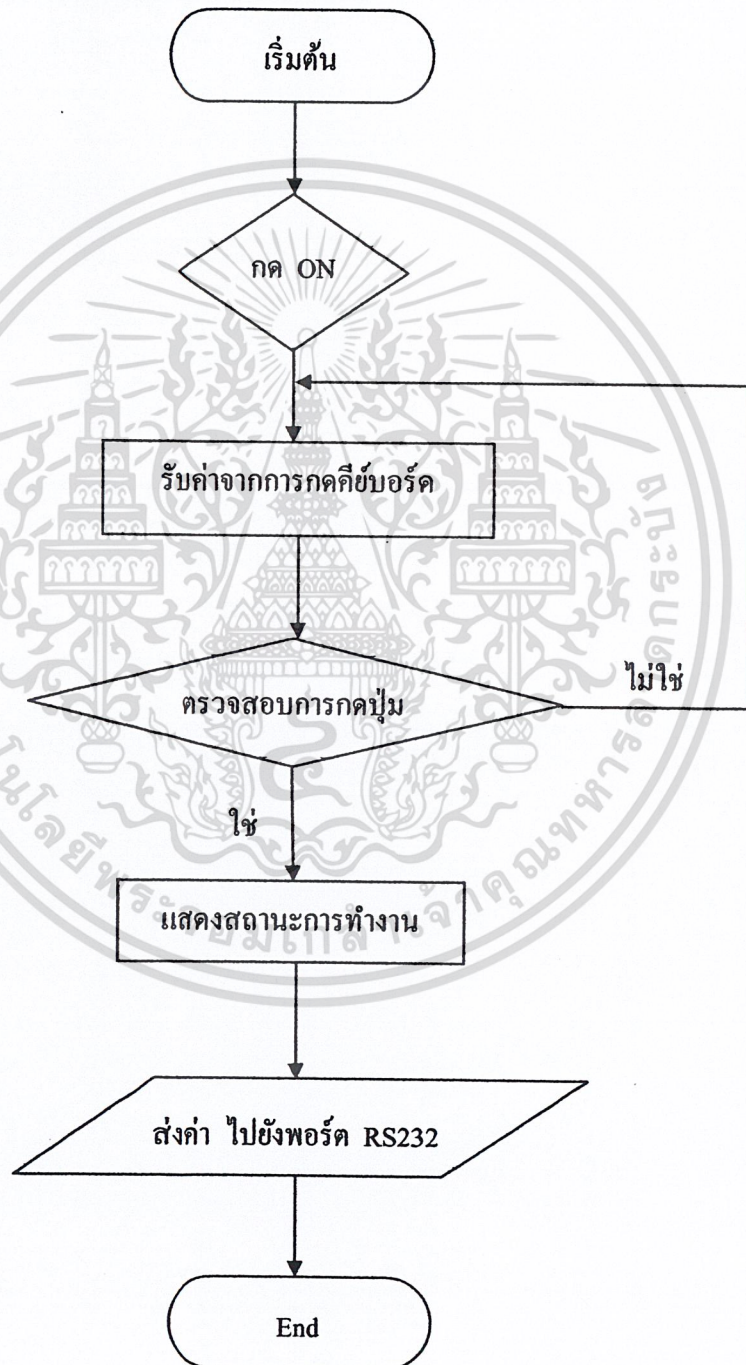


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การออกแบบโปรแกรมในส่วน Visual Basic 6

ในการออกแบบนี้สามารถอธิบายได้ด้วยโฟลชาร์ตดังนี้

โฟลชาร์ตแสดงการส่งข้อมูลผ่านพอร์ตอนุกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

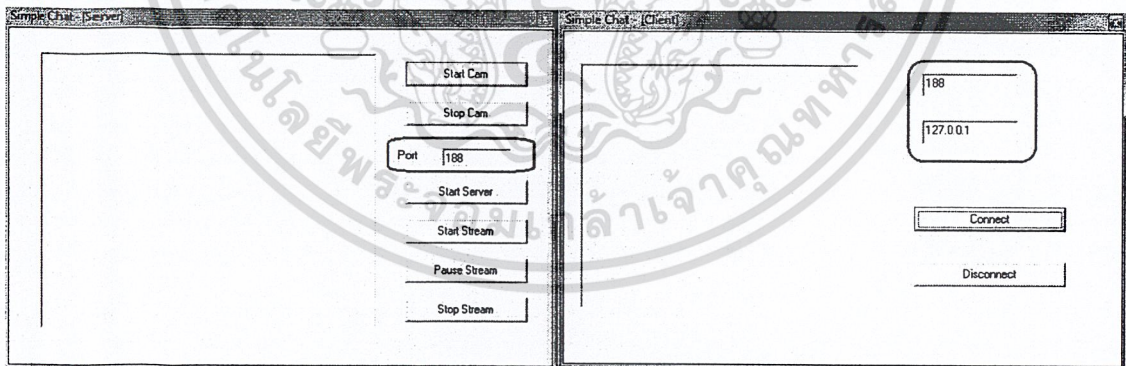
4.1 การทดลองในส่วนของโปรแกรม Visual Basic

4.1.1 โปรแกรม Visual Basic ในส่วนของการรับ ส่งภาพ

ในการทดลองนี้ได้ทดลองรับ-ส่งภาพโดยใช้คอมพิวเตอร์เครื่องเดียว ซึ่งทำได้โดยการกำหนด Port ของทั้งสองโปรแกรมให้เหมือนกัน แล้วในช่อง IP ให้ใส่เป็นหมายเลข 127.0.0.1 ซึ่งเป็นเลข Local Host IP แต่หากจะต้องการรับ-ส่งภาพ โดยใช้คอมพิวเตอร์สองตัวสามารถทำได้โดยเชื่อมต่อคอมพิวเตอร์ทั้ง 2 เข้า Workgroup เดียวกัน ในช่อง Port ของทั้งสองโปรแกรมให้ระบุเลขเดียวกัน แล้วในโปรแกรมส่วน Client ช่อง IP ให้ใส่เลข IP ของเครื่องที่เปิดโปรแกรม Server อยู่

ขั้นตอนการทดลอง

1. ทำการเปิดโปรแกรมทั้งสองขึ้นมา แล้วระบุเลข Port ของทั้งสองโปรแกรมให้เหมือนกัน ส่วนในช่อง IP ให้ใส่เลข 127.0.0.1 ดังรูป

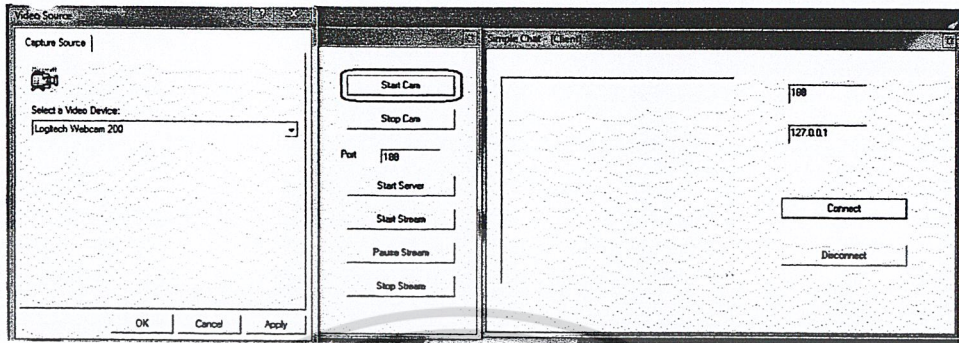


รูปที่ 4.1 ภาพแสดงขั้นตอนที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ที่โปรแกรมส่วน Server กดปุ่ม Start Cam จะมีหน้าต่างให้เลือกกล้องที่จะใช้ในการจับภาพขึ้นมา

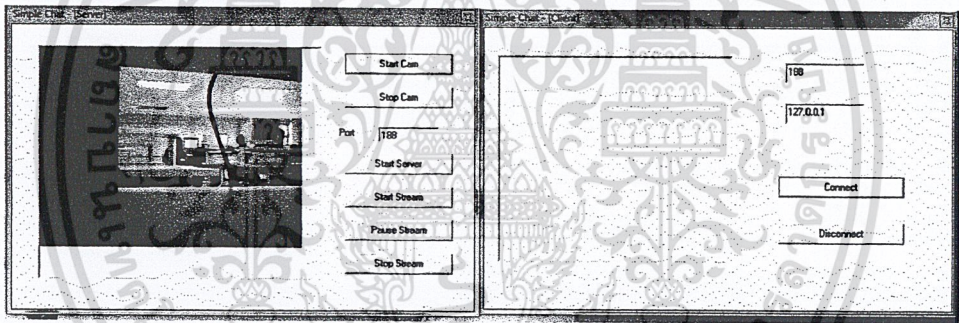
ดังรูป



รูปที่ 4.2 ภาพแสดงขั้นตอนที่ 2

3. เมื่อทำการเลือกกล้องได้แล้วกดปุ่ม OK ที่โปรแกรมส่วน Server จะแสดงภาพขึ้นมาที่ Picture Box

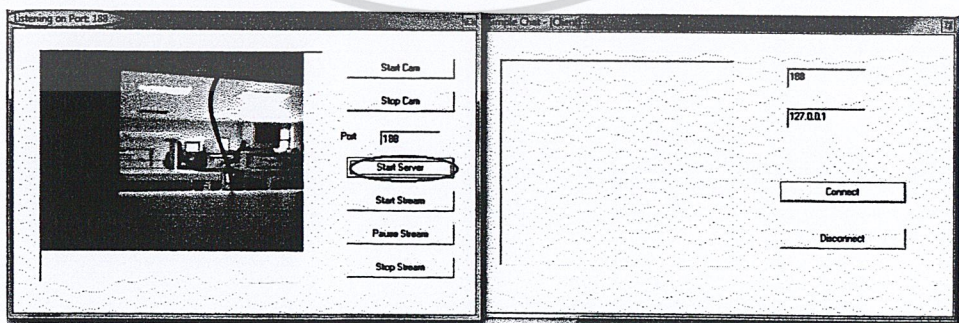
ดังรูป



รูปที่ 4.3 ภาพแสดงขั้นตอนที่ 3

4. กดปุ่ม Start Server ที่โปรแกรมส่วน Server บนหัวโปรแกรมส่วน Server จะขึ้นมาว่า Listening on

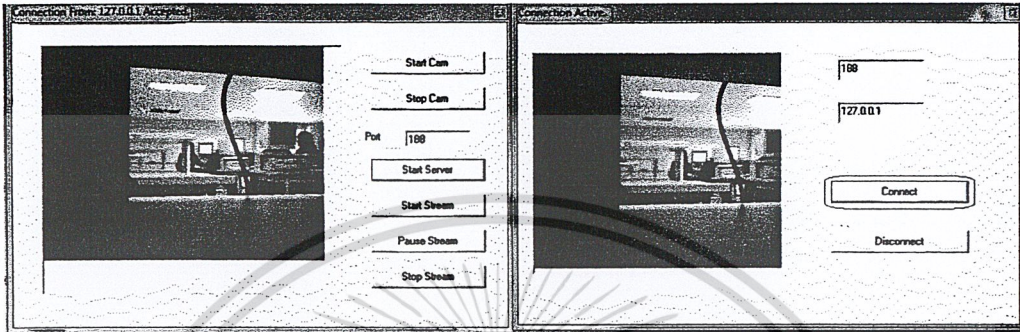
Port : ดังรูป



รูปที่ 4.4 ภาพแสดงขั้นตอนที่ 4

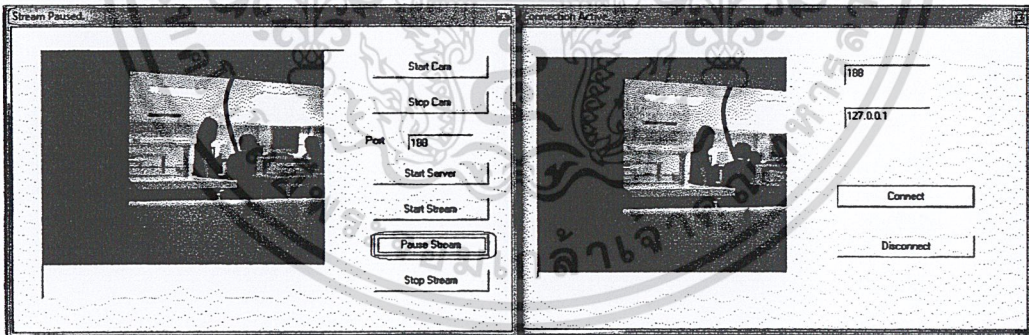
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ที่โปรแกรม Client กดปุ่ม Connect ที่ Picture Box ของส่วนclient จะมีภาพปรากฏขึ้นมาพร้อมกับหัวโปรแกรมส่วน Server จะแสดงข้อความว่า Connection form: ... Accepted และหัวโปรแกรมส่วน Client จะแสดงข้อความว่า Connection Active ดังรูป



รูปที่ 4.5 ภาพแสดงขั้นตอนที่ 5

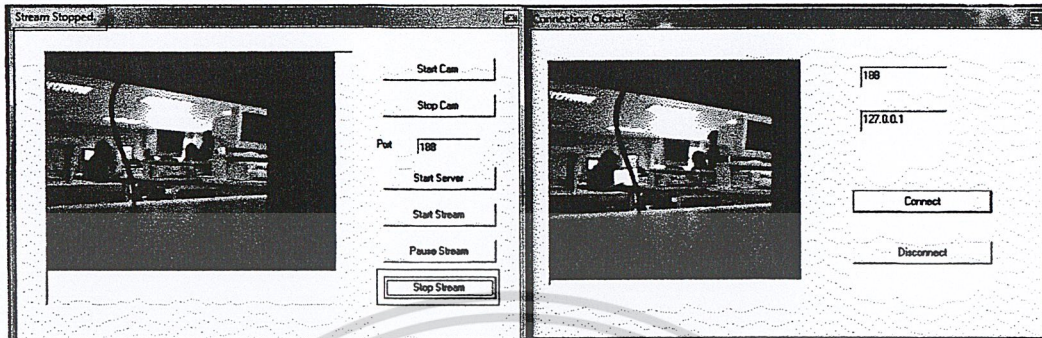
6. เมื่อกดปุ่ม Start Stream โปรแกรมจะเริ่ม Stream ภาพไปยัง Client
7. เมื่อกดปุ่ม Pause Stream โปรแกรม Server จะหยุดการส่งภาพ ไปยังส่วนของโปรแกรม Client และบนส่วนหัวโปรแกรม Server จะขึ้นคำว่า Stream Paused ดังรูป



รูปที่ 4.6 ภาพแสดงขั้นตอนที่ 7

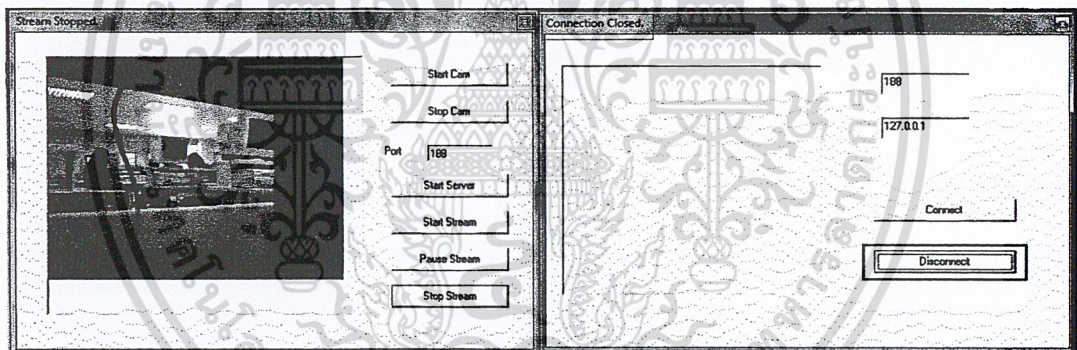
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อกดปุ่ม Stop Stream โปรแกรม Server จะหยุดการส่งภาพไปยังส่วนของโปรแกรม Client และบนส่วนหัวโปรแกรม Server จะขึ้นคำว่า Stream Stopped ดังรูป



รูปที่ 4.7 ภาพแสดงขั้นตอนที่ 8

9. ทำการกดปุ่ม Disconnect ที่โปรแกรมส่วน Client บนภาพหายไปจาก Picture Box ของโปรแกรม ส่วน Client และบนหัวโปรแกรมส่วน Client จะขึ้นว่า Connection Closed



รูปที่ 4.8 ภาพแสดงขั้นตอนที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

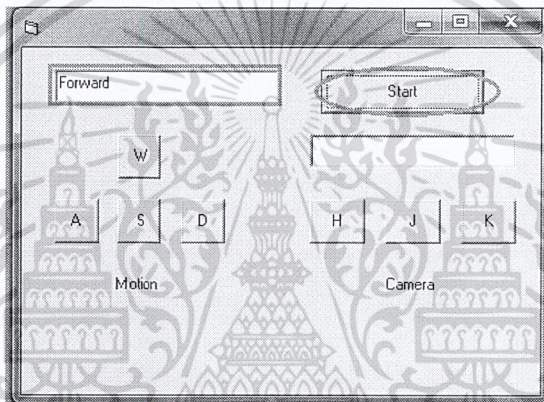
4.1.2 โปรแกรม Visual Basic ในส่วนการควบคุม

เมื่อทำการกดปุ่ม Start บนหน้าต่างโปรแกรมเราจะสามารถกดปุ่มต่างๆเพื่อทำการสั่งการหุ่นยนต์ตามที่เราต้องการได้ดังนี้

ขั้นตอนการทดลอง

1. เมื่อกดปุ่ม W บนคีย์บอร์ด รถจะเคลื่อนที่ไปข้างหน้าและในช่อง Text1 จะแสดงข้อความว่า “Forward”

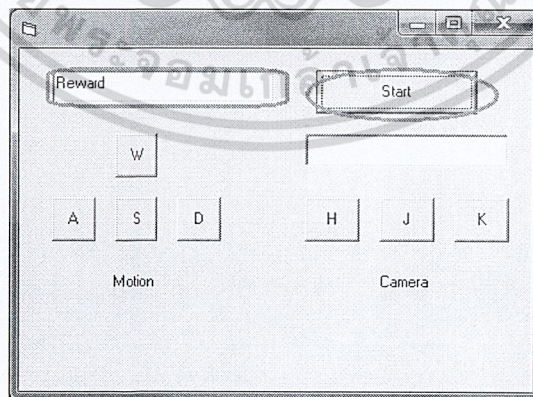
ดังรูป



รูปที่ 4.9 แสดงขั้นตอนที่ 1

2. เมื่อกดปุ่ม S บนคีย์บอร์ด รถจะเคลื่อนที่ไปข้างหลังและในช่อง Text1 จะแสดงข้อความว่า “Reward”

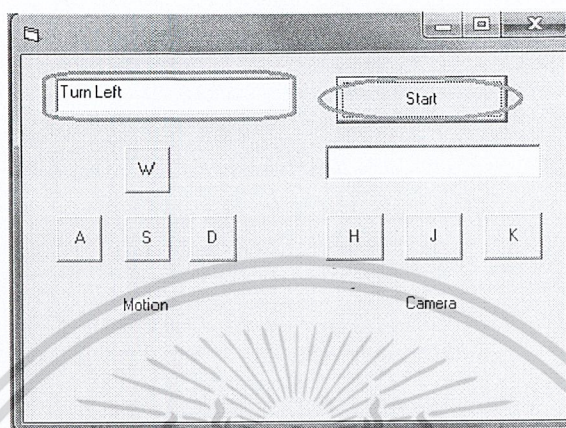
ดังรูป



รูปที่ 4.10 แสดงขั้นตอนที่ 2

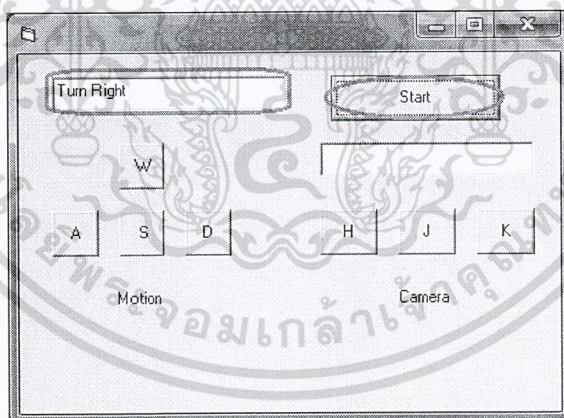
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อกดปุ่ม A บนคีย์บอร์ด รถจะเคลื่อนที่ไปทางซ้ายและในช่อง Text1 จะแสดงข้อความว่า “Turn Left” ดังรูป



รูปที่ 4.11 แสดงขั้นตอนที่ 3

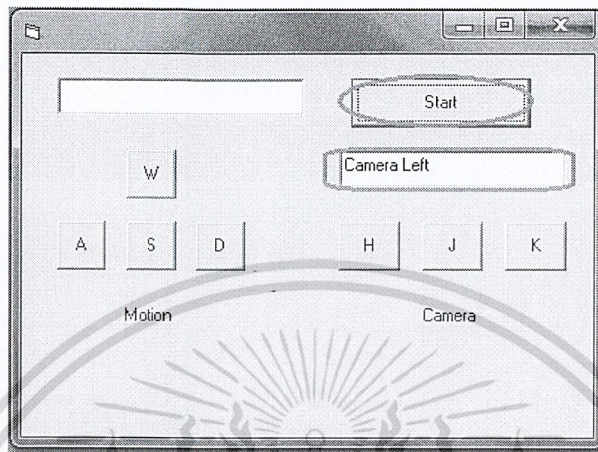
4. เมื่อกดปุ่ม D บนคีย์บอร์ด รถจะเคลื่อนที่ไปทางซ้ายและในช่อง Text1 จะแสดงข้อความว่า “Turn Right” ดังรูป



รูปที่ 4.12 แสดงขั้นตอนที่ 4

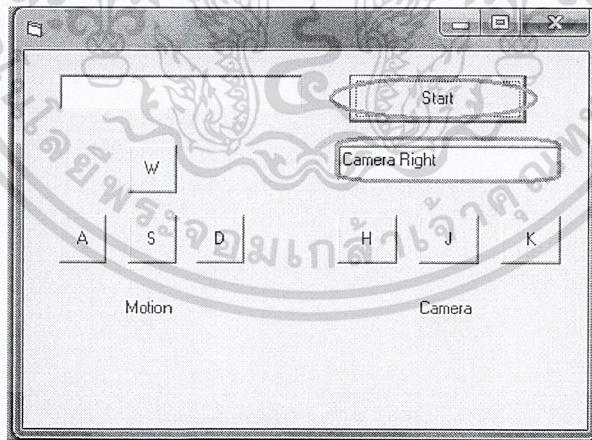
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เมื่อกดปุ่ม H บนคีย์บอร์ด ก็ลองจะหมุนไปทางซ้ายและในช่อง Text2 จะแสดงข้อความว่า “Camera Left” ดังรูป



รูปที่ 4.13 แสดงขั้นตอนที่ 5

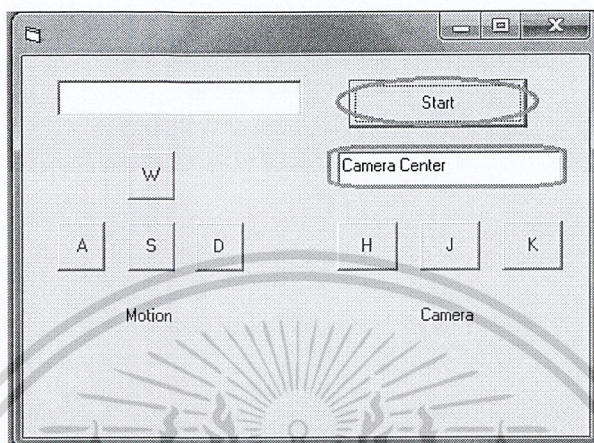
6. เมื่อกดปุ่ม K บนคีย์บอร์ด ก็ลองจะหมุนไปทางขวาและในช่อง Text2 จะแสดงข้อความว่า “Camera Right” ดังรูป



รูปที่ 4.14 แสดงขั้นตอนที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

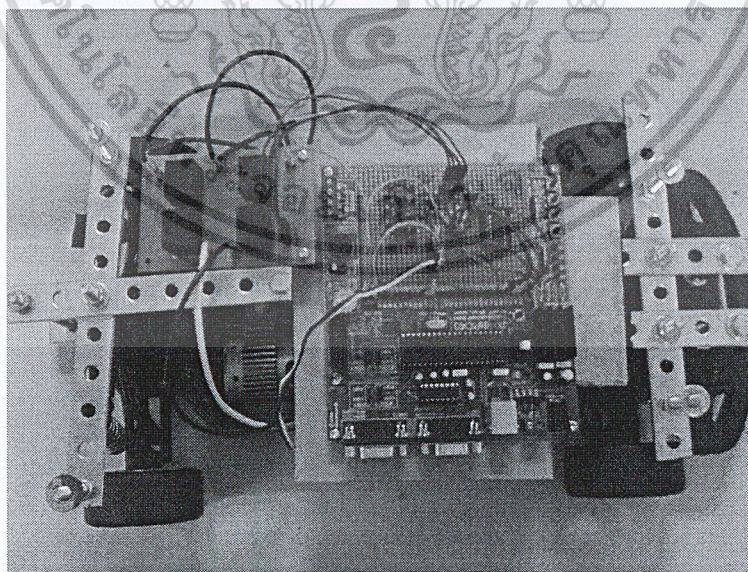
7. เมื่อกดปุ่ม J บนคีย์บอร์ดกล้องจะหมุนกลับมาตรงกลางและในช่อง Text2 จะแสดงข้อความว่า “Camera Center” ดังรูป



รูปที่ 4.15 แสดงขั้นตอนที่ 7

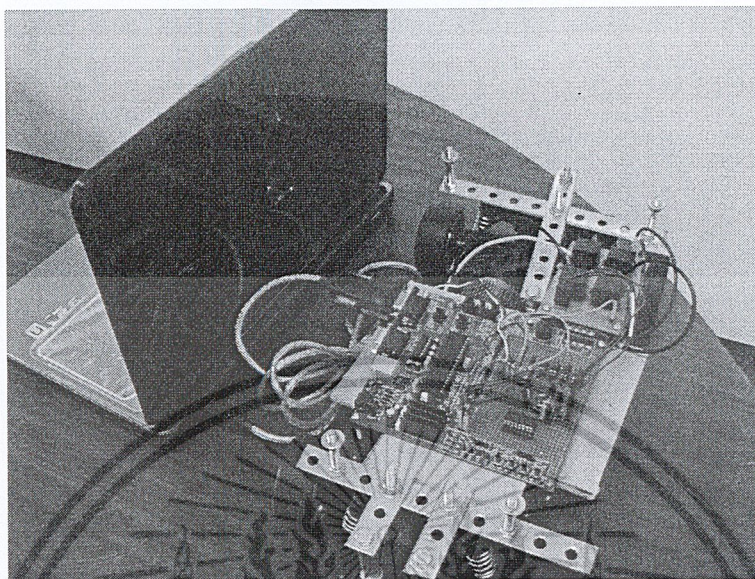
4.1 การทดลองในส่วนการควบคุมการทำงานของหุ่นยนต์ทั้งหมด

ทางผู้จัดทำได้ทำการทดลองวางอุปกรณ์ และทดสอบการทำงานทั้งหมดของหุ่นยนต์ ซึ่งแสดงดังรูปภาพต่อไปนี้

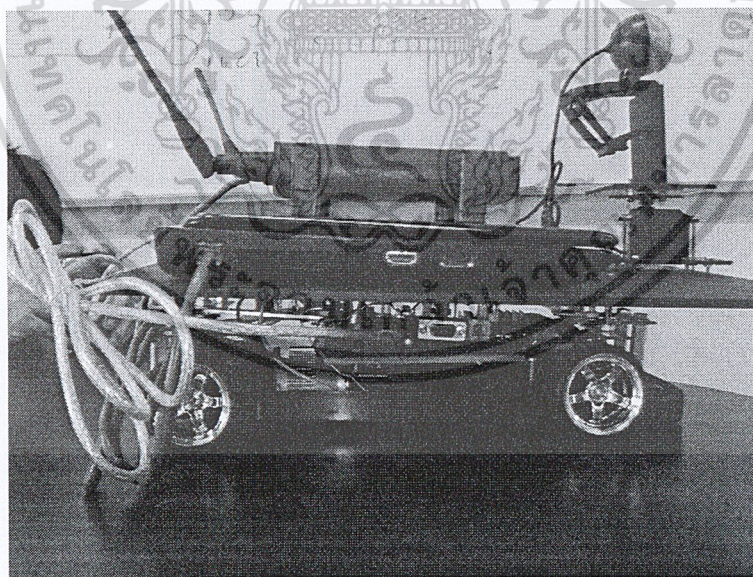


รูปที่ 4.16 แสดงการเชื่อมต่อระหว่างบอร์ด dsPIC และวงจรขับเคลื่อนมอเตอร์กระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

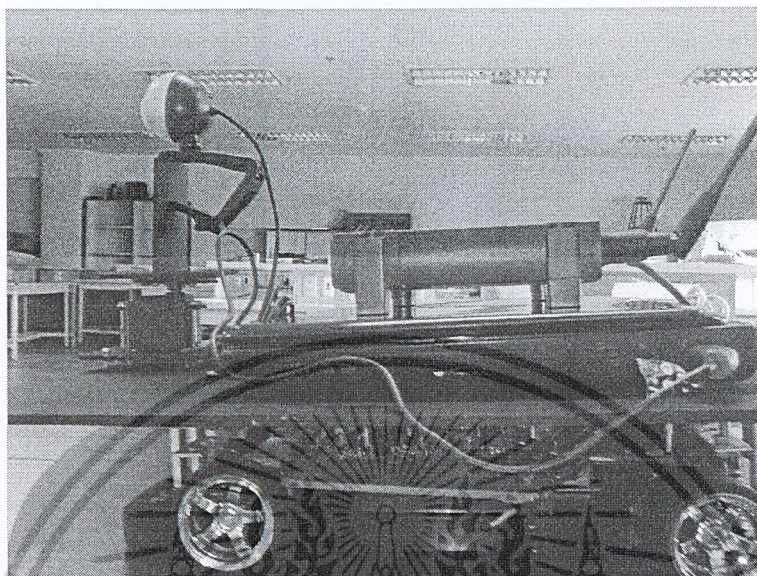


รูปที่ 4.17 แสดงการเชื่อมระหว่างบอร์ด dsPIC และ Computer Notebook ผ่านทาง RS-232

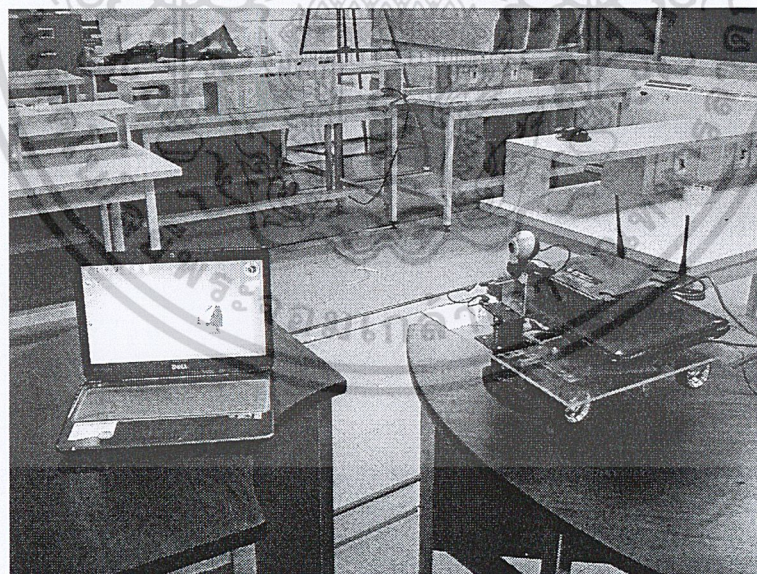


รูปที่ 4.18 แสดงการเชื่อมระหว่างบอร์ด dsPIC และ Computer Notebook เมื่อทำการประกอบเข้ากับตัวรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

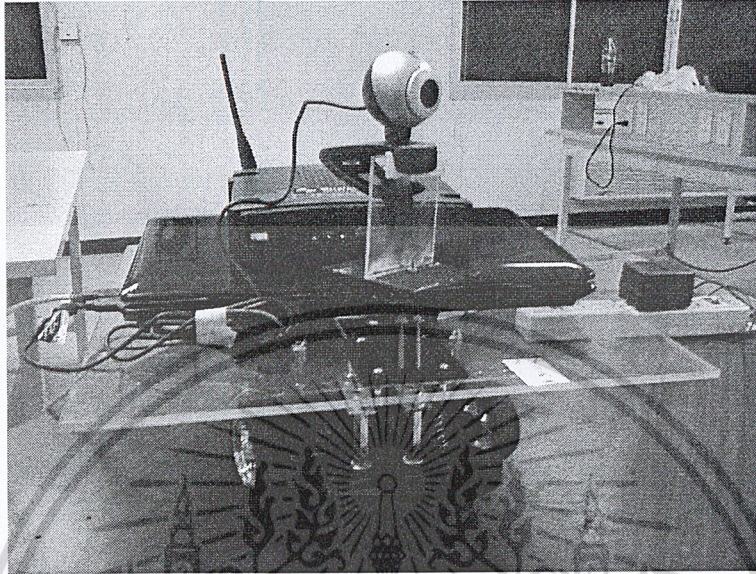


รูปที่ 4.19 แสดงการเชื่อมระหว่างกล้อง Webcam และ Computer Notebook

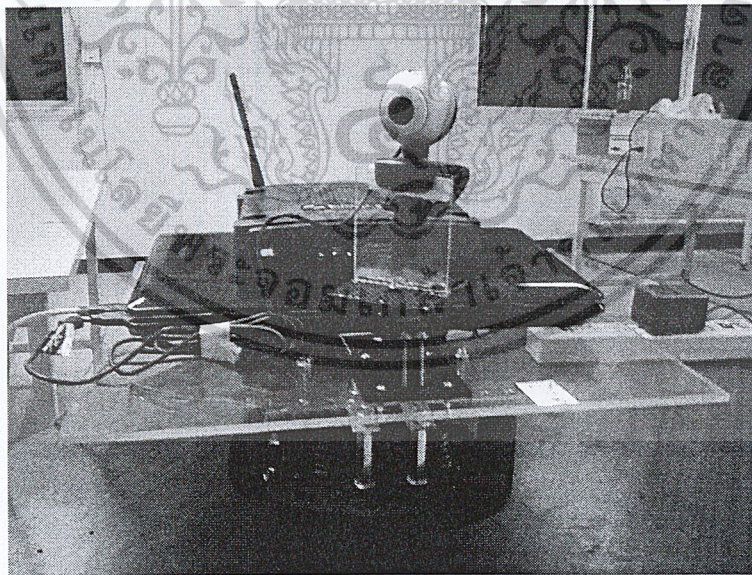


รูปที่ 4.20 หุ่นยนต์บริการสั่งงานจากเครื่อง Client ผ่านทาง Access Point

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 แสดงการสั่งงานให้ตัวรถเลี้ยวซ้าย และหมุนกลิ้งซ้าย



รูปที่ 4.22 แสดงการสั่งงานให้ตัวรถเลี้ยวขวา และหมุนกลิ้งขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

ในการทำปริญญานิพนธ์ฉบับนี้ ได้จัดทำขึ้นเพื่อการควบคุมกระบวนการแบบไร้สาย การควบคุมมอเตอร์กระแสตรง ซึ่งสามารถทำการควบคุม สั่งงาน การเคลื่อนที่ของหุ่นยนต์ ซึ่งใช้ในการส่งข้อมูล และรับข้อมูลแบบไร้สาย โดยผ่านตัวส่ง Access point Wireless – G 2.4GHz การควบคุมผ่านหน้าจอกอมพิวเตอร์ โดยใช้โปรแกรม Visual Basic ควบคุมการทำงาน

จากการทดลองหุ่นยนต์สำรวจไร้สาย พบว่า การควบคุมมอเตอร์กระแสตรงยังทำงานได้ไม่ดีเท่าที่ควร เนื่องจากมอเตอร์กระแสตรงที่ใช้มี rpm สูง เมื่อมีการนำ computer notebook มาวางบนตัวหุ่นยนต์ มอเตอร์กระแสตรงจะมีการกินกระแสที่สูงมาก ซึ่งส่งผลให้แหล่งจ่ายไฟเกิดการร้อนในการทดลองครั้งนี้จึงได้จำลองแค่การเคลื่อนที่ของหุ่นยนต์ ส่วนการควบคุมการเคลื่อนที่ของกล้อง สามารถควบคุมได้ตามทิศทางที่ต้องการ

5.1 ปัญหาที่พบ

1. มอเตอร์กระแสตรงมีการใช้กระแสที่มากเกินไป ทำให้แหล่งจ่ายไฟเกิดการร้อน
2. โปรแกรม Visual Basic ในส่วนของ การรับ – ส่งภาพ ในบางโปรแกรมหยุดการทำงาน
3. โปรแกรม Visual Basic ไม่สามารถทำงานทั้งหมดได้พร้อมกัน

5.2 วิธีแก้ไขปัญหา

1. เปลี่ยนวงจรมอเตอร์ เป็นวงจรที่ขับเคลื่อนด้วย MOTFET ซึ่งสามารถขับ Motor ที่ใช้กระแสสูงได้
2. ทำการเปลี่ยน code การรับ – ส่ง เป็นรูปแบบอื่น หรือ ทำการเปลี่ยนระบบปฏิบัติการเป็น

WindowsXP

3. เปลี่ยนโปรแกรมสั่งงานเป็น Visual C++

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Code Program Visual Basic และ Code dsPIC30F4011

Code Program Visual Basic

Server Code

ในส่วนของ Module

Option Explicit

Public Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA"
 (ByVal lpszWindowName As String, ByVal dwStyle As Long, ByVal X As Long, ByVal Y As Long, ByVal
 nWidth As Long, ByVal nHeight As Long, ByVal hwndParent As Long, ByVal nID As Long) As Long

Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal
 wParam As Long, ByVal lParam As Long, lParam As Any) As Long

Public mCapHwnd As Long

Public ListDisplay As Variant

Public requested As String

Public Clipboard As Variant

Public Const CONNECT As Long = 1034

Public Const DISCONNECT As Long = 1035

Public Const GET_FRAME As Long = 1084

Public Const COPY As Long = 1054

Public Function GetFile(ByVal StrFileName As String) As String

Dim StrFile As Long

Dim GetFil As String

StrFile = FreeFile

Open StrFileName For Binary As #StrFile

GetFil = Space\$(LOF(StrFile))

Get #StrFile, , GetFil

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Close #StrFile
GetFile = GetFil
End Function

```

ในส่วนของ Form

Option Explicit

```

-----
Private Sub cmdStartServer_Click()           //เมื่อกดปุ่ม Start Server
    Winsock.LocalPort = txtPort.Text        //ปรับ Port ของ Winsock ให้เป็นเหมือน
                                             ในของ Port Text
    Winsock.Listen                          //Winsock รอรับค่า
    Me.Caption = "Listen on Port: " & Winsock.LocalPort //หัวโปรแกรมขึ้นว่า Listen on Port ...
End Sub

```

```

-----
Private Sub cmdStartCam_Click()             //เมื่อกดปุ่ม Start Cam
    mCapHwnd = capCreateCaptureWindow("WebcamCapture", 0, 0, 0, 640, 480, picDisplay.hwnd, 0)
                                             //ทำการเชื่อมต่อกับ Webcam
    DoEvents
    SendMessage mCapHwnd, CONNECT, 0, 0    //เชื่อมต่อ Webcam
    TmrPreview.Enabled = True              //Timer ที่ทำหน้าที่ควบคุมการแสดงผลภาพ
                                             ทำงาน
End Sub

```

```

-----
Private Sub cmdStopCam_Click()              //เมื่อกดปุ่ม Stop Cam
    DoEvents: SendMessage mCapHwnd, DISCONNECT, 0, 0
                                             //ยกเลิกการเชื่อมต่อกับ Webcam
    TmrPreview.Enabled = False             //Timer ที่ทำหน้าที่ควบคุมการแสดงผลภาพ
                                             หยุดทำงาน
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub cmdStartStream_Click()
    TmrStream.Enabled = True
    Me.Caption = "Stream Active."
End Sub
-----
Private Sub cmdPause_Click()
    TmrStream.Enabled = False
    Me.Caption = "Stream Paused."
End Sub
-----
Private Sub cmdStopStream_Click()
    TmrStream.Enabled = False
    Winsock.Close
    Me.Caption = "Stream Stopped."
    Kill App.Path & "\temp.bmp"
End Sub
-----
Private Sub Winsock_Close()
    Winsock.Close
    ListDisplay.AddItem "Disconnected"
End Sub
-----
Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    If Winsock.State <> sckClosed Then Winsock.Close

    Winsock.Accept requested
    Me.Caption = "Connect From: " & _
    Winsock.RemoteHostIP & " Accepted. "
End Sub

```

//เมื่อกดปุ่ม Start Stream
//Timer ที่ควบคุมการ Stream ทำงาน
//หัวโปรแกรมขึ้นว่า "Stream Active"

//เมื่อกดปุ่ม Pause Stream
//Timer ที่ควบคุมการ Stream หยุดการทำงาน
//หัวโปรแกรมขึ้นว่า "Stream Pause"

//เมื่อกดปุ่ม Stop Stream
//Timer ที่ควบคุมการ Stream หยุดการทำงาน
//Winsock หยุดการเชื่อมต่อ
//หัวโปรแกรมขึ้นว่า "Stream Stopped"
//ลบ ไฟล์รูปที่สร้างมาเพื่อ Stream

//อธิบายความหายของ Winsock Close
//Winsock หยุดการเชื่อมต่อ
//ตัดการเชื่อมต่อกับ Picture box

//การขอเชื่อมต่อจาก Client
//ถ้าสถานะของ Winsock อยู่ใน
ภาวะ close
//Winsock รับคำขอการเชื่อมต่อ
//หัวโปรแกรมแสดงข้อความ
// "Connect form ... Accept"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)           //อธิบายข้อมูลที่รับมาจาก Client
    Dim data As String                                             //กำหนดdata เป็นข้อมูลแบบ String
    Winsock.GetData data                                           //Winsock รับ data เป็นข้อมูล
    If data = "CAMSTREAM" Then                                     //ถ้า data คือ CAMSTREAM
    Else                                                            //นอกเหนือจากนี้
        Winsock.Close                                             //Winsock ไม่ทำงาน
    End If
End Sub

```

```

Private Sub TmrPreview_Timer()                                     //อธิบายการทำงานของ TmrPreview
    SendMessage mCapHwnd, GET_FRAME, 0, 0                         //ส่งจำขอเชื่อมต่อกับ Webcam
    SendMessage mCapHwnd, COPY, 0, 0
    picDisplay.Picture = Clipboard.GetData                       //แสดงภาพที่ได้มาจาก Webcam บน
Clipboard
    Clipboard.Clear
End Sub

```

```

Private Sub TmrStream_Timer()
    Dim BinFile As String
    If Winsock.State = sckConnected Then
        SavePicture picDisplay.Picture, App.Path & "\temp.bmp"
        DoEvents
        BinFile = GetFile(App.Path & "\temp.bmp")
        DoEvents
        Winsock.SendData BinFile
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub Form_Unload(Cancel As Integer)

 If Dir(App.Path & "\temp.bmp", vbNormal) = "temp.bmp" Then Kill App.Path & "\temp.bmp"

End Sub



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Client Code

ในส่วนของ Module

Option Explicit

Public Function SetFile(ByVal StrFileName As String, ByVal FileData As String)

Dim StrFile As Long

StrFile = FreeFile

Kill StrFileName

Open StrFileName For Binary Access Write As #StrFile

Put #StrFile, , FileData

Close #StrFile

End Function

(ทั้งหมดนี้คือส่วนแปลงไฟล์ Binary กลับมาเป็นภาพ)

ในส่วนของ Form

Option Explicit

Dim iData As String

Private Sub cmdConnect_Click()

Winsock.Close

Winsock.RemoteHost = txtIP.Text

Winsock.RemotePort = txtPort.Text

Winsock.Connect

End Sub

Private Sub cmdDisconnect_Click()

Winsock.Close

Me.Caption = "Connect Closed."

picDisplay.Picture = Nothing

End Sub

//เมื่อกดปุ่ม Connect

//Winsock หยุดการทำงาน

//บันทึกค่า IP ใน Winsock ตามที่ระบุในช่อง IP Text

//บันทึกค่า Port ใน Winsock ตามที่ระบุใน Port Text

//เชื่อมต่อ Winsock

//เมื่อกดปุ่ม Disconnect

//Winsock หยุดการทำงาน

//หัวโปรแกรมขึ้นว่า "Connect Closed"

//ไม่มีการแสดงภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Winsock_Connect()                                //อธิบายความหมายของ Winsock Connect
    Winsock.SendData "CAMSTREAM"                            //Winsock ส่งข้อมูล "CAMSTREAM"
    Me.Caption = "Connection Active"                       //หัวข้อโปรแกรมขึ้นว่า "Connection Active"
End Sub

```

```

Private Sub Winsock_Close()                                //อธิบายความหมายของ Winsock Close
    Winsock.Close                                           //Winsock หยุดการทำงาน
    Me.Caption = "Connection Closed"                       //หัวข้อโปรแกรมขึ้นว่า "Connection Closed"
End Sub

```

```

Private Sub Winsock_DataArrival(ByVal bytesTotal As Long)  //อธิบายความหมายของ Winsock
DataArrival
    Dim data As String                                     //กำหนดตัวแปร data เป็น string
    Winsock.GetData data                                  //Winsock รับข้อมูล data
    iData = iData + data                                 //กำหนดให้ iData = iData + data
    DoEvents
    SetImg iData                                         //กำหนดให้ iData เป็นภาพ
End Sub

```

```

Sub SetImg(Imgdata As String)                             //อธิบายความหมายของ SetImg
    SetFile App.Path & "\temp.bmp", Imgdata             //เรียกใช้ฟังก์ชันการแปลงไฟล์ Binary เป็นภาพ
    DoEvents
    picDisplay.Picture = LoadPicture(App.Path & "\temp.bmp") //ทำการแสดงภาพบน โปรแกรมส่วน Client
    iData = ""
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub Form_Unload(Cancel As Integer)

 If Dir(App.Path & "\temp.bmp", vbNormal) = "temp.bmp" Then Kill App.Path & "\temp.bmp"

End Sub

//เป็นส่วนที่ใช้สั่งการแสดงผลภาพบนโปรแกรมClient



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code Program Visual Basic ในส่วนของการควบคุม

Option Explicit

Private Sub Form_Load()

MSComm1.Settings = "9600,N,8,1" //กำหนดค่า default

MSComm1.CommPort = 4 //กำหนดการติดต่อผ่านพอร์ต COM4

MSComm1.InputLen = 1 //กำหนดชนิดของข้อมูล Input

MSComm1.PortOpen = True //กำหนดการเปิด Port

MSComm1.RThreshold = 1 //กำหนดการเกิด Event ของ Port

End Sub

Private Sub Start_KeyDown(KeyCode As Integer, Shift As Integer) //เมื่อมีการกดปุ่มคีย์บอร์ด

Select Case KeyCode //แยกเป็นกรณีการกดแต่ละปุ่ม

Case vbKeyA //เมื่อกดปุ่ม "A"

Text1.Text = "Turn Left" //ในช่อง Text1 ขึ้นข้อความว่า "Turn Left"

MSComm1.Output = "A" //ส่งข้อมูล "A" ออกไปทาง Port

Case vbKeyD //เมื่อกดปุ่ม "D"

Text1.Text = "Turn Right" //ในช่อง Text1 ขึ้นข้อความว่า "Turn Right"

MSComm1.Output = "D" //ส่งข้อมูล "D" ออกไปทาง Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case vbKeyH //เมื่อกดปุ่ม "H"

Text2.Text = "Camera Left" //ในช่อง Text2 ขึ้นข้อความว่า "Camera Left"

MSComm1.Output = "H" //ส่งข้อมูล "H" ออกไปทาง Port

Case vbKeyJ //เมื่อกดปุ่ม "J"

Text2.Text = "Camera Center" //ในช่อง Text2 ขึ้นข้อความว่า "Camera Center"

MSComm1.Output = "J" //ส่งข้อมูล "J" ออกไปทาง Port

Case vbKeyK //เมื่อกดปุ่ม "K"

Text2.Text = "Camera Right" //ในช่อง Text2 ขึ้นข้อความว่า "Camera Right"

MSComm1.Output = "K" //ส่งข้อมูล "K" ออกไปทาง Port

Case vbKeyW //เมื่อกดปุ่ม "W"

Text1.Text = "Forward" //ในช่อง Text1 ขึ้นข้อความว่า "Forward"

MSComm1.Output = "W" //ส่งข้อมูล "W" ออกไปทาง Port

Case vbKeyS //เมื่อกดปุ่ม "S"

Text1.Text = "Reward" //ในช่อง Text1 ขึ้นข้อความว่า "Reward"

MSComm1.Output = "S" //ส่งข้อมูล "S" ออกไปทาง Port

End Select

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub move_KeyUp(KeyCode As Integer, Shift As Integer) //เมื่อปล่อยปุ่ม
Text1.Text = "" //ในช่อง Text1 ไม่มีข้อความ
Text2.Text = "" //ในช่อง Text2 ไม่มีข้อความ
End Sub

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code dsPIC30F4011

```

#include <p30f4011.h> // dsPIC30F4011 MPU Register
#include <uart.h> // UART module
#include <OutCompare.h> // Outcompare module

//-----Config fuse
_FOSC(CSW_FSCM_OFF & XT_PLL4); // Sw Disable, Mon Disable,XT w/PLL 4x
_FWDT(WDT_OFF); // Watchdog timer off

//-----Calc Buad Rate Generator
#define Fcy 4000000.0 // Fosc 4.0MHz
#define BAUD_RATE 9600.0 // Baud Rate 9600 bps
#define BAUD_RATE_GEN (Fcy/(16.0*BAUD_RATE))-1 // Baud Rate Generator

void _ISR_U1TXInterrupt(void)
{
    _U1RXIF = 0; // Clear TX interrupt flag
}

void _ISR_U1RXInterrupt(void)
{
    _U1RXIF = 0; // Clear RX interrupt flag
}

void _ISR_PWMInterrupt(void)
{
    _PWMIF = 0; // Clear PWM interrupt flag
}

//Delay 1ms (XT w/PLL 4x)
void Delay_MS(unsigned int ms)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

{
    unsigned int i;

    for (; ms>0; ms--)
        for (i=0; i<728; i++)
            Nop ();
}

//-----DC Motor
void Motor_Forward()
{
    _LATB0 = 1;           //configuration for RB0 to high
    _LATB1 = 0;           //configuration for RB1 to low
    _LATB2 = 0;           //configuration for RB2 to low
    _LATB3 = 1;           //configuration for RB3 to high
}

void Motor_Backward()
{
    _LATB0 = 0;           //configuration for RB0 to low
    _LATB1 = 1;           //configuration for RB1 to high
    _LATB2 = 1;           //configuration for RB2 to high
    _LATB3 = 0;           //configuration for RB3 to low
}

void Motor_Stop()
{
    _LATB0 = 0;           //configuration for RB0 to low
    _LATB1 = 0;           //configuration for RB1 to low
    _LATB2 = 1;           //configuration for RB2 to high
    _LATB3 = 1;           //configuration for RB3 to high
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

//-----Servo_Open
void Servo_Open(void)
{
    CloseOC1();                // Close Timer1
    CloseOC2();                // Close Timer2
    OpenOC1(OC_IDLE_CON &    // continue operation in idle mode
            OC_TIMER2_SRC &  // Timer2 source for OutCom1
            OC_PWM_FAULT_PIN_DISABLE, // PWM mode on,fault pin disable
            0,                // OCxRS data secondary
Register
            0);              // ORxR Data Main Register
    TMR2 = 0x0000;            // Clear Timer2
    T2CON = 0x8030;          // Timer2 ON, Prescale 1:256

    OpenOC2(OC_IDLE_CON &    // continue operation in idle mode
            OC_TIMER2_SRC &  // Timer2 source for OutCom1
            OC_PWM_FAULT_PIN_DISABLE, // PWM mode on,fault pin disable
            0,                // OCxRS data secondary
Register
            0);              // ORxR Data Main Register
    TMR2 = 0x0000;            // Clear Timer2
    T2CON = 0x8030;          // Timer2 ON, Prescale 1:256
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----Servo_Direction1_Steering
void Servo_Left()
{
    PR2 = 81;
    SetDCOC1PWM(18);                // Pulse High 1.0ms
}
void Servo_Right()
{
    PR2 = 81;
    SetDCOC1PWM(29);                // Pulse High 2.0ms
}
void Servo_Middle()
{
    PR2 = 81;
    SetDCOC1PWM(23);                // Pulse High 1.5ms
}
//-----Servo_Direction2_Camera
void Servo_Cameraleft()
{
    PR2 = 81;
    SetDCOC2PWM(18);                // Pulse High 1.0ms
}
void Servo_Cameraright()
{
    PR2 = 81;
    SetDCOC2PWM(29);                // Pulse High 2.0ms
}
void Servo_Cameramiddle()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    PR2 = 81;
    SetDCOC2PWM(23);                // Pulse High 1.5ms
}

//-----Initial UART
void Uart1_Init()
{
    unsigned int configU1MODE;
    unsigned int configU1STA;
    unsigned int BaudRate;

    CloseUART1();                    // Close UART1

    configU1MODE = UART_EN &        // enable module
                  UART_IDLE_CON &  // Work in IDELE mode
                  //UART_RX_TX &   // Communication through the
normal pins
                  //UART_ALTRX_ALTTX & // Communication through ALT
bit
                  UART_DIS_WAKE & // Disable Wake-up on START
                  UART_DIS_LOOPBACK & // Loop back disable
                  UART_DIS_ABAUD & // Input to capture module
                  UART_NO_PAR_8BIT & // no parity 8 bit
                  UART_1STOPBIT;    // 1 stop bit

    configU1STA = UART_INT_TX_BUF_EMPTY & // Interrupt on TXBUF empty
                  UART_TX_PIN_NORMAL & // UART TX pin normally

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UART_INT_RX_CHAR & // Interrupt every char
reciveed

UART_ADR_DETECT_DIS & // address detect disable
UART_RX_OVERRUN_CLEAR; // RX buffer Overrun
clear

BaudRate = BAUD_RATE_GEN; // Baudrate 9600 pbs

OpenUART1 (configU1MODE, configU1STA, BaudRate);
}
/* pototype section */
void init_uart(void); // Initial UART1 Function

int main(void)
{
char rx_data; // Character Receive Buffer

TRISD = 0; // set Port D to output
TRISB = 0; // set Port B to output
LATD = 0; // clear port D
LATB = 0; // clear port B

Servo_Open(); // Open Servo
Uart1_Init(); // Initial UART = 9600,N,8,1

while(1) // Loop Continue
{

//UART1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (DataRdyUART1()) // If RX Receive Complete
{
  rx_data = ReadUART1(); // Get UART1 Data Receive
  if (rx_data == 'A') // If Receive "A"
  {
    Servo_Left(); // call Function
    Delay_MS(2000);
    Servo_Middle();

    while(BusyUART1()); // Wait putsUART1 Complete
  }
  else if (rx_data == 'D') // If Receive "D"
  {
    Servo_Right(); // call Function
    Delay_MS(2000);
    Servo_Middle();

    while(BusyUART1()); // Wait putsUART1 Complete
  }
  else if (rx_data == 'W') // If Receive "W"
  {
    Motor_Forward(); // call Function
    Delay_MS(2000);
    Motor_Stop();

    while(BusyUART1()); // Wait putsUART1 Complete
  }
  else if (rx_data == 'S') // If Receive "S"
  {
    Motor_Backward(); // call Function
    Delay_MS(2000);
    Motor_Stop();
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    while(BusyUART1());              // Wait putsUART1 Complete
  }
else if (rx_data == 'H')            // If Receive "H"
{
    Servo_Cameraleft();             // call Function
    while(BusyUART1());             // Wait putsUART1 Complete
}
else if (rx_data == 'J')            // If Receive "J"
{
    Servo_Cameramiddle();           // call Function
    while(BusyUART1());             // Wait putsUART1 Complete
}
else if (rx_data == 'K')            // If Receive "K"
{
    Servo_Cameraright();            // call Function
    while(BusyUART1());             // Wait putsUART1 Complete
}
}
}
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์



MICROCHIP

dsPIC30F4011/4012

Data Sheet

High Performance
Digital Signal Controllers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JX-dsPIC40

40-pin dsPIC microcontroller Project board

Features

- Use a 40-pin DIP dsPIC30F device, dsPIC30F4011 on-board Clock rate of 4MHz
- Debugging and programming via LCD jack with ICDX-30 or ICD2 (optional)
- Select PROG and RUN mode with a switch.
- Prepare 5-ports as : RB0-RB8, RC13-RC14, RD0-RD3, RE0-RE5, RE8 and RF0-RF6
- +5V 800mA regulated DC supply on-board
- RESET switch
- LCD modul connection : D4-D7 with RD0-RD3, RS with RF0 and E with RF1 port pin.
- 2 of RS-232 serial port interface
 - UART1 connect with RC13 and RC14, selected by jumper
 - UART2 connect with RF4 and RF5, selected by jumper
- Free experimental device as :
 - 4-LED monitor active logic "high"
 - Piezo speaker
 - 4-Push button switches with pull-up resistors
 - 4-points Terminal block for connect with QB (Quadrature Encoder Input)
 - 0-5V reference source for A/D testing
- Proto area 3x2.5 Inches with 170 solder pads. This area support medium size breadboard (KH-102 from K&H)
- Required +9Vdc 500mA supply (up to +12Vdc)

Packing List

- Experiment board
- Documentation

Optional :

- +12V 500mA DC adapter
- ICDX-30 Debugger/Programmer for PIC/dsPIC with DC adapter
- RS-232 serial port cable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 • JX-dsPIC40 project board technical info

1. JX-dsPIC40 schematic diagram

Figure 1 is complete schematic diagram of JX-dsPIC40 project board. The dsPIC chip on-board is dsPIC30F4011. It has 29 input/output port pins.

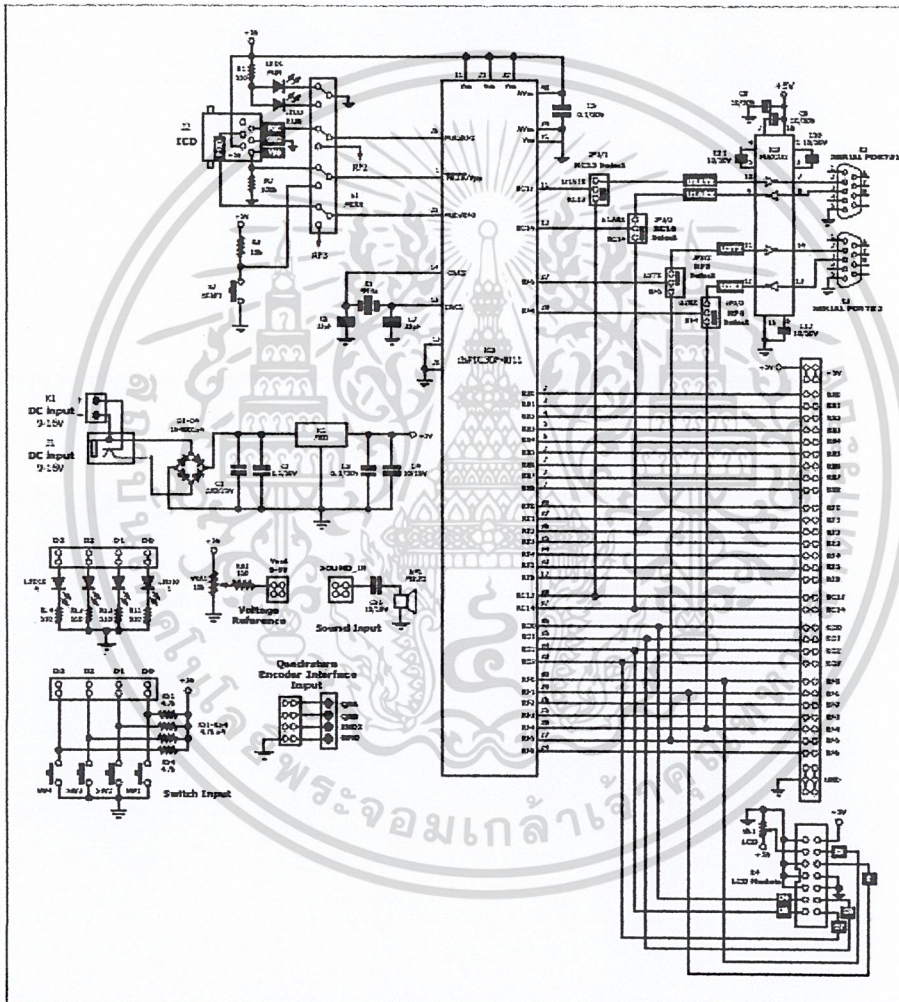


Figure 1 JX-dsPIC40 Project board schematic diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

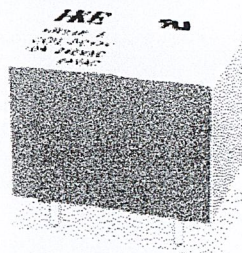
HRS4E Relay

1. COIL DATA

1-1. Nominal Voltage	3 VDC to 48 VDC
1-2. Coil Resistance	Refer to Table 1
1-3. Operate Voltage	Refer to Table 1
1-4. Release Voltage	Refer to Table 1
1-5. Nominal Power Consumption	380 mW

2. CONTACT DATA

2-1. Contact Arrangement	1 Form A , 1 Form C
2-2. Contact Material	AuCdo
2-3. Contact Rating	12A 120VAC / 24VDC 10A 240VAC / 24VDC
2-4. Max. Switching Voltage	110 VDC / 250 VAC
2-5. Max. Switching Current	15A
2-6. Max. Switching Power	2500 VA , 240W
2-7. Contact Resistance (Initial)	1.000 m Ω , at 6VDC 1A
2-8. Life Expectancy	Electrical 100,000 operations at nominal load Mechanical 10,000,000 operations

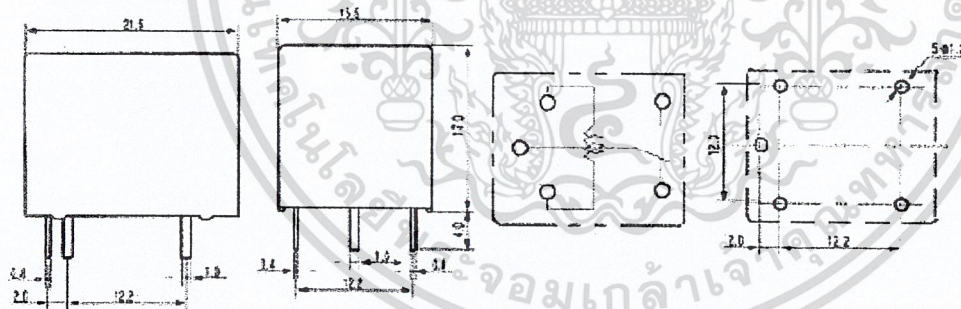


HRS4E Relay

3. GENERAL DATA

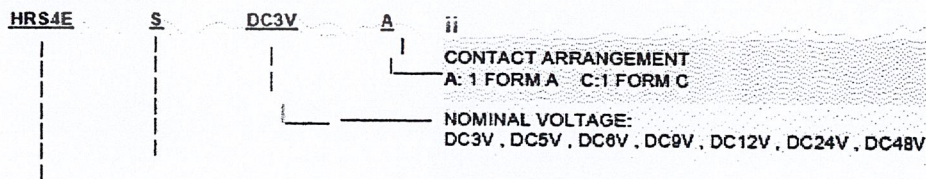
3-1. Insulation Resistance	Min. 100M Ω , at 500 VDC
3-2. Dielectric Strength	1.000VAC , 1min between open contacts 2.500VAC , 1min between contacts and coil
3-3. Operate Time	Max. 10ms
3-4. Release Time	Max. 5ms
3-5. Temperature Range	-30 to +70 μ e
3-6. Shock Resistance	10G
3-7. Vibration Resistance	10 - 55 Hz , Amplitude 1.5mm
3-8. Weight	10 gr.
3-9. Safety Standard	Conform to VDE0110 AND 0435 , UL , CSA

4. DIMENSIONS (in mm)



ii

5. ORDERING CODE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NPN General Purpose Transistors
 **Lead(Pb)-Free**
TO-92
**1. EMITTER
2. BASE
3. COLLECTOR**

ABSOLUTE MAXIMUM RATINGS (Ta=25 °C)

Rating	Symbol	Value	Unit
Collector-Emitter Voltage	V _{CEO}	25	V _{dc}
Collector-Base Voltage	V _{CBO}	40	V _{dc}
Emitter-Base Voltage	V _{EBO}	5.0	V _{dc}
Collector Current	I _C	500	mA _{dc}
Total Device Dissipation Ta=25 °C	P _D	0.625	W
Junction Temperature	T _j	150	°C
Storage Temperature	T _{stg}	-55 to +150	°C

ELECTRICAL CHARACTERISTICS

Characteristics	Symbol	Min	Max	Unit
Collector-Emitter Breakdown Voltage (I _C = 0.1 mA _{dc} , I _B =0)	V _{(BR)CEO}	25	-	V _{dc}
Collector-Base Breakdown Voltage (I _C = 100 μA _{dc} , I _E =0)	V _{(BR)CBO}	40	-	V _{dc}
Emitter-Base Breakdown Voltage (I _E = 100 μA _{dc} , I _C =0)	V _{(BR)EBO}	5.0	-	V _{dc}
Collector Cutoff Current (V _{CE} = 20 V _{dc} , I _B =0)	I _{CE0}	-	0.1	μA _{dc}
Collector Cutoff Current (V _{CB} = 40 V _{dc} , I _E =0)	I _{CBO}	-	0.1	μA _{dc}
Emitter Cutoff Current (V _{EB} = 3.0V _{dc} , I _C =0)	I _{EBO}	-	0.1	μA _{dc}

WEITRON
<http://www.weitron.com.tw>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- dsPIC Microcontroller ด้วยโปรแกรมภาษา C , นคร ภัคดีชาติ , ชัยวัฒน์ ถิ่มพรจิตรวิไล
บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
- คู่มือ Visual Basic 6 ฉบับทฤษฎี พีชผล , พิชิต สันติกุลานนท์ , พร้อมเลิศ หล่อวิจิตร , พิมพ์ครั้งที่ 11
บริษัท โปรวิชั่น จำกัด
- สิริตม์ บรุพาแสงสุรย์ , สุพจน์ เอนกรนโรจน์กุล และ อภิวัฒน์ วันคา. 2550. “ระบบควบคุมไร้สาย”.
ปริญญาานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้า
คุณทหารลาดกระบัง.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้