

สถานจอดรถอัจฉริยะ
INTELLIGENT CAR PARKING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ลานจอดรถอัจฉริยะ
INTELLIGENT CAR PARKING



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะของหน่วยงานนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามให้ข้อมูลไปสู่อื่นๆ และต้องนำเอกสารฉบับนี้คืนเจ้าขอเอกสารที่แท้จริงที่มีการนำไปใช้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

INTELLIGENT CAR PARKING



TARINEE SURACHAREONCHAIKUL
PANWADEE CHANPANICH
SAROCHA KETKAEW

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากสถาบันฯ
เอกสารทุกครั้งที่มีการนำไปใช้

FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

ปริญญาโทบริหารศึกษา 2556

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ลานจอดรถอัจฉริยะ
INTELLIGENT CAR PARKING

ผู้จัดทำ นางสาวธาริณี สุรเจริญชัยกุล 53010734
 นางสาวพรรณวดี จันทร์พานิช 53011059
 นางสาวสโรชา เกษแก้ว 53011651



.....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลานจอดรถอัจฉริยะ

โดย

นางสาวธาริณี สุระเจริญชัยกุล 53010734

นางสาวพรรณวดี จันทร์พานิช 53011059

นางสาวสโรชา เกษแก้ว 53011651

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.วรวงศ์ ตั้งศรีรัตน์

ปีการศึกษา 2556

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการนำเสนอแนวคิดในการออกแบบ และสร้างระบบลานจอดรถ โดยแสดงผลบนหน้าจอคอมพิวเตอร์ โครงสร้างของระบบประกอบด้วยคอมพิวเตอร์ บอร์ด ไมโครคอนโทรลเลอร์อาดูโน่ อัลตราโซนิกเซนเซอร์ เซอร์โวมอเตอร์ และวงจรอิเล็กทรอนิกส์ ประมวลผลสัญญาณ จุดมุ่งหมายของการทำโครงการนี้คือให้ผู้ใช้สามารถเห็นภาพลานจอดรถก่อนที่จะนำรถเข้าไปจอด และสามารถกดจองตำแหน่งช่องจอดรถที่ยังว่างอยู่ ผ่านทางบอร์ดแสดงผลอิเล็กทรอนิกส์ ควบคุมด้วยโปรแกรมไมโครคอนโทรลเลอร์ ซึ่งเป็นการอำนวยความสะดวก และประหยัดเวลาแก่ผู้ใช้ในการหาที่จอดรถมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTELLIGENT CAR PARKING

By

Ms.Tarinee Suracharoenchaikul 53010734

Ms.Panwadee Chanpanich 53011059

Ms.Sarocha Ketkaew 53011651

Advisor

Assoc.Prof.Dr. Worapong Tangsirat

Academic Year 2013

ABSTRACT

This thesis presents the design and construction of an intelligent car parking systems. The presented structure consists of a computer, operated with Arduino microcontroller board, as well as signal processing electronic circuit, i.e., ultrasonic sensor, servo motor and its driver. The major goal of this project is to establish a program via Visual Basic that can indicate and reserve available parking slots from electronic displaying board. The project provides the advantages of convenience and saving for the drivers in finding available parking slots.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปฏิญานพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดีเพราะได้รับความช่วยเหลือเป็นอย่างดี จากรองศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์ ที่ได้กรุณาให้คำปรึกษา และคำแนะนำที่ดีมาโดยตลอดตั้งแต่ต้นรวมถึงความช่วยเหลืออื่น ๆ ที่เป็นประโยชน์ต่อโครงการ ผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์ และขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณคณาจารย์ และเจ้าหน้าที่ทุกท่านของสาขาวิชาวิศวกรรมการวัดและควบคุม ที่ท่านให้ความรู้มาโดยตลอดในระยะเวลาสี่ปีที่ผ่านมา รวมทั้งอำนวยความสะดวก และเอื้อเฟื้ออุปกรณ์ต่าง ๆ

ขอขอบคุณพี่ ๆ เพื่อน ๆ และน้อง ๆ ทุกคนในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยเตือน คอยช่วยเหลือ และเป็นกำลังใจให้เสมอมา

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยห่วงใย เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่อง ตลอดจนเป็นแรงผลักดันให้ปฏิญานพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

ผู้จัดทำ

นางสาวธารณี สรเจริญชยกุล

นางสาวพรรณวดี จันทร์พานิช

นางสาวสโรชา เกษแก้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.1 คุณสมบัติของวิซวลเบสิก	16
2.6.2 ส่วนประกอบของวิซวลเบสิก	17
2.6.3 ขั้นตอนพื้นฐานในการเขียนโปรแกรมด้วยวิซวลเบสิก	19
2.6.3.1 สร้าง, ย้าย, เปลี่ยนขนาด และลบคอนโทรลบนฟอร์ม	19
2.6.3.2 กำหนดคุณสมบัติของคอนโทรล	20
2.6.3.3 เขียนโปรแกรมตอบสนองต่อเหตุการณ์	20
2.6.3.4 รันโปรแกรมประยุกต์	20
2.6.3.5 บันทึกโปรแกรม	20
2.7 เซอร์ไวโมเตอร์	21
2.7.1 ลักษณะของเซอร์ไวโมเตอร์	21
2.7.2 หลักการทำงานของเซอร์ไวโมเตอร์	21
บทที่ 3 การคำนวณและการสร้าง	23
3.1 โครงสร้างของระบบในส่วนต่าง ๆ ของโครงการ	27
3.1.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	27
3.1.2 อัลตราโซนิกเซนเซอร์ HC-SR04	27
3.1.3 หน้าจอแสดงผลที่ออกแบบโดยวิซวลเบสิก	28
3.1.3.1 หน้าจอแสดงผลลานจอดรถ	28
3.1.3.2 หน้าจอสำหรับใส่รหัสเปิดปิดไม้กั้น	29
3.1.4 เซอร์ไวโมเตอร์ SG90	30
3.2 ออกแบบโครงสร้างลานจอดรถ	31
บทที่ 4 การทดลองและผลการทดลอง	33
4.1 แบบจำลองการทำงาน	33
4.2 การแสดงภาพของตำแหน่งที่จอดรถยนต์	34
4.2.1 ช่องจอดรถว่าง	34
4.2.2 ช่องจอดรถมีการจอง	35
4.2.3 ช่องจอดรถมีรถเข้าจอด	37

เอกสารนี้เป็นเอกสารที่การทดลองวงจรตรวจจับรถเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ 40 การค้า
ไม่ว่ากรณีใดๆทั้ง 4.4 การทดลองวงจรเปิดปิดไม้กั้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำ 41

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทวิจารณ์และสรุป	42
5.1 สรุปผลการทดลอง	42
5.2 สรุปผลการดำเนินงาน	42
5.3 ปัญหาและอุปสรรคที่พบ	42
5.4 แนวทางแก้ไข	43
5.5 ข้อเสนอแนะและแนวทางในการพัฒนา	43
เอกสารอ้างอิง	44
ภาคผนวก	45
ภาคผนวก ก โปรแกรมการประมวลผลภาพและโปรแกรมควบคุม	46
ก.1 โปรแกรมวิซวลเบสิกที่ใช้ในส่วนของหน้าจอแสดงผลสำหรับการจอง	46
ก.2 โปรแกรมวิซวลเบสิกที่ใช้ในส่วนของหน้าจอแสดงผลสำหรับกรอกรหัสเปิด ไม้กัน	59
ก.3 โปรแกรมควบคุมเซนเซอร์ และแอลซีดี	61
ก.4 โปรแกรมควบคุมเซอร์โว	69
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	71
ข.1 เอกสารคู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	71
ข.2 เอกสารคู่มือการใช้งานอัลตราโซนิกเซนเซอร์ HC-SR04	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 โครงสร้างของไมโครคอนโทรลเลอร์	4
2.2 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์	5
2.3 ไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์ เบอร์ ATmega328 ขนาด 28 ขา	7
2.4 วงจรของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	7
2.5 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	8
2.6 ส่วนประกอบต่าง ๆ ของสภาพแวดล้อมของโปรแกรมอาดูโน่	9
2.7 สภาพแวดล้อมของโปรแกรมอาดูโน่	10
2.8 การกำหนดบอร์ดที่ใช้ในการทดลอง	11
2.9 การกำหนดพอร์ตการใช้งาน	11
2.10 ตัวอย่างชุดคำสั่งของโปรแกรม	12
2.11 แสดงผลการรันโปรแกรมที่คอมไพล์ผ่าน	12
2.12 โครงสร้างภายในตัวอัลตราโซนิกทรานสดิวเซอร์แบบเปียโซอิเล็กทริกที่ใช้สารเซรามิก	14
2.13 แสดงตัวอย่างการเขียนสัญลักษณ์ของอัลตราโซนิกทรานสดิวเซอร์	15
2.14 ชนิดของโปรแกรมประยุกต์	18
2.15 สภาพแวดล้อมของโปรแกรมมิชวลเบสิก	18
2.16 ส่วนประกอบต่าง ๆ ของสภาพแวดล้อมของโปรแกรมมิชวลเบสิก	19
2.17 หลักการทำงานของเซอร์โวมอเตอร์	20
3.1 การทำงานของระบบทั้งหมด	23
3.2 การทำงานของส่วนประมวลของเซนเซอร์	24
3.3 การทำงานของส่วนประมวลของเซอร์โวมอเตอร์	25
3.4 การทำงานของหน้าจอแสดงผล	26
3.5 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3	27
3.6 อัลตราโซนิกเซนเซอร์ HC-SR04	28
3.7 การเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์กับอัลตราโซนิกเซนเซอร์	28
3.8 หน้าจอแสดงผลโดยใช้โปรแกรมมิชวลเบสิก	29
3.9 หน้าจอแสดงรหัสเปิดปิดไม้กั้น	29
3.10 หน้าจอสำหรับใส่รหัสเปิดปิดไม้กั้น	30
3.11 เซอร์โวมอเตอร์ SG90 รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่น ๆ	30
3.12 การเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์กับเซอร์โวมอเตอร์	31

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.13 มุมมองแบบจำลองลานจอดรถแบบไอโซเมตริก (Isometric)	31
3.14 มุมมองแบบจำลองลานจอดรถด้านบน (Top view)	32
3.15 มุมมองแบบจำลองลานจอดรถด้านหน้า (Front View)	32
3.16 มุมมองแบบจำลองลานจอดรถด้านข้าง (Side view)	32
4.1 มุมมองโครงสร้างแบบจำลองลานจอดรถจากด้านหน้า	33
4.2 มุมมองโครงสร้างแบบจำลองลานจอดรถจากด้านบน	33
4.3 มุมมองโครงสร้างแบบจำลองลานจอดรถจากด้านซ้ายและด้านขวา	34
4.4 มุมมองโครงสร้างแบบจำลองลานจอดรถแบบไอโซเมตริก	34
4.5 ตำแหน่งของช่องจอดรถที่ไม่มีรถเข้าจอด	35
4.6 ตำแหน่งของช่องจอดรถที่มีการจอง	35
4.7 หน้าจอแสดงผลสำหรับกรอกหมายเลขทะเบียน	36
4.8 ตำแหน่งของช่องจอดรถที่มีการจองพร้อมกับหมายเลขทะเบียน	36
4.9 รหัสใช้ในการเปิด - ปิด ที่กั้นช่อง	37
4.10 หน้าจอแสดงผลสำหรับกรอกรหัสที่ใช้เปิดปิดที่กั้นช่อง	37
4.11 กรอกรหัสที่ได้รับมาจากการจอง	38
4.12 รหัสที่กรอกถูกต้อง	38
4.13 รหัสที่กรอกไม่ถูกต้อง	39
4.14 ตำแหน่งของช่องจอดรถที่มีรถเข้าจอด	39
4.15 แบบจำลองลานจอดรถขณะมีรถเข้าจอด	40
4.16 ตำแหน่งของช่องจอดรถที่มีรถจอดตามที่เซนเซอร์วัดได้	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ในยุคปัจจุบัน ยานพาหนะในการเดินทางเป็นสิ่งจำเป็น และเริ่มเข้ามามีบทบาทในชีวิตประจำวัน โดยเฉพาะในเมือง หรือย่านธุรกิจ ผู้คนเริ่มมีรถส่วนตัวใช้กันมากขึ้น ทำให้เกิดปัญหาของผู้ที่ไร้รถตามมา ไม่ว่าจะเป็น ที่จอดรถไม่ว่าง ที่จอดรถมีรถจอดจนเต็มแล้ว ซึ่งเป็นเหตุหนึ่งที่ทำให้เสียเวลาในการขับรถหาที่จอดรถ จากเหตุดังกล่าวจึงเป็นที่มาในการจัดทำโครงการนี้ขึ้นมา เพื่ออำนวยความสะดวกของผู้ใช้รถเวลาหาที่จอดรถ โดยการออกแบบลานจอดรถที่มีระบบการจองที่จอดรถก่อนที่จะขับเข้าไปในลานจอดรถ ซึ่งลานจอดรถนี้จะแสดงตำแหน่งต่าง ๆ ของที่จอดรถในลาน โดยใช้เซนเซอร์ และคอนโทรลเลอร์ มาแสดงผลผ่านทางหน้าจอคอมพิวเตอร์ เพื่อให้ผู้ใช้งานเห็นภาพลานจอดรถก่อนที่จะนำรถเข้าไปที่จอด และสามารถจองที่จอดรถในตำแหน่งที่ว่างอยู่ และต้องการจะจอดได้เลยจากหน้าจอแสดงผล ซึ่งเป็นการช่วยให้ผู้ใช้รถประหยัดเวลาในการหาที่จอดรถได้

1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบที่จอดรถให้มีความทันสมัย และอำนวยความสะดวกให้กับผู้ใช้บริการที่มาจอดรถ
2. เพื่อช่วยลดระยะเวลาในการวนหาที่จอดรถ และปัญหาในการหาที่จอดรถไม่ได้
3. เพื่อศึกษาการเขียน เพื่อประยุกต์ใช้โปรแกรมวิซวลเบสิก 6.0 (Visual Basic 6.0) ในงานควบคุมระบบทางวิศวกรรม

1.3 ขอบเขตของโครงการ

1. ตัวคอนโทรลเลอร์ที่ใช้ในการควบคุมเซนเซอร์ เซอร์โวมอเตอร์ และโปรแกรมวิซวลเบสิก
2. สามารถแสดงภาพลานจอดรถบนหน้าจอแสดงผล พร้อมทั้งสามารถจองตำแหน่งที่ต้องการจอดรถได้
3. แสดงผลเมื่อมีรถเข้ามาจอด
4. สามารถใช้โปรแกรมวิซวลเบสิก ในการสร้างหน้าจอแสดงผลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินการ

1. ศึกษา และวางแผนเกี่ยวกับรายละเอียดต่าง ๆ ของชิ้นงาน โดยมีอาจารย์คอยให้คำแนะนำ และเป็นพี่ปรึกษาโครงการ
2. ศึกษาคุณสมบัติ และการใช้งานของเซนเซอร์ เซอร์โวมอเตอร์ โปรแกรมวิซวลเบสิก และโปรแกรมควบคุมตัวคอนโทรลเลอร์
3. พิจารณาเซนเซอร์ เซอร์โวมอเตอร์ และคอนโทรลเลอร์ให้เหมาะสมกับโครงการนี้
4. ออกแบบลานจอดรถจำลองในคอมพิวเตอร์โดยใช้โปรแกรม AutoCAD และออกแบบวงจรอิเล็กทรอนิกส์ต่าง ๆ ให้สอดคล้องกับลานจอดรถ
5. เขียนโปรแกรมควบคุมตัวคอนโทรลเลอร์ และออกแบบหน้าจอแสดงผลโดยใช้โปรแกรมวิซวลเบสิก
6. สร้างลานจอดรถจำลองโดยให้มีขนาด และตำแหน่งต่าง ๆ ตามที่ออกแบบไว้
7. ติดตั้งเซนเซอร์ เซอร์โวมอเตอร์ ไมโครคอนโทรลเลอร์ และวงจรอิเล็กทรอนิกส์ต่าง ๆ บนลานจอดรถจำลอง
8. ทำการทดสอบระบบของลานจอดรถจำลองให้เป็นไปตามที่กำหนด และแก้ไขในส่วนที่ยังไม่สมบูรณ์
9. สรุปผลการทดลอง และจัดทำรายงาน

1.5 ผลที่คาดว่าจะได้รับ

1. สามารถนำความรู้ที่ได้จากการศึกษามาออกแบบ และประยุกต์ใช้ให้เกิดประโยชน์
2. สามารถเขียนโปรแกรมควบคุมตัวคอนโทรลเลอร์ และโปรแกรมวิซวลเบสิกได้
3. สามารถนำไปพัฒนา และประยุกต์ใช้ในลานจอดรถจริงได้ในอนาคต
4. ได้เรียนรู้การทำงานของเซนเซอร์ เซอร์โวมอเตอร์ และตัวคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีหรือหลักการ

2.1 ไมโครคอนโทรลเลอร์

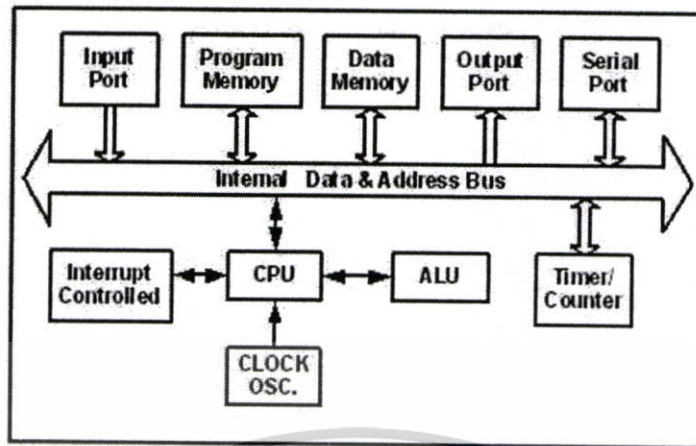
ไมโครคอนโทรลเลอร์ คืออุปกรณ์ควบคุมขนาดเล็ก เป็นอุปกรณ์อิเล็กทรอนิกส์ที่สามารถทำงานได้ตามเงื่อนไขต่าง ๆ ตามที่ได้เขียนโปรแกรมไว้ สามารถนำไปประยุกต์ใช้ในงานควบคุมต่าง ๆ ได้มากมาย เช่น การควบคุมเซนเซอร์ การควบคุมมอเตอร์ การควบคุมการทำงานของหุ่นยนต์ เป็นต้น ปัจจุบันไมโครคอนโทรลเลอร์มีหลายค่ายหลายเบอร์ให้เลือกใช้งาน และมีราคาถูกลง สามารถเขียนโปรแกรมได้หลายภาษา เช่น ภาษาซี เป็นต้น

2.1.1 ส่วนประกอบทั่วไปของไมโครคอนโทรลเลอร์

ส่วนประกอบโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งเป็น 5 ส่วนใหญ่ ๆ ดังนี้

1. ส่วนประมวลผลกลาง หรือซีพียู (CPU: Central Processing Unit) ทำหน้าที่คำนวณทางคณิตศาสตร์ และการตัดสินใจแบบมีเงื่อนไข (Logic)
2. ส่วนเก็บข้อมูล (Memory) ทำหน้าที่เก็บข้อมูลต่าง ๆ แบ่งออกเป็น
 - หน่วยความจำแรม (RAM: Random Access Memory) จะเก็บข้อมูลได้ ก็ต่อเมื่อมีไฟเลี้ยงอยู่ และเมื่อไม่มีไฟเลี้ยง ข้อมูลก็จะสูญหายไป
 - หน่วยความจำอีพรอม (EPROM: Erasable Programmable Read Only Memory) จะใช้ในการเก็บโค้ดเป็นส่วนใหญ่ สามารถเก็บข้อมูลได้ แม้วามีไฟเลี้ยงหรือไม่ก็ตาม
3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต สามารถแบ่งได้เป็นพอร์ตอินพุต และพอร์ตเอาต์พุต
4. ช่องทางเดินของสัญญาณ หรือบัส (Bus) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำ และพอร์ต โดยแบ่งเป็น บัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)
5. ส่วนกำเนิดสัญญาณนาฬิกา ตัวไมโครคอนโทรลเลอร์นั้นจะสามารถทำงานได้เมื่อมีสัญญาณนาฬิกา โดยจะขึ้นอยู่กับข้อกำหนดจังหวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 โครงสร้างของไมโครคอนโทรลเลอร์

2.2 ไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์

ไมโครคอนโทรลเลอร์เอวีอาร์ เป็นไอซีไมโครคอนโทรลเลอร์ของบริษัท Atmel มีสถาปัตยกรรมภายในเป็นแบบอาร์ไอเอสซี (RISC: Reduced Instruction Set Computer) โดยใช้สัญญาณนาฬิกาเพียง 1 ลูก ในการปฏิบัติงานใน 1 คำสั่ง โดยจะประกอบด้วยหน่วยความจำโปรแกรมภายในที่เป็นแบบแฟลช โปรแกรมข้อมูลได้แบบ In-System Programmable และในบางเบอร์ยังสามารถมีการกำหนดตำแหน่งของหน่วยความจำที่สร้างเป็นบูตโหลดเดอร์ กล่าวคือสามารถเขียนโปรแกรมเพื่อติดต่อกับ PC หรือไอซีตัวอื่น ๆ และยังสามารถโปรแกรมให้กับตัวเองได้

2.2.1 ประเภทของไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์

ไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์สามารถแบ่งออกได้เป็น 4 กลุ่ม ใหญ่ ๆ ดังนี้

2.2.1.1 TinyAVR - ATtiny Series

เช่นเบอร์ Tiny13, Tiny2313

- มีหน่วยความจำโปรแกรมขนาด 1 – 8 กิโลไบต์ (KB)
- มีจำนวนขาใช้งาน 6 – 32 ขา
- มีส่วนของอุปกรณ์เสริมที่ค่อนข้างจำกัด

2.2.1.2 MegaAVR - ATmega Series

เช่นเบอร์ ATmega8, ATmega16, ATmega32, ATmega64

- มีหน่วยความจำโปรแกรมขนาด 4 – 256 KB
- มีจำนวนขาใช้งาน 28 – 100 ขา
- มีชุดคำสั่งที่สามารถจัดการกับหน่วยความจำที่มีขนาดใหญ่มากขึ้น
- มีส่วนของอุปกรณ์เสริมมากในตัวไอซี

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุผลเบี่ยงเบนนอกจากนี้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3 XMEGA - ATxmega Series

เช่นเบอร์ ATxmega64A1, ATxmega128A1

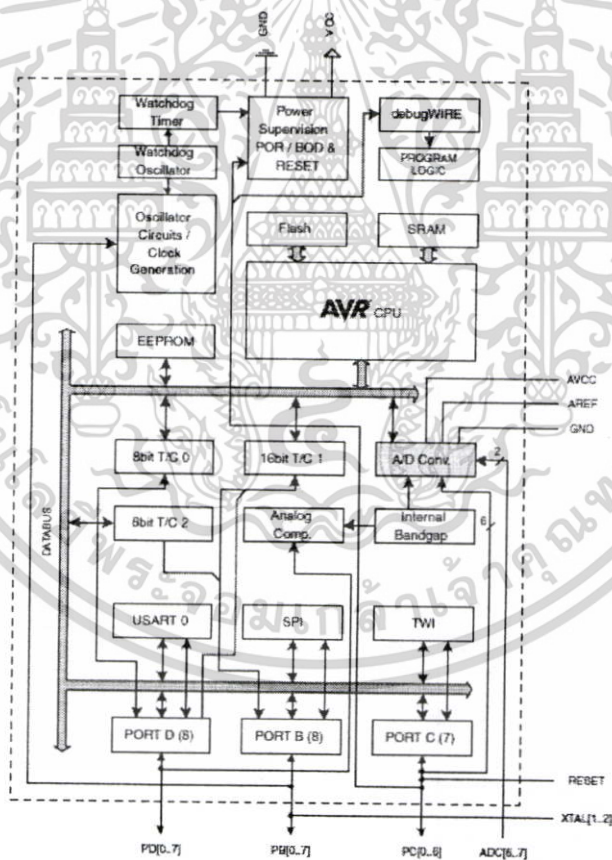
- มีหน่วยความจำโปรแกรมขนาด 16 – 384 KB
- มีจำนวนขาใช้งาน 44, 64 และ 100 ขา
- มีชุดคำสั่งที่สามารถจัดการกับระบบ DMA และการเข้าถึงเหตุการณ์ได้เร็วมากขึ้น โดย

ใช้การสื่อสารในแบบต่าง ๆ ได้หลายรูปแบบ

- มีส่วนของอุปกรณ์เสริมมากในตัวไอซีโดยการใช้งานกับ Digital-to-Analog Converter (DACs) และยังสามารถเขียนรหัสเฉพาะ โดยเข้ากันได้กับไฟล์แบบ AES และ DES

2.2.1.4 Application Specific AVR

เป็นไอซีที่สร้างเพื่อใช้งานเฉพาะ เช่น CAN AVR (ATmega64C1), LCD AVR (ATmega3290P/V), USB AVR (AT90USB1287) เป็นต้น



รูปที่ 2.2 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 อาดูโน

อาดูโนเป็นบอร์ดไมโครคอนโทรลเลอร์โดยใช้เอวีอาร์ ขนาดเล็กเป็นตัวประมวลผล และสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์อินพุต และอุปกรณ์เอาต์พุตต่าง ๆ ได้มากมาย ทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่น ๆ เช่น คอมพิวเตอร์พีซี ทั้งนี้ก็เนื่องมาจากอาดูโนสนับสนุนการเชื่อมต่อกับอุปกรณ์อินพุต และอุปกรณ์เอาต์พุต ทั้งแบบดิจิทัล และแอนะล็อก เช่น การรับค่าจากสวิตช์ หรืออุปกรณ์เซนเซอร์แบบต่าง ๆ รวมไปถึง การควบคุมอุปกรณ์เอาต์พุตต่าง ๆ ตั้งแต่ LED หลอดไฟ มอเตอร์ รีเลย์ ฯลฯ อาดูโนเป็นโครงการไมโครคอนโทรลเลอร์ระบบ โดยจะมีชุดพัฒนาของอาดูโนโดยเฉพาะ ภายในชุดพัฒนาของอาดูโนนี้ จะมีไลบรารีฟังก์ชันภาษาซีสำหรับติดต่อกับฮาร์ดแวร์ไว้ให้เป็นจำนวนมาก ซึ่งทำให้สามารถเขียนโปรแกรมสั่งงานอุปกรณ์ต่าง ๆ ได้ง่าย และสามารถเพิ่มไลบรารีได้ เพื่อให้สะดวกต่อการใช้งานมากขึ้น การทำงานของอาดูโนสามารถทำงานได้ทั้งบนวินโดวส์ (Windows) ลินุกซ์ (Linux) และแมคอินทอช (Macintosh OSX)

ส่วนภาษาในการเขียนโปรแกรมลงบนอาดูโนนั้น จะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) อื่น ๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรม และให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่าย และสะดวกมากขึ้น

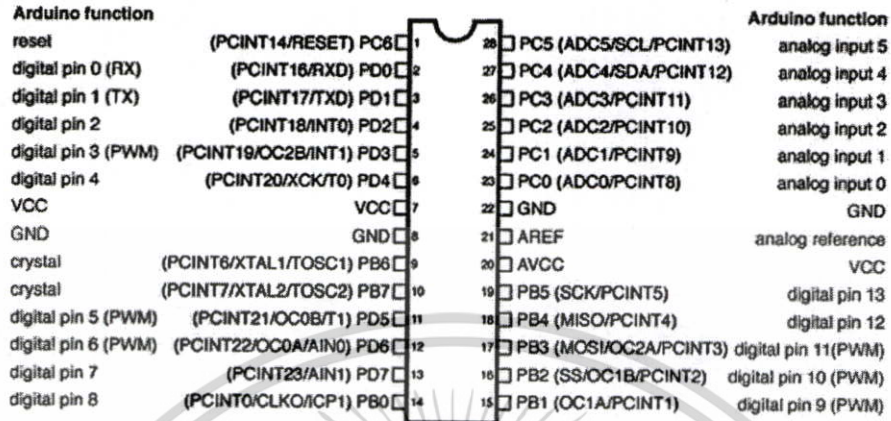
2.3.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล AVR ของบริษัท Atmel ซึ่งบอร์ดนี้เลือกใช้ CPU เบอร์ ATmega328 ขนาด 28 ขา มีความถี่ในการประมวลผล 16 เมกะเฮิร์ตซ์ (MHz) มี 14 ขาดิจิทัล ที่สามารถใช้เป็นอินพุตหรือเอาต์พุต ซึ่ง 6 ขาจาก 14 ขาดิจิทัล สามารถใช้เป็นสัญญาณ PWM (Pulse width modulation) เอาต์พุตได้ และมี 6 ขาเป็นแอนะล็อก หน่วยความจำถาวร 32 กิโลบิต (Kb) ใช้ USB ในการเชื่อมต่อกับคอมพิวเตอร์ มีปุ่ม Reset และสามารถใช้ไฟเลี้ยงผ่านการเชื่อมต่อ USB หรือต่อจากแหล่งจ่ายไฟภายนอกได้ โดยบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มีคุณสมบัติดังนี้

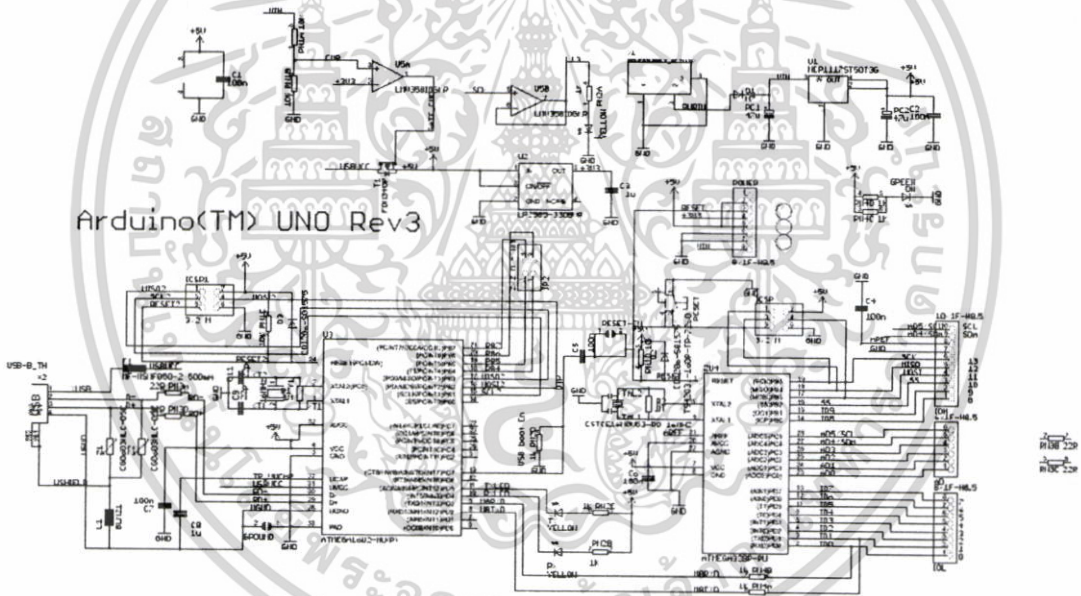
- หน่วยความจำโปรแกรมแบบ FLASH ขนาด 32 KB
- หน่วยความจำข้อมูลแบบ SRAM ขนาด 2 KB
- หน่วยความจำข้อมูลแบบ EEPROM ขนาด 1 KB
- สนับสนุนการเชื่อมต่อแบบ I²C bus
- วงจรสื่อสารอนุกรม

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากท่านได้ตั้งย่านแรงดัน 1.8-5.5 โวลต์ (V) อย่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความถี่ใช้งานสูงสุด 20 MHz

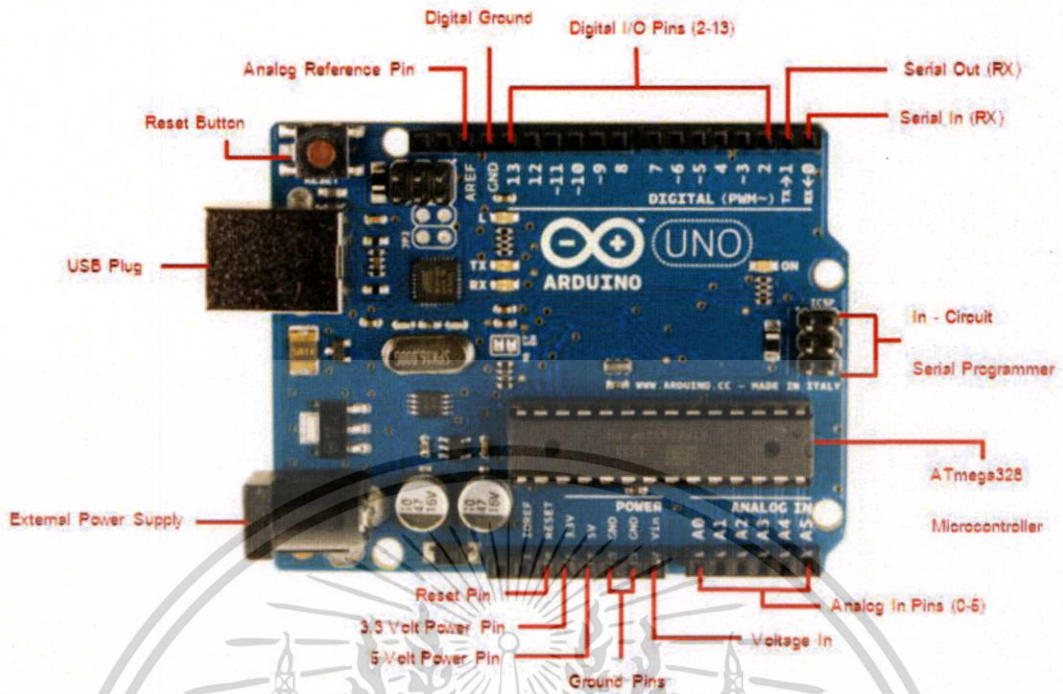


รูปที่ 2.3 ไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์ เบอร์ ATmega328 ขนาด 28 ขา



รูปที่ 2.4 วงจรของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

2.3.1.1 จุดต่อการใช้งานของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

จุดต่อการใช้งานดังรูปที่ 2.5 ประกอบด้วย

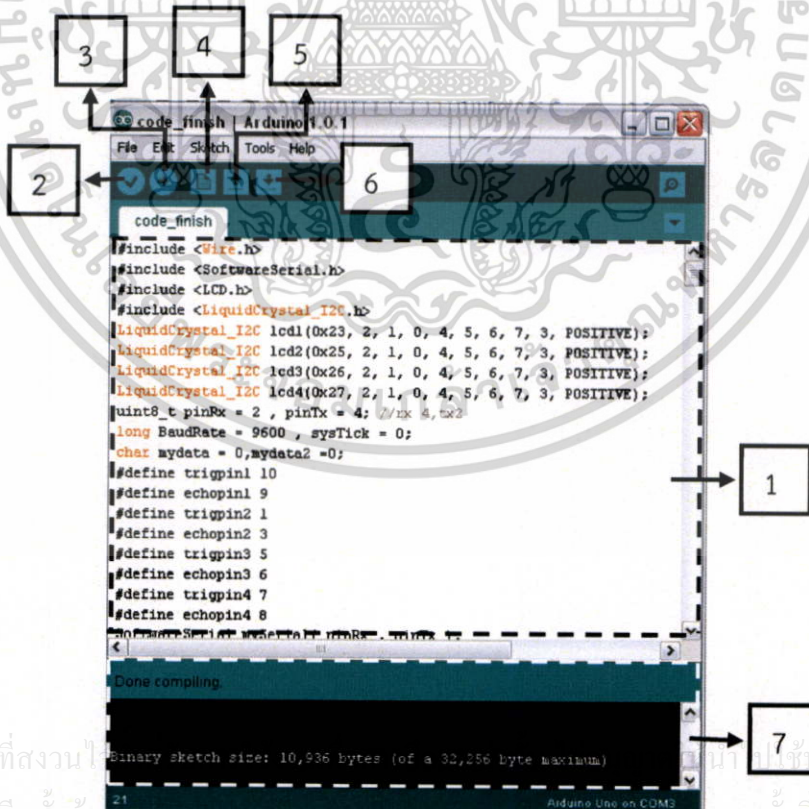
- Vin คือแรงดันไฟฟ้าที่จ่ายเข้าไปยังบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เมื่อใช้แหล่งพลังงานจากภายนอก
- GND คือจุดเชื่อมต่อกราวด์
- 5V คือแรงดันไฟฟ้าเอาต์พุตขนาด 5V
- 3.3V คือแรงดันไฟฟ้า 3.3V กระแสไฟฟ้าสูงสุด 50 มิลลิแอมป์ (mA)
- IOREF คือพอร์ตที่จะอ้างอิงแรงดันไฟฟ้ากับการทำงานของไมโครคอนโทรลเลอร์ว่าจะทำงานที่แรงดันไฟฟ้าขนาด 5V หรือ 3.3V
 - มี 14 ขาดิจิตัล คือหมายเลข 0 ถึง 13 จะสามารถใช้เป็นขาอินพุตหรือเอาต์พุตได้ซึ่งมี 6 ขาที่สามารถใช้เป็นสัญญาณ PWM เอาต์พุตได้ ประกอบด้วยหมายเลข 3, 5, 6, 9, 10 และ 11 โดยใช้ฟังก์ชัน pinMode(), digitalWrite() และ digitalRead() โดยแต่ละขาจะทำงานที่แรงดันไฟฟ้า 5V และแต่ละขาจะรับหรือให้กระแสไฟฟ้าสูงสุด 40 mA
 - มี 6 ขาแอนะล็อกอินพุต คือหมายเลข A0 ถึง A5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โปรแกรมอาduino

โปรแกรมอาduino เป็นซอฟต์แวร์ที่พัฒนาขึ้นมาเพื่อการเขียนโปรแกรมลงบนบอร์ดอาduino ซึ่งจะสามารถดาวน์โหลดได้ที่ <http://arduino.cc/en/Main/Software> โดยจะสามารถรองรับได้ทั้งระบบปฏิบัติการวินโดวส์ (Windows) ลินุกซ์ (Linux) และแมคอินทอช (Macintosh OSX) ซึ่งตัวภาษาของอาduino เป็นการนำเอาโอเพนซอร์สโปรเจกต์ชื่อ wiring มาพัฒนาต่อ ภายในชุดพัฒนาของอาduino จะมีไลบรารีฟังก์ชันภาษาซีสำหรับติดต่อกับฮาร์ดแวร์จำนวนมากไว้ให้ ทำให้สามารถเขียนโปรแกรมสั่งงานอุปกรณ์ต่าง ๆ ได้ง่าย โดยไม่จำเป็นต้องศึกษาลงไปในรายละเอียดมากนัก โดยตัวโปรแกรมมีส่วนประกอบดังรูปที่ 2.6 ดังนี้

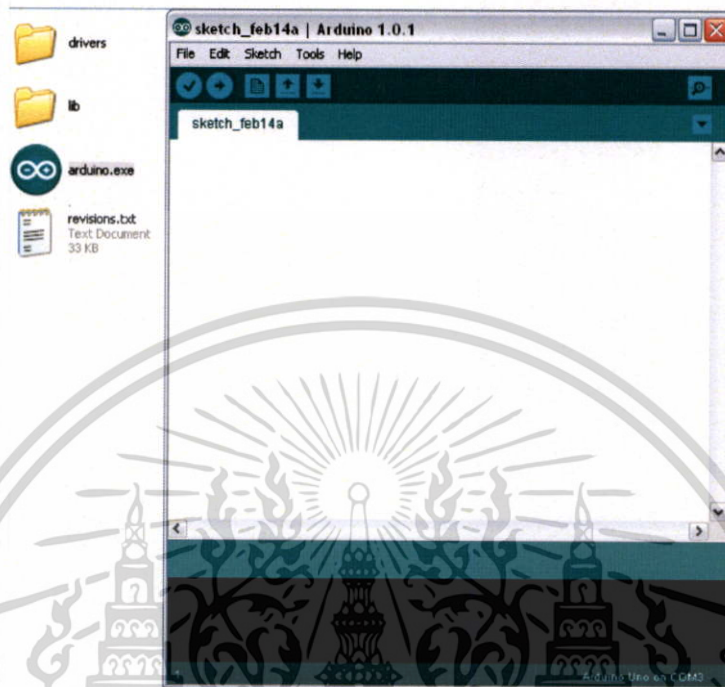
- หมายเลข 1 : ชุดคำสั่งของโปรแกรม ใช้สำหรับการเขียนโค้ดต่าง ๆ ลงโปรแกรม
- หมายเลข 2 : Verify ใช้สำหรับตรวจสอบความถูกต้องของชุดคำสั่งของโปรแกรม
- หมายเลข 3 : Upload ใช้สำหรับลงชุดคำสั่งในบอร์ดการทดลอง
- หมายเลข 4 : New ใช้สำหรับสร้างไฟล์งานใหม่
- หมายเลข 5 : Open ใช้สำหรับเปิดไฟล์งานที่มีอยู่แล้ว
- หมายเลข 6 : Save ใช้สำหรับบันทึกไฟล์งาน
- หมายเลข 7 : ส่วนแสดงผลการทำงาน



รูปที่ 2.6 ส่วนประกอบต่าง ๆ ของสภาพแวดล้อมของโปรแกรมอาduino

2.4.1 การทดสอบโปรแกรมบนอาดูโน่

1. เรียกโปรแกรมขึ้นมา



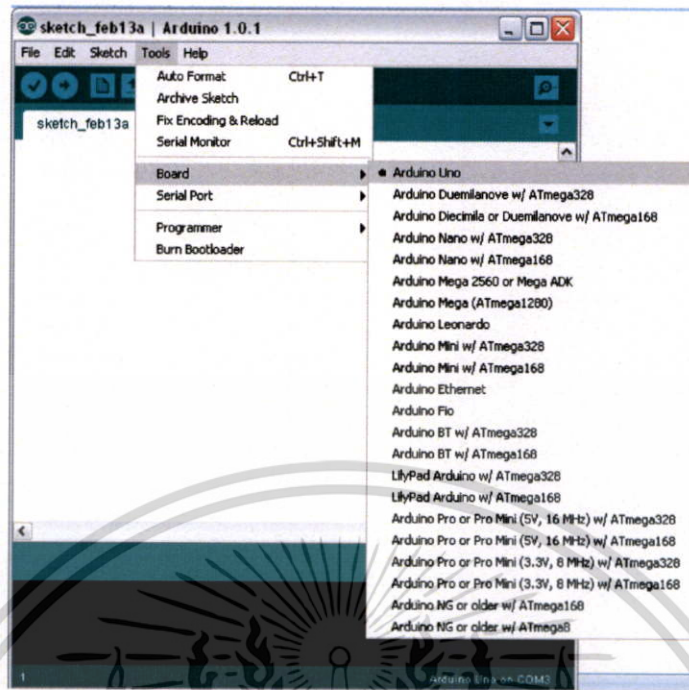
รูปที่ 2.7 สภาพแวดล้อมของโปรแกรมอาดูโน่

2. กำหนดค่าทางฮาร์ดแวร์ที่ใช้

- เลือกเบอร์ของไมโครคอนโทรลเลอร์ โดยเลือกเมนู Tools --> Board -->

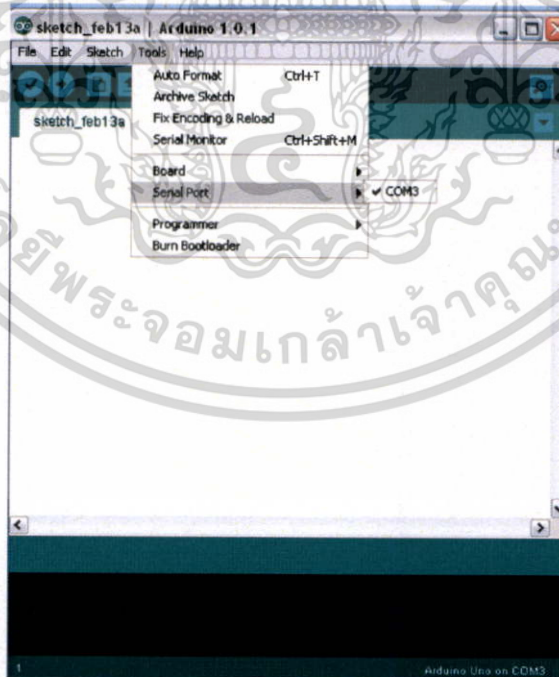
Arduino Uno

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การกำหนดบอร์ดที่ใช้ในการทดลอง

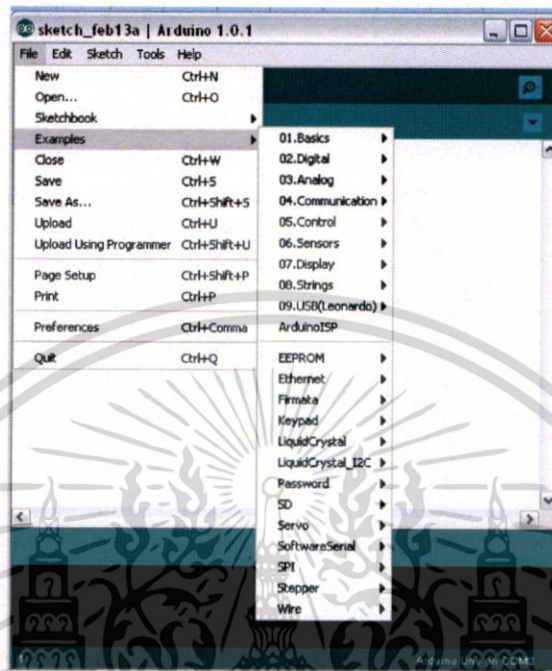
- กำหนดพอร์ตอนุกรมที่ใช้ในการติดต่อ โดยเลือกเมนู Tools --> Serial Port
--> COMxxx ในที่นี้คือ COM3



รูปที่ 2.9 การกำหนดพอร์ตการใช้งาน

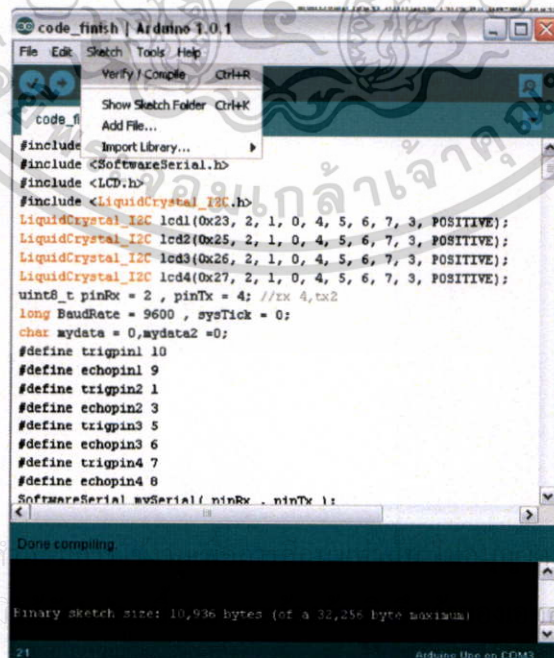
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ...
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปิดโปรแกรมการทำงาน โดยในตัวโปรแกรม จะมีตัวอย่างโปรแกรมให้เลือกทดลอง อยู่มากมาย โดยเลือกไปที่เมนู File --> Examples แล้วเลือกตัวอย่างที่ต้องการจะทดสอบ



รูปที่ 2.10 ตัวอย่างชุดคำสั่งของโปรแกรม

4. คอมไพล์โปรแกรม เมื่อเปิดไฟล์โปรแกรมแล้ว จากนั้นทำการคอมไพล์โปรแกรม โดยเลือกเมนู Sketch --> Verify/Compile ส่วนแสดงผลการทำงานจะขึ้นข้อความว่า Done compiling



รูปที่ 2.11 ผลการรันโปรแกรมที่คอมไพล์ผ่าน

2.5 คลื่นอัลตราโซนิก

คลื่นอัลตราโซนิก (Ultrasonic) หมายถึง คลื่นเสียงที่มีความถี่สูงกว่า 20 กิโลเฮิร์ตซ์ (kHz) ขึ้นไป เป็นความถี่ที่สูงเกินกว่าที่ประสาทหูมนุษย์จะได้ยิน ซึ่งโดยทั่วไปแล้วหูของมนุษย์จะได้ยินเสียงสูงสุดเพียงแค่ประมาณ 15 kHz เพราะว่าหูชั้นกลางของคนเราซึ่งประกอบด้วยกระดูกค้อน กระดูกทั่ง และกระดูกโกลน จะไม่มีการสั่นสะเทือนเมื่อมีความถี่สูงเกิน 20 kHz เปรียบเสมือนกรองความถี่ต่ำผ่าน (Low-pass filter)

คุณสมบัติเด่นของคลื่นย่านอัลตราโซนิกคือ เป็นคลื่นที่มีทิศทางที่สามารถเล็งคลื่นเสียงไปยังเป้าหมายที่เราต้องการได้โดยเจาะจง ยิ่งถ้าคลื่นมีความถี่สูงขึ้นความยาวคลื่นก็จะมีขนาดสั้นลงดังสมการที่ 2.1 จะเห็นได้ว่าความยาวคลื่นมีค่าแปรผกผันกับความถี่ของคลื่น

$$\lambda = \frac{c}{f}$$

(2.1)

โดยที่ λ คือความยาวคลื่นมีหน่วยเป็นเมตร (m)

c คือความเร็วในการเดินทางของคลื่นมีหน่วยเป็นเมตรต่อวินาที (m/s)

f คือความถี่ของคลื่นมีหน่วยเป็นเฮิร์ตซ์ (Hz)

ตัวอย่างเช่น ความถี่ 40 kHz จะทำให้มีความยาวคลื่นในอากาศ 8 มิลลิเมตร (mm) ซึ่งมีขนาดเล็กกว่ารูเปิดของตัวที่ให้กำเนิดเสียงความถี่นั้นมาก คลื่นเสียงจะไม่มีการเลี้ยวเบนที่ขอบจึงพุ่งออกมาเป็นลำแคบ ๆ หรือที่เราเรียกว่า มีทิศทาง แต่ถ้าคลื่นมีความถี่ต่ำทำให้ความยาวคลื่นยาวกว่าช่องที่ให้คลื่นเสียงออกมาจากตัวกำเนิดเสียง โดยทั่วไปคลื่นจะหักเบนที่ขอบด้านนอกของตัวกำเนิดเสียงทำให้เกิดการกระจายทิศทางของคลื่นการนำไปใช้งานของคลื่นเสียงย่านอัลตราโซนิกโดยใช้คุณสมบัติการมีทิศทางได้หลากหลาย เช่น นำไปใช้ในเครื่องวัดความหนาของวัตถุโดยส่งเกตระยะเวลาที่คลื่นสะท้อนกลับมา เครื่องมือวัดความลึก และทำแผนที่ใต้ท้องทะเล เครื่องหาตำแหน่งอวัยวะบางส่วนในร่างกาย ใช้ทดสอบการรั่วไหลของท่อ เครื่องควบคุมระยะไกล (Ultrasonic remote control) เครื่องล้างอุปกรณ์ (Ultrasonic cleaner) โดยการส่งคลื่นความถี่สูงลงไปของเหลว เป็นผลทำให้มีฟองอากาศเล็ก ๆ จำนวนมากผุดขึ้น ฟองอากาศเล็ก ๆ นี้จะมีพลังงานซึ่งสามารถชะล้างคราบหรือสิ่งฝังที่อยู่บนผิวอุปกรณ์โดยไม่ทำลายชิ้นงาน เป็นต้น โดยความถี่ที่ใช้ขึ้นอยู่กับการใช้งาน โดยถ้าคลื่นเสียงที่ต้องเดินทางผ่านอากาศ ความถี่ที่ใช้มักจะจำกัดไม่เกิน 50 kHz เพราะถ้าความถี่สูงเกินกว่านี้อากาศจะดูดกลืนคลื่นเสียงเพิ่มมาก ทำให้ระดับความแรงของคลื่นที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานด้านการแพทย์ซึ่งต้องการรัศมีทำการสั้น ๆ อาจใช้ความถี่ในช่วง 1 ถึง 10 MHz ขณะที่ความถี่เป็น จิกะเฮิร์ตซ์ (GHz) ก็มีใช้กันในหลาย ๆ การใช้งานที่ตัวกลางที่คลื่นเสียงเดินทางผ่านไม่ใช่

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การคัดลอกหรือการนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตจากสถาบันฯ จะถือว่าผิดกฎหมายและไม่มีการรับประกันใดๆ

2.5.1 อัลตราโซนิกทรานสดิวเซอร์

อัลตราโซนิกทรานสดิวเซอร์ (Ultrasonic Transducer) คืออุปกรณ์ที่สามารถเปลี่ยนพลังงานในรูปอื่นให้มาเป็นพลังงานทางกลโดยการสั่นไปมา ที่ทำให้เกิดคลื่นเสียงย่านอัลตราโซนิกกระจายไปในอากาศหรือแปลงพลังงานทางกลเป็นพลังงานในรูปอื่น คลื่นอัลตราโซนิกสามารถสร้างได้โดยตัวทรานสดิวเซอร์ ซึ่งแปลงพลังงานไฟฟ้า และพลังงานกล โดยหลักการสร้างจะมีหลายวิธีดังนี้

1. แบบแมกนีโตสตริกทีฟ (Magnetostrictive Transducer) ซึ่งเปลี่ยนแปลงระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้น

2. แบบอิเล็กโตรสตริกทีฟ (Electrostrictive Transducer) ซึ่งเปลี่ยนแปลงระหว่างพลังงานไฟฟ้ากับพลังงานทางกล

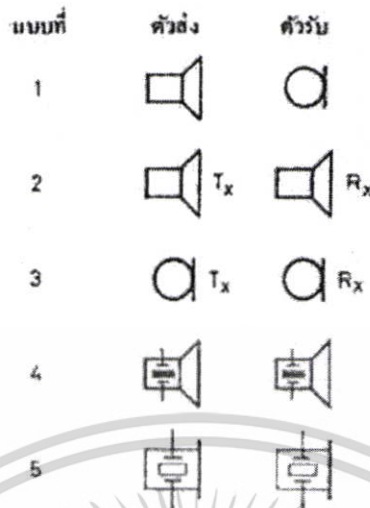
3. แบบเพียโซอิเล็กทริก (Piezo-electric Transducer) ซึ่งเปลี่ยนแปลงระหว่างพลังงานไฟฟ้า และพลังงานทางกล โดยมีความถี่เรโซแนนซ์คงที่อยู่ค่าหนึ่ง

ทรานสดิวเซอร์แบบเพียโซอิเล็กทริก ภายในตัวนี้ที่ใช้กันในปัจจุบันประกอบด้วยชั้นสารเซรามิกสี่เหลี่ยมซึ่งมีผิวโลหะเงินฉาบอยู่ทั้ง 2 หน้าเพื่อให้ต่อสายไฟออกมาเป็นขา 2 ขา ชั้นสารเซรามิกนี้ประกอบขึ้นจากสารเซรามิก 2 ชั้น ประกอบกันอยู่โดยวางให้ขั้วไดโพลทางไฟฟ้าภายในอะตอมของมันมีทิศทางตรงข้ามกันดังรูปที่ 2.12 โดยคุณสมบัติโดยทั่วไปของอัลตราโซนิกทรานสดิวเซอร์แบบเพียโซอิเล็กทริก คือ มีความต้านทานแรงดันไฟตรงสูงมาก โดยมีค่าประมาณ 100 เมกะโอห์ม (MΩ) แต่เมื่อในขณะที่มันทำงานความต้านทานไฟสลบจะมีค่าลดลง



รูปที่ 2.12 โครงสร้างภายในตัวอัลตราโซนิกทรานสดิวเซอร์แบบเพียโซอิเล็กทริกที่ใช้สารเซรามิก

เมื่อมีการป้อนแรงดันให้แก่ตัวเพียโซอิเล็กทริกจะทำให้ชั้นสารเซรามิกมีการโก่งงอไปมาดังรูปที่ 2.12 ทำให้เกิดคลื่นเสียงอัลตราโซนิกกระจายไปในอากาศ ด้านหน้าทำเป็นช่องเปิดมีตะแกรงติดอยู่เพื่อให้คลื่นอัลตราโซนิกออกมาเกิดการกดอัดอากาศโดยรอบเกิดเป็นคลื่นเสียงที่มีความถี่ขอสัญญานตรงกับความถี่เรโซแนนซ์ ซึ่งเป็นความถี่ทางกลตามธรรมชาติของชั้นสารเซรามิก โดยทำหน้าที่คล้ายลำโพงคือเป็นตัวส่ง (Transmitter) ของคลื่นเสียง ในทำนองกลับกันเมื่อมีคลื่นเสียงที่มีความถี่ตรงกับความถี่เรโซแนนซ์ของชั้นสารเซรามิกเข้ามา จะทำให้ชั้นสารโก่งงอไปมา และเกิดสัญญาณแรงดันซึ่งมีขนาดเล็กขึ้นมาคร่อมขั้วทั้งสองของตัวมันเองได้ โดยทำหน้าที่คล้ายไมโครโฟนคือเป็นตัวรับ (Receiver) ของคลื่นเสียง



รูปที่ 2.13 ตัวอย่างการเขียนสัญลักษณ์ของอัลตราโซนิกทรานสดิวเซอร์

2.5.2 ข้อควรรู้ในการใช้งานของอัลตราโซนิกทรานสดิวเซอร์

1. ไม่ควรให้ตัวทรานสดิวเซอร์ได้รับการกระทบหรือตกจากที่สูง เพื่อป้องกันโครงสร้างภายในมิให้เสียหาย

2. ทรานสดิวเซอร์ที่มีขายโดยทั่วไปจะทนแรงดันตกคร่อมตัวมันสูงสุดได้ไม่เกินกว่า 20 V_{rms} ดังนั้นขนาดของสัญญาณที่จะป้อนให้กับทรานสดิวเซอร์ก็ควรจะอยู่ในขีดจำกัดอันนี้

3. ความถี่เรโซแนนซ์ของทรานสดิวเซอร์ 40 kHz ที่มีขายกันโดยทั่วไปจะผิดพลาดไปไม่เกิน ± 1 kHz และมีแถบความถี่ (Bandwidth) ประมาณ 4.5 kHz สำหรับตัวส่ง และมีแถบความถี่ประมาณ 5.0 kHz สำหรับตัวรับ จะเห็นได้ว่าแถบความถี่ของตัวรับจะกว้างกว่าของตัวส่งอยู่เล็กน้อย เพื่อให้แน่ใจว่าตัวรับจะสามารถรับความถี่ทั้งหมดที่ออกมาจากตัวส่งได้

4. อุณหภูมิใช้งานของตัวทรานสดิวเซอร์ควรอยู่ในช่วง -20 ถึง $+60$ องศาเซลเซียส

5. ทั้งตัวส่ง และตัวรับจะมีทิศทางการคล้ายคลึงกันมากกล่าวคือ ที่ตำแหน่งเบนจากแนวแกนของตัวส่งไปประมาณ 30 องศา ความแรงของคลื่นเสียงที่ถูกส่งออกไปจะลดลงจากแนวแกนประมาณ 10 เดซิเบล (dB) ในทำนองเดียวกันถ้าคลื่นเสียงพุ่งเข้ามาในแนวที่เบี่ยงเบนไปจากแนวแกนของตัวรับไปประมาณ 30 องศา ความไวหรือขนาดแรงดันที่ออกมาก็ลดลงไปประมาณ 10 dB ด้วยเช่นกัน ดังนั้นในการใช้งานที่เป็นการควบคุมระยะไกลในที่โล่งแจ้งจึงควรพยายามให้ทั้งตัวรับ และตัวส่งอยู่ในแนวที่พุ่งตรงเข้าหากันให้มากที่สุด อย่างไรก็ตามในกรณีที่อยู่ในห้องอาจจะเบี่ยงเบนจากกันได้มาก เพราะคลื่นเสียงอัลตราโซนิกสามารถสะท้อนกับกำแพง พื้น และวัตถุที่อยู่ในห้อง ทำให้คลื่นเสียงเข้าไปหาตัวรับได้หลายทาง

6. ในกรณีที่ใช้งานตัวรับจะต้องมีตัวต้านทานต่อขนานกับตัวรับเพื่อทำหน้าที่เป็นโหลดตามปกติแล้วตัวต้านทานตัวนี้ควรมีค่าอยู่ในช่วงจาก 10 ถึง 100 กิโลวัตต์ (kW) จากการทดลองพบว่า ถ้าเปลี่ยนโหลดจาก 100 kW มาเป็น 10 kW ความไวจะลดลงประมาณ 10 ถึง 20 dB แต่แถบความถี่จะกว้างขึ้น ถ้าใช้ค่าความต้านทานต่ำลงไปอีกความถี่เรโซแนนซ์จะลดลงไปจากที่ระบุไว้ ถ้าการใช้งานมีสัญญาณรบกวนมากควรใช้โหลดที่มีความต้านทานสูง เพื่อให้ตัวส่งมีความไวสูง และมีแถบความถี่แคบ

7. ตามปกติแล้วเราสามารถนำเอาตัวส่ง และรับมาใช้แทนกันได้ในการใช้งานส่วนใหญ่ และตัวส่งหรือตัวรับของยี่ห้อใด รุ่นใด ก็สามารถที่จะนำมาใช้แทนกันได้ในงานส่วนใหญ่ของเพียงแต่ให้มีความถี่หรือแชนเนลเดียวกันเท่านั้น อย่างไรก็ตามในบางกรณีอาจต้องเปลี่ยนแปลงค่าความต้านทานสมมูลทางด้านไฟสลบเพื่อให้ลักษณะผลตอบสนองทางความถี่สอดคล้องกับของเดิม

2.6 วิชาลเบสิก

วิชาลเบสิก (Visual Basic) เป็นภาษาที่ใช้สำหรับสร้างหรือพัฒนาโปรแกรมประยุกต์ (Application) ที่ทำงานภายใต้ระบบปฏิบัติการวินโดวส์ (Windows) โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกว่า คอนโทรล (Control) ถูกนำมาใช้สำหรับสร้างหรือออกแบบส่วนต่อประสานกราฟิกกับผู้ใช้ (Graphical User Interface หรือ GUI) ได้ตามต้องการ ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย

2.6.1 คุณสมบัติของวิชาลเบสิก

1. วิชาลเบสิกเป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมที่ใช้งานได้ตั้งแต่ระดับต้น เพื่อใช้งานโปรแกรมง่าย ๆ บนระบบปฏิบัติการวินโดวส์ ตลอดจนโปรแกรมเมอร์ระดับมืออาชีพ ที่จะพัฒนาโปรแกรมในระดับสูง โดยการใช้เซตข้อมูลวัตถุ (Object Linking and Embedding หรือ OLE) และส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface หรือ API) ของระบบปฏิบัติการวินโดวส์มาประกอบการเขียนโปรแกรม

2. วิชาลเบสิกเป็นภาษาที่มีรูปแบบโครงสร้างคำสั่งเขียนง่าย อ่านเข้าใจได้ง่าย โดยการสร้างองค์ประกอบต่าง ๆ ของโปรแกรมประยุกต์ด้วยคอนโทรล โดยมีรูปแบบที่สื่อด้วยภาพบนหน้าจอเพื่อติดต่อกับผู้ใช้งาน (User Interface) ได้ตามต้องการ และเมื่อวาดหน้าจอเสร็จก็เขียนโปรแกรมเพื่อเชื่อมโยงความสัมพันธ์ขององค์ประกอบแต่ละตัวบนหน้าจอเข้าด้วยกัน ให้ทำงานสอดคล้องกันตามที่เรากำลังต้องการ

3. วิชาลเบสิกสามารถนำไปประยุกต์ใช้ในงานโปรแกรมฐานข้อมูลได้ เนื่องจากมีเครื่องมือต่าง ๆ เกี่ยวกับฐานข้อมูลอย่างครบถ้วน เช่นเครื่องมือในการติดต่อกับฐานข้อมูลทั้ง Microsoft Access หรือฐานข้อมูลบนระบบ Client Server เช่น Microsoft SQL Server โดยการติดต่อกับฐานข้อมูลนั้น เพียงแต่กำหนดตำแหน่งของฐานข้อมูลพร้อมกับข้อมูลที่จำเป็นในการติดต่อกับฐานข้อมูลเท่านั้น ก็สามารถติดต่อกับฐานข้อมูลได้ทันที

4. วิชาลเบสิกสามารถนำไปประยุกต์ใช้งานด้านกราฟิก, เกม, หรือโปรแกรมที่รันบนระบบเครือข่ายอินเทอร์เน็ต หรืออินเทอร์เน็ตผ่านเว็บเบราว์เซอร์ได้

5. วิชาลเบสิกสามารถนำไปประยุกต์ใช้งานในงานเขียนโปรแกรมควบคุมอัตโนมัติผ่าน

เอกสารนี้ทางฟอร์ตชนานี้, พอร์ตอนุกรม ตลอดจนพอร์ตยูเอสบีได้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. วิชาลเบสิกสามารถใช้งานกับระบบปฏิบัติการคอมพิวเตอร์รุ่นเก่าจนถึงปัจจุบันได้ เนื่องจากถูกพัฒนามาจากภาษาเบสิก (Basic) ซึ่งเป็นภาษาที่มีมาตั้งแต่เดิมในสมัยที่ยังใช้ดอส (Dos) ในการพัฒนาโปรแกรม ส่วนสำคัญที่ทำให้วิชาลเบสิกกลายเป็นโปรแกรมที่นิยมใช้กันมากในปัจจุบันเพราะภาษาเบสิกถูกนำมาใช้เขียนโปรแกรมกับวินโดวส์ ทำให้ผู้ใช้สามารถใช้งานโปรแกรมได้อย่างสะดวก และมีความหลากหลายมากขึ้น

2.6.2 ส่วนประกอบของวิชาลเบสิก

เมื่อเปิดใช้งานโปรแกรมวิชาลเบสิก จะขึ้นหน้าต่างดังรูปที่ 2.14 ซึ่งแสดงถึงชนิดของโปรแกรมประยุกต์ โดยเบื้องต้นจะพัฒนาโปรแกรมประยุกต์ชนิด Standard.EXE ก่อนเพราะเป็นโปรแกรมประยุกต์ที่ใช้งานทั่ว ๆ ไป เมื่อคอมไพล์โปรเจค จะได้โปรแกรมประยุกต์ที่มีนามสกุล *.exe เมื่อเลือกชนิดของโปรแกรมประยุกต์แล้วจะเข้าสู่สภาพแวดล้อมของโปรแกรมวิชาลเบสิก (Integrated Development Environment หรือ IDE) ดังรูปที่ 2.15 และส่วนประกอบต่าง ๆ ของสภาพแวดล้อมของโปรแกรมวิชาลเบสิกจะแสดงดังรูปที่ 2.16 โดยที่

หมายเลข 1 คือ แถบเมนู (Menu Bar) เป็นส่วนในการเก็บคำสั่งที่สามารถใช้งานได้ทั้งหมดในวิชาลเบสิก ประกอบไปด้วยเมนูทำงานกับ แฟ้ม (File), มุมมอง (View), หน้าต่าง (Windows), ฯลฯ

หมายเลข 2 คือ แถบเครื่องมือ (Toolbar) ประกอบด้วยปุ่มคำสั่งต่าง ๆ ที่ช่วยให้ใช้งานคำสั่งของวิชาลเบสิก ได้รวดเร็วยิ่งขึ้น

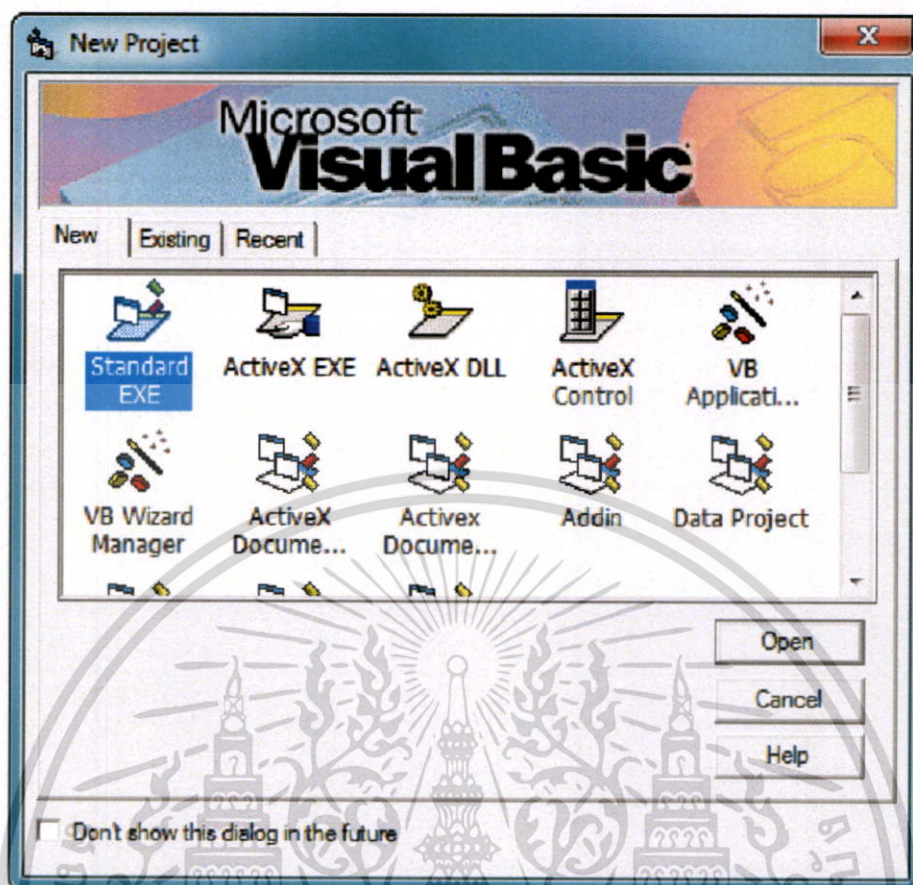
หมายเลข 3 คือ กล่องเครื่องมือ (Toolbox) เป็นที่แสดงเครื่องมือต่าง ๆ ที่เรียกว่า คอนโทรล ซึ่งเป็นเครื่องมือที่สามารถเลือกไปวางลงบนฟอร์มได้ เพื่อออกแบบหน้าจอของโปรแกรม โดยจะต้องเลือกคอนโทรลให้ตรงกับจุดประสงค์ในการใช้งาน

หมายเลข 4 คือ ฟอร์มว่างสำหรับออกแบบหน้าต่างของโปรแกรม (Form)

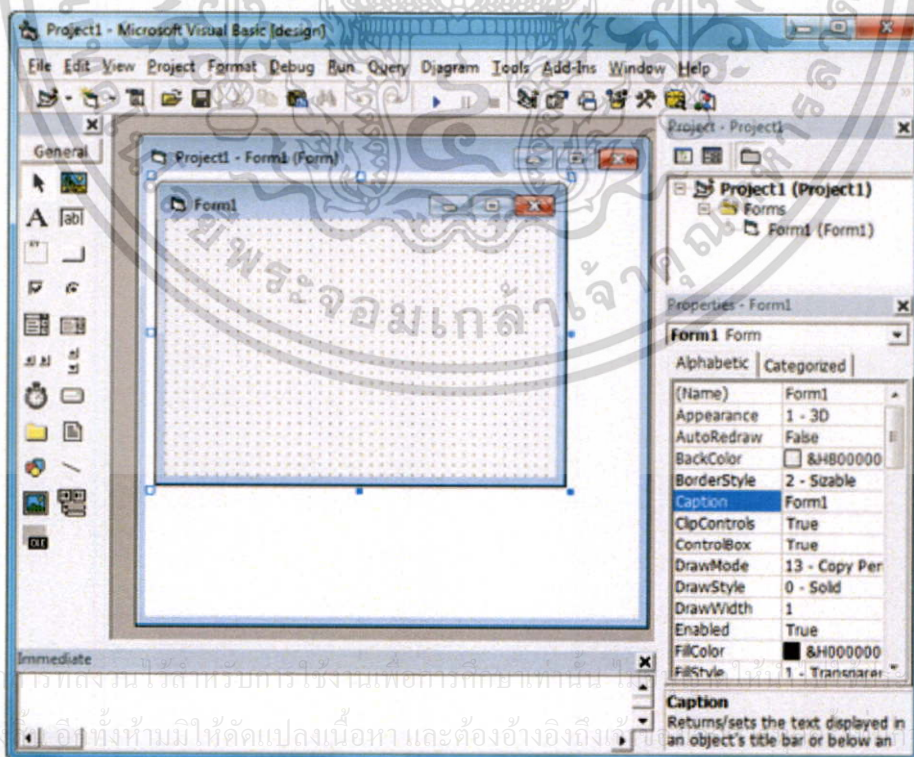
หมายเลข 5 คือ หน้าต่างโปรเจค (Project Explorer) เป็นหน้าต่างที่แสดงโมดูลต่าง ๆ ที่มีอยู่ในโปรเจคทั้งหมด

หมายเลข 6 คือ หน้าต่างคุณสมบัติ (Properties Window) เป็นหน้าต่างที่แสดงคุณสมบัติของคอนโทรลที่เลือกอยู่ขณะนั้น

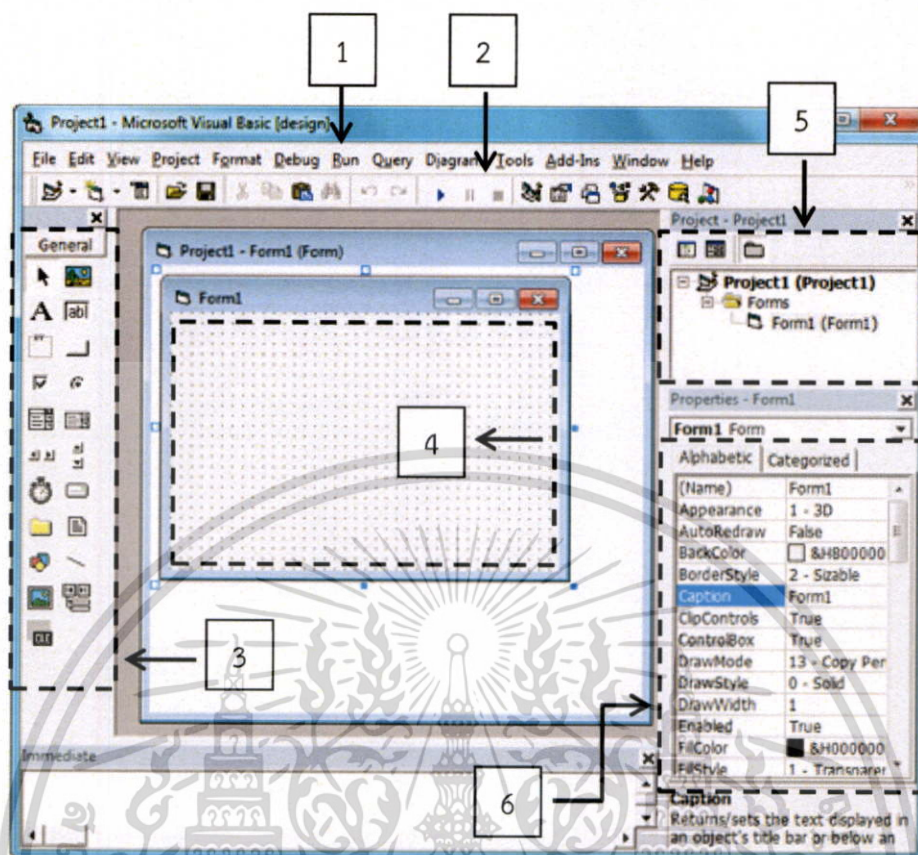
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ชนิดของโปรแกรมประยุกต์



รูปที่ 2.15 สภาพแวดล้อมของโปรแกรมวิซวลเบสิก



รูปที่ 2.16 ส่วนประกอบต่าง ๆ ของสภาพแวดล้อมของโปรแกรมวิซวลเบสิก

2.6.3 ขั้นตอนพื้นฐานในการเขียนโปรแกรมด้วยวิซวลเบสิก

ขั้นตอนพื้นฐานในการเขียนโปรแกรมด้วยวิซวลเบสิกนั้นมีอยู่ 5 ขั้นตอนหลัก ๆ ได้แก่

2.6.3.1 สร้าง, ย้าย, เปลี่ยนขนาด และลบคอนโทรลบนฟอร์ม

การสร้าง, ย้าย, เปลี่ยนขนาด และลบคอนโทรลบนฟอร์มสำหรับออกแบบหน้าต่างตาของโปรแกรมสามารถทำได้ดังนี้

1. การสร้างคอนโทรล

การสร้างคอนโทรลนั้น ให้คลิกเมาส์เลือกคอนโทรลที่ต้องการจากกล่องเครื่องมือนำมาวางบนฟอร์ม แล้วลากเมาส์เพื่อกำหนดขนาดของคอนโทรลตามที่ต้องการบนฟอร์ม

2. การย้ายคอนโทรล

เมื่อต้องการเปลี่ยนตำแหน่งของคอนโทรลที่วางไปแล้วบนฟอร์มไปยังตำแหน่งใหม่ ให้คลิกเมาส์เลือกคอนโทรลที่ต้องการย้าย และลากเมาส์ไปยังตำแหน่งใหม่ได้เลย

3. การเปลี่ยนขนาดของคอนโทรล

คลิกเมาส์เลือกคอนโทรลที่ต้องการเปลี่ยนขนาด เลื่อนเมาส์ไปยังปุ่มสี่เหลี่ยมรอบคอนโทรลที่เลือก จนตัวชี้เมาส์เป็นรูปลูกศร 2 ทาง จากนั้นให้คลิกแล้วลากเมาส์เพื่อเปลี่ยนขนาดคอนโทรล เมื่อได้ขนาดของคอนโทรลตามที่ต้องการแล้วให้ปล่อยเมาส์

4. การลบคอนโทรล

คลิกเมาส์เลือกคอนโทรลที่ต้องการลบออก แล้วกดปุ่ม Delete บนคีย์บอร์ด หรือคลิกเมาส์เลือกคำสั่งแก้ไข (Edit) แล้วเลือกคำสั่งลบ (Delete) จากแถบเมนูได้

2.6.3.2 กำหนดคุณสมบัติของคอนโทรล

การกำหนดคุณสมบัติของคอนโทรลสามารถทำได้ผ่านหน้าต่างคุณสมบัติ โดยคลิกเมาส์เลือกคอนโทรลบนหน้าจอที่ต้องการกำหนดคุณสมบัติ ให้สังเกตว่า หน้าต่างคุณสมบัติจะแสดงคุณสมบัติของคอนโทรลที่เลือก จากนั้นให้เลือกคุณสมบัติที่ต้องการเปลี่ยนในหน้าต่างคุณสมบัติ และป้อนค่าใหม่ทางด้านขวาของหน้าต่างคุณสมบัติ หากยังไม่มีหน้าต่างคุณสมบัติ ให้กดปุ่ม F4 บนคีย์บอร์ด หรือคลิกเมาส์เลือกคำสั่งมุมมอง แล้วเลือกคำสั่งหน้าต่างคุณสมบัติ จากแถบเมนูได้

2.6.3.3 เขียนโปรแกรมตอบสนองต่อเหตุการณ์

การเขียนโปรแกรมสำหรับวิซวลเบสิก เป็นการใช้นโยบายกำหนดให้คอนโทรลตอบสนองต่อเหตุการณ์บางอย่างที่เกิดขึ้น เมื่อโปรแกรมที่สร้างถูกนำมาใช้งาน เช่นถ้าต้องการให้โปรแกรมเกิดการตอบสนองเมื่อปุ่มคำสั่งถูกคลิก ต้องเขียนคำสั่งที่ตอบสนองต่อเหตุการณ์เหล่านั้นให้สอดคล้องกัน ทำได้โดยเลือกคำสั่งมุมมอง แล้วเลือกคำสั่งโค้ด (Code) จากแถบเมนู เพื่อเปิดหน้าต่างแก้ไขโค้ด เลือกชื่อคอนโทรลที่ต้องการให้ตอบสนองต่อเหตุการณ์ในช่องรายการด้านซ้าย แล้วเลือกเหตุการณ์ของคอนโทรลที่จะถูกตอบสนองในช่องรายการด้านขวา ใส่คำสั่งลงไประหว่างข้อความ Private Sub ชื่อคอนโทรล_ชื่อเหตุการณ์ กับ End Sub โดยคำว่า Private Sub และ End Sub เป็นตัวบ่งบอกว่าคำสั่งที่ตอบสนองเริ่มต้น และจบลงที่ใด

2.6.3.4 รันโปรแกรมประยุกต์ (Run application)

เมื่อสร้างโปรแกรมเสร็จเรียบร้อยแล้ว สามารถรันโปรแกรมได้โดยกดปุ่ม F5 บนคีย์บอร์ด หรือคลิกเมาส์เลือกคำสั่งรัน แล้วเลือกคำสั่งเริ่ม จากแถบเมนูได้

2.6.3.5 บันทึกโปรเจค

การบันทึกโปรเจคสามารถทำได้ 2 แบบคือ

1. บันทึกโปรเจคในชื่อเดิม

เลือกคำสั่งแฟ้ม แล้วเลือกคำสั่งบันทึกโปรเจค (Save Project) จากแถบเมนู หากมีไฟล์อื่น ๆ ในโปรเจคที่ยังไม่ได้บันทึก จะต้องตั้งชื่อไฟล์เพื่อบันทึกด้วย

2. บันทึกโปรเจคในชื่อใหม่

เลือกคำสั่งแฟ้ม แล้วเลือกคำสั่งบันทึกโปรเจคเป็น (Save Project As) จากแถบเมนู และต้องตั้งชื่อไฟล์เพื่อบันทึกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาค้นคว้าเท่านั้น มิอนุญาตให้นำไปเผยแพร่ ใช้งาน หรือดัดแปลงในทางอื่น ๆ ในโปรเจคที่ยังไม่ได้บันทึก จะต้องตั้งชื่อไฟล์เพื่อบันทึกด้วย

2.7 เซอร์โวมอเตอร์

2.7.1 ลักษณะของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ (Servo motor) คือ มอเตอร์ที่ทำงานโดยใช้สัญญาณพัลส์ โดยภายในเซอร์โวมอเตอร์จะประกอบไปด้วย มอเตอร์ไฟฟ้ากระแสตรง, ชุดเกียร์ และชุดควบคุม โดยจะประกอบกันไว้ภายในโมดูลเดียวกันหรือภายในกล่องพลาสติกเดียวกัน โครงสร้างภายในของเซอร์โวมอเตอร์จะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (Variable Resistor) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์ และตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ซึ่งทำให้สามารถควบคุมให้มอเตอร์หมุนไปตำแหน่งองศาที่ต้องการได้ แต่เซอร์โวมอเตอร์นี้จะหมุนได้แค่ประมาณ 180 องศาหรือครึ่งรอบเท่านั้น อาจมีบางรุ่นที่สามารถหมุนได้ถึง 210 องศาแต่จะไม่สามารถหมุนเป็นวงรอบได้ เพื่อป้องกันความเสียหายที่จะเกิดกับตัวต้านทานปรับค่าได้ โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานทั้งหมดเพียง 3 เส้นเท่านั้น คือ เส้นสีแดง จะเป็นสายไฟเลี้ยงโดยสามารถต่อแรงดันอยู่ระหว่าง 4V ถึง 6V ซึ่งขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว, เส้นสีน้ำตาล จะเป็นสายกราวด์ และเส้นสีส้ม จะเป็นสายสัญญาณที่ใช้ควบคุมตำแหน่งแกนของเซอร์โวมอเตอร์ โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณแบบ พัลส์วidthมอดูเลชัน (Pulse Width Modulation) ข้อดีของมอเตอร์ชนิดนี้คือจะมีขนาดเล็ก น้ำหนักเบา, ให้แรงบิดสูง, กินพลังงานน้อย และสามารถควบคุมด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่น ๆ เพราะมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว

2.7.2 หลักการทำงานของเซอร์โวมอเตอร์

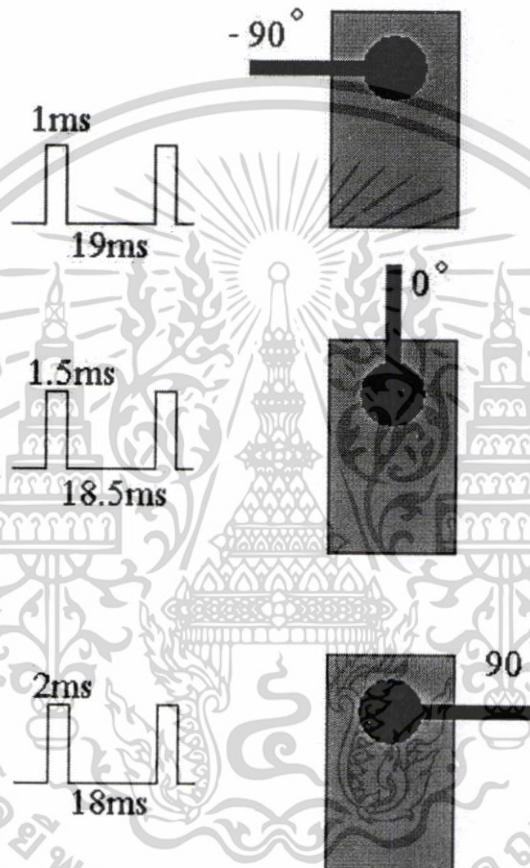
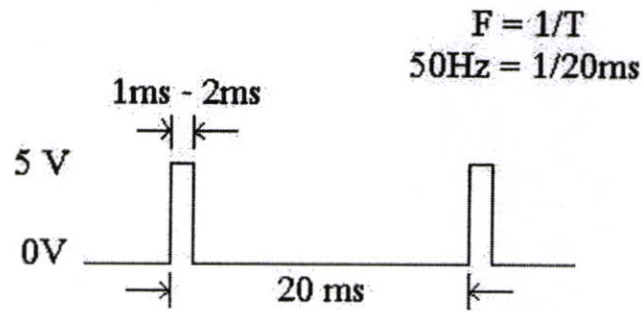
สัญญาณพัลส์ที่ใช้ควบคุมตำแหน่งของเซอร์โวมอเตอร์จะเป็นสัญญาณที่มีความถี่ 50 Hz มีคาบเวลาคือ 20 มิลลิวินาที (ms) โดยสัญญาณช่วงบวกจะมีค่าอยู่ระหว่าง 1 ถึง 2 ms และสัญญาณช่วงลบจะมีค่าอยู่ประมาณ 18 ms ดังรูปที่ 2.17 ซึ่งบ่งบอกถึงลักษณะของสัญญาณพัลส์ที่จ่ายไปยังขาสัญญาณของเซอร์โวมอเตอร์ โดยความกว้างของสัญญาณพัลส์ช่วงบวกจะมีผลต่อตำแหน่งของมอเตอร์ดังนี้

ความกว้างของสัญญาณพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม -90 องศา หรือทิศทางทวนเข็มนาฬิกา

ความกว้างของสัญญาณพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือจุดกึ่งกลางของมอเตอร์

ความกว้างของสัญญาณพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม +90 องศา หรือทิศทางตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารทสงวน ไว้สำหรับกร ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



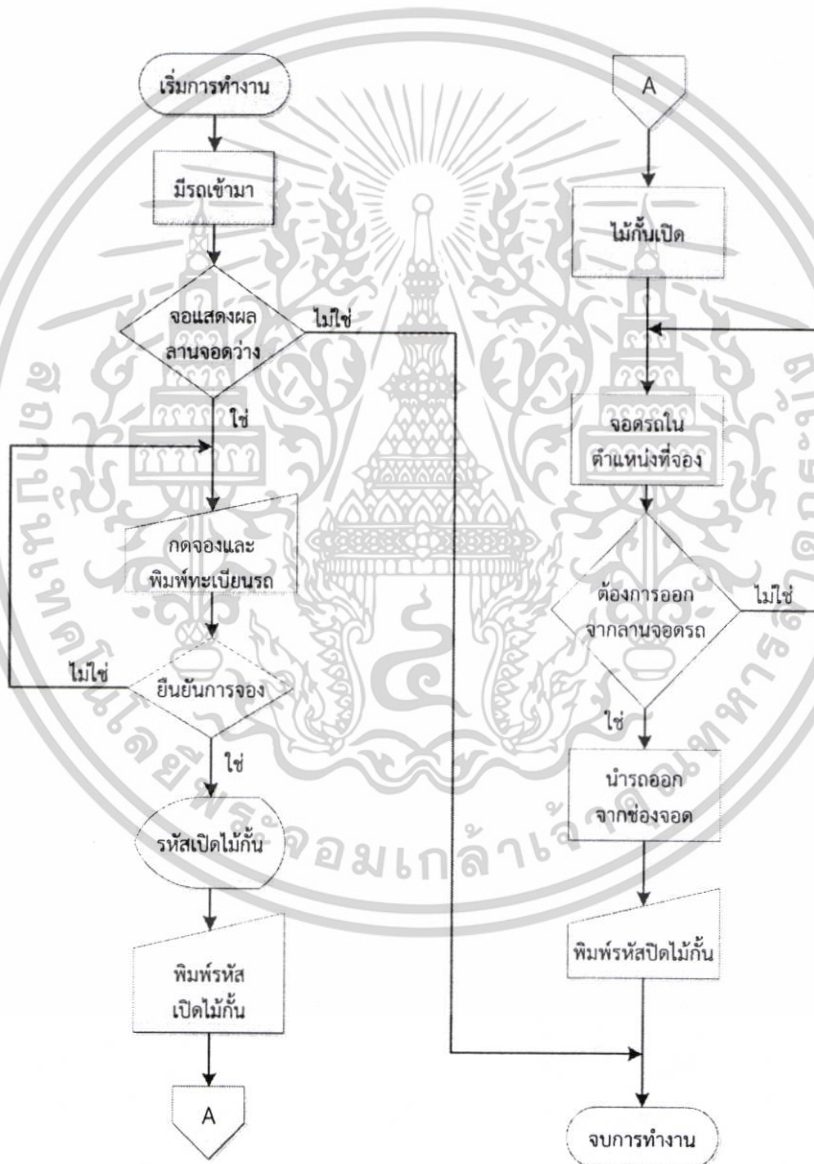
รูปที่ 2.17 หลักการทำงานของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

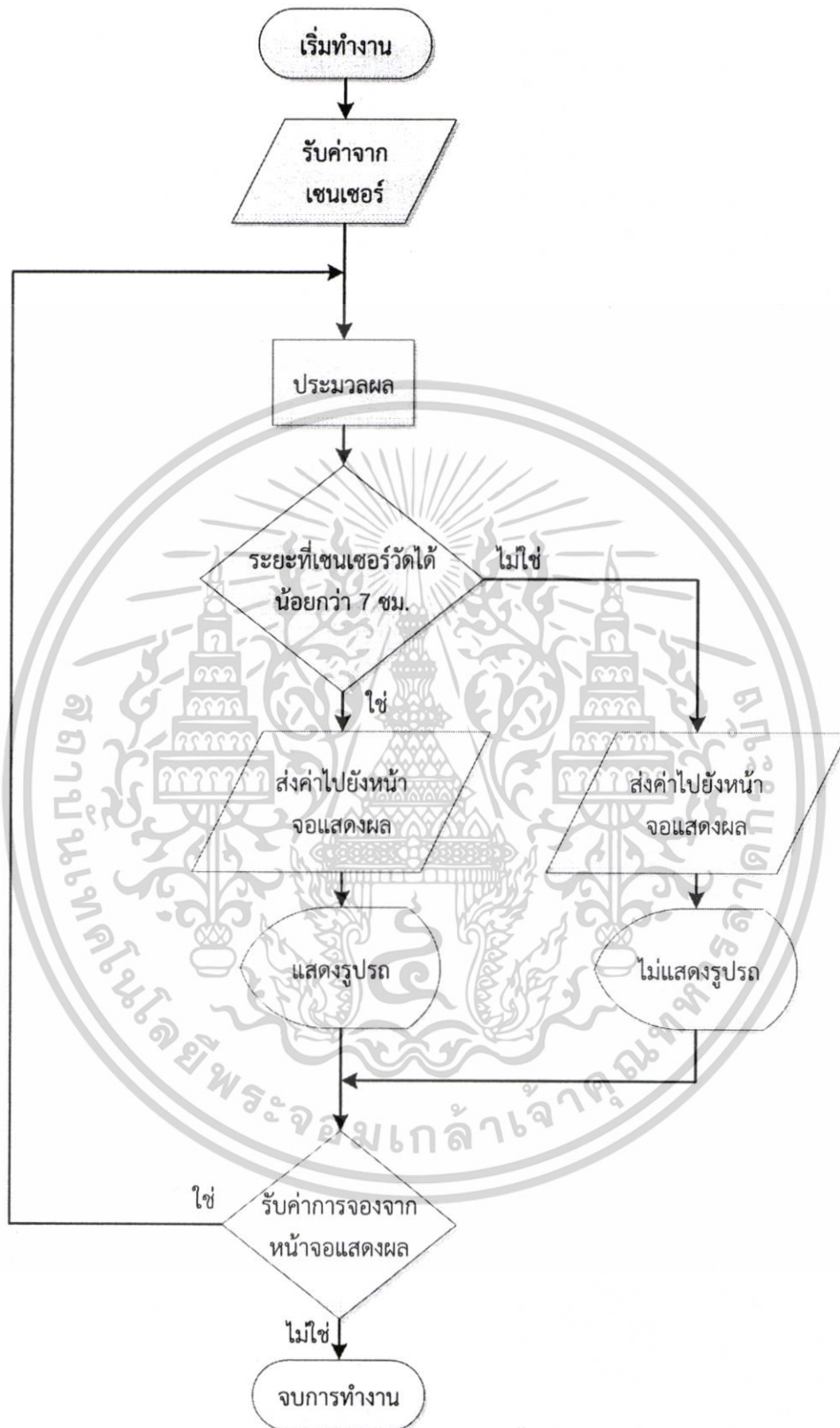
บทที่ 3

การคำนวณและการสร้าง

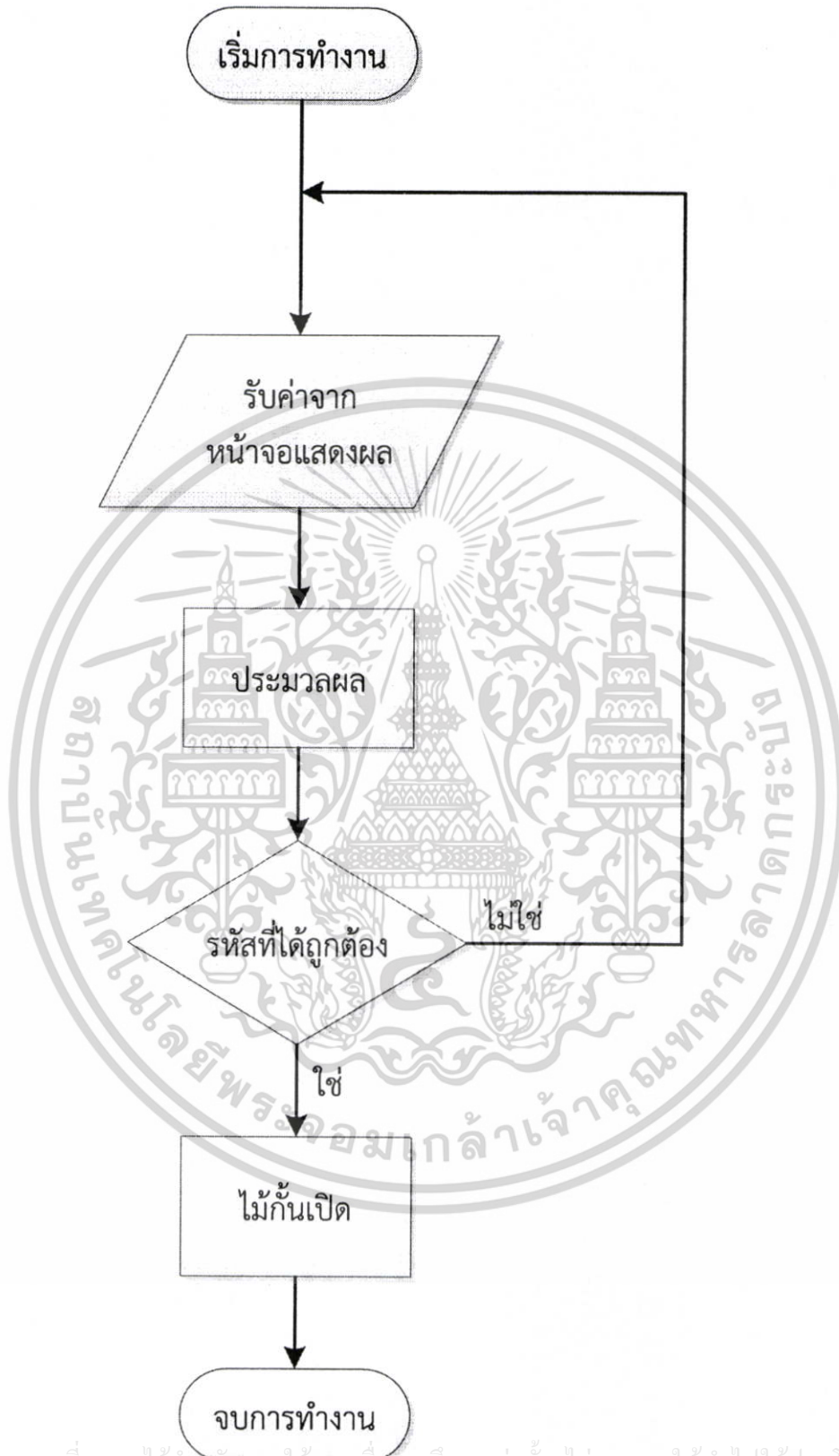
ขั้นตอนของการออกแบบ และโครงสร้างของระบบในโครงการนี้ สามารถเขียนเป็นแผนภาพการทำงานของระบบทั้งหมด การทำงานของส่วนประมวลผลของเซนเซอร์ การทำงานของส่วนประมวลผลของเซอร์โวมอเตอร์ และการทำงานของหน้าจอแสดงผล ได้ดังรูปที่ 3.1 ถึงรูปที่ 3.4 ตามลำดับ



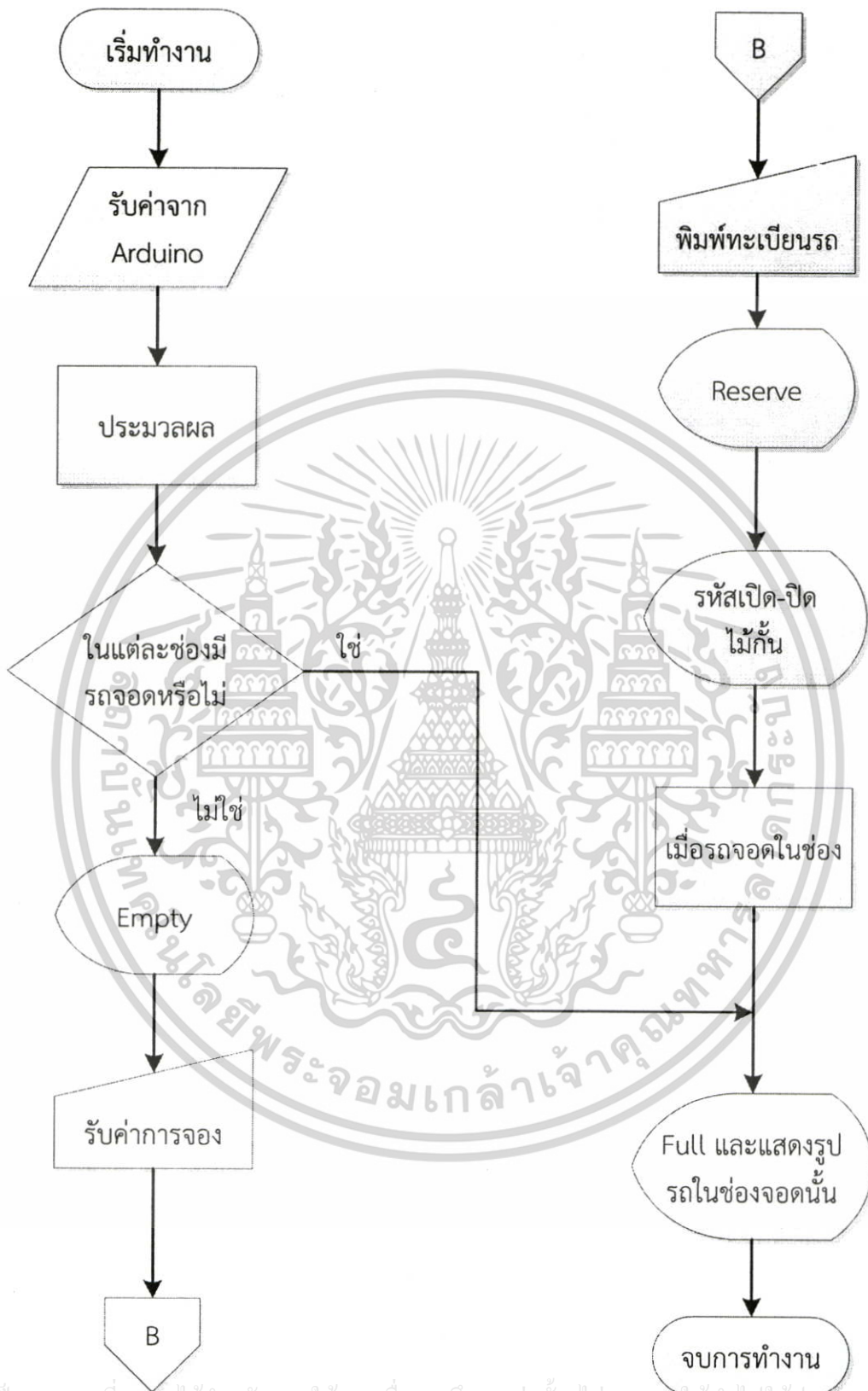
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **รูปที่ 3.1** การทำงานของระบบทั้งหมด กรุณาอย่าให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.2 การทำงานของส่วนประมวลผลของเซนเซอร์
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.3 การทำงานของส่วนประมวลของเซิร์ฟเวอร์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ทุกฉบับลงมือทำและต้องให้ของเดิมทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.4 การทำงานของหน้าจอแสดงผล
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

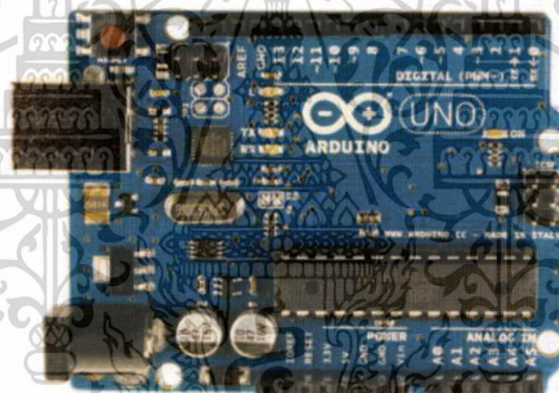
3.1 โครงสร้างของระบบในส่วนต่าง ๆ ของโครงการ

โครงสร้างของระบบในส่วนต่าง ๆ ของโครงการแบ่งออกเป็น 4 ส่วนย่อย ๆ คือ

1. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
2. อัลตราโซนิกเซนเซอร์ HC-SR04
3. หน้าจอแสดงผลที่ออกแบบโดยวิซวลเบสิก
4. เซอร์โวมอเตอร์ SG90

3.1.1 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

ระบบนี้ใช้การควบคุมการตรวจจับ และแสดงผล รวมทั้งควบคุมเซอร์โวมอเตอร์ ซึ่งจำเป็นต้องใช้ไมโครคอนโทรลเลอร์เพื่อที่จะทำให้วงจรทำงานได้อย่างสมบูรณ์ในโครงการนี้เลือกใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ดังรูปที่ 3.5 จำนวน 2 บอร์ดโดยบอร์ดที่ 1 จะใช้ในส่วนของการตรวจจับรถ และบอร์ดที่ 2 จะใช้ในส่วนของเซอร์โวมอเตอร์สำหรับการเปิดปิดไม้กั้น โดยเขียนโปรแกรมลงไปในไมโครคอนโทรลเลอร์เพื่อใช้ควบคุมส่วนต่าง ๆ ของระบบ

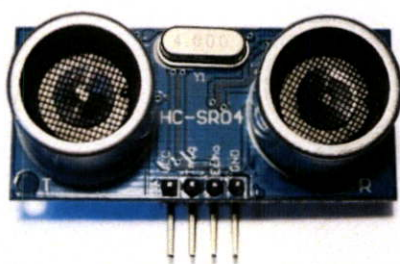


รูปที่ 3.5 บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

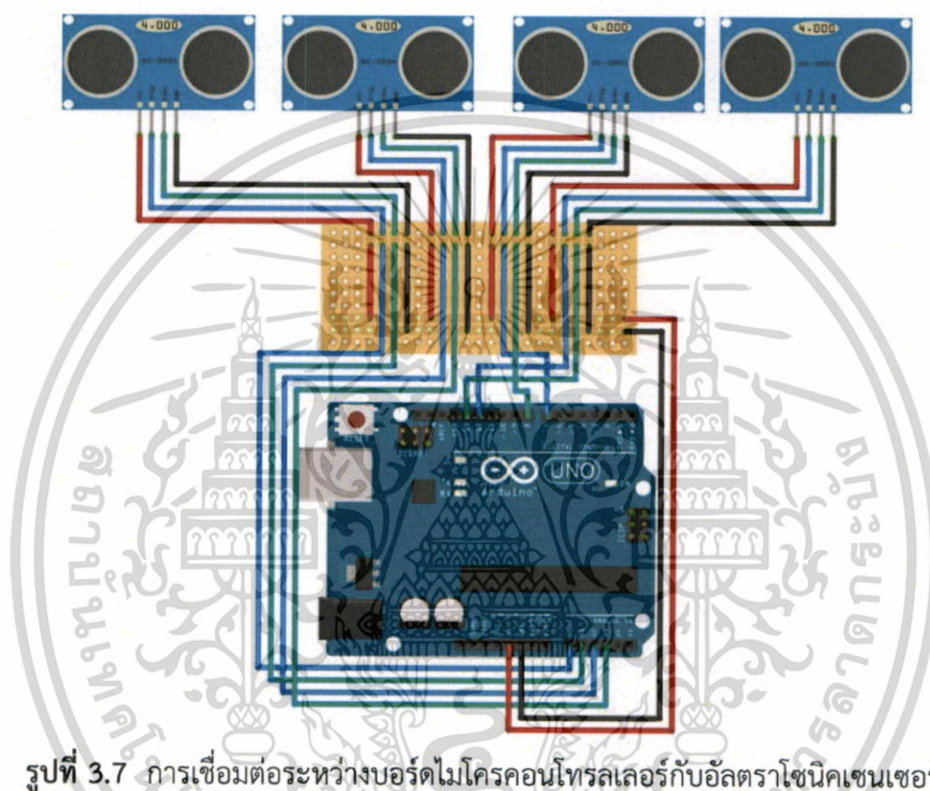
3.1.2 อัลตราโซนิกเซนเซอร์ HC-SR04

การตรวจจับบริเวณช่องจอดรถจะต้องใช้เซนเซอร์วัดในที่นี้เลือกใช้อัลตราโซนิกเซนเซอร์ HC-SR04 ดังรูปที่ 3.6 เนื่องจากระยะทางที่สามารถวัดได้ของเซนเซอร์ตัวนี้อยู่ที่ 2 ถึง 400 เซนติเมตร ซึ่งนอกจากจะเหมาะสมกับโครงสร้างลานจอดรถที่ได้ทำการออกแบบแล้ว ยังสามารถนำไปใช้ในลานจอดรถจริงได้อีกด้วย ถ้ายังไม่มีรถยนต์มาจอดเซนเซอร์จะวัดระยะได้ 7 เซนติเมตร แต่ถ้ามีรถมาจอดในช่องจอดรถยนต์ระยะที่เซนเซอร์วัดได้จะมีค่าน้อยกว่า 7 เซนติเมตร ซึ่งเซนเซอร์จะส่งค่าไปยังไมโครคอนโทรลเลอร์เพื่อทำการประมวลผลโดยจะทำการเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์กับอัล

ตราโซนิกเซนเซอร์ ดังรูปที่ 3.7 การใช้งานเพื่อการศึกษานี้ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 อัลตราโซนิกเซนเซอร์ HC-SR04



รูปที่ 3.7 การเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์กับอัลตราโซนิกเซนเซอร์

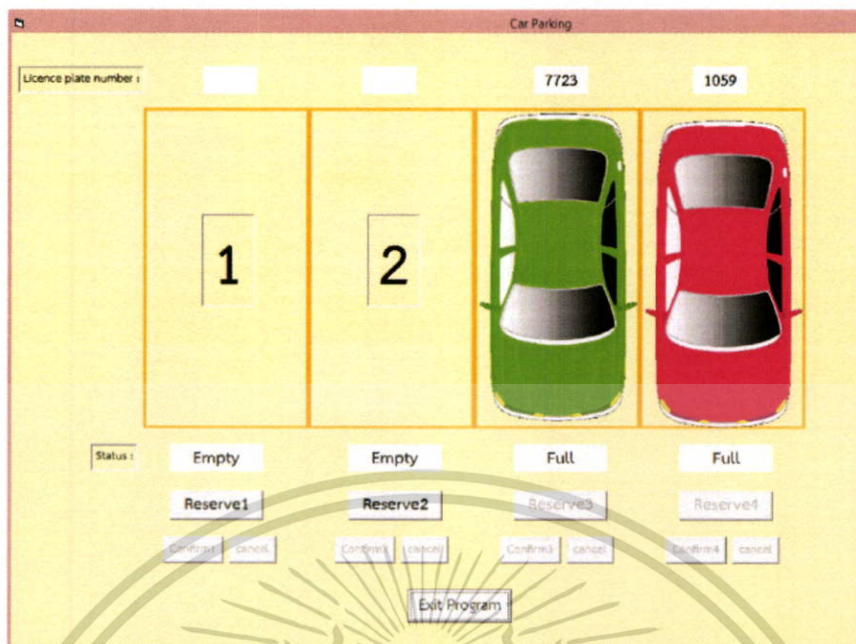
3.1.3 หน้าจอแสดงผลที่ออกแบบโดยวิซวลเบสิก

ในโครงการนี้จะมีหน้าจอแสดงผลอยู่ 2 ส่วนด้วยกันคือ

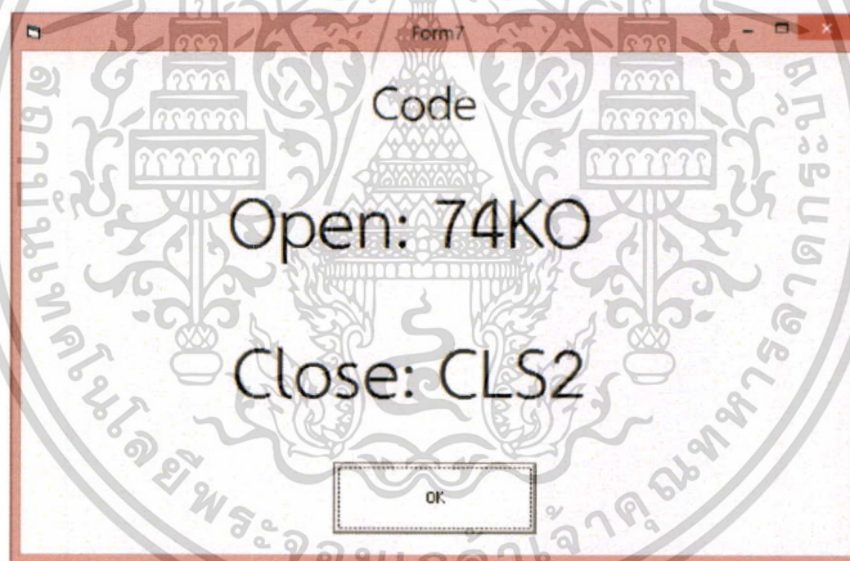
3.1.3.1 หน้าจอแสดงผลลานจอดรถ

หน้าจอแสดงผลที่ออกแบบโดยใช้โปรแกรมวิซวลเบสิกจะแสดงช่องจอดรถทั้งหมดเพื่อให้ผู้ใช้งานทราบว่าตำแหน่งของช่องจอดรถใดบ้างที่สามารถนำรถเข้าจอดได้ ดังรูปที่ 3.8 จากนั้นผู้ใช้งานจะทำการกดจองช่องจอดรถที่ว่างอยู่ ผู้ใช้งานสามารถกดปุ่มยืนยันเพื่อรับรหัสเปิดปิดไม้กั้นหรือกดปุ่มยกเลิกช่องจอดรถนั้นเพื่อจองช่องจอดรถอื่นได้ โดยรหัสเปิดปิดไม้กั้นจะแสดงเฉพาะผู้ใช้งานกดปุ่มยืนยันเท่านั้น ดังรูปที่ 3.9

เอกสารผู้ใช้งานกดปุ่มยืนยันเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 หน้าจอแสดงผลโดยใช้โปรแกรมวิซวลเบสิก

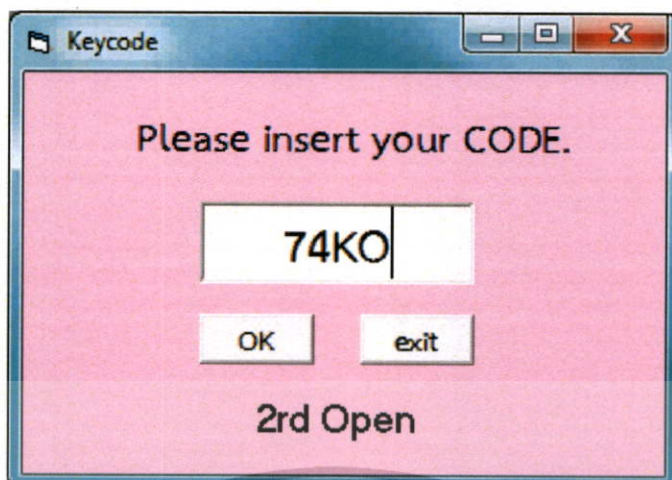


รูปที่ 3.9 หน้าจอแสดงรหัสเปิดปิดไม้กั้น

3.1.3.2 หน้าจอสำหรับใส่รหัสเปิดปิดไม้กั้น

หน้าจอสำหรับใส่รหัสเปิดปิดไม้กั้นที่ออกแบบโดยใช้โปรแกรมวิซวลเบสิกจะให้ผู้ใช้งานนำรหัสที่ได้จากการกดปุ่มยืนยัน ดังรูปที่ 3.9 มาใส่ในกล่องข้อความของหน้าจอสำหรับใส่รหัสเปิดปิดไม้กั้น ดังรูปที่ 3.10 จากนั้นกดปุ่มตกลงเพื่อเปิดไม้กั้นขึ้น แล้วจึงจะสามารถนำรถยนต์เข้าจอดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 หน้าจอสำหรับใส่รหัสเปิดปิดไม้กั้น

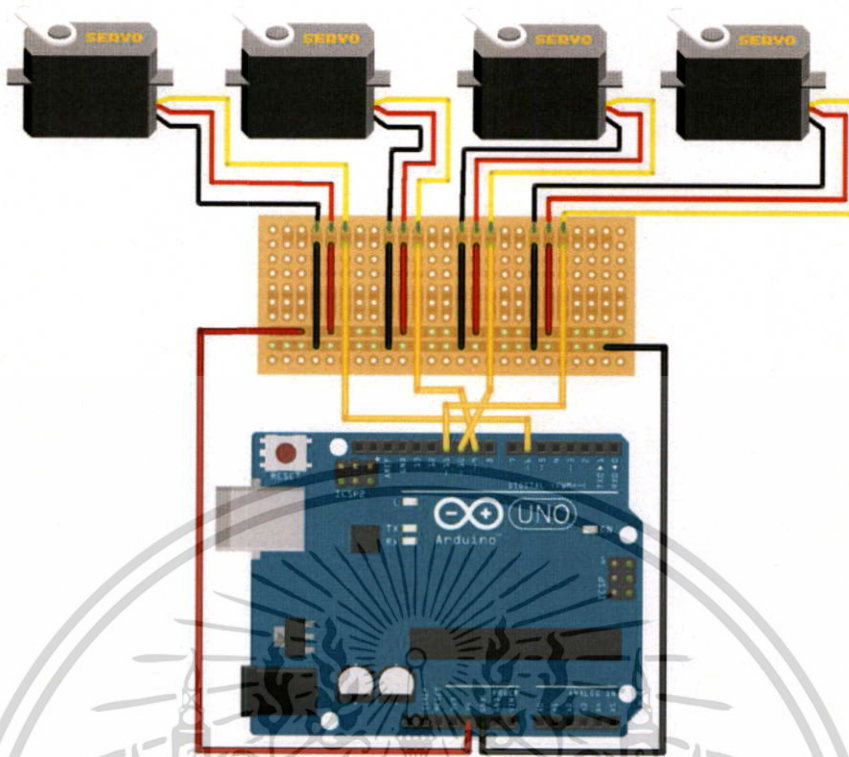
3.1.4 เซอร์โวมอเตอร์ SG90

ในการที่รถยนต์จะสามารถเข้าไปจอดในช่องจอดรถที่ได้ทำการจองได้นั้น จะต้องทำการใส่รหัสเพื่อเปิดไม้กั้นก่อน ในที่นี้เลือกใช้เซอร์โวมอเตอร์ SG90 ดังรูปที่ 3.11 สำหรับการเปิดปิดไม้กั้น เหตุผลที่เลือกใช้เซอร์โวมอเตอร์ SG90 เนื่องจากเซอร์โวมอเตอร์ตัวนี้มีขนาดเล็ก น้ำหนักเบา เหมาะกับโครงสร้างลานจอดรถที่ได้ทำการออกแบบไว้ หลังจากทำการใส่รหัสแล้ว ไมโครคอนโทรลเลอร์จะตรวจสอบว่ารหัสที่ใส่เข้ามาตรงกันกับรหัสสำหรับเปิดไม้กั้นแล้ว ไมโครคอนโทรลเลอร์ก็จะทำการแปลงค่าจากโปรแกรมที่กำหนดค่าขององศาไว้แล้วส่งสัญญาณพัลส์ไปยังเซอร์โวมอเตอร์ ซึ่งเซอร์โวมอเตอร์ก็จะทำงานตามค่าของสัญญาณพัลส์ที่ได้รับจากไมโครคอนโทรลเลอร์ โดยทำการเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์กับเซอร์โวมอเตอร์ดังรูปที่ 3.12



รูปที่ 3.11 เซอร์โวมอเตอร์ SG90

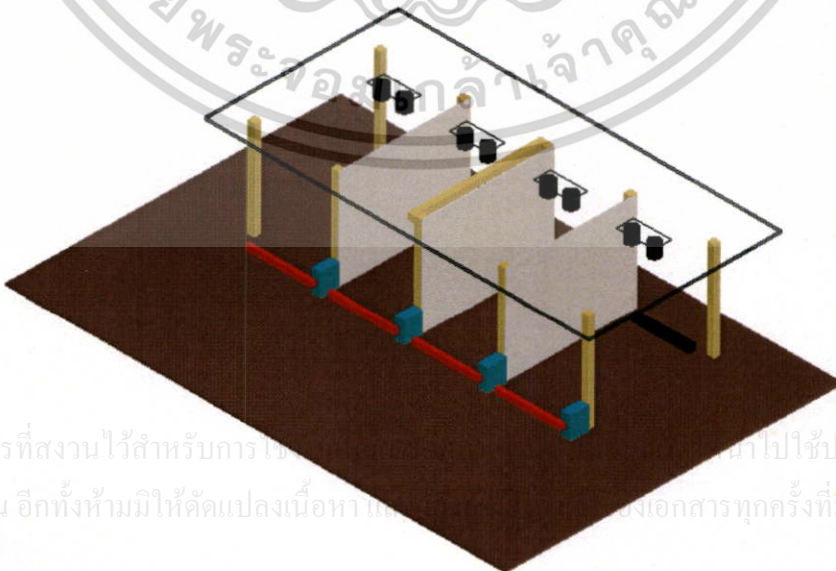
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์กับเซอร์โวมอเตอร์

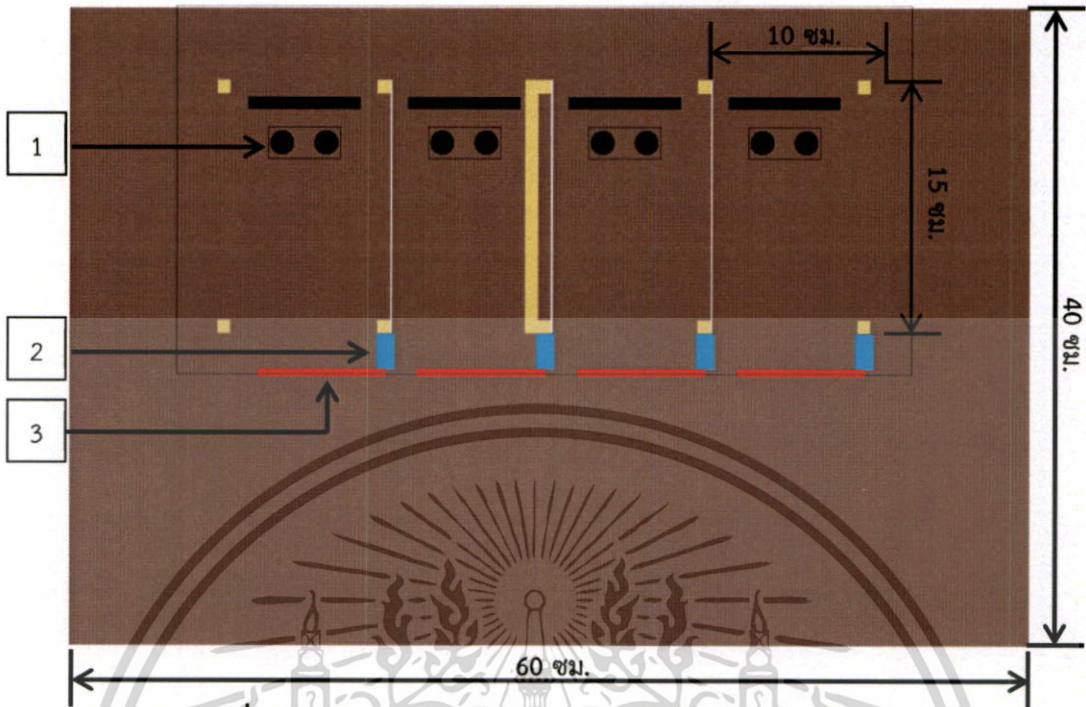
3.2 ออกแบบโครงสร้างลานจอดรถ

การออกแบบโครงสร้างลานจอดรถใช้แผ่นไม้ขนาดกว้าง 40 เซนติเมตร ยาว 60 เซนติเมตร เป็นฐานลานจอดรถทั้งหมด ในแต่ละช่องจอดรถมีขนาดกว้าง 10 เซนติเมตร ยาว 15 เซนติเมตร สูง 12 เซนติเมตร ดังรูปที่ 3.13 ถึงรูปที่ 3.16 จากรูปที่ 3.14 หมายเลข 1 คือ อัลตราโซนิกเซนเซอร์ หมายเลข 2 คือ เซอร์โวมอเตอร์ และหมายเลข 3 คือ ไม้กั้นรถยนต์

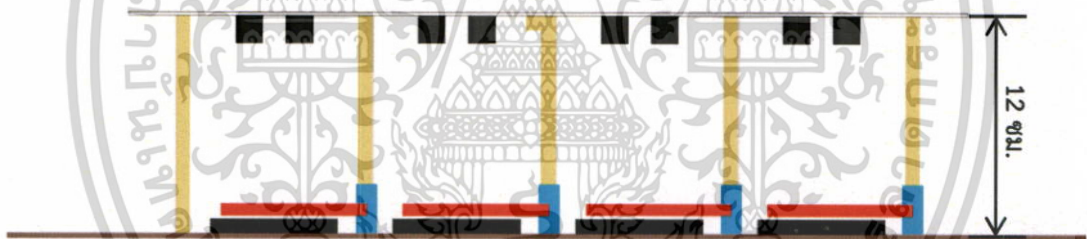


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและข้อมูลของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.13 มุมมองแบบจำลองลานจอดรถแบบไอโซเมตริก (Isometric)



รูปที่ 3.14 มุมมองแบบจำลองลานจอดรถด้านบน (Top view)



รูปที่ 3.15 มุมมองแบบจำลองลานจอดรถด้านหน้า (Front View)



รูปที่ 3.16 มุมมองแบบจำลองลานจอดรถด้านข้าง (Side view)

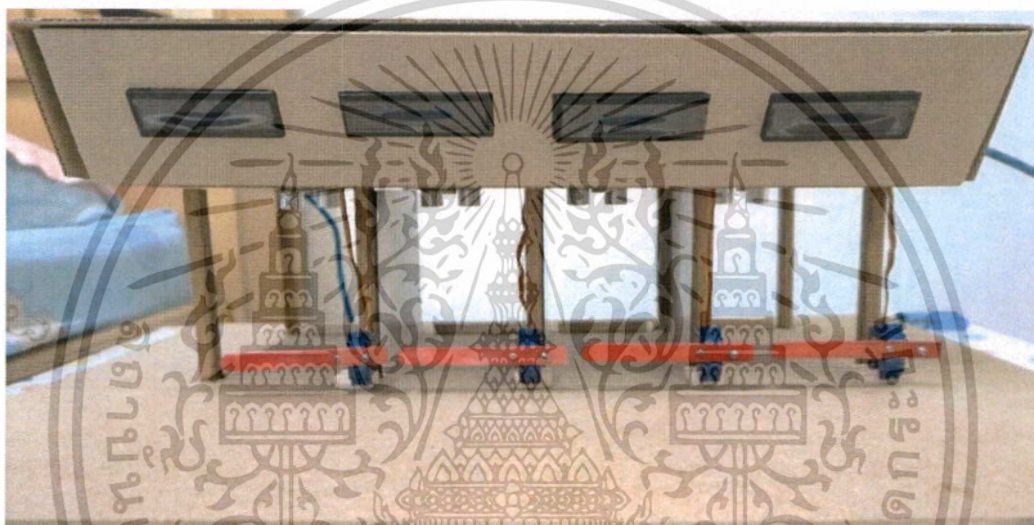
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 แบบจำลองการทำงาน

สร้างแบบจำลองการทำงานเป็นลานจอดรถยนต์ 1 ชั้น ซึ่งมีจำนวนตำแหน่งที่จอดรถยนต์ 4 ตำแหน่ง ติดตั้งเซนเซอร์ ไมโครคอนโทรลเลอร์ เซอร์โวมอเตอร์ จอแสดงผลแอลซีดี และคอมพิวเตอร์ ดังแสดงในรูปที่ 4.1 ถึงรูปที่ 4.4 ตามลำดับ

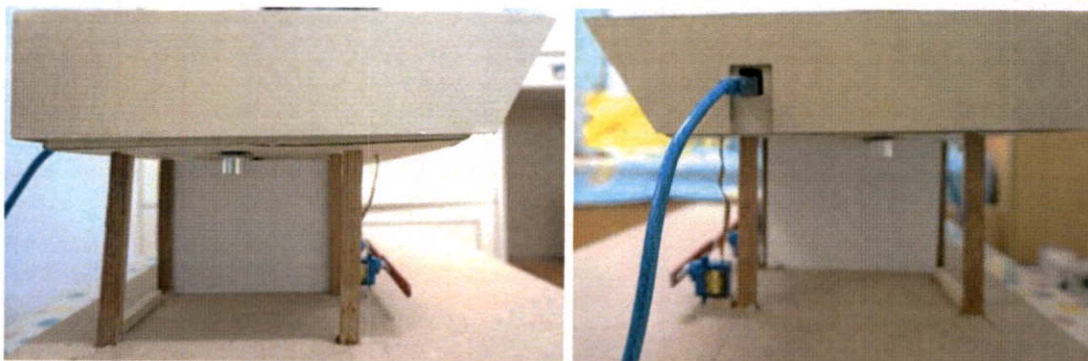


รูปที่ 4.1 มุมมองโครงสร้างแบบจำลองลานจอดรถจากด้านหน้า

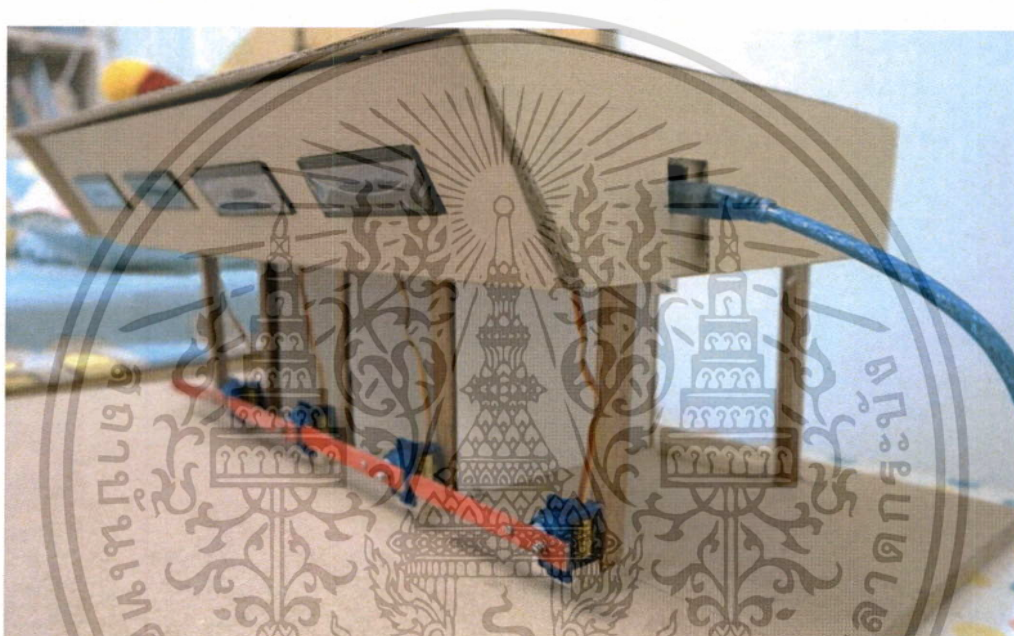


รูปที่ 4.2 มุมมองโครงสร้างแบบจำลองลานจอดรถจากด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 มุมมองโครงสร้างแบบจำลองลานจอดรถจากด้านซ้าย และด้านขวา



รูปที่ 4.4 มุมมองโครงสร้างแบบจำลองลานจอดรถแบบไอโซเมตริก

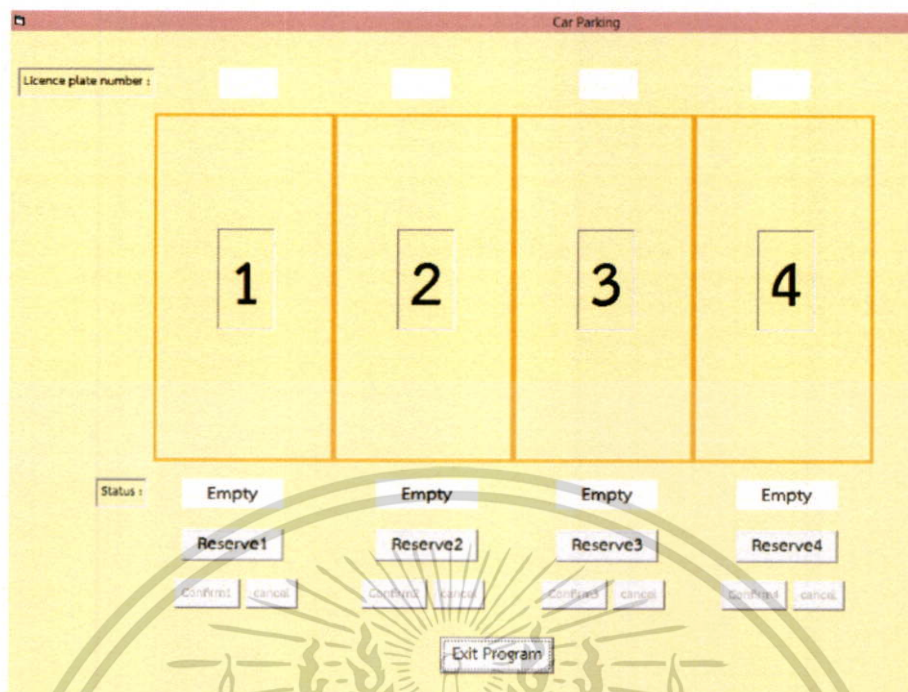
4.2 การแสดงภาพของตำแหน่งที่จอดรถยนต์

ส่วนของหน้าจอแสดงผลที่ใช้โปรแกรมวิซวลเบสิกในการออกแบบ จะแสดงภาพของตำแหน่งที่จอดรถได้ 3 แบบ คือ

- ช่องจอดรถว่าง
- ช่องจอดรถมีการจอง
- ช่องจอดรถมีรถเข้าจอด

4.2.1 ช่องจอดรถว่าง

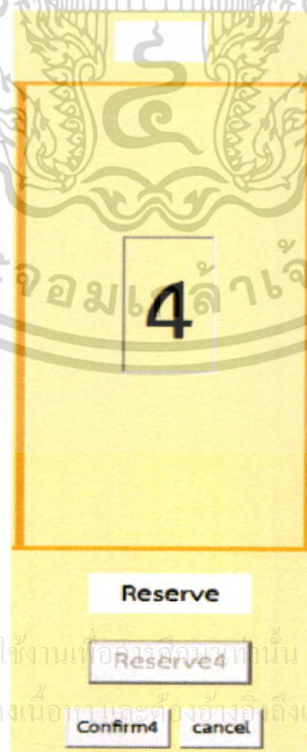
ในกรณีที่ไม่มีรถเข้ามาจอด ภาพในลานจอดรถจะแสดงเป็นตัวเลขที่บ่งบอกถึงตำแหน่งของช่องจอดรถ และแสดงข้อความว่า 'Empty' อยู่ใต้ภาพ ซึ่งสามารถจอง (Reserve) ตำแหน่งของช่องจอดรถได้ ดังรูปที่ 4.5



รูปที่ 4.5 ตำแหน่งของช่องจอดรถที่ไม่มีรถเข้าจอด

4.2.2 ช่องจอดรถมีการจอง

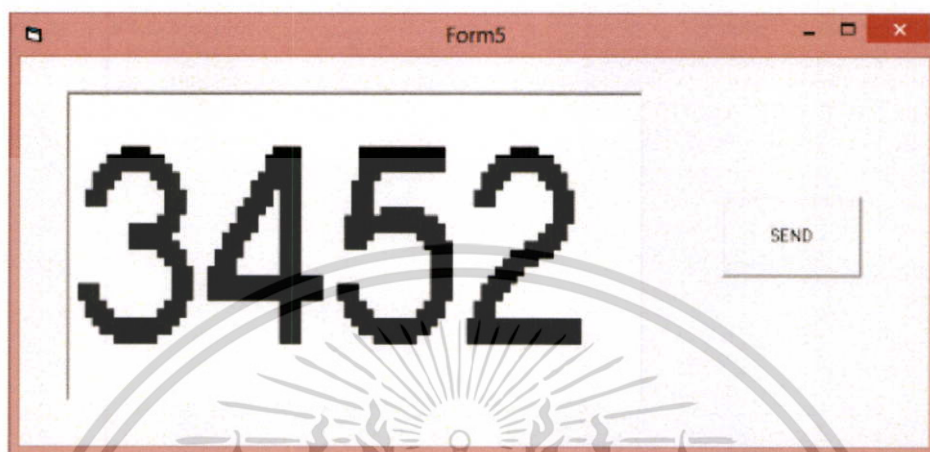
เมื่อผู้ใช้งานได้ทำการกดจองในตำแหน่งที่ต้องการแล้ว ตำแหน่งนั้นจะไม่สามารถกดจองได้อีก และข้อความใต้ภาพจากคำว่า 'Empty' จะเปลี่ยนเป็นคำว่า 'Reserve' ดังรูปที่ 4.6



รูปที่ 4.6 ตำแหน่งของช่องจอดรถที่มีการจอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่กดจองแล้วจะมีหน้าต่างใหม่แสดงขึ้นมา เพื่อให้พิมพ์ทะเบียนรถที่จะเข้าจอดในช่องนั้น ดังรูปที่ 4.7 และเมื่อพิมพ์ทะเบียนเสร็จแล้ว ก็จะกลับมายังหน้าจอแสดงผลเดิม พร้อมกับแสดงหมายเลขทะเบียนบนช่องจองที่ทำการจอง ดังรูปที่ 4.8



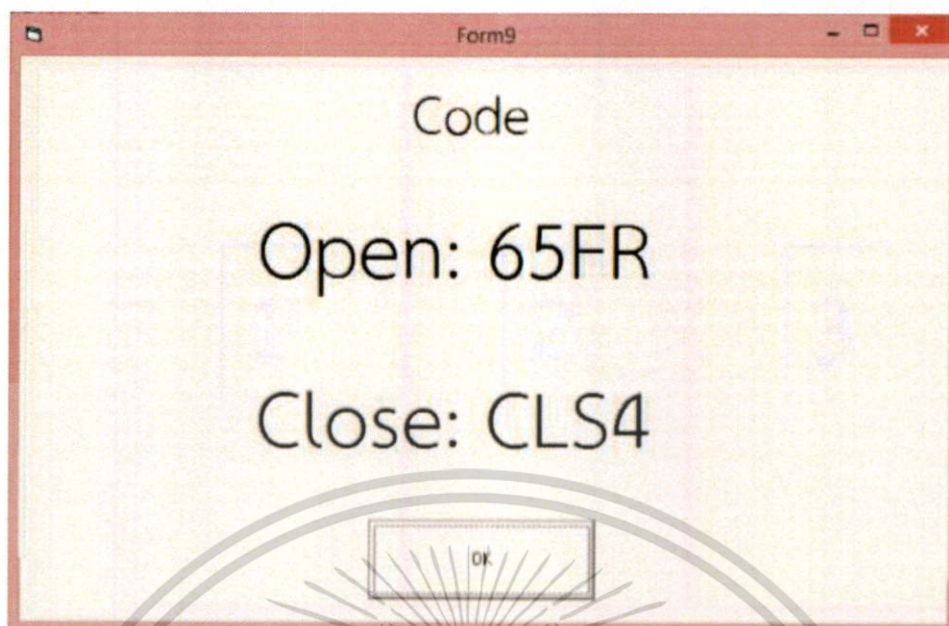
รูปที่ 4.7 หน้าจอแสดงผลสำหรับกรอกหมายเลขทะเบียน



รูปที่ 4.8 ตำแหน่งของช่องจองรถที่มีการจองพร้อมด้วยหมายเลขทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ หากมีข้อสงสัยหรือข้อผิดพลาด กรุณาติดต่อฝ่ายเอกสารหรือฝ่ายที่ดูแลระบบ

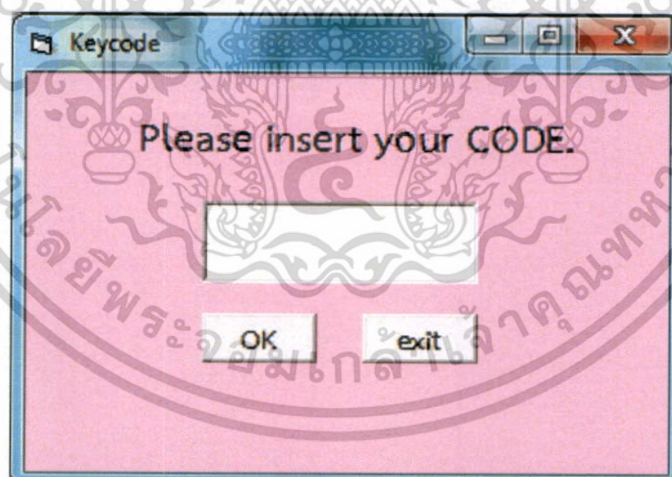
เมื่อผู้ใช้งานได้กดปุ่ม 'Confirm' เพื่อยืนยันที่จะเข้าจอดในตำแหน่งนั้น จะได้รับรหัสที่จะนำไปใช้ในการเปิดปิดที่กั้นของช่องนั้น ๆ ดังรูปที่ 4.9 หรือถ้าผู้ใช้งานกดปุ่ม 'cancel' ช่องจองนั้นจะถูกทำการยกเลิกการจองโดยอัตโนมัติ



รูปที่ 4.9 รหัสใช้ในการเปิด - ปิด ที่กั้นช่อง

4.2.3 ช่องจอดรถมีรถเข้าจอด

ก่อนที่จะนำรถเข้ามาจอด จะมีหน้าจอแสดงผลสำหรับให้กรอกรหัสที่ได้รับมาจากการจอง เพื่อใช้สำหรับเปิดปิดที่กั้นของช่องจอดนั้น ดังรูปที่ 4.10 จากนั้นก็กรอกรหัสที่ได้รับมา ดังรูปที่ 4.11



รูปที่ 4.10 หน้าจอแสดงผลสำหรับกรอกรหัสที่ใช้เปิดปิดที่กั้นช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



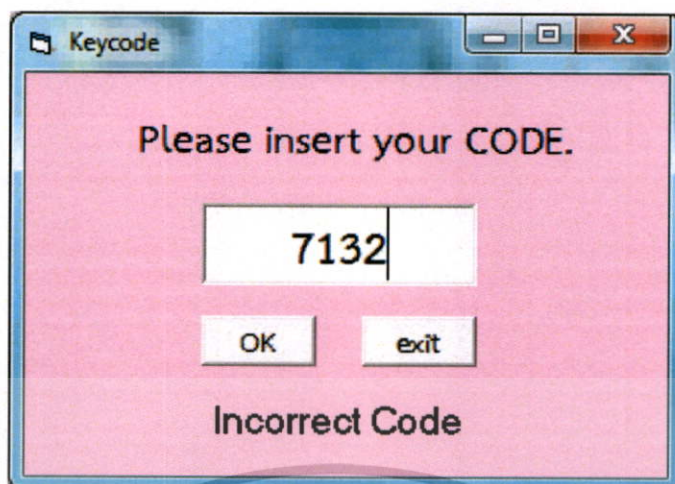
รูปที่ 4.11 กรอกรหัสที่ได้รับมาจากการจอง

ถ้ากรอกรหัสที่ได้รับมาถูก ที่หน้าจอแสดงผลจะขึ้นดังรูปที่ 4.12 พร้อมกับเซอร์โวมอเตอร์ก็จะทำงานตามค่าของสัญญาณพัลส์ที่ได้รับจากไมโครคอนโทรลเลอร์ ทำให้ที่กั้นในช่องนั้นเปิดขึ้น แต่ถ้ากรอกรหัสที่ได้รับมาไม่ถูกต้อง ที่หน้าจอแสดงผลจะขึ้นดังรูปที่ 4.13



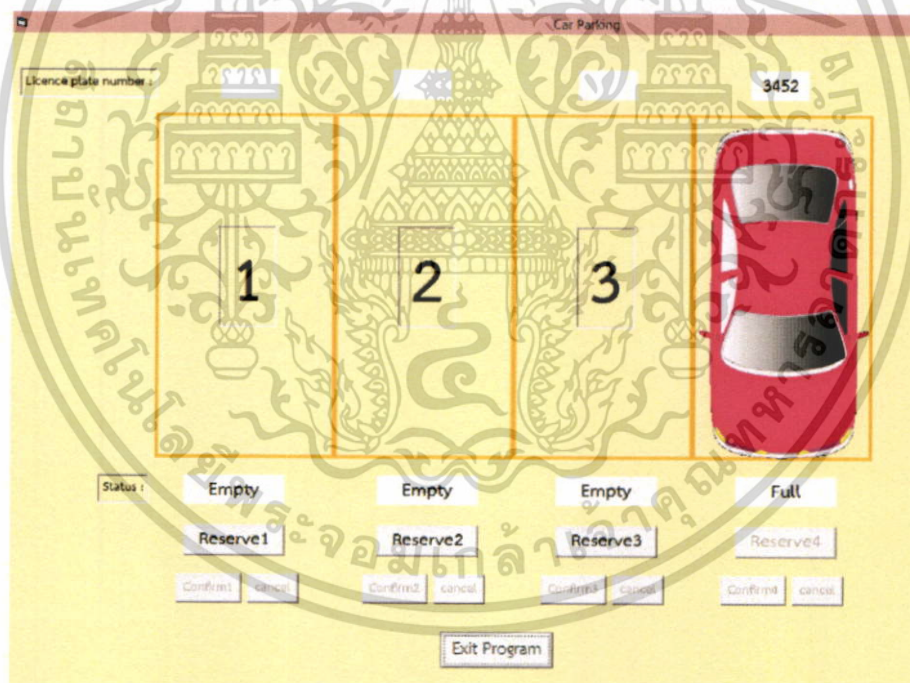
รูปที่ 4.12 รหัสที่กรอกถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 รหัสที่กรอกไม่ถูกต้อง

หลังจากกรอกรหัสที่ใช้เปิดที่กั้นเสร็จ และนำรถเข้ามาจอด ตำแหน่งช่องจอดรถนั้นจะแสดงเป็นภาพรถ และแสดงข้อความว่า 'Full' อยู่ใต้ภาพ ดังรูปที่ 4.14



รูปที่ 4.14 ตำแหน่งของช่องจอดรถที่มีรถเข้าจอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองวงจรตรวจจับรถ

บริเวณส่วนของลานจอดรถได้ทำการติดตั้งเซนเซอร์สำหรับตรวจจับรถไว้บริเวณหลังคาของลานจอดรถที่ห่างจากพื้นลานจอดรถ 7 เซนติเมตร โดยเซนเซอร์ทำการเชื่อมต่อกับ บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และส่งข้อมูลไปยังคอมพิวเตอร์ ภายในบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้ทำการบรรจุโปรแกรมไว้ว่า เซนเซอร์จะทำการวัดระยะจากเซนเซอร์ไปยังวัตถุทุก ๆ 2 ไมโครวินาที และส่งค่าให้ไมโครคอนโทรลเลอร์ประมวลผล หากเซนเซอร์ตรวจพบว่าระยะวัตถุต่ำกว่า 7 เซนติเมตร ซึ่งหมายถึงมีรถเข้ามาจอด ไมโครคอนโทรลเลอร์ จะทำการประมวลผล และส่งข้อมูลไปยังโปรแกรมวิซวลเบสิก ผ่านซิกบี โดยหน้าจอแสดงผลจะทำการแสดงภาพรถในช่องจอดรถที่มีรถจอดอยู่ ดังรูปที่ 4.16



รูปที่ 4.15 แบบจำลองลานจอดรถขณะมีรถเข้าจอด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังเป็นการผิดกฎหมาย และต้องรับผิดชอบถึงเจ้าของเอกสารหากมีใครนำออกไปใช้

รูปที่ 4.16 ตำแหน่งของช่องจอดรถที่มีรถจอดตามที่เซนเซอร์วัดได้

4.4 การทดลองวงจรเปิดปิดไม้กั้น

บริเวณส่วนด้านหน้าของช่องจอดรถได้ทำการติดตั้งเซอร์โวมอเตอร์ โดยเซอร์โวมอเตอร์ทำการเชื่อมต่อกับ บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 และส่งข้อมูลไปยังคอมพิวเตอร์ ภายในบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ได้ทำการบรรจุโปรแกรมไว้ว่าเมื่อผู้ใช้กรอกรหัสแล้ว ไมโครคอนโทรลเลอร์จะตรวจสอบว่ารหัสที่ใส่เข้ามาตรงกันกับรหัสสำหรับเปิดปิดไม้กั้นหรือไม่ ถ้ารหัสตรงกัน ไมโครคอนโทรลเลอร์จะทำการแปลงค่าจากโปรแกรมที่กำหนดค่าขององศาไว้แล้วส่งสัญญาณพัลส์ไปยังเซอร์โวมอเตอร์ โดยเซอร์โวมอเตอร์จะทำงานตามค่าของสัญญาณพัลส์ที่ได้รับจากไมโครคอนโทรลเลอร์ จะทำให้ไม้กั้นเปิดขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากการทำโครงงานปริญญานิพนธ์ พบว่าระบบแสดงผลลานจอดรถนั้นเป็นไปตามที่ต้องการได้ โดยในส่วนของลานจอดรถสามารถนำรถเข้าไปจอดได้จำนวน 4 คัน โดยจะมีอุปกรณ์เซนเซอร์ติดตั้งอยู่บริเวณหลังคาของลานจอดรถที่สามารถตรวจจับรถยนต์ที่เข้ามาจอดในช่องจอดรถยนต์ได้โดยการควบคุมผ่านทางไมโครคอนโทรลเลอร์ Arduino Uno R3 และส่งค่าจากการตรวจจับไปยังคอมพิวเตอร์ผ่านซิกบี เพื่อแสดงผลออกทางโปรแกรมมิชวลเบสิก โดยโปรแกรมดังกล่าวสามารถแสดงตำแหน่งที่มีรถยนต์จอดภายในบริเวณช่องจอดแต่ละตำแหน่งได้จึงทำให้ทราบว่าภายในบริเวณลานจอดรถนี้มีช่องจอดรถใดบ้างที่สามารถนำรถเข้าไปจอดได้ และเซอร์โวมอเตอร์ยังสามารถเปิดปิดไม้กั้นในแต่ละช่องจอดรถได้หากผู้ใช้งานกรอกรหัสถูกต้อง

5.2 สรุปผลการดำเนินงาน

1. สามารถออกแบบลานจอดรถ และสร้างแบบจำลองตามทีออกแบบไว้ได้
2. สามารถเขียนโปรแกรมควบคุมบอร์ดไมโครคอนโทรลเลอร์โดยใช้โปรแกรม Arduino และออกแบบหน้าจอส่วนประสานกราฟิกกับผู้ใช้งานโดยโปรแกรมมิชวลเบสิกได้
3. สามารถนำไปประยุกต์ใช้ในลานจอดรถจริงได้ เนื่องจากผู้จัดทำเลือกใช้เซนเซอร์ที่เหมาะสมทั้งแบบจำลอง และลานจอดรถขนาดจริง
4. ผู้จัดทำมีความเข้าใจเกี่ยวกับการทำงานของเซนเซอร์ เซอร์โวมอเตอร์ และตัวคอนโทรลเลอร์มากขึ้น

5.3 ปัญหาและอุปสรรคที่พบ

1. ปัญหา การติดตั้งอัลตราโซนิกเซนเซอร์มากกว่า 1 ตัว ไว้ใกล้ ๆ กันจะก่อให้เกิดการรบกวนกันเอง มีผลทำให้ตัวเซนเซอร์อ่านค่าระยะทางผิดพลาด
2. ปัญหา ทางผู้จัดทำใช้เวลาในการศึกษาโปรแกรมมิชวลเบสิกค่อนข้างนานเนื่องจากผู้จัดทำไม่เคยศึกษาโปรแกรมนี้มาก่อน
3. ปัญหา เซอร์โวมอเตอร์ทำงานผิดปกติในช่วงเวลาหนึ่ง เนื่องจากความรู้เท่าไม่ถึงการณ์ของผู้จัดทำที่ใช้แหล่งจ่ายไฟเดียวกันสำหรับบอร์ดไมโครคอนโทรลเลอร์ 2 บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 แนวทางแก้ไข

1. นำกระดาษหรือวัสดุที่บวมกันระหว่างช่องจอตรง เพื่อป้องกันไม่ให้อัลตราโซนิกเซนเซอร์แต่ละตัวรบกวนการทำงานของกันและกัน
2. ในการศึกษาโปรแกรมวิซวลเบสิก ทางผู้จัดทำได้แก้ไขปัญหาโดยศึกษาค้นคว้าข้อมูลเพิ่มเติมจากหนังสือ อินเทอร์เน็ต และสอบถามผู้เชี่ยวชาญทำให้สามารถใช้งานโปรแกรมวิซวลเบสิกได้อย่างมีประสิทธิภาพ
3. แยกแหล่งจ่ายไฟสำหรับบอร์ดไมโครคอนโทรลเลอร์แต่ละตัว

5.5 ข้อเสนอแนะและแนวทางในการพัฒนา

1. สามารถนำไปประยุกต์ใช้กับลานจอดรถที่มีขนาดใหญ่ได้จริง เพราะเซนเซอร์สามารถวัดระยะของลานจอดรถจริงได้
2. ไม่เกินสำหรับแต่ละช่องจอดรถ อาจพัฒนาให้มีขนาดเล็กลงได้
3. การป้อนเลขทะเบียนเพื่อให้แสดงบนหน้าจอแอลซีดี อาจเปลี่ยนไปใช้เทคโนโลยีการประมวลผลภาพหรือวิธีอื่น ๆ ที่สามารถสแกนทะเบียนรถยนต์เพื่อแสดงบนหน้าจอแอลซีดีได้เลย เพื่อสะดวกต่อผู้ใช้งาน
4. การใส่รหัสเปิดปิดไม้กั้น อาจเปลี่ยนเป็นเครื่องอ่านบัตรประเภทอาร์เอฟไอดีหรือวิธีอื่น ๆ ที่สะดวกต่อผู้ใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] อีระศักดิ์ สุโชตินันท์, ประยุทธ อินแบน. โปรแกรมเมอร์มือใหม่ ทัดเขียนโปรแกรม Microsoft Visual Basic 6.0 Enterprise Edition. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 2549.
- [2] วรพงศ์ ตั้งศรีรัตน์. เซนเซอร์และทรานสดิวเซอร์. พิมพ์ครั้งที่ 5. กรุงเทพมหานคร : สำนักพิมพ์ สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2551.
- [3] อภิชาติ ภู่อัลป์. เริ่มต้นเขียนโปรแกรมติดต่อ และควบคุมฮาร์ดแวร์ด้วย Visual Basic. พิมพ์ครั้งที่ 1. นนทบุรี : อินโฟเพรส. 2546.
- [4] อภิชาติ ภู่อัลป์. สนุกกับการประยุกต์ใช้ Visual Basic. พิมพ์ครั้งที่ 1. นนทบุรี : อินโฟเพรส. 2546.
- [5] อภิชาติ ภู่อัลป์. รวมคอนโทรลเลอร์ สำหรับ VB 6. พิมพ์ครั้งที่ 1. นนทบุรี : ไอดีซี. 2547.
- [6] อภิรัตน์ บางศิริ. Autocad 2012 สำหรับผู้เริ่มต้น. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : ชิมพลีฟาย. 2555.
- [7] ยณลักษณ์ ตันตีสวรรณา, รุ่งวิทย์ เผือกขุ่ม, วัฒนวิรัช เขียวอยู่, สะท้านไกล ฤงวิชา. “ระบบแสดงผลสถานะจอตลอด.” ปริญญาโทวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมระบบควบคุม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2553.
- [8] Adisak Chinawong. “เริ่มต้น AVR Microcontroller.” [Online]. Available : <http://www.adisak51.com/avr2.html>. 2552.
- [9] Mondop Chaibundit. “Arduino Microcontroller electronics and telecommunication.” [Online]. Available : <http://mondop.blogspot.com/2013/11/arduino-microcontroller.html>. 2556.
- [10] “รู้จักไมโครคอนโทรลเลอร์.” [Online]. Available : http://www.semi-shop.com/knowledge/knowledge_detail.php?sk_id=13. 2554
- [11] “16F877 ควบคุมเซอร์โวมอเตอร์.” [Online]. Available : http://www.mcuthailand.com/articles/pic/Ex_Servo.html. 2556.
- [12] “Arduino กับ การขับเซอร์โวมอเตอร์เบื้องต้น.” [Online]. Available : http://www.semi-shop.com/knowledge/knowledge_detail.php?sk_id=105. 2556.
- [13] “Simple Arduino and HC-SR04 Example.” [Online]. Available : <http://www.instructables.com/id/Simple-Arduino-and-HC-SR04-Example>. 2556.

เอกสารนี้ [14] “Ultrasonic Theory.” [Online]. Available : <http://www.compomax.co.th/product/ultrasonic-theory>. 2556. ทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมการประมวลผลภาพ และโปรแกรมควบคุม

ก.1 โปรแกรมวิซวลเบสิกที่ใช้ในส่วนของหน้าจอแสดงผลสำหรับการจอง

- Form 1

```
Private Sub can1_Click()
```

```
lblsend.Caption = "i"
```

```
MSComm1.Output = lblsend.Caption
```

```
Command1.Enabled = True
```

```
confirm1.Enabled = False
```

```
can1.Enabled = False
```

```
Form2.E = ""
```

```
lb1.Caption = Form2.E
```

```
lbl_status(0).Caption = "Empty"
```

```
End Sub
```

```
Private Sub can2_Click()
```

```
lblsend.Caption = "j"
```

```
MSComm1.Output = lblsend.Caption
```

```
Command2.Enabled = True
```

```
confirm2.Enabled = False
```

```
can2.Enabled = False
```

```
Form3.F = ""
```

```
lb2.Caption = Form3.F
```

```
lbl_status(1).Caption = "Empty"
```

```
End Sub
```

```
Private Sub can3_Click()
```

```
lblsend.Caption = "k"
```

```
MSComm1.Output = lblsend.Caption
```

```
Command3.Enabled = True
```

```
confirm3.Enabled = False
```

เอกสารนี้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อผิดพลาดหรือต้องการแจ้งเรื่อง กรุณาแจ้งไปยังฝ่ายพัฒนาระบบสารสนเทศของกรมส่งเสริมการค้าระหว่างประเทศ

```

can3.Enabled = False
Form4.G = ""
lb3.Caption = Form4.G
lbl_status(2).Caption = "Empty"
End Sub

```

```

Private Sub can4_Click()
lblsend.Caption = "m"
MSComm1.Output = lblsend.Caption
Command4.Enabled = True
confirm4.Enabled = False
can4.Enabled = False
Form5.H = ""
lb4.Caption = Form5.H
lbl_status(3).Caption = "Empty"
End Sub

```

```

Private Sub cmd_EXIT_Click()
End
End Sub

```

```

Private Sub Command1_Click()
lbl_status(0).Caption = "Reserve"
lblsend.Caption = "i"
MSComm1.Output = lblsend.Caption
Form2.Show
Command1.Enabled = False
can1.Enabled = True
confirm1.Enabled = True
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Private Sub Command2_Click()
 ไม่ว่าจะทำได้จากทั้งต้น อีกทั้งห้ามมิให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 lbl_status(1).Caption = "Reserve"

```

lblsend.Caption = "j"
MSComm1.Output = lblsend.Caption
Form3.Show
Command2.Enabled = False
can2.Enabled = True
confirm2.Enabled = True
End Sub

```

```

Private Sub Command3_Click()
lbl_status(2).Caption = "Reserve"
lblsend.Caption = "k"
MSComm1.Output = lblsend.Caption
Form4.Show
Command3.Enabled = False
can3.Enabled = True
confirm3.Enabled = True
End Sub

```

```

Private Sub Command4_Click()
lbl_status(3).Caption = "Reserve"
lblsend.Caption = "m"
MSComm1.Output = lblsend.Caption
Form5.Show
Command4.Enabled = False
can4.Enabled = True
confirm4.Enabled = True
End Sub

```

```

Private Sub confirm1_Click()
"MsgBox "Code is 4562", , _
""Code"
Form6.Show
AA = Form2.E

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะตีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSComm1.Output = AA
confirm1.Enabled = False
can1.Enabled = False
End Sub

```

```

Private Sub confirm2_Click()
Form7.Show
BB = Form3.F
MSComm1.Output = BB
confirm2.Enabled = False
can2.Enabled = False
End Sub

```

```

Private Sub confirm3_Click()
Form8.Show
CC = Form4.G
MSComm1.Output = CC
confirm3.Enabled = False
can3.Enabled = False
End Sub

```

```

Private Sub confirm4_Click()
Form9.Show
DD = Form5.H
MSComm1.Output = DD
confirm4.Enabled = False
can4.Enabled = False
End Sub

```

```

Private Sub Form_Activate()
lb1.Caption = Form2.E
lb1.Alignment = vbCenter
lb2.Caption = Form3.F

```

```

lb2.Alignment = vbCenter
lb3.Caption = Form4.G
lb3.Alignment = vbCenter
lb4.Caption = Form5.H
lb4.Alignment = vbCenter
End Sub

```

```

Private Sub Form_Load()
MSComm1.RThreshold = 1
MSComm1.InputLen = 1
MSComm1.Settings = "9600,n,8,1"
MSComm1.CommPort = 4
MSComm1.PortOpen = True
MSComm1.DTREnable = False
can1.Enabled = False
confirm1.Enabled = False
can2.Enabled = False
confirm2.Enabled = False
can3.Enabled = False
confirm3.Enabled = False
can4.Enabled = False
confirm4.Enabled = False
End Sub


```

```

Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False
End Sub

```

```

Private Sub MSComm1_OnComm()
If MSComm1.CommEvent = comEvReceive Then
Label1.Caption = MSComm1.Input
If Label1.Caption = "a" Then
.Visible = True

```

เอกสารนี้เป็นเอกสารทูลงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ลิงก์ทั้งห้านี้ให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lbl_status(0).Caption = "Full"
Elseif Label1.Caption = "b" Then
  Img_car(0).Visible = False
If lbl_status(0).Caption = "Full" Then
  lbl_status(0).Caption = "Empty"
  If Img_car(0).Visible = False Then
    Form2.E = ""
    lb1.Caption = Form2.E
  End If
Command1.Enabled = True
End If

```

```

Elseif Label1.Caption = "c" Then
  Img_car(1).Visible = True
  lbl_status(1).Caption = "Full"
Elseif Label1.Caption = "d" Then
  Img_car(1).Visible = False
If lbl_status(1).Caption = "Full" Then
  lbl_status(1).Caption = "Empty"
  If Img_car(1).Visible = False Then
    Form3.F = ""
    lb2.Caption = Form3.F
  End If
Command2.Enabled = True
End If

```

```

Elseif Label1.Caption = "e" Then
  Img_car(2).Visible = True
  lbl_status(2).Caption = "Full"
Elseif Label1.Caption = "f" Then

```

```

  Img_car(2).Visible = False
  If lbl_status(2).Caption = "Full" Then
    lbl_status(2).Caption = "Empty"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If Img_car(2).Visible = False Then
Form4.G = ""
lb3.Caption = Form4.G
End If
Command3.Enabled = True
End If

```

```

Elseif Label1.Caption = "g" Then
Img_car(3).Visible = True
lbl_status(3).Caption = "Full"
Elseif Label1.Caption = "h" Then
Img_car(3).Visible = False
If lbl_status(3).Caption = "Full" Then
lbl_status(3).Caption = "Empty"
If Img_car(3).Visible = False Then
Form5.H = ""
lb4.Caption = Form5.H
End If
Command4.Enabled = True
End If
End If
End If
End Sub

```

- Form 2

```
Public E As String
```

```
Private Sub send1_Click()
```

```
E = data1.Text
```

```
Form2.Hide
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub data1_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
KeyAscii = 0
```

```
send1_Click
```

```
Elseif KeyAscii = 27 Then
```

```
Form1.Show
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
data1.SetFocus
```

```
data1.Text = E
```

```
data1.Text = ""
```

```
End Sub
```

```
Private Sub data1_Change()
```

```
data1.Font.Size = 150
```

```
End Sub
```

- Form 3

```
Public F As String
```

```
Private Sub send2_Click()
```

```
F = data2.Text
```

```
Form3.Hide
```

```
End Sub
```

```
Private Sub data2_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
KeyAscii = 0
```

```
send2_Click
```

```
Elseif KeyAscii = 27 Then
```

```
Form1.Show
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
End Sub
```

```
Private Sub Form_Activate()
data2.SetFocus
data2.Text = F
data2.Text = ""
End Sub
```

```
Private Sub data2_Change()
data2.Font.Size = 150
End Sub
```

- Form 4

```
Public G As String
```

```
Private Sub send3_Click()
G = data3.Text
Form4.Hide
End Sub
```

```
Private Sub data3_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
KeyAscii = 0
send3_Click
Elseif KeyAscii = 27 Then
Form1.Show
End If
End Sub
```

```
Private Sub Form_Activate()
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 data3.SetFocus
 data3.Text = G

```
data3.Text = ""
```

```
End Sub
```

```
Private Sub data3_Change()
```

```
data3.Font.Size = 150
```

```
End Sub
```

- Form 5

```
Public H As String
```

```
Private Sub send4_Click()
```

```
H = data4.Text
```

```
Form5.Hide
```

```
End Sub
```

```
Private Sub data4_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
KeyAscii = 0
```

```
send4_Click
```

```
Elseif KeyAscii = 27 Then
```

```
Form1.Show
```

```
End If
```

```
End Sub
```

```
Private Sub Form_Activate()
```

```
data4.SetFocus
```

```
data4.Text = H
```

```
data4.Text = ""
```

```
End Sub
```

```
Private Sub data4_Change()
```

```
data4.Font.Size = 150
```

```
End Sub
```

เอกสารนี้เป็นเอกสารทสงาน... ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Form 6

```
Private Sub Command1_Click()
```

```
Form1.Show
```

```
Form6.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Label1.Font.Size = 60
```

```
Label1.Alignment = vbCenter
```

```
Label2.Font.Size = 60
```

```
Label2.Alignment = vbCenter
```

```
Label3.Font.Size = 40
```

```
Label3.Alignment = vbCenter
```

```
End Sub
```

```
Private Sub data1_KeyPress(KeyAscii As Integer)
```

```
If KeyAscii = 13 Then
```

```
KeyAscii = 0
```

```
Command1_Click
```

```
End If
```

```
End Sub
```

- Form 7

```
Private Sub Command3_Click()
```

```
Form1.Show
```

```
Form7.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Label1.Font.Size = 40
```

```
Label1.Alignment = vbCenter
```

```
Label2.Font.Size = 60
```

```
Label2.Alignment = vbCenter
```

```

Label3.Font.Size = 60
Label3.Alignment = vbCenter
Label2.Caption = "Open: 74KO"
Label3.Caption = "Close: CLS2"
End Sub

```

```

Private Sub data1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
KeyAscii = 0
Command3_Click
End If
End Sub

```

- Form 8

```

Private Sub Command2_Click()
Form1.Show
Form8.Hide
End Sub

```

```

Private Sub Form_Load()
Label1.Font.Size = 40
Label1.Alignment = vbCenter
Label2.Font.Size = 60
Label2.Alignment = vbCenter
Label3.Font.Size = 60
Label3.Alignment = vbCenter
Label2.Caption = "Open: 98ED"
Label3.Caption = "Close: CLS3"
End Sub

```

```

Private Sub data1_KeyPress(KeyAscii As Integer)

```

```

If KeyAscii = 13 Then

```

```

KeyAscii = 0

```

```

Command2_Click
End If
End Sub

```

- Form 9

```

Private Sub Command1_Click()
Form1.Show
Form9.Hide
End Sub

```

```

Private Sub Form_Load()
Label1.Font.Size = 40
Label1.Alignment = vbCenter
Label2.Font.Size = 60
Label2.Alignment = vbCenter
Label3.Font.Size = 60
Label3.Alignment = vbCenter
Label2.Caption = "Open: 65FR"
Label3.Caption = "Close: CLS4"
End Sub

```

```

Private Sub data1_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
KeyAscii = 0
Command1_Click
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.2 โปรแกรมวิชาเบสิกที่ใช้ในส่วนของหน้าจอแสดงผลสำหรับรอกรหัสเปิดไม้กั้น

```

Private Sub Command1_Click()
Dim datas As String
datas = Text1.Text
If datas = "8976" Then
Label2.Caption = "1st open"
Elseif datas = "CLS1" Then
Label2.Caption = "1st close"
Elseif datas = "74KO" Then
Label2.Caption = "2rd Open"
Elseif datas = "CLS2" Then
Label2.Caption = "2rd Close"
Elseif datas = "98ED" Then
Label2.Caption = "3nd Open"
Elseif datas = "CLS3" Then
Label2.Caption = "3rd Close"
Elseif datas = "65FR" Then
Label2.Caption = "4th Open"
Elseif datas = "CLS4" Then
Label2.Caption = "4th Close"
Else
Label2.Caption = "Incorrect Code"
End If
MSComm1.Output = Text1.Text
End Sub

```

```

Private Sub Command2_Click()
End
End Sub

```

```

Private Sub MSComm1_OnComm()
If MSComm1.CommEvent = comEvReceive Then
Text1.Text = MSComm1.Input

```

```
End If  
End Sub
```

```
Private Sub Form_Load()  
MSComm1.RThreshold = 1  
MSComm1.InputLen = 1  
MSComm1.Settings = "9600,n,8,1"  
MSComm1.CommPort = 3  
MSComm1.PortOpen = True  
MSComm1.DTREnable = False  
End Sub
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)  
If KeyAscii = 13 Then  
KeyAscii = 0  
Command1_Click  
End If  
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.3 โปรแกรมควบคุมเซนเซอร์ และแอลซีดี

```

#include <Wire.h>
#include <SoftwareSerial.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd1(0x23, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
LiquidCrystal_I2C lcd2(0x25, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
LiquidCrystal_I2C lcd3(0x26, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
LiquidCrystal_I2C lcd4(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
uint8_t pinRx = 2 , pinTx = 4; //rx 4,tx2
long BaudRate = 9600 , sysTick = 0;
char mydata = 0,mydata2 =0;
#define trigpin1 A0
#define echopin1 A1
#define trigpin2 A2
#define echopin2 A3
#define trigpin3 7
#define echopin3 8
#define trigpin4 12
#define echopin4 13
SoftwareSerial mySerial( pinRx , pinTx );

void setup()
{
  Serial.begin(BaudRate);
  pinMode(trigpin1,OUTPUT);
  pinMode(echopin1,INPUT);
  pinMode(trigpin2,OUTPUT);
  pinMode(echopin2,INPUT);
  pinMode(trigpin3,OUTPUT);
  pinMode(echopin3,INPUT);
  pinMode(trigpin4,OUTPUT);
  pinMode(echopin4,INPUT);

```

```
mySerial.begin(BaudRate);
mySerial.println("Powered by NewSoftSerial !");
```

```
lcd1.begin(16,2);
lcd1.clear();
lcd1.backlight();
lcd1.setCursor(5, 0);
lcd1.print("LOT #1 ");
lcd1.setCursor(6, 1);
```

```
lcd2.begin(16,2);
lcd2.clear();
lcd2.backlight();
lcd2.setCursor(5, 0);
lcd2.print("LOT #2 ");
lcd2.setCursor(6, 1);
```

```
lcd3.begin(16,2);
lcd3.clear();
lcd3.backlight();
lcd3.setCursor(5, 0);
lcd3.print("LOT #3 ");
lcd3.setCursor(6, 1);
```

```
lcd4.begin(16,2);
lcd4.clear();
lcd4.backlight();
lcd4.setCursor(5, 0);
lcd4.print("LOT #4 ");
lcd4.setCursor(6, 1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น "ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น" อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void loop()
{
  sysTick++ ; // a system timer
/* //for debug
  Serial.print("Xbee Timer :");
  Serial.println(sysTick);
  mySerial.print("Xbee Timer :");
  mySerial.println(sysTick);
*/
// Monitor Rx from PC , if the data is available then read
// it to "GotChar". Then ask XBee send the data out
// wirelessly.
  long duration1, distance1, duration2, distance2, duration3, distance3, duration4,
  distance4;
  digitalWrite(trigpin1,LOW);
  delayMicroseconds(2);
  digitalWrite(trigpin1,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin1,LOW);
  duration1 = pulseIn(echopin1, HIGH);
  distance1 = duration1/58.2;

  digitalWrite(trigpin2,LOW);
  delayMicroseconds(2);
  digitalWrite(trigpin2,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigpin2,LOW);
  duration2 = pulseIn(echopin2, HIGH);
  distance2 = duration2/58.2;

  digitalWrite(trigpin3,LOW);
  delayMicroseconds(2);
  digitalWrite(trigpin3,HIGH);

```

```

delayMicroseconds(10);
digitalWrite(trigpin3,LOW);
duration3 = pulseIn(echopin3, HIGH);
distance3 = duration3/58.2;

```

```

digitalWrite(trigpin4,LOW);
delayMicroseconds(2);
digitalWrite(trigpin4,HIGH);
delayMicroseconds(10);
digitalWrite(trigpin4,LOW);
duration4 = pulseIn(echopin4, HIGH);
distance4 = duration4/58.2;

```

```

if (distance1 <=7)
{
  mySerial.print('a');
  mySerial.print(distance1);
}
else
if (distance1 >7)
{
  mySerial.print('b');
  mySerial.print(distance1);
}

```

```

if (distance2 <=7)
{
  mySerial.print('c');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 mySerial.print(distance2);
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 }

```

else
if (distance2 >7)
{
mySerial.print('d');
mySerial.print(distance2);

}

if (distance3 <=7)
{
mySerial.print('e');
mySerial.print(distance3);
}
else
if (distance3 >7)
{
mySerial.print('f');
mySerial.print(distance3);
}

if (distance4 <=7)
{
mySerial.print('g');
mySerial.println(distance4);

}

else
if (distance4 >7)
{
mySerial.print('h');
mySerial.println(distance4);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

```

```
delay(1000);

```

```
if ( mySerial.available() ) {
  mydata = mySerial.read();
  delay(100);

```

```

    if(mydata == 'i'){
      lcd1.clear();
      lcd1.setCursor(5, 0);
      lcd1.print("LOT #1 ");
      lcd1.setCursor(6, 1);
      Serial.println("geta");
      delay(10000);
      while ( mySerial.available()>0 ) {
        mydata2 = mySerial.read();
        if(mydata2 == 'i'){
          lcd1.clear();
          lcd1.setCursor(5, 0);
          lcd1.print("LOT #1 ");
          lcd1.setCursor(6, 1); }
        else
          { lcd1.print(mydata2,byte(0));
          }
      }

```

```

}

```

```

}

```

```
if(mydata == 'j'){

```

```
  lcd2.clear();

```

```
  lcd2.setCursor(5, 0);

```

```
  lcd2.print("LOT #2 ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lcd2.setCursor(6, 1);
Serial.println("getb");
delay(10000);
while ( mySerial.available()>0 ) {
    mydata2 = mySerial.read();
    if(mydata2 == 'j'){
        lcd2.clear();
        lcd2.setCursor(5, 0);
        lcd2.print("LOT #2 ");
        lcd2.setCursor(6, 1); }
    else
    { lcd2.print(mydata2,byte(0));
}
}
}
}
if(mydata == 'k'){
    lcd3.clear();
    lcd3.setCursor(5, 0);
    lcd3.print("LOT #3 ");
    lcd3.setCursor(6, 1);
    Serial.println("getc");
    delay(10000);
    while ( mySerial.available()>0 ) {
        mydata2 = mySerial.read();
        if(mydata2 == 'k'){
            lcd3.clear();
            lcd3.setCursor(5, 0);
            lcd3.print("LOT #3 ");
            lcd3.setCursor(6, 1); }
        else
        { lcd3.print(mydata2,byte(0));
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแบบลงมือทำ และต้องขออนุญาตก่อนนำเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

if(mydata == 'm'){
  lcd4.clear();
  lcd4.setCursor(5, 0);
  lcd4.print("LOT #4 ");
  lcd4.setCursor(6, 1);
  Serial.println("getd");
  delay(10000);
  while ( mySerial.available()>0 ) {
    mydata2 = mySerial.read();
    if(mydata2 == 'm'){
      lcd4.clear();
      lcd4.setCursor(5, 0);
      lcd4.print("LOT #4 ");
      lcd4.setCursor(6, 1); }
    else
    { lcd4.print(mydata2,byte(0));
    }
  }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.4 โปรแกรมควบคุมเซอร์โว

```
#include <Servo.h>
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
char datas[5];
int i;
```

```
void setup()
```

```
{
  Serial.begin(9600);
  myservo1.attach(6);
  myservo1.write(85);
  myservo2.attach(9);
  myservo2.write(90);
  myservo3.attach(10);
  myservo3.write(90);
  myservo4.attach(11);
  myservo4.write(100);
}
```

```
void loop()
```

```
{
  if(Serial.available())
  {
    for (i=0; i <4; i++)
    {
      datas[i] = Serial.read();
    }
  }
  datas[4] = '\0';
```

```
if (strcmp(datas, "8976") == 0)
```

```
myservo1.write(0);
```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if (strcmp(datas, "CLS1") == 0)
{
myservo1.write(85);
}
else if (strcmp(datas, "74KO") == 0)
{
myservo2.write(0);
}
else if (strcmp(datas, "CLS2") == 0)
{
myservo2.write(90);
}
else if (strcmp(datas, "98ED") == 0)
{
myservo3.write(0);
}
else if (strcmp(datas, "CLS3") == 0)
{
myservo3.write(90);
}
else if (strcmp(datas, "65FR") == 0)
{
myservo4.write(10);
}
else if (strcmp(datas, "CLS4") == 0)
{
myservo4.write(100);
}
}
delay(200);


```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

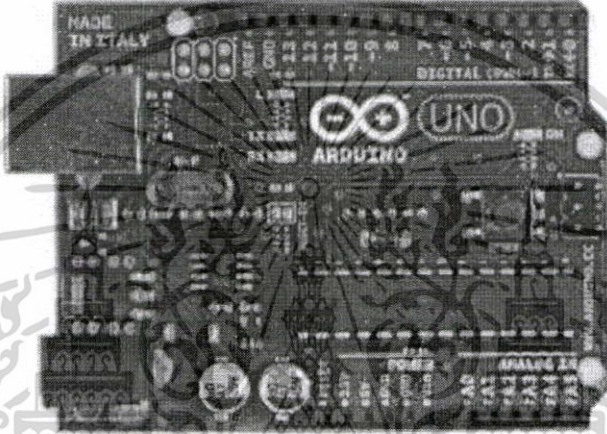
ภาคผนวก ข


เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 เอกสารคู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3



Arduino UNO





Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 8 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button: it contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications	Page 2
How to use Arduino Programming Environment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Environmental Policies half sqm of green via Impatto Zero®	Page 7

Technical Specification

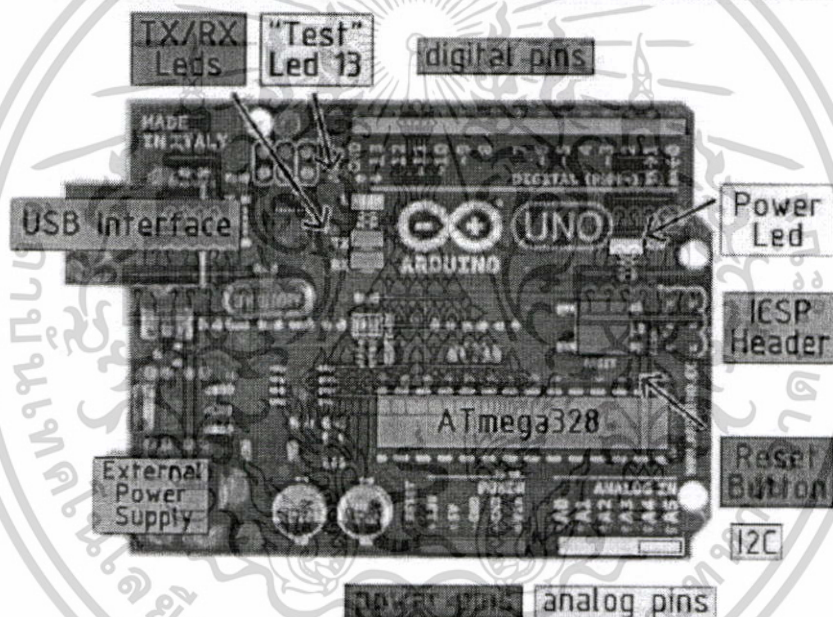


EAGLE file: [arduino-uno-rev103a-uno-testy.via](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

เอกสารนี้เป็นเอกสาร
ไม่ว่ากรณีใดๆทั้งสิ้น

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C:** 4 (SDA) and 5 (SCL). Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I²C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I²C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

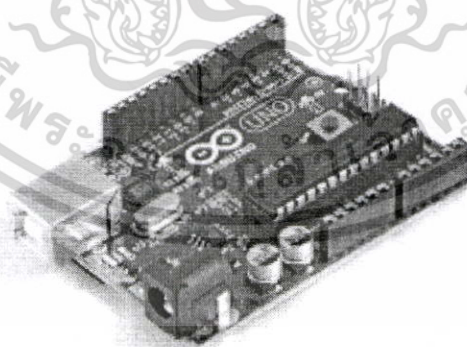
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 100 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

เอกสารนี้เป็น
ไม่ว่ากรณีใด

ชนด้านการค้า
นำไปใช้

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](http://arduino.cc/en/Guide/HomePage) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

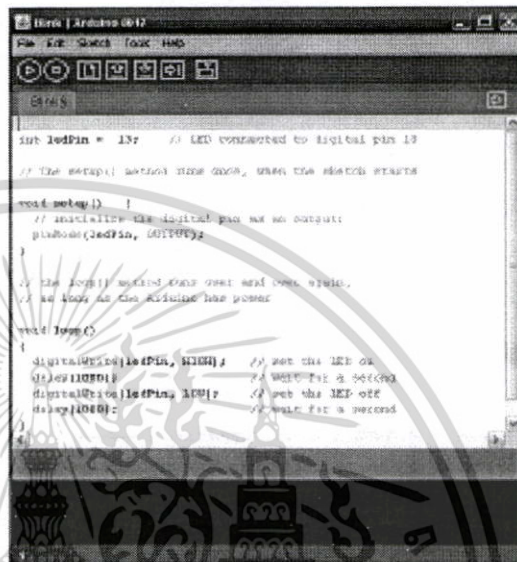
Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink

Once you have your sketch you'll see something very close to the screenshot on the right.

In Tools>Board select

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



```

Sketch | Arduino 0017
File Edit Sketch Tools Help

Serial

// Blink sketch
int ledPin = 13; // LED connected to digital pin 13

// The setup() section runs once, when the sketch starts
void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

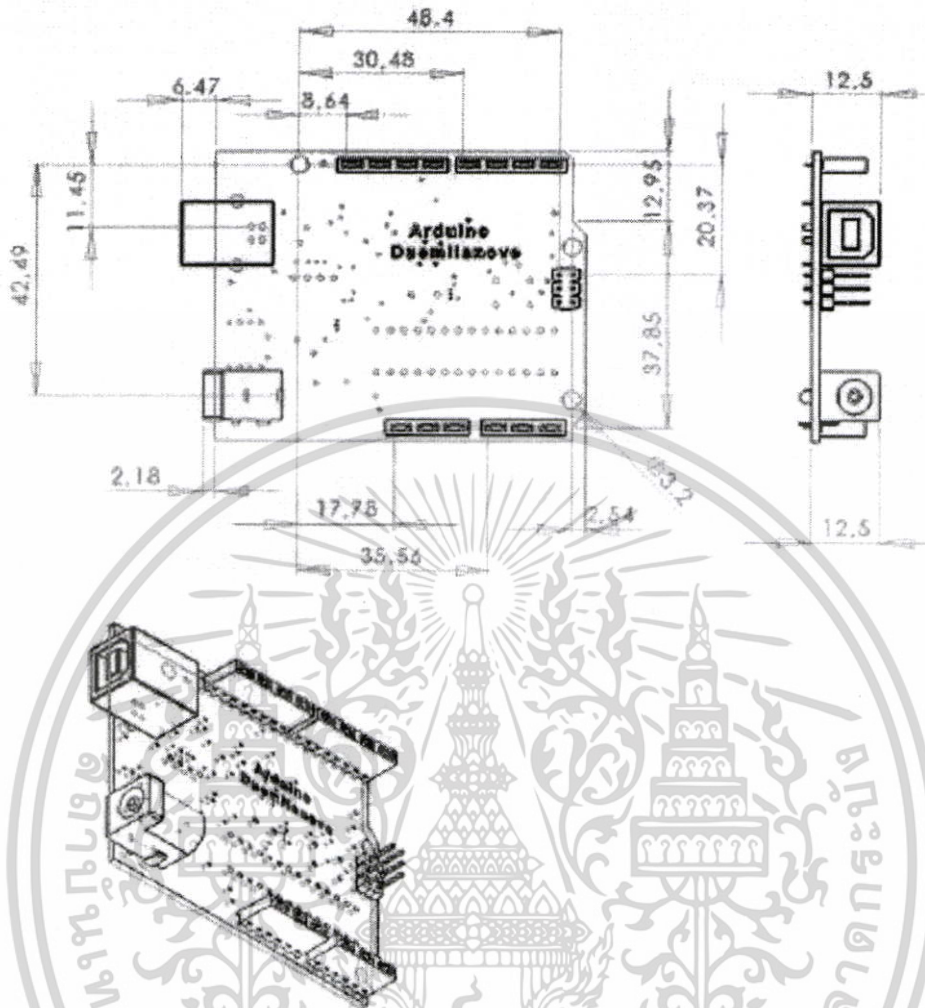
// The loop() section runs over and over again,
// as long as the Arduino has power
void loop() {
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
  
```



Dimensioned Drawing



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Terms & Conditions



1. Warranty

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

เอกสารนี้เป็นเอกสารของบริษัทผู้ผลิตและจำหน่ายสินค้าอิเล็กทรอนิกส์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

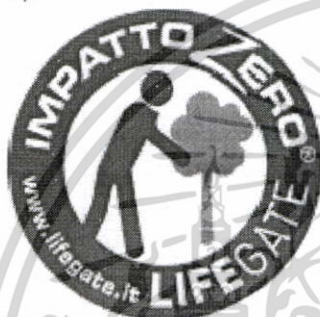
In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



RS

radiospares

RADIONICS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 เอกสารคู่มือการใช้งานอัลตราโซนิกเซนเซอร์ HC-SR04



Tech Support: services@elecfreaks.com

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal.
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2.

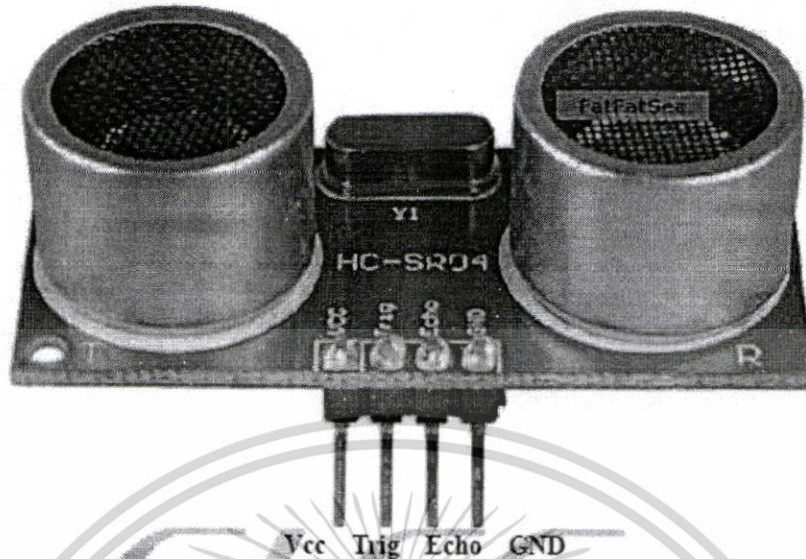
Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

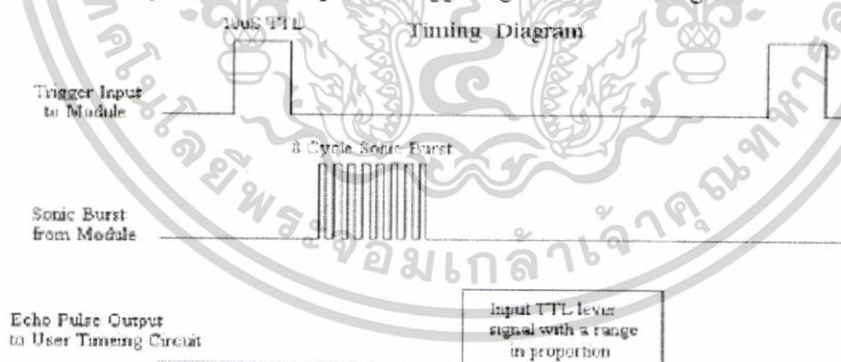
Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Timing diagram

The Timing diagram is shown below. You only need to supply a short 10 μ S pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that pulse width and the range in proportion. You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: $\mu\text{S} / 58 = \text{centimeters}$ or $\mu\text{S} / 148 = \text{inch}$; or the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Attention:

- The module is not suggested to connect directly to electric, if connected electric, the GND terminal should be connected the module first, otherwise, it will affect the normal work of the module.
- When tested objects, the range of area is not less than 0.5 square meters and the plane requests as smooth as possible, otherwise, it will affect the results of measuring.