ป้ายไฟแสดงประสาน
SYNCHRONIZE ELECTRONIC LABEL

ภวิษย์ภร ภูพันธุ์อนันต์
PAWITPORN POOPAN-ANANT
รวิพร แดงชาติ
RAWIPORN DAENGCHAT
รัชนีกร ธรรมนิยม
RATCHANEEKORN THAMNIYOM

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ป้ายไฟสอดประสาน
SYNCHRONIZE ELECTRONIC LABEL

ภวิษย์พร ฟูพันธุ์อนันต์
PAWITPORN FOOPAN-ANANT

รวิพร แดงชาติ
RAWIPORN DAENGCHAT

รัชนีกร ธรรมนิยม
RATCHANEEKORN THAMNIYOM

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

SYNCHRONIZE ELECTRONIC LABEL

PAWITPORN FOOPAN-ANANT

RAWIPORN DAENGCHAT

RATCHANEEKORN THAMNIYOM

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

หัวข้อปริญญานิพนธ์	ป้ายไฟสอดประสาน Synchronize Electronic Label	
รายชื่อนักศึกษา	นางสาวภวิษย์พร ฟูพันธุ์อนันต์	รหัสนักศึกษา 53011200
	นางสาววิพร แดงชาติ	รหัสนักศึกษา 53011335
	นางสาวรัชนีกร ธรรมนิยม	รหัสนักศึกษา 53011346
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2556	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.กฤดากร กล่อมการ	

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

กฤดากร

.....

(ผศ.ดร.กฤดากร กล่อมการ)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	ป้ายไฟสอดประสาน Synchronize Electronic Label	
รายชื่อนักศึกษา	นางสาวภวิษย์พร ฟูพันธุ์อนันต์	รหัสนักศึกษา 53011200
	นางสาวรวิพร แดงชาติ	รหัสนักศึกษา 53011335
	นางสาวรัชนีกร ธรรมนิยม	รหัสนักศึกษา 53011346
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2556	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.กฤดากร กล่อมการ	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาเกี่ยวกับการแสดงผลตัวอักษรและตัวเลข ผ่านทางบอร์ดแสดงผลไดโอดเปล่งแสงซึ่งได้ถูกพัฒนาให้มีความสามารถในการทำงานติดต่อสอดประสานรับส่งข้อมูลกันได้ โดยรูปแบบของการแสดงข้อมูลจะแสดงผลลงบนบอร์ดแสดงผลแบบจุด (Dot Matrix Display Board) โดยใช้ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3 และ Leonardo Pro Micro ATmega32U4 Arduino เป็นตัวควบคุมและประมวลผลข้อมูลที่ต้องการแสดงผลโดยตัวป้ายไฟสอดประสานจะเรียนรู้ตำแหน่งของตนเอง และส่งข้อมูลตำแหน่งไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผลและส่งไปยังตัวส่งข้อมูล เพื่อบอกตำแหน่งของตนไปให้ตัวควบคุมรับทราบและทำการส่งข้อความที่ต้องการแสดงกลับมา และไมโครคอนโทรลเลอร์จะทำการแปลงรหัสที่ได้รับจากอินฟราเรดนำมาประมวลผลคำสั่ง เพื่อสร้างเป็นตัวอักษรและตัวเลขตามแบบของรหัสข้อความที่ได้รับมาแล้วนำข้อความหรือตัวอักษรที่ได้ไปแสดงผลที่บอร์ดไดโอดเปล่งแสง ซึ่งในโครงงานนี้มีส่วนประกอบที่สำคัญได้แก่ แสดงผล (Display Board) , ส่วนตัวขับ (Driver Board) , ส่วนควบคุมและประมวลผล (Microcontroller) และ ส่วนส่งข้อมูลผ่านอินฟราเรด (Infrared)

Thesis Title	SYNCHRONIZE ELECTRONIC LABEL	
Student	Miss Pavitporn Foopan-anant	Student ID. 53011200
	Miss Rawiporn Daengchat	Student ID. 53011335
	Miss Ratchaneekorn Thamniyom	Student ID. 53011346
Degree	Bachelor of Engineering	
Program	Information Engineering	
Year	2013	
Thesis Advisor	Asst.Prof. Kitdakorn Klomkarn	

ABSTRACT

This project is the study of displaying of letters and numbers through beamed diodes displayed boards. The boards have been developed to be compatible so it can synchronize, receive, and transfer the data. The format of displaying of the data will be performed on the displayed board as dots (Dot Matrix Display Board) by using microcontroller Arduino Mega 2560 R3 and Leonardo Pro Micro ATmega32U4 Arduino as controller and processor of the needed data to show on the boards.

The synchronized displayed boards will learn their own positions and will send the information of their positions to the microcontroller to process and send to the data transmitter via radio waves to indicate their positions to the controller and send back the needed data to display. After that microcontroller will convert the code that received from the infrared to process for forming into letters and numbers as the form of received code. Finally, the message or letters will display on beamed diodes displayed boards. In addition, the important compositions in this project are the display (Display Board), the driver (Driver Board), the controller and processor (Microcontroller), the data transmitter via infrared (Infrared).

กิตติกรรมประกาศ

การออกแบบและการสร้างโครงงานบอร์ดแสดงผลโดย ไดโอดเปล่งแสงแบบเมทริกซ์ 8 คูณ 8 นี้ประกอบไปด้วยตัวชิ้นงานและเอกสารประกอบโครงงาน ซึ่งโครงงานจะไม่สำเร็จลุล่วงไปด้วยดี หากขาดอาจารย์ ผศ.ดร. กฤดากร กล่อมการ อาจารย์ที่ปรึกษา ซึ่งให้คำแนะนำชี้แนะ ดูแลช่วยเหลือในด้านอุปกรณ์และให้คำปรึกษามาโดยตลอด รวมถึงการสนับสนุนด้านทุนทรัพย์จากบิดา มารดา ตลอดจนกำลังใจและความช่วยเหลือจากเพื่อนร่วมภาคผู้ซึ่งให้คำปรึกษา ชี้แนะ และคอยเป็นกำลังใจให้เสมอมา

จึงใคร่ขอขอบพระคุณทุกท่านดังกล่าวข้างต้นเป็นอย่างสูง ที่ให้การสนับสนุนและความอนุเคราะห์ การช่วยเหลือทุกๆ ด้าน จึงทำให้การออกแบบและการสร้างโครงงานบอร์ดแสดงผลโดยไดโอดเปล่งแสงแบบเมทริกซ์ 8 คูณ 8 และโปรแกรมควบคุมข้อมูลการทำงานของบอร์ดแสดงผลไดโอดเปล่งแสงนี้และสำเร็จลุล่วงไปด้วยดี

นางสาวภวิษย์พร ฟูพันธุ์อนันต์
นางสาววริพร แดงชาติ
นางสาวรัชนีกร ธรรมนิยม

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญรูป VIII
บทที่ 1 บทนำ 1
1.1 แนวคิดและที่มาของปัญหา 1
1.2 จุดประสงค์ 2
1.3 ขอบเขตของโครงงาน 2
1.4 ผลที่คาดว่าจะได้รับ 3
1.5 อุปกรณ์ที่ต้องใช้ 3
1.5.1 ฮาร์ดแวร์ 3
1.5.2 ซอร์ฟแวร์ 3
1.6 ขั้นตอนการดำเนินงาน 3
บทที่ 2 ทฤษฎีพื้นฐาน 5
2.1 ไดโอดเปล่งแสง (Light Emitting Diode : LED) 5
2.1.1 คุณสมบัติของไดโอดเปล่งแสง 6
2.2 เรจิสเตอร์แบบอนุกรม (ไอซีเบอร์ 74HC595) 7
2.3 ไอซีขับกระแส (ไอซี เบอร์ ULN2803) 9
2.4 ทรานซิสเตอร์ (TRANSISTORS) 9
2.4.1 ทรานซิสเตอร์แบบพีเอ็นพี (pnp Transistor) 9
2.4.2 ทรานซิสเตอร์แบบเอ็นพีเอ็น (npn Transistor) 10
2.5 อินฟราเรดรีโมตคอนโทรล (Infrared Remote Control : IR) 11
2.6 ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3 12
2.6.1 คุณสมบัติ 13
2.7 ไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega32U4 Arduino 18
2.7.1 คุณสมบัติ 19
2.8 มอดูลส่งแสงอินฟาเรด 20
2.9 มอดูลรับแสงอินฟราเรด (TSOP 4838) 20

สารบัญ (ต่อ)

หน้า

2.9.1 การทำงานของมอดูลรับแสงอินฟราเรด 38 กิโลเฮิร์ท 20
2.10 แสงใต้แดง(infrared) 21
2.11 การกล้ำสัญญาณมอดูเลตสัญญาณ 22
2.11.1 ประเภทการกล้ำสัญญาณ 23
2.11.2 การกล้ำสัญญาณทางแอมพลิจูด (AM) 23
2.11.3 การกล้ำสัญญาณทางความถี่ (FM) 24
2.11.4 การกล้ำสัญญาณทางเฟส (PM) 24
2.12 ตัวต้านทาน หรือ เรซิสเตอร์ (Resistor) 25
2.12.1 ตัวต้านทานแบบมีค่าคงที่ 25
2.12.2 ตัวต้านทานแบบปรับค่าได้ 26
2.12.3 การอ่านค่าความต้านทาน 26
2.12.4 ค่าที่พึงประสงค์ 28
2.12.5 ตัวต้านทานแบบเอสเอ็มที (SMT) 29
2.12.6 กฎของโอห์ม 29
บทที่ 3 การวิเคราะห์และออกแบบระบบ 31
3.1 การออกแบบวงจร 31
3.1.1 การออกแบบ 31
3.1.2 หลักการทำงาน 33
3.1.3 การออกแบบวงจรบอร์ดแสดงผลแบบ 8x8 ด๊อทเมทริกซ์ (LED DOT MATRIX) 33
3.1.4 การออกแบบแผนภาพเค้าร่าง (Schematic) ของวงจร 33
3.1.5 การออกแบบวงจรพีซีบี (PCB) ของวงจร 34
3.2 การควบคุมการทำงานของอินฟราเรด 36
3.2.1 หลักการทำงานของอินฟราเรดของตัวแม่ 36
3.2.2 หลักการทำงานของอินฟราเรดของตัวลูก 36
3.2.3 ขั้นตอนวิธีของโปรแกรมอินฟราเรดตัวลูก 37
3.2.4 หลักการกล้ำสัญญาณ 38
3.2.5 หลักการดีมอดูเลตสัญญาณ 38
3.3 การควบคุมการแสดงผลหลอดไดโอดเปล่งแสง (LED) 40
บทที่ 4 ผลการทดลอง 41
4.1 โปรแกรมสำหรับเรียนรู้ตำแหน่งของตนเองโดยอินฟราเรด 41

สารบัญ (ต่อ)

หน้า

4.1.1 สำหรับวงจรที่ใช้ทดลอง 41
4.1.2 สำหรับบอกตำแหน่งป้ายชื่อ 0xA0 42
4.1.3 สำหรับบอกตำแหน่งป้ายชื่อ 0xB0 45
4.1.4 สำหรับบอกตำแหน่งป้ายชื่อ 0xC0 49
4.2 ส่วนแสดงผลบนหลอดไดโอดเปล่งแสง (LED) 52
4.2.1 ส่วนของโปรแกรมที่ใช้แสดงผล 52
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง 65
5.1 รูปแบบวงจรด๊อทเมทริกซ์ที่สร้างได้จริง 65
5.2 รูปแบบการแสดงผลของด๊อทเมทริกซ์ 68
5.2.1 รูปส่วนแสดงผลบอร์ดที่ยังไม่ทำการสอดประสานกัน 68
5.2.2 รูปส่วนแสดงผลบอร์ดที่ทำการสอดประสานกัน 69
5.3 วิจารณ์ปัญหาที่พบจากการทดลอง 70
5.3.1 ปัญหาจากวงจรอินฟราเรด 70
5.3.2 ปัญหาจากวงจรหลอดไดโอดเปล่งแสง 70
บรรณานุกรม 71

สารบัญตาราง

ตารางที่ หน้า
1.1 แสดงระยะเวลาในการดำเนินงาน 3
2.1 ตารางขาการส่ง Arduino Mega 2560 14
2.2 ตารางอ่านค่าตัวต้านทาน 27

สารบัญรูป

รูปที่ หน้า

1.1 การทำงานของระบบป้ายไฟสอดประสาน 2
2.1 แสดงลักษณะของไดโอดเปล่งแสง 5
2.2 รูปไดโอดเปล่งแสงเมทริกซ์ขนาด 8x8 6
2.3 ลักษณะการทำงาน และการต่อภายในของเมทริกซ์ขนาด 8X8 7
2.4 ตำแหน่งขา ไอซีเบอร์ 74HC595 8
2.5 แผนภาพเวลา ของ ไอซี 74HC595 8
2.6 ตำแหน่งขา ไอซี ULN2803 9
2.7 ทรานซิสเตอร์แบบพีเอ็นพี 10
2.8 ทรานซิสเตอร์แบบเอ็นพีเอ็น 10
2.9 การดูขอบขาของสัญญาณว่าเป็นขึ้นหรือลง 11
2.10 ดูระยะห่างของสัญญาณเพ้าท์ 11
2.11 ดูความกว้างของสัญญาณเพ้าท์ 11
2.12 ตัวส่งและตัวรับ อินฟราเรดรีโมตคอนโทรล (Infrared Remote Control : IR) 12
2.13 ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3 13
2.14 แผนภาพขา Arduino Mega 2560 14
2.15 ไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega32U4 19
2.16 แผนภาพขา Leonardo Pro Micro ATmega32U4 19
2.17 หลอดไดโอดเปล่งแสง ตัวส่งอินฟราเรด 20
2.18 มอดูลรับแสงอินฟราเรด 38 กิโลเฮิร์ท (TSOP 4838) 21
2.19 แถบเส้นสเปคตรัม 21
2.20 แถบเส้นเสปคตรัมที่ถูกนำไปใช้งาน 22
2.21 สัญญาณคลื่นพาห์ สัญญาณข้อมูล สัญญาณกล้าสัญญาณแอมพลิจูด 23
2.22 สัญญาณคลื่นพาห์ สัญญาณข้อมูล สัญญาณกล้าสัญญาณความถี่ 24
2.23 สัญญาณคลื่นพาห์ สัญญาณข้อมูล สัญญาณกล้าสัญญาณทางเฟส 25
2.24 รูปวิธีอ่านค่าตัวต้านทาน 28
2.25 รูปการต่อตัวต้านทานแบบขนาน 29
2.26 รูปการต่อตัวต้านทานแบบอนุกรม 30
2.27 รูปการต่อตัวต้านทานแบบขนานและอนุกรม 30
3.1 แบบวงจรเมทริกซ์ 8 X 8 จำนวน 1 วงจร 31
3.2 แบบวงจรเมทริกซ์ 8 X 8 แบบ 1 ชิ้น 32

สารบัญรูป(ต่อ)

รูปที่ หน้า

3.3 แบบวงจรเมทริกซ์ 8X8 แบบ 1 ชิ้น 32
3.4 วงจร หลอดไดโอดเปล่งแสง ด๊อท เมทริกซ์ 8X8 34
3.5 วงจรแผนภาพเค้าร่างของวงจรแอลอีดี ด๊อทเมทริกซ์ ชั้นแรก (LED DOT MATRIX) 35
3.6 วงจรภาพพีซีบี(PCB) ของวงจรแอลอีดี ด๊อทเมทริกซ์ (LED DOT MATRIX) 35
3.7 แผนภาพการทำงานของอินฟราเรด 36
3.8 แผนภาพระบบการส่งสัญญาณระยะไกล 38
3.9 แผนภาพการส่งสัญญาณขาออก 38
3.10 แผนภาพการส่งสัญญาณขาเข้า 39
3.11 แผนภาพการสัญญาณข้อมูลของอินฟราเรด 39
4.1 รูปวงจรการทำงานของอินฟราเรด 41
4.2 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 1 42
4.3 รูปส่วนแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 1 43
4.4 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 2 43
4.5 รูปส่วนแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 2 44
4.6 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 3 44
4.7 รูปส่วนแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 3 45
4.8 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 1 46
4.9 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 1 46
4.10 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 2 47
4.11 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 2 47
4.12 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 3 48
4.13 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 3 48
4.14 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 1 49
4.15 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 1 49
4.16 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 2 50
4.17 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 2 50
4.18 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 3 51
4.19 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 3 51
5.1 รูปวงจรการต่อจริง 65
5.2 รูปวงจรแผ่นลายปริ้นทั้งหมด 66

สารบัญรูป(ต่อ)

รูปที่ หน้า

5.3 รูปวงจรแผ่นลายปริ้นชั้นที่ 1 66
5.4 รูปวงจรแผ่นลายปริ้นชั้นที่ 2 67
5.5 รูปวงจรแผ่นลายปริ้นที่ใส่อุปกรณ์ 67
5.6 รูปบอร์ดแสดงผลปกติ 68
5.7 รูปบอร์ดแสดงผลการสอดประสาน 2 บอร์ด 69
5.8 รูปบอร์ดแสดงผลการสอดประสาน 3 บอร์ด 69

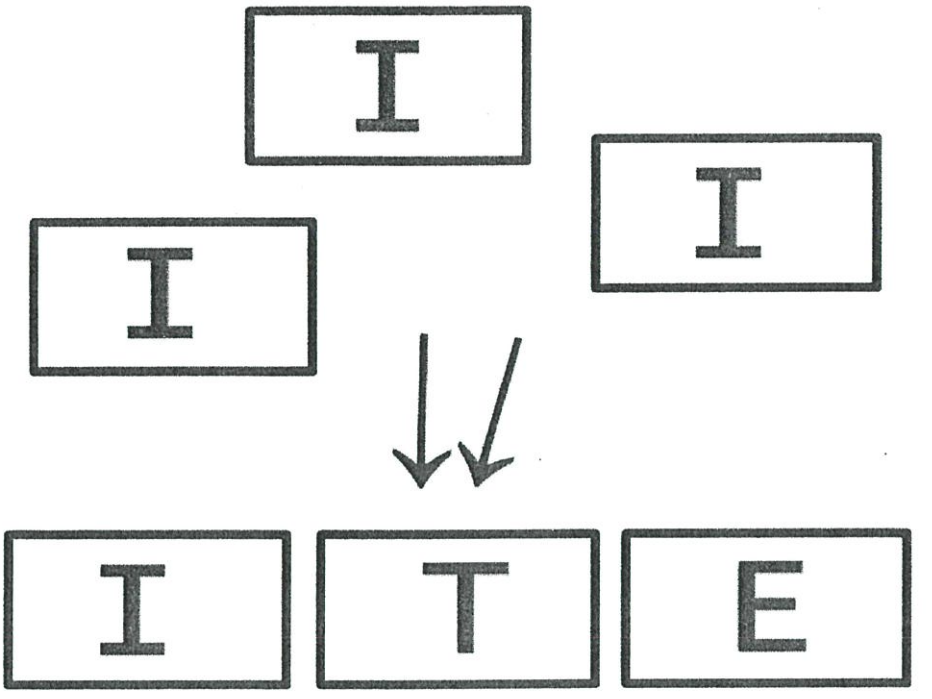
บทที่ 1
บทนำ

1.1 แนวคิดและที่มาของปัญหา

เนื่องจากปัจจุบันสื่อบันเทิงต่างประเทศได้เข้ามามีบทบาทกับชีวิตของวัยรุ่นไทยค่อนข้างมาก ทำให้สื่อบันเทิงต่างๆ ศิลปินนักร้องทั้งไทยและต่างประเทศต่างก็เป็นที่ได้รับความนิยมอย่างสูงในหมู่กลุ่มวัยรุ่น หรือที่เรียกว่าแฟนคลับ ทำให้อุปกรณ์จำพวกป้ายไฟต่างก็กำลังได้รับความนิยมและสนใจกันอย่างแพร่หลายในหมู่ของกลุ่มผู้คนเหล่านี้

โดยปัญหาที่พบในปัจจุบันของเหล่าแฟนคลับเองที่กำลังนิยมป้ายไฟจำพวกนี้คือขนาดของป้ายไฟที่มักมีขนาดใหญ่เทอะทะทำให้ยากลำบากในการพกพา หรือนำเข้าบริเวณการแสดงดนตรี และการทำงานของป้ายไฟธรรมดาที่มีการใช้งานได้น้อย เพียงแค่เปิดและปิดป้ายไฟเพียงเท่านั่น ทำให้ขาดความพร้อมเพรียงในการเปิดปิดให้พร้อมเพรียงกันสำหรับจำนวนแฟนคลับที่มีจำนวนมาก หรือจำนวนแฟนคลับที่มีเป็นกลุ่มใหญ่ที่ตั้งใจจะนัดแนะกัน ทำให้ผู้จัดทำโครงงานคิดจัดทำป้ายไฟสอดคล้องกันที่ตอบสนองกับความต้องการของแฟนคลับกลุ่มใหญ่ที่สามารถใช้ป้ายไฟเปิดปิดให้เป็นจังหวะสอดประสานกันได้อัตโนมัติ โดยป้ายไฟจะมีการประสานงานกันตามแต่จำนวนป้ายที่จัดทำขึ้น และเพราะป้ายไฟที่สามารถนำเข้าชมการแสดงนั่นจำเป็นต้องมีขนาดเล็กทำให้ป้ายไฟสอดประสานชิ้นนี้จึงสามารถที่จะต่อคำเป็นข้อความที่มีขนาดยาวเพิ่มขึ้นได้ด้วยป้ายไฟหลายชิ้น เหมาะสำหรับกลุ่มแฟนคลับที่มีจำนวนมาก สามารถประกอบกันเป็นข้อความขนาดใหญ่ที่เรียงต่อกันเป็นข้อความตามที่ต้องการได้

ดังนั้นทางกลุ่มผู้จัดทำโครงงานจึงจัดทำป้ายไฟสอดประสานเพื่อเป็นต้นแบบสำหรับผลิตในอุตสาหกรรมต่อไป



รูปที่ 1.1 การทำงานของระบบป้ายไฟสอดประสาน

1.2 จุดประสงค์

1. เพื่อพัฒนาให้การใช้งานป้ายไฟสามารถใช้ได้พร้อมเพรียงกันมากยิ่งขึ้นสำหรับคนกลุ่มใหญ่
2. เพื่อพัฒนาให้ป้ายไฟมีประสิทธิภาพและมีลูกเล่นเพิ่มมากยิ่งขึ้น
3. เพื่อพัฒนาให้ป้ายไฟสามารถแสดงข้อความที่มีความยาวเพิ่มมากขึ้นและมีความต่อเนื่องกันได้

1.3 ขอบเขตของโครงงาน

1. มีป้ายไฟที่มีอินฟราเรดสามารถเรียนรู้ตำแหน่งของตัวเองขณะนั่นได้
2. มีป้ายไฟที่สามารถรับส่งข้อมูลด้วยสัญญาณอินฟราเรดและสามารถแสดงข้อความที่ต้องการได้
3. มีป้ายไฟที่สามารถแสดงข้อความที่มีความยาวต่อเนื่องกันได้
4. มีป้ายไฟที่สามารถติดต่อกันเป็นแบบจุดต่อจุดได้

1.4 ผลที่คาดว่าจะได้รับ

1. ป้ายไฟที่พัฒนาขึ้นทำให้ง่ายต่อการใช้งานจริง
2. ป้ายไฟที่พัฒนาขึ้นสามารถนำไปต่อยอดทำการประดิษฐ์ให้มีประสิทธิภาพเพิ่มขึ้นได้ในอนาคต
3. ป้ายไฟที่พัฒนาขึ้นสามารถเป็นกรณีศึกษาให้กับผู้ที่สนใจได้
4. ป้ายไฟที่พัฒนาขึ้นสามารถเรียนรู้ตำแหน่งของตัวเองได้เองซึ่งง่ายต่อการแสดงข้อความ

1.5 อุปกรณ์ที่ต้องใช้

1.5.1 ฮาร์ดแวร์

1. ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3
2. ไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega32U4 Arduino
3. ป้ายไดโอดเปล่งแสง
3. ตัวรับสัญญาณอินฟราเรด
4. ตัวส่งสัญญาณอินฟราเรด
5. เครื่องคอมพิวเตอร์แบบพกพา

1.5.2 ซอร์ฟแวร์

1. ภาษาซี

1.6 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 แสดงระยะเวลาในการดำเนินงาน

ID	TASKNAME	2013							2014		
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1	เสนอหัวข้อโครงงาน										
2	ศึกษาค้นคว้าและทำความเข้าใจ										
3	วิเคราะห์และออกแบบองค์ประกอบ										

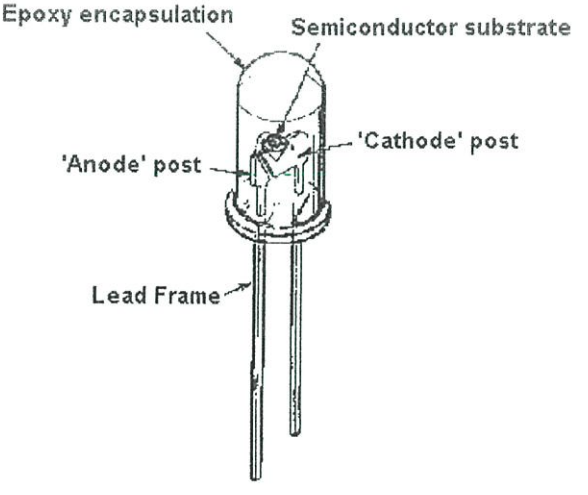
4	ทดลอง ฮาร์ดแวร์และซอร์ฟแวร์								
5	ทดลองการใช้งานและปรับปรุงแก้ไข								
6	จัดการปริญญานิพนธ์								

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้

โครงงานนี้ประกอบด้วยส่วนควบคุมการแสดงผลและส่วนแสดงผล โดยส่วนควบคุมการแสดงผลจะใช้ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3 และไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega 32U4 Arduino ตัวส่งและรับข้อมูลด้วยสัญญาณอินฟราเรดเป็นส่วนประกอบที่สำคัญ และในส่วนแสดงผลจะมี ไดโอดเปล่งแสง เป็นส่วนประกอบสำคัญ จึงต้องทำความเข้าใจเกี่ยวกับคุณสมบัติและการทำงานของทั้งสองส่วนนี้ให้เข้าใจ

2.1 ไดโอดเปล่งแสง (Light Emitting Diode : LED)

แอลอีดี (LED) ย่อมาจาก Light Emitting Diodes หลอดไดโอดเปล่งแสง คือ หลอดไฟขนาดเล็กแต่มีหลักการทำงานแตกต่างจากหลอดไฟมีไส้ เพราะไม่มีการเผาไส้หลอด ดังนั้นหลอดไดโอดเปล่งแสง จึงไม่เกิดความร้อน แสงเกิดขึ้นจากการเคลื่อนที่ของอิเล็กตรอนภายในสารกึ่งตัวนำ ซึ่งเป็นวัสดุแบบเดียวกับที่ใช้ในการทำทรานซิสเตอร์ ไดโอดเปล่งแสงเป็นสิ่งประดิษฐ์สารกึ่งตัวนำชนิดหนึ่งที่สามารถแปลงพลังงานไฟฟ้าเป็นพลังงานแสง การเปล่งแสงเกิดขึ้นเมื่อปลดปล่อยพลังงานออกมาเพื่อลดระดับลงไปอยู่ที่ระดับพลังงานต่ำกว่า อิเล็กตรอนที่กระโดดจากวงโคจรนอกสู่วงโคจรในมันจะปลดปล่อยพลังงานออกมา

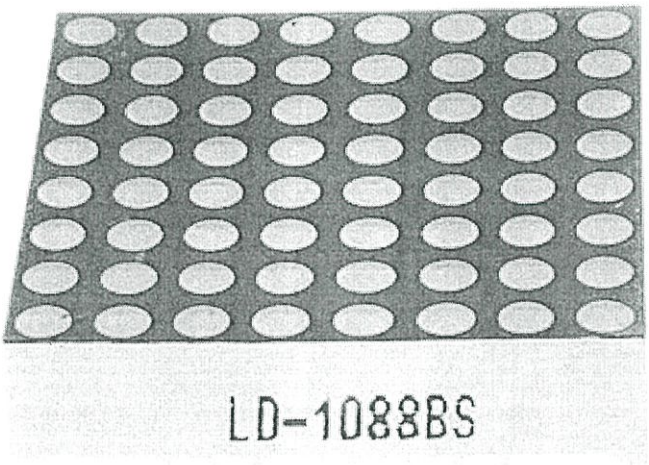


รูปที่ 2.1 แสดงลักษณะของไดโอดเปล่งแสง

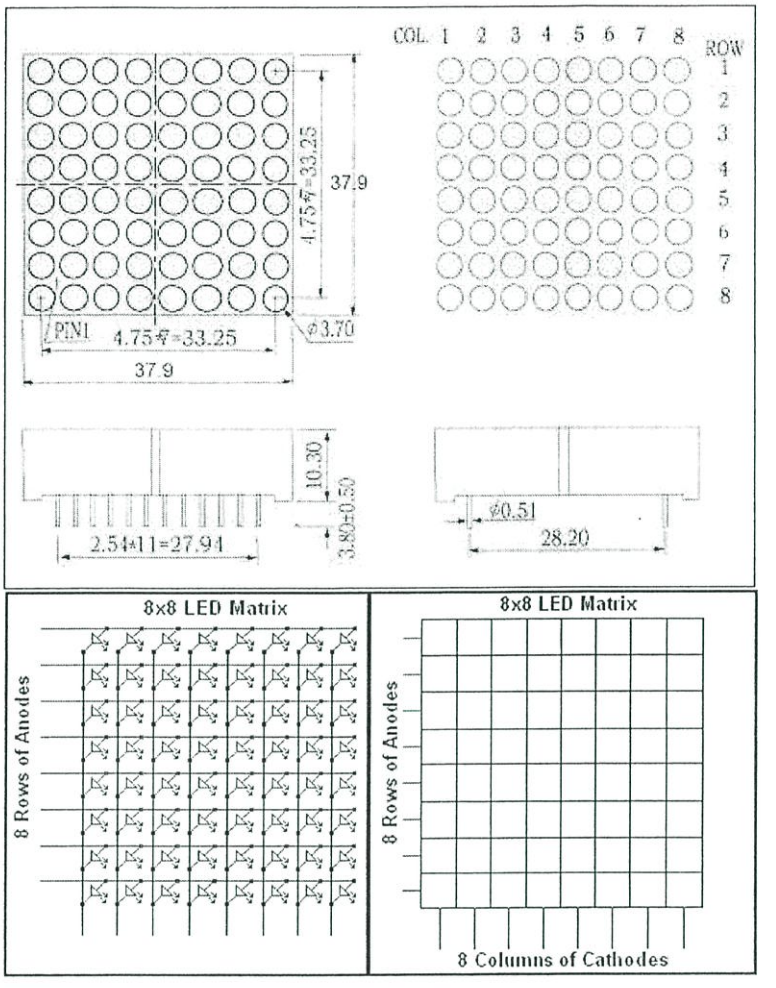
2.1.1 **คุณสมบัติของไดโอดเปล่งแสง คือ**

1. เป็นแหล่งกำเนิดแสงขนาดเล็กแต่จะมีความทนทานสูง
2. ใช้แรงดันที่มาตั้งในจุดทำงาน ไดโอดเปล่งแสงที่ต่ำ
3. สามารถนำมาใช้งานในการกล้ำสัญญาณความเร็วสูง

สำหรับในโครงงานนี้จะใช้ ไดโอดเปล่งแสง ชนิด ด็อทเมทริกซ์ขนาด 8 คูณ 8 ซึ่งเป็นอุปกรณ์ แสดงผลแบบเดียวกับไดโอดเปล่งแสง โดยการนำเอาไดโอดเปล่งแสงหลายตัวมาต่อเรียงกัน



รูปที่ 2.2 รูปไดโอดเปล่งแสงเมทริกซ์ขนาด 8x8



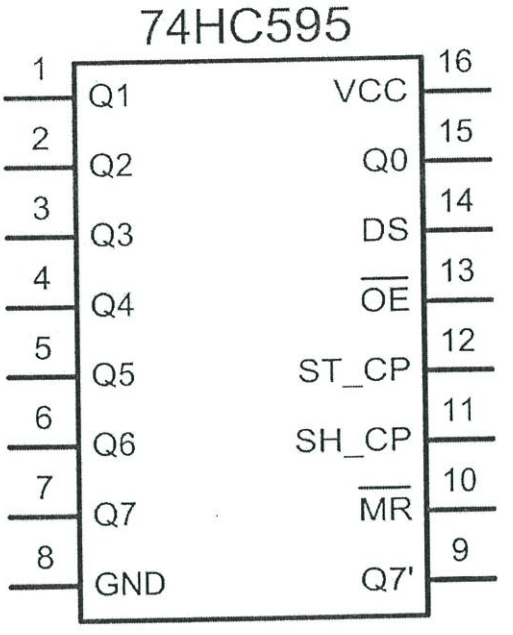
รูปที่ 2.3 ลักษณะการทำงาน และการต่อภายในของเมทริกซ์ขนาด 8X8

2.2 เรจิสเตอร์แบบอนุกรม (ไอซีเบอร์ 74HC595)

ใช้เรจิสเตอร์แบบอนุกรมในการขยายช่องทางเข้าออกโดยจะใช้ขาข้อมูลเพียง 3 ขาเท่านั้น ประกอบไปด้วย สัญญาณนาฬิกา , ข้อมูล , สโตรบ โดยไอซีแต่ละผู้ผลิตอาจจะตั้งชื่อขาสัญญาณแตกต่างกันออกไปบ้าง โดยข้อมูลจะส่งไปพร้อมกับสัญญาณนาฬิกา หรือที่เรียกการส่งข้อมูลแบบนี้ว่า การประสานเวลา

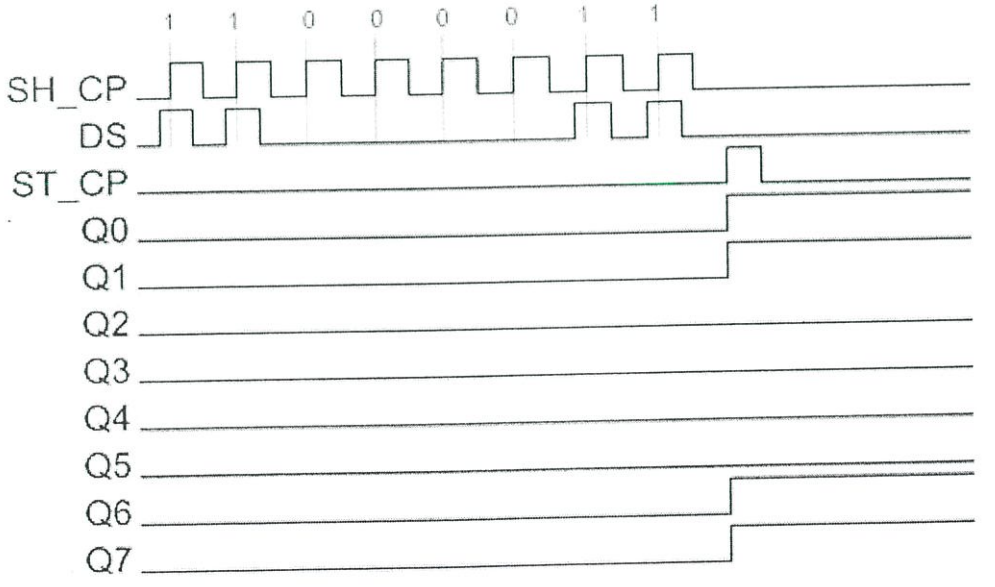
ไอซีเบอร์ 74HC595 จะใช้งานโดยการต่อใช้งานขาข้อมูลจากไมโครคอนโทรลเลอร์ จะต่อเข้ากับ ข้อมูลขาเข้า ของตัวที่ 1 และ ข้อมูลขาออก ของตัวที่ 1 จะต่อเข้ากับข้อมูลขาเข้าของตัวที่ 2 เป็นอย่างนี้ไปเรื่อยๆจนถึงตัวสุดท้าย ของข้อมูลขาออกของตัวสุดท้ายจะไม่ถูกต่อ ไอซีเบอร์ 74HC595 จะใช้การส่งข้อมูลแบบ ดี ฟลิปฟล๊อบ คือ มีสายสัญญาณ 2 เส้น ป้อนข้อมูล และสัญญาณนาฬิกา เส้นหนึ่งต่อกับขา 11 เพื่อเป็นสัญญาณนาฬิกา อีกเส้น ต่อขา 14 เป็นสายส่งข้อมูล

หลักการทำงานคือ เมื่อขา สัญญาณนาฬิกา ขา 11 ถูกกระตุ้น ตัวไอซีจะรับข้อมูลจากขาข้อมูล ขา14 ไปเก็บไว้ในเรจิสเตอร์ ซึ่งมี 8 บิตในไอซีตัวนี้ และเลื่อนข้อมูลที่อยู่ก่อนหน้าไป 1 ช่อง เป็นลำดับตลอดเวลา



รูปที่ 2.4 ตำแหน่งขา ไอซีเบอร์ 74HC595

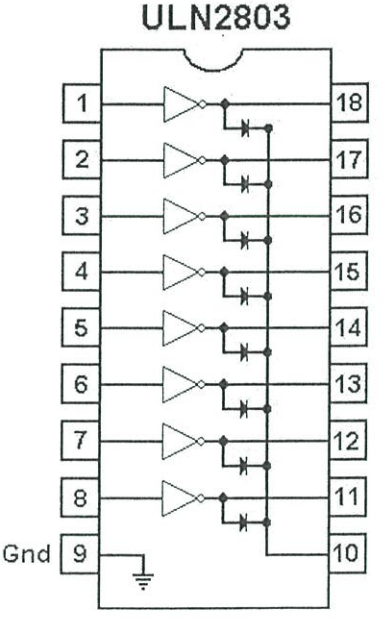
การเลื่อนข้อมูลนี้จะไม่ส่งผลต่อขาส่งออก ทั้ง 8 เพราะเป็นการนำข้อมูลไปเก็บเอาไว้ในเรจิสเตอร์เท่านั้น แต่จนถึงตอนนี้ข้อมูลส่งออกที่ต้องการก็ยังไม่ออกจากขา Q0 ถึง Q7 ต้องทำการกระตุ้นขา 12 ต่ออีกทีหนึ่งเป็นการบอกไอซี ว่าส่งข้อมูลครบเรียบร้อยแล้ว จะส่งสัญญาณพัลซ์ 1 ลูกไป จากนั้นข้อมูลจะถูกโหลด จากเรจิสเตอร์ ไปยังขา Q0 ถึง Q7 เพื่อนำไปใช้งานต่อไป



รูปที่ 2.5 แผนภาพเวลา ของ ไอซี 74HC595

2.3 ไอซีขับกระแส (ไอซี เบอร์ ULN2803)

ไอซีเบอร์ ULN2803 ซึ่งเป็นไอซีขับกระแสขนาด 8 บิต สามารถขับกระแสให้กับโหลดได้ถึง 500 มิลลิแอมแปร์ต่อ 50 โวลต์ มีโครงสร้างเป็นตัวผกผันแบบ เปิดคอลเลคเตอร์ (Open Collector) ทำหน้าที่เป็นตัวขับกระแส ทำการขับกระแสไฟฟ้าให้แต่ละแถวของไดโอดเปล่งแสง แบบเมทริกซ์ 8 คูณ 8 เพราะจะต้องใช้ไฟเลี้ยงจ่ายไปยังทุกตำแหน่งของไดโอดเปล่งแสง แบบเมทริกซ์ 8 คูณ 8 ซึ่งหากไม่มีตัวช่วยขับกระแสจะส่งผลให้ความสว่างของไดโอดเปล่งแสงลดลง หรือมีความสว่างไม่เท่ากัน หรืออาจจะไม่สว่างเลยก็ได้



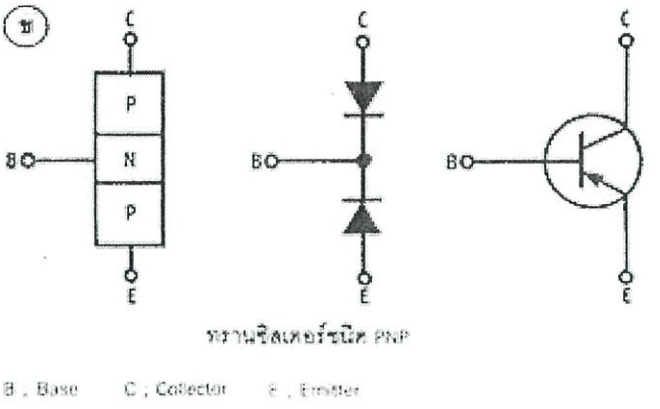
รูปที่ 2.6 ตำแหน่งขา ไอซีเบอร์ ULN280

2.4 ทรานซิสเตอร์ (TRANSISTORS)

ทรานซิสเตอร์เป็นอุปกรณ์อิเลคทรอนิกส์ซึ่งมีรอยต่อของสารกึ่งตัวนำ พีเอ็น จำนวน 2 ตำแหน่ง จึงมีชื่อเรียกอีกอย่างหนึ่งว่า "ทรานซิสเตอร์รอยต่อสองขั้ว"

2.4.1 ทรานซิสเตอร์แบบพีเอ็นพี (pnp Transistor)

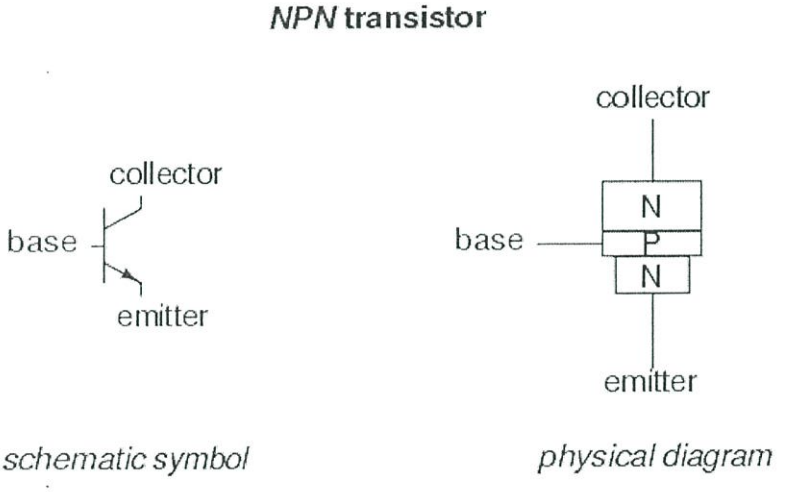
ทรานซิสเตอร์พีเอ็นพีประกอบด้วยสารกึ่งตัวนำชนิดพีจำนวน 2 ชิ้นต่อเชื่อมกับสารกึ่งตัวนำชนิดเอ็นจำนวน 1 ชิ้น



รูปที่ 2.7 ทรานซิสเตอร์แบบพีเอ็นพี

2.4.2 ทรานซิสเตอร์แบบเอ็นพีเอ็น (npn Transistor)

ทรานซิสเตอร์แบบ พีเอ็นพี ประกอบด้วยสารกึ่งตัวนำชนิด พี จำนวน 2 ชิ้นต่อเชื่อมกับสารกึ่งตัวนำชนิด เอ็น จำนวน 1 ชิ้น



รูปที่ 2.8 ทรานซิสเตอร์แบบเอ็นพีเอ็น

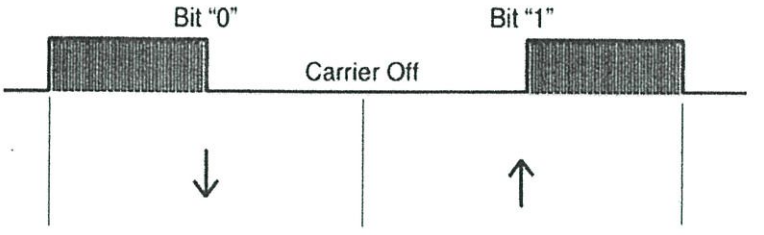
ทรานซิสเตอร์ มี 3 ขั้ว การต่อแรงดันไฟฟ้าที่ขั้วเพื่อให้ทรานซิสเตอร์ทำงานจึงเป็นไปได้ 3 แบบคือ

- การให้ทรานซิสเตอร์ทำงานที่บริเวณคัตออฟ (Cut-off Region)
- การให้ทรานซิสเตอร์ทำงานที่บริเวณอิ่มตัว (Saturation Region)
- การให้ทรานซิสเตอร์ทำงานที่บริเวณแอกทีฟ (Active Region)

2.5 อินฟราเรดรีโมตคอนโทรล (Infrared Remote Control : IR)

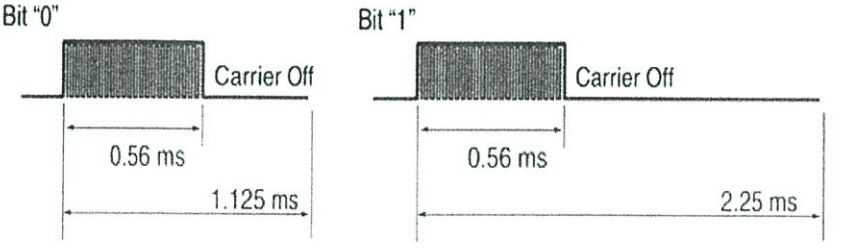
สัญญาณอินฟราเรดระยะไกลที่ใช้งานโดยทั่วไปจะส่งสัญญาณด้วยแสงอินฟราเรดออกมาในรูปแบบของข้อมูลเชิงเลขที่มีคลื่นพาหะขี่อยู่ซึ่งโดยทั่วไปจะมีค่าความถี่ 36 กิโลเฮิรตซ์ หรือ 38 กิโลเฮิรตซ์ ส่วนการแยกข้อมูล 1 และ 0 มีอยู่หลายรูปแบบซึ่งแบ่งออกเป็นได้ดังนี้

1. การดูขอบขาของสัญญาณว่าเป็นขึ้นหรือลงในเวลาที่กำหนด (บีไอ เฟส การเข้ารหัส)



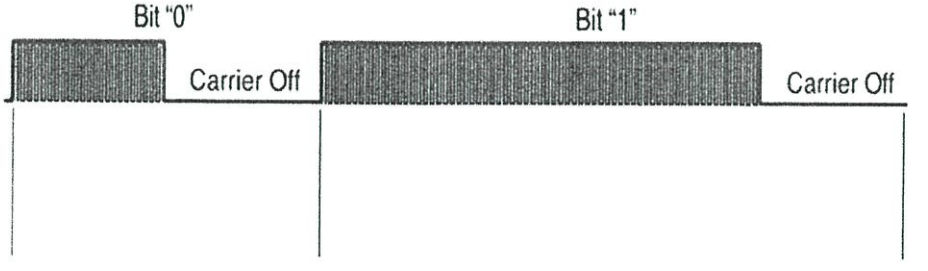
รูปที่ 2.9 การดูขอบขาของสัญญาณว่าเป็นขึ้นหรือลง

2. ดูระยะห่างของขอบสัญญาณเพ้าท์ (Pulse Distance)



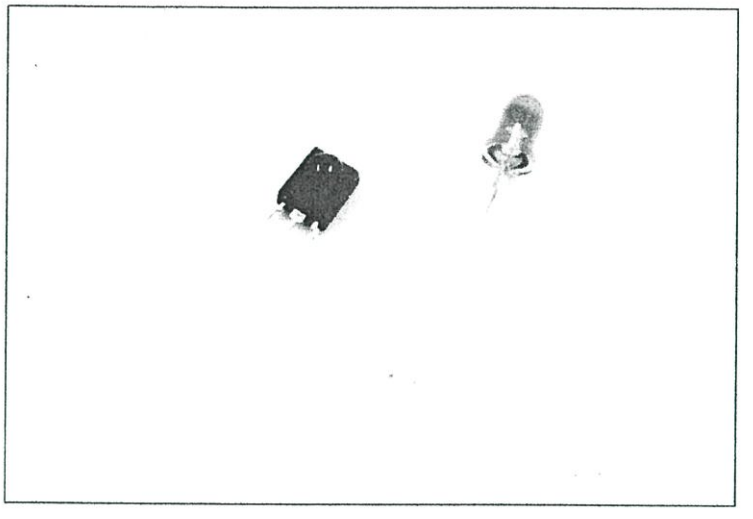
รูปที่ 2.10 ดูระยะห่างของของสัญญาณเพ้าท์

3. ดูจากความกว้างของสัญญาณเพ้าท์ (Pulse Length)



รูปที่ 2.11 ดูความกว้างของสัญญาณเพ้าท์

ไออาร์ตัวรับค่าจำเพาะ ที่ใช้ในการรับสัญญาณรีโมทจะมีหน้าที่ในการรับสัญญาณความถี่ของคลื่นพาหะที่ตรงกับค่านั่น ขนาดจำเพาะของตัวนั้นแล้วทำการกรองเอาคลื่นพาหะออก แล้วส่งสัญญาณออกมาเฉพาะส่วนที่เป็นข้อมูลเชิงเลข ดังนั้นในการจะเลือกให้จำเพาะจะต้องดูขนาด



รูปที่ 2.12 ตัวส่งและตัวรับ อินฟราเรดรีโมตคอนโทรล (Infrared Remote Control : IR)

2.6 ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3

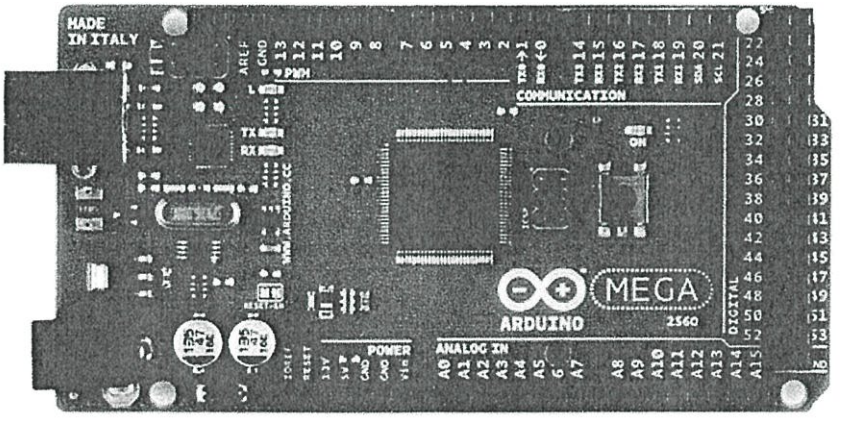
ไมโครคอนโทรลเลอร์ คือ อุปกรณ์ที่มีหน่วยประมวลผลและความจำขนาดเล็กภายในตัวเองสามารถรับและส่งข้อมูลได้ทั้งแบบดิจิตอลและอนาล็อก ใช้พลังงานน้อย ทำให้เป็นที่นิยมการใช้งานในรูปแบบที่เรียกว่าเอ็มเบ็ดเด็ด

อา-เดีย-โน (Arduino) เป็นชื่อโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์ แบบเปิดต้นทาง ที่ได้รับความนิยมจากผู้ใช้งานทั่วโลกเป็นมาก เพราะระบบวงจรของบอร์ดมีขนาดเล็กใช้อุปกรณ์น้อย ทำให้ง่ายต่อการต่อวงจรและประหยัดต้นทุนในการสร้างบอร์ดซึ่งเป็นไมโครคอนโทรลเลอร์แบบที่เรียกว่า การเปิดส่วนอุปกรณ์ เป็นอุปกรณ์ที่มีส่วนประกอบเป็นมาตรฐานที่เปิดเผย สามารถทำได้เองโดยใช้แบบที่มีการเปิดเผยทั่วไป สามารถหาซื้อง่าย มีราคาถูก มีชุดคำสั่งให้ใช้งานฟรี สามารถนำไปใช้งานทั่วไป หรือแบบธุรกิจได้โดยจะไม่เสียค่าลิขสิทธิ์ เป็นรูปแบบที่มีข้อมูลมากที่สุดบนอินเตอร์เน็ต การพัฒนาได้ง่าย สามารถใช้คำสั่งเขียนโปรแกรมได้เสมือนโปรแกรมชั้นสูงที่ใช้กัน นอกจากนี้ยังมีจุดเด่นในเรื่องของความง่ายในการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่างๆขึ้นมาสนับสนุนการ ด้วยรูปแบบที่ง่าย ไม่ซับซ้อน สามารถนำไปใช้งานได้จริง และยังสามารถสร้างคำสั่งและคลังใหม่ขึ้นมาใช้เองเมื่อมีความชำนาญมากขึ้น รองรับการทำงานทั้ง วินโดวส์ลีนุกซ์ และ มาร์ซินทอร์ช โอเอสเอ็กซ์

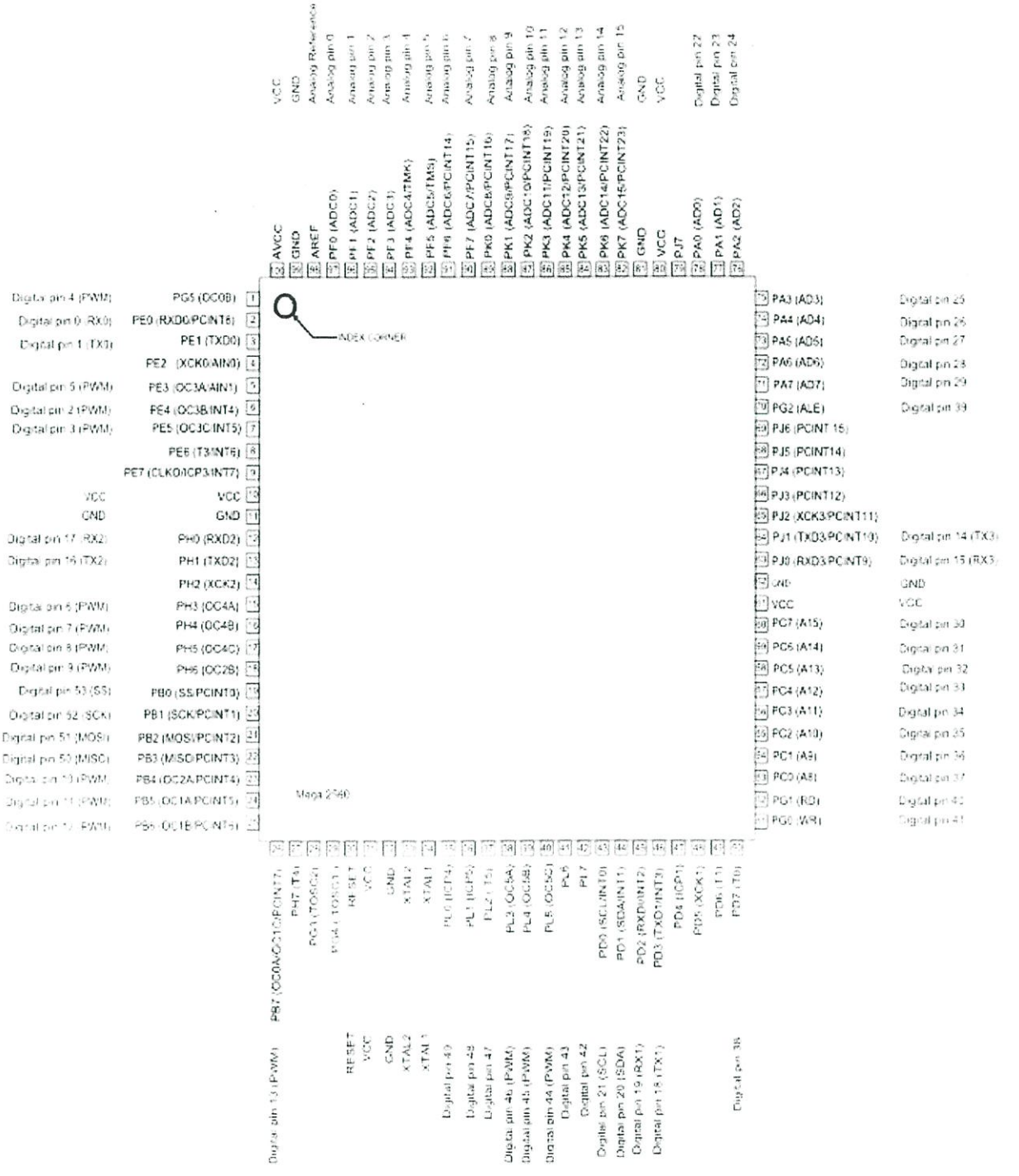
2.6.1 คุณสมบัติ

- ไมโครคอนโทรลเลอร์ ATMEGA2560
- แรงดันไฟฟ้า 5 โวลต์
- แรงดันไฟฟ้าขา 7 ถึง 12 โวลต์
- เชิงเลข ไอต่อโอ หมุด 54 (จาก15ซึ่งให้เอาท์พุทพีดับเบิ้ลยู เอ็ม)
- ใส่หมุดเชิงอุปมาน 16
- หน่วยความจำสะเก็ดวาบ 256 กิโลไบต์ที่ใช้โดยกิโลบูทโหลด
- ความเร็วสัญญาณนาฬิกา 16 เมกกะเฮิรตซ์
- กระแสไฟตรงต่อผมขา0 40 มิลลิแอมแปร์
- ปัจจุบันกระแสตรงสำหรับ3.3สำหรับขา 50 มิลลิแอมแปร์
- แรมสถิต 8 กิโล
- หน่วยความจำอ่านอย่างเดียวชนิด 4 กิโล

โปรแกรมและลบได้ด้วยกระแสไฟฟ้า



รูปที่ 2.13 ไมโครคอนโทรลเลอร์ Arduino Mega 2560 R3



รูปที่ 2.14 แผนภาพขา Arduino Mega 2560

ตารางที่ 2.1 ตารางขาการส่ง Arduino Mega 2560

Pin Number	Pin Name	Mapped Pin Name
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)

4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1

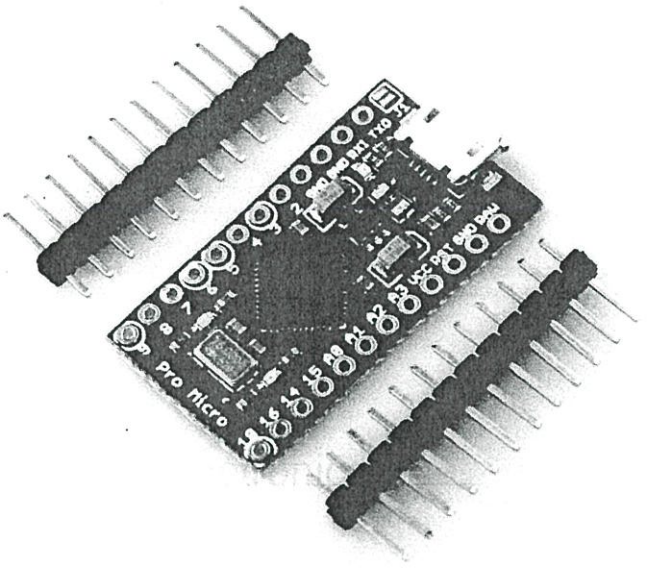
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	

66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1

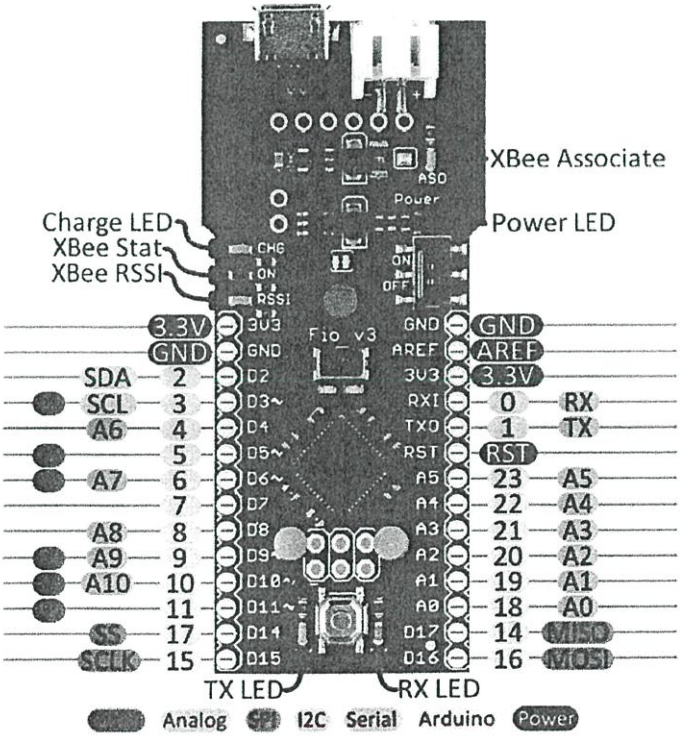
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

2.7 ไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega32U4 Arduino

ไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega32U4 Arduino เป็นรับส่งสัญญาณยูเอชบี (USB) ภายใน สามสิบสองยูสี่ (32U4) ช่วยให้สามารถ เพิ่ม การเชื่อมต่อยูเอชบีบนกระดาน และทำไปด้วยส่วนต่อประสานภายนอก บอร์ดนี้มีขนาดเล็ก 4 ช่องทางของ เอดีซี 10 บิต 5 ขา บีดับเบิ้ลยูเอ็ม 12 ไดออส เป็นฮาร์ดแวร์ที่เชื่อมต่อแบบอนุกรม และตัวรับ ตัวส่ง ทำงานที่ 16 เมกกะเฮิร์ท และ 5 โวลล์ มีการควบคุมแรงดันไฟฟ้าเพื่อสามารถรับแรงดัน ได้ถึง 12 โวลล์



รูปที่ 2.15 ไมโครคอนโทรลเลอร์ Leonardo Pro Micro ATmega32U4



รูปที่ 2.16 แผนภาพขา Leonardo Pro Micro ATmega32U4

2.7.1 คุณสมบัติ

- ATmega32U4 ทำงานที่ 5 โวลล์ ต่อ 16 เมกกะเฮิร์ท
- 4 ขา เอดีซี (ADC) 10 บิต

- 12 ขา ดิจิตอล I ต่อ Os
- รับและส่ง และ ฮาร์ดแวร์ การเชื่อมต่อแบบอนุกรม
- ขนาด 1.3 คูณ 0.7 นิ้ว

2.8 มอดูลส่งแสงอินฟาเรด

ตัวส่งอินฟราเรด มีลักษณะคล้ายกันกับ ไดโอดเปล่งแสง สีใส ขนาดเส้นผ่านศูนย์กลาง 5 มิลลิเมตร ความยาวคลื่นประมาณ 940 นาโนเมตร ซึ่งป้อนความถี่ใบอัสให้กับตัวส่งอินฟราเรด เพื่อให้เกิดการส่งแสงอินฟราเรดออกมา และสามารถทดสอบแสงที่มันเปล่งออกมาโดยการส่องผ่าน กล้องดิจิตอล ซึ่งถ้ามีการส่งอินฟราเรดออกมาจากหลอด ก็จะมีแสงสีชมพูอมม่วงสว่างออกมาที่ตัวหลอด โดยการส่งอินฟราเรด จะทำการสร้างสัญญาณพาห์ความถี่ 38 กิโลเฮิร์ท ผสมกับข้อมูล ส่งออกมาโดยผ่าน ไดโอดเปล่งแสง อินฟราเรด

รูปที่ 2.17 หลอดไดโอดเปล่งแสง ตัวส่งอินฟราเรด

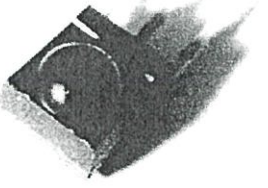
2.9 มอดูลรับแสงอินฟราเรด (TSOP 4838)

มอดูลรับแสงอินฟราเรด ทำงานในลักษณะคล้ายโฟโต้ทรานซิสเตอร์ (Phototransistor) มี 3 ขา คือ ขาออกข้อมูล (OUT) สายดิน (GND) ไฟ (VCC) ขาหมายเลข 1,2,3 ตามลำดับ สามารถใช้กับแรงดัน 5 โวลล์ ได้

2.9.1 การทำงานของมอดูลรับแสงอินฟราเรด 38 กิโลเฮิร์ท

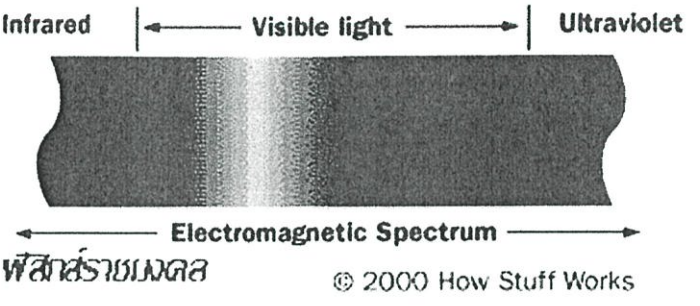
เมื่อจ่ายไฟแล้วยังไม่มีการส่งแสงอินฟราเรดที่ผสมสัญญาณพาห์ 38 กิโลเฮิร์ท มายังรับตัวรับแสงอินฟราเรด ข้อมูลส่งออกของวงจรเป็นตรรกะ “ 1 ” ตลอดเวลา เมื่อมีการส่งแสงอินฟราเรดที่ผสมสัญญาณพาห์ 38 กิโลเฮิร์ทผสมอยู่ด้วย โมดูลรับแสงอินฟราเรดทำงานให้ข้อมูลส่งออกเป็นตรรกะ “0”

P/N: TSOP4838



รูปที่ 2.18 มอดูลรับแสงอินฟราเรด 38 กิโลเฮิร์ท (TSOP 4838)

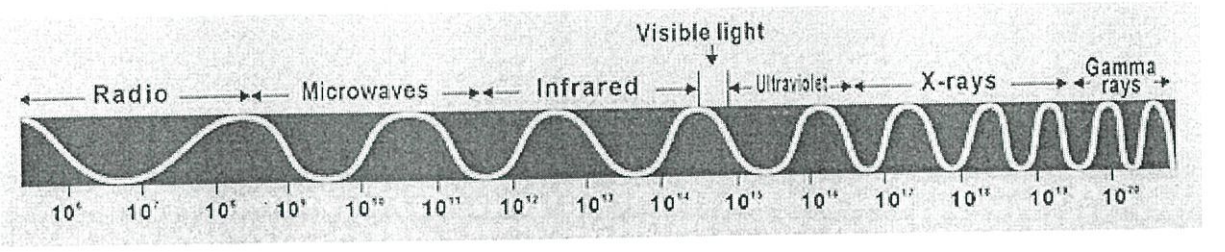
2.10 แสงใต้แดง(infrared)



รูปที่ 2.19 แถบเส้นสเปคตรัม

นักวิทยาศาสตร์ชาวอังกฤษ ได้ค้นพบ อินฟราเรดสเปคตรัม ในปีคริสศักด์ราช 1800 โดยเขาได้ทำการทดลองวัดอุณหภูมิของแถบสีต่างที่เปล่งออกมาเป็นสีรุ้งจากทรงปริซึม พบว่าอุณหภูมิความร้อนจะเพิ่มขึ้นตามลำดับจากสีม่วงและสูงสุดที่แถบสีสีแดง ซึ่งเรียกว่า “อินฟราเรด” (ของเขตที่ต่ำกว่าแถบเส้นสีแดง)

รังสีอินฟราเรด (Infrared) มีชื่อเรียกอีกชื่อว่า รังสีใต้แดง หรือรังสีความร้อน เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นอยู่ในระหว่างคลื่นวิทยุและแสงมีความถี่ในช่วง 1011 เฮิรตซ์ ถึง 1014 เฮิรตซ์ มีความถี่ในช่วงเดียวกับคลื่นไมโครเวฟ มีความยาวคลื่นอยู่ระหว่างแสงสีแดงกับคลื่นวิทยุ สสารทุกชนิดที่มีอุณหภูมิระหว่างลบ 200 องศาเซลเซียส ถึง 4,000 องศาเซลเซียส จะปล่อยรังสีอินฟราเรด คุณสมบัติเฉพาะตัวของรังสีอินฟราเรด ยกตัวอย่างเช่น ไม่เบี่ยงเบนในสนามแม่เหล็กไฟฟ้า แต่ที่แตกต่างกันก็คือ คุณสมบัติที่ขึ้นอยู่กับความถี่ คือยิ่งความถี่สูงมากขึ้น พลังงานก็สูงขึ้นมากด้วย



รูปที่ 2.20 แถบเส้นเสปคตรัมที่ถูกนำไปใช้งาน

แสงในย่านอินฟราเรด แตกออกเป็น 3 ส่วนดังนี้

1. แสงใกล้แดง (Near infrared) อยู่ใกล้กับช่วงที่ตาสามารถมองเห็นได้ความยาวคลื่นที่อยู่ในช่วง 0.7 ไมครอน ถึง 1.3 ไมครอน (1 ไมครอน = 0.000001 เมตร) หรือ 700 นาโนเมตร ถึง 1300 นาโนเมตร (1 นาโนเมตร = 0.000000001 เมตร)
2. แสงกลางแดง (Mid-IR) มีความยาวคลื่นอยู่ในช่วง 1.3 ไมครอน ถึง 3 ไมครอน ทั้งแสงใกล้แดงกับแสงกลางแดง นิยมใช้อุปกรณ์ทางอิเล็กทรอนิกส์ เช่นการ เครื่องควบคุมระยะไกล (รีโมท คอนโทรล)
3. เทอร์มอล อินฟราเรด (Thermal infrared) หรือที่เรียกว่า แสงร้อน มีความยาวคลื่นอยู่ในช่วง 3 ไมครอน ถึง 30 ไมครอน

2.11 การกล้ำสัญญาณ

เมื่อต้องการจะส่งสัญญาณหรือข้อมูลผ่านช่องทางสื่อสาร จำเป็นที่จะต้องอาศัยหลักพลังงานไฟฟ้าในการช่วยพาสัญญาณเหล่านั้นให้ช่วยเคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่ง ขบวนการในการเพิ่มพลังงานไฟฟ้าดังกล่าว หรือที่เรียกว่า การมอดูเลต หรือหลักการกล้ำสัญญาณที่เป็นพลังงานไฟฟ้ามีความถี่สูงและคงที่ รวมทั้งแอมพลิจูดสูงด้วยเรียกว่า สัญญาณคลื่นพาห์ (Signal Carrier) อุปกรณ์สำหรับกล้ำสัญญาณ (Modulator) จะสร้างสัญญาณคลื่นพาห์และรวมเข้ากับสัญญาณข้อมูลเพื่อให้สัญญาณมีแรงพอที่จะส่งผ่านสื่อกลางไปยังอีกจุดหนึ่ง เมื่อถึงปลายทางจะมีอุปกรณ์ในการแยกสัญญาณคลื่นพาห์ออก เรียกการแยกสัญญาณนี้ว่าการดีมอดูเลต (Demodulation)

2.11.1 ประเภทของการกล้ำสัญญาณ

การกล้ำสัญญาณแปลงสัญญาณข้อมูลเชิงอุปมานเป็นสัญญาณเชิงอุปมานเพื่อส่งผ่านไปในช่องทางสื่อสารเชิงอุปมานมี 3 วิธี คือ

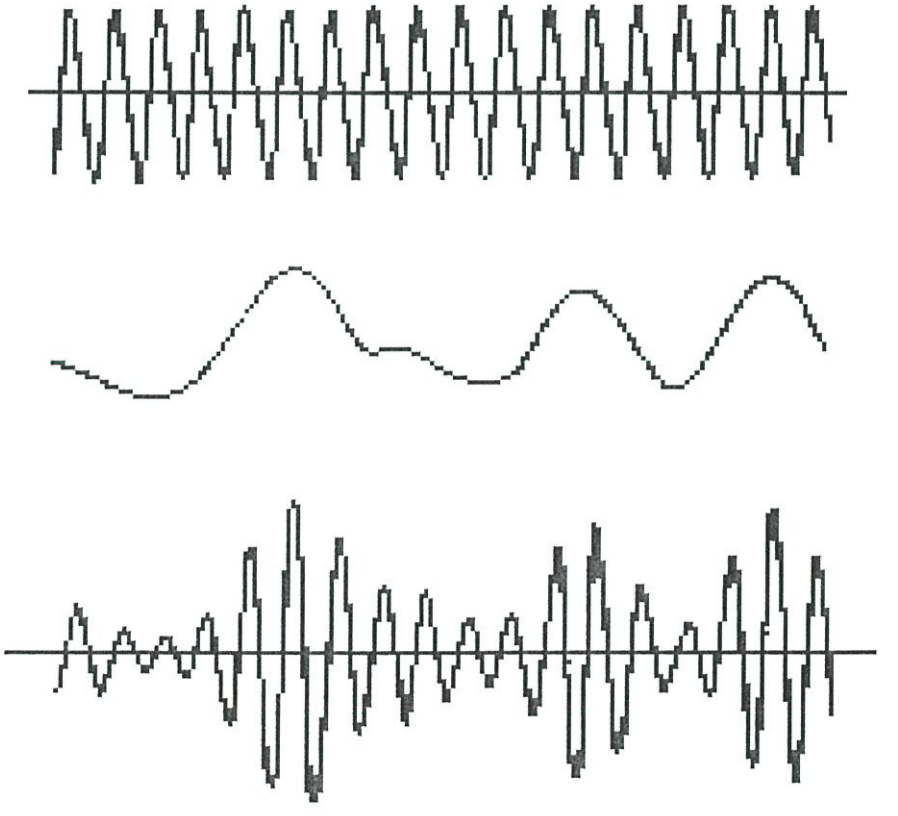
1. การกล้ำสัญญาณทางแอมพลิจูด (Amplitude Modulation : AM)
2. การกล้ำสัญญาณทางความถี่ (Frequency Modulation : FM)
3. การกล้ำสัญญาณทางเฟส (Phase Modulation : PM)

2.11.2 การกล้ำสัญญาณทางแอมพลิจูด (AM)

สัญญาณของคลื่นพาห์จะมีความถี่สูงกว่าความถี่ของสัญญาณข้อมูล เพื่อให้สามารถพาสัญญาณข้อมูลไปได้ไกล สัญญาณแอมพลิจูดที่กล้ำแล้วจะมีความถี่เท่ากับความถี่ของสัญญาณคลื่นพาห์ โดยมีขนาดของสัญญาณเปลี่ยนแปลงไปตามแอมพลิจูดของสัญญาณด้วย

ข้อเสียของการกล้ำสัญญาณแบบแอมพลิจูด

1. ความกว้างแถบความถี่ของสัญญาณแอมพลิจูด เป็นย่านความถี่ไม่สูงทำให้มีสัญญาณรบกวนจากภายนอกเข้ามาได้ง่ายๆ
2. การส่งสัญญาณแบบแอมพลิจูด สิ้นเปลืองพลังงานมากส่วนใหญ่ใช้ในการส่งคลื่นพาห์



รูปที่ 2.21 สัญญาณคลื่นพาห์ สัญญาณข้อมูล สัญญาณกล้ำสัญญาณแอมพลิจูด

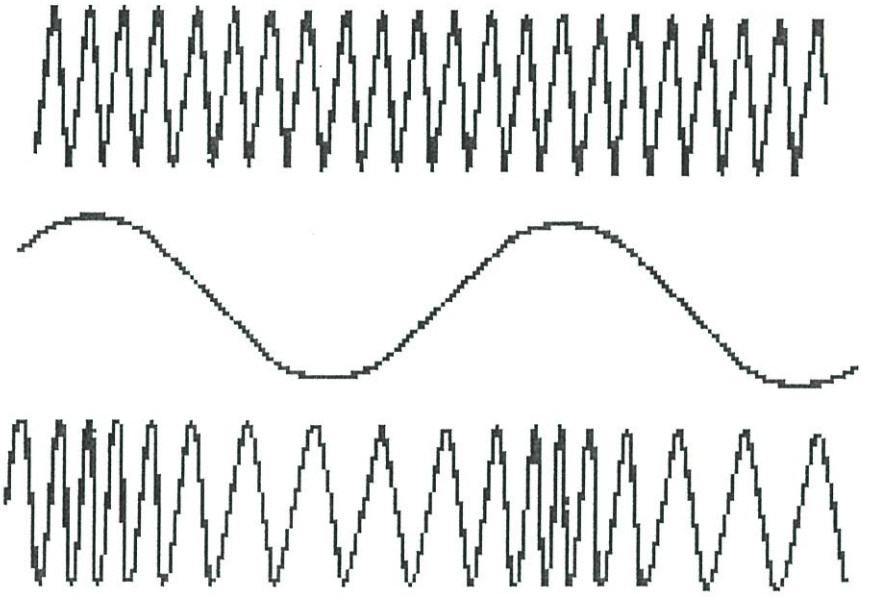
2.11.3 การกล้ำสัญญาณทางความถี่ (FM)

สัญญาณทางความถี่ที่กล้ำสัญญาณแล้วจะมีแอมพลิจูดคงที่ แต่ความถี่ของสัญญาณจะไม่คงที่ เปลี่ยนแปลงตามแอมพลิจูดของสัญญาณข้อมูล

ข้อเสียของการกล้ำสัญญาณแบบสัญญาณทางความถี่คือ

1. ต้องการความกว้างแถบความถี่ที่มีขนาดกว้าง เนื่องจากสัญญาณข้อมูลมีหลากหลายความถี่

2. คุณภาพดีกว่าการกล้ำสัญญาณแบบแอมพลิจูด แต่มีการทำงานจะซับซ้อนกว่า

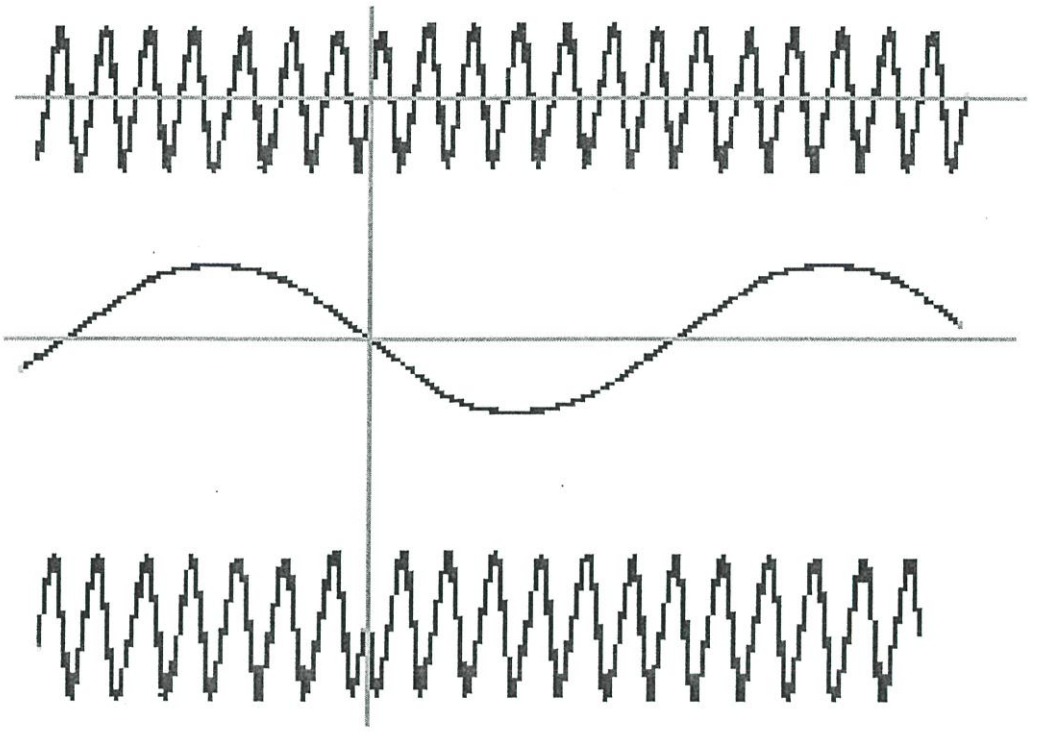


รูปที่ 2.22 สัญญาณคลื่นพาห์ สัญญาณข้อมูล สัญญาณกล้ำสัญญาณความถี่

2.11.4 การกล้ำสัญญาณทางเฟส (PM)

การกล้ำสัญญาณแบบทางเฟส จะให้ครึ่งรอบของสัญญาณเป็นมุม 180 องศา และเมื่อครบ 1 รอบ จะกลายเป็น 360 องศา สัญญาณกล้ำสัญญาณจะมีการเปลี่ยน มุมเฟสทุกครั้งที่มุมเฟสของสัญญาณ ค่าข้อมูลต่างจากมุมเฟสของสัญญาณคลื่นพาห์เท่ากับ 180 องศา การกล้ำสัญญาณแบบทางเฟส นิยมใช้ในการแพร่ภาพสีทีวี แต่แอมพลิจูด กับทางเฟส นิยมใช้การกระจายเสียงวิทยุ การกล้ำสัญญาณทางเฟส เป็นวิธีการที่ดีแต่วงจรจะซับซ้อนจึงไม่นิยมใช้ส่งสัญญาณข้อมูลเชิงอุปมา

การกล้ำสัญญาณทางเฟสมักจะนำมาใช้ในการส่งสัญญาณข้อมูลคอมพิวเตอร์ที่ต้องการความเร็วในการส่งข้อมูลสูงมากๆ



รูปที่ 2.23 สัญญาณคลื่นพาห์ สัญญาณข้อมูล สัญญาณกล้ำสัญญาณทางเฟส

2.12 ตัวต้านทาน หรือ เรซิสเตอร์ (Resistor)

ตัวต้านทานเป็นอุปกรณ์ไฟฟ้าชนิดสองขั้ว ที่สร้างความต่างศักย์ทางไฟฟ้าขึ้นคร่อมขั้วทั้งสองขั้ว โดยมีสัดส่วนมากน้อยตามกระแสไฟที่ไหลผ่าน อัตราส่วนระหว่างความต่างศักย์ และปริมาณกระแสไฟฟ้า ก็คือ ค่าความต้านทานทางไฟฟ้า

หน่วยค่าความต้านทานไฟฟ้าตามระบบเอสไอ (SI Unit) คือ โอห์ม (Ohm) อุปกรณ์ที่มีความ ต้านทาน ค่า 1 โอห์ม หากมีความต่างศักย์ 1 โวลต์ไหลผ่าน จะให้กระแสไฟฟ้า 1 แอมแปร์ ซึ่ง เท่ากับการไหลของประจุไฟฟ้า 1 คูลอมบ์ (ประมาณ 6.241506 คูณ 1018 อิเล็กตรอน) ต่อวินาที ตัวต้านทานอาจแบ่งออกเป็น ตัวต้านทานที่มีค่าคงที่ และ ตัวต้านทานแบบปรับค่าได้

2.12.1 ตัวต้านทานแบบมีค่าคงที่

ตัวต้านทานทั่วไปอาจมีรูปร่างทรงกระบอก โดยที่มีสารตัวต้านทานอยู่ที่แกนกลาง หรือเป็นฟิล์มอยู่ที่ผิว และมีแกนโลหะตัวนำออกมาจากปลายทั้งสอง ตัวต้านทานที่มีรูปร่างนี้เรียกว่า ตัวต้านทานรูปร่างแบบ แอกเซียล ดังในรูปด้านขวามือ ตัวต้านทานใช้สำหรับกำลังสูงจะถูกออกแบบให้มีรูปร่างที่ถ่ายเทความร้อนได้ดี โดยมักจะเป็น ตัวต้านทานแบบขดลวด ตัวต้านทานที่มักจะพบเห็นบนแผงวงจร ตัวอย่างเช่น คอมพิวเตอร์นั้น โดยปกติจะมีลักษณะเป็น ตัวต้านทานแบบประกบผิวหน้า (surface-mount) ขนาดเล็ก และ จะไม่มีขาโลหะตัวนำยื่นออกมา

นอกจากนั้นตัวต้านทานอาจจะถูกรวมภายใน อุปกรณ์วงจรรวม (IC - integrated circuit) โดยตัวต้านทานจะถูกสร้างขึ้นระหว่างกระบวนการผลิต และแต่ละไอซี มีตัวต้านทานถึงหลายล้านตัวอยู่ภายในตัวต้านทาน

2.12.2 ตัวต้านทานแบบปรับค่าได้

ตัวต้านทานสามารถปรับค่าได้ เป็นตัวต้านทาน ที่ค่าความต้านทานสามารถเปลี่ยนได้ โดยอาจมีปุ่มสำหรับเลื่อนเพื่อปรับค่าความต้านทานและบางครั้งก็เรียก โพเทนติโอมิเตอร์ (potentiometers) หรือ รีโอสแตต (rheostats)

ตัวต้านทานแบบปรับค่าได้ มีทั้งแบบหมุนได้เพียงรอบเดียว ถึง แบบที่หมุนแบบเป็นเกลียวได้หลายๆรอบ บางชนิดมีอุปกรณ์แสดงนับรอบหมุน เนื่องจากตัวต้านทานปรับค่าได้ มีส่วนของโลหะที่ขัดสีสึกกร่อน บางครั้งอาจขาดความน่าเชื่อถือ ในตัวต้านทานปรับค่าได้รุ่นใหม่ๆจะใช้วัสดุซึ่งทำจากพลาสติกทนทานต่อการสึกกร่อนจากการขัดสี และ กัดกร่อน

- รีโอสแตต (rheostat)

  ตัวต้านทานปรับค่าได้จะมี 2 ขา โดยที่ขาหนึ่งถูกยึดตายตัว ส่วนอีกขาที่เหลือเลื่อนไปๆมาๆได้ ปกติใช้สาหรับที่มีปริมาณกระแสไฟไหลผ่านสูง

- โพเทนติโอมิเตอร์ (potentiometer)

  ตัวต้านทานปรับค่าได้ ที่พบเห็นได้ทั่วๆไป โดยเป็นปุ่มสามารถปรับความดังสำหรับเครื่องขยายเสียงได้

2.12.3 การอ่านค่าความต้านทาน

ตัวต้านทานแบบแอกเซียลส่วนใหญ่จะมีการระบุค่าความต้านทานด้วยแถบสีส่วนแบบประกบผิวหน้าจะระบุค่าด้วยตัวเลข

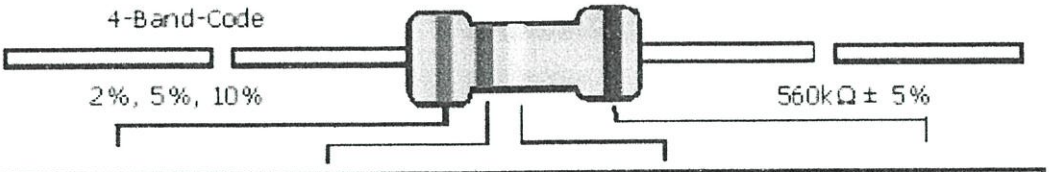
- ตัวต้านทานแบบมี 4 แถบสี

ตัวต้านทานแบบมี 4 แถบสีนั้นเป็นแบบที่นิยมมากที่สุด โดยจะมีแถบสีซึ่งระบายเป็นเส้น 4 เส้นรอบๆตัวต้านทาน โดยค่าตัวเลขของ 2 แถบแรกนั่นจะเป็น ค่าสองหลักแรกของความต้านทาน และ แถบที่ 3 จะเป็นตัวคูณ และ แถบที่ 4 จะเป็นค่าขอบเขตของความเบี่ยงเบน ซึ่งมีค่าเป็น 2 เปอร์เซ็นต์ 5 เปอร์เซ็นต์ และ 10 เปอร์เซ็นต์ ค่าของรหัสสีตามมาตรฐาน EIA EIA-RS-279

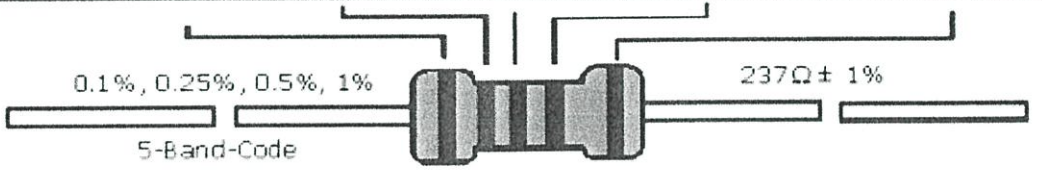
ตารางที่ 2.2 ตารางอ่านค่าตัวต้านทาน

สี	แถบ 1	แถบ 2	แถบ 3 (ตัวคูณ)	แถบ 4 (ขอบเขตความเบี่ยงเบน)	สัมประสิทธิ์ของอุณหภูมิ
ดำ	0	0	$\times 10^0$		
เหลือง	4	4	$\times 10^4$		25 ppm
เขียว	5	5	$\times 10^5$	±0.5% (D)	
ขาว	9	9	$\times 10^9$		
ทอง			×0.1	±5% (J)	
เงิน			×0.01	±10% (K)	
ไม่มีสี				±20% (M)	

สีแดง ถึง ม่วง เป็นสีรุ้ง โดยที่สีแดงเป็นสีพลังงานต่ำ และ สีม่วงเป็นสีพลังงาน



COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	±0.5% (D)
Blue	6	6	6	1MΩ	±0.25% (C)
Violet	7	7	7	10MΩ	±0.10% (B)
Grey	8	8	8		±0.05%
White	9	9	9		
Gold				0.1	± 5% (J)
Silver				0.01	± 10% (K)



รูปที่ 2.24 รูปวิธีอ่านค่าตัวต้านทาน

2.12.4 ค่าที่พึงประสงค์

ตัวต้านทานมาตรฐานที่ผลิต มีค่าตั้งแต่มิลลิโอห์มถึงจิกะโอห์ม ซึ่งในช่วงนี้จะมีเพียงบางค่าที่เรียกว่า ค่าที่พึงประสงค์เท่านั้นที่ถูกผลิตออกมา และตัวทรานซิสเตอร์ที่เป็นอุปกรณ์แยกภายในท้องตลาดเหล่านี้นั้น ทางปฏิบัติแล้วไม่ได้มีค่าตามอุดมคติ ดังนั้นเลยมีการระบุเขตของการเบี่ยงเบนจากค่าที่ระบุ โดยการใช้แถบสีแถบสุดท้าย

เงิน 10 เปอร์เซ็นต์

ทอง 5 เปอร์เซ็นต์

แดง 2 เปอร์เซ็นต์

น้ำตาล 1 เปอร์เซ็นต์

ตัวต้านทานที่มีความแม่นยำมากกว่าปกติ ก็มีขายในท้องตลาดทั่วไป

- ตัวต้านทานแบบมี 5 แถบสี

ตัวต้านทาน 5 แถบสีนั้นปกติใช้สำหรับตัวต้านทานมีความแม่นยำสูง โดยมีค่าขอบเขตของความ เบี่ยงเบน 1เปอร์เซ็นต์ 0.5เปอร์เซ็นต์ 0.25เปอร์เซ็นต์ 0.1เปอร์เซ็นต์ แถบสีทั้งหมด 3 แถบแรกนั้นใช้ระบุตัวความต้านทาน แถบที่ 4 จะใช้ระบุค่าตัวคูณ และ แถบที่ 5 จะใช้

ระบุขอบเขตของความเบี่ยงเบน ส่วนตัวต้านทานแบบ 5 แถบสีที่จะมีความแม่นยำปกติ พบได้ในตัวต้านทานรุ่นเก่าๆ หรือตัวต้านทานชนิดพิเศษ ซึ่งค่าขอบเขตของความเบี่ยงเบน จะอยู่ในตำแหน่งปกติคือ แถบที่ 4 ส่วนแถบที่ 5 จะนั้นใช้บอกค่าสัมประสิทธิ์ของอุณหภูมิ

2.12.5 ตัวต้านทานแบบเอสเอ็มที (SMT)

ตัวต้านทานแบบประกบผิวหน้า จะระบุค่าความต้านทานด้วยรหัสตัวเลข โดยตัวต้านทานเอสเอ็มที(SMT) ความแม่นยำปกติจะระบุด้วยรหัสเลข 3 หลัก สองหลักแรกบอกค่าสองหลักแรกของความต้านทาน และ หลักค่าที่ 3 คือค่าเลขยกกาลังของ 10 ตัวอย่างเช่น 472 ใช้หมายถึง 47 เป็นค่าสองหลักแรกของค่าความต้านทานคูณด้วยเลข 10ยกกาลังสอง 47 X 10^2 = 47 X 100 = 4700 โอห์ม ส่วนตัวต้านทานเอสเอ็มที(SMT) มีความแม่นยำสูง จะใช้รหัสเลข 4 หลัก โดยที่เลข 3 หลักแรกบอกค่าสามหลักแรกของความต้านทาน และหลักที่ 4 คือค่าเลขยกกำลังของ 10

2.12.6 กฎของโอห์ม

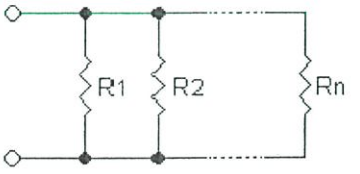
กฎของโอห์ม (Ohm's law) คือสมการที่อธิบายความสัมพันธ์ระหว่างความต่างศักย์, กระแสไฟฟ้า และความต้านทานของวัตถุต่างๆ ที่มีอยู่ว่า V = I x R

แรงดันไฟฟ้า คือความต่างศักย์ในวัตถุ ในหน่วยโวลต์ กระแสไฟฟ้าที่ไหลผ่านวัตถุ ในหน่วยแอมแปร์ และความต้านทานในหน่วยโอห์ม

- วงจรขนาน

ตัวต้านทานต่อแบบขนาน จะมีค่าความต่างศักย์เท่ากันทุกตัว จึงหาความต้านทานที่ สมมูล (R_{eq}) เสมือนว่ามีตัวต้านทานเพียงตัวเดียว ได้ดังตัวอย่างนี้

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n} \qquad R_{eq} = R_1 \| R_2 = \frac{R_1 R_2}{R_1 + R_2}$$

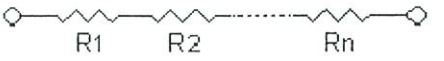


รูปที่ 2.25 รูปการต่อตัวต้านทานแบบขนาน

ตัวต้านทานสามารถแทนตัวต้านทานที่ต่อกันแบบขนานกัน ด้วยเส้นตรง 2 เส้น ได้ สำหรับตัวต้านทาน 2ตัว สามารถเขียนได้ดังนี้

- วงจรแบบอนุกรม

กระแสไฟฟ้าไหลผ่านตัวต้านทานแบบอนุกรมจะเท่ากันเสมอ แต่ความต่างศักย์ของตัวต้านทานแต่ละตัวไม่เท่ากัน ดังนั้น ความต่างศักย์ทั้งหมดจึงจะเท่ากับผลรวมของความต่างศักย์ จึงสามารถหาความต้านทานได้



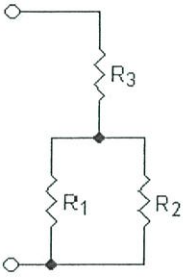
รูปที่ 2.26 รูปการต่อตัวต้านทานแบบอนุกรม

$$R_{eq} = R_1 + R_2 + \cdots + R_n$$

- ตัวต้านทานที่ต่อแบบขนานและแบบอนุกรม

ตัวต้านทานที่ต่อกันแบบขนานและแบบอนุกรมรวมกันนั้น สามารถแบ่งได้เป็นส่วนเล็กๆก่อนแล้วจึงคำนวณความต้านทานทีละส่วนๆได้ ดังตัวอย่างนี้

$$R_{eq} = (R_1 \| R_2) + R_3 = \frac{R_1 R_2}{R_1 + R_2} + R_3$$



รูปที่ 2.27 รูปการต่อตัวต้านทานแบบขนานและอนุกรม

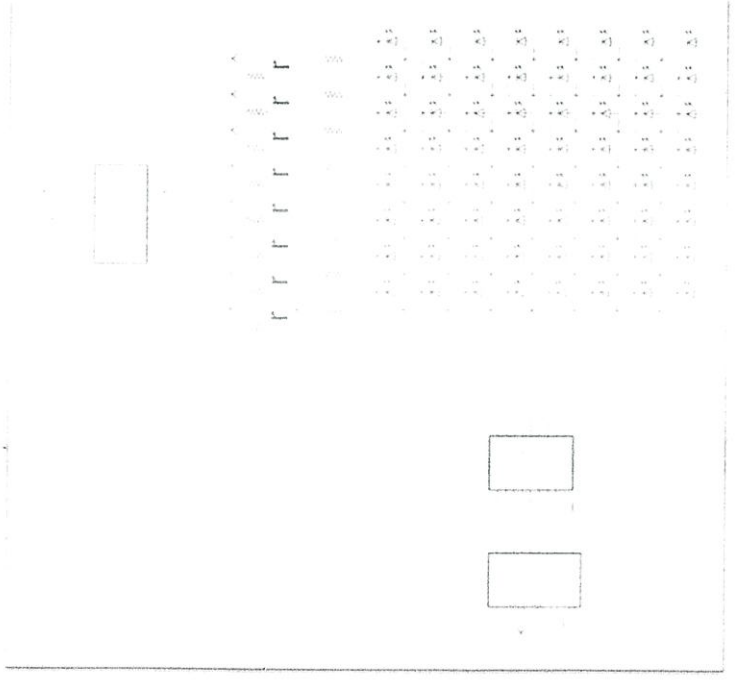
บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การออกแบบวงจร

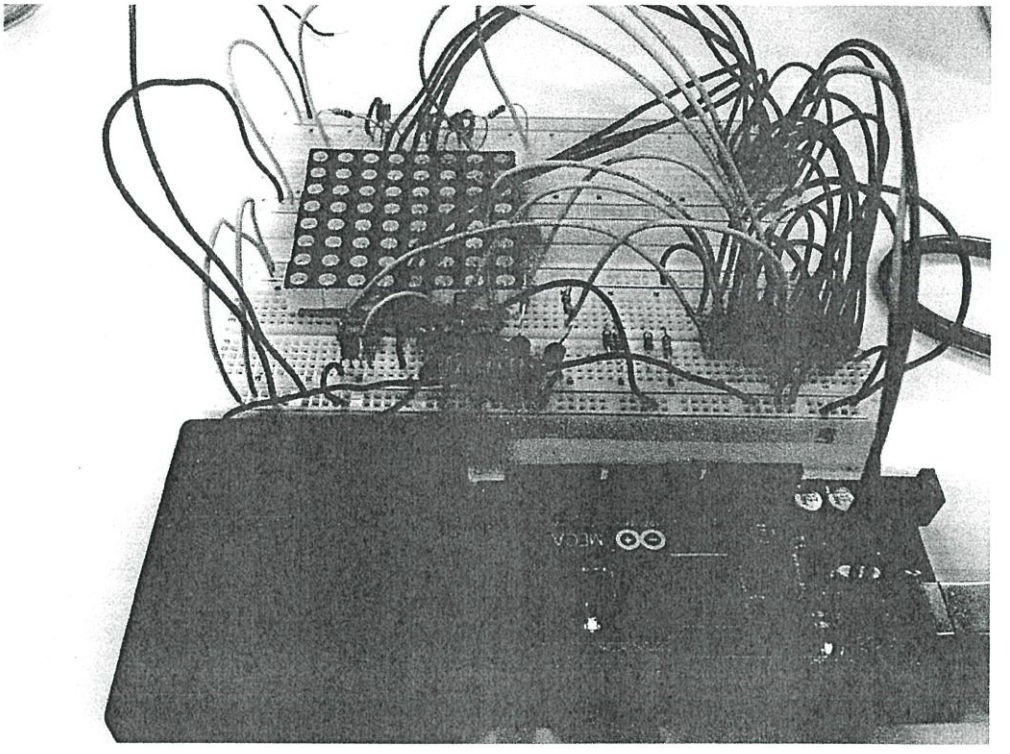
การจัดทำโครงงานในการปฏิบัติงานต่างๆให้สำเร็จจะต้องมีการวางแผนเป็นส่วนๆและทดสอบในส่วนของฮาร์ดแวร์ และ ซอร์ฟแวร์ ให้สามารถทำงานร่วมกันได้ โดยมีการกำหนดเค้าโครงของโครงงาน ส่วนประกอบที่จำเป็น และนำมาออกแบบและดำเนินงานให้บรรลุเป้าหมาย

3.1.1 การออกแบบ

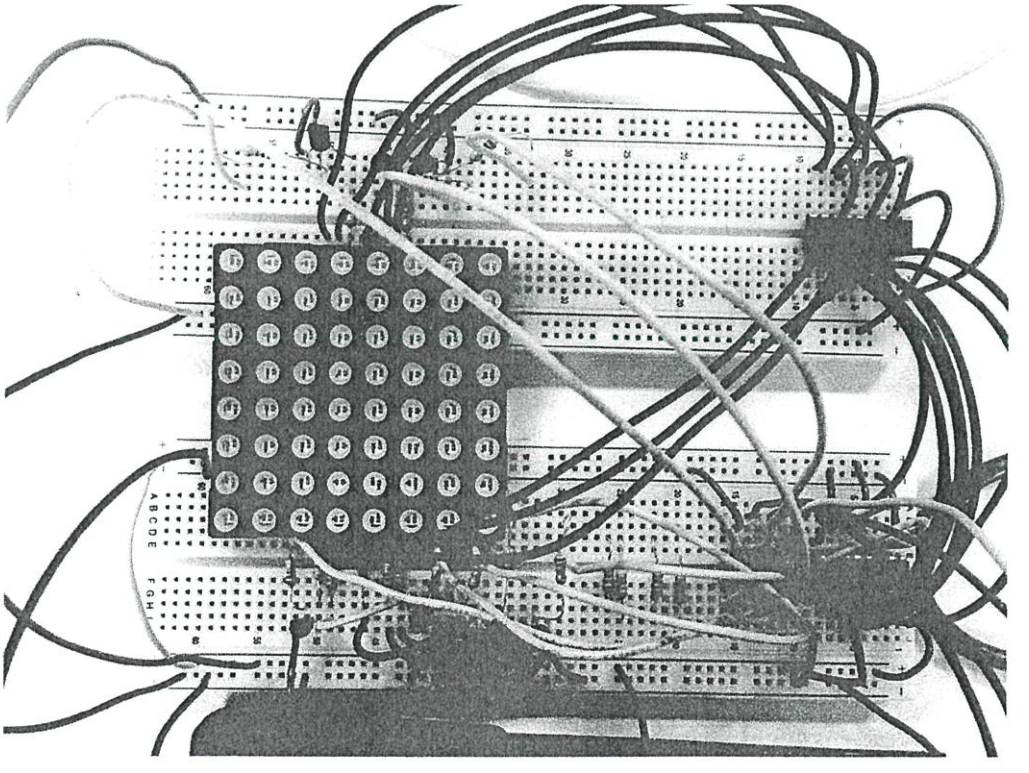
การออกแบบป้ายไฟส่องสว่าง โดยใช้หลอดไดโอดเปล่งแสง ด๊อท เมทริกซ์ จำนวน 2 ตัว เพื่อต่อกันให้เป็นข้อมูลที่มีได้ขนาดใหญ่เพิ่มมากยิ่งขึ้น โดยสำหรับวงจรไดโอดเปล่งแสง ด๊อท เมทริกซ์ 1 ตัวจะใช้ ตัว วงจรเลื่อนข้อมูล ไอซี เบอร์ 74HC595 ตัวขับเคลื่อนแบบแปดขา ไอซีเบอร์ ULN2803 ในการขับเคลื่อนวงจรได้ดังรูป



รูปที่ 3.1 แบบวงจร เมทริกซ์ 8 X 8 จำนวน 1 วงจร



รูปที่ 3.2 แบบวงจรเมทริกซ์ 8 X 8 แบบ 1 ชิ้น



รูปที่ 3.3 แบบวงจรเมทริกซ์ 8X8 แบบ 1 ชิ้น

3.1.2 หลักการทำงาน

ขาสัญญาณที่ทำการรับข้อมูลจากไมโครคอนโทรเลอร์คือขา 1,2,5,7,8,9,12 และขา 14 โดยเมื่อรับข้อมูลออกจากไมโครคอนโทรลเลอร์ต่อเนื่องไปทำให้ได้รับข้อมูลในแนวนอนโดยมีไอซีเบอร์ 74HC595 ทำหน้าที่ขับกระแสให้แก่แอลอีดี (LED)

ขาสัญญาณ 3,4,6,10,11,13,15 และ 16 โดยควบคุมการแสดงผลของไดโอดเปล่งแสง ในแนวตั้งด้วยการส่งตรรกะ '0' ให้แก่พีเอ็นพี ทรานซิสเตอร์ในคอลัมน์ที่ต้องการให้ไดโอดเปล่งแสงสว่าง

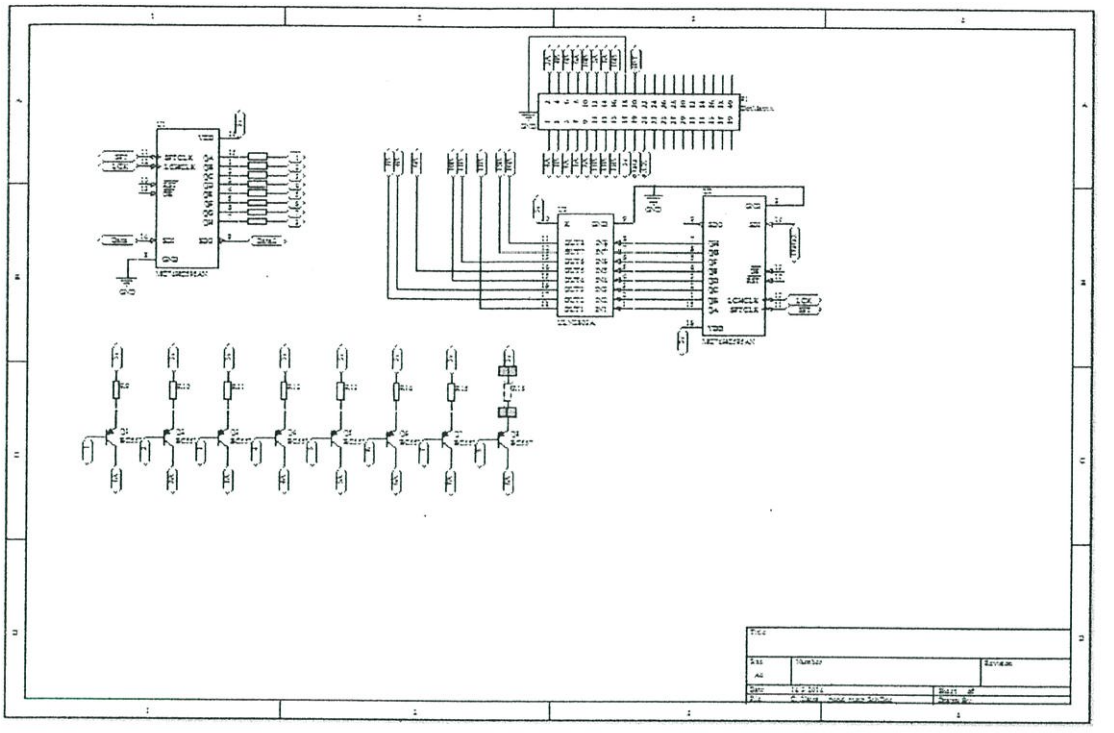
เมื่อต้องการควบคุมให้ไดโอดเปล่งแสงแถวที่ 1 และคอลัมน์ที่ 1 สว่าง ต้องส่งข้อมูล 0x01(00000001B) ไปยังขา 15 ของไอซีเบอร์ 74HC595 เป็นบิตที่ 1 ทำให้ไดโอดเปล่งแสงทุกตัวที่ต่ออยู่ในแนวนอนของแถวที่ 1 ได้รับตรรกะ "1" ทุกตัว และเมื่อส่งข้อมูล 0x00(000000000B) ไปยังขา 13 ของไอซี ULN2803 จะทำให้คอลัมน์ที่ 1 เป็นตรรกะ "0" ทรานซิสเตอร์ได้รับไบอัสตรง ทำให้ไดโอดเปล่งแสงในคอลัมน์ 1 ได้รับตรรกะ"0" ทุกตัว แต่จะมีไดโอดเปล่งแสงจำนวนแถว 1 คอลัมน์ 1 เท่านั้นที่ไฟจะส่องสว่างเนื่องจากจะมีกระแสไฟไหลครบวงจรโดยผ่านตรรกะ "1" ไปครบวงจรโดยผ่านทรานซิสเตอร์

3.1.3 การออกแบบวงจรบอร์ดแสดงผลแบบ 8x8 ด๊อทเมทริกซ์ (LED DOT MATRIX)

วงจรนั่นจะประกอบไปด้วย หลอดไดโอดเปล่งแสง ด๊อท เมทริกซ์ จำนวน 1 ตัว วงจรเลื่อนข้อมูล ไอซี เบอร์ 74HC595 จำนวน 2 ตัวเพื่อที่ตัวขับเคลื่อนแบบแปดขา ไอซีเบอร์ ULN2803 จำนวน 1 ตัว

3.1.4 การออกแบบแผนภาพเค้าร่าง (Schematic) ของวงจร

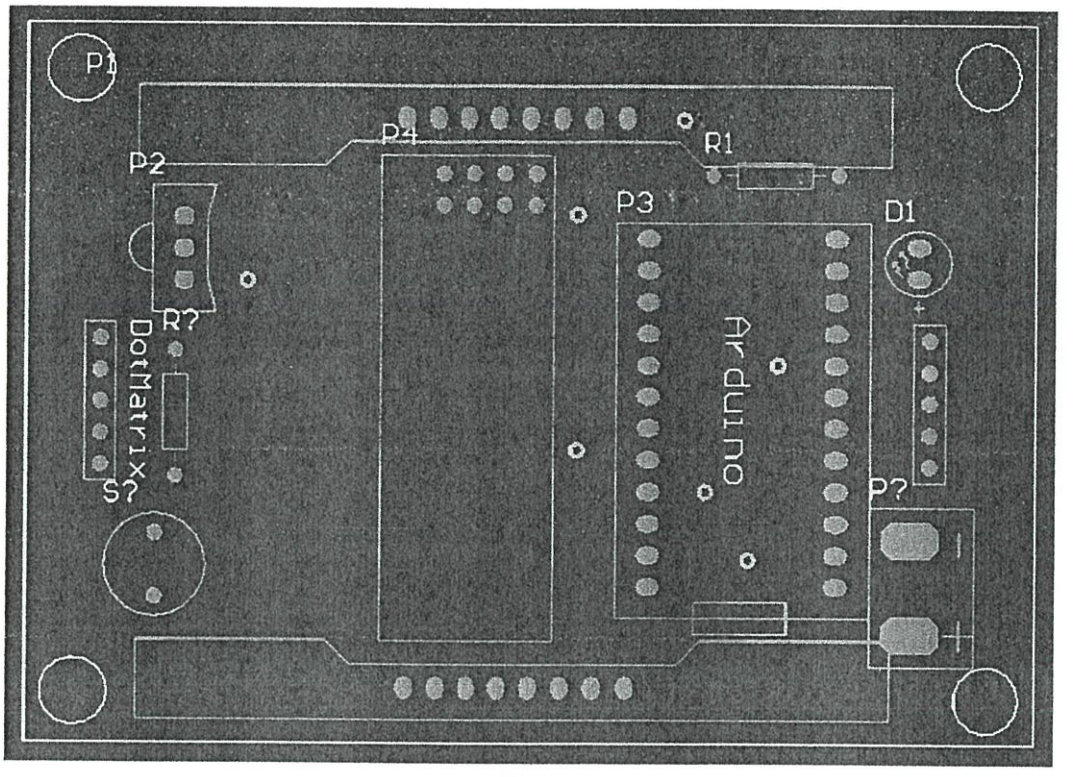
การออกแบบแผนภาพเค้าร่างของวงจรหลอดไดโอดเปล่งแสง ด๊อท เมทริกซ์แบบ 8X8 จะใช้โปรแกรมโพรทิอุส (Proteus) ในการออกแบบวงจร



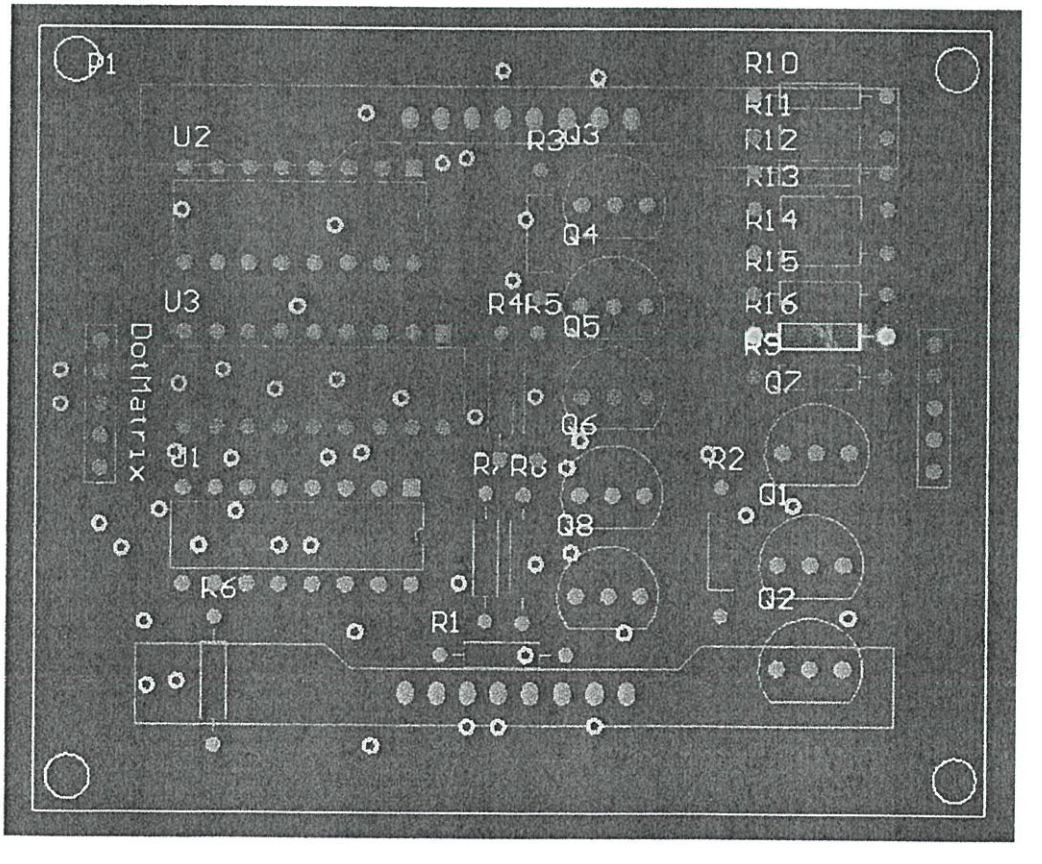
รูปที่ 3.4 วงจร หลอดไดโอดเปล่งแสง ด๊อท เมทริกซ์ 8X8

3.1.5 การออกแบบวงจรพีซีบี (PCB) ของวงจร

การออกแบบวงจรพีซีบี ของวงจรของวงจรแอลอีดี ด๊อทเมทริกซ์ แบบ 8X8 จะใช้โปรแกรมโพรทิอุส ในการออกแบบวงจร ลากสายไฟทองแดงขนาด 0.6 มิลลิเมตร และกำหนดจุดที่เชื่อมต่อของวงจรขนาด 1.6 มิลลิเมตร โดยขนาดแผงวงจรที่ได้จะมีความกว้าง 8.25 เซนติเมตร ยาว 24.00 เซนติเมตร ต่อจำนวน 1 แผ่น

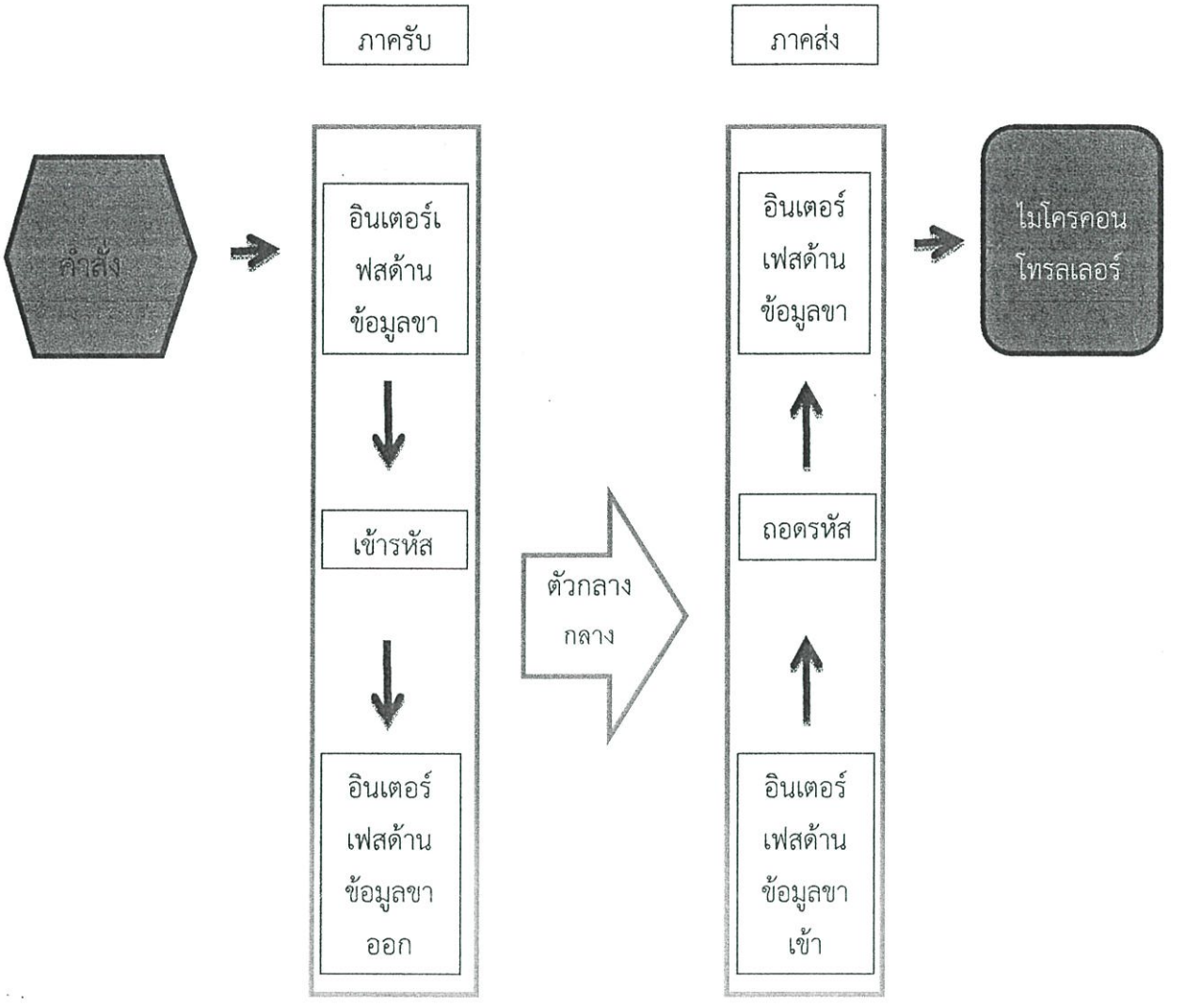


รูปที่ 3.5 วงจรแผนภาพเค้าร่างของวงจรแอลอีดี ด๊อทเมทริกซ์ ชั้นแรก (LED DOT MATRIX)



รูปที่ 3.6 วงจรภาพพีซีบี(PCB) ของวงจรแอลอีดี ด๊อทเมทริกซ์ (LED DOT MATRIX)

3.2 การควบคุมการทำงานของอินฟราเรด



รูปที่ 3.7 แผนภาพการทำงานของอินฟราเรด

3.2.1 หลักการทำงานของอินฟราเรดของตัวแม่

1.ตัวแม่จะมีตัวส่งอินฟราเรดเพื่อบอกว่าตัวเองคือแป้นหมายเลข 1 อยู่เสมอ

2.แป้นบอร์ดด๊อทเมทริกซ์จะแสดงข้อความที่ถูกกำหนดให้ เพื่อรอป้ายอื่นๆมาต่อเป็นข้อความที่ยาวขึ้น

3.2.2 หลักการทำงานของอินฟราเรดของตัวลูก

การควบคุมระยะไกลแบบไร้สายทิศทางเดียวโดยอินฟราเรด

อินฟราเรดเป็นการสั่งงานจากจุดหนึ่งแล้วเกิดการทำงานอีกจุดหนึ่ง โดยในวงจรของเราที่ทำคือ

1. อินฟราเรดจะทำการรับสัญญาณข้อมูล
2. อินฟราเรดจากนั่นนำข้อมูลที่รับได้จากตัวรับอินฟราเรดมาวิเคราะห์

2.1 อินฟราเรดหากได้รับสัญญาณหมายเลขแป้นบอร์ดด๊อทเมทริกซ์ใดก็ตามจะทำการบวกนำหมายเลขสุดท้ายของเลขที่รับได้ มาบวกเพิ่ม 1 และนำไปแทนที่ตำแหน่งสุดท้ายของตน

2.2 แป้นบอร์ดด๊อทเมทริกซ์แป้นใดที่ไม่ได้รับคำสั่งให้ทำการหยุดส่งสัญญาณใดๆ เพื่อรอรับ

3. แป้นบอร์ดตัวที่รอรับข้อมูลเมื่อได้รับข้อมูลมาก็ตรวจสอบว่าหมายเลขบอร์ดตัวสุดท้ายที่ได้รับมาคือหมายเลขใด

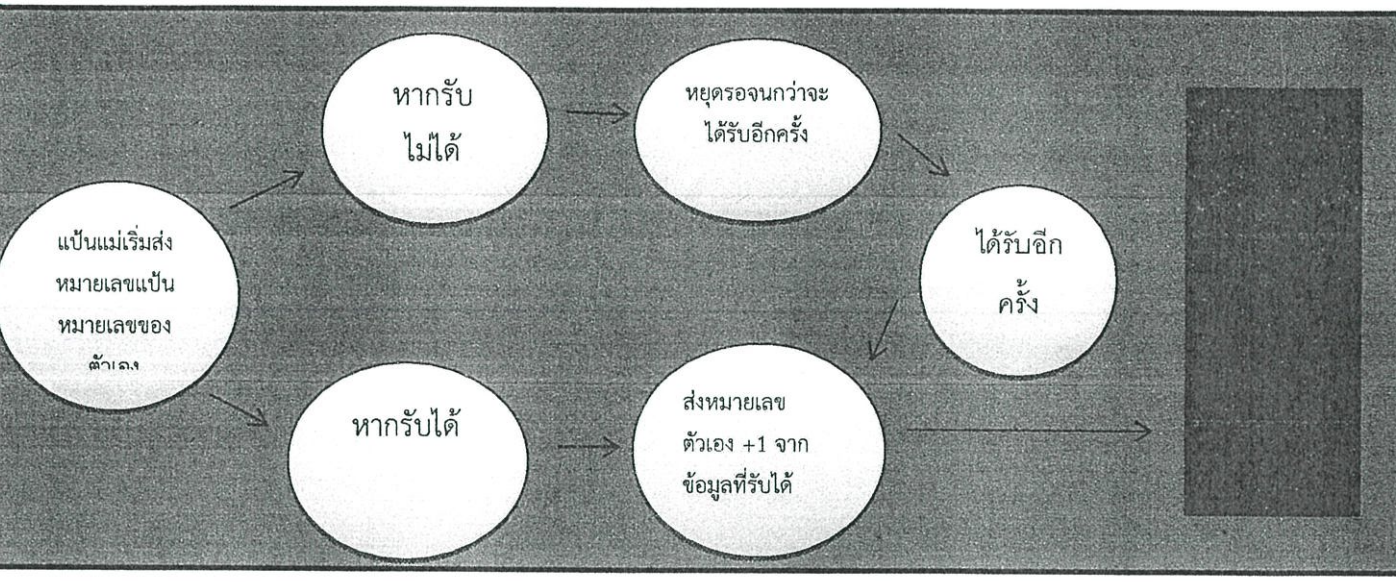
3.1 แป้นบอร์ดหากได้รับรหัสจากแป้นลงท้ายได้หมายเลข 1 แป้นก็จะทำการวิเคราะห์แล้วแสดงผลบนบอร์ด

3.2 แป้นบอร์ดหากได้รับรหัสจากแป้นลงท้ายได้หมายเลข 2 แป้นก็จะทำการวิเคราะห์แล้วแสดงผลบนบอร์ด

4. เมื่อแป้นบอร์ดได้แสดงผลแล้วจะทำการส่งสัญญาณอินฟราเรดที่กำหนดไว้ออกไป

3.2.3 ขั้นตอนวิธีของโปรแกรมอินฟราเรดตัวลูก

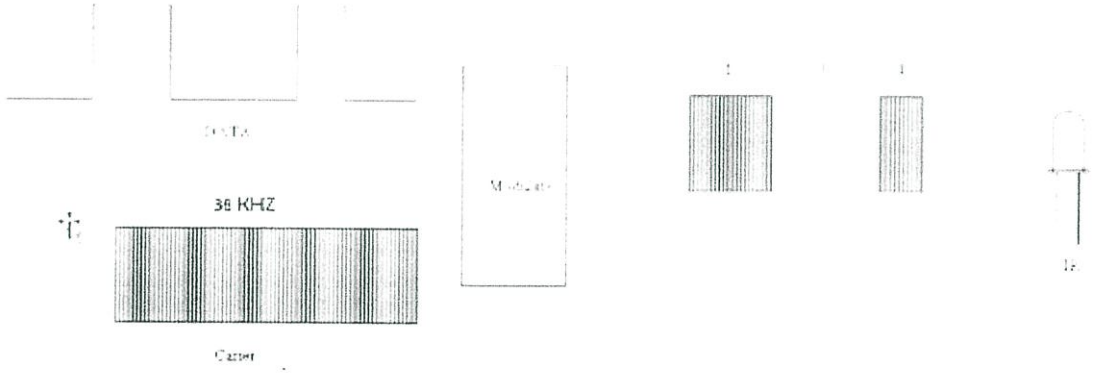
ป้ายที่ 1	ป้ายที่ 2	ป้ายที่ 3
1. รอรับ	รอรับ	รอรับ
2. รับได้	รับไม่ได้	รับไม่ได้
3. ส่งเลขตัวเอง #2	รอรับไปเรื่อยๆ	รอรับไปเรื่อยๆ
4. สั่งออกยังไมโครคอนโทรลเลอร์	ได้เลขจากป้าย #3	รอรับไปเรื่อยๆ
5. จบการทำงาน	ส่งเลขตัวเอง #4	รอรับไปเรื่อยๆ
6. -	สั่งออกยังไมโครคอนโทรลเลอร์	ได้เลขจากป้าย #3
7. -	จบการทำงาน	ส่งเลขตัวเอง #4
8. -	-	สั่งงาน
9. เริ่มทำงานใหม่	เริ่มทำงานใหม่	เริ่มทำงานใหม่



รูปที่ 3.8 แผนภาพระบบการส่งสัญญาณระยะไกล

3.2.4 หลักการกล้ำสัญญาณ

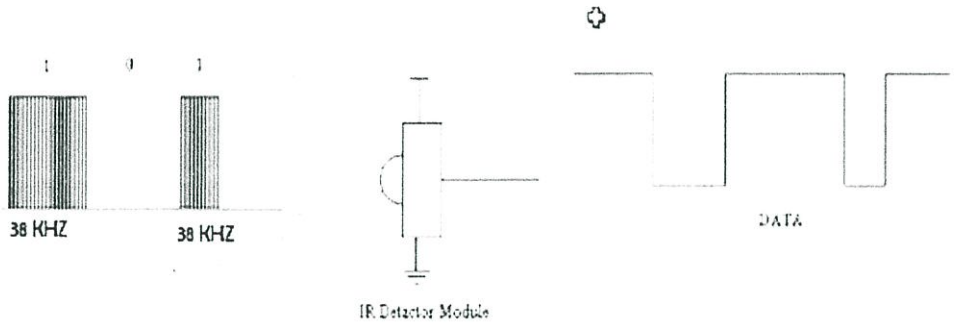
การนำสัญญาณข้อมูลที่จะส่งมาผสมรวมกับคลื่นพาห์ที่มีขนาดความถี่ 38 กิโลเฮิรตซ์แบบแอนด์กัน วิธีการนี้เป็น การเปลี่ยน ความถี่ของคลื่นพาห์ตามสัญญาณของข้อมูลที่เข้ามา



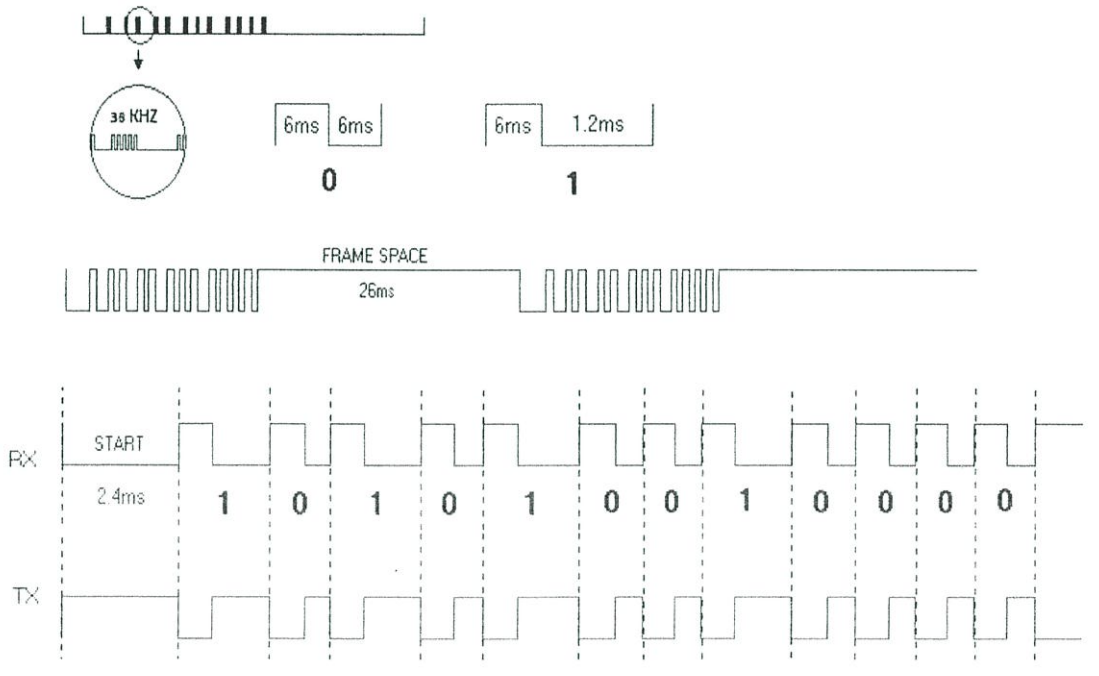
รูปที่ 3.9 แผนภาพการส่งสัญญาณขาออก

3.2.5 หลักการดีมอดูเลตสัญญาณ

จากสัญญาณข้อมูลที่ผสมกันกับสัญญาณคลื่นพาห์ถูกส่งมาในอากาศเข้ามาที่เครื่องตรวจจับเฉพาะไออาร์เพื่อทำการถอดรหัสและอินเวิร์สข้อมูลออกมาแล้วจะส่งเข้าสู่ไมโครคอนโทรลเลอร์



รูปที่ 3.10 แผนภาพการส่งสัญญาณขาเข้า



รูปที่ 3.11 แผนภาพการสัญญาณข้อมูลของอินฟราเรด

3.3 การควบคุมการแสดงผลหลอดไดโอดเปล่งแสง (LED)

การแสดงผลของหลอดไดโอดเปล่งแสง หลักของการแสดงผลของ LED ส่วนแสดงผลจะทำงานเมื่อไมโครคอนโทรลเลอร์นั่นได้รับคำสั่งการแสดงผลจากอินฟราเรดจะนำคำสั่งที่ประมวลผลมาได้ แล้วนำมาแสดงผลบน LED

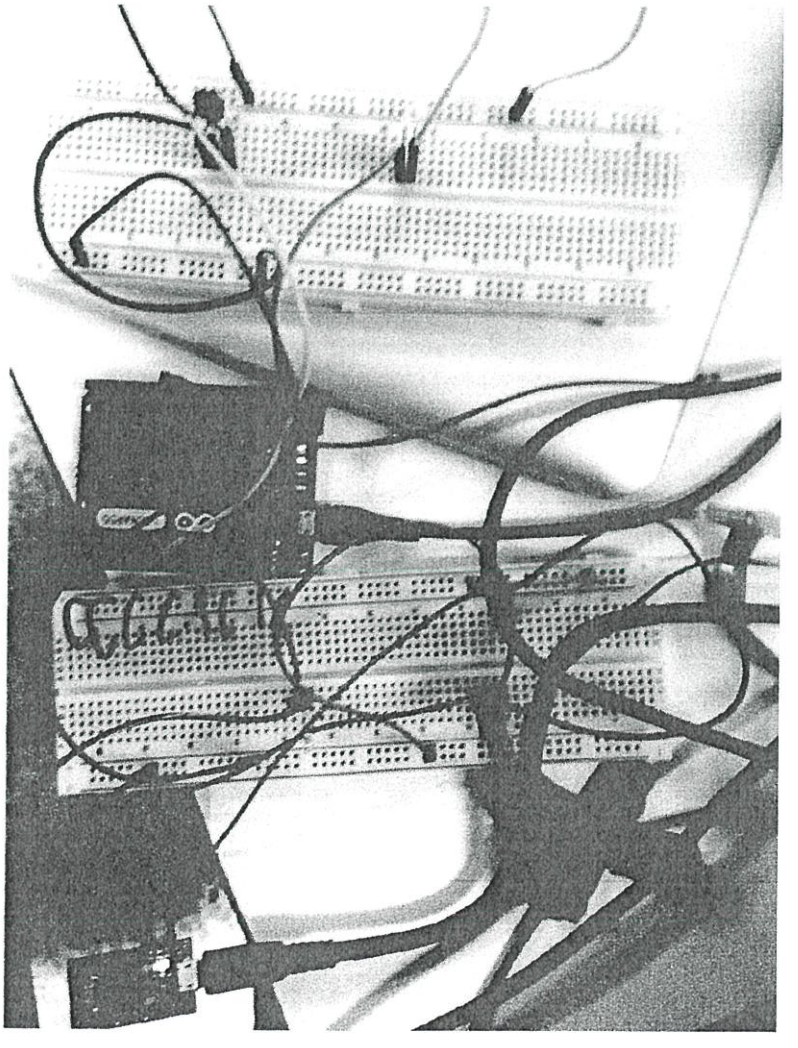
บทที่ 4
ผลการทดลอง

การทดลองป้ายไฟสอดประสานประกอบไปด้วยการทดลองหลัก 3 หัวข้อด้วยกัน ซึ่งในการทดลองใช้โปรแกรม Arduino 1.0.5-r2 เพื่อใช้ประกอบการทดลองหัวข้อดังต่อไปนี้

4.1 โปรแกรมสำหรับเรียนรู้ตำแหน่งของตนเองโดยอินฟราเรด

หลักการทำงานของโปรแกรมคือแต่ละป้ายจะสามารถเรียนรู้ตำแหน่งของตนเองจากอินฟราเรดตัวส่ง ซึ่งมีวงจรการทำงานของอินฟราเรดดังรูปต่อไปนี้

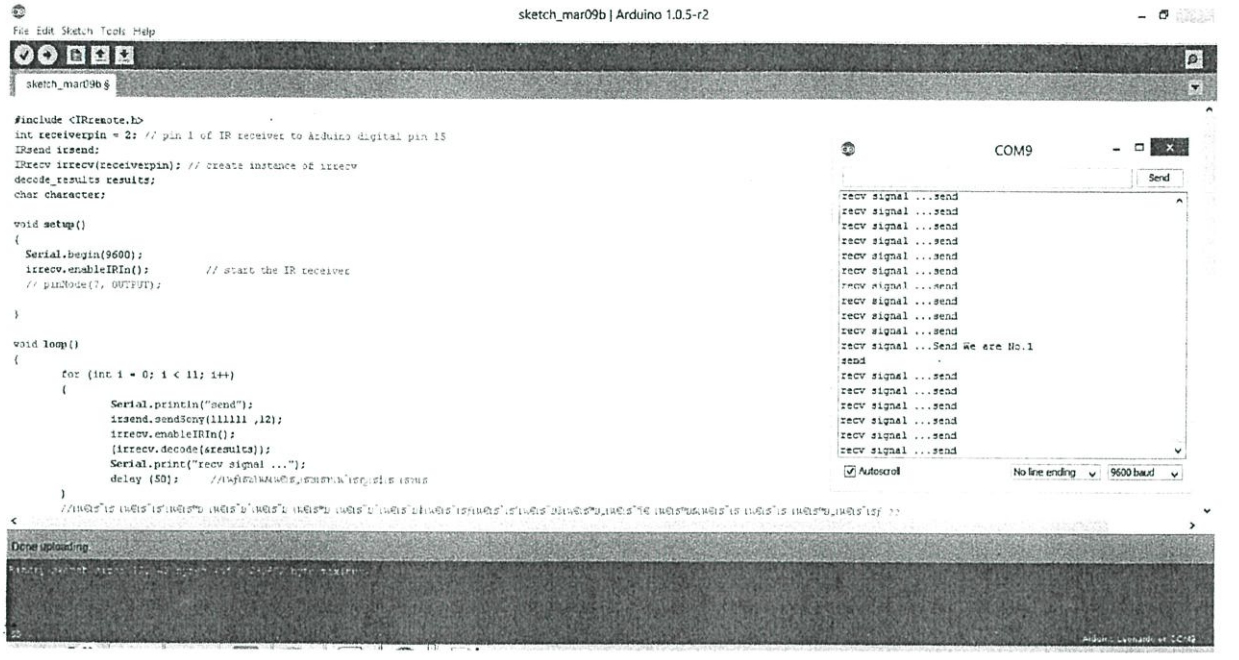
4.1.1 สำหรับวงจรที่ใช้ทดลอง



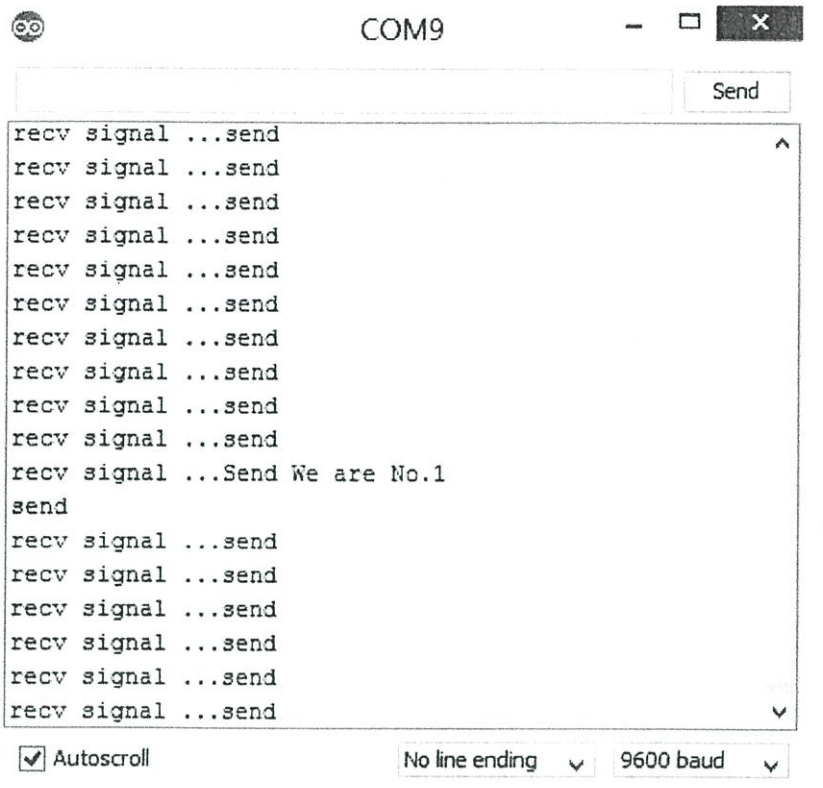
รูปที่ 4.1 รูปวงจรการทำงานของอินฟราเรด

4.1.2 สำหรับบอกตำแหน่งป้ายชื่อ 0xA0

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 1 จะแสดงคำพูด “ We are No. 1”

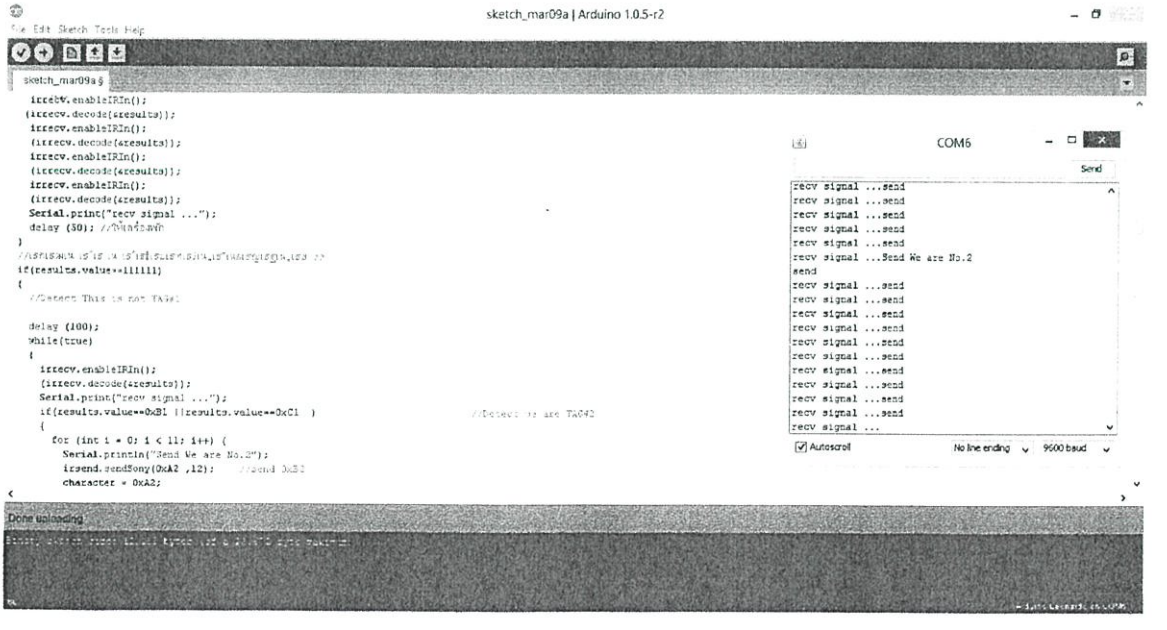


รูปที่ 4.2 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 1

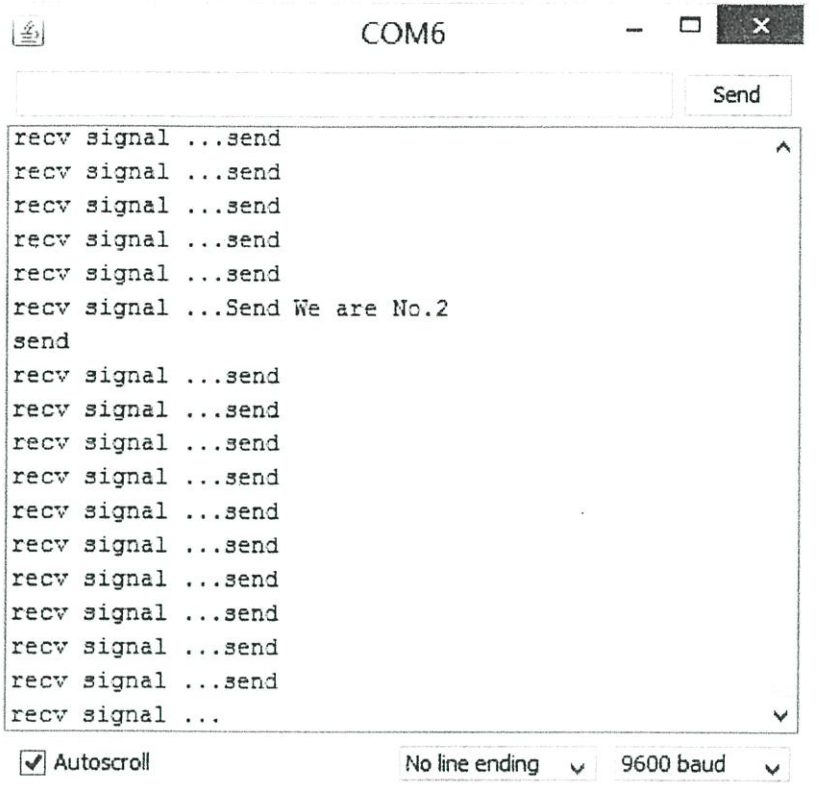


รูปที่ 4.3 รูปส่วนแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 1

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ จะแสดงคำพูด " We are No. 2"

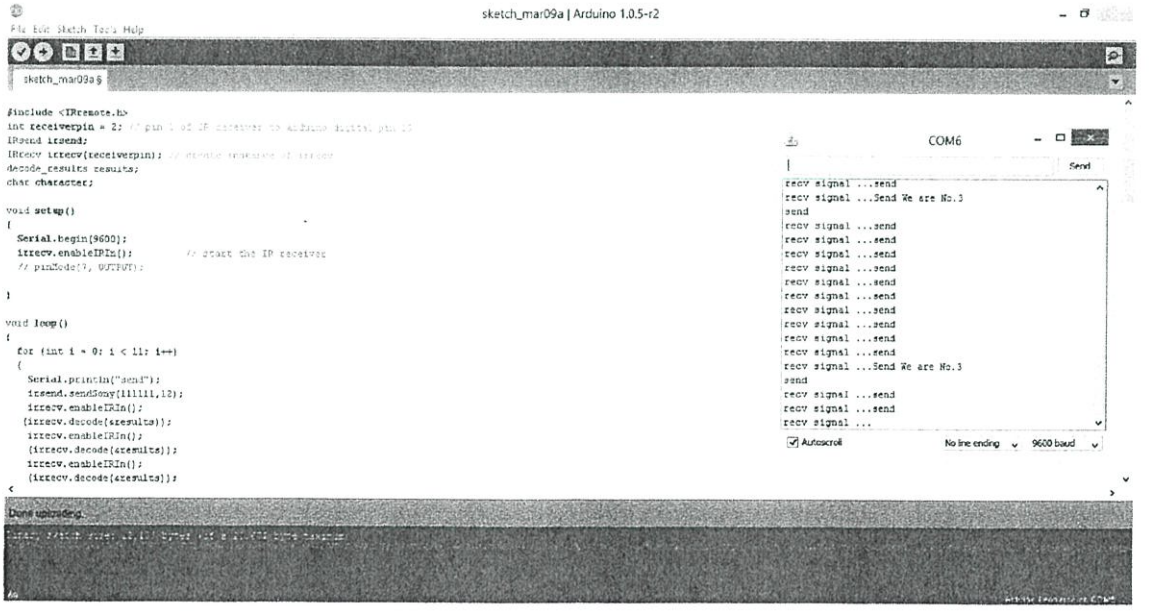


รูปที่ 4.4 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้าย ที่ 2

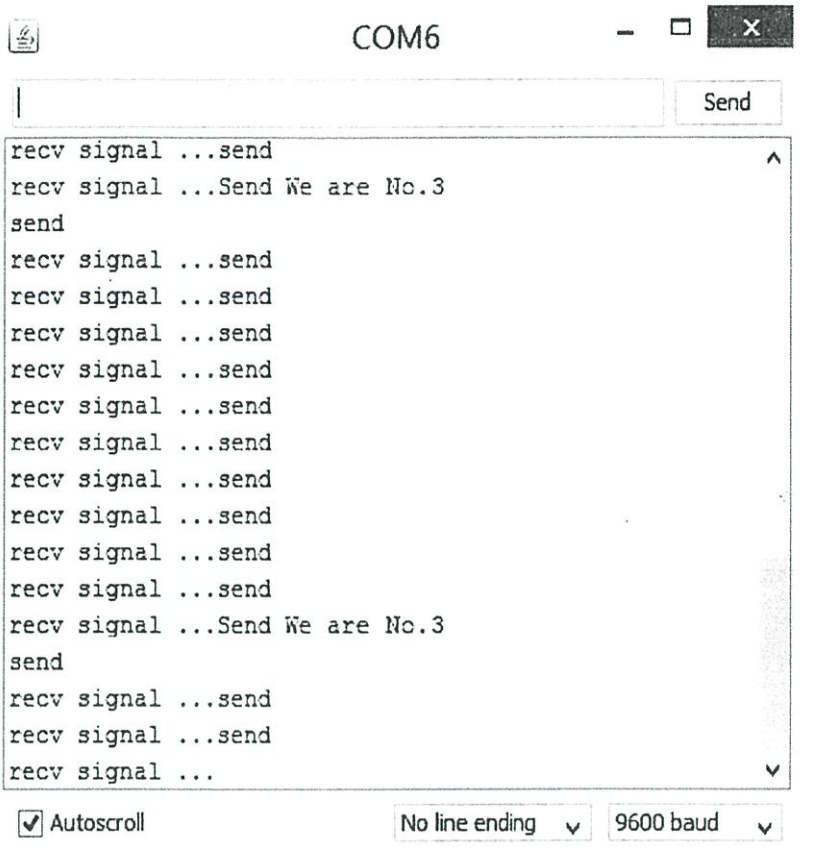


รูปที่ 4.5 รูปส่วนแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 2

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 3 จะแสดงคำพูด " We are No. 3"



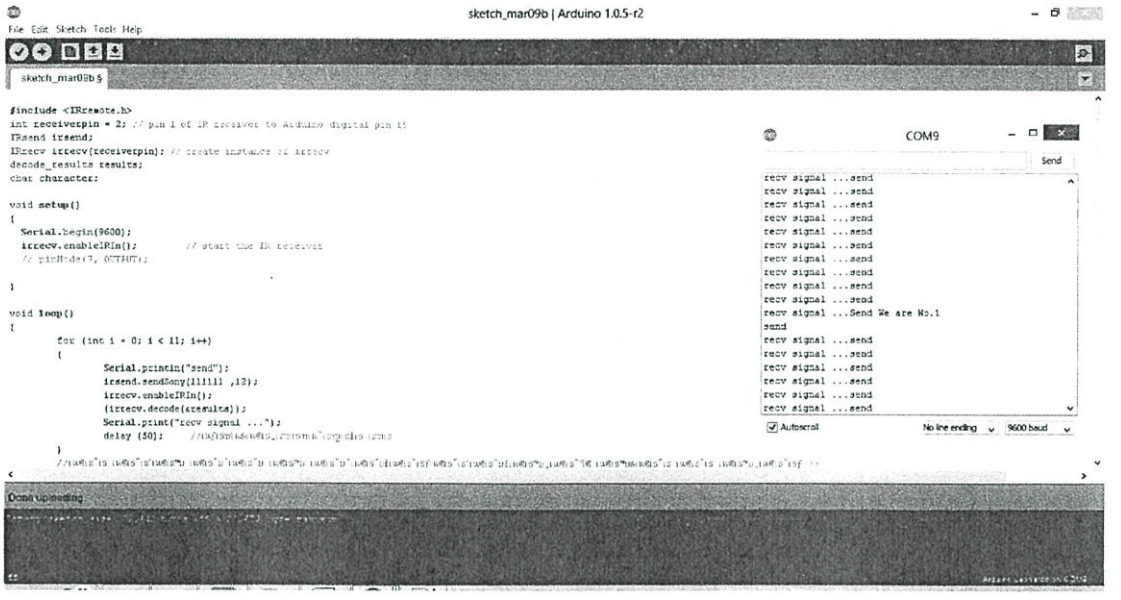
รูปที่ 4.6 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 3



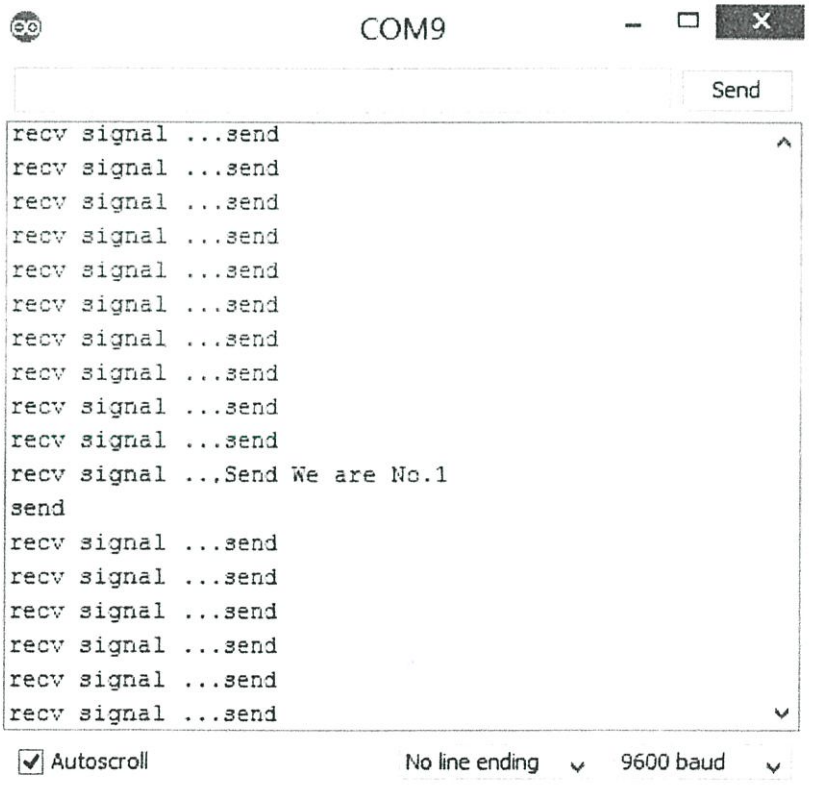
รูปที่ 4.7 รูปส่วนแสดงผลตำแหน่งของอินฟราเรดป้าย 0xA0 ตำแหน่งเป็นป้ายที่ 3

4.1.3 สำหรับบอกตำแหน่งป้ายชื่อ 0xB0

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 1 จะแสดงคำพูด " We are No. 1"

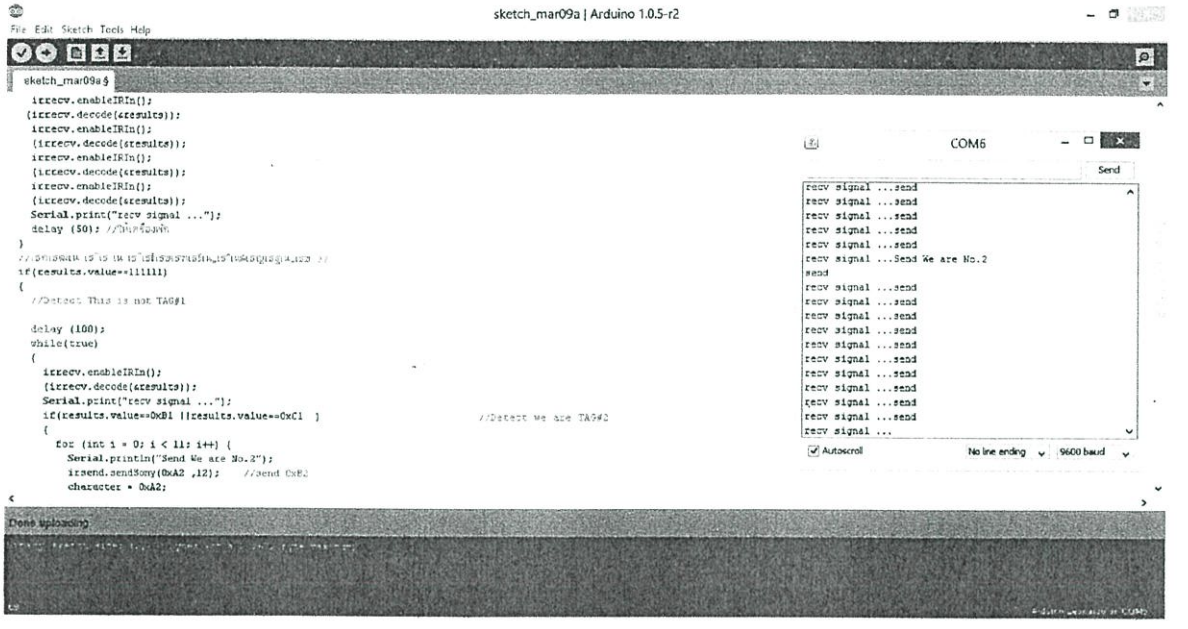


รูปที่ 4.8 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 1



รูปที่ 4.9 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 1

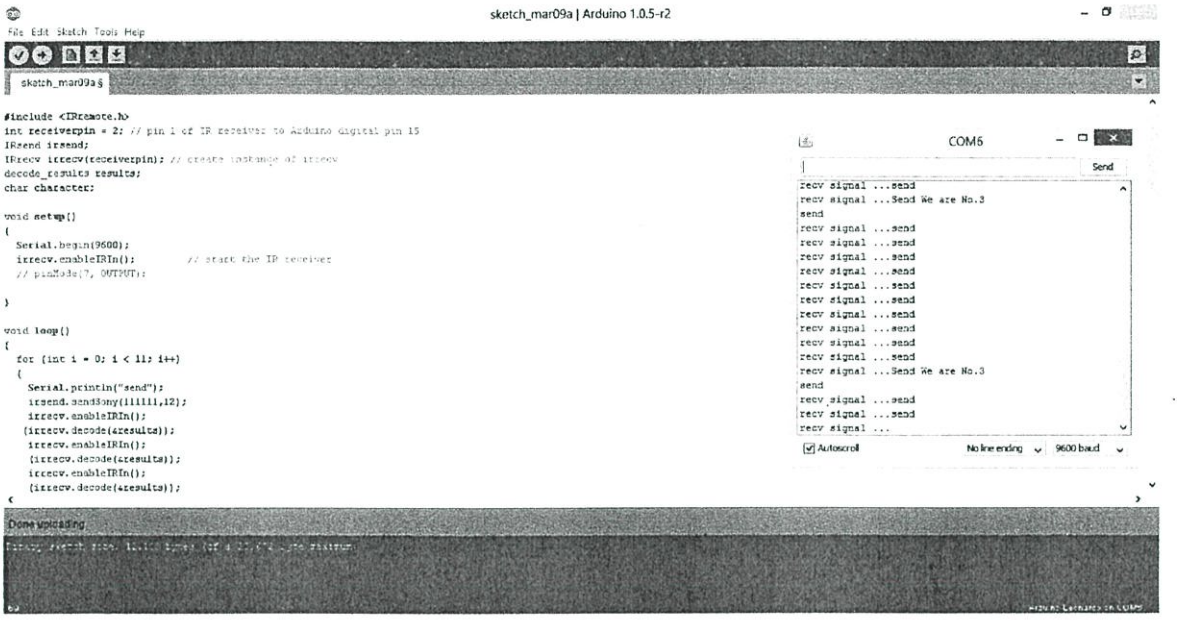
สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 2 จะแสดงคำพูด “ We are No. 2”



รูปที่ 4.10 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 2

รูปที่ 4.11 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 2

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 3 จะแสดงคำพูด “ We are No. 3”

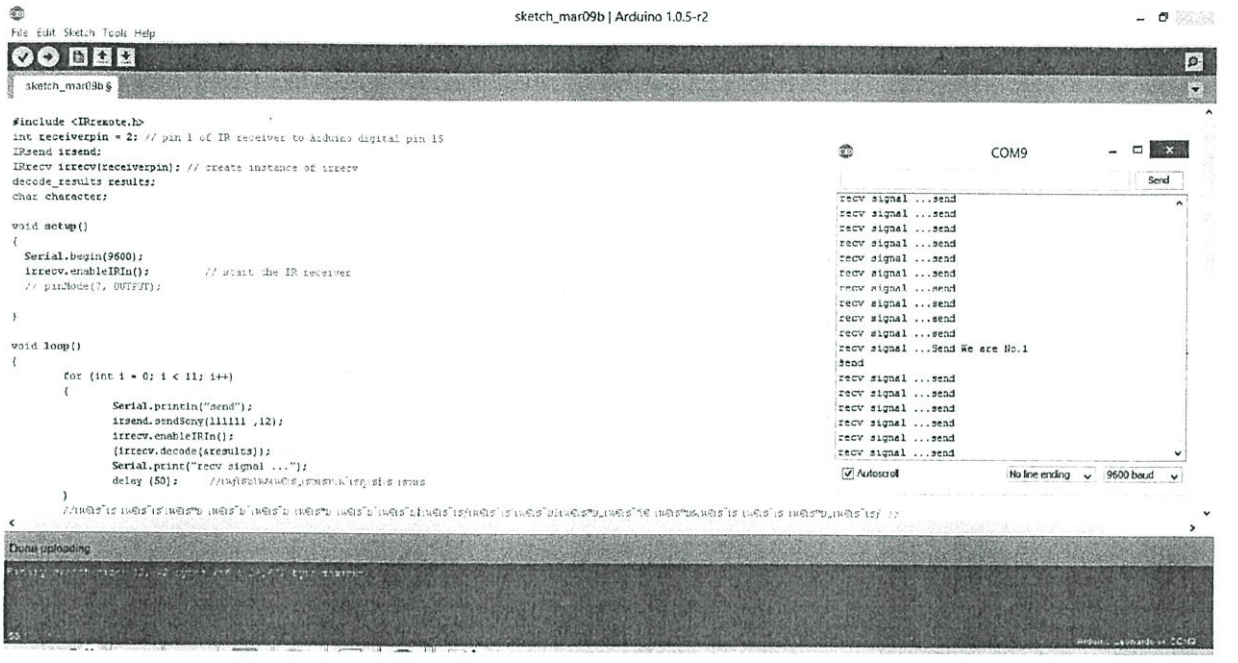


รูปที่ 4.12 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 3

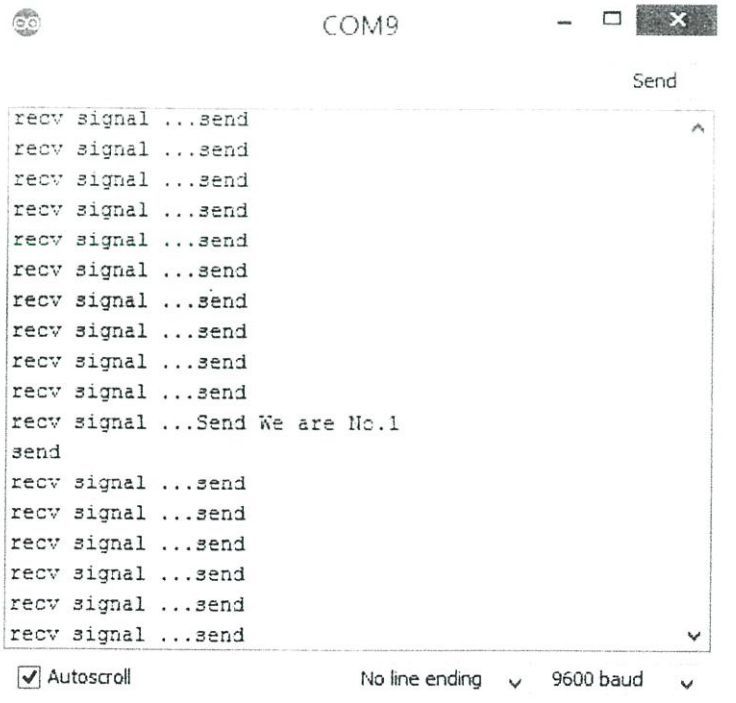
รูปที่ 4.13 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xB0 ตำแหน่งเป็นป้ายที่ 3

4.1.4 สำหรับบอกตำแหน่งป้ายชื่อ 0xC0

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 1 จะแสดงคำพูด “ We are No. 1”

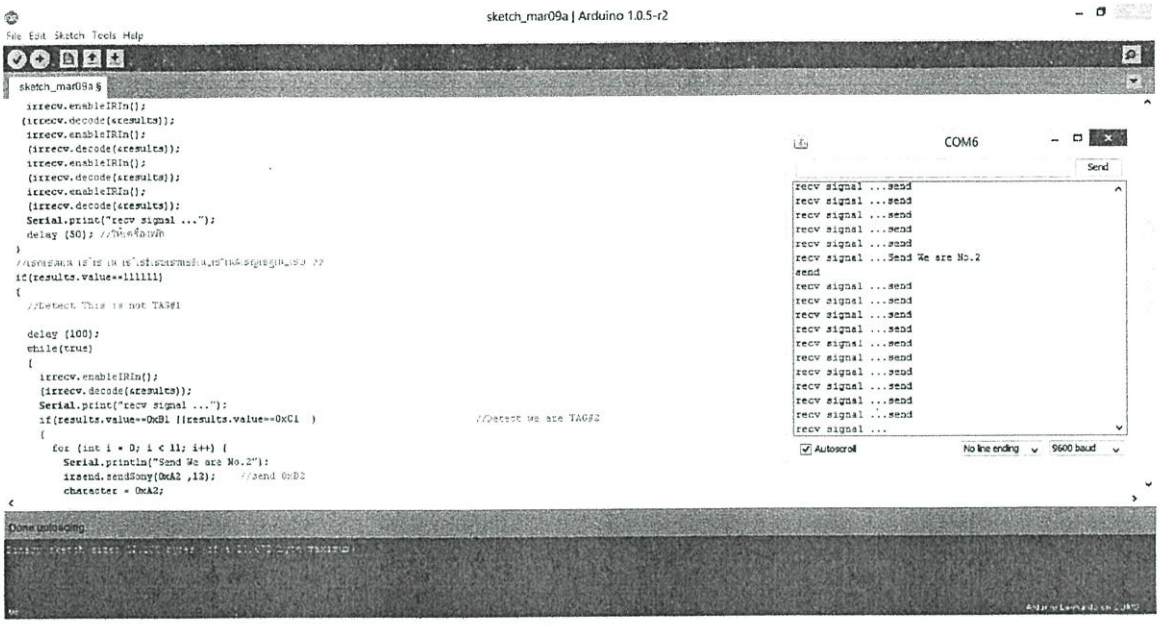


รูปที่ 4.14 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 1

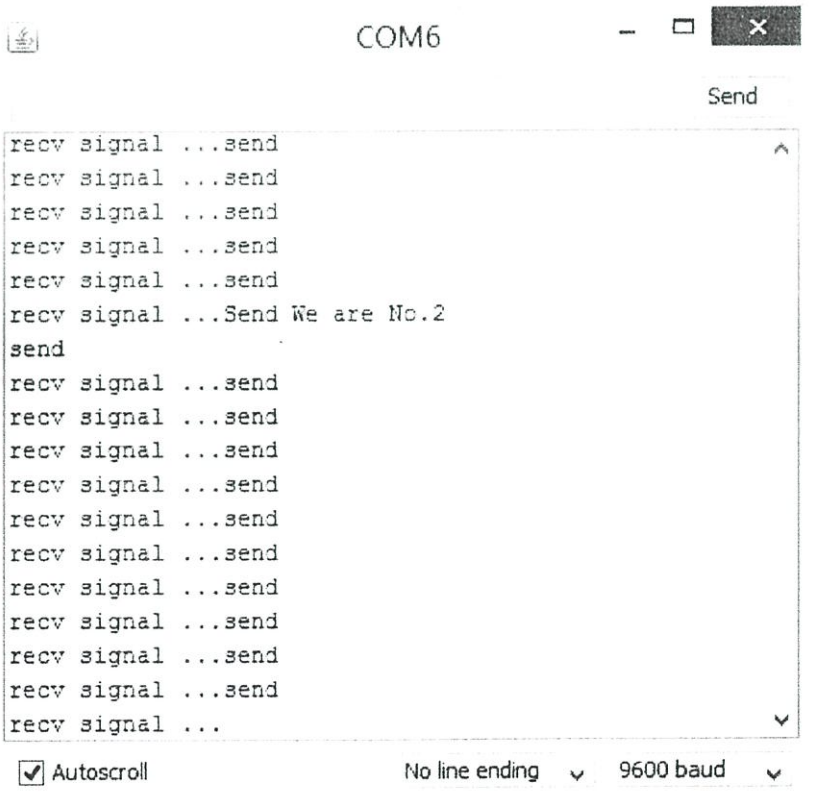


รูปที่ 4.15 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 1

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 2 จะแสดงคำพูด “ We are No. 2”

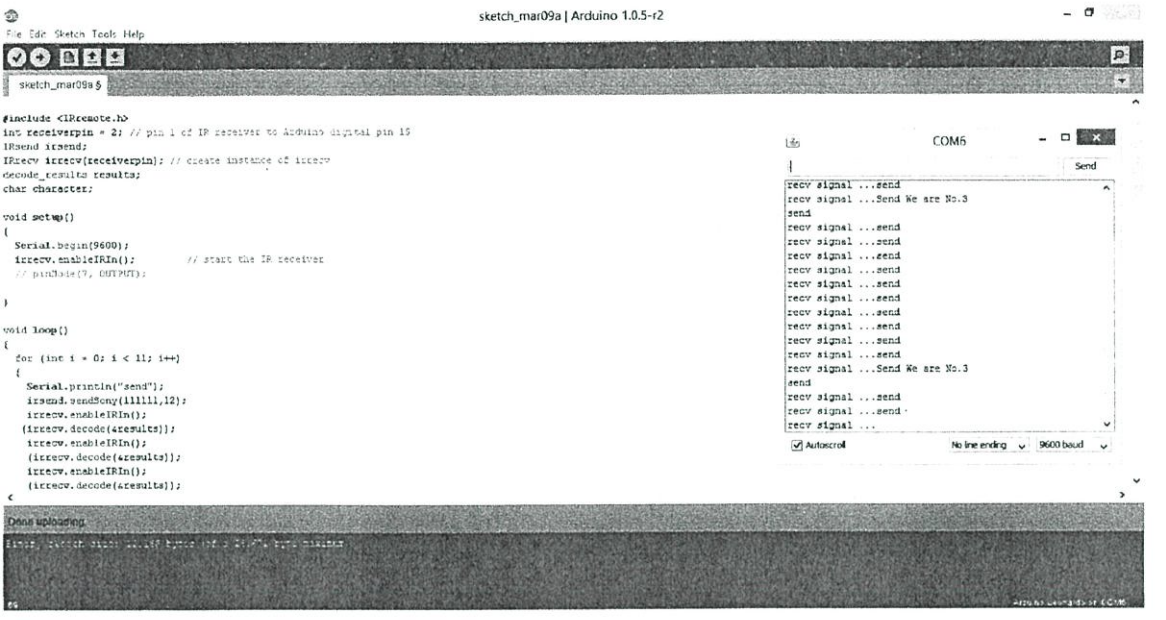


รูปที่ 4.16 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 2



รูปที่ 4.17 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 2

สำหรับการแสดงผลการระบุตำแหน่งเมื่อเป็นป้ายที่ 3 จะแสดงคำพูด “ We are No. 3”



รูปที่ 4.18 รูปส่วนโปรแกรมและการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 3

รูปที่ 4.19 รูปส่วนการแสดงผลตำแหน่งของอินฟราเรดป้าย 0xC0 ตำแหน่งเป็นป้ายที่ 3

4.2 ส่วนแสดงผลบนหลอดไดโอดเปล่งแสง (LED)

4.2.1 ส่วนของโปรแกรมที่ใช้แสดงผล

```
int letterA[8][8] = {
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  0,  1,  1,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
};

int letterB[8][8] = {
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
};

int letterC[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1 }
```

```
};

int letterD[8][8] = {
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
};


int letterE[8][8] = {
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
};

int letterF[8][8] = {
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
};
```

```
int letterG[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1 }
};

int letterH[8][8] = {
  { 1,  0,  0,  1,  1,  0,  0,  1 }
  ,
  { 1,  0,  0,  1,  1,  0,  0,  1 }
  ,
  { 1,  0,  0,  1,  1,  0,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  0,  1,  1,  0,  0,  1 }
  ,
  { 1,  0,  0,  1,  1,  0,  0,  1 }
  ,
  { 1,  0,  0,  1,  1,  0,  0,  1 }
};

int letterI[8][8] = {
  { 0,  0,  0,  0,  0,  0,  0,  0 }
  ,
  { 0,  0,  0,  0,  0,  0,  0,  0 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 0,  0,  0,  0,  0,  0,  0,  0 }
  ,
  { 0,  0,  0,  0,  0,  0,  0,  0 }
  ,
};
```

```
int letterJ[8][8] = {
  { 1,  0,  0,  0,  0,  0,  0,  0 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  0,  0,  1,  1,  1,  1 }
  ,
};

int letterK[8][8] = {
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  0 }
  ,
};


int letterL[8][8] = {
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
```

```
};

int letterM[8][8] = {
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
  { 0,  0,  1,  1,  1,  1,  0,  0 }
  ,
  { 0,  1,  0,  1,  1,  0,  1,  0 }
  ,
  { 0,  1,  1,  0,  0,  1,  1,  0 }
  ,
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
};

int letterN[8][8] = {
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
  { 0,  0,  1,  1,  1,  1,  1,  0 }
  ,
  { 0,  1,  0,  1,  1,  1,  1,  0 }
  ,
  { 0,  1,  1,  0,  1,  1,  1,  0 }
  ,
  { 0,  1,  1,  1,  0,  1,  1,  0 }
  ,
  { 0,  1,  1,  1,  1,  0,  1,  0 }
  ,
  { 0,  1,  1,  1,  1,  1,  0,  0 }
  ,
  { 0,  1,  1,  1,  1,  1,  1,  0 }
  ,
};

int letterO[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1 }
```

```
  ,
};
int letterP[8][8] = {
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
};

int letterQ[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  0 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  0 }
  ,
};

int letterR[8][8] = {
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
```

```
  ,
};

int letterS[8][8] = {
  { 1,  1,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
};

int letterT[8][8] = {
  { 0,  0,  0,  0,  0,  0,  0,  0 }
  ,
  { 0,  0,  0,  0,  0,  0,  0,  0 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  0,  1,  1,  1 }
  ,
};

int letterU[8][8] = {
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
```

```
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  1,  0,  0,  0,  0,  1,  1 }
    ,
};

int letterW[8][8] = {
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  0,  1,  0,  0,  1,  0,  1 }
    ,
    { 1,  0,  1,  0,  0,  1,  0,  1 }
    ,
    { 1,  0,  0,  1,  1,  0,  0,  1 }
    ,
};

int letterX[8][8] = {
    { 0,  1,  1,  1,  1,  1,  1,  0 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  1,  0,  1,  1,  0,  1,  1 }
    ,
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
    { 1,  1,  0,  1,  1,  0,  1,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 0,  1,  1,  1,  1,  1,  1,  0}
    ,
};


int letterY[8][8] = {
    { 0,  1,  1,  1,  1,  1,  1,  0 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  1,  0,  1,  1,  0,  1,  1 }
    ,
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
```

```
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  0,  0,  1,  1,  1}
    ,
};

int letterZ[8][8] = {
    { 0,  0,  0,  0,  0,  0,  0,  0 }
    ,
    { 1,  1,  1,  1,  1,  1,  0,  1 }
    ,
    { 1,  1,  1,  1,  1,  0,  1,  1 }
    ,
    { 1,  1,  1,  1,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  0,  1,  1,  1,  1 }
    ,
    { 1,  1,  0,  1,  1,  1,  1,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  1,  1 }
    ,
    { 0,  0,  0,  0,  0,  0,  0,  0}
    ,
};

int letter1[8][8] = {
    { 1,  1,  1,  0,  0,  1,  1,  1 }
    ,
    { 1,  1,  0,  1,  0,  1,  1,  1 }
    ,
    { 1,  0,  1,  1,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  1,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  1,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  1,  0,  1,  1,  1 }
    ,
    { 1,  1,  1,  1,  0,  1,  1,  1 }
    ,
    { 0,  0,  0,  0,  0,  0,  0,  0}
    ,
};


int letter2[8][8] = {
    { 1,  1,  0,  0,  0,  0,  1,  1 }
    ,
    { 1,  0,  1,  1,  1,  1,  0,  1 }
```

```
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  0,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  1,  1,  1,  1 }
  ,
  { 1,  1,  0,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1}
  ,
};

int letter3[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  0,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  0,  1,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1}
  ,
};

int letter4[8][8] = {
  { 1,  1,  1,  1,  1,  0,  0,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  0,  1 }
  ,
  { 1,  1,  1,  0,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1}
  ,
};


int letter5[8][8] = {
  { 1,  0,  0,  0,  0,  0,  0,  1 }
```

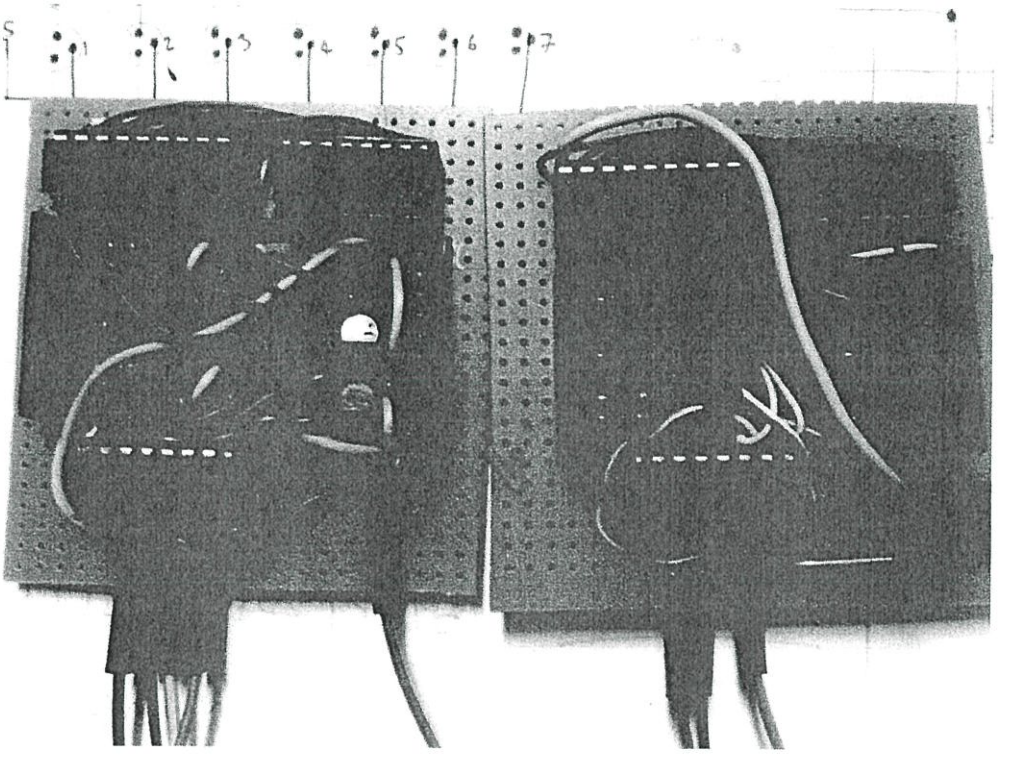
```
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1}
  ,
};


int letter6[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  1,  1 }
  ,
  { 1,  0,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1}
  ,
};


int letter7[8][8] = {
  { 1,  0,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  0,  1,  1 }
  ,
  { 1,  1,  1,  1,  0,  1,  1,  1 }
  ,
  { 1,  1,  1,  0,  1,  1,  1,  1 }
  ,
  { 1,  1,  0,  1,  1,  1,  1,  1 }
  ,
  { 1,  1,  0,  1,  1,  1,  1,  1 }
  ,
  { 1,  1,  0,  1,  1,  1,  1,  1}
  ,
};
```

```
int letter8[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1}
  ,
};

int letter9[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  1,  0,  0,  0,  0,  1,  1}
  ,
};

int letter0[8][8] = {
  { 1,  1,  0,  0,  0,  0,  1,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  1,  0,  1 }
  ,
  { 1,  0,  1,  1,  1,  0,  0,  1 }
  ,
  { 1,  0,  1,  1,  0,  1,  0,  1 }
  ,
  { 1,  0,  1,  0,  1,  1,  0,  1 }
  ,
```

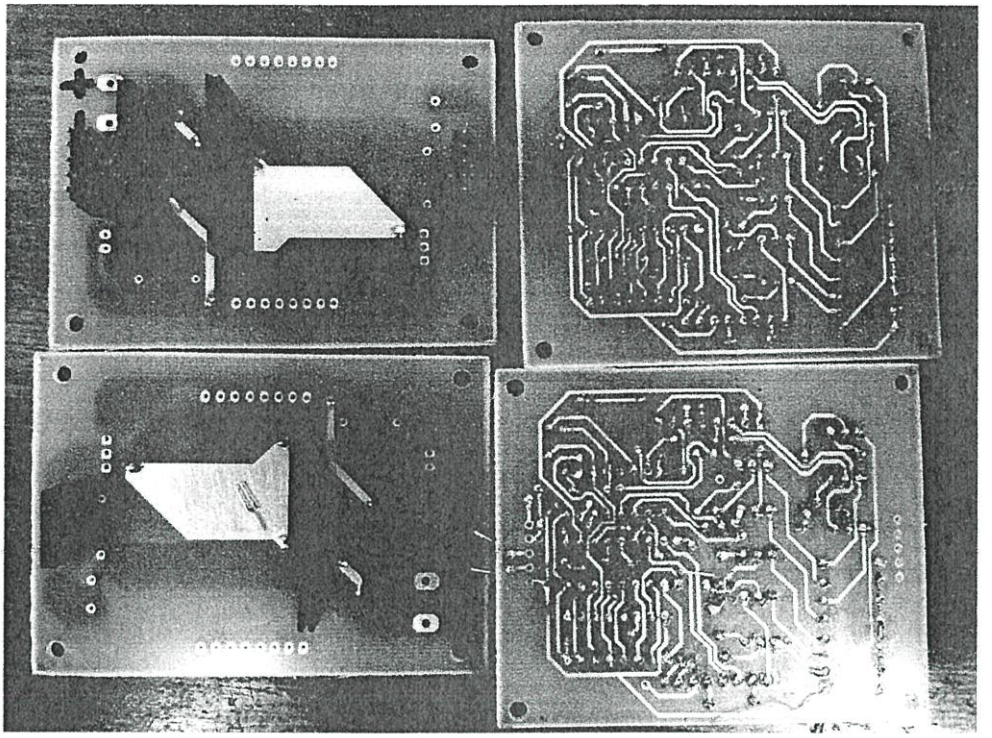
```
    { 1,   0,   0,   1,   1,   1,   0,   1 }
    ,
    { 1,   1,   0,   0,   0,   0,   1,   1 }
    ,
};
```

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

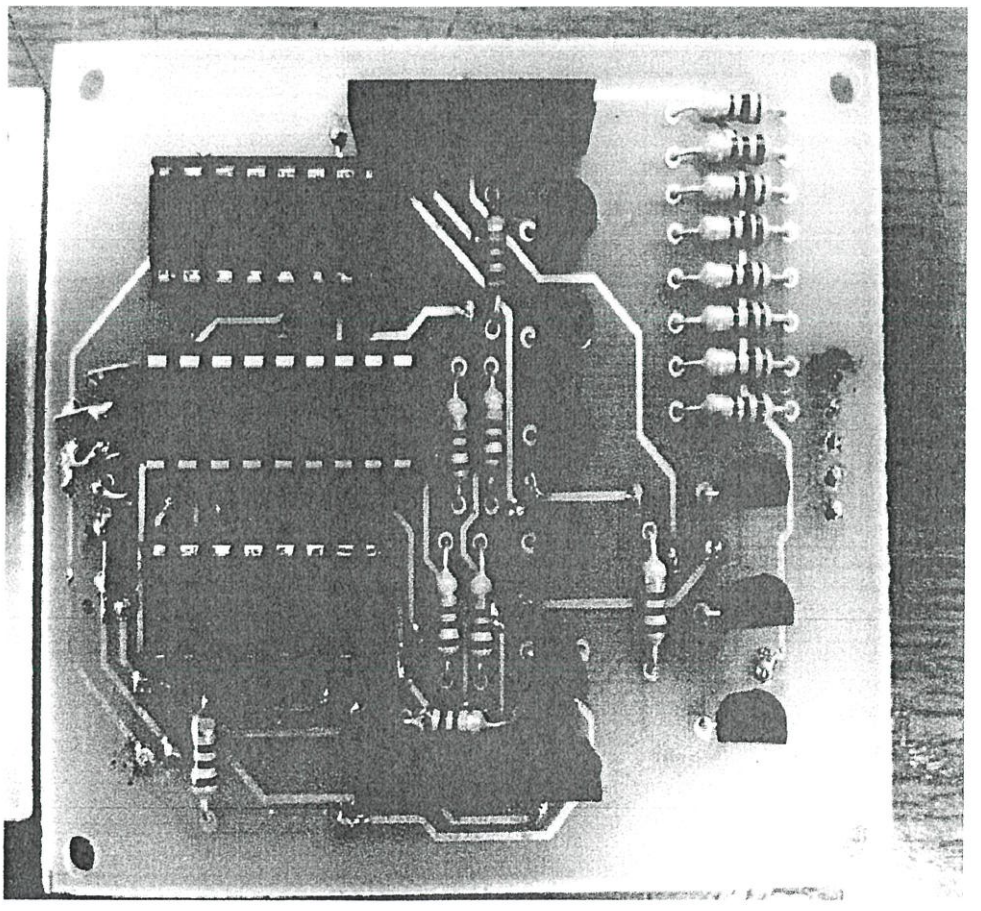
5.1 รูปแบบวงจรด๊อทเมทริกซ์ที่สร้างได้จริง



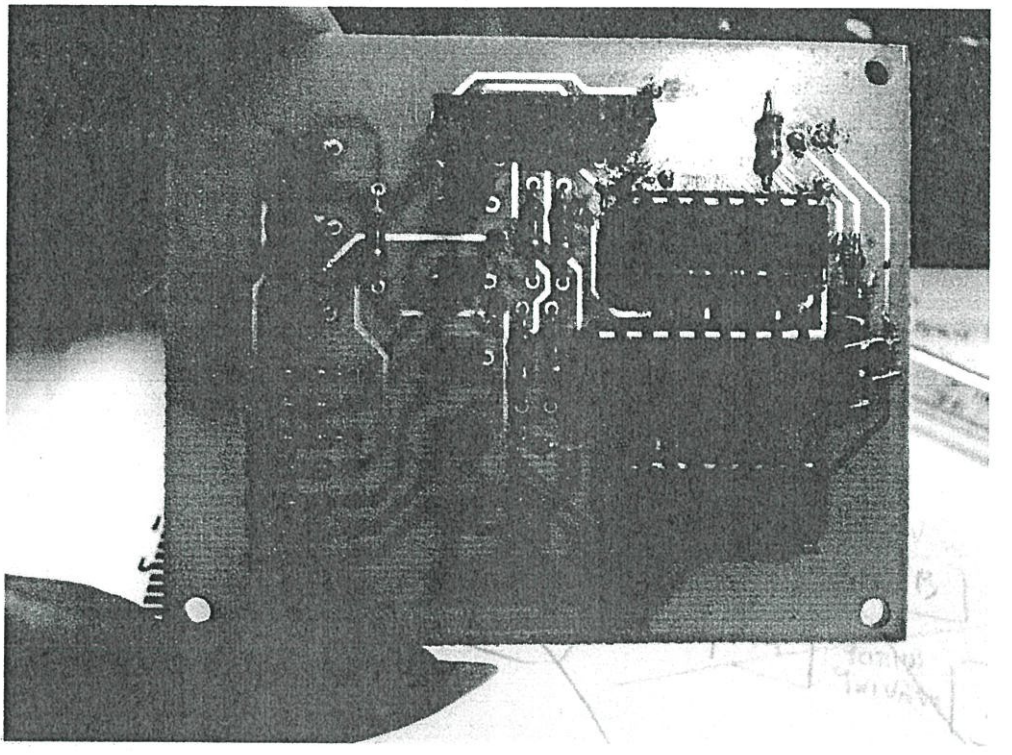
รูปที่ 5.1 รูปวงจรการต่อจริง



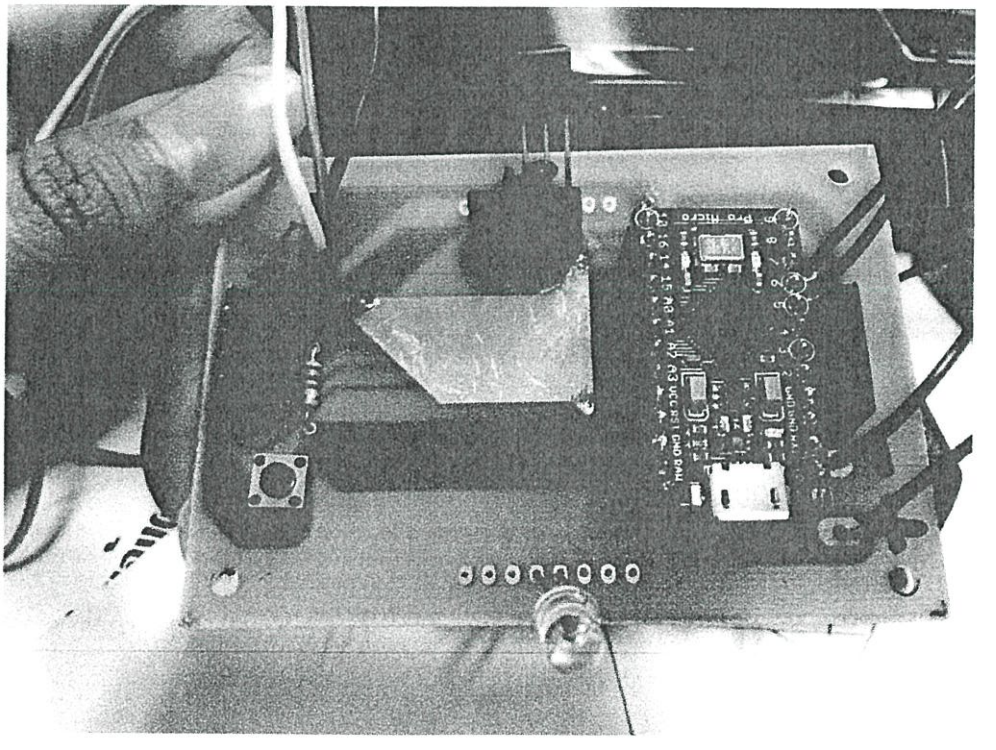
รูปที่ 5.2 รูปวงจรแผ่นลายปริ้นทั้งหมด



รูปที่ 5.3 รูปวงจรแผ่นลายปริ้นชั้นที่ 1



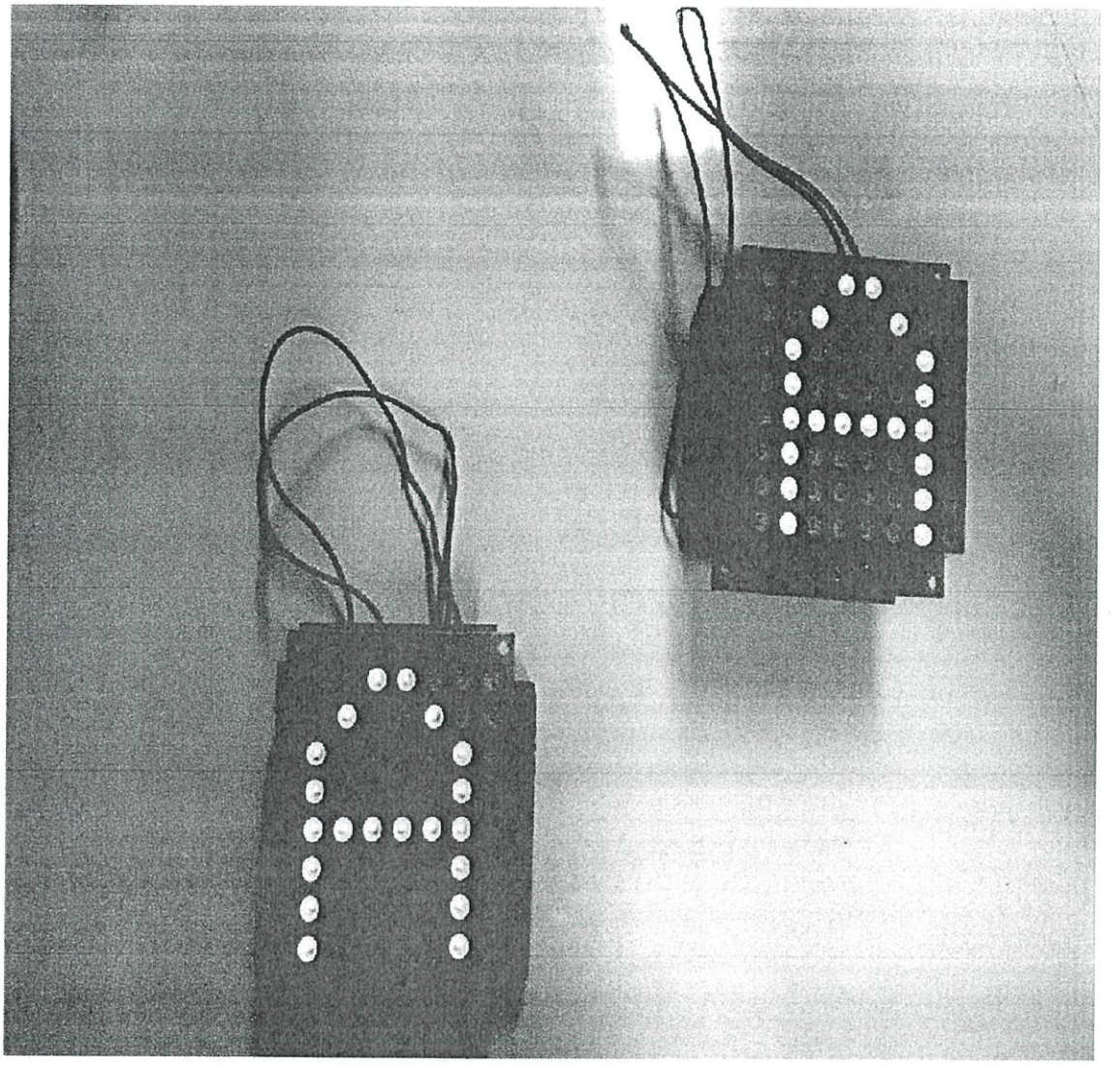
รูปที่ 5.4 รูปวงจรแผ่นลายปริ้นชั้นที่ 2



รูปที่ 5.5 รูปวงจรแผ่นลายปริ้นที่ใส่อุปกรณ์

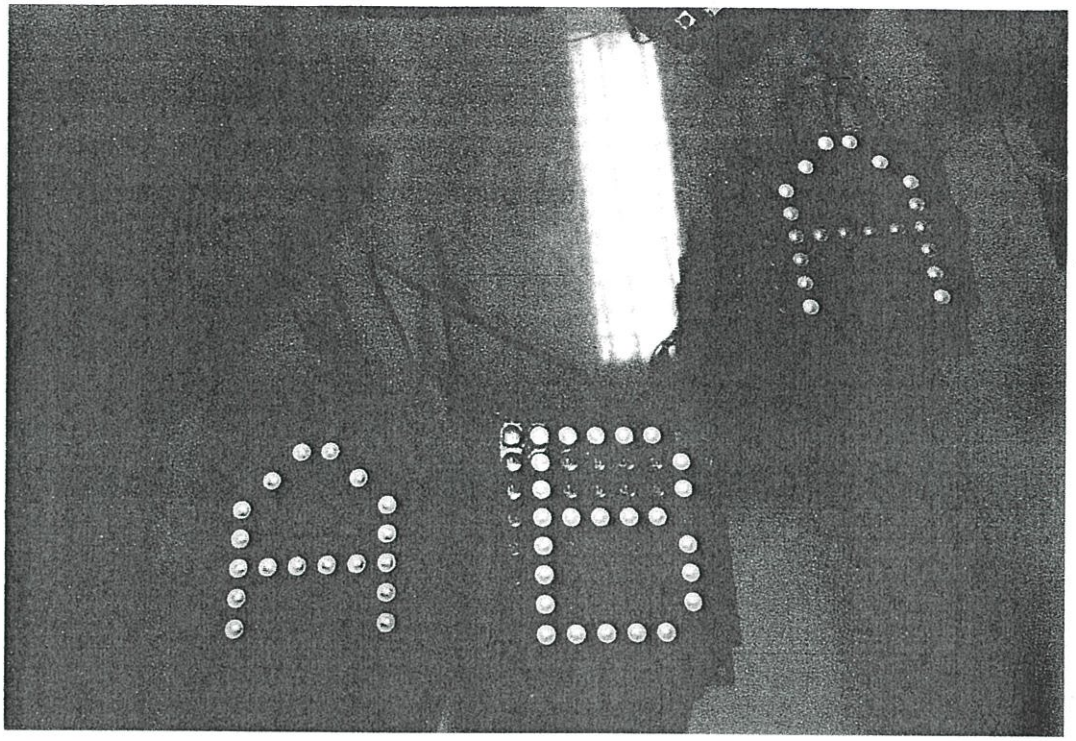
5.2 รูปแบบการแสดงผลของด๊อทเมทริกซ์

5.2.1 รูปส่วนแสดงผลบอร์ดที่ยังไม่ทำการสอดประสานกัน

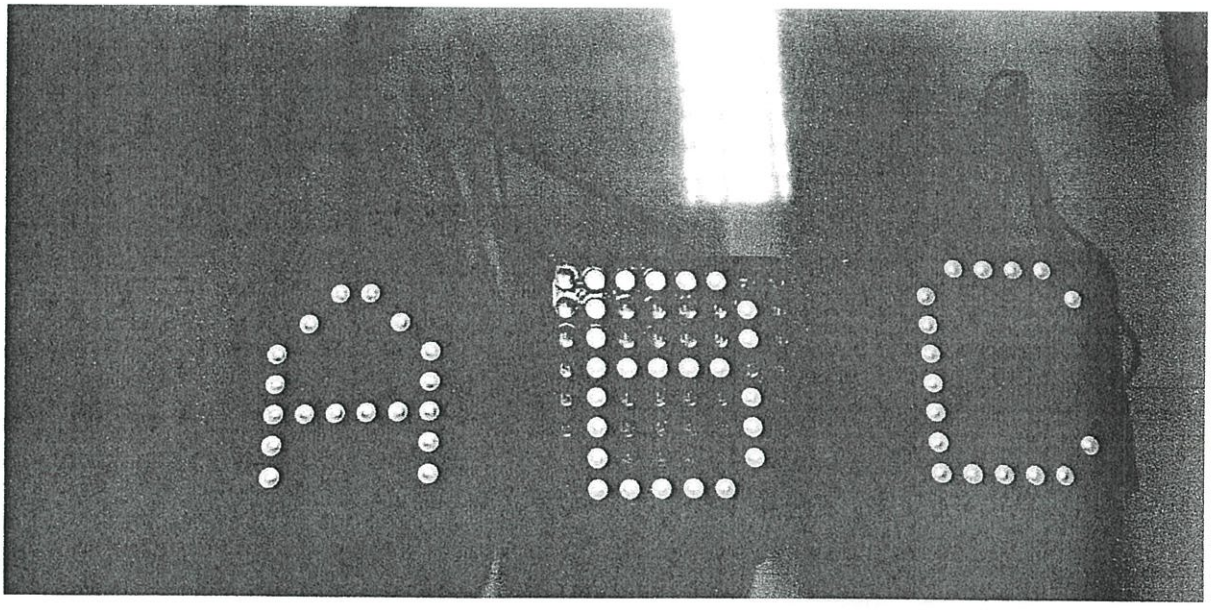


รูปที่ 5.6 รูปบอร์ดแสดงผลปกติ

5.2.2 รูปส่วนแสดงผลบอร์ดที่ทำการสอดประสานกัน



รูปที่ 5.7 รูปบอร์ดแสดงผลการสอดประสาน 2 บอร์ด



รูปที่ 5.8 รูปบอร์ดแสดงผลการสอดประสาน 3 บอร์ด

5.3 วิจารณ์ปัญหาที่พบจากการทดลอง

จากการทดลองจะพบปัญหาดังต่อไปนี้

5.3.1 ปัญหาจากวงจรอินฟราเรด

เพราะวงจรอินฟราเรดสามารถถูกแทรกซ้อนและถูกรบกวนได้ง่าย ทำให้การทดลองรับและส่งอินฟราเรดมักจถูกรบกวนได้ง่าย และเปอร์เซ็นต์ในความผิดพลาดสูง

5.3.2 ปัญหาจากวงจรหลอดไดโอดเปล่งแสง

จากวงจรมักจะพบปัญหาในการต่อวงจรที่มักจะเกิดการผิดพลาด เกิดจากสายไฟและสายดินหลวม ทำให้ความสว่างของหลอดไฟมีปัญหาในการสว่างที่อาจจะไม่เท่ากัน หรือเกิดการช๊อตทำให้แสดงผลไม่ถูกต้อง

บรรณานุกรม

[1] LED Dot Matrix Display[ออนไลน์], เข้าถึงได้จาก: http://www.instructables.com/id/LED-Dot-Matrix-Display-1/

[2]Dot matrix[ออนไลน์], เข้าถึงได้จาก: http://en.wikipedia.org/wiki/Dot_matrix

[3] การควบคุมการติด-ดับของไดโอดเปล่งแสงด้วยไอซี 74HC595[ออนไลน์], เข้าถึงได้จาก:http://www-.ee.kmutnb.ac.th/eerobot/esl/learning/index.php?article=74hc595-pwm-leds

[4] ทรานซิสเตอร์ (TRANSISTORS) [ออนไลน์], เข้าถึงได้จาก: http://www.cpe.ku.ac.th/~yuen/-204471/device/diode_transistor/transistor.htm

[5]Infrared Remote control (IR)[ออนไลน์], เข้าถึงได้จาก:http://www.thaimicrotron.com/CCS-628/EXAM/TVRemote.htm

[6]Arduino Mega 2560 R3[ออนไลน์], เข้าถึงได้จาก:http://arduino.cc/en/Main/arduinoBoard-Mega2560#.UylibKh_sts

[7]การมอดูเลตสัญญาณ (Signal Modulation)[ออนไลน์], เข้าถึงได้จาก:http://tanathep014.blog-spot.com/2009/11/signal-modulation-2-1-modulation.html

[8]ตัวต้านทาน (RESISTOR)[ออนไลน์], เข้าถึงได้จาก:http://school.obec.go.th/tkp/sci/kl_web-/m3/resistor.html

[9]การอ่านค่าตัวต้านทานบอกแถบสี[ออนไลน์], เข้าถึงได้จาก:http://www.elecnet.chandra.ac.th-/courses/ELTC1203/resistor/colorcoderesistor.htm

[10]กฎของโอห์ม[ออนไลน์], เข้าถึงได้จาก:http://www.elecnet.chandra.ac.th/courses/ELTC12-03/electricbasic/ohm.htm