ป้ายประชาสัมพันธ์ไร้สาย

WIRELESS BOARD

ชานนท์ บุญสุขานนท์
CHANON BOONSUKANON
ฐาปนันท์ หนองหลวง
THAPANAN NONGLUANG

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

ป้ายประชาสัมพันธ์ไร้สาย

WIRELESS BOARD

ชานนท์ บุญสุขานนท์
CHANON BOONSUKANON
ฐาปนันท์ หนองหลวง
THAPANAN NONGLUANG

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2557

WIRELESS BOARD

CHANON BOONSUKANON
THAPANAN NONGLUANG

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONTKUT'S OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2014

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	ป้ายประชาสัมพันธ์ไร้สาย
Thesis Title	WIRELESS BOARD
ชื่อนักศึกษา	นายชานนท์ บุญสุขานนท์
	นายฐาปนันท์ หนองหลวง
ระดับปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมสารสนเทศ
ปริญญานิพนธ์ปีการศึกษา	2557

(.....)
ผศ.ไพศาล สิทธิโยภาสกุล
อาจารย์ที่ปรึกษาปริญญานิพนธ์

(.....)
อ.สรพงษ์ วชิรรัตนพรกุล
อาจารย์ที่ปรึกษาปริญญานิพนธ์

หัวข้อปริญญานิพนธ์	ป้ายประชาสัมพันธ์ไร้สาย		
Thesis Title	WIRELESS BOARD		
ชื่อนักศึกษา	นายชานนท์ บุญสุขานนท์	รหัสนักศึกษา	54010324
	นายฐาปนันท์ หนองหลวง	รหัสนักศึกษา	54011355
ระดับปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมสารสนเทศ		
ปีการศึกษา	2557		
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ไพศาล สิทธิโยภาสกุล		
อาจารย์ที่ปรึกษาปริญญานิพนธ์ร่วม	อ.สรพงษ์ วชิรรัตนพรกุล		

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นโครงงานการประยุกต์ใช้อุปกรณ์ไมโครคอนโทรลเลอร์กับเทคโนโลยีไร้สายมาใช้งานให้เกิดประโยชน์ในด้านของการสื่อสารและการประชาสัมพันธ์เพื่อให้ง่ายต่อการสื่อสารกับคนจำนวนมากและสามารถเข้าใจไปในทิศทางเดียวกันได้เป็นอย่างดีซึ่งสิ่งนี้แต่เดิมมีการใช้งานกันอยู่แล้วแต่ยังไม่ได้รับความสะดวกในการใช้งานเท่าที่ควร

ด้วยเหตุนี้ทางผู้จัดทำจึงได้ประยุกต์สร้างโครงงานดังกล่าวขึ้นเพื่อใช้งานการส่งข้อมูลแบบไร้สายมารวมกับความสามารถของไมโครคอนโทรลเลอร์ซึ่งจะต่อเข้ากับป้ายประชาสัมพันธ์เพื่อให้สามารถเข้าใจในการทำงานและคำสั่งของไมโครคอนโทรลเลอร์มากขึ้นโดยผู้ใช้งานนั้นสามารถที่จะพิมพ์ข้อความใดๆที่ต้องการให้ปรากฏบนป้ายได้ทันทีและความสะดวกของการสั่งงานคือใช้การติดต่อกันระหว่างป้ายกับคอมพิวเตอร์เป็นแบบไร้สายทำให้เราไม่ต้องคอยกังวลเรื่องสายสัญญาณที่เชื่อมต่อกันอีกด้วย

Thesis Title	Wireless Board		
Student	Mr.Chanon Boonsukanon	Student ID.	54010324
	Mr.Thapanan Nongluang	Student ID.	54011355
Degree	Bachelor of Engineering		
Program	Information Engineering		
Academic Year	2557		
Thesis Advisor	Asst.Prof.Dr.Paisarn Sittiyopaskul		
Thesis Co-Advisor	Mr.Sorapong Wachirarattanapornkul		

ABSTRACT

The purpose of this project is focus on using Microcontroller device with Wireless LAN technology for working out in the fields of communication and public relations board to make it easier to communicate with many people and can be understand for working as well. And the method has originally in usage but has not been easy to work as it should.The proposed design of the project was created a framework for such applications to use wireless data transmission combined with the capabilities of a microcontroller and then connected with the public relation. This method that using by the user can input the message into the public relation board on the real times and convenient for connecting between the board and client computer by using wireless technology that has less problem about the cable signal lose.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ.ไพศาล สิทธิโยภาสกุล และ อ.สรพงษ์ วชิรรัตนพรกุลที่ท่านอาจารย์คอยให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประการณ์ที่ดีแก่ผู้จัดทำ ทางผู้จัดทำขอขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอบพระคุณคุณพ่อ คุณแม่ และครอบครัวที่คอยอยู่เบื้องหลังในการเรียน คอยให้กำลังใจแก่ผู้จัดทำ และสนับสนุนในการเรียน จนทุกอย่างสำเร็จไปได้ด้วยดีตลอดมา และสิ่งหนึ่งที่ขาดไม่ได้คือผู้จัดทำต้องขอบคุณเพื่อนๆทุกคนที่คอยช่วยเหลือ ให้คำแนะนำ ช่วยแก้ปัญหา และคอยให้กำลังใจกันและกัน จนทำให้โครงงานออกมาสำเร็จลุล่วงไปได้ด้วยดี

ผู้จัดทำ
ชานนท์ บุญสุขานนท์
ฐาปนันท์ หนองหลวง

สารบัญ

สารบัญ(ต่อ)

สารบัญรูป

หน้า

สารบัญรูป(ต่อ)

หน้า

สารบัญรูป(ต่อ)

หน้า

สารบัญตาราง

หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

เนื่องจากปัจจุบันนี้เทคโนโลยีทางด้านสารสนเทศเข้ามามีบทบาทเป็นอย่างมากในชีวิตประจำวัน เพราะปัจจุบันเป็นโลกของข้อมูลข่าวสารที่เป็นกันหนึ่งเดียวทั่วโลก ที่โดยเฉพาะอย่างยิ่งด้านธุรกิจและการโฆษณา ดังนั้นในปัจจุบันจึงต้องการสื่อที่สามารถแสดงข้อมูลข่าวสารต่างๆ เพื่อให้สามารถดึงดูดความสนใจและสามารถสื่อสารให้ทุกคนได้เข้าใจในข้อมูลข่าวสารนั้นๆ กระดานหรือแผ่นป้ายแสดงข่าวสาร (Display Board) เป็นอีกสื่อหนึ่งที่เป็นที่นิยมใช้กันมากในปัจจุบันซึ่งสามารถพบเห็นได้ตามแหล่งชุมชน ย่านธุรกิจ ศูนย์การค้า โรงพยาบาลและบริษัทห้างร้านต่างๆ ในอดีตยังคงใช้ระบบการแสดงข้อมูลเป็นแบบ เซเวนเซกเมนต์ (7-Segment) และต่อมาจึงมีการพัฒนาเป็นแบบจุดแสดงผล (Dot Matrix Display) ซึ่งจะมีความละเอียดของภาพมากกว่าแบบเดิม การแสดงผลจะแสดงเป็นอักษรภาษาอังกฤษและตัวเลข นอกจากนี้แล้วปัจจุบันป้ายโฆษณาที่มีลักษณะเป็นตัวอักษรเลื่อนก็เป็นที่นิยม ทำให้สามารถดึงดูดความสนใจได้จากผู้พบเห็นได้มากกว่าป้ายโฆษณาทั่วๆ ไป

1.2 ภาพรวม หรือโครงสร้างรวมของโครงงาน

เป็นป้ายโฆษณาประชาสัมพันธ์ ที่รับข้อมูลในการแสดงผลจากคอมพิวเตอร์และตัวแป้นพิมพ์ โดยข้อมูลจะถูกป้อนมาจากโปรแกรมคอมพิวเตอร์ผ่านการส่งข้อมูลแบบไร้สายมาที่ป้ายโฆษณา ซึ่งแสดงผลเป็นแบบ Matrix โดยใช้หลอดไฟ LED เป็นตัวแสดงผลและใช้ไมโครคอนโทรลเลอร์ตระกูล Arduino เป็นตัวควบคุมการทำงานในการแปลงค่าที่ได้รับจากโปรแกรมคอมพิวเตอร์ เพื่อแสดงออกมาเป็นข้อความบนป้ายLEDโดยเลื่อนจากขวาไปซ้าย

องค์ประกอบในการทำโครงงานชิ้นนี้จะแบ่งเป็นสองส่วนได้แก่

1. ฮาร์ดแวร์ (Hardware)
   - LED Dot Matrix ขนาด4x4 จำนวน320 ตัว
   - บอร์ดขนาดใหญ่สำหรับ LED Dot Matrix 2 ตัว
   - nRF24L01p Module สื่อสารไร้สาย 2 ตัว
   - Arduino UNO R3 1 ตัว
   - Arduino DUE 1 ตัว
   - Arduino Mega2560 1 ตัว
   - คอมพิวเตอร์ 1 ตัว
2. ซอฟต์แวร์ (Software)
   - โปรแกรม Visual Basic 2010 เพื่อเขียนโปรแกรมบนคอมพิวเตอร์
   - โปรแกรม Arduino IDE เพื่อเขียนคำสั่งลงบน Arduino

1.3 วัตถุประสงค์ของโครงงาน

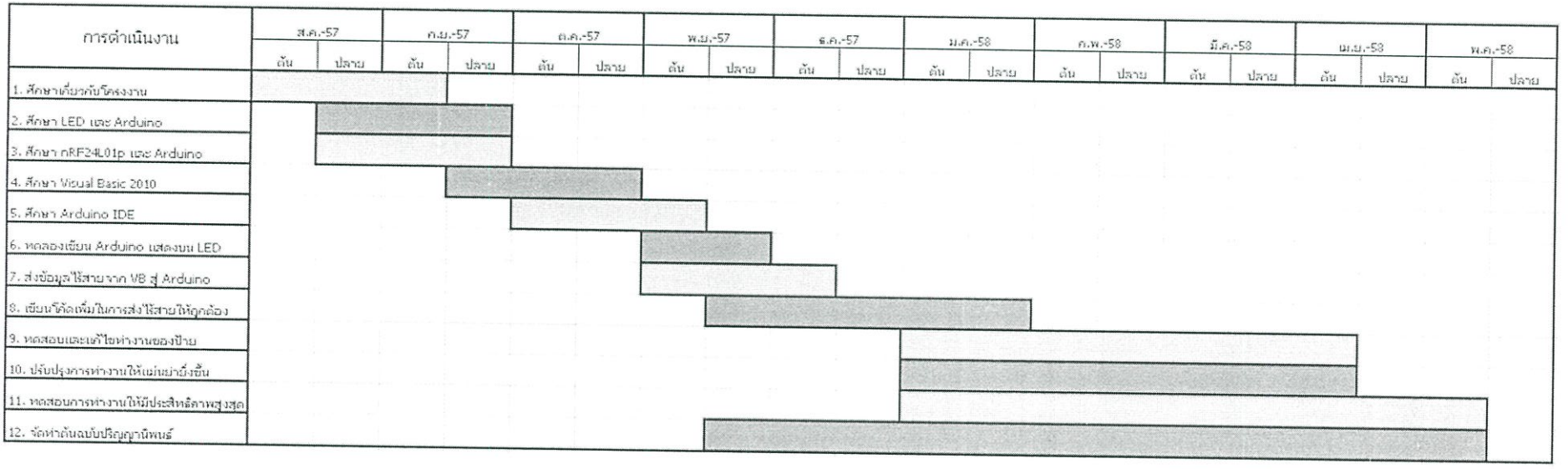
1. เพื่อศึกษาหลักการทำงานของหลอดLED
2. เพื่อศึกษาการทำงานของเทคโนโลยีไร้สาย
3. เพื่อนำความรู้ทางไมโครคอนโทรลเลอร์ และเทคโนโลยีLED มาใช้ในการประชาสัมพันธ์
4. สามารถนำไปใช้ได้การประชาสัมพันธ์ได้จริง

1.4 ขั้นตอนการดำเนินงานโครงงาน

1. ศึกษาเกี่ยวกับโครงงานโดยรวม
2. ศึกษาการประยุกต์ใช้หลอดไฟLED กับ Arduino ในการแสดงผลแบบ Matrix
3. ศึกษาวิธีการรับส่งแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ nRF24L01p Moduleและ Arduino
4. ศึกษาการใช้งานโปรแกรม Visual Basic 2010
5. ศึกษาการใช้งานโปรแกรม Arduino IDE
6. ทดลองเขียนคำสั่งลงบอร์ด Arduino เพื่อแสดงผลบน LED
7. ทดลองการส่งข้อมูลไร้สายจากโปรแกรม VisualBasic สู่ตัว Arduino
8. เขียนโค้ดเพิ่มเติมในการแก้ไขการใช้งานไร้สายให้ถูกต้อง
9. ทดสอบและแก้ไขรูปแบบการทำงานของป้าย
10. ปรับปรุงการทำงานของอุปกรณ์ให้สามารถสื่อสารกันได้อย่างแม่นยำยิ่งขึ้น
11. ทดสอบการทำงานของระบบให้มีประสิทธิภาพสูงสุดและสามารถใช้งานได้สะดวก
12. จัดทำต้นฉบับปริญญานิพนธ์

1.5 แผนผัง หรือตารางเวลาการดำเนินงานโครงงาน

ตารางที่ 1.1 ตารางระยะเวลาการดำเนินงาน



บทที่ 2
ทฤษฎีพื้นฐาน

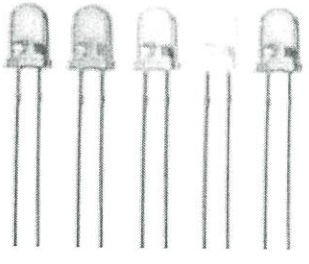
2.1 ไดโอดเปล่งแสง

2.1.1 ประวัติความเป็นมา

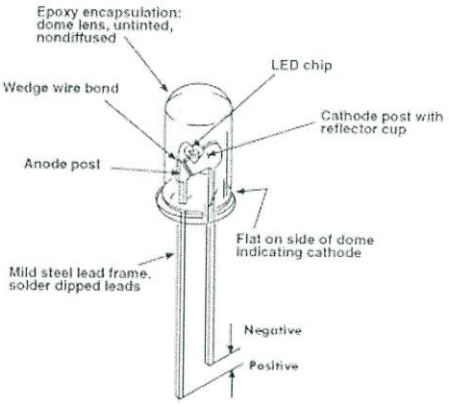
ความเป็นมาของหลอด LED (Light Emitting Diode) หรือไดโอดเปล่งแสง เริ่มมาจากจุดกำเนิดคือเป็นไดโอดก่อน ซึ่งไดโอดนั้นเป็นอุปกรณ์อิเล็กทรอนิกส์ชนิดสองขั้ว ที่ออกแบบและมีการควบคุมทิศทางการไหลของประจุไฟฟ้า โดยยอมให้กระแสไฟฟ้าไหลในทิศทางเดียว และกั้นการไหลในทิศทางตรงกันข้าม เมื่อกล่าวถึงไดโอด มักจะหมายถึงไดโอดที่ทำมาจากสารกึ่งตัวนำ (semiconductor diode) ซึ่งก็คือผลึกของสารกึ่งตัวนำที่ต่อกันได้ทางขั้วไฟฟ้าทั้งสองขั้ว

ส่วนหลอด LED หรืออาจเรียกว่า solid-state lighting (SSL) เป็นอุปกรณ์สารกึ่งตัวนำประเภทหนึ่ง จัดอยู่ในจำพวกไดโอดที่สามารถเปล่งแสงในช่วงสเปกตรัมแคบ ในรูปของอิเล็กโตรลูมิเนสเซนต์ (electroluminescence) สีของแสงที่เปล่งออกมานั้นขึ้นอยู่กับองค์ประกอบทางเคมีของวัสดุกึ่งตัวนำที่ใช้ และเปล่งแสงได้ใกล้ช่วงอัลตราไวโอเล็ต (ultraviolet) ช่วงแสงที่มองเห็น (visible light) และช่วงอินฟราเรด (infrared) ผู้พัฒนาไดโอดเปล่งแสงขึ้นเป็นคนแรก คือ นิก โฮโลยัก (Nick Holonyak Jr.) แห่งบริษัทเจเนรัลอิเล็กทริก (General Electric Company) โดยได้พัฒนาไดโอดเปล่งแสงในช่วงแสงสีแดงที่มองเห็น และสามารถใช้งานได้ในเชิงปฏิบัติเป็นครั้งแรก เมื่อ ค.ศ. 1962 จนกระทั่งช่วงทศวรรษที่ 1970 จอร์จ คราฟอร์ด (George Craford) จึงได้คิดค้น LED สีเหลือง (amber) ขึ้นเป็นครั้งแรกและได้พัฒนาความสว่างของ LED สีแดงและสีแดงอมส้มด้วย

ในช่วงแรกๆ นั้นหลอด LED ใช้เป็นตัวบ่งบอกสัญญาณ (indicator light) ในการทำงานของอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์ เช่น นาฬิกา เครื่องคิดเลข รีโมทคอนโทรล และกระติกน้ำร้อน เป็นต้น เพราะตัวหลอด LED มีขนาดเล็กจิ๋ว และใช้กระแสไฟฟ้าน้อยมากเมื่อเทียบกับปริมาณแสงที่ออกมา ทำให้ในเวลาต่อมามีผู้พัฒนาหลอด LED อย่างต่อเนื่อง จากแรกเริ่มที่ให้สีโทนร้อน คือ สีแดง ส้ม เหลือง ต่อมาได้มีการคิดค้นวิธีการสร้างหลอดที่ให้สีโทนเย็น คือ สีเขียวและน้ำเงินและได้แสงขาวโทนเย็นขึ้น จึงมีการนำมาใช้งานทดแทนหลอดไฟฟ้าชนิดอื่นอย่างจริงจัง ทั้งที่ใช้เป็นแสงขาวโทนสีต่างๆ รวมทั้งใช้เป็นไฟเปลี่ยนสีจากการผสมสี RGB



รูปที่ 2.1 แสดงหลอดไฟ LED



รูปที่ 2.2 แสดงโครงสร้างภายในของหลอด LED

2.1.2 หลักการทำงานของหลอด LED

เมื่อเปิดสวิตช์ไฟ กระแสไฟฟ้าจะผ่านอุปกรณ์เพื่อแปลงไฟฟ้ากระแสสลับให้เป็นไฟฟ้ากระแสตรง และเปลี่ยนจากความต่างศักย์ไฟฟ้าสูงไปสู่ความต่างศักย์ไฟฟ้าที่ค่อนข้างต่ำ ประมาณ 2.5-3 โวลต์ แล้วจึงจ่ายเข้าตัวชิปของหลอด LED ซึ่งมีเพียงตัวนำแคโทด (Cathod) และแอโนด (Anode) เท่านั้น โดยหลอด LED จะมีกระแสไฟฟ้าไหลผ่านน้อยมาก ประมาณ 20 มิลลิแอมป์ ในตัวชิปของ LED ประกอบด้วยสารกึ่งตัวนำขั้วประจุบวกชนิด P (Positively Changed Material) ที่อยู่ห่างจากสารกึ่งตัวนำขั้วประจุลบชนิด N (Negatively Changed Material) เล็กน้อย จุดนี้เรียกว่ารอยต่อ (Junction) เมื่อปล่อยกระแสไฟฟ้าผ่านหลอด LED ตัวนำแอโนดจะไปดันขั้วประจุบวก และตัวนำแคโทดไปดันขั้วประจุลบให้มาชนกัน เมื่อประจุบวกและประจุลบมาชนกันที่รอยต่อของสารกึ่งตัวนำทั้งสองชนิดจับตัวกันและคายพลังงานออกมาในรูปของแสงสว่าง ซึ่งเรียกว่า “อิเล็กโตรลูมิเนสเซนต์” ทำให้เกิดแสงสว่างที่บริเวณด้านหน้าตัวหลอด ซึ่งมีอุณหภูมิในการทำงานที่ประมาณ 25 องศาเซลเซียส ถ้าอุณหภูมิสูงเกินไป แสงสว่างที่ออกมาจะลดลงแสงจากหลอด LED มีลักษณะพุ่งออกในทิศทางเดียว แต่ในกรณีที่ต้องการให้แสงกระจายออกในมุมแคบหรือกว้างเพิ่มขึ้น ก็จะใช้อุปกรณ์ครอบหลอด LED ในลักษณะของเลนส์ (Package) ไว้เพื่อบังคับทิศทางของการกระจายแสง หลอด LED สามารถเปิดปิดได้ทันที ไม่ต้องใช้ระยะเวลาในการจุดติดเหมือนหลอดไส้ที่ต้องเผาไส้หลอด หรือหลอดดิสชาร์จที่ต้องปรับแรงดันก๊าซภายใน หลอด LED สามารถปรับความเข้มของแสงได้ด้วยอุปกรณ์หรี่ไฟ (Dimmer) โดยขึ้นอยู่กับรุ่นและอุปกรณ์ควบคุมซึ่งจะต้องตรวจสอบให้แน่ใจก่อนเลือกใช้

2.1.3 ข้อดีของหลอด LED

มีประสิทธิภาพการให้แสงสว่างสูง และทิศทางแสงสว่างของ LED จะส่องไปเฉพาะด้านหน้าเท่านั้น ซึ่งจะลดการสูญเปล่าของแสงสว่าง

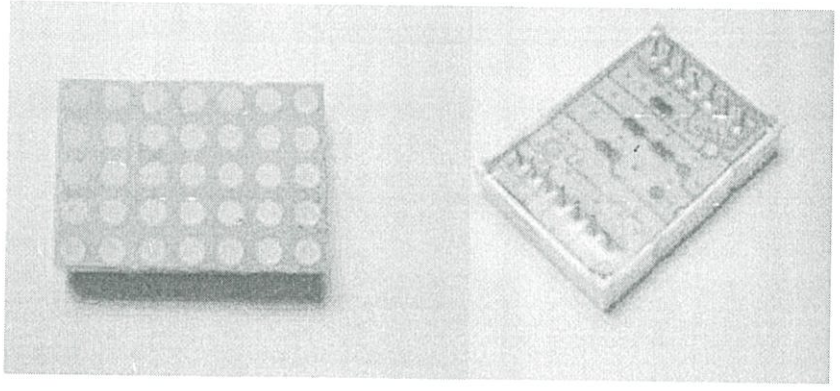
- ใช้พลังงานน้อย
- การดูแลรักษาต่ำ

- น้ำหนักเบา,ขนาดเล็ก
- อายุการใช้งานยาวนานถึง 100,000 ชั่วโมง
- สามารถเปิดปิดได้บ่อยครั้ง และเมื่อเปิดจะให้แสงสว่างโดยทันที
- ทนต่อการสั่นสะเทือนและแรงกระแทก จึงเหมาะสมสำหรับติดตั้งในเครื่องบินหรือรถยนต์
- ปล่อยความร้อนออกมาน้อยมาก ทำให้ลดการสูญเสียพลังงานไฟฟ้าในส่วนเครื่องปรับอากาศ
- สามารถควบคุมคุณภาพของแสงที่ปล่อยออกมาได้จึงสามารถนำไปใช้ให้แสงสว่างในบางสถานที่ได้

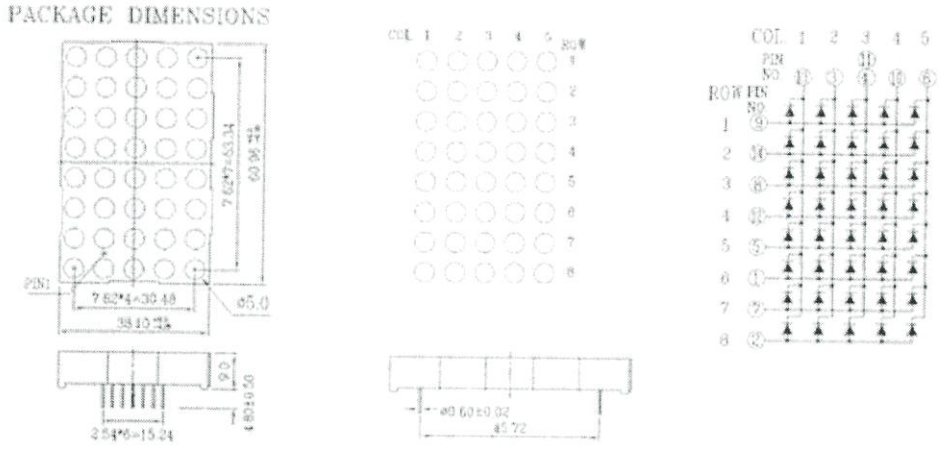
เช่น การให้แสงสว่างกับภาพเขียน เนื่องจากสามารถควบคุมแสงสว่างจาก LED ไม่ให้มีส่วนผสมของแสงที่เป็นอันตรายต่อภาพเขียน เช่น แสงอินฟาเรด และแสงอุลตร้าไวโอเลท

2.1.4 ดอทเมตริกซ์

ดอทเมตริกซ์ (Dot Matrix) เป็นอุปกรณ์แสดงผล แบบเดียวกับ LED ครับ คือว่ากันไปแล้วคือการนำเอา LED หลายตัวมาต่อเรียงกัน เป็นหลัก เป็นแถว ซึ่งเราจะเห็นการใช้งานดอทเมตริกซ์ในการทำป้ายไฟวิ่ง โดยเราจะนำเอาดอทเมตริกซ์ หลายๆตัวมาต่อกัน แล้วเขียนโปแกรมผ่านไมโครคอนโทลเลอร์ หรือ คอมพิวเตอร์ผ่านวงจรขับดอทเมตริกซ์



รูปที่ 2.3 แสดงลักษณะแผ่น LED ดอทเมตริกซ์



รูปที่ 2.4 แสดงลักษณะและการต่อภายในของชิป LED ดอทเมตริกซ์

คุณสมบัติของดอทเมตริกซ์โดยเฉลี่ย

- กำลังวัตต์ / จุด 75W
- กระแสใช้งาน 20mA ต่อ จุด
- แรงดันใช้งาน 5V

2.2 การขับ LED ด้วยวิธีแบบสแตติกและแบบมัลติเพล็กซ์

2.2.1 แบบสแตติก (Static Drive)

เป็นวิธีที่ง่ายโดยการขับ LED แต่ละดวงจะแยกอิสระจากกัน โดยการต่อตัวต้านทานตัวหนึ่งเข้ากับแหล่งจ่ายแรงดัน หรือแหล่งจ่ายกระแสคงที่ มาทำการป้อนไบแอสตรง ให้กับ LED ซึ่ง LED ก็จะติดสว่างได้ตามต้องการเราเรียกว่า สแตติค (Static) เพราะว่ามีกระแสไหลผ่าน LED อย่างต่อเนื่องตลอดเวลา เหมาะสำหรับการขับ LED ไม่กี่ตัว LED ทั่วไปมีความสว่างปกติต้องการกระแสประมาณ 2 mA ซึ่งเอาต์พุตของไมโครคอนโทรลเลอร์ส่วนใหญ่จะสามารถขับกระแสให้กับ LED ได้โดยตรง

2.2.2 แบบมัลติเพล็กซ์ (Multiplex Drive) หรือการขับแบบพัลส์ (Pulse Drive)

เป็นการลดความยุ่งยากของวงจร โดยการป้อนกระแสไฟฟ้าไบแอสให้กับ LED ในแต่ละเซกเมนต์เพียงกลุ่มหนึ่ง(เพื่อให้แสดงผลสมบูรณ์ในหน่วยนั้นๆ) ในแต่ละช่วงเวลาสั้นๆจึงค่อยเลื่อนไปไบแอสให้กับ LED เซกเมนต์ในกลุ่มต่อไป อย่างไรก็ตามการไบแอสแบบให้ดวงไฟติดสว่างในระยะเวลาสั้นๆ ต้องมีคาบในการติดสว่างซ้ำ (Repeatition Rate) ที่รวดเร็วเพียงพอจนสายตามนุษย์มองเห็นเหมือนแสงสว่างนั้นติดตลอดเวลา ประโยชน์ อีกข้อของจากขับ LED แบบ Multiplex คือการปรับปรุงเรื่องความเข้มแสง ในการแสดงผลของจอแสดงผลในขณะที่สิ้นเปลืองพลังงานโดยเฉลี่ย ต่ำกว่าหรือเทียบเท่ากัน

2.3 การสแกน

การสแกนหมายถึง จำนวนเส้นการสแกนต่อหนึ่งภาพ และจำนวนที่ส่งออกไปต่อวินาที ถ้าเราส่งที่จำนวนต่อวินาทีมากมายเท่าไรการกระพริบของภาพก็จะลดลงเท่านั้น หน้าที่ของการสแกน คือการเลือก LED ที่สว่างให้ได้ภาพตามชัดเจนที่ต้องการ

ภาพที่เราเห็นหรือตัวอักษรต่างๆในป้ายโฆษณานั้น จะประกรอบไปด้วยจุดเล็กๆจำนวนหนึ่งที่ทำให้เกิดขึ้นโดยเส้นแนวนอนและแนวตั้ง ตาคนเราจะมองเห็น เอาต์พุต LED เป็นแสงต่อเนื่องได้จะต้องป้อนกระแส พลัส์มีความถี่มากกว่า 30 Hz แต่ถ้าเราต้องการให้เห็นเป็นภาพต่อเนื่อง หรือตัวอักษรวิ่งที่ไม่สามารถสังเกตเห็นการกระพริบได้เลย เราควรป้อนพลัส์ ที่มีความถี่ประมาณ 50-60 Hz จึงจะไม่เห็นภาพเกิดการพลิ้ว

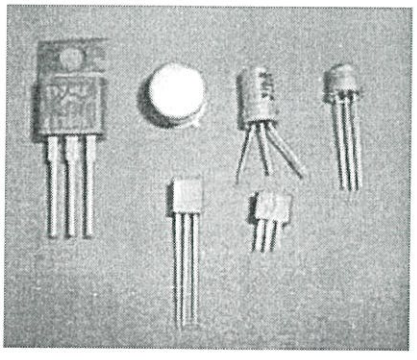
2.3.1 การสแกนทางแนวตั้ง (Column Scan)

การสแกนทางแถวแนวตั้งจะทำการส่งข้อมูลออกไปทางแถวแนวนอน โดยส่งข้อมูลตัวที่หนึ่งออกไปแล้วให้แถวแนวตั้งที่หนึ่งแสดงผล จากนั้นก็ทำการส่งข้อมูลตัวที่สองออกไป แล้วให้แถวแนวตั้งที่สองแสดงผล ทำเช่นนี้ไปจนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกแถวแนวตั้ง ก็จะเป็นการสแกนครบหนึ่งรอบดังนั้นถ้าจำนวนหลักที่จะแสดงผลออกมาเป็นตัวอักษร ที่มีจำนวนหลายหลัก วิธีนี้ไม่เหมาะสมที่จะนำมาใช้งาน เพราะว่าเมื่อให้ LED ในแถวแนวตั้งที่หนึ่งติดกว่าที่ LED ที่แถวแนวตั้งสุดท้ายจะติดต้องใช้เวลานาน

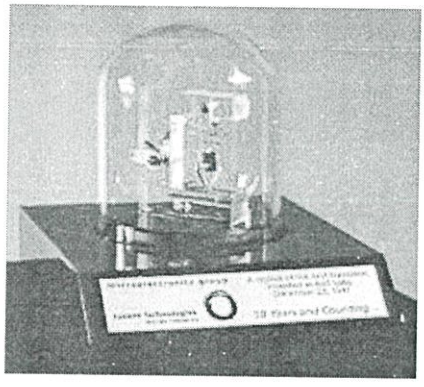
2.3.2 การสแกนทางแนวนอน (Row Scan)

การสแกนทางแถวแนวนอนจะทำการส่งข้อมูลออกไปครบทุกหลักก่อนแล้วให้แถวแนวนอนที่หนึ่งแสดงผลจากนั้นก็ทำการส่งข้อมูลชุดถัดไปออกไปจนครบหมดทุกหลัก แล้วให้แถวแนวนอนที่สองแสดงผล ทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปจนครบหมดทุกแถวแนวนอน ก็จะเป็นการสแกนครบหนึ่งรอบ วิธีนี้มีข้อดีคือ สามารถแสดงผลเป็นตัวอักษรพร้อมกันได้หลายหลักและถ้าจัดเวลาให้เหมาะสมแล้ว เวลาทำการสแกนจะไม่เกิดอาการส่ายของภาพ แต่ก็มีข้อเสียคือการเขียนโปรแกรมควบคุมให้ตัวอักษรเลื่อนทำให้ยากกว่าแบบแรก

2.4 ทรานซิสเตอร์



รูปที่ 2.5 แสดงทรานซิสเตอร์แบบต่างๆ



รูปที่ 2.6 แสดงแบบจำลองของทรานซิสเตอร์ที่ทำงานได้ครั้งแรก

2.4.1 ทรานซิสเตอร์ คืออะไร

ทรานซิสเตอร์ (Transistor) เป็นอุปกรณ์สารกึ่งตัวนำที่สามารถควบคุมการไหลของอิเล็กตรอนได้ ใช้ทำหน้าที่ ขยายสัญญาณไฟฟ้า, เปิด/ปิดสัญญาณไฟฟ้า, ควบคุมแรงดันไฟฟ้าให้คงที่, หรือกล้ำสัญญาณไฟฟ้า (Modulate) เป็นต้น การทำงานของทรานซิสเตอร์เปรียบได้กับวาล์วควบคุมที่ทำงานด้วยสัญญาณไฟฟ้าที่ขาเข้า เพื่อปรับขนาดกระแสไฟฟ้าขาออกที่จ่ายมาจากแหล่งจ่ายไฟทรานซิสเตอร์ประกอบด้วยวัสดุเซมิคอนดักเตอร์ที่มีอย่างน้อยสามขั้วไฟฟ้าเพื่อเชื่อมต่อกับวงจร ภายนอก แรงดันหรือกระแสไฟฟ้าที่ป้อนให้กับขั้วทรานซิสเตอร์หนึ่งคู่ จะมีผลให้เกิดการเปลี่ยนแปลงในกระแสที่ไหลผ่านในขั้วทรานซิสเตอร์อีกคู่หนึ่ง เนื่องจากพลังงานที่ถูกควบคุม (เอาต์พุต) จะสูงกว่าพลังงานที่ใช้ในการควบคุม (อินพุท) ทรานซิสเตอร์จึงสามารถขยายสัญญาณได้ ปัจจุบัน บางทรานซิสเตอร์ถูกประกอบขึ้นมาต่างหาก แต่ยังมีอีกมากที่พบฝังอยู่ใน แผงวงจรรวม ทรานซิสเตอร์เป็นการสร้างบล็อกพื้นฐานของอุปกรณ์อิเล็กทรอนิกส์ที่ทันสมัย และเป็นที่แพร่หลายในระบบอิเล็กทรอนิกส์สมัยใหม่. หลังจากถูกพัฒนาขึ้นในช่วงต้นทศวรรษที่ 1950, ทรานซิสเตอร์ได้ปฏิวัติสาขาอิเล็กทรอนิกส์และปูทางสำหรับวิทยุ เครื่องคิดเลข และคอมพิวเตอร์ ให้มีขนาดเล็กลงและราคาที่ถูกกว่า

2.4.2 การทำงานของ ทรานซิสเตอร์

ประโยชน์ที่สำคัญของทรานซิสเตอร์มาจากความสามารถในการใช้สัญญาณขนาดเล็กที่ป้อนให้ระหว่างขั้วไฟฟ้าคู่หนึ่ง เพื่อควบคุมสัญญาณที่มีขนาดใหญ่กว่ามากที่อีกคู่หนึ่งของขั้วไฟฟ้า คุณสมบัติแบบนี้ถูกเรียกว่าอัตราการขยาย (Gain) (สามารถคำนวณได้จากนำสัญญาณเอาต์พุต หารด้วยอินพุต ถ้าได้ผลลัพธ์มากกว่า 1 แสดงว่าวงจรนั้นเป็นวงจรขยาย) ทรานซิสเตอร์สามารถควบคุมสัญญาณเอาต์พุตให้เป็นสัดส่วนกับสัญญาณอินพุต นั่นคือมันสามารถทำหน้าที่เป็นเครื่องขยาย หรืออีกแบบหนึ่ง ทรานซิสเตอร์สามารถใช้ในการเปิดหรือปิดกระแสในวงจร(สวิตช์)ควบคุมระบบไฟฟ้าที่ปริมาณ ของกระแสไฟฟ้าจะถูกกำหนดโดยองค์ประกอบวงจรอื่น ๆ

ทรานซิสเตอร์มีสองประเภท ซึ่งมีความแตกต่างกันเล็กน้อยในวิธีการที่ พวกมันจะถูกใช้ในวงจร แบบแรกเป็น ทรานซิสเตอร์สองขั้วที่มีขา เบส, คอลเล็กเตอร์ และอิมิตเตอร์ กระแสขนาดเล็ก ที่ ขาเบส (ที่ไหลระหว่าง เบส กับ อิมิตเตอร์) สามารถควบคุม หรือ สวิตช์ กระแสที่มีขนาดใหญ่มากที่ไหลระหว่าง คอลเล็กเตอร์ กับ อิมิตเตอร์. สำหรับทรานซิสเตอร์ ฟิลด์-เอฟเฟ็กต์ ขาจะมีป้ายกำกับเป็น เกท, ซอส และ เดรน แรงดันไฟฟ้าที่เกทสามารถควบคุมกระแส ระหว่างซอส และ เดรน

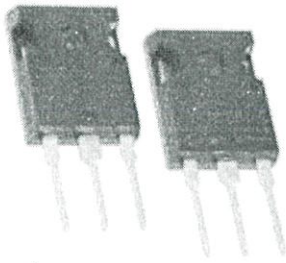
2.4.2.1 ทรานซิสเตอร์เป็นสวิตช์

BJT ใช้เป็นสวิตช์อิเล็กทรอนิกส์ในรูปแบบอิมิตเตอร์ ลงกราวนด์ทรานซิสเตอร์ถูกใช้กันทั่วไปให้ทำหน้าที่เป็นสวิตช์อิเล็กทรอนิกส์, ทั้งสำหรับการใช้งาน พลังงานสูง เช่น Switched-Mode Power Supplies และ สำหรับการใช้งานพลังงานต่ำ เช่น ลอจิกเกตในวงจร ทรานซิสเตอร์แบบอิมิตเตอร์ ลงกราวด์ เป็นวงจรสวิตช์ไฟแสงสว่างที่ในสถานะปกติจะ OFF หลอดไฟก็จะปิด เมื่อแรงดันไฟฟ้าที่เบส สูงขึ้น, กระแสอิมิตเตอร์ และ คอลเล็กเตอร์ (I_{CE}) เพิ่มขึ้นแบบเอ็ดโพเนนเชียลจนอิ่มตัว (Saturate) แรงดันที่คอลเล็กเตอร์ จะลดลงเข้าใกล้อิมิตเตอร์ (หรือใกล้ศูนย์) กระแส I_{CE} จะไหลผ่านโหลดเต็มที่ ซึ่งในวงจรนี้คือหลอดไฟ ทำให้หลอดไฟ "เปิด" เราจึงเรียกสถานะของสวิตช์ในขณะนี้ว่า ON การให้กระแสที่เบส (I_{BE}) อย่างเพียงพอเป็นปัญหาที่สำคัญในการใช้ทรานซิสเตอร์ให้ทำงานเป็นสวิตช์. ทรานซิสเตอร์ให้เกน เป็นกระแส จึงได้กระแสค่อนข้างมากที่คอลเล็กเตอร์ ที่จะถูกสลับ โดยกระแสที่มีขนาดเล็กในเบสอัตราส่วนของกระแสเหล่านี้แตกต่างกันไป ขึ้นอยู่กับชนิด ของทรานซิสเตอร์และแม้กระทั่งทรานซิสเตอร์ประเภทเดียวกันก็แตกต่างกัน ขึ้นอยู่กับกระแสในคอลเล็กเตอร์ จะมีตัวต้านทานที่ต้องเลือกให้มีขนาดที่ให้กระแสที่เบส มีเพียงพอ เพื่อให้แน่ใจว่าทรานซิสเตอร์จะทำงานอิ่มตัวในวงจรสวิตช์ใดๆ ค่าของแรงดันไฟฟ้า อินพุต จะถูกจ่ายให้มีขนาดที่จะทำให้ได้ เอาต์พุต เป็น OFF หรือ ON โดยสมบูรณ์ ทรานซิสเตอร์จึงจะทำหน้าที่เป็นสวิตช์ที่ดีและการทำงานแบบนี้ เป็นเรื่องธรรมดาใน วงจรดิจิตอลที่ต้องการเพียง "OFF" และ "ON" เท่านั้น

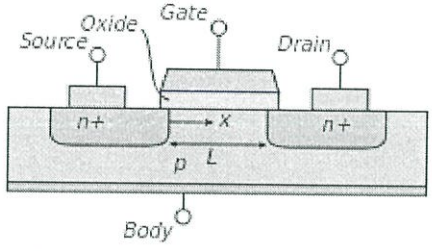
2.4.2.2 ทรานซิสเตอร์เป็นตัวขยายสัญญาณ

เครื่องขยายแบบอีมิตเตอร์ร่วม (Common-Emitter) ได้รับการออกแบบเพื่อที่ว่าการเปลี่ยนแปลงเล็กๆ ใน แรงดันไฟฟ้า (Vin) ทำให้เกิดการเปลี่ยนแปลงเล็กๆ ในกระแสเบสของทรานซิสเตอร์; การขยายกระแสของทรานซิสเตอร์รวมกับคุณสมบัติของวงจรทำให้การเปลี่ยนแปลงขนาดเล็กของ V_{in} ทำให้เกิดการเปลี่ยนแปลงขนาดใหญ่ของ V_{out} วงจรขยายด้วยทรานซิสเตอร์ตัวเดียวมีรูปแบบหลายอย่าง มีทั้งแบบขยายกระแส หรือแบบขยาย แรงดันไฟฟ้า หรือทั้งสองแบบตั้งแต่โทรศัพท์มือถือไปยังโทรทัศน์ ผลิตภัณฑ์จำนวนมากรวมทั้งเครื่องขยายเสียง เครื่องส่งวิทยุและเครื่องประมวลสัญญาณ เครื่องขยายสัญญาณเสียงด้วยทรานซิสเตอร์เครื่องแรกให้กำลังไม่กี่ร้อยมิลลิวัตต์ แต่กำลังและความชัดเจนของเสียงค่อยๆ เพิ่มขึ้น เมื่อทรานซิสเตอร์ที่ดีกว่าถูกผลิตขึ้น และสถาปัตยกรรมเครื่องขยายได้รับการพัฒนาขึ้นเครื่องขยายเสียงทรานซิสเตอร์ที่ทันสมัยที่มีกำลังเป็นร้อยวัตต์ขึ้นไป เป็นเรื่องธรรมดาและราคาก็ไม่แพงนัก

2.4.3 มอสเฟต



รูปที่ 2.7 แสดงลักษณะมอสเฟต



รูปที่ 2.8 แสดงโครงสร้างของมอสเฟต

มอสเฟต (Metal–Oxide–Semiconductor Field-Effect Transistor : MOSFET) เป็นทรานซิสเตอร์ ที่ใช้อิทธิพลสนามไฟฟ้าในการควบคุมสัญญาณไฟฟ้า โดยใช้ออกไซด์ของโลหะในการทำส่วน GATE นิยมใช้ในวงจรดิจิตอล โดยนำไปสร้างลอจิกเกตต่างๆ เพราะมีขนาดเล็ก

โครงสร้างของมอสเฟตประกอบด้วย 3 ส่วนคือ

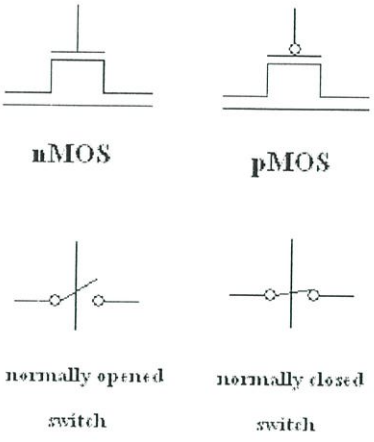
- SOURCE เป็นส่วนขาเข้าของสัญญาณ
- DRAIN เป็นส่วนขาออกของสัญญาณ
- GATE เป็นส่วนที่ทำมาจากออกไซด์ของโลหะ โดยสร้างให้เกิดความต่างศักย์ตกคร่อมระหว่างแผ่นสองแผ่นเพื่อ สร้างสนามไฟฟ้าเพื่อควบคุมการเข้าออกของสัญญาณไฟฟ้า

มอสเฟตแบ่งเป็น 2 ประเภท ได้แก่

- NMOS (Negative MOSFET) เป็นทรานซิสเตอร์ประเภท NPN เมื่อมีความต่างศักย์เป็นบวก (สนามไฟฟ้าแรง) สัญญาณไฟฟ้าจึงจะไหลจากซอส ไป เดรน ได้
- PMOS (Positive MOSFET) เป็นทรานซิสเตอร์ประเภท PNP เมื่อมีความต่างศักย์ต่ำหรือเป็นลบ (สนามไฟฟ้าอ่อน) สัญญาณไฟฟ้าจึงจะไหลจาก ซอส ไป เดรน ได้

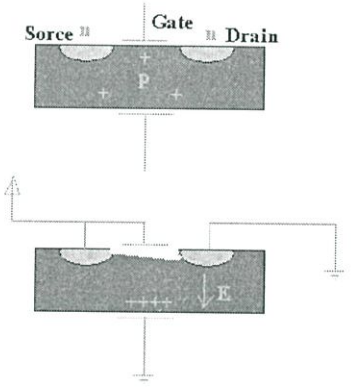
2.4.3.1 สัญลักษณ์แทน MOSFET

MOSFET ในทางดิจิตอลถูกมองว่าเป็นสวิตซ์ โดย NMOS จะเป็นสวิตซ์ที่เมื่อสัญญาณเข้าเป็น "1" สวิตซ์ก็จะปิด ถ้าไม่สวิตซ์ก็ยังเปิดอยู่ (Normal Opened Switch) ส่วน PMOS จะเป็นสวิตซ์ที่เมื่อสัญญาณเข้าเป็น "1" สวิตซ์ก็จะเปิด ถ้าไม่สวิตซ์ก็จะปิดอยู่ (Normal Closed Switch) และสัญลักษณ์ทั่วไปจะมีสามขา ขากลางเป็น เกทส่วนอีกสองขาคือซอส และเดรน โดยใช้ใน NMOS เป็นหลักเพื่อสื่อสัญลักษณ์เดียวกับทรานซิสเตอร์ทั่วไปคือ ไฟขาเบส ไหล ขาคอลเล็กเตอร์จะต่อกับอิมิตเตอร์



รูปที่ 2.9 แสดงสัญลักษณ์แทนมอสเฟต

2.4.3.2 การทำงานของมอสเฟต

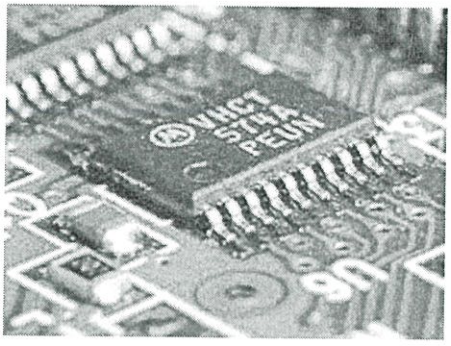


รูปที่ 2.10 แสดงสัญลักษณ์แทนมอสเฟต

NMOS เมื่อปล่อยความต่างศักย์สูง จะเกิดสนามไฟฟ้าในทิศลงอย่างแรง โฮลใน p-type จะถูกผลักลงมาอยู่ด้านล่าง (ตามรูปที่ประกอบข้างบน) ประกอบกับมีอิเล็กตรอนอิสระบางส่วนถูกดูดขึ้นไปด้านบน ส่งผลให้บริเวณด้านบนมีอิเล็กตรอนอิสระมากจนเป็น n-type ได้เรียกว่า channel สัญญาณไฟฟ้าก็จะไหลผ่านช่วง channel นี้ซึ่งเป็น n-type เหมือนกับเดรน และซอส ได้โดยใช้อิเล็กตรอนอิสระเป็นพาหะ

PMOS จะทำงานกลับกับ NMOS โดยเมื่อปล่อยความต่างศักย์ต่ำ (โดยมากมักจะติดลบ) จะเกิดสนามไฟฟ้าในทิศขึ้นอย่างแรง อิเล็กตรอนอิสระใน n-type จะถูกผลักลงมาอยู่ด้านล่าง ประกอบกับมีโฮลบางส่วนถูกดูดขึ้นไปด้านบน ส่งผลให้บริเวณด้านบนมีโฮลมากจนเป็น p-type ได้เรียกว่า Channel สัญญาณไฟฟ้าก็จะไหลผ่านช่วง Channel นี้ซึ่งเป็น p-type เหมือนกับ เดรน และ ซอส ได้โดยใช้โฮลเป็นพาหะ

2.5 วงจรรวม



รูปที่ 2.11 แสดงลักษณะวงจรรวม

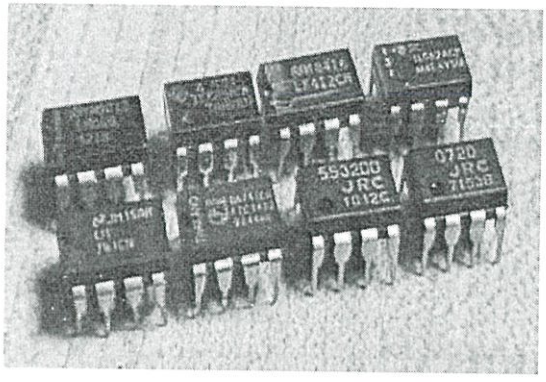
วงจรรวม หรือ วงจรเบ็ดเสร็จ (Integrated Circuit : IC) หมายถึง วงจรที่นำเอาไดโอด , ทรานซิสเตอร์ตัวต้านทาน, ตัวเก็บประจุ และองค์ประกอบวงจรต่าง ๆ มาประกอบรวมกันบนแผ่นวงจรขนาดเล็ก ในปัจจุบันแผ่นวงจรนี้จะทำด้วยแผ่นซิลิคอน บางทีอาจเรียก ชิป (Chip) และสร้างองค์ประกอบวงจรต่าง ๆ ฝังอยู่บนแผ่นผลึกนี้ ส่วนใหญ่เป็นชนิดที่เรียกว่า โมโนลิธิค (Monolithic) การสร้างองค์ประกอบวงจรบนผิวผลึกนี้ จะใช้กรรมวิธีทางด้านการถ่ายภาพอย่างละเอียด ผสมกับขบวนการทางเคมีทำให้ลายวงจรมีความละเอียดสูงมาก สามารถบรรจุองค์ประกอบวงจรได้จำนวนมาก ภายในไอซีจะมีส่วนของลอจิกมากมาย ในบรรดาวงจรเบ็ดเสร็จที่ซับซ้อนสูง เช่น ไมโครโปรเซสเซอร์ ที่ใช้ทำงานควบคุม คอมพิวเตอร์ จนถึงโทรศัพท์มือถือ แม้กระทั่งเตาอบไมโครเวฟแบบดิจิทัล สำหรับชิปหน่วยความจำ (RAM) เป็นอีกประเภทหนึ่งของวงจรเบ็ดเสร็จ ที่มีความสำคัญมากในยุคปัจจุบันไอซีกำเนิดขึ้นโดย Geoffrey W.A. Dummer นักวิทยาศาสตร์เรดาร์จากอังกฤษ ต่อมาได้ย้ายไปทำการค้นคว้าต่อที่สหรัฐอเมริกา โดยสามารถสร้างไอซีจากเซรามิกส์ตัวแรกได้ในปี ค.ศ. 1956 แต่ยังไม่ประสบผลสำเร็จนัก ต่อมาในปี ค.ศ. 1957 กองทัพสหรัฐอเมริกานำโดยแจ็ก คิลบี ได้ทำการค้นคว้าทดลองต่อ ในวันที่ 6 กุมภาพันธ์ ค.ศ. 1959 คิลบี ได้จดสิทธิบัตรไอซีที่ทำจากเจอร์มาเนียม และในพัฒนาการสุดท้ายของไอซี โรเบิร์ต นอยซ์ได้จดสิทธิบัตรไอซีที่ทำจากซิลิคอน ในวันที่ 25 เมษายน ค.ศ. 1961

2.5.1 ประเภทของไอซี

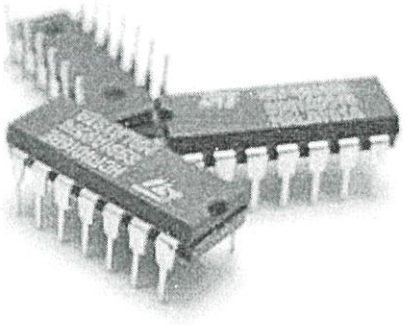
ประเภทของไอซีนั้นแบ่งตามจำนวนขนาดของเกท จำนวนของเกทต่อไอซีจะกำหนดประเภทของไอซี (IC) 1 เกท เท่ากับชิ้นส่วนอิเล็กทรอนิกส์ 1 ชิ้น

- ขนาด SSI (Small Scale Integration) จะมีตั้งแต่ 1 ถึง 10 เกท
- ขนาด MSI (Medium Scale Integration) จะมีตั้งแต่ 10 ถึง 100 เกท
- ขนาด LSI (Large Scale Integration) จะมีตั้งแต่ 100 ถึง 10,000 เกท
- ขนาด VLSI (Very Large Scale Integration) จะมีตั้งแต่ 100,000 ถึง 10,000,000 เกท

- ขนาด ULSI (Ultra-Large Scale Integration) จะมีตั้งแต่ 1,000,000 เกทขึ้นไป

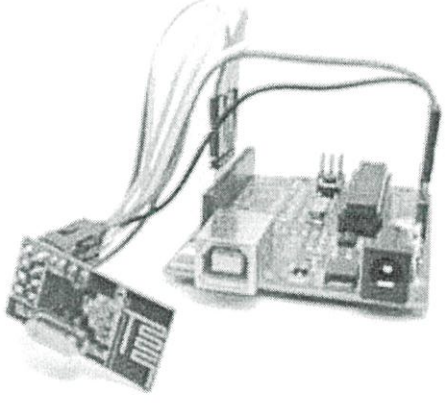


รูปที่ 2.12 แสดงลักษณะของไอซี (1)

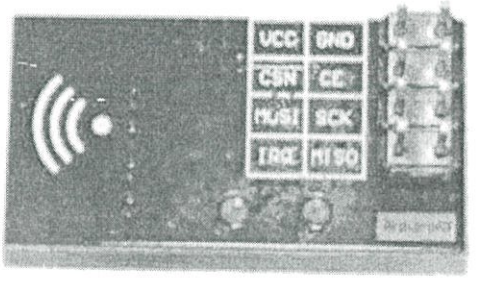


รูปที่ 2.13 แสดงลักษณะของไอซี (2)

2.6 โมดูล NRF24L01

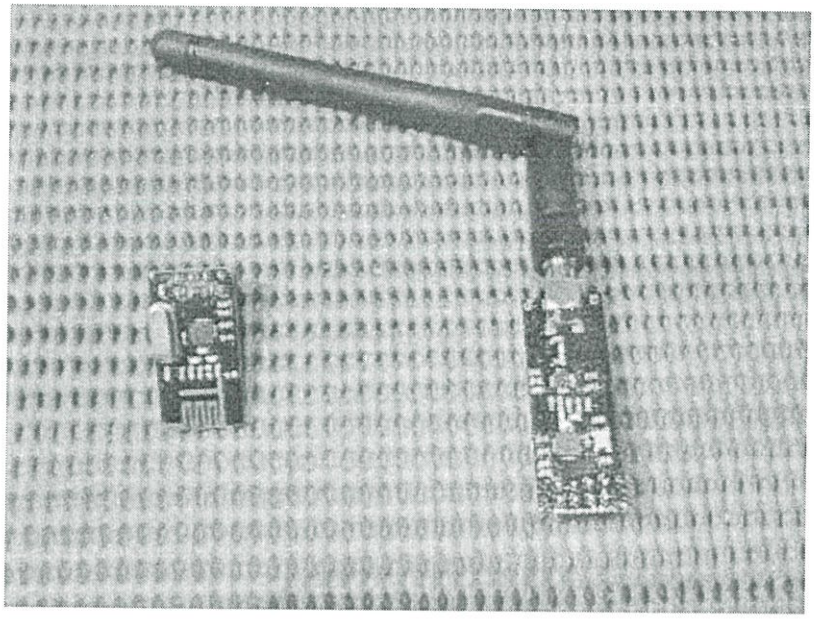


รูปที่ 2.14 แสดงลักษณะของโมดูล NRF24L01 (1)



รูปที่ 2.15 แสดงลักษณะของโมดูNRF24L01 (2)

NRF24L01 เป็น NF ชนิดไร้สายที่ใช้ช่วงความยาวคลื่นที่ 2.4 GHz ซึ่งเป็นย่านความถี่เดียวกับ WiFi, GPS หรือวิทยุสมัครเล่น มีสองรูปแบบคือแบบที่ไม่มีเสาอากาศแบบ Zig-Zag (ซ้ายมือ) และแบบที่มีตัวขยายสัญญาณ (ขวามือ) ซึ่งมีราคาแตกต่างกัน แต่การใช้งานและตำแหน่งของ Pin เหมือนกัน



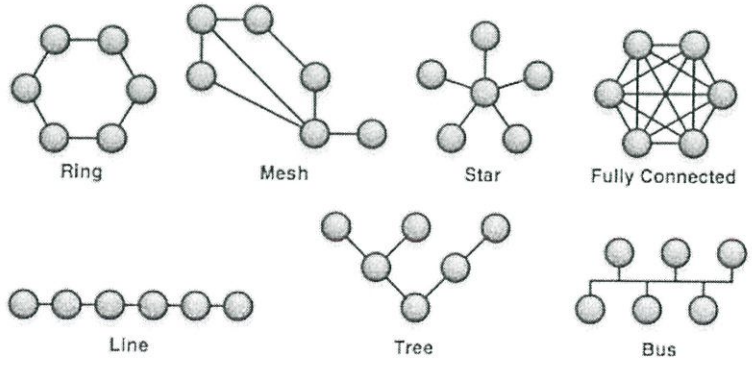
รูปที่ 2.16 แสดงลักษณะของโมดูล NRF24L01 แบบมีเสาและไม่มีเสา

ที่แตกต่างคือตัวเลขระยะห่างหรือระยะการส่งข้อมูล ระหว่างโมดูล ตามเอกสารจากผู้ผลิตหรือผู้จำหน่าย จะอ้างถึงตัวเลขที่แตกต่างกันเล็กน้อย แต่ก็อยู่ที่ประมาณ 100 เมตรในที่โล่ง (ในที่มีสิ่งกีดขวางเยอะระยะก็จะลดลง) สำหรับเสาอากาศแบบ Zig-Zag แบบที่มีตัวขยายสัญญาณ ก็จะได้ระยะเพิ่มอีกเท่าตัว

2.6.1 Network Topology

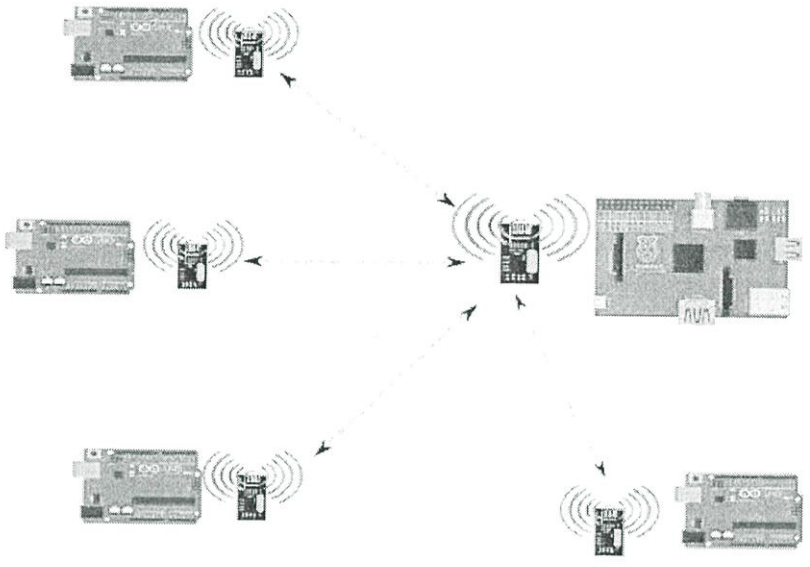
Network Topology คือรูปแบบการเชื่อมโยงระหว่างอุปกรณ์หรือ Node ต่าง ๆ ในข่ายงานสื่อสาร รูปแบบที่ว่านี้ได้แก่ Point to Point เป็นรูปแบบที่ง่ายที่สุด เป็นการสื่อสารระหว่าง Node 2 Node

, Bus, Star, Ring, Tree และ Mesh (เอกสารบางฉบับอาจมีการผสมกันระหว่างรูปแบบเหล่านี้ได้เพิ่มมาอีกหลายรูปแบบ)

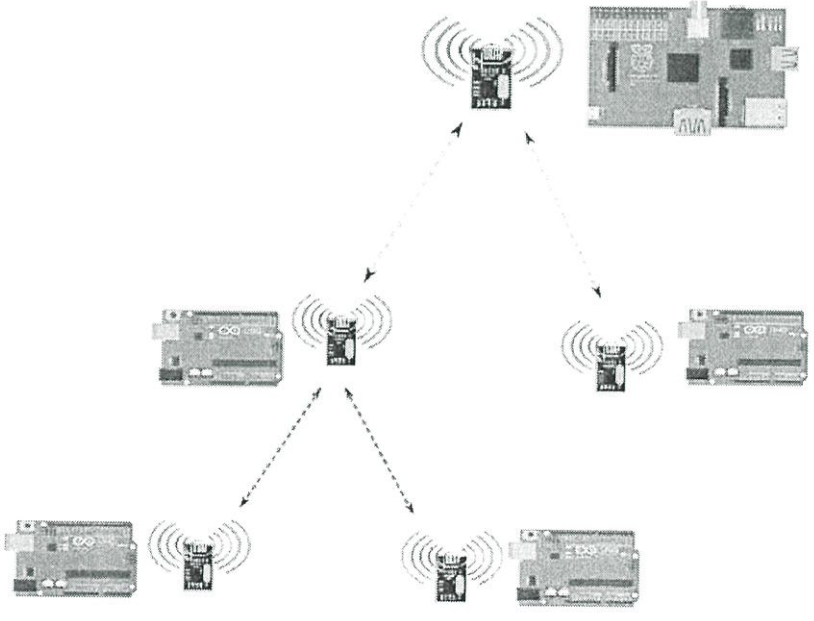


รูปที่ 2.17 แสดงลักษณะของเครือข่ายแบบต่างๆ

ที่กล่าวถึงเรื่องรูปแบบเครือข่าย นี้ก็เพื่อจะบอกให้ทราบว่าข้อดีของการสื่อสารด้วย NRF24L01 ว่าช่วยให้เราสามารถสร้างโครงข่ายการสื่อสารระหว่าง Raspberry PI กับ Arduino ได้หลายรูปแบบมากขึ้น (ยกเว้นรูปแบบ Fully Mesh ซึ่งต้องใช้ Zigbee) ไม่ได้จำกัดอยู่กับอุปกรณ์เพียงสองชิ้นดังนั้นเราจึงสร้างประโยชน์จากรูปแบบการสื่อสารได้มากขึ้นตามไปด้วย



รูปที่ 2.18 แสดงตัวอย่างการเชื่อมต่อเครือข่ายแบบ Star



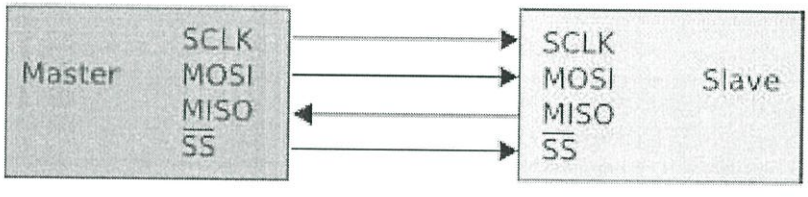
รูปที่ 2.19 แสดงตัวอย่างการเชื่อมต่อเครือข่ายแบบ Tree

2.6.2 SPI

Serial Peripheral Interface (SPI) คือโปรโตคอลการสื่อสาร (Communication Protocol) ที่ใช้Arduino และ Raspberry PI ใช้สื่อสารกับ NRF24L01 การสื่อสารผ่าน SPI นี้อุปกรณ์จะมีบทให้เล่นสองบทคือ Master โดยจะเป็น Arduino หรือ Raspberry PI และ Slave ซึ่งก็คือตัว NRF24L01 และต้องอาศัยเส้นทางเดินของสัญญาณหลักอย่างน้อย 3 เส้นทางคือ

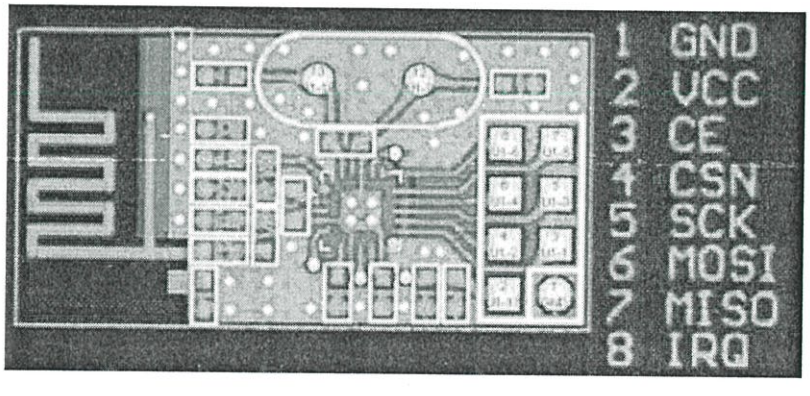
1. MOSI (Master Out Slave In) นำข้อมูลจาก Master ไปสู่ Slave
2. MISO (Master In Slave Out) นำข้อมูลจาก Slave ไปสู่ Master
3. SCLK (Serial Clock) เป็นสัญญาณนาฬิกาที่ Master เป็นผู้สร้างเพื่อให้กำกับเวลาในการสื่อสาร

สำหรับ SS (Slave Select) ไม่ได้เกี่ยวข้องกับการสื่อสารแต่เป็นตัวที่ใช้สำหรับการ ควบคุมเส้นทาง ทั้ง Arduino และ Raspberry PI ต่างก็มี PIN สำหรับใช้งานกับ SPI มาพร้อมแล้ว



รูปที่ 2.20 แสดงรูปแบบการสื่อสารที่เป็นหน้าที่ตัวหลัก (Master) และ ตัวรอง (Slave)

2.6.3 การต่อสายสัญญาณ



รูปที่ 2.21 แสดงรูปแบบการต่อขาพินของโมดูล NRF24L01

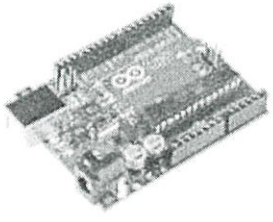
ตารางที่ 2.1 แสดงขาพินต่างๆที่ต่อร่วมกัน
ระหว่างโมดูล NRF24L01Plus และ Arduino Mega 2560

NRF24L01 Plus	Arduino Mega 2560
VCC	3.3V
GRD	GRD
CE	9
CSN	10
SCK	52
MOSI	51
MISO	50
IRQ	No use

NRF24L01 จะมี 2 ขา ที่ใช้ในการควบคุมการสื่อสารคือ CS กับ CSN โดย CS จะมี Mode เป็น อินพุทเสมอ หากให้ค่าเป็น HIGH NRF24L01 จะทำการตรวจสอบอากาศเพื่อรับข้อมูล หากเป็น LOW ก็จะอยู่ในสภาพรอรับคำสั่ง และพร้อมจะส่งข้อมูลออกไป ส่วน CSN จะเกี่ยวข้องกับการรับ-ส่งข้อมูลผ่าน SPI Bus ในการทำงานจริงนั้น Library จะรับหน้าที่จัดการกำหนดค่าให้กับ ขาเหล่านี้ตามความเหมาะสมเอง

2.7 Arduino

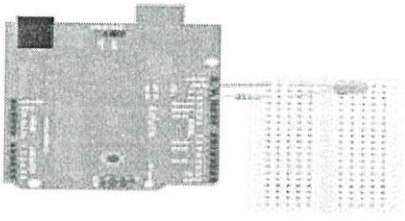
2.7.1 Arduino คืออะไร



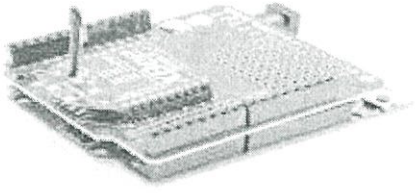
รูปที่ 2.22 แสดงบอร์ดไมโครคอนโทรเลอร์ Arduino

Arduino อ่านว่า (อา-ดู-อิ-โน่ หรือ อาดุยโน่) เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์ และซอฟต์แวร์ ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิคส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Arduino Shield) ประเภทต่างๆ เช่น Arduino XBee Shield, Arduino Music Shield, Arduino Relay Shield, Arduino Wireless Shield, Arduino GPRS Shield เป็นต้น มาเสียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย



รูปที่ 2.23 แสดงบอร์ด Arduino ต่อกับ โฟโตบอร์ด

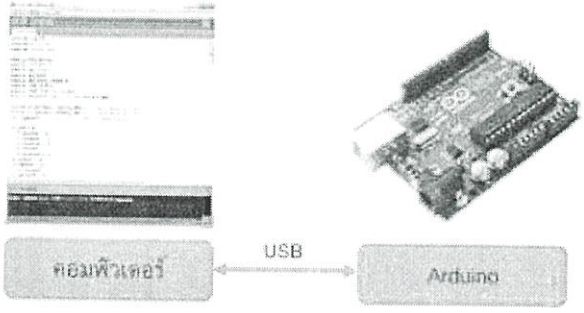


รูปที่ 2.24 แสดงบอร์ดArduino ต่อกับ X-Bee

2.7.2 จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแรง
- Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- ราคาไม่แพง
- Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

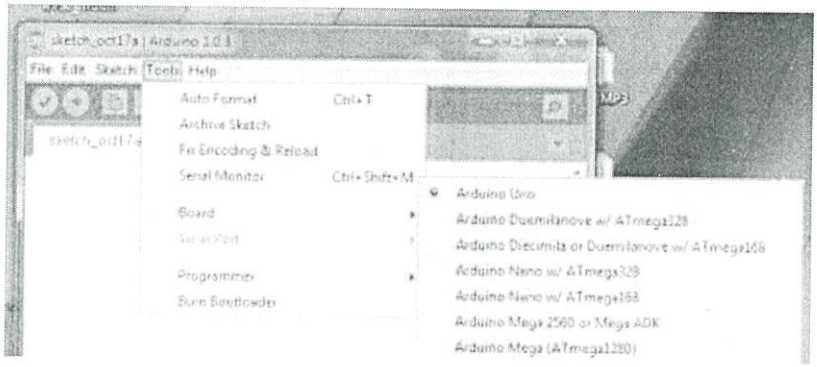
2.7.3 รูปแบบการเขียนโปรแกรมบน Arduino



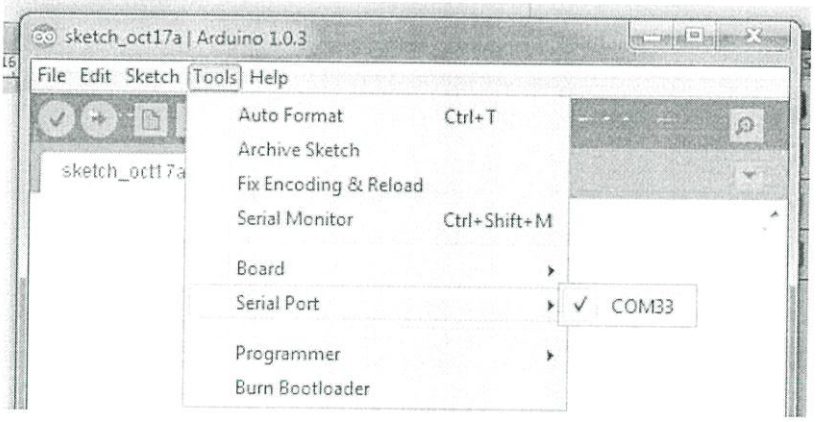
รูปที่ 2.25 แสดงการต่อ Arduino กับคอมพิวเตอร์

1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE ซึ่งสามารถดาวน์โหลดได้จาก Arduino.cc/en/main/software

2. หลังจากที่เขียนโค้ดโปรแกรมเรียบร้อยแล้วจะให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port

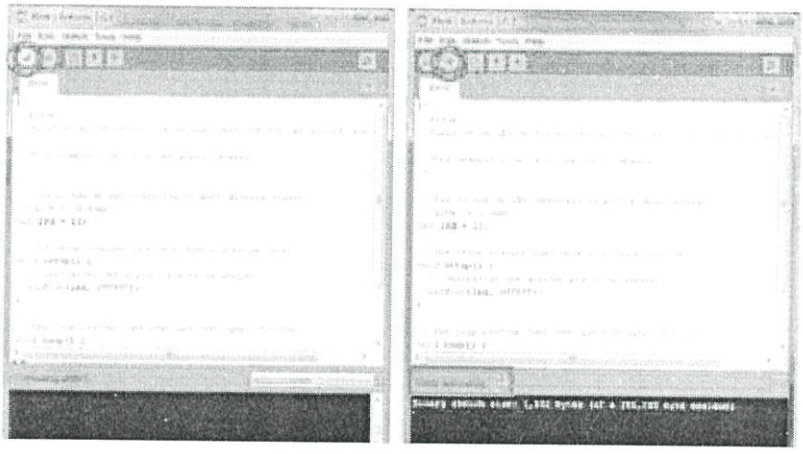


รูปที่ 2.26 แสดงการติดตั้งโปรแกรมที่ใช้ติดต่อ Arduino (1)



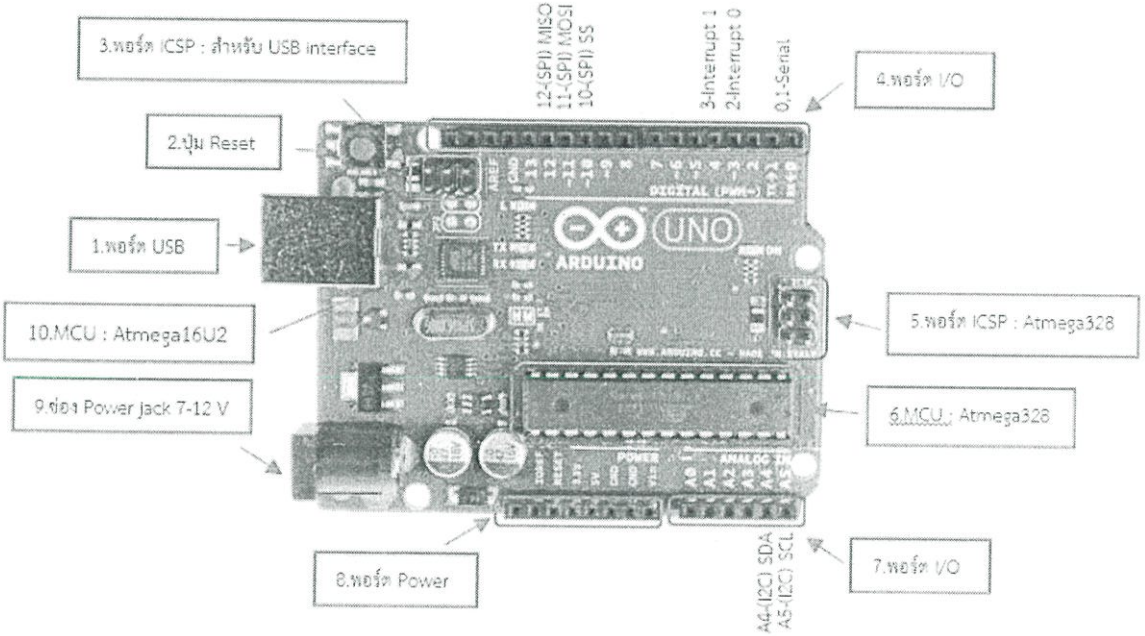
รูปที่ 2.27 แสดงการติดตั้งโปรแกรมที่ใช้ติดต่อ Arduino (2)

3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduinoผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความแถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



รูปที่ 2.28 แสดงการติดตั้งโปรแกรมที่ใช้ติดต่อ Arduino (3)

2.7.4 Layout & Pin out Arduino Board



รูปที่ 2.29 แสดงส่วนประกอบต่างๆ ของบอร์ด Arduino

1. USBPort : ใช้สำหรับต่อกับ Computer เพื่ออับโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่

3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

4. I/OPort : Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx, Rx Serial, Pin 3, 5, 6, 9, 10 และ 11 เป็นขา PWM

5. ICSP Port: Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader

6. MCU: Atmega328 เป็น MCU ที่ใช้บนบอร์ด Arduino

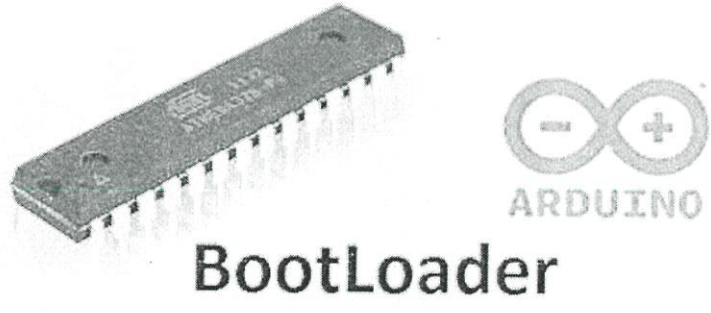
7. I/OPort: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5

8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin

9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V

10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อ กับ Computer ผ่านAtmega16U2

สรุป Arduino คือ การเอา Atmel chip คือตัว Atmega 168/328 มาบรรจุ Bootloader ซึ่งทำให้การ Burn หรือ Flash หรือจะทำการป้อน ภาษาเครื่อง (Hex File) ง่ายขึ้น ยิ่งไปกว่านั้น ยังมีสิ่งอำนวยความสะดวก ในเรื่อง Library ให้ใช้มากมาย



รูปที่ 2.30 แสดงรูปภาพของ Bootloader

2.8 ภาษาที่ใช้สำหรับการเขียนโปรแกรมบน Arduino

โปรแกรมภาษาของ Arduino จะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับ ภาษาซีมาตรฐาน (ANSI-C) อื่นๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความ

ยุ่งยากในการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรง

2.8.1 โครงสร้างการเขียนโปรแกรม ภาษาซี ของ Arduino

ภาษาซีของ Arduino จะจัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็นส่วนย่อยๆ หลายๆส่วน โดยเรียกแต่ละส่วนว่า ฟังก์ชั่น และ เมื่อนำฟังก์ชั่น มารวมเข้าด้วยกัน ก็จะเรียกว่าโปรแกรม โดยโครงสร้างการเขียนโปรแกรมของ Arduino นั้น ทุกๆโปรแกรมจะต้องประกอบไปด้วยฟังก์ชั่นจำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมีฟังก์ชั่น จำนวน 2 ฟังก์ชั่น คือ setup() และ loop() ดังตัวอย่าง

```
#include <Servo.h> // สั่งผนวกไฟล์ ชื่อ Servo.hเข้ามาใช้ในโปรแกรม
int Servo1 = 9; // กำหนดให้ Servo1 แทน Pin Digit = 9
Servo myservo; // สร้าง object ชื่อ myservoเพื่อควบคุม Servo
void setup()
{
myservo.attach(Servo1); // กำหนดให้ใช้ขา Digital-9 สร้างสัญญาณควบคุม Servo
}
void loop()


{
myservo.write(180); // กำหนดค่าตำแหน่งให้กับ Servo = 180 องศา
}
```

2.9 เทคโนโลยีการสื่อสารแบบไร้สาย

องค์ประกอบหลักของการทำงาน มีดังต่อไปนี้

- ข้อมูล (Data) ข่าวสารต่างๆ สามารถถูกส่งแบบไร้สาย โดยเปลี่ยนจากสัญญาณวิทยุและสัญญาณโทรทัศน์เป็นเสียงและภาพ หรือเป็นข้อมูลด้านคอมพิวเตอร์ข่าวสารต่างๆ จะถูกส่งโดยผสมไปกับคลื่นวิทยุ ซึ่งคลื่นวิทยุเป็นแค่ส่วนหนึ่งในสเปกตรัมคลื่นแม่เหล็กไฟฟ้าในช่วงที่เรียกว่า ความถี่วิทยุ (Radio frequency) ข้อมูลทุกชนิดสามารถถูกส่งโดยการใช้ความถี่วิทยุ ความถี่วิทยุมีมากมายไม่ใช่มีแค่คลื่นวิทยุเอเอ็มหรือเอฟเอ็มที่เรารู้จักเท่านั้น โมดูเลชัน (Modulation) ข้อมูลจะถูกส่งผสมไปคลื่นความถี่วิทยุโดยกระบวนที่เรียกว่า โมดูเลชัน เมื่อได้รับคลื่นสัญญานจะผ่านกระบวนการ ดีโมดูเลชัน (Demodulation) เพื่อแยกเอาข้อมูลออกมา สถานีฐาน (Base Station) ภายในแต่ละเซลล์จะมีสถานีฐานซึ่งจะส่งและรับสัญญาณสื่อสารทั้งรับและส่งไปยังโทรศัพท์มือถือ (หรือโทรศัพท์เซลลูล่าร์) ภายในเซลล์นั้นๆ เซลล์ (Cells) โทรศัพท์เซลลูลาร์ หรือโทรศัพท์แบบพกพาหรือโทรศัพท์มือถือที่เรารู้จัก มีแนวความคิดมาจากเซลล์นี่เอง ซึ่งเป็นการแบ่งพื้นที่ออกเป็นส่วนย่อยๆ เมื่อโทรศัพท์เซลล์ลูลาร์รับ-ส่งสัญญาณจะมีการ

ติดต่อสื่อสารภายในเซลล์นั้นๆเองเท่านั้น ข่าวสารต่างๆๆ จะถูกส่งภายในเซลล์นั้นไปที่เป้าหมายภายในเซลล์เท่านั้น

- อุปกรณ์ส่งสัญญาณและอุปกรณ์รับสัญญาณ คลื่นความถี่วิทยุพร้อมด้วยข้อมูลข่าวสารจะถูกส่งโดยอุปกรณ์ส่งสัญญาณ (Transmitters) และรับโดยอุปกรณ์ที่เรียกว่าอุปกรณ์รับสัญญาณ (Receivers)

2.9.1 ระบบวิทยุ 2.4 GHz Spread Spectrum(FHSS,DSSS)

ปัจจุบันเทคโนโลยีการสื่อสารแบบไร้สายได้มีความต้องการใช้งานที่มากขึ้นแต่ความถี่ (Frequency Spectrum)ที่สามารถใช้งานได้ก็ยังคงมีอยู่อย่างจำกัด ไม่เว้นแม้แต่ในวงการวิทยุบังคับซึ่งก็ใช้คลื่นวิทยุ(RF)ในการติดต่อสื่อสารเช่นเดียวกับการสื่อสารไร้สายทั่วไปเช่น วิทยุ,โทรทัศน์,ไวเลสแลนด์ เป็นต้น

ดังนั้นทุกๆประเทศจึงต้องมีการจัดสรรความถี่ในการใช้งานแต่ละสาขาของการใช้งาน เพื่อป้องกันการรบกวนกันของสัญญาณ ตัวอย่างเช่น, วิทยุระบบ FM ใช้คลื่นความถี่อยู่ที่ 87.5 – 108.0 MHz เป็นต้น

สำหรับคลื่นความถีที่ใช้กับอุปกรณ์ประเภทวิทยุบังคับก็ได้ถูกกำหนดมาแล้วเช่นกันซึ่งเราจะคุ้นเคยกันอยู่กับคลื่น 27MHz 49MHz 50MHz 72MHz 75MHz ซึ่งระบบวิทยุแบบเดิมด้วยการเข้ารหัสสัญญาณคลื่นวิทยุแบบ FM(Frequency Modulation)แน่นอนว่าในการเล่นวิทยุบังคับในแต่ละพื้นที่ที่ระยะส่งสัญญาณไปถึงจะต้องไม่มีการใช้วิทยุบังคับที่มีคลื่นความถี่ซ้ำกันซึ่งเป็นข้อจำกัดอย่างหนึ่งของระบบวิทยุนี้ แม้จะมีการแก้ปัญหาโดยการแบ่งความถี่ให้ย่อยลงไปอีกเพื่อให้มีการใช้งานพร้อมกันได้มากขึ้นเช่นความถี่ 72 MHz ที่แบ่งได้อีกถึง ช่องสัญญาณ

72.010 MHz - Ch 11

72.030 MHz - Ch 12

..... MHz - Ch xx

72.970 MHz - Ch 59

72.990 MHz - Ch 60

ถึงแม้จะมีช่องความถี่ให้ใช้มากขึ้นถึง 50 ความถี่ แต่โอกาสที่จะไปเล่นเครื่องบินที่สนามแล้วเจอผู้เล่นคนอื่นที่มีความถี่ตรงกับเราก็ยังมีเยอะอยู่ดีและทุกครั้งก่อนเล่นก็ต้องมีระบบจัดการคลื่นความถี่ของสนามนั้นๆเพื่อป้องกันการเล่นคลื่นความถี่เดียวกันพร้อมๆกันโดยส่วนมากก็จะเป็นการใช้ไม้หนีบหรือการเขียนลงบอร์ดไว้เพื่อให้ผู้เล่นท่านอื่นที่มาเล่นทีหลังได้ทราบว่าในตอนนี้มีคลื่นไหนบ้างที่กำลังเล่นอยู่ บางครั้งก็ต้องแก้ปัญหาหากความถี่ที่เราต้องการใช้นั้นมีคนอื่นใช้งานอยู่ด้วยการเปลี่ยนความถี่เป็นความถี่ที่ไม่มีคนอื่นใช้ด้วยการเปลี่ยนแร่ความถี่

และแล้วเมื่อไม่นานมานี้ก็มีเทคโนโลยีใหม่เข้ามาช่วยให้วงการวิทยบังคับได้ก้าวข้ามพ้นความลำบากเหล่านี้ ด้วยเทคโนโลยี Spread Spectrum บนคลื่นความถี่ 2.4GHz ทำให้ผู้เล่นวิทยุบังคับไม่ต้องกังวลเรื่องคลื่นวิทยุจะตรงกันและนอกเหนือจากนั้นรีซีฟยังสามารถใช้ร่วมกันได้กับวิทยุที่ใช้ระบบเดียวกันได้อีกด้วย

2.10 Visual basic 2010

รูปที่ 2.31 แสดงสัญลักษณ์โปรแกรม Visual Basic 2010

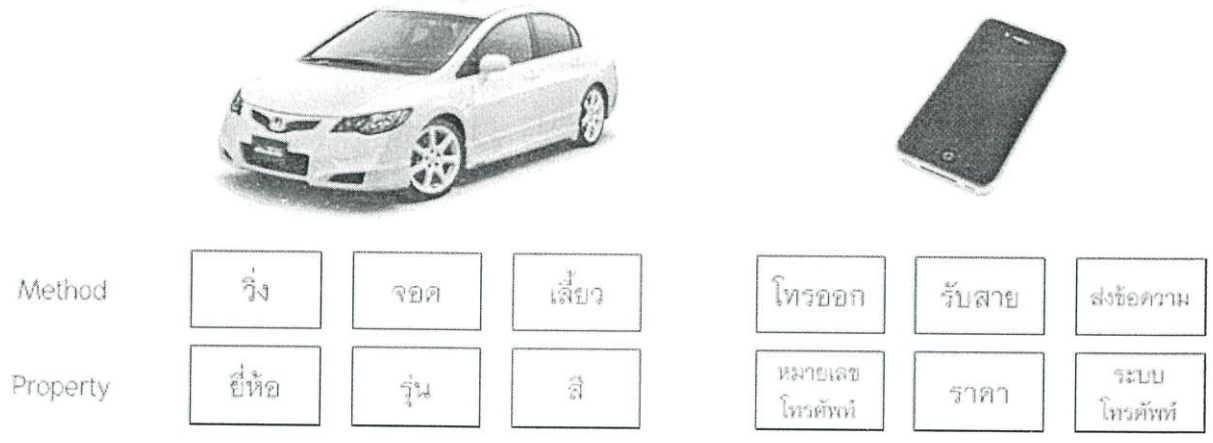
2.10.1 แนวทางการเขียนโปรแกรมภาษา Visual Basic

ปัจจุบันการพัฒนาซอฟต์แวร์ให้ทำงานบนระบบปฏิบัติการ Windows บนระบบอินเทอร์เน็ต หรือบนโทรศัพท์มือถือ สามารถใช้แนวคิดของการเขียนโปรแกรมแบบ Visual Programming ได้เป็นอย่างดี แนวทางการเขียนโปรแกรมดังกล่าวมีความแตกต่างจากแนวทางการเขียนโปรแกรมแบบดั้งเดิม เช่นการเขียนโปรแกรมภาษา C หรือ Java เป็นอย่างมาก สิ่งที่มีความสำคัญที่สุดสำหรับการเขียนโปรแกรมแบบ Visual Programming คือ เครื่องมือช่วยพัฒนาแอพพลิเคชัน ในปัจจุบันมีให้ใช้งานมากมายหลายตัว แต่ละตัวก็มีความสามารถในการสนับสนุนกระบวนการพัฒนาซอฟต์แวร์ในขั้นตอนต่างๆ เริ่มจากการออกแบบแอพพลิเคชัน การเขียนโปรแกรม ตัวแปลภาษา เครื่องมือทดสอบ เครื่องมือที่ใช้ติดต่อฐานข้อมูล เป็นต้น การเขียนโปรแกรมแบบ Visual Programming มี 2 แนวทางที่เป็นที่นิยม คือ การเขียนโปรแกรมแบบ Object Oriented และ การเขียนโปรแกรมแบบ Event Driven

2.10.2 แนวทางการเขียนโปรแกรมแบบ Object Oriented

แนวทางการเขียนโปรแกรมแบบ Object Oriented สามารถเรียกได้อีกชื่อหนึ่งว่า การเขียนโปรแกรมเชิงวัตถุ การจะทำความเข้าใจกับแนวทางการเขียนโปรแกรมเชิงวัตถุนั้น ต้องทำความเข้าใจกับคำว่า ออบเจ็กต์ (Object) คุณสมบัติเฉพาะตัว (Property) และความสามารถที่เรียกใช้งานได้ (Method) เสียก่อน

ออบเจ็กต์ (Object) คือสิ่งที่สมมติขึ้นมาแทนวัตถุอย่างใดอย่างหนึ่ง เช่น ออบเจ็กต์รถยนต์ ออบเจ็กต์โทรศัพท์มือถือ ออบเจ็กต์บ้าน เป็นต้น ซึ่งออบเจ็กต์จะต้องมีลักษณะที่บ่งบอกคุณสมบัติเฉพาะตัว (Property) และความสามารถที่เรียกใช้งานได้ (Method)

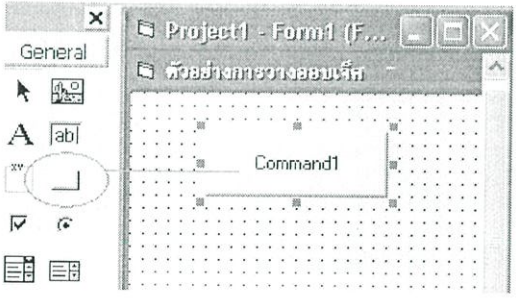


รูปที่ 2.32 แสดงการเปรียบเทียบสิ่งของเป็น Object

2.10.3 การเขียนโปรแกรมด้วย Visual Basic

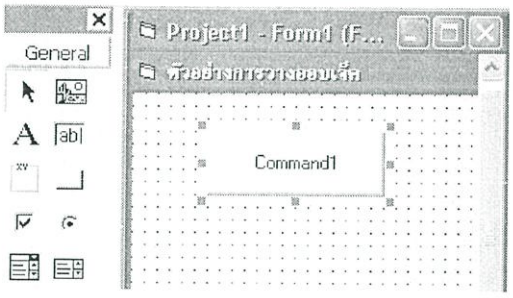
1. ทำการออกแบบหน้าตาของโปรแกรมที่ต้องการติดต่อกับผู้ใช้ โดยการนา Control Object ต่าง ๆ ที่อยู่ใน Toolbox มาวางในฟอร์ม มีวิธีการควบคุม Object ต่าง ๆดังนี้

- การเลือกและวางออบเจ็ค คลิกเมาส์ที่ออบเจ็คที่ต้องการใน Toolbox แล้วนาเมาส์มายังตำแหน่งที่ต้องการวางออบเจ็คนั้นบนฟอร์ม คลิกเมาส์ซ้ายค้างไว้และลากเมาส์จนได้ขนาดที่ต้องการ



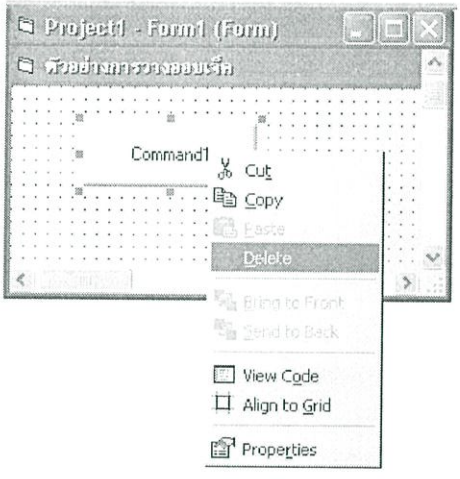
รูปที่ 2.33 แสดงการออกแบบหน้าตาของโปรแกรม

- การจัดตำแหน่งและปรับขนาดออบเจ็ค ทาได้โดยเปลี่ยนเครื่องมือเลือกให้เป็น Pointer รูปลูกศร แล้วจึงนาเมาส์ไปคลิกและลากที่ออบเจ็คที่จะเลือกเพื่อจัดให้อยู่ในตำแหน่งที่ต้องการ สาหรับการปรับขนาดจะทาโดยคลิกเลือกที่ออบเจ็ค จากนั้นนาเมาส์ไปที่ขอบของออบเจ็ค จะสามารถคลิกและลากเพื่อเปลี่ยนขนาดได้ทั้งสี่ด้าน
ของออบเจ็ค



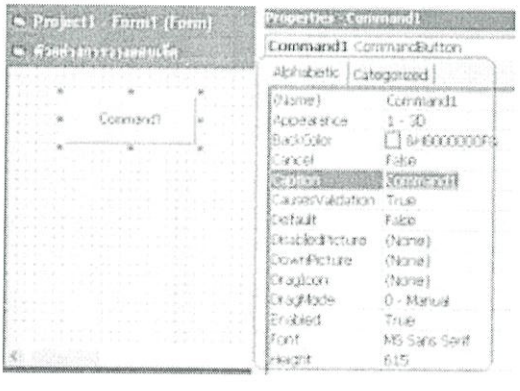
รูปที่ 2.34 แสดงการจัดตำแหน่งและปรับขนาดออบเจ็ค

- การลบออบเจ็ค ออกจากฟอร์ม ทาโดยคลิกเลือกออบเจ็คที่ต้องการลบ กดปุ่ม Delete หรือคลิกขวาเลือกเมนู Delete



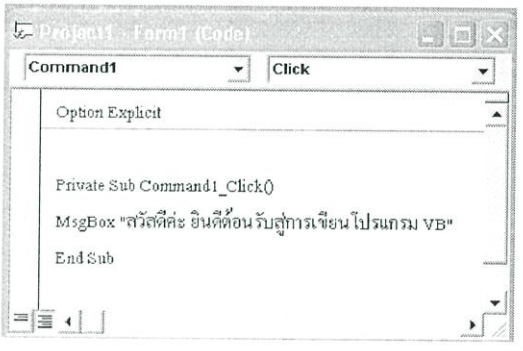
รูปที่ 2.35 แสดงการลบออบเจ็ค

2. กำหนดค่า Properties ของออบเจ็คต่าง ๆ ตามความต้องการและความเหมาะสม โดยใช้วินโดวส์ Properties ซึ่งเรียกโดยกด F4 หรือจากเมนู View > Properties Windows



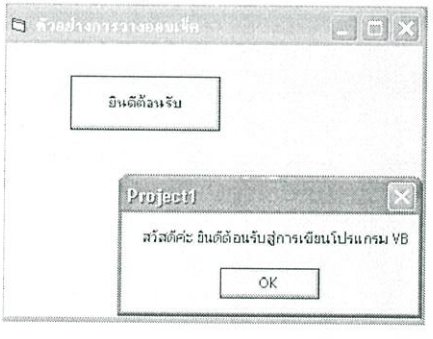
รูปที่ 2.36 แสดงการกำหนดค่า Properties

3. เขียนโปรแกรมให้กับเหตุการณ์ของออบเจ็คต่าง ๆ ตามที่ต้องการ โดยที่หัวของ Code Editor จะเป็นส่วนบอกว่าโค้ดที่แสดงอยู่นั้นเป็นของออบเจ็คและเหตุการณ์ใด โดยการดับเบิลคลิกที่ปุ่มคาสั่งที่ต้องการให้เกิดการทำงาน เช่น ดับเบิลคลิกที่ปุ่ม Command1 เพื่อเขียนคำสั่งดังนี้



รูปที่ 2.37 แสดงเขียนโปรแกรมให้กับเหตุการณ์ของออบเจ็คต่างๆ

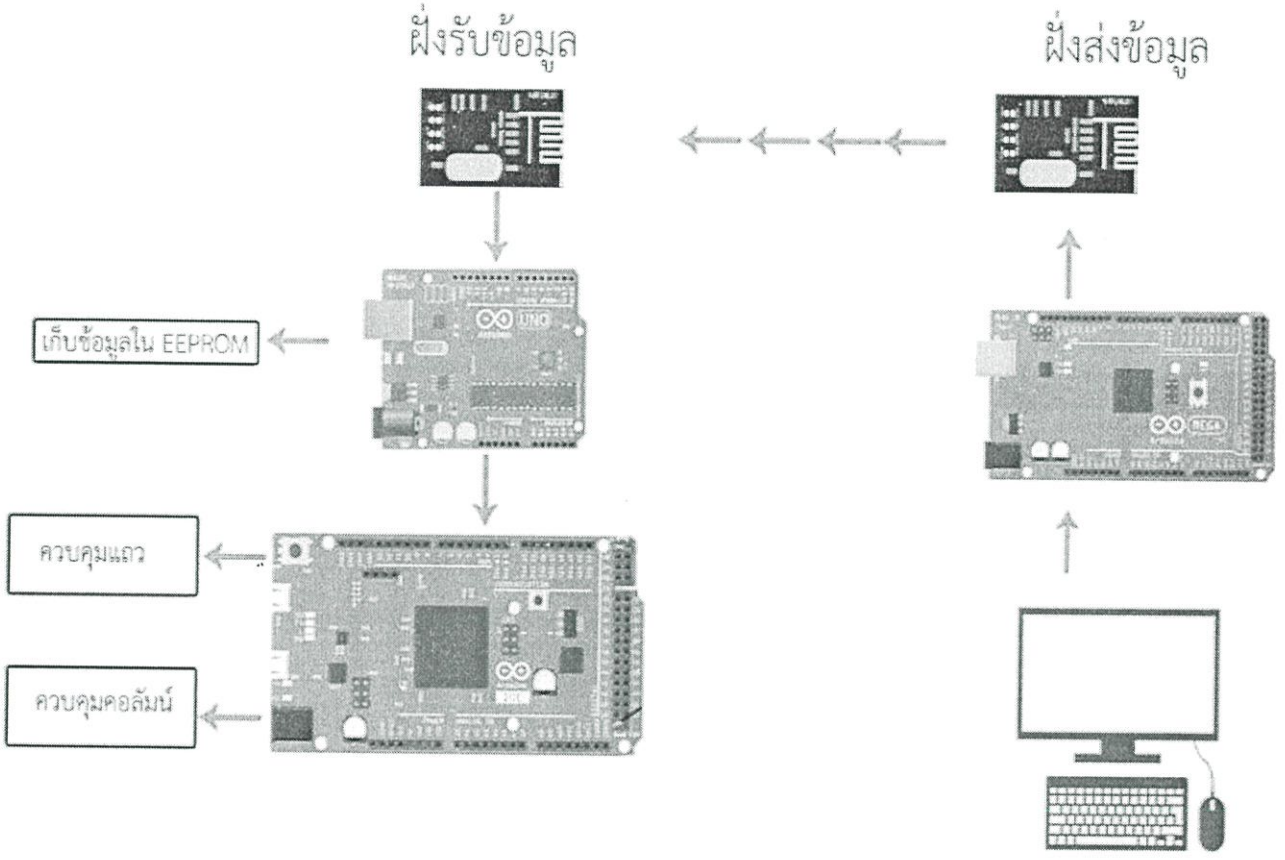
5. ทดสอบและรันโปรแกรมว่าทางานถูกต้องตามที่ต้องการหรือไม่ โดยคลิกที่ปุ่ม Run



รูปที่ 2.38 แสดงการทดสอบและรันโปรแกรม

บทที่ 3
การออกแบบและการทำงานของระบบ

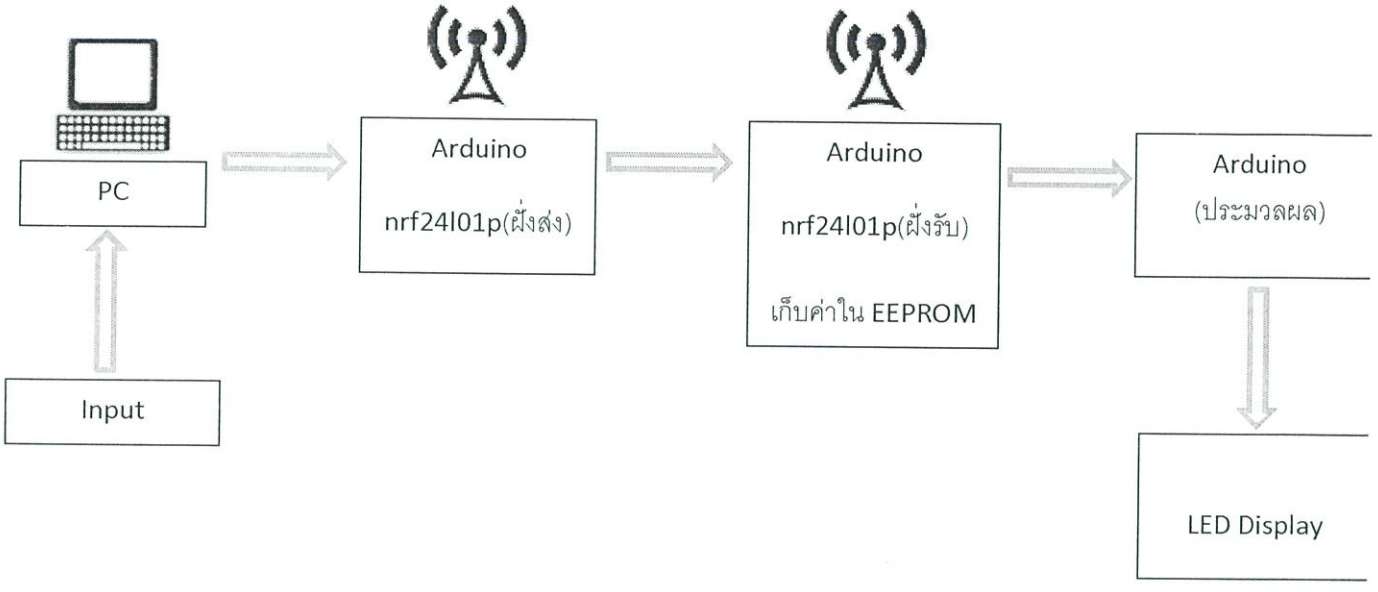
3.1 ภาพรวมของระบบ



รูปที่ 3.1 ภาพรวมการทำงานของระบบ

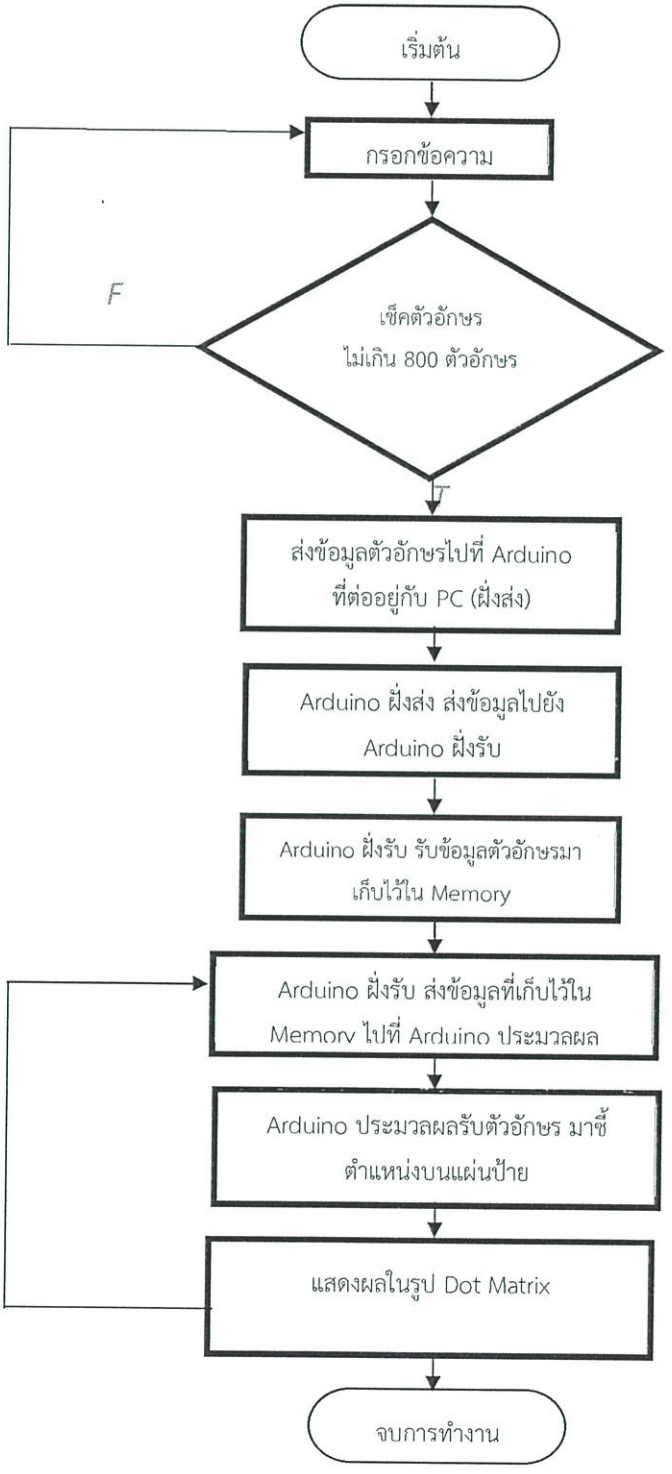
3.1.1 การทำงานของระบบ

เมื่อ USER ทำการกรอกข้อความผ่านโปรแกรม ในคอมพิวเตอร์ลงไป แล้วกดส่งข้อความ Arduino ฝั่งส่งข้อมูลที่เสียบอยู่กับคอมพิวเตอร์จะทำการส่งข้อมูล Text ตามที่ User ป้อนไปยัง Arduino ฝั่งรับที่อยู่ในตัวป้ายประชาสัมพันธ์จากนั้น Arduino ฝั่งรับจะนำข้อมูล Text ที่ USER ป้อนไปเก็บข้อมูลไว้ใน Eeprom Memory แล้วจึงจะเอา Text ส่งให้ Arduino อีกตัวที่ทำหน้าแปลง Text เป็นตำแหน่งที่แสดงผลบนป้าย โดยเอา Text ที่ได้มาแปลงและชี้ตัวอักษรตามโค้ดที่เขียนไว้และ แสดงผลอักษรตามที่ User ป้อนในที่ต้องการให้แสดงบนป้าย



รูปที่ 3.2 บล็อกไดอะแกรมการทำงานของระบบ

3.1.2 โฟลว์ชาร์ต

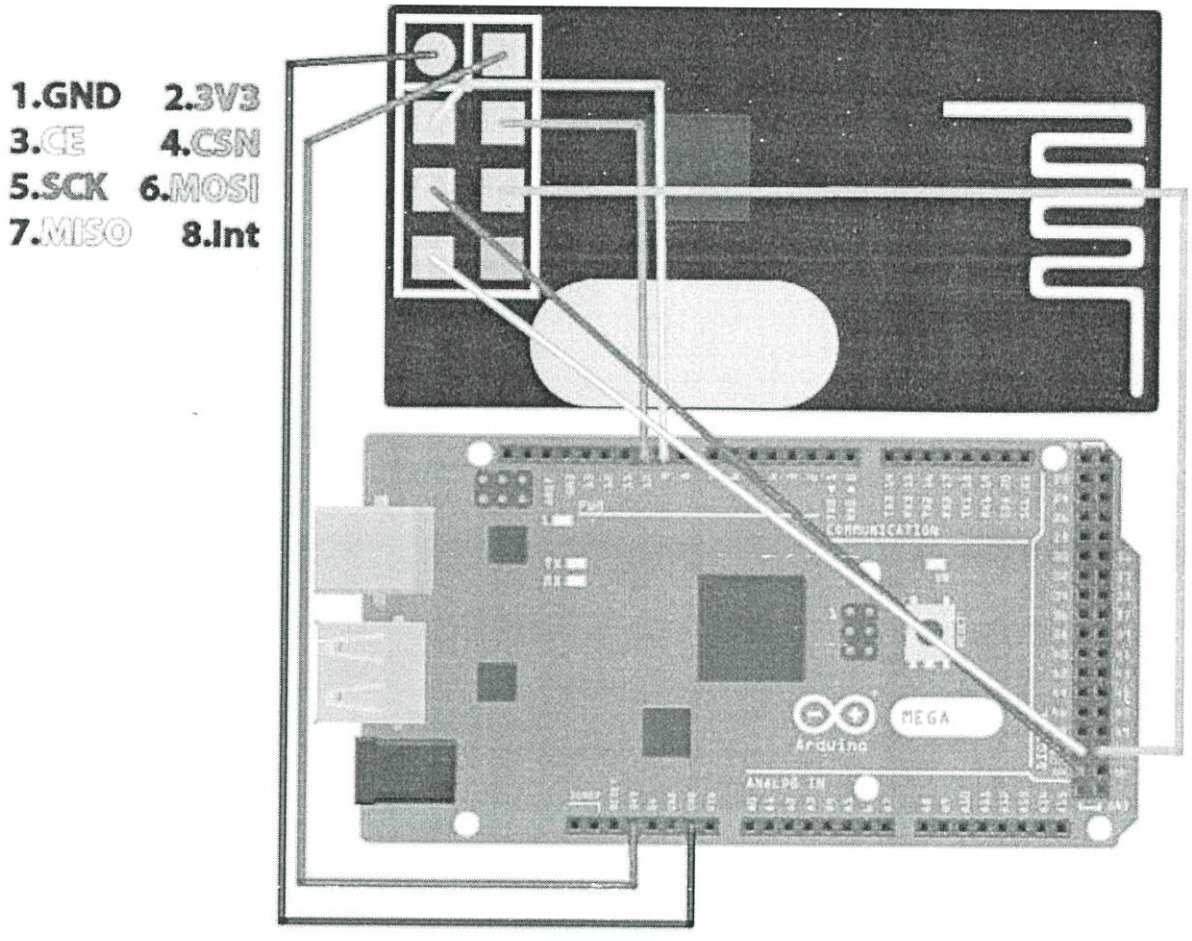


รูปที่ 3.3 โฟลว์ชาร์ตแสดงการทำงานของระบบ

3.2 การออกแบบวงจร

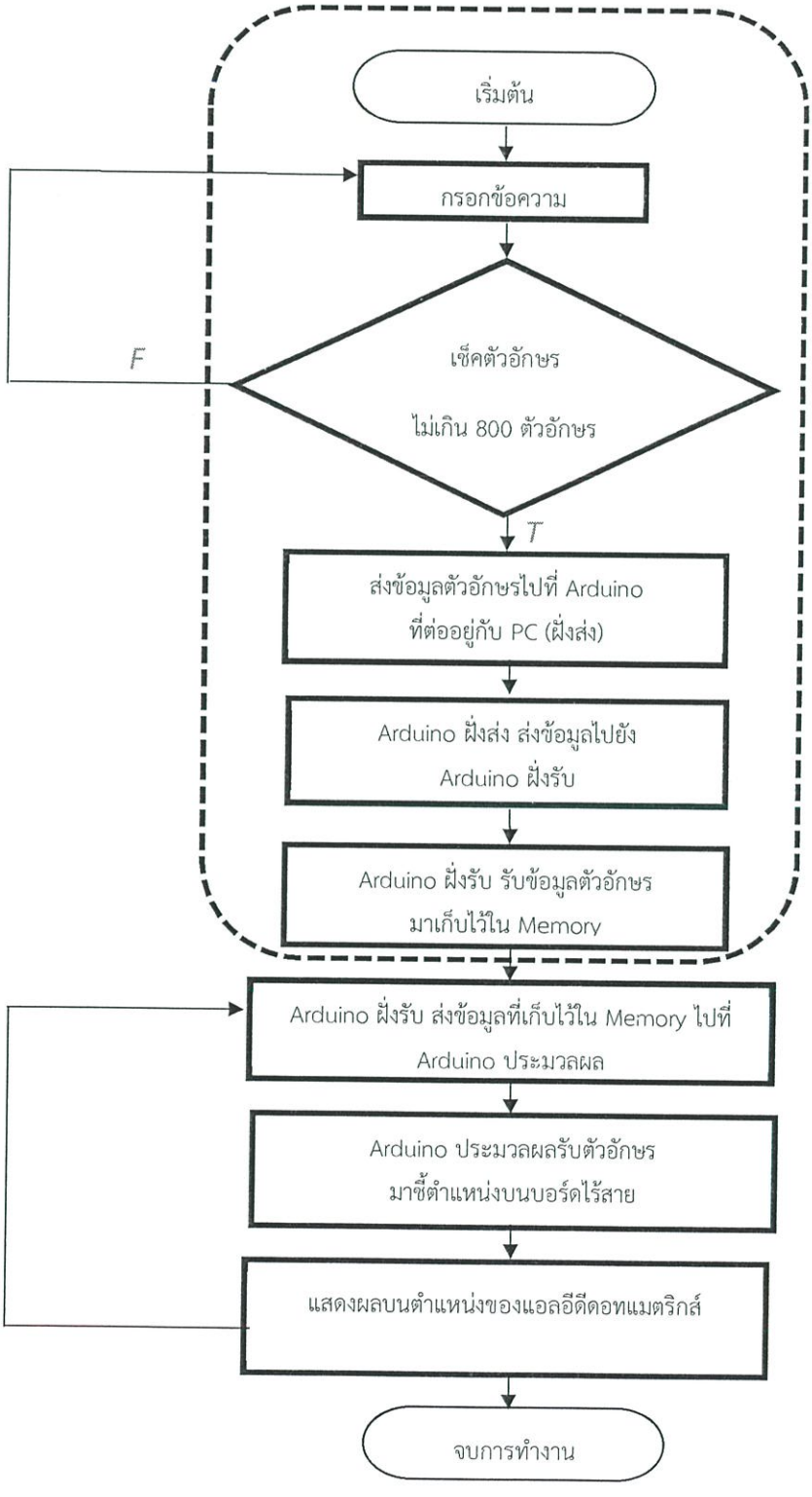
3.2.1 ภาคการส่งข้อมูล

Arduino mega 2560 ต่ออยู่กับคอมพิวเตอร์และตัวnRF24L01P (ฝั่งส่ง) ทำหน้าที่รับข้อมูล
ตัวอักษรจากคอมพิวเตอร์และส่งให้ nRF24L01P (ฝั่งรับ)



รูปที่ 3.4 แสดงการต่อวงจรภาคส่งข้อมูล

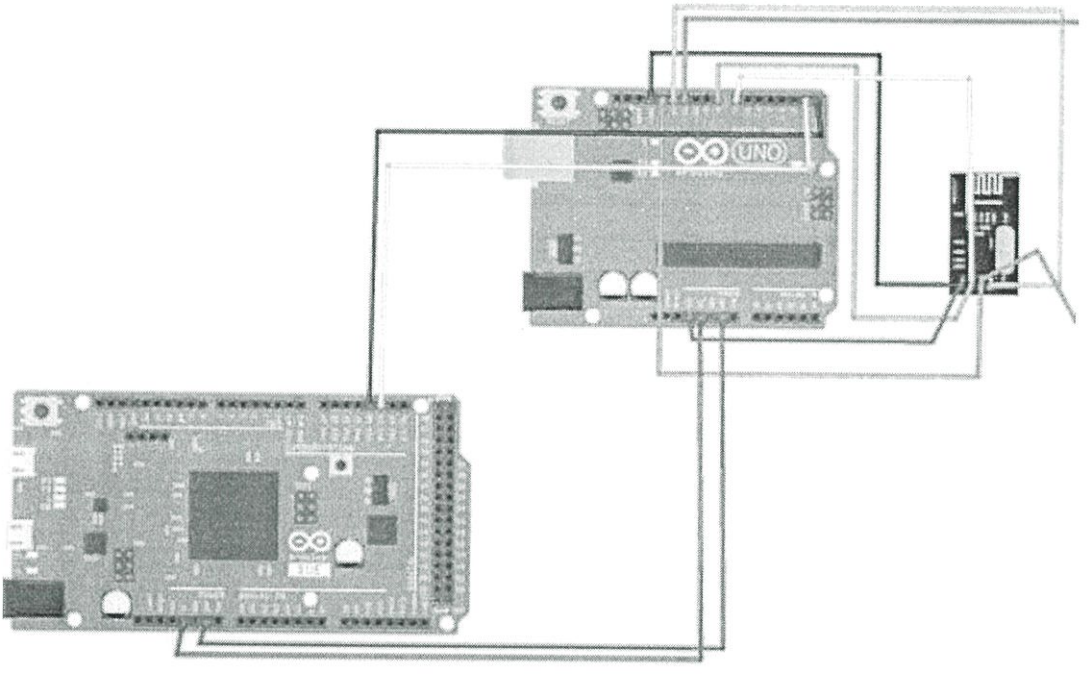
3.2.1.1 โฟลว์ชาร์ต



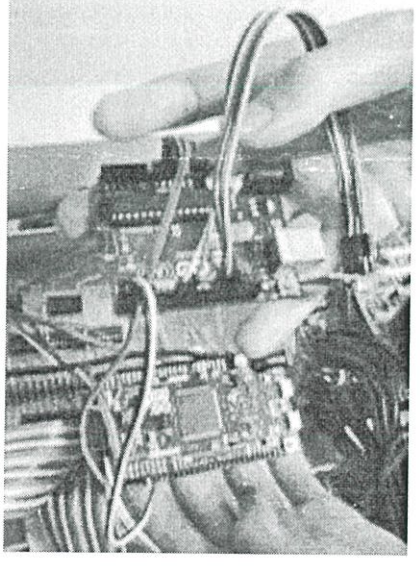
รูปที่ 3.5 โฟลว์ชาร์ตแสดงการทำงานภาคการส่งข้อมูล

3.2.2 ภาคการรับข้อมูล

Arduino UNO ต่ออยู่กับตัว nRF24L01P (ฝั่งรับ) และ Arduino DUE (ประมวลผล) โดยจะทำการรับข้อมูลจาก Arduino Mega 256 มาเก็บไว้ใน EEPROM ซึ่งเวลาปิดบอร์ดประชาสัมพันธ์แล้วจะทำให้ข้อมูลล่าสุดที่ป้อนเข้าไปยังคงอยู่ จากนั้นจะส่งข้อมูลไปยัง Arduino DUE เพื่อประมวลผลต่อไป

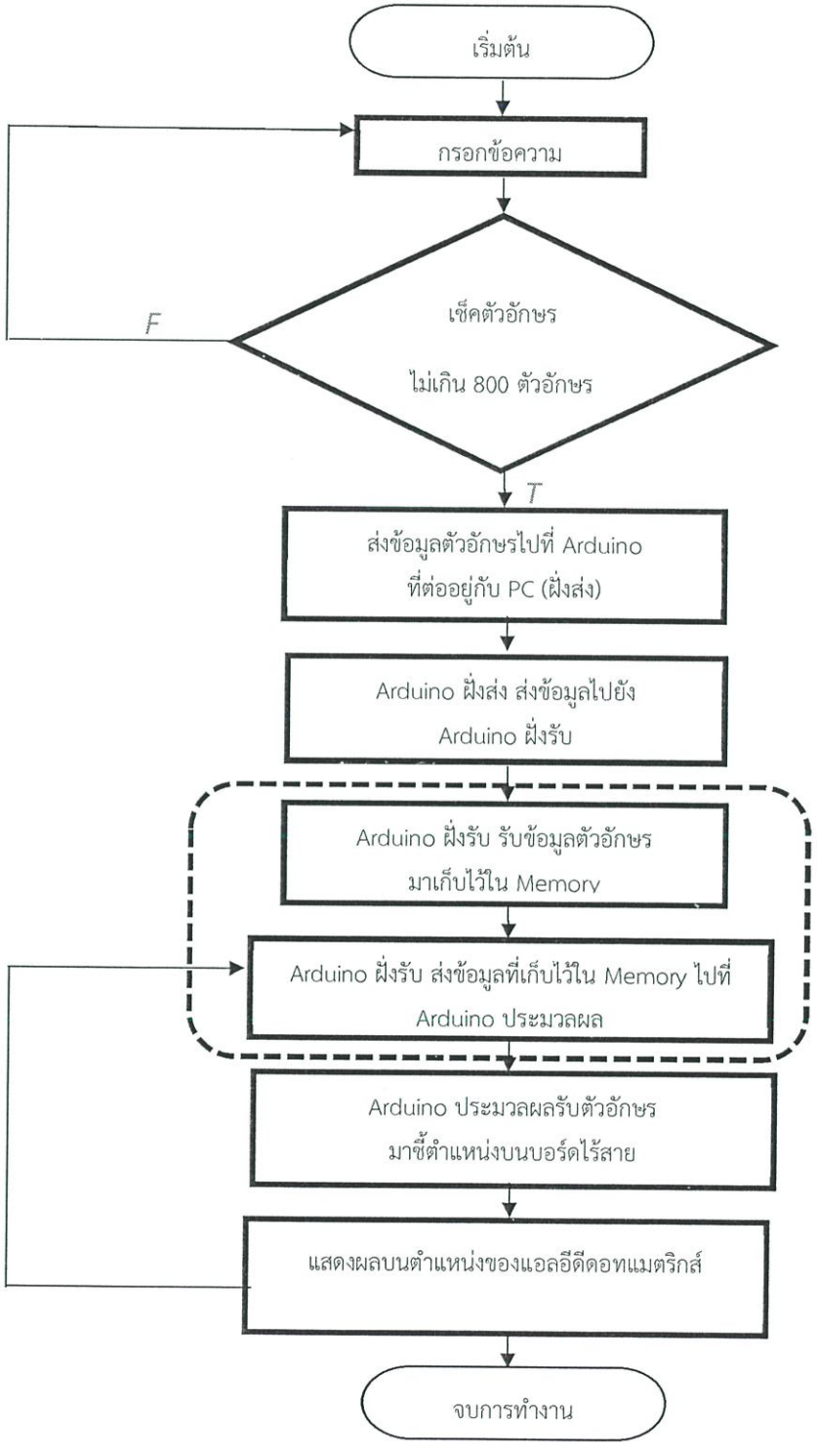


รูปที่ 3.6 แสดงวงจรภาครับข้อมูล



รูปที่ 3.7 แสดงวงจรภาครับข้อมูล

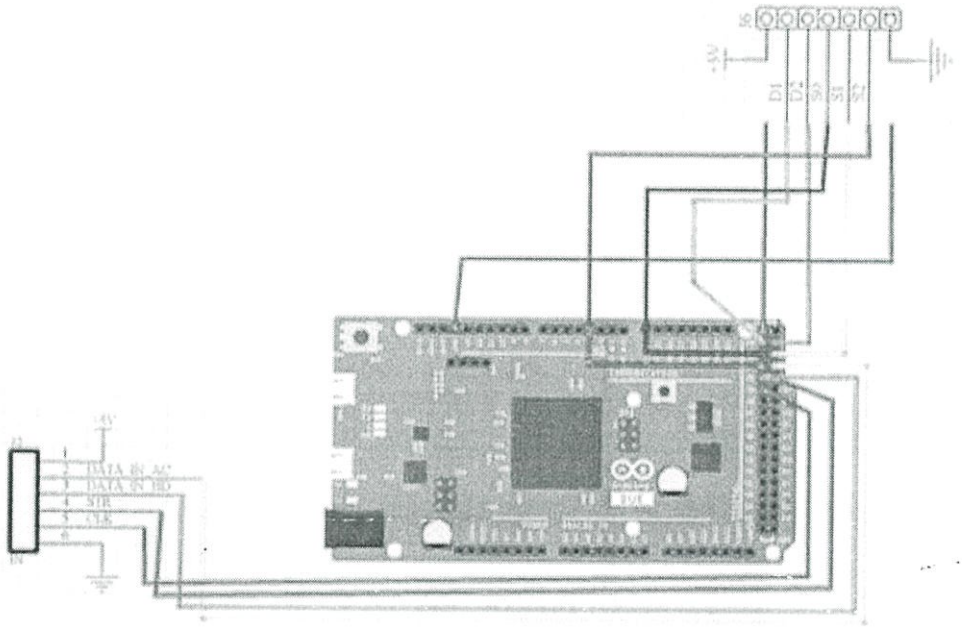
3.2.2.1 โฟลว์ชาร์ต



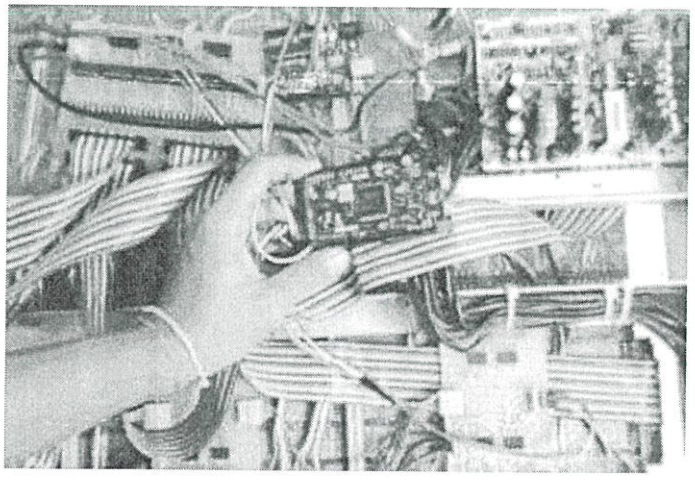
รูปที่ 3.8 โฟลว์ชาร์ตแสดงการทำงานภาครับข้อมูล

3.2.3 ภาคประมวลผล

Arduino DUEต่ออยู่กับ Arduino UNO และและตัวควบคุมฝั่งแถวแนวตั้ง กับ แถว โดยรับข้อมูลมาจาก Arduino UNO มาแปลงเป็นตัวอักษรในรูปของ Dot Matrix และแสดงผลตามโค้ดที่เขียนไว้

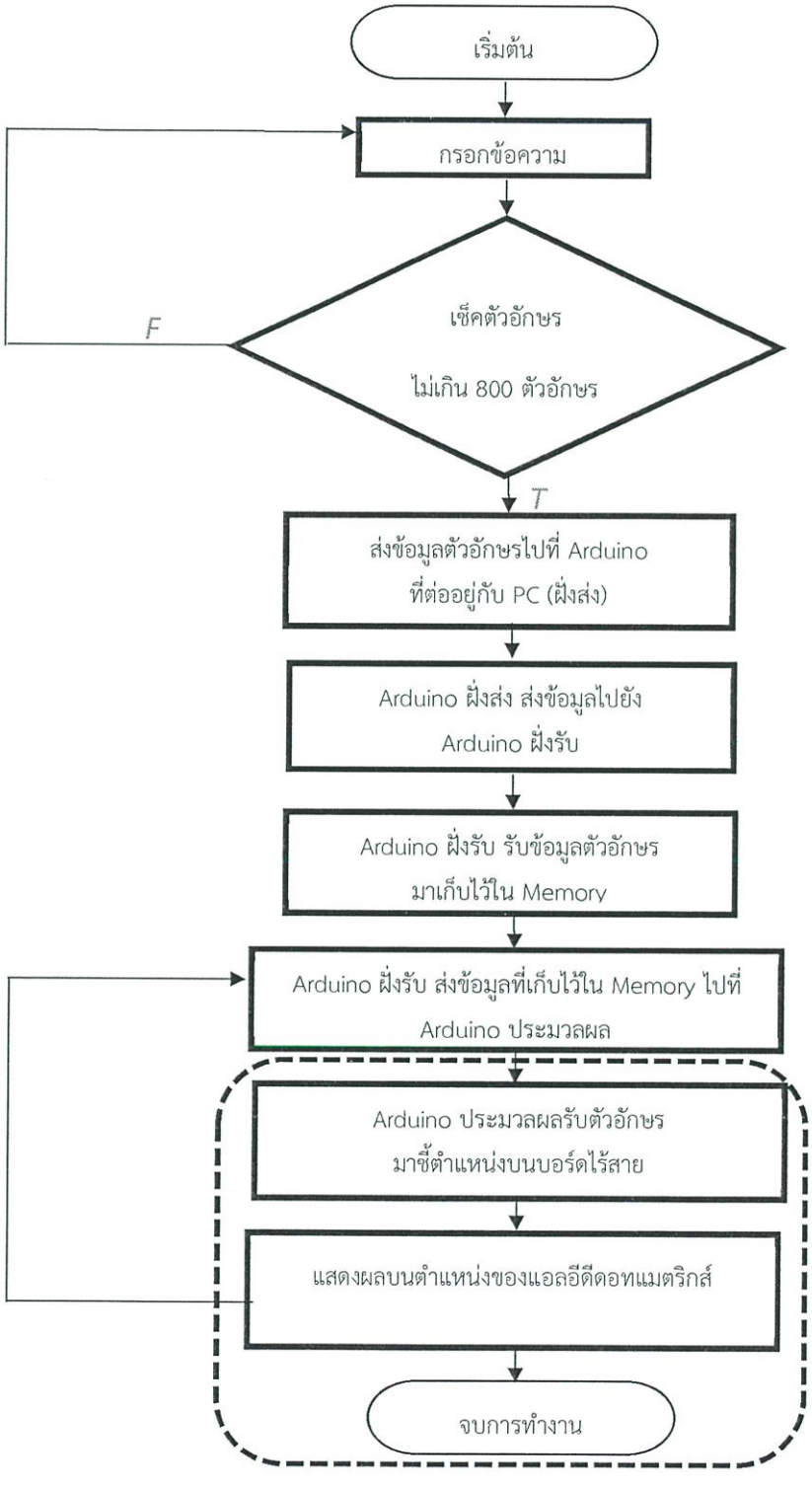


รูปที่ 3.9 แสดงวงจรประมวลผล



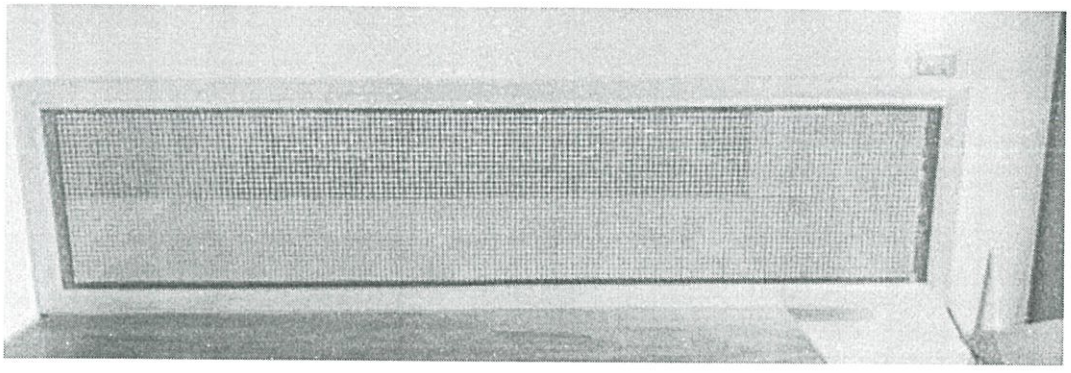
รูปที่ 3.10 แสดงวงจรประมวลผล

3.2.3.1 โฟลว์ชาร์ต



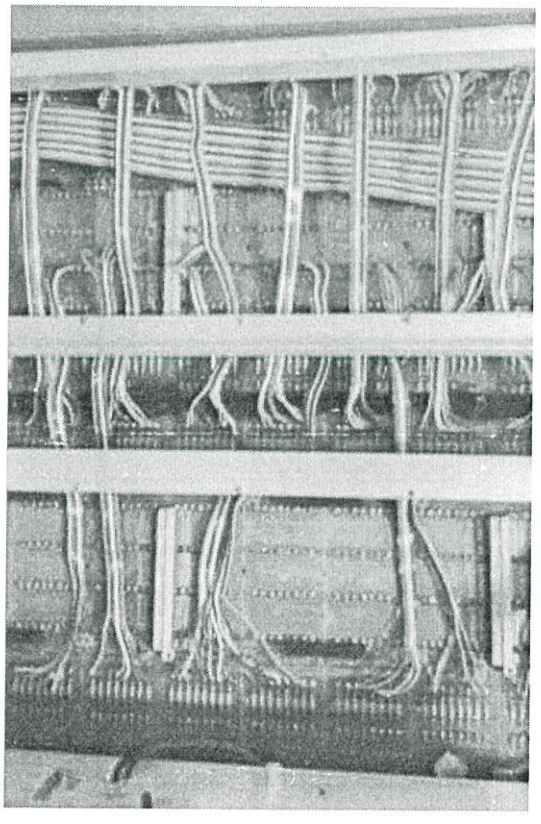
รูปที่ 3.11 โฟลว์ชาร์ตแสดงการทำงานภาคประมวลผล

3.2.4 ภาคแสดงผล



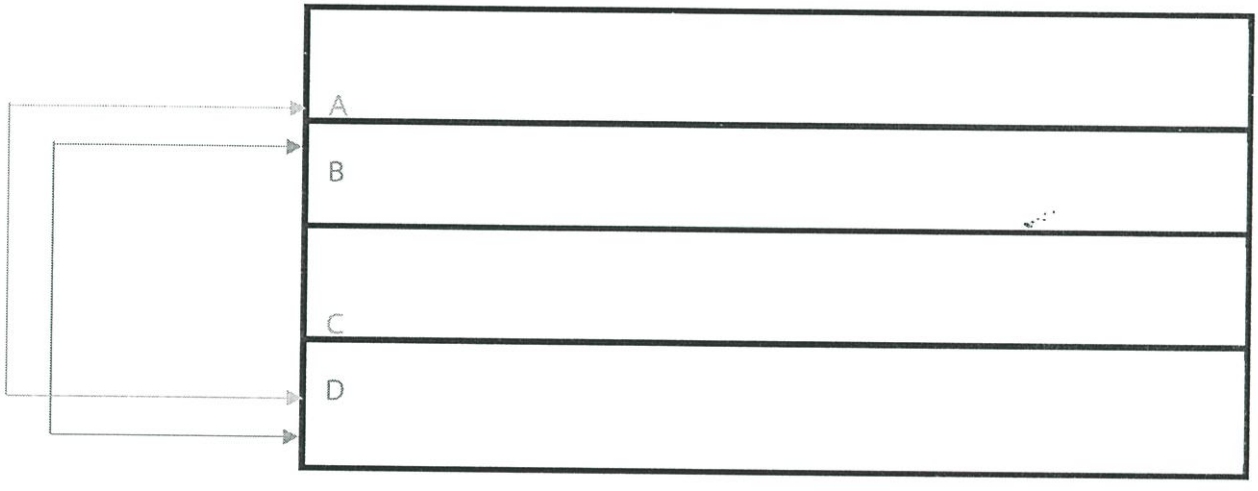
รูปที่ 3.12 แสดงด้านหน้าของแผ่นป้ายประชาสัมพันธ์

ประกอบไปด้วย LED ขนาด 4X4 จำนวน 320 ตัวโดยต่อฝั่งแถวแนวนอนจำนวน 8 ตัวฝั่งแถวจำนวน 40 ตัว ดังนั้นมีดวงไฟแนวนอนเท่ากับ 160 ดวง และแนวตั้งเท่ากับ 32 ดวง ซึ่งLED ทั้งหมดต่ออยู่กับแผ่น PCB ขนาดใหญ่ที่ไว้ควบคุมการแสดงผลของ LED



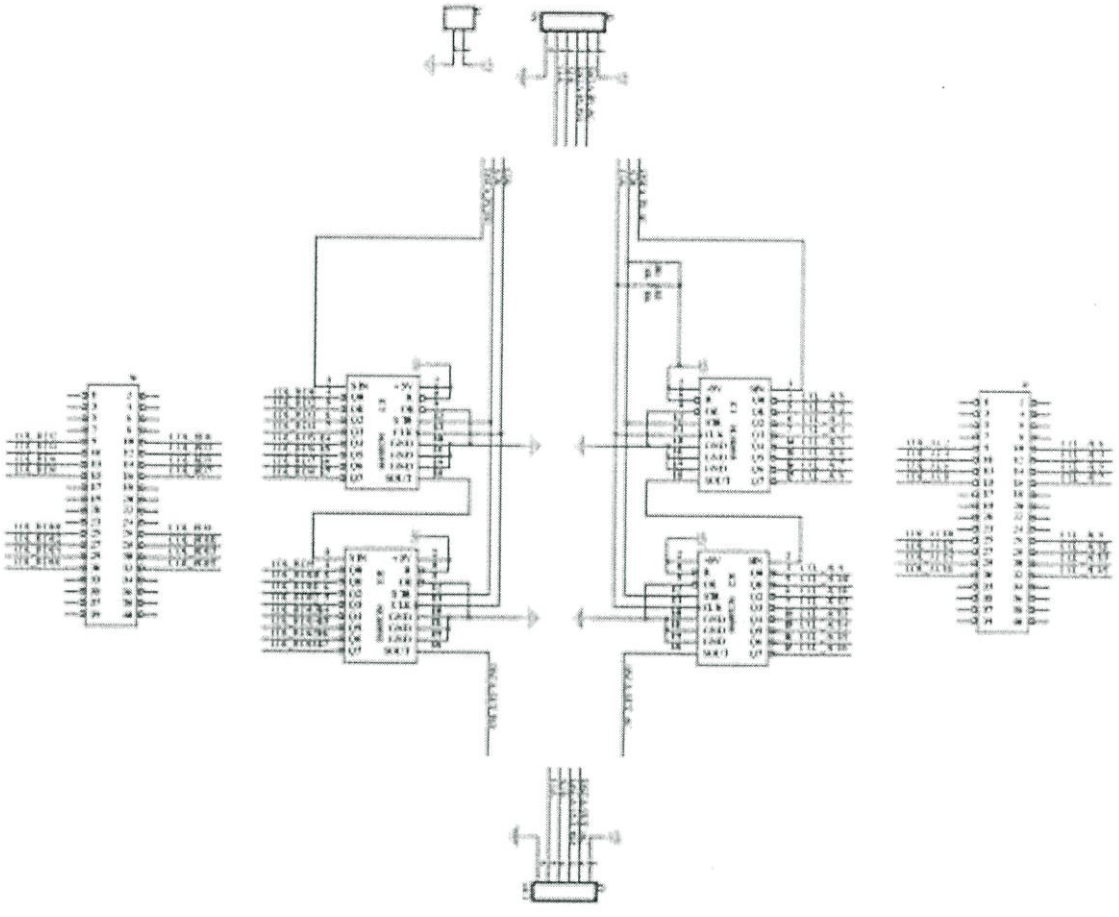
รูปที่ 3.13 แสดงวงจรด้านหลังของแผ่นป้ายประชาสัมพันธ์

โดยจะเห็นว่า PCB บอร์ดใหญ่ที่ติดอยู่กับตัว LED แบ่งเป็น 2 แผ่นมาต่อกัน และภายใน 1 ส่วนก็แบ่งเป็น2ส่วนย่อยอีก จึงมีทั้งหมด4ส่วนย่อย A B C D ซึ่งมีการต่อวงจรไฟสลับกันดังรูปภาพ ซึ่งภายในวงจรก็จะมี Slot เชื่อมถึงกัน ทั้งด้านข้างและด้านบนดังรูปภาพ โดยหลักการทำงานคือ LED จะติดพร้อมกันตามวงจรที่ต่อไว้เป็นคู่ คือ AB และ CD ตามลำดับโดยทั้งแผ่นป้ายถ้าดูจำนวนดวงไฟจะมีทั้ง 32x160 ดวง การทำงานหนึ่งครั้งสแกนเพียง 16x160 ดวง ซึ่งในการทำงานหนึ่งครั้งจะมีการสแกนพร้อมกัน2ครั้ง จึงเป็น 32x160 ดวง

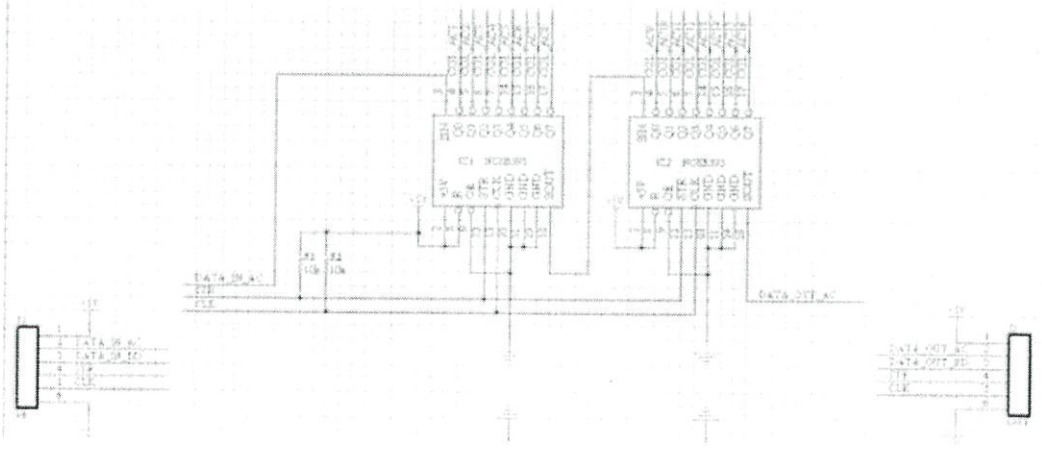


รูปที่ 3.14 แสดงแบบการต่อวงจรรวมภายใน

3.2.5 วงจรควบคุมแถวแนวตั้ง

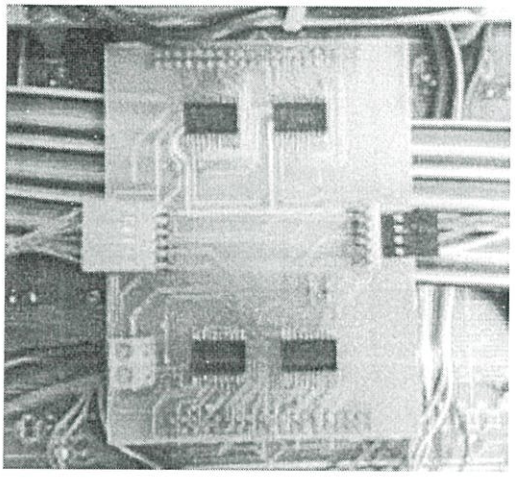


รูปที่ 3.15 แสดงวงจรควบคุมแถวแนวตั้ง

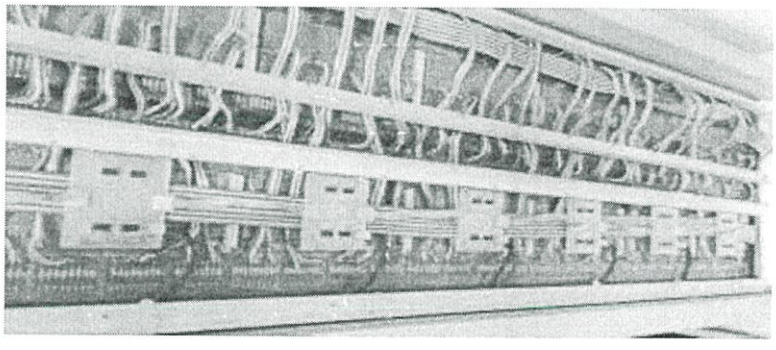


รูปที่ 3.16 แสดงวงจรควบคุมแถวแนวตั้ง

ข้อมูลจากตัว Arduino ภาคประมวลผล แบ่งเป็น DATA_IN_AC กับ DATA_IN_BD ซึ่งมาทางขาที่ 2 กับ 3 ตามลำดับซึ่ง DATA_IN_AC จะต่อไปที่ขาที่3 ของตัว NC6B595 (ตัวที่1)โดยจะมีการส่งข้อมูลจนครบ 8 บิตแล้วจึงจะส่งต่อไปที่ NC6B595 (ตัวที่2) จนครบทั้งหมด 16 บิต แต่มีจำนวนLED แนวนอนทั้งหมด 32 ดวงจึงต้องใช้ NC6B595 จำนวน 4 ตัว(หนึ่งตัวรับได้ 8 บิต) ต่อหนึ่งวงจรดังภาพ

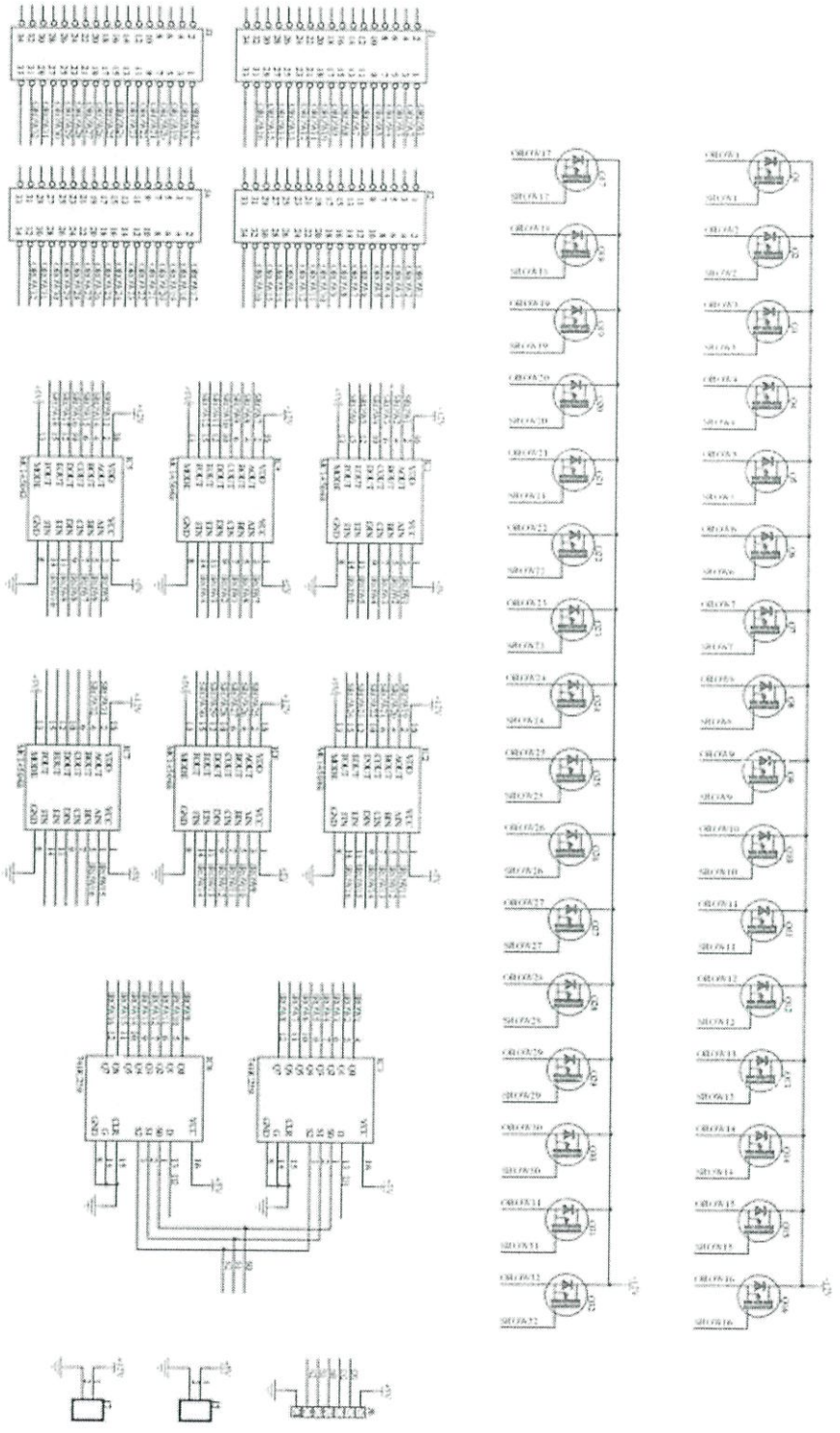


รูปที่ 3.17 แสดงชิ้นงานวงจรควบคุมแถวแนวตั้ง

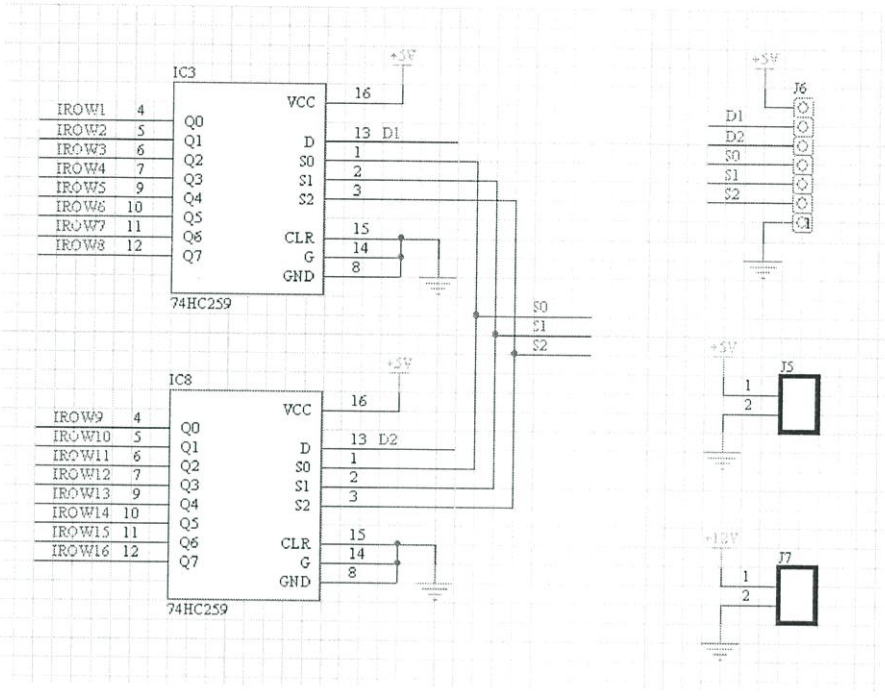


รูปที่ 3.18 แสดงชิ้นงานวงจรควบคุมแถวแนวตั้ง

3.2.6 วงจรควบคุมแถวแนวนอน

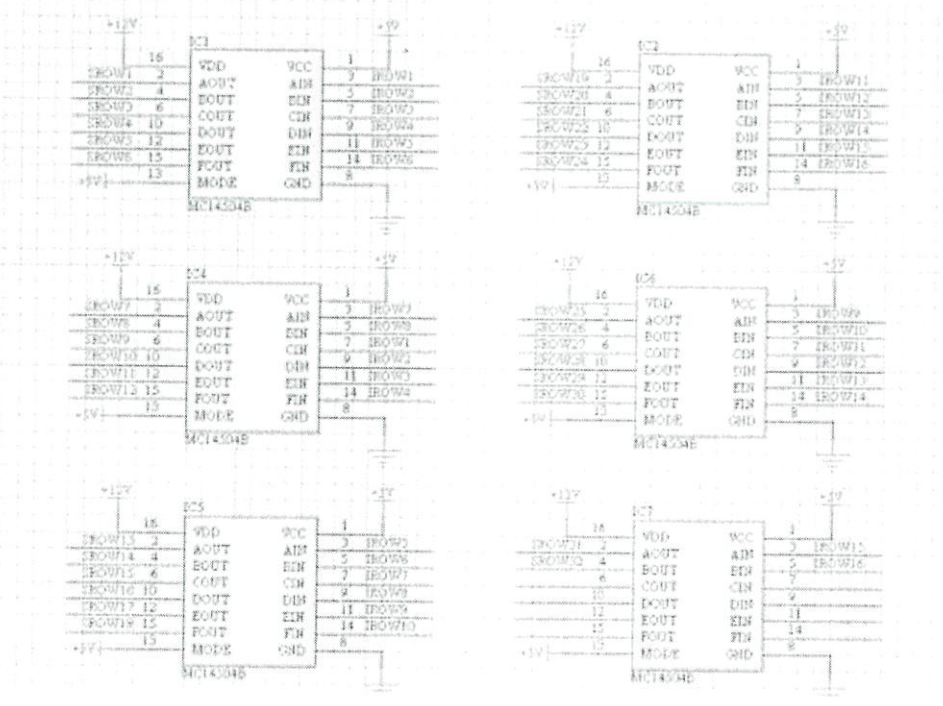


รูปที่ 3.19 แสดงวงจรควบคุมแถวแนวนอน



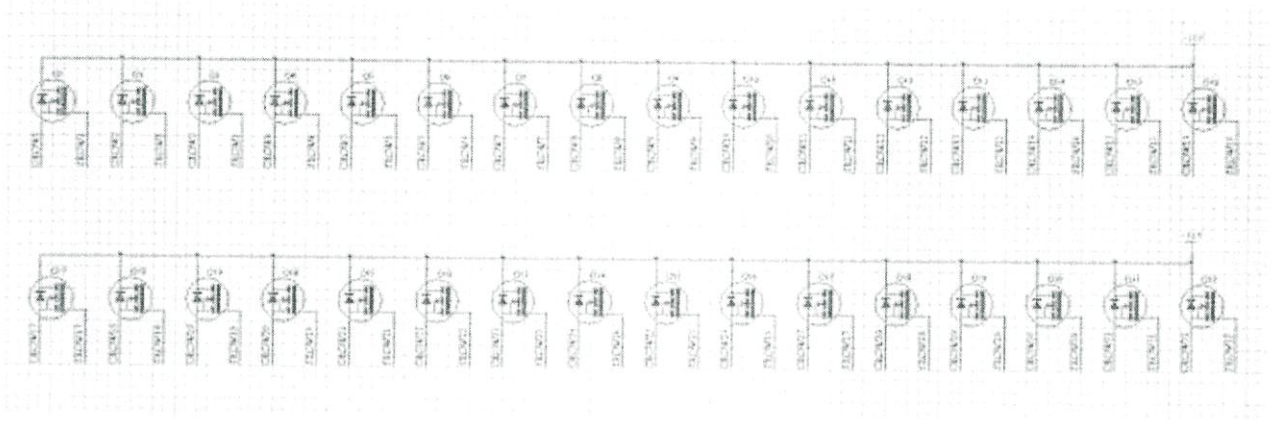
รูปที่ 3.20 แสดงวงจรควบคุมแถวแนวนอน

ข้อมูลจากตัว Arduino ภาคประมวลผล เข้ามาทางขา D1 และ D2 ตามลำดับโดยจะส่งข้อมูลมีละบิตจนครบ 8 บิตที่ 74HC259 ตัวที่หนึ่งจากนั้นจึงส่งข้อมูลต่อไปยัง 74HC259 ตัวที่สองจนครบทั้งหมด



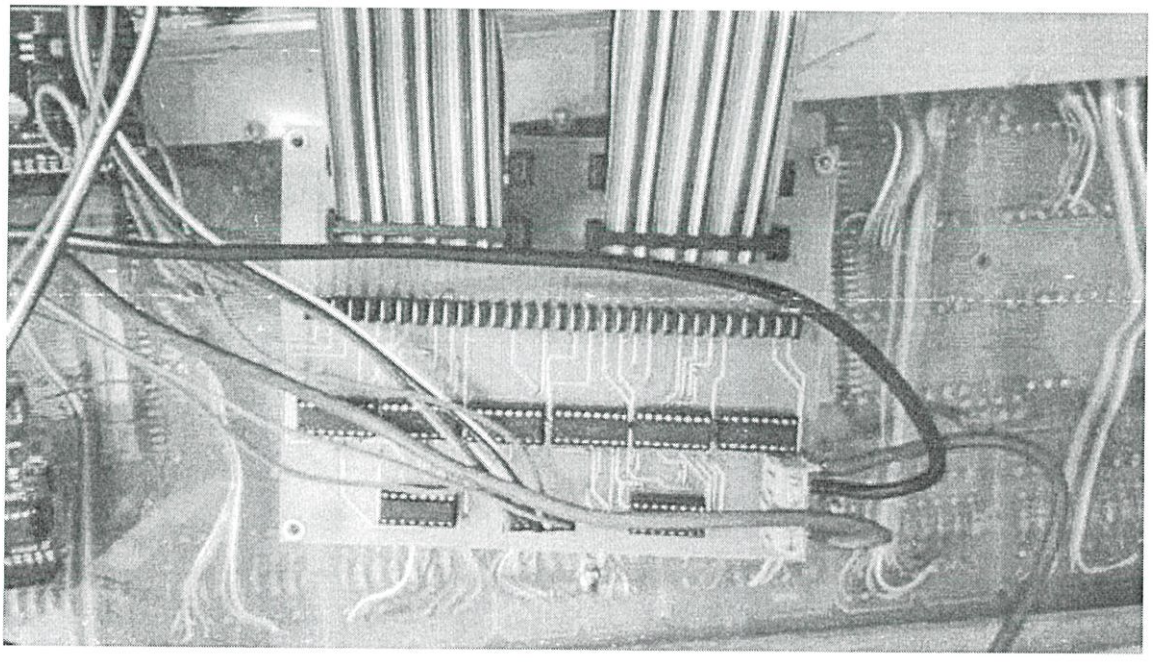
รูปที่ 3.21 แสดงวงจรควบคุมแถวแนวนอน

ข้อมูลทั้งหมดจากตัว 74HC259 จะถูกส่งเข้าตัว MC14504B ซึ่งเป็นตัว Shift Level เพื่อแปลง V จาก 5 เป็น 12



รูปที่ 3.22 แสดงวงจรควบคุมแถวแนวนอน

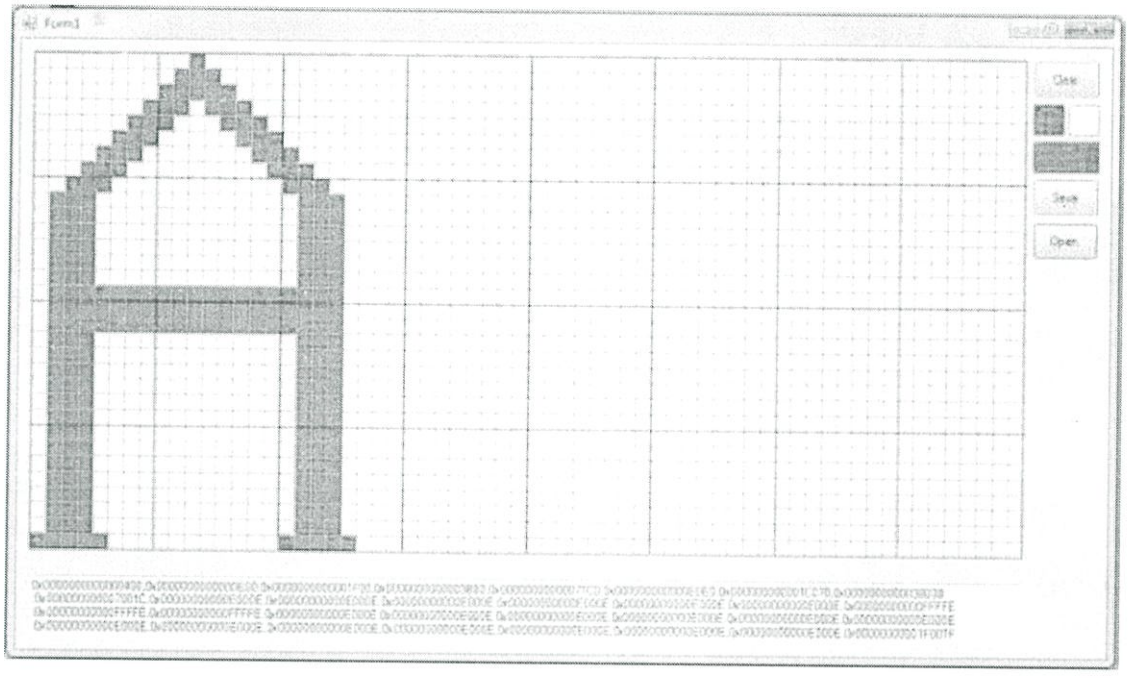
ตัวมอสเฟตซึ่งทำหน้าที่เป็นเหมือนสวิตซ์เปิดปิดการทำงานของวงจรเพื่อเพิ่มค่าความต่างศักย์กลายเป็น 12 V จะทำให้ตัวมอสเฟตปิดและเปิดให้กระแสไหล ทำให้วงจรนี้ทำงาน



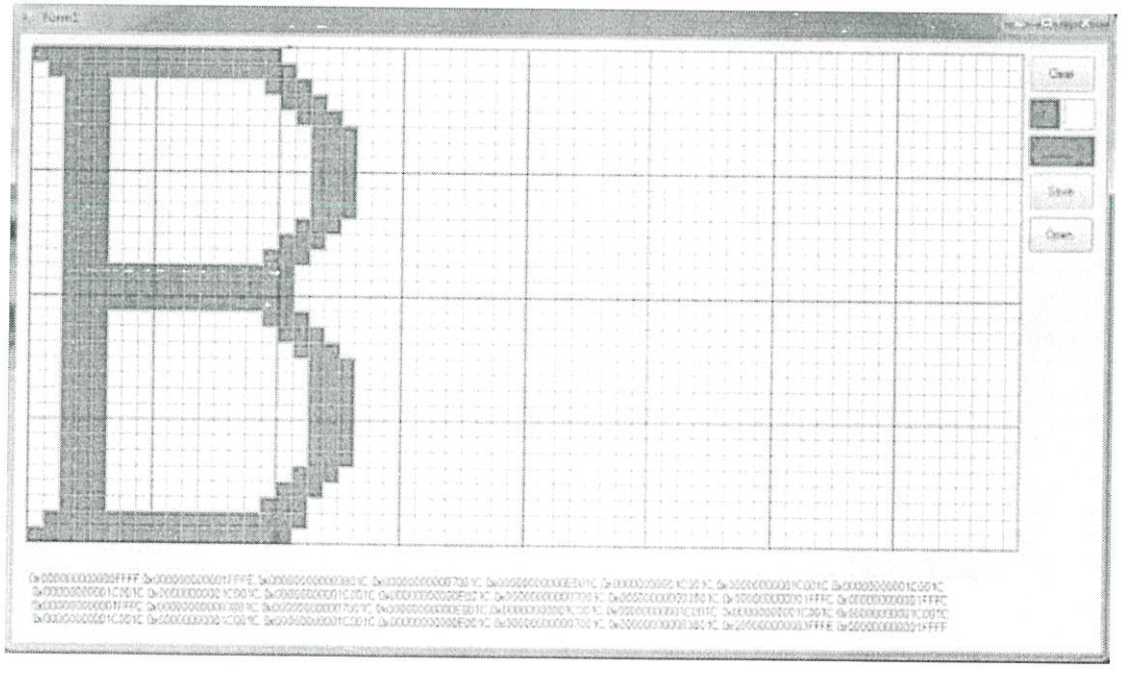
รูปที่ 3.23 แสดงชิ้นงานวงจรควบคุมแถวแนวนอน

3.3 การออกแบบอักษร

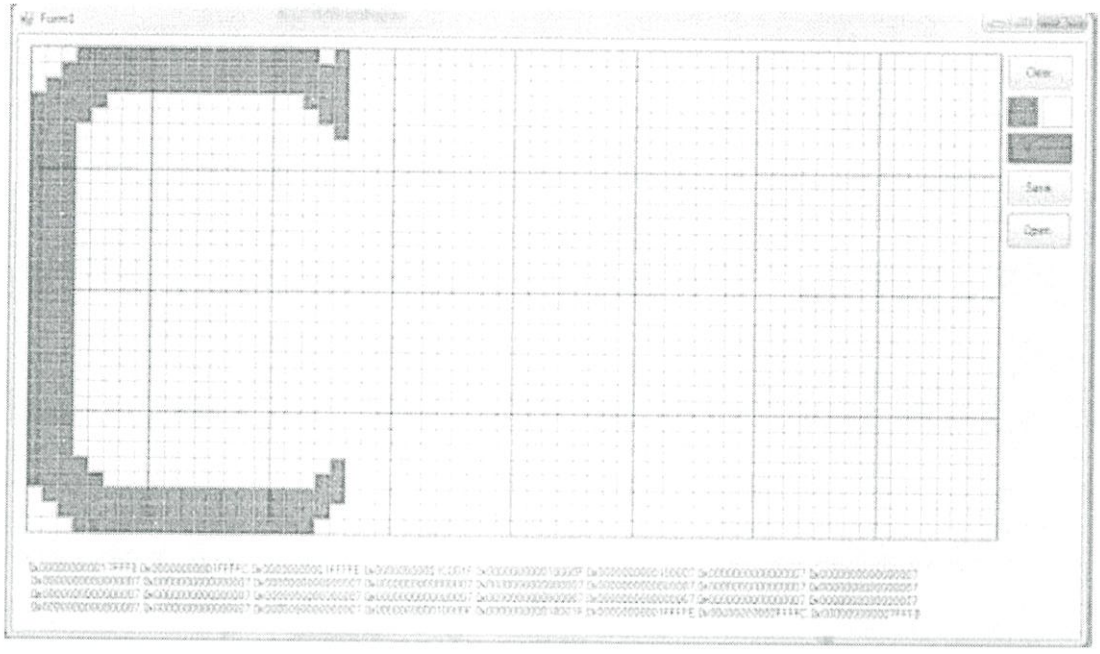
ใช้โปรแกรม Image2Lcd ในการออกแบบและสร้างอักษร



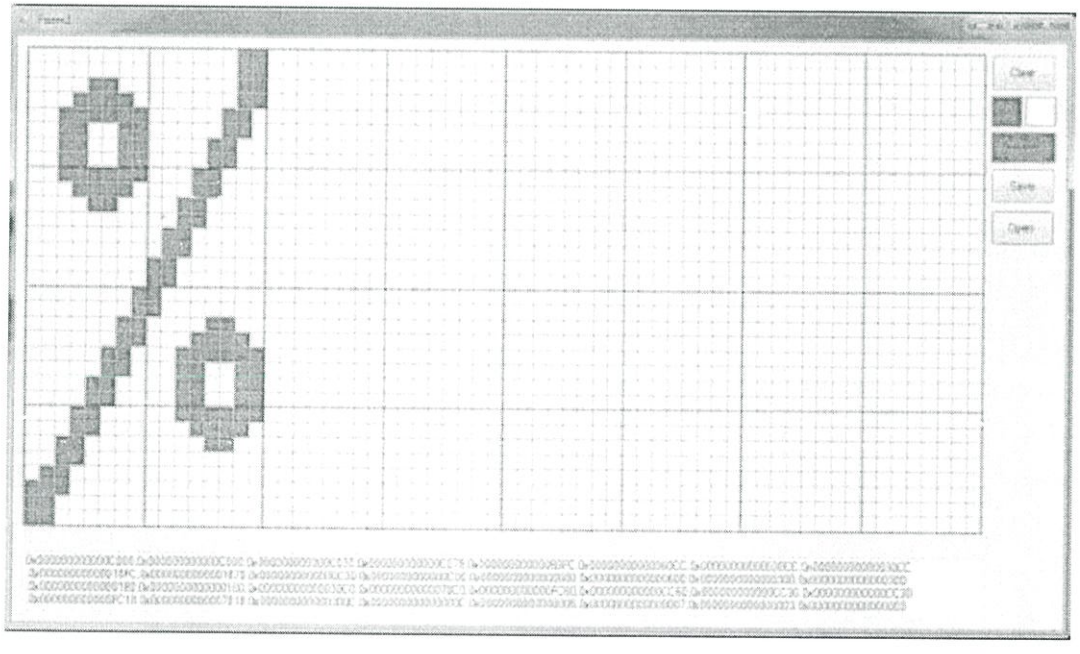
รูปที่ 3.24 การออกแบบอักษร A



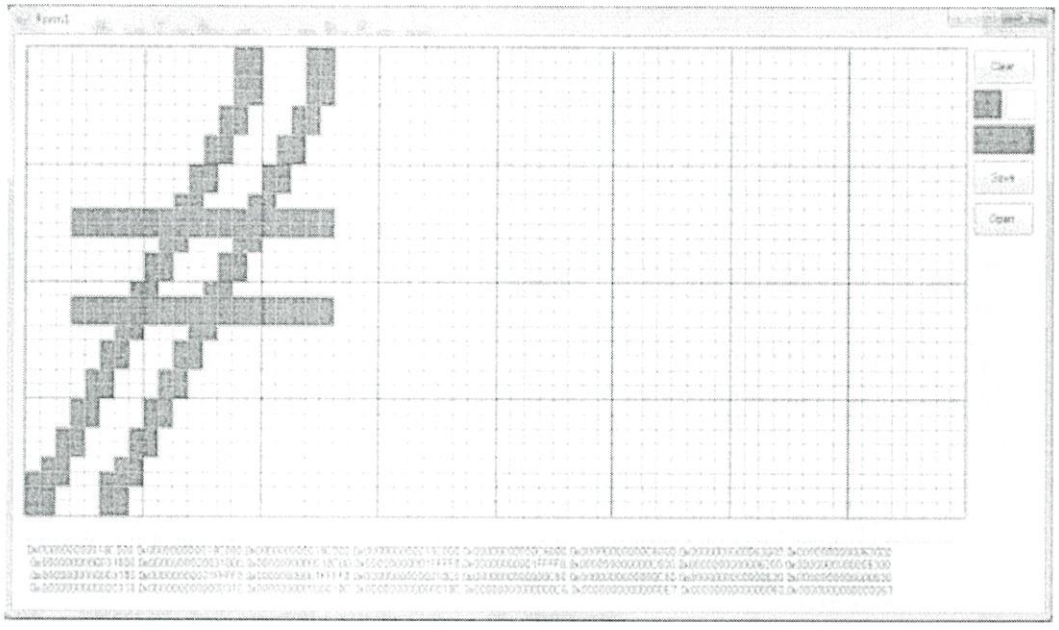
รูปที่ 3.25 การออกแบบอักษร B



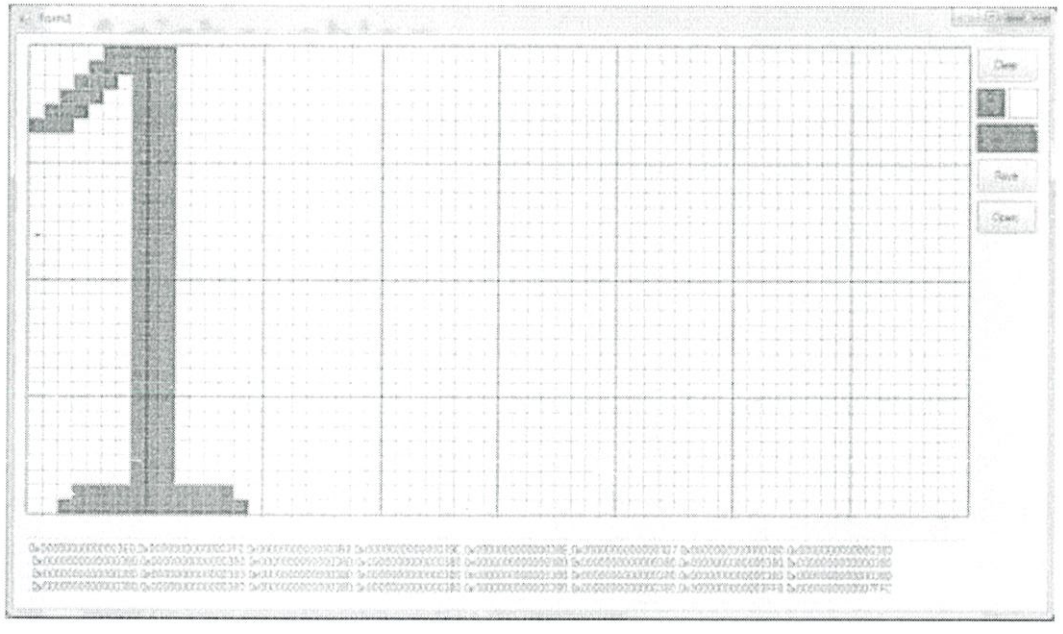
รูปที่ 3.26 การออกแบบอักษร C



รูปที่ 3.27 การออกแบบอักษร %



รูปที่ 3.28 การออกแบบอักษร #



รูปที่ 3.29 การออกแบบตัวเลข 1

จะได้ Virtual Code เป็นเลขฐานสิบหกด้านล่างที่สามารถนำไปใส่ในโปรแกรมของ Arduino เพื่อใช้ในการแสดงผลติดดับของหลอด LED ตามจุดที่สร้างไว้ โดยสร้างตัวอักษรตามตาราง ASCII CODE จนครบ

ตารางที่ 3.1 แสดงค่าของเลขฐานสองและอักขระ

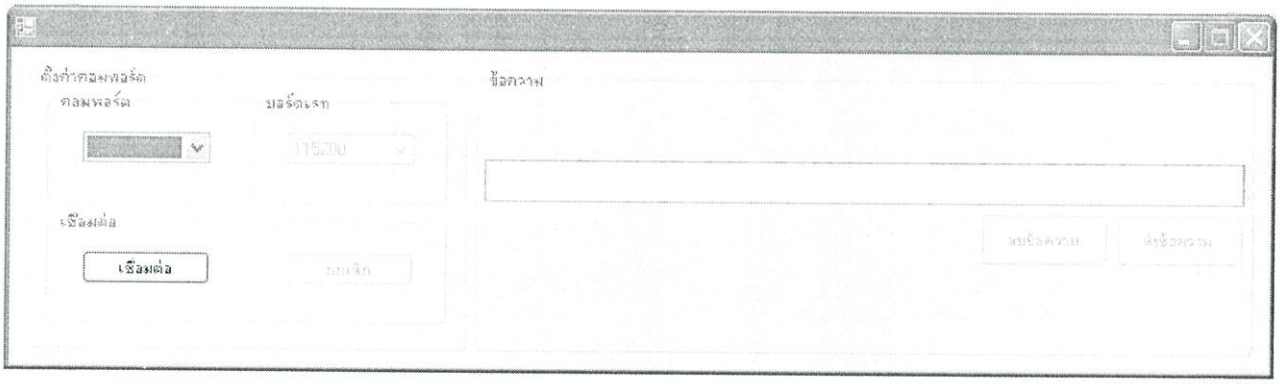
ฐานสอง	ฐานสิบ	ฐานสิบหก	อักขระ
0010 0000	32	20	(ช่องว่าง)
0010 0001	33	21	!
0010 0010	34	22	"
0010 0011	35	23	#
0010 0100	36	24	$
0010 0101	37	25	%
0010 0110	38	26	&
0010 0111	39	27	'
0010 1000	40	28	(
0010 1001	41	29	)
0010 1010	42	2A	*
0010 1011	43	2B	+
0010 1100	44	2C	,
0010 1101	45	2D	-
0010 1110	46	2E	.
0010 1111	47	2F	/
0011 0000	48	30	0
0011 0001	49	31	1
0011 0010	50	32	2
0011 0011	51	33	3
0011 0100	52	34	4
0011 0101	53	35	5
0011 0110	54	36	6
0011 0111	55	37	7
0011 1000	56	38	8
0011 1001	57	39	9
0011 1010	58	3A	:
0011 1011	59	3B	;
0011 1100	60	3C	<
0011 1101	61	3D	=
0011 1110	62	3E	>
0011 1111	63	3F	?

ฐานสอง	ฐานสิบ	ฐานสิบหก	อักขระ
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U
0101 0110	86	56	V
0101 0111	87	57	W
0101 1000	88	58	X
0101 1001	89	59	Y
0101 1010	90	5A	Z
0101 1011	91	5B	[
0101 1100	92	5C	\
0101 1101	93	5D	]
0101 1110	94	5E	^
0101 1111	95	5F	_

ฐานสอง	ฐานสิบ	ฐานสิบหก	อักขระ
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h
0110 1001	105	69	i
0110 1010	106	6A	j
0110 1011	107	6B	k
0110 1100	108	6C	l
0110 1101	109	6D	m
0110 1110	110	6E	n
0110 1111	111	6F	o
0111 0000	112	70	p
0111 0001	113	71	q
0111 0010	114	72	r
0111 0011	115	73	s
0111 0100	116	74	t
0111 0101	117	75	u
0111 0110	118	76	v
0111 0111	119	77	w
0111 1000	120	78	x
0111 1001	121	79	y
0111 1010	122	7A	z
0111 1011	123	7B	{
0111 1100	124	7C	|
0111 1101	125	7D	}
0111 1110	126	7E	~

ตัวอักษรทั้งหมดที่ออกแบบนั้นสามารถนำมาใช้แสดงผลบนบอร์ดได้ โดยแบ่งเป็น ตัวอักษรภาษาอังกฤษ ตัวเลข และ สัญลักษณ์

3.4 โปรแกรม



รูปที่ 3.30 หน้าต่างการทำงานของโปรแกรม

ประกอบด้วย 4 ส่วน

1. คอมพอร์ต

   เป็นการเลือกพอร์ตการเชื่อมต่อของ Arduino ฝั่งส่งที่ต่อกับคอมพิวเตอร์

2. บอร์ดเรท

   ความเร็วในการส่งข้อมูลในที่นี้ตั้งไว้คือ 115200 โดยในตัวโปรแกรมสามารถเลือกได้ตั้งแต่

   4800 ถึง 115200

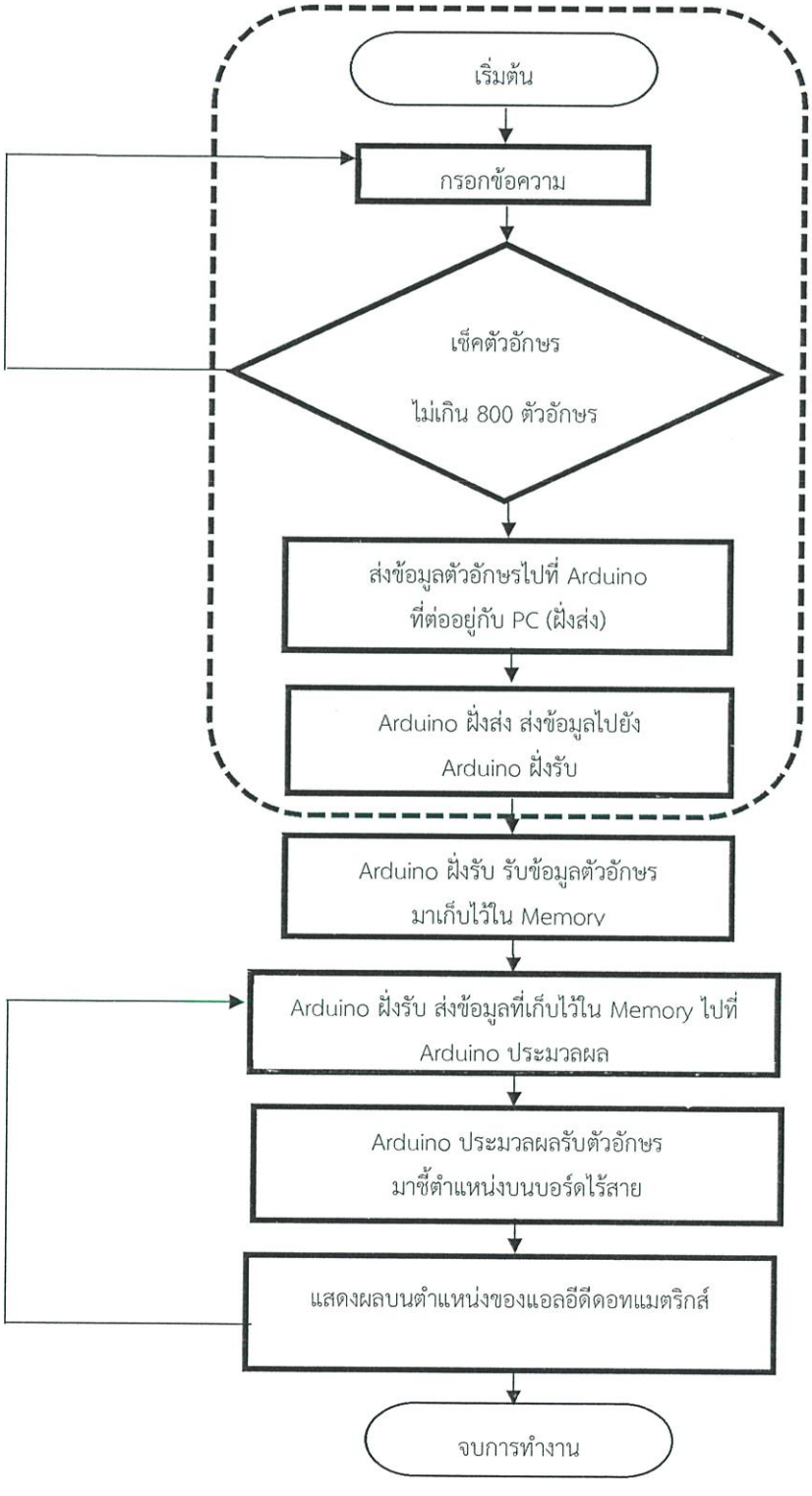
3. เชื่อมต่อ

   เอาไว้เชื่อมต่อตัวโปรแกรมกับ Arduino ฝั่งส่งที่ต่อกับคอมพิวเตอร์

4. ข้อความ

   เป็นกล่องข้อความเอาไว้พิมพ์ข้อความที่ต้องการลงไปซึ่งสามารถเพิ่มลบข้อความได้ตามต้องการก่อนที่จะส่ง

3.4.1 โฟลว์ชาร์ต



รูปที่ 3.31 โฟลว์ชาร์ตแสดงการทำงานของโปรแกรม

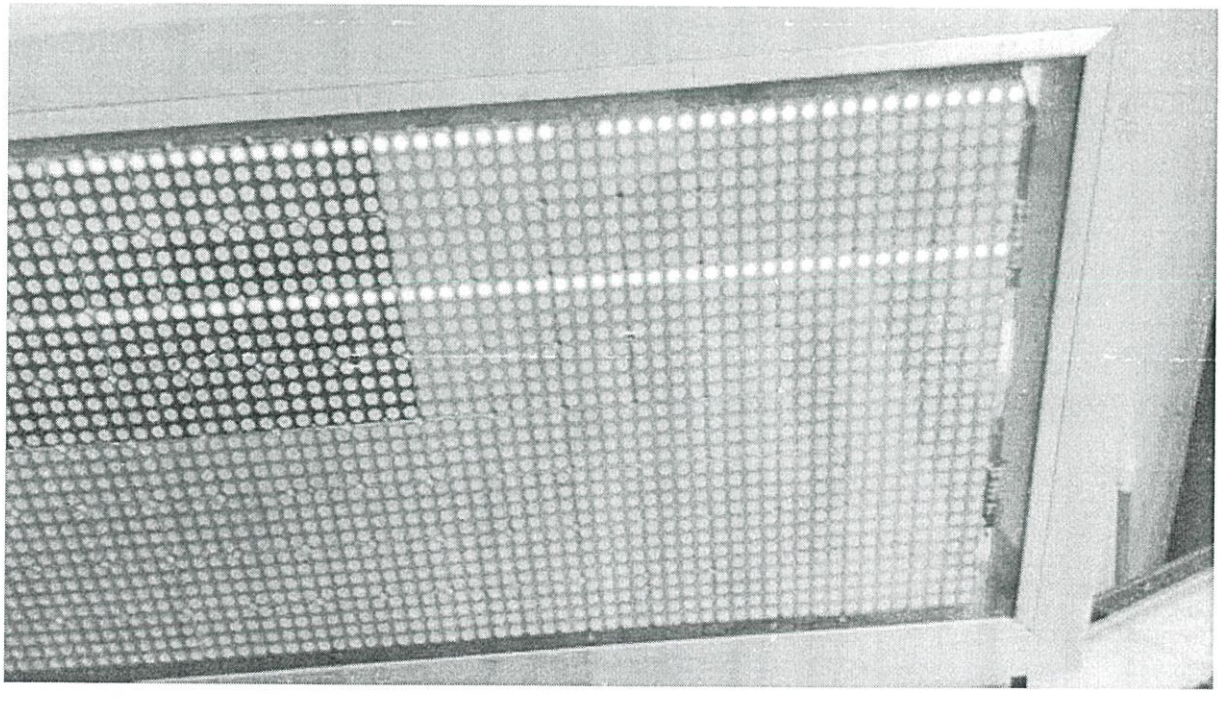
บทที่ 4
การทดลองและผลการทดลอง

ผลการทดลองแบ่งออกเป็น 4 ส่วนคือ

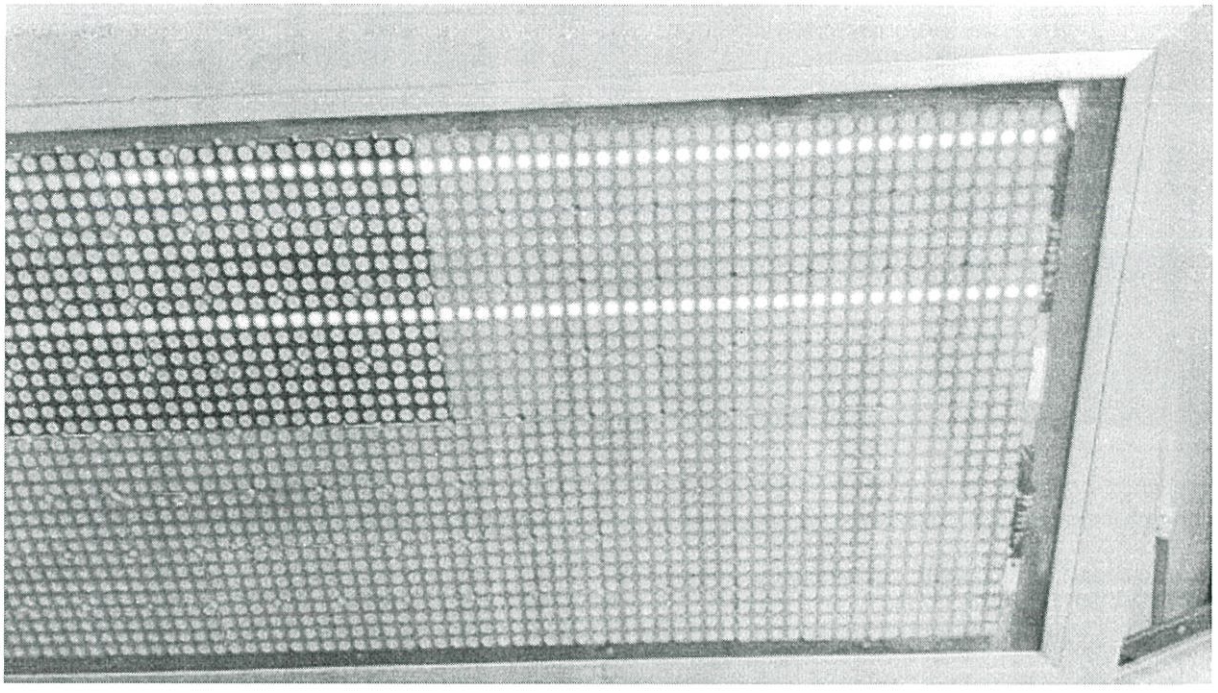
1. การแสดงผลแบบแถวแนวนอน
2. การแสดงผลแบบแถวแนวตั้ง
3. การแสดงผลตัวอักษร
4. ระยะการทำงาน

4.1 ผลการทดลองการแสดงผลแบบแถวแนวนอน

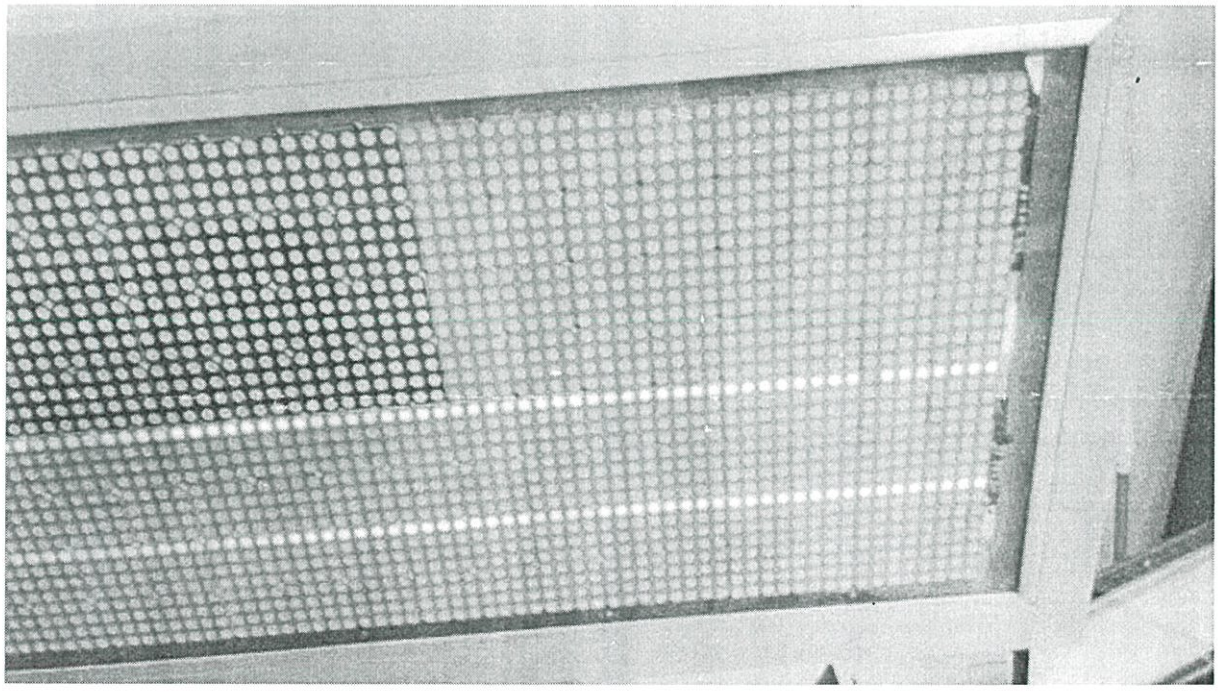
เมื่อเปิดการใช้งานโค้ดในการทดสองการแสดงผลแบบแถว หรือการสแกนแถว เพื่อทดสอบการทำงานของวงจรจะเห็นว่าในการทำงานหนึ่งรอบมีการสแกนแถวพร้อมกัน2ครั้ง โดยสแกนครั้งแรกเริ่มจากแถวที่ 1 และสแกนครั้งที่สอง เริ่มจากแถวที่ 9 โดยการสแกนครั้งที่หนึ่งจะสแกนไล่ไปทุกแถวตั้งแต่แถวที่ 1 จนถึงแถวที่ 8 และการสแกนครั้งที่สองจะสแกนเริ่มจากแถวที่ 9 จนถึงแถวที่ 16 จากนั้นการสแกนครั้งที่หนึ่งจะเริ่มใหม่แถวที่ 17 จนถึงแถวที่ 24 และการสแกนครั้งที่สองจะเริ่มใหม่แถวที่ 25 จนถึงแถวที่ 32 ทำให้สแกนแถวครบทั้งป้ายประชาสัมพันธ์ ซึ่งมีทั้งหมด 32แถว



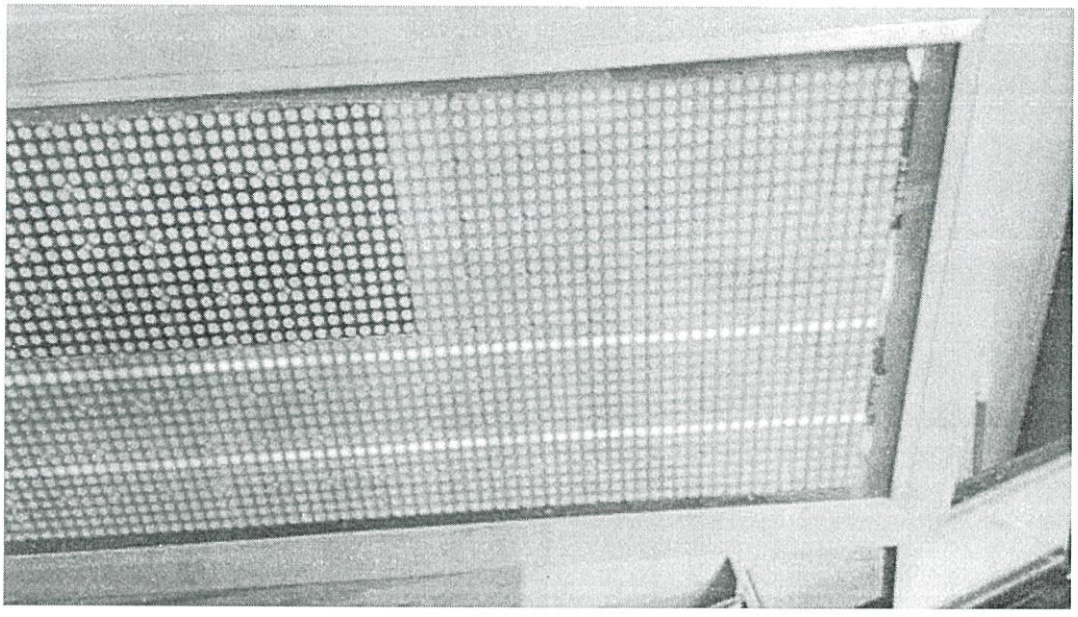
รูปที่ 4.1 แสดงการสแกนแถวแนวนอนที่ 1 และแถวแนวนอนที่ 9



รูปที่ 4.2 แสดงการสแกนแถวแนวนอนที่ 2 และแถวแนวนอนที่ 10



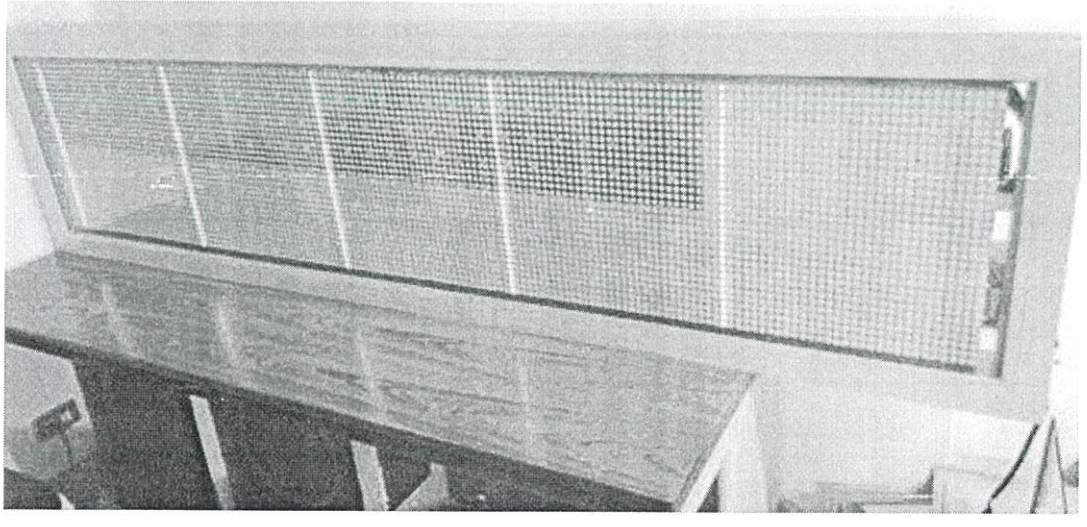
รูปที่ 4.3 แสดงการสแกนแถวแนวนอนที่ 17 และแถวแนวนอนที่ 25



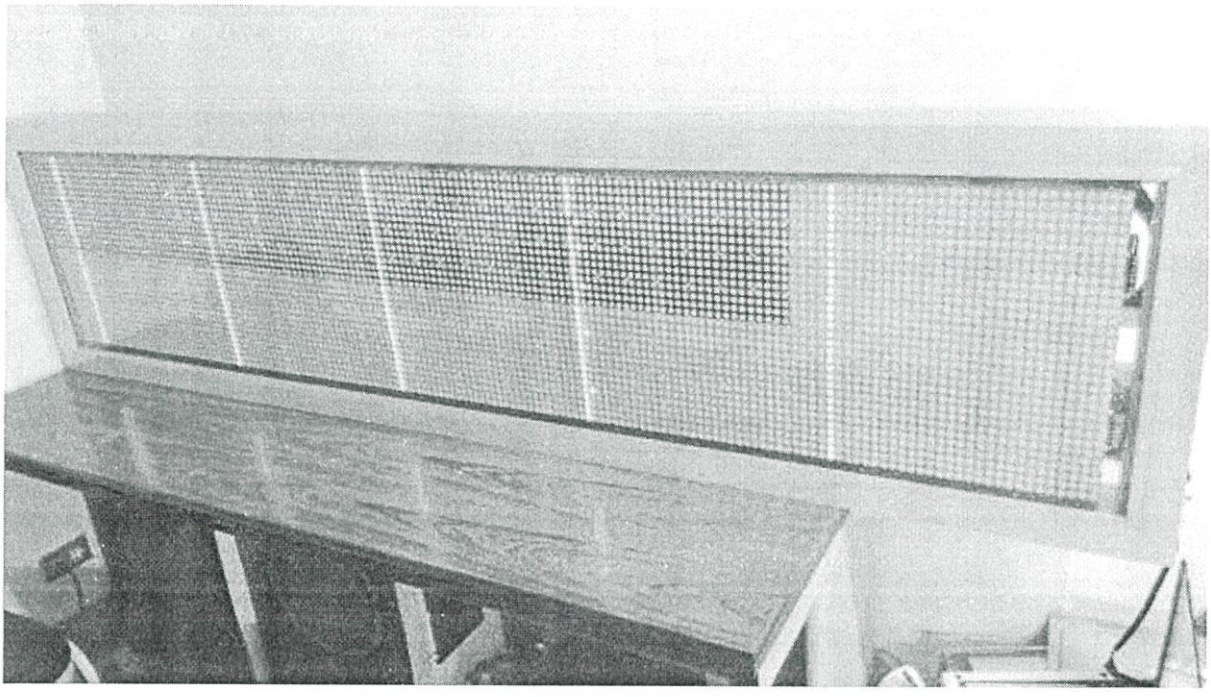
รูปที่ 4.4 แสดงการสแกนแถวแนวนอนที่ 18 และแถวแนวนอนที่ 26

4.2 ผลการทดลองการแสดงผลแบบแถวแนวตั้ง

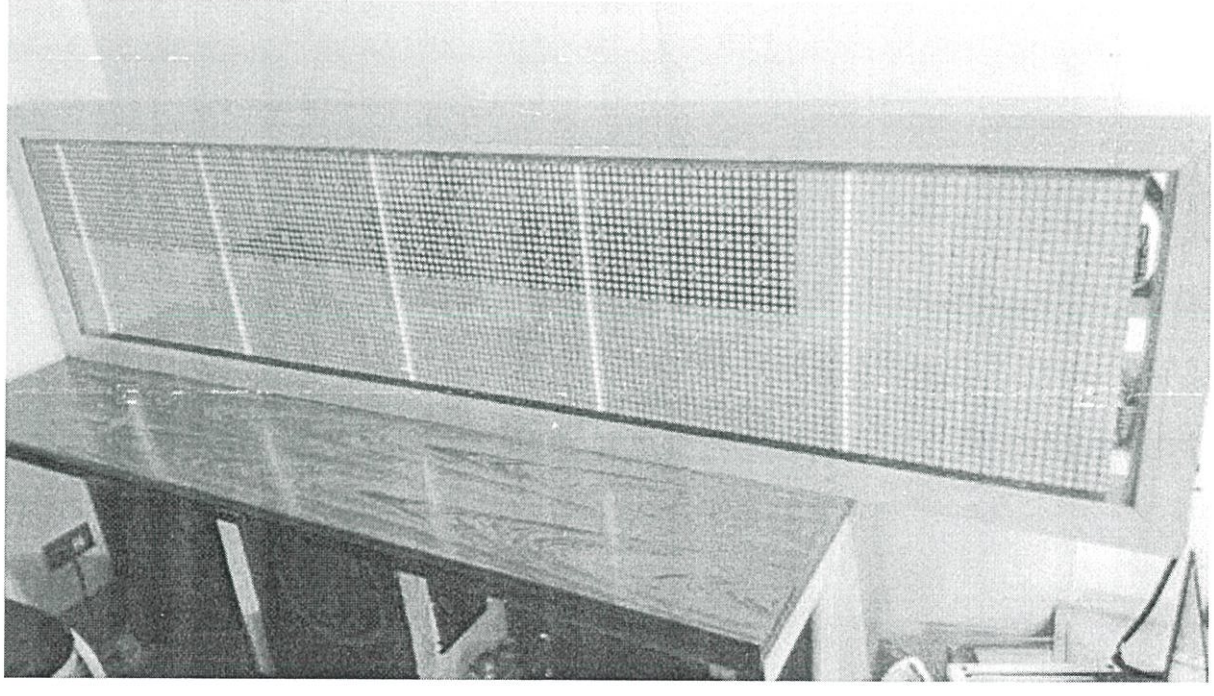
เมื่อเปิดการใช้งานโค้ดในการทดสองการแสดงผลแบบแถวแนวตั้งโดยในป้ายมีทั้งหมด 160 แถว ดังนั้นเพื่อความสะดวกรวดเร็วในการสแกนแถวแนวตั้ง ในการทำงานการแสดงผลแถวแนวตั้งหนึ่งครั้ง จึงให้มีการสแกนแถวแนวตั้งพร้อมกัน 5 แถว โดยสแกนไล่ทีละ 32 แถว โดยจะเห็นว่าบางตำแหน่งมีการช็อตกันของวงจรและมี LED บางดวงในแต่ละแถวแนวตั้งที่ไม่แสดงผล



รูปที่ 4.5 แสดงการการสแกนแถวแนวตั้ง (1)

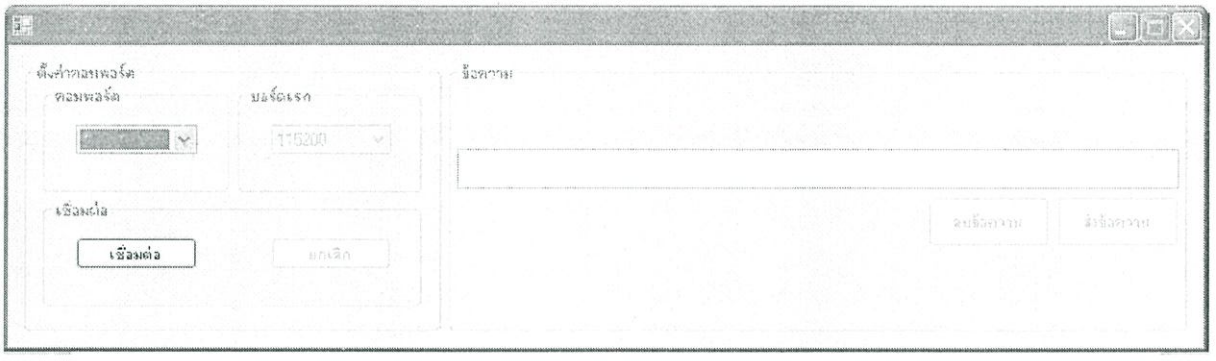


รูปที่ 4.6 แสดงการการสแกนแถวแนวตั้ง (2)



รูปที่ 4.7 แสดงการการสแกนแถวแนวตั้ง (3)

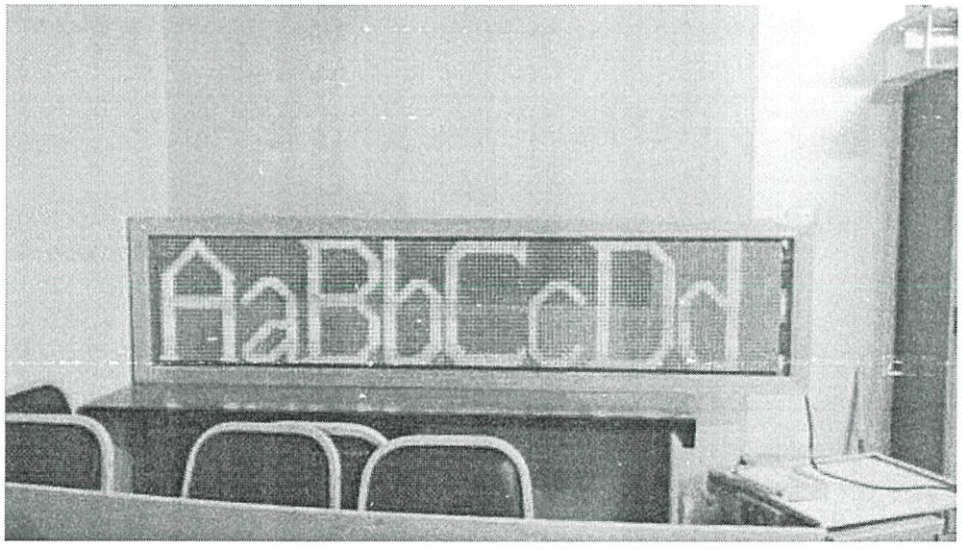
4.3 ผลการทดลองการแสดงผลแบบตัวอักษร



รูปที่ 4.8 แสดงหน้าต่างการทำงานของโปรแกรมส่งข้อความ

4.3.1 ตัวอักษรภาษาอังกฤษ

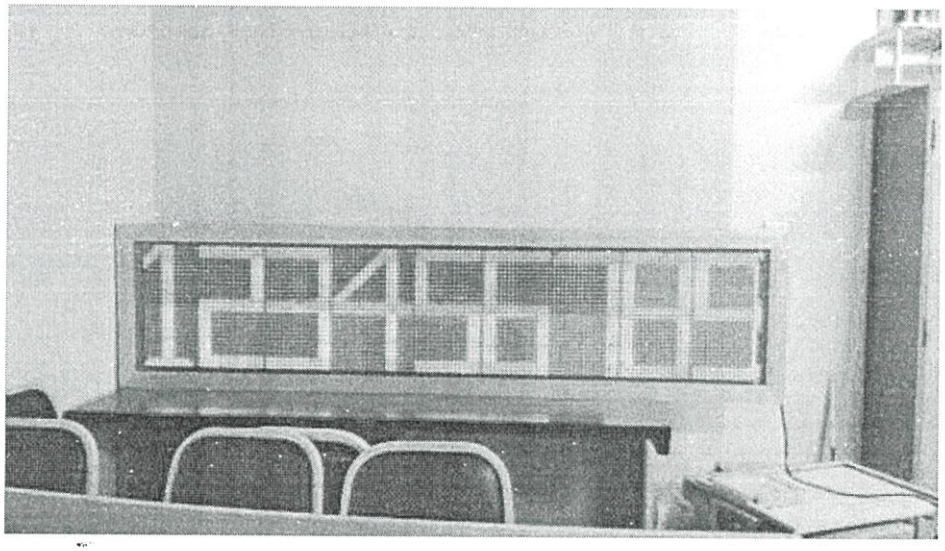
เมื่อส่งข้อความที่เป็นข้อมูลตัวอักษรภาษาอังกฤษผ่านโปรแกรม วิชวลเบสิกแผ่นป้ายประชาสัมพันธ์สามารถแสดงผลได้อย่างถูกต้องดังภาพ ซึ่งตัวอักษรภาษาอังกฤษสามารถรับได้ตั้งแต่ A ถึง Z จำนวน 26 ตัว



รูปที่ 4.9 ป้ายแสดงผลเป็นตัวอักษรภาษาอังกฤษ

4.3.2 ตัวเลข

เมื่อส่งข้อความที่เป็นข้อมูลตัวเลขผ่านโปรแกรม วิชวลเบสิกแผ่นป้ายประชาสัมพันธ์สามารถแสดงผลได้อย่างถูกต้องดังภาพ ซึ่งตัวเลขสามารถรับได้ตั้งแต่ 0 ถึง 9 จำนวน 10 ตัว



รูปที่ 4.10 ป้ายแสดงผลเป็นตัวเลข

4.3.3 สัญลักษณ์ตามตารางแอสกี

เมื่อส่งข้อความที่เป็นข้อมูลสัญลักษณ์ตามตารางแอสกี้ผ่านโปรแกรม วิชวลเบสิกโดยแผ่นป้ายประชาสัมพันธ์สามารถแสดงผลได้อย่างถูกต้องดังภาพ

รูปที่ 4.11 ป้ายแสดงผลเป็นสัญลักษณ์ตามตารางแอสกี

4.3.4 รวมทุกตัวอักขระ

เมื่อส่งข้อความที่เป็นข้อมูลรวมทุกตัวอักขระผ่านโปรแกรม วิชวลเบสิกแผ่นป้ายประชาสัมพันธ์สามารถแสดงผลได้อย่างถูกต้องดังภาพโดยมีตัวอักษรภาษาอังกฤษ ตัวเลข และสัญลักษณ์

รูปที่ 4.12 ป้ายแสดงผลรวมทุกตัวอักขระ

ตารางที่ 4.1 แสดงตัวอักษรทั้งหมดที่ป้ายประชาสัมพันธ์สามารถรับคำสั่งและแสดงผลได้

ฐานสอง	ฐานสิบ	ฐานสิบหก	อักขระ
0010 0000	32	20	(ช่องว่าง)
0010 0001	33	21	!
0010 0010	34	22	"
0010 0011	35	23	#
0010 0100	36	24	$
0010 0101	37	25	%
0010 0110	38	26	&
0010 0111	39	27	'
0010 1000	40	28	(
0010 1001	41	29	)
0010 1010	42	2A	*
0010 1011	43	2B	+
0010 1100	44	2C	,
0010 1101	45	2D	-
0010 1110	46	2E	.
0010 1111	47	2F	/
0011 0000	48	30	0
0011 0001	49	31	1
0011 0010	50	32	2
0011 0011	51	33	3
0011 0100	52	34	4
0011 0101	53	35	5
0011 0110	54	36	6
0011 0111	55	37	7
0011 1000	56	38	8
0011 1001	57	39	9
0011 1010	58	3A	:
0011 1011	59	3B	;
0011 1100	60	3C	<
0011 1101	61	3D	=
0011 1110	62	3E	>
0011 1111	63	3F	?

ฐานสอง	ฐานสิบ	ฐานสิบหก	อักขระ
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U
0101 0110	86	56	V
0101 0111	87	57	W
0101 1000	88	58	X
0101 1001	89	59	Y
0101 1010	90	5A	Z
0101 1011	91	5B	[
0101 1100	92	5C	\
0101 1101	93	5D	]
0101 1110	94	5E	^
0101 1111	95	5F	_

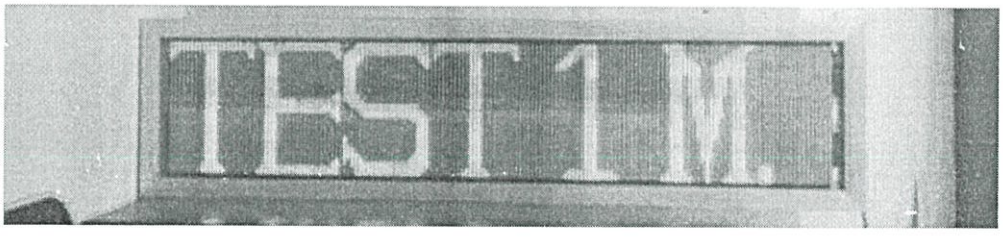
ฐานสอง	ฐานสิบ	ฐานสิบหก	อักขระ
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h
0110 1001	105	69	i
0110 1010	106	6A	j
0110 1011	107	6B	k
0110 1100	108	6C	l
0110 1101	109	6D	m
0110 1110	110	6E	n
0110 1111	111	6F	o
0111 0000	112	70	p
0111 0001	113	71	q
0111 0010	114	72	r
0111 0011	115	73	s
0111 0100	116	74	t
0111 0101	117	75	u
0111 0110	118	76	v
0111 0111	119	77	w
0111 1000	120	78	x
0111 1001	121	79	y
0111 1010	122	7A	z
0111 1011	123	7B	{
0111 1100	124	7C	|
0111 1101	125	7D	}
0111 1110	126	7E	~

4.4 ผลการทดลองระยะการทำงาน

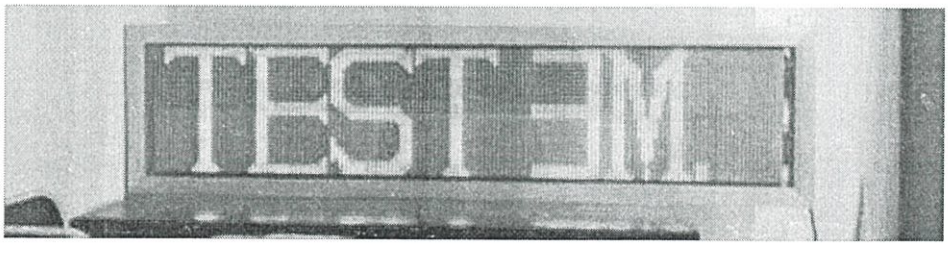
ตารางที่ 4.2 แสดงผลการทดลองระยะทางที่อุปกรณ์ภาครับสัญญาณสามารถรับข้อมูลได้

ระยะการส่งข้อความ (เมตร)	ระยะเวลาที่ได้รับข้อความ(วินาที)			
	ครั้งที่1	ครั้งที่2	ครั้งที่3	เฉลี่ย
1	0.12	0.11	0.12	0.12
3	0.16	0.16	0.14	0.15
5	0.24	0.26	0.25	0.25
10	0.52	0.58	0.54	0.55
20	0.81	0.81	0.83	0.82

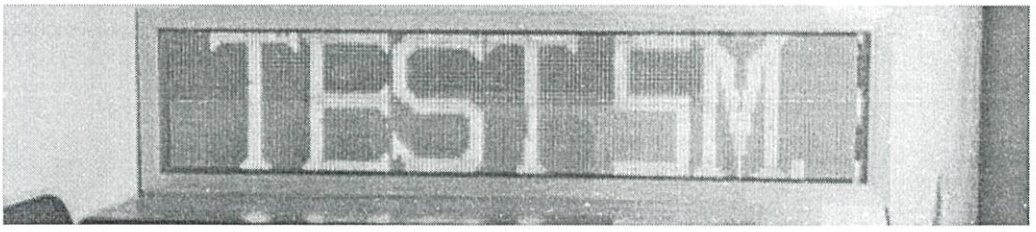
จากการทดลองทั้งหมด 3 ครั้งต่อระยะทางการส่งข้อความ 1 ระยะทางทั้งหมด 5 ระยะทางทำให้ทราบได้ว่าระยะทางมีผลต่อการแสดงผลของข้อความ กล่าวคือยิ่งระยะทางมากขึ้นระยะเวลาที่แผ่นป้ายประชาสัมพันธ์ได้รับข้อความก็ยิ่งมากขึ้นด้วย



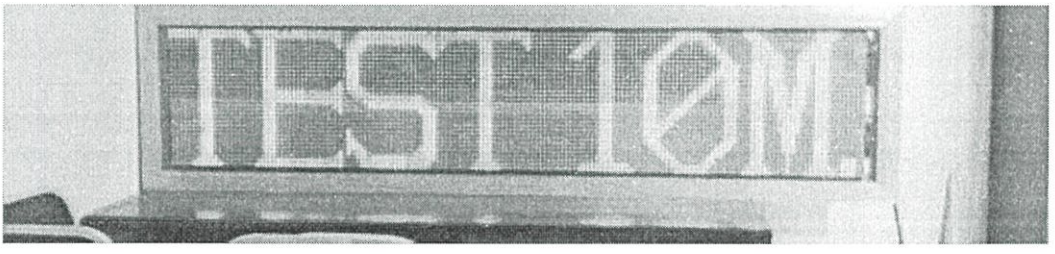
รูปที่ 4.13 แสดงผลการทดลองการส่งข้อความระยะ 1 เมตร



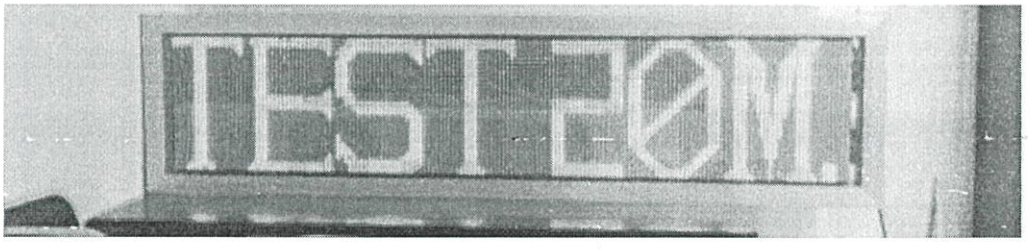
รูปที่ 4.14 แสดงผลการทดลองการส่งข้อความระยะ 3 เมตร



รูปที่ 4.15 แสดงผลการทดลองการส่งข้อความระยะ 5 เมตร



รูปที่ 4.16 แสดงผลการทดลองการส่งข้อความระยะ 10 เมตร



รูปที่ 4.17 แสดงผลการทดลองการส่งข้อความระยะ 20 เมตร

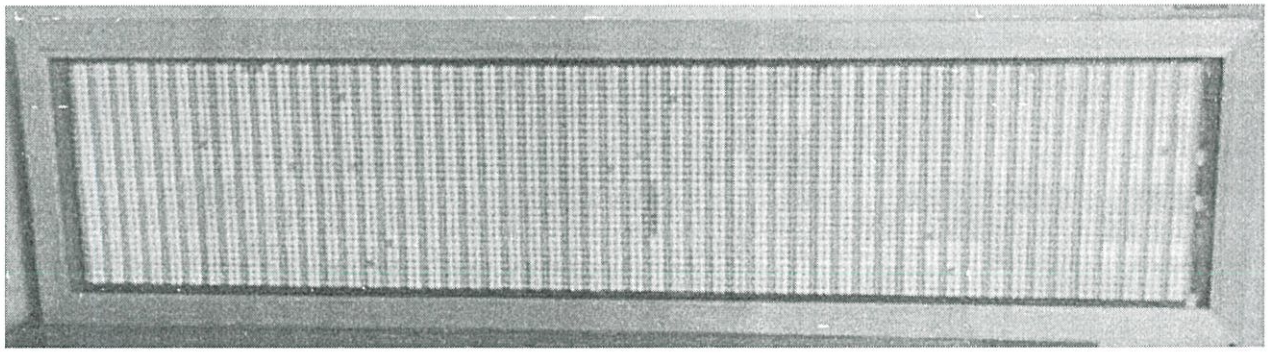
บทที่ 5
บทสรุปและวิจารณ์

5.1 สรุปผลการดำเนินงาน

โครงงานชิ้นนี้ได้นำเสนอเกี่ยวกับการจัดทำบอร์ดประชาสัมพันธ์ไร้สายที่มีการสั่งการด้วยโปรแกรมคอมพิวเตอร์ เพื่อความสะดวกในการติดตั้งบอร์ดนี้ในสถานที่ต่างๆได้อย่างสะดวก โดยความสะดวกของมันจะอยู่ที่เราไม่ต้องต่อสายใดๆจากคอมพิวเตอร์ไปที่ตัวป้ายเลย ซึ่งในระบบการทำงานของมันนั้นจะแบ่งออกเป็นสองส่วน ส่วนแรกคือที่ส่วนของตัวบอร์ดประชาสัมพันธ์ ภายในป้ายนั้นจะประกอบไปด้วยส่วนแสดงผลเป็นตัวแผง LED Dot Matrix ที่เรียงกันอยู่ขนาด 8x40มีไมโครคอนโทรลเลอร์ควบคุมการทำงานเป็น Arduino และมีส่วนเสริมในการใช้งานคือตัว nRF24L01p ซึ่งเป็นโมดูลที่ใช้ในการส่งข้อมูลแบบไร้สาย ส่วนที่สองคือในทางฝั่งของการสั่งงานจะเป็นโปรแกรมคอมพิวเตอร์ที่มีการสั่งงานไปที่ตัว nRF24L01pที่ต่อเข้ากับพอร์ท Serialของคอมพิวเตอร์ โดยการติดต่อระหว่างภาคส่งและภาครับนั้นจะอาศัยหลักการของระบบเครือข่ายไร้สายที่ความถี่ 2.4GHz

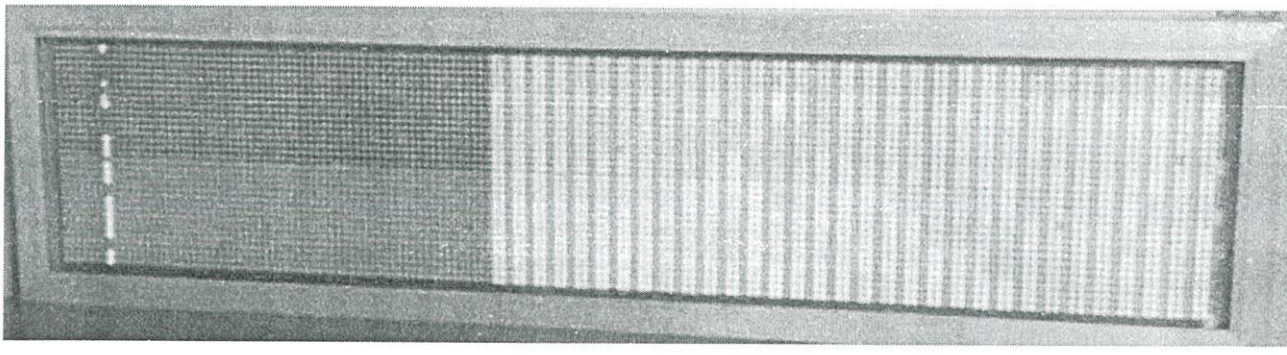
5.2 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

เนื่องจากเป็นชิ้นงานเก่าทำให้มีไฟบางดวงบนแผ่นป้ายที่ไม่สามารถใช้งานได้แล้วซึ่งจะเห็นเป็นจุดบอดและยังมีตำแหน่งที่เกิดการลัดวงจรดังภาพ



รูปที่ 5.1 แสดงจุดบอดที่ตำแหน่งต่างๆ บนแผ่นป้าย

แนวทางการแก้ไขปัญหานี้คือเปลี่ยนหลอดไฟเฉพาะดวงที่ไม่สามารถใช้งานได้แล้วซึ่งทำให้เกิดจุดบอดบนภาพ แต่เนื่องจากหลอดไฟ LED 1อันเป็นแบบ 4 X 4 ทำให้ต้องเปลี่ยนใหม่ทั้งหมด ทำให้ใช้งบประมาณที่ค่อนข้างสูง



รูปที่ 5.2 แสดงจุดที่มีการลัดวงจรของแผ่นป้าย

แนวทางการแก้ไขปัญหานี้คือเปลี่ยนวงจรด้านหลังที่เป็นวงจรสำเร็จที่ต่อกับ LED ขนาด 4X4 ซึ่งอาจจะต้องเปลี่ยนทั้งวงจรซึ่งใช้งบประมาณมากอีกเช่นกัน

5.3 แนวทางการพัฒนาต่อไป

โครงงานนี้สามารถพัฒนาต่อไปได้หลายรูปแบบทั้งในเรื่องของการพัฒนารูปแบบตัวอักษรในการแสดงผล หรือรองรับในการแสดงผลภาษาหลากหลายขึ้น และยังสามารถพัฒนาในด้านของระยะการรับส่งข้อมูลให้มีระยะที่ไกลขึ้น มีความยืดหยุ่นในการทำงานมากขึ้น โดยจุดประสงค์หลักในการพัฒนาเพิ่มเติมนั่นคือเพิ่มขีดความสามารถที่หลากหลายและความสะดวกสบายให้มากขึ้นนั่นเอง

บรรณานุกรม

[1] Arduino. [ออนไลน์].
เข้าถึงได้จาก : http://www.arduino.cc/en/Main/ArduinoBoard (23 เมษายน 2558)
[2] บทความ "ตัวอย่างการควบคุมไฟ LED ด้วย Arduino". [ออนไลน์].
เข้าถึงได้จาก : http://goo.gl/a8NBHs (23 เมษายน 2558)
[3] ฤทธิ์ ธีระโกเมน. (ม.ป.ป.). โซลินอยด์.
เข้าถึงได้จาก : http://goo.gl/29PQuJ (23 เมษายน 2558)
[4] nRF24L01p โมดูลไร้สาย. [ออนไลน์]. เข้าถึงได้จาก : http://goo.gl/SMcCJW (23 เมษายน 2558)
[5] Doungjai Khuntasamon. (2556). อุปกรณ์คอมพิวเตอร์.
เข้าถึงได้จาก : http://goo.gl/kjXTOJ (24 เมษายน 2558)
[6] ภาษา C. (2556). เข้าถึงได้จาก : http://guru.sanook.com/6394/ (24 เมษายน 2558)
[7] ภาษาซีพลัสพลัส. [ออนไลน์]. เข้าถึงได้จาก : http://goo.gl/Bqa5Ja (24 เมษายน 2558)
[8] IIS คืออะไร. (2557). เข้าถึงได้จาก : http://goo.gl/q2oCL3 (25 เมษายน 2558)
[9] ISAREEYA KEATWUTTIKAN. (2557). โปรแกรม ภาษา VISUAL BASIC คืออะไร.
เข้าถึงได้จาก : http://goo.gl/TjrAbl (26 เมษายน 2558)
[10] ประยุกต์ใช้ LED ในชีวิตประจำวัน (2556).
เข้าถึงได้จาก : https://www.blognone.com/node/49398 (27 เมษายน 2558)
[11] VB พื้นฐาน. [ออนไลน์].
เข้าถึงได้จาก : http://www.men9ch.com/?page_id=561 (27 เมษายน 2558)
[12] การเขียนโปรแกรมคอมพิวเตอร์. [ออนไลน์].
เข้าถึงได้จาก : http://www.lks.ac.th/anchalee/cindex-n.htm

ภาคผนวก

โค้ด ArduinoEeprom ฝั่งรับ

```
//

#include <SPI.h>

#include <EEPROM.h>

#include <nRF24L01P.h>

//

nRF24L01p RF(7,8);//CSN,CE

//

String textFont;

//

int LED = 6;

//

////////////////////////////////////////////////////////////////////////////////

void readData()

{

  for(int i = 0; i < 900; i ++)

  {

    char d = EEPROM.read(i);


    if(d == '\r') return;
```

```
    else textFont += d;
  }
}
////////////////////////////////////////////////////////////////////////////////////////
void saveData()
{
  int len = textFont.length();

  for(int i = 0;i < len;i ++)
  {
    EEPROM.write(i,(char)textFont[i]);
  }
  EEPROM.write(len,0x0d);
}
////////////////////////////////////////////////////////////////////////////////////////
void sendDataText(boolean s)
{
  digitalWrite(LED,HIGH);
  if(textFont == "") return;
  Serial.print(textFont);
  delay(100);
```

```
  Serial.print("\r");
  if(s == true) saveData();
  textFont = "";
  delay(200);
  digitalWrite(LED,LOW);
}
////////////////////////////////////////////////////////////////////////////////
boolean getDataText()
{
  if(RF.available())
  {
    String BUFRF = "";
    RF.read();
    RF.rxPL(BUFRF);

    if(BUFRF == "\r") return true;
    else textFont += BUFRF;
  }
  return false;
}
////////////////////////////////////////////////////////////////////////////////
```

```
void setup()
{
  delay(2000);
  Serial.begin(115200);
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  RF.channel(90);
  //RF.TXaddress("KCC");
  RF.RXaddress("KCC");
  RF.dataRate(250);
  RF.init();

  pinMode(LED,OUTPUT);

  textFont = "";

  readData();
  sendDataText(false);
}
////////////////////////////////////////////////////////////////////////////////
void loop()
```

```
{
  boolean sOk = getDataText();

  if(sOk == true) sendDataText(true);
}
```

โค้ด Arduino ฝั่งส่ง

```
//
#include <SPI.h>
#include <nRF24L01P.h>
//
nRF24L01p RF(7,8);//CSN,CE
//
String textFont;
//
////////////////////////////////////////////////////////////////////////////////
void sendRf()
{
  String BUFRF;

  int len = textFont.length();
```

```
int lenS = len;
int countLen = 0;

while(true)
{
  if(lenS > 20)
  {
    BUFRF = textFont.substring(countLen,countLen + 20);
    RF.txPL(BUFRF);
    RF.send(FAST);
    countLen = countLen + 20;

    lenS = lenS - 20;
  }
  else
  {
    BUFRF = textFont.substring(countLen,len);
    RF.txPL(BUFRF);
    RF.send(FAST);
    delay(100);
    RF.txPL("\r");
```

```
      RF.send(FAST);
      textFont = "";
      return;
    }
  }
}
////////////////////////////////////////////////////////////////////////////////
boolean getDataText()
{
  if (Serial.available())
  {
    char d = Serial.read();

    if(d == '\r')
    {
      Serial.println("OK");
      return true;
    }
    else if(d >= 0x20 && d <= 0x7f)
    {
      textFont += d;
```

```
    }
  }
  return false;
}
////////////////////////////////////////////////////////////////////////////////////////
void setup()
{
  Serial.begin(115200);
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  RF.channel(90);
  RF.TXaddress("KCC");
  //RF.RXaddress("KCC");
  RF.dataRate(250);
  RF.init();

  textFont = "";
}
////////////////////////////////////////////////////////////////////////////////////////
void loop()
{
```

```
  boolean sOk = getDataText();


  if(sOk == true) sendRf();
}
```

โค้ด Arduino ฝั่งส่ง

```
//
#include <SPI.h>
#include <nRF24L01P.h>
//
nRF24L01p RF(7,8);//CSN,CE
//
String textFont;
//
////////////////////////////////////////////////////////////////////////////////
void sendRf()
{
  String BUFRF;


  int len = textFont.length();


  int lenS = len;
```

```
int countLen = 0;


while(true)
{
 if(lenS > 20)
 {
   BUFRF = textFont.substring(countLen,countLen + 20);
   RF.txPL(BUFRF);
   RF.send(FAST);
   countLen = countLen + 20;


   lenS = lenS - 20;
 }
 else
 {
   BUFRF = textFont.substring(countLen,len);
   RF.txPL(BUFRF);
   RF.send(FAST);
   delay(100);
   RF.txPL("\r");
   RF.send(FAST);
```

```
      textFont = "";
      return;
    }
  }
}
////////////////////////////////////////////////////////////////////////////////
boolean getDataText()
{
  if (Serial.available())
  {
    char d = Serial.read();

    if(d == '\r')
    {
      Serial.println("OK");
      return true;
    }
    else if(d >= 0x20 && d <= 0x7f)
    {
      textFont += d;
    }
```

```
  }
  return false;
}
////////////////////////////////////////////////////////////////////////////////////////
void setup()
{
  Serial.begin(115200);
  SPI.begin();
  SPI.setBitOrder(MSBFIRST);
  RF.channel(90);
  RF.TXaddress("KCC");
  //RF.RXaddress("KCC");
  RF.dataRate(250);
  RF.init();

  textFont = "";
}
////////////////////////////////////////////////////////////////////////////////////////
void loop()
{
  boolean sOk = getDataText();
```

```
  if(sOk == true) sendRf();
}
```

โค้ด Arduino ประมวลผล

```
#include <Arduino.h>
#include <avr/pgmspace.h>
//
#define  MAXDIGIT 5  //32 x 5 = 160 bit
//
int    D1    =  22;
int    D2    =  23;
int    S0    =  24;
int    S1    =  25;
int    S2    =  26;

int    INAC  =  27;
int    INBD  =  28;
int    STR   =  29;
int    CLK   =  30;
//
```

```
PROGMEM prog_uint32_t FONT_000[33] = {8
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000
000,0x00000000,0x00000000,0x00000000};//

PROGMEM prog_uint32_t FONT_001[33] =
{22,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x0000000
7,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,
0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x
00000007,0x00000007,0x00000007,0x00000000,0x00000000,0x00000000,0x00000000,0x00
000000,0x00000007,0x00000007,0x00000007};//!

PROGMEM prog_uint32_t FONT_002[33] = {9
,0x000000E7,0x000000E7,0x000000E7,0x000000E7,0x00000000,0x00000000,0x00000000,0x
00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00
000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000
0000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000000
00,0x00000000,0x00000000,0x00000000};//"

PROGMEM prog_uint32_t FONT_003[33] =
{22,0x0018C000,0x0018C000,0x0018C000,0x0018C000,0x000C6000,0x000C6000,0x0006300
0,0x00063000,0x00031800,0x00031800,0x00018C00,0x001FFFF8,0x001FFFF8,0x0000C600,
0x00006300,0x00006300,0x00003180,0x001FFFF8,0x001FFFF8,0x000018C0,0x00000C60,0x
00000C60,0x00000630,0x00000630,0x00000318,0x00000318,0x0000018C,0x0000018C,0x0
00000C6,0x000000E7,0x00000063,0x00000063};//#

PROGMEM prog_uint32_t FONT_004[33] =
{22,0x00001F80,0x00003FC0,0x000076E0,0x0000E670,0x0001C638,0x0003861C,0x0007060
E,0x000E0607,0x001C0607,0x00000607,0x00000607,0x0000060E,0x0000061C,0x00000638,
0x00000670,0x00007FE0,0x0000FFC0,0x0001C600,0x00038600,0x00070600,0x000E0600,0x
```

001C0603,0x001C0603,0x001C0603,0x001C0603,0x000E0607,0x0007060E,0x0003861C,0x0
001C638,0x0000E670,0x00007FE0,0x00003FC0};//$

PROGMEM prog_uint32_t FONT_005[33] =
{17,0x0000C000,0x0000C000,0x0000C030,0x0000C078,0x000060FC,0x000060CC,0x000030
CC,0x000030CC,0x000018FC,0x00001878,0x00000C30,0x00000C00,0x00000600,0x0000060
0,0x00000300,0x00000300,0x00000180,0x00000180,0x000030C0,0x000078C0,0x0000FC60,
0x0000CC60,0x0000CC30,0x0000CC30,0x0000FC18,0x00007818,0x0000300C,0x0000000C,0
x00000006,0x00000007,0x00000003,0x00000003};//%

PROGMEM prog_uint32_t FONT_006[33] =
{22,0x00000F80,0x00001FC0,0x000038E0,0x00007070,0x0000E038,0x0001C01C,0x0003800
E,0x00070007,0x00070003,0x00070003,0x00038003,0x0001C003,0x0000E003,0x00007007,
0x0000380E,0x00001C1C,0x00000E38,0x00000770,0x000003E0,0x000003E0,0x000003F0,0x
00000738,0x00000E1C,0x00181C0E,0x001C3807,0x000E7003,0x0007E003,0x0003C007,0x0
003E00E,0x0007701C,0x000E3FFC,0x001C1FF8};//&

PROGMEM prog_uint32_t FONT_007[33] = {4
,0x00000007,0x00000007,0x00000007,0x00000007,0x00000000,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000
000,0x00000000,0x00000000,0x00000000};//'

PROGMEM prog_uint32_t FONT_008[33] = {7
,0x00000030,0x00000018,0x0000000C,0x00000006,0x00000003,0x00000003,0x00000003,0
x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x0
0000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x000
00003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000
006,0x0000000C,0x00000018,0x00000030};//(

PROGMEM prog_uint32_t FONT_009[33] = {7
,0x00000003,0x00000006,0x0000000C,0x00000018,0x00000030,0x00000030,0x00000030,0

x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x0
0000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x000
00030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000
018,0x0000000C,0x00000006,0x00000003};//)

PROGMEM prog_uint32_t FONT_010[33] =
{22,0x00000000,0x00000000,0x00000000,0x00000E00,0x00000E00,0x00000E00,0x00180E0
3,0x001C0E07,0x000E0E0E,0x00070E1C,0x00038E38,0x0001CE70,0x0000EEE0,0x00007FC0,
0x00003F80,0x001FFFFF,0x001FFFFF,0x001FFFFF,0x00003F80,0x00007FC0,0x0000EEE0,0x
0001CE70,0x00038E38,0x00070E1C,0x000E0E0E,0x001C0E07,0x00180E03,0x00000E00,0x00
000E00,0x00000000,0x00000000,0x00000000};//*

PROGMEM prog_uint32_t FONT_011[33] =
{22,0x00000000,0x00000000,0x00000000,0x00000E00,0x00000E00,0x00000E00,0x00000E0
0,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0
x00000E00,0x001FFFFF,0x001FFFFF,0x001FFFFF,0x00000E00,0x00000E00,0x00000E00,0x00
000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x0000
0E00,0x00000000,0x00000000,0x00000000};//+

PROGMEM prog_uint32_t FONT_012[33] = {7
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000
038,0x0000001C,0x0000000E,0x00000007};//,

PROGMEM prog_uint32_t FONT_013[33] =
{22,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,
0x00000000,0x001FFFFF,0x001FFFFF,0x001FFFFF,0x00000000,0x00000000,0x00000000,0x
00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00
000000,0x00000000,0x00000000,0x00000000};//-

PROGMEM prog_uint32_t FONT_014[33] = {4
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000
007,0x00000007,0x00000007,0x00000007};//.

PROGMEM prog_uint32_t FONT_015[33] =
{17,0x0000C000,0x0000C000,0x0000C000,0x0000C000,0x00006000,0x00006000,0x0000300
0,0x00003000,0x00001800,0x00001800,0x00000C00,0x00000C00,0x00000600,0x00000600,
0x00000300,0x00000300,0x00000180,0x00000180,0x000000C0,0x000000C0,0x00000060,0x
00000060,0x00000030,0x00000030,0x00000018,0x00000018,0x0000000C,0x0000000C,0x00
000006,0x00000007,0x00000003,0x00000003};///

PROGMEM prog_uint32_t FONT_016[33] =
{22,0x00003F80,0x00007FC0,0x0000E0E0,0x0001C070,0x00038038,0x0007001C,0x000E000
E,0x001E0007,0x001F0007,0x001F8007,0x001DC007,0x001CE007,0x001C7007,0x001C3807,
0x001C1C07,0x001C0E07,0x001C0707,0x001C0387,0x001C01C7,0x001C00E7,0x001C0077,0
x001C003F,0x001C001F,0x001C000F,0x001C0007,0x000E000E,0x0007001C,0x00038038,0x
0001C070,0x0000E0E0,0x00007FC0,0x00003F80};//0

PROGMEM prog_uint32_t FONT_017[33] =
{17,0x000003E0,0x000003F0,0x000003B8,0x0000039C,0x0000038E,0x00000387,0x0000038
0,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,
0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x
00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00000380,0x00
000380,0x00000380,0x00003FF8,0x00007FFC};//1

PROGMEM prog_uint32_t FONT_018[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C00
0,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,
0x0001FFFF,0x0001FFFF,0x0001FFFF,0x00000007,0x00000007,0x00000007,0x00000007,0x

00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00
000007,0x0001FFFF,0x0001FFFF,0x0001FFFF};//2

PROGMEM prog_uint32_t FONT_019[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C00
0,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,
0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x
0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0
001C000,0x0001FFFF,0x0001FFFF,0x0001FFFF};//3

PROGMEM prog_uint32_t FONT_020[33] =
{22,0x0001E000,0x0001F000,0x0001F800,0x0001DC00,0x0001CE00,0x0001C700,0x0001C38
0,0x0001C1C0,0x0001C0E0,0x0001C070,0x0001C038,0x0001C01C,0x0001C00E,0x0001C007
,0x001FFFFF,0x001FFFFF,0x001FFFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0
x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x
0001C000,0x0001C000,0x0001C000,0x0001C000};//4

PROGMEM prog_uint32_t FONT_021[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x00000007,0x00000007,0x00000007,0x00000007
,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0
x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0
001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x00
01C000,0x0001FFFF,0x0001FFFF,0x0001FFFF};//5

PROGMEM prog_uint32_t FONT_022[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x00000007,0x00000007,0x00000007,0x00000007
,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0
x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0
001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x00
01C007,0x0001FFFF,0x0001FFFF,0x0001FFFF};//6

PROGMEM prog_uint32_t FONT_023[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C00

0,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,
0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0
x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x
0001C000,0x0001C000,0x0001C000,0x0001C000};//7

PROGMEM prog_uint32_t FONT_024[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C007,0x0001C007,0x0001C007,0x0001C00
7,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,
0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x
0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0
001C007,0x0001FFFF,0x0001FFFF,0x0001FFFF};//8

PROGMEM prog_uint32_t FONT_025[33] =
{18,0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C007,0x0001C007,0x0001C007,0x0001C00
7,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,0x0001C007,
0x0001FFFF,0x0001FFFF,0x0001FFFF,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x
0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0001C000,0x0
001C000,0x0001FFFF,0x0001FFFF,0x0001FFFF};//9

PROGMEM prog_uint32_t FONT_026[33] = {4
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000007,0x00000007,0
x00000007,0x00000007,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000007,0x00000007,0x00000007,0x00000007,0x00000000,0x00000
000,0x00000000,0x00000000,0x00000000};//:

PROGMEM prog_uint32_t FONT_027[33] = {7
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000038,0x00000038,0
x00000038,0x00000038,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000038,0x0000001C,0x0000000E,0x00000007,0x00000000,0x00000
000,0x00000000,0x00000000,0x00000000};//;

PROGMEM prog_uint32_t FONT_028[33] =
{19,0x00038000,0x0001C000,0x0000E000,0x00007000,0x00003800,0x00001C00,0x00000E0
0,0x00000700,0x00000380,0x000001C0,0x000000E0,0x00000070,0x00000038,0x0000001C,
0x0000000E,0x00000007,0x00000007,0x0000000E,0x0000001C,0x00000038,0x00000070,0x
000000E0,0x000001C0,0x00000380,0x00000700,0x00000E00,0x00001C00,0x00003800,0x00
007000,0x0000E000,0x0001C000,0x00038000};//<

PROGMEM prog_uint32_t FONT_029[33] =
{22,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x001FFFFF,0
x001FFFFF,0x001FFFFF,0x00000000,0x00000000,0x001FFFFF,0x001FFFFF,0x001FFFFF,0x00
000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000
0000,0x00000000,0x00000000,0x00000000};//=

PROGMEM prog_uint32_t FONT_030[33] =
{19,0x00000007,0x0000000E,0x0000001C,0x00000038,0x00000070,0x000000E0,0x000001C
0,0x00000380,0x00000700,0x00000E00,0x00001C00,0x00003800,0x00007000,0x0000E000,
0x0001C000,0x00038000,0x00038000,0x0001C000,0x0000E000,0x00007000,0x00003800,0x
00001C00,0x00000E00,0x00000700,0x00000380,0x000001C0,0x000000E0,0x00000070,0x00
000038,0x0000001C,0x0000000E,0x00000007};//>

PROGMEM prog_uint32_t FONT_031[33] =
{17,0x0000FFFF,0x0000FFFF,0x0000FFFF,0x0000FFFF,0x0000E007,0x0000E007,0x0000E007
,0x0000E007,0x0000E007,0x0000E007,0x0000E007,0x0000E007,0x0000E007,0x0000E000,0x
0000E000,0x0000FFF8,0x0000FFF8,0x0000FFF8,0x00000038,0x00000038,0x00000038,0x00
000038,0x00000038,0x00000038,0x00000000,0x00000000,0x00000038,0x00000038,0x0000
0038,0x00000038,0x00000038,0x00000000};//?

PROGMEM prog_uint32_t FONT_032[33] =
{22,0x0000FF00,0x0001FF80,0x0003FFC0,0x000700E0,0x000E0070,0x001C0038,0x001C001
C,0x001D000E,0x001DFE07,0x001DFF07,0x001DC787,0x001DC1C7,0x001DC0E7,0x001DC0E
7,0x001DC0E7,0x001DC0E7,0x001DC0E7,0x001DC0E7,0x001CE1C7,0x001C7387,0x000E3F0

```
7,0x00077E07,0x0003FC07,0x0001C00E,0x0000001C,0x00000038,0x00000070,0x000E00E0,
0x000701C0,0x00038380,0x0001FF00,0x0000FF00};//@

PROGMEM prog_uint32_t FONT_033[33] =
{22,0x00000400,0x00000E00,0x00001F00,0x00003B80,0x000071C0,0x0000E0E0,0x0001C07
0,0x00038038,0x0007001C,0x000E000E,0x000E000E,0x000E000E,0x000E000E,0x000E000E,0
x000E000E,0x000FFFFE,0x000FFFFE,0x000FFFFE,0x000E000E,0x000E000E,0x000E000E,0x00
0E000E,0x000E000E,0x000E000E,0x000E000E,0x000E000E,0x000E000E,0x000E000E,0x000E
000E,0x000E000E,0x000E000E,0x001F001F};//A

PROGMEM prog_uint32_t FONT_034[33] =
{22,0x0000FFFF,0x0001FFFE,0x0003801C,0x0007001C,0x000E001C,0x001C001C,0x001C001
C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x000E001C,0x0007001C,0x0003801
C,0x0001FFFC,0x0001FFFC,0x0001FFFC,0x0003801C,0x0007001C,0x000E001C,0x001C001C,
0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x000E001C,
0x0007001C,0x0003801C,0x0003FFFE,0x0001FFFF};//B

PROGMEM prog_uint32_t FONT_035[33] =
{22,0x0017FFF8,0x001FFFFC,0x001FFFFE,0x001C001F,0x0018000F,0x00100007,0x0000000
7,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,
0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x
00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x0010000F,0x00
18001F,0x001FFFFE,0x000FFFFC,0x0007FFF8};//C

PROGMEM prog_uint32_t FONT_036[33] =
{22,0x0000FFFF,0x0001FFFE,0x0003801C,0x0007001C,0x000E001C,0x001C001C,0x001C001
C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001
C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001
C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x000E001
C,0x0007001C,0x0003801C,0x0003FFFE,0x0001FFFF};//D

PROGMEM prog_uint32_t FONT_037[33] =
{22,0x001FFFFF,0x001FFFFE,0x0018001C,0x0010001C,0x0000001C,0x0000001C,0x0000001
```

C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0010001C,0x0018001C
,0x001FFFFC,0x001FFFFC,0x001FFFFC,0x0018001C,0x0010001C,0x0000001C,0x0000001C,0
x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x
0010001C,0x0018001C,0x001FFFFE,0x001FFFFF};//E

PROGMEM prog_uint32_t FONT_038[33] =
{22,0x001FFFFF,0x001FFFFE,0x0018001C,0x0010001C,0x0000001C,0x0000001C,0x0000001
C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0010001C,0x0018001C
,0x001FFFFC,0x001FFFFC,0x001FFFFC,0x0018001C,0x0010001C,0x0000001C,0x0000001C,0
x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x
0000001C,0x0000001C,0x0000003E,0x0000007F};//F

PROGMEM prog_uint32_t FONT_039[33] =
{22,0x0000FFF8,0x0005FFFC,0x0007FFFE,0x0007001F,0x0006000F,0x00040007,0x0000000
7,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,
0x00000007,0x00000007,0x001FC007,0x000F8007,0x00070007,0x00070007,0x00070007,0x
00070007,0x00070007,0x00070007,0x00070007,0x00070007,0x00070007,0x0007000F,0x00
07801F,0x0003FFFE,0x0001FFFC,0x0000FFF8};//G

PROGMEM prog_uint32_t FONT_040[33] =
{22,0x001FC07F,0x000F803E,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001
C,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001C
,0x0007001C,0x0007FFFC,0x0007FFFC,0x0007FFFC,0x0007001C,0x0007001C,0x0007001C,0
x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x0007001C,0x
0007001C,0x0007001C,0x000F803E,0x001FC07F};//H

PROGMEM prog_uint32_t FONT_041[33] = {8
,0x0000007F,0x0000003E,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0
x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x
0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0
000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x00
00001C,0x0000001C,0x0000003E,0x0000007F};//I

```
PROGMEM prog_uint32_t FONT_042[33] =
{22,0x001FC000,0x000F8000,0x00070000,0x00070000,0x00070000,0x00070000,0x0007000
0,0x00070000,0x00070000,0x00070000,0x00070000,0x00070000,0x00070000,0x00070000,
0x00070000,0x00070000,0x00070000,0x00070000,0x00070000,0x00070000,0x00070000,0x
00070004,0x00070006,0x00070007,0x00070007,0x00070007,0x00070007,0x00070007,0x00
038007,0x0001C00E,0x0000FFFC,0x00007FF8};//J

PROGMEM prog_uint32_t FONT_043[33] =
{22,0x001FC07F,0x000F803E,0x0007001C,0x0007001C,0x0003801C,0x0001C01C,0x0000E01
C,0x0000701C,0x0000381C,0x00001C1C,0x00000E1C,0x0000071C,0x0000039C,0x000001DC
,0x000000FC,0x0000007C,0x0000003C,0x0000007C,0x000000FC,0x000001DC,0x0000039C,
0x0000071C,0x00000E1C,0x00001C1C,0x0000381C,0x0000701C,0x0000E01C,0x0001C01C,0
x0003801C,0x0007001C,0x000F803E,0x001FC07F};//K

PROGMEM prog_uint32_t FONT_044[33] =
{22,0x0000007F,0x0000003E,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001
C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C
,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0
x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0010001C,0x0018001C,0x
001C001C,0x000E001C,0x0007FFFE,0x0003FFFF};//L

PROGMEM prog_uint32_t FONT_045[33] =
{22,0x001F001F,0x000F001E,0x0007803C,0x0007C07C,0x0007E0FC,0x0007F1FC,0x0007F1F
C,0x000771DC,0x000771DC,0x000771DC,0x000771DC,0x000771DC,0x00073B9C,0x00073B9
C,0x00073B9C,0x00073B9C,0x00073B9C,0x00071F1C,0x00071F1C,0x00071F1C,0x00071F1C
,0x00071F1C,0x00070E1C,0x00070E1C,0x00070E1C,0x00070E1C,0x00070E1C,0x0007041C,0
x0007041C,0x0007041C,0x000F843E,0x001FC47F};//M

PROGMEM prog_uint32_t FONT_046[33] =
{22,0x001FC01F,0x000F803E,0x0007003C,0x0007003C,0x0007007C,0x0007007C,0x0007007
C,0x000700FC,0x000700DC,0x000700DC,0x000701DC,0x0007019C,0x0007019C,0x0007039
C,0x0007031C,0x0007031C,0x0007071C,0x0007061C,0x0007061C,0x00070E1C,0x00070C1C
```

,0x00070C1C,0x00071C1C,0x0007181C,0x0007181C,0x0007381C,0x0007301C,0x0007301C,
0x0007701C,0x0007F01C,0x0007E03E,0x0007E07F};//N

PROGMEM prog_uint32_t FONT_047[33] =
{22,0x0003FFF8,0x0007FFFC,0x000FFFFE,0x001E001F,0x001C000F,0x001C0007,0x001C000
7,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,
0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0
x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C000F,0x
001E001F,0x000FFFFE,0x0007FFFC,0x0003FFF8};//O

PROGMEM prog_uint32_t FONT_048[33] =
{22,0x0000FFFF,0x0001FFFE,0x0003801C,0x0007001C,0x000E001C,0x001C001C,0x001C001
C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x000E001C,0x0007001C,0x0003801
C,0x0001FFFC,0x0000FFFC,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,
0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0x0000001C,0
x0000001C,0x0000001C,0x0000003E,0x0000007F};//P

PROGMEM prog_uint32_t FONT_049[33] =
{22,0x0003FFF8,0x0007FFFC,0x000FFFFE,0x001E001F,0x001C000F,0x001C0007,0x001C000
7,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,
0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001C0007,0x001E000F,0x000F001F,0
x0007803E,0x0003C07C,0x0001E0F0,0x0000F1E0,0x00007BC0,0x00003F80,0x00001F00,0x0
0003E00,0x00007C00,0x0000F000,0x0003E000};//Q

PROGMEM prog_uint32_t FONT_050[33] =
{22,0x0000FFFF,0x0001FFFE,0x0003801C,0x0007001C,0x000E001C,0x001C001C,0x001C001
C,0x001C001C,0x001C001C,0x001C001C,0x001C001C,0x000E001C,0x0007001C,0x0003801
C,0x0001FFFC,0x0000FFFC,0x00007FFC,0x00001C1C,0x00001C1C,0x0000381C,0x0000381C
,0x0000701C,0x0000701C,0x0000E01C,0x0000E01C,0x0001C01C,0x0001C01C,0x0003801C,
0x0003801C,0x0007001C,0x000F003E,0x001E007F};//R

PROGMEM prog_uint32_t FONT_051[33] =
{22,0x0017FFF8,0x001FFFFC,0x001FFFFE,0x001C001F,0x0018000F,0x00100007,0x0000000

7,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x0000000F,
0x0000001E,0x0003FFFC,0x0007FFF0,0x000FFFE0,0x001E0000,0x001C0000,0x001C0000,0x
001C0000,0x001C0000,0x001C0001,0x001C0001,0x001C0003,0x001C0007,0x001E000F,0x0
01F001D,0x001FFFFC,0x000FFFF8,0x0007FFF8};//S

PROGMEM prog_uint32_t FONT_052[33] =
{22,0x001FFFFF,0x001FFFFF,0x00180E03,0x00100E01,0x00000E00,0x00000E00,0x00000E00
,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x
00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00
000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x0000
0E00,0x00000E00,0x00001F00,0x00003F80};//T

PROGMEM prog_uint32_t FONT_053[33] =
{22,0x001F003F,0x000E001E,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C00
1C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C00
1C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C00
1C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C001C,0x000C00
18,0x000E0018,0x00070070,0x0003FFE0,0x0001FFC0};//U

PROGMEM prog_uint32_t FONT_054[33] =
{22,0x001FC07F,0x000F803F,0x000F001E,0x0007001C,0x0007001C,0x0007001C,0x0007001
C,0x00038038,0x00038038,0x00038038,0x00038038,0x0001C070,0x0001C070,0x0001C070,
0x0001C070,0x0000E0E0,0x0000E0E0,0x0000E0E0,0x0000E0E0,0x000071C0,0x000071C0,0x
000071C0,0x000071C0,0x00003B80,0x00003B80,0x00003B80,0x00003B80,0x00001F00,0x0
0001F00,0x00001F00,0x00000E00,0x00000E00};//V

PROGMEM prog_uint32_t FONT_055[33] =
{22,0x001FC07F,0x000F803E,0x0007001C,0x0007001C,0x0007061C,0x0007061C,0x0007061
C,0x0007061C,0x00038638,0x00038638,0x00039F38,0x00039F38,0x00039F38,0x00039F38,
0x0001DF70,0x0001DB70,0x0001DB70,0x0001DB70,0x0001DB70,0x0001DB70,0x0000F1E0,0
x0000F1E0,0x0000F1E0,0x0000F1E0,0x0000F1E0,0x0000F1E0,0x0000F1E0,0x000071C0,0x0
00071C0,0x000071C0,0x000071C0,0x000071C0};//W

```
PROGMEM prog_uint32_t FONT_056[33] =
{22,0x0007C01F,0x0003800E,0x0001800E,0x0000C01C,0x0000401C,0x00006038,0x0000203
8,0x00003070,0x00001070,0x000018E0,0x000008E0,0x00000DC0,0x00000FC0,0x00000F80,
0x00000780,0x00000700,0x00000700,0x00000F80,0x00000F80,0x00001CC0,0x00001C40,0x
00003860,0x00003820,0x00007030,0x00007010,0x0000E018,0x0000E008,0x0001C00C,0x00
03C00C,0x0007C00E,0x000FE01E,0x001FF03F};//X

PROGMEM prog_uint32_t FONT_057[33] =
{22,0x001F001F,0x000E000E,0x000E000E,0x0007001C,0x0007001C,0x0007001C,0x0003803
8,0x00038038,0x00038038,0x0001C070,0x0001C070,0x0001C070,0x0000E0E0,0x0000E0E0,
0x0000E0E0,0x00007FC0,0x00007FC0,0x00001F00,0x00000E00,0x00000E00,0x00000E00,0x
00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00000E00,0x00
000E00,0x00000E00,0x00001F00,0x00003F80};//Y

PROGMEM prog_uint32_t FONT_058[33] =
{22,0x001FFFFF,0x001FFFFF,0x001C0003,0x001C0001,0x001C0000,0x001C0000,0x001C000
0,0x001C0000,0x001C0000,0x001E0000,0x000F0000,0x00078000,0x0003C000,0x0001E000,
0x0000F000,0x00007800,0x00003C00,0x00001E00,0x00000F00,0x00000780,0x000003C0,0x
000001E0,0x000000F0,0x00000078,0x0000003C,0x0000001E,0x0000000F,0x0000000F,0x00
100007,0x00180007,0x001FFFFF,0x001FFFFF};//Z

PROGMEM prog_uint32_t FONT_059[33] = {9
,0x000000FF,0x0000003F,0x0000001F,0x0000000F,0x00000007,0x00000007,0x00000007,0x
00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00
000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x0000
0007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x000000
0F,0x0000001F,0x0000003F,0x000000FF};//[

PROGMEM prog_uint32_t FONT_060[33] = {9
,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000006,0x00000006,0
x00000006,0x00000006,0x00000006,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x
0000000C,0x00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x00000030,0x00
```

000030,0x00000030,0x00000030,0x00000030,0x00000060,0x00000060,0x00000060,0x0000
0060,0x00000060,0x000000C0,0x000000C0};//\

PROGMEM prog_uint32_t FONT_061[33] = {9
,0x000000FF,0x000000FC,0x000000F8,0x000000F0,0x000000E0,0x000000E0,0x000000E0,0x
000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x00
0000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x0000
00E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000
F0,0x000000F8,0x000000FC,0x000000FF};//]

PROGMEM prog_uint32_t FONT_062[33] =
{22,0x000000FF,0x000000FC,0x000000F8,0x000000F0,0x000000E0,0x000000E0,0x000000E
0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0
x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x0
00000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000000E0,0x000
000F0,0x000000F8,0x000000FC,0x000000FF};//^

PROGMEM prog_uint32_t FONT_063[33] =
{30,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,
0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x
00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00
000000,0x1FFFFFFF,0x1FFFFFFF,0x00000000};//_

PROGMEM prog_uint32_t FONT_064[33] = {8
,0x00000007,0x0000000E,0x0000001C,0x00000038,0x00000070,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0
0000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x000
00000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000
000,0x00000000,0x00000000,0x00000000};//'

PROGMEM prog_uint32_t FONT_065[33] =
{16,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000

0,0x00000000,0x00000000,0x00000000,0x000003E0,0x000007F0,0x00000E38,0x00001C1C,
0x0000380E,0x00007007,0x00007003,0x00007000,0x00007000,0x00007000,0x00007000,0x
00007F80,0x00007FC0,0x000070E0,0x00007070,0x00007038,0x0000781C,0x00007C0E,0x00
007E07,0x00007703,0x000073FF,0x000071FE};//a

PROGMEM prog_uint32_t FONT_066[33] =
{16,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x00000007,0x0000000
7,0x00000007,0x00000007,0x00000007,0x00000787,0x00000FC7,0x00001CE7,0x00003877,
0x0000703F,0x0000701F,0x0000700F,0x00007007,0x00007007,0x00007007,0x00007007,0x
00007007,0x00007007,0x00007007,0x00007007,0x00007007,0x00007007,0x00007807,0x00
003C07,0x00001E07,0x00000FFF,0x000007FF};//b

PROGMEM prog_uint32_t FONT_067[33] =
{15,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000003E0,0x000007F0,0x00000E38,0x00001C1C,
0x0000380E,0x00003007,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x
00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00003003,0x00003803,0x00
001C07,0x00000E0E,0x000007FC,0x000003F8};//c

PROGMEM prog_uint32_t FONT_068[33] =
{16,0x00007000,0x00007000,0x00007000,0x00007000,0x00007000,0x00007000,0x0000700
0,0x00007000,0x00007000,0x00007000,0x000070F0,0x000071F8,0x0000739C,0x0000770E,
0x00007E07,0x00007C07,0x00007807,0x00007007,0x00007007,0x00007007,0x00007007,0x
00007007,0x00007007,0x00007007,0x00007007,0x00007007,0x00007007,0x0000700F,0x00
00701E,0x0000703C,0x00007FF8,0x00007FF0};//d

PROGMEM prog_uint32_t FONT_069[33] =
{17,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000007E0,0x00000FF0,0x00001C3C,0x0000380E,
0x00007007,0x0000E003,0x0000C003,0x0000FFFF,0x0000FFFF,0x00000003,0x00000003,0x
00000003,0x00000003,0x00000003,0x00000003,0x0000C003,0x0000E007,0x0000700E,0x00
00381C,0x00001C38,0x00000FF0,0x000007E0};//e

```
PROGMEM prog_uint32_t FONT_070[33] =
{17,0x00001F80,0x00003FC0,0x000070E0,0x0000E070,0x0000C038,0x00000018,0x0000001
8,0x00000018,0x00000018,0x00000018,0x00000018,0x000000FF,0x000000FF,0x00000018,
0x00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x
00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x00000018,0x00
000018,0x00000018,0x00000018,0x00000018};//f

PROGMEM prog_uint32_t FONT_071[33] =
{15,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000001F0,0x000003F8,0x0000071C,0x00000E0E,
0x00001C07,0x00003803,0x00003003,0x00003003,0x00003003,0x00003003,0x00003803,0x
00003C07,0x00003E0E,0x000037FC,0x000033F8,0x00003000,0x00003003,0x00003807,0x00
001C0E,0x00000E1C,0x000007F8,0x000001F0};//g

PROGMEM prog_uint32_t FONT_072[33] =
{12,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x0000000
3,0x00000003,0x00000003,0x00000003,0x000000F3,0x000001FB,0x0000039F,0x0000070F,
0x00000607,0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x
00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00
000603,0x00000603,0x00000603,0x00000603};//h

PROGMEM prog_uint32_t FONT_073[33] = {3
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x00000003,0x00000003,0x00000000,0x00000000,0x0
0000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x000
00003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000
003,0x00000003,0x00000003,0x00000003};//i

PROGMEM prog_uint32_t FONT_074[33] = {9
,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0
x00000000,0x00000000,0x00000000,0x000000C0,0x000000C0,0x00000000,0x00000000,0x0
00000C0,0x000000C0,0x000000C0,0x000000C0,0x000000C0,0x000000C0,0x000000C0,0x00
```

0000C0,0x000000C0,0x000000C0,0x000000C3,0x000000C3,0x000000C3,0x000000C3,0x000
000E3,0x00000077,0x0000003E,0x0000001C};//j

PROGMEM prog_uint32_t FONT_075[33] =
{11,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x0000000
3,0x00000003,0x00000003,0x00000003,0x00000303,0x00000383,0x000001C3,0x000000E3,
0x00000073,0x0000003B,0x0000001F,0x0000000F,0x00000007,0x00000003,0x00000003,0x
00000003,0x00000003,0x00000007,0x0000000F,0x0000001F,0x0000003B,0x00000073,0x00
0000E3,0x000001C3,0x00000383,0x00000303};//k

PROGMEM prog_uint32_t FONT_076[33] = {3
,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0
x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x0
0000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x000
00003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000
003,0x00000003,0x00000003,0x00000003};//l

PROGMEM prog_uint32_t FONT_077[33] =
{23,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x0003C0E3,0x0007E1F3,0x000E73BB,0x001C3F1F,
0x00381E0F,0x00300C07,0x00300C03,0x00300C03,0x00300C03,0x00300C03,0x00300C03,0
x00300C03,0x00300C03,0x00300C03,0x00300C03,0x00300C03,0x00300C03,0x00300C03,0x
00300C03,0x00300C03,0x00300C03,0x00300C03};//m

PROGMEM prog_uint32_t FONT_078[33] =
{13,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000000E3,0x000001F3,0x000003BB,0x0000071F,
0x00000E0F,0x00000C07,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0
x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x
00000C03,0x00000C03,0x00000C03,0x00000C03};//n

PROGMEM prog_uint32_t FONT_079[33] =
{18,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000

0,0x00000000,0x00000000,0x00000000,0x00000FE0,0x00001FF0,0x00003838,0x0000701C,
0x0000E00E,0x0001C007,0x00018003,0x00018003,0x00018003,0x00018003,0x00018003,0x
00018003,0x00018003,0x00018003,0x00018003,0x00018003,0x00018003,0x0001C007,0x00
00E00E,0x0000701C,0x00003FF8,0x00001FF0};//o

PROGMEM prog_uint32_t FONT_080[33] =
{12,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000000FF,0x000001FF,0x00000383,0x00000703,
0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00000703,0x
00000383,0x000001FF,0x000000FF,0x00000003,0x00000003,0x00000003,0x00000003,0x00
000003,0x00000003,0x00000003,0x00000003};//p

PROGMEM prog_uint32_t FONT_081[33] =
{12,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000007F8,0x000007FC,0x0000060E,0x00000607,
0x00000603,0x00000603,0x00000603,0x00000603,0x00000603,0x00000607,0x0000060E,0x
0000061C,0x000007F8,0x000007F0,0x00000600,0x00000600,0x00000600,0x00000600,0x00
000600,0x00000600,0x00000600,0x00000600};//q

PROGMEM prog_uint32_t FONT_082[33] =
{13,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x000003C3,0x000007E3,0x00000E73,0x00000C3B,
0x0000001F,0x0000000F,0x00000007,0x00000003,0x00000003,0x00000003,0x00000003,0x
00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00000003,0x00
000003,0x00000003,0x00000003,0x00000003};//r

PROGMEM prog_uint32_t FONT_083[33] =
{17,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000FF0,0x00001FF8,0x0000381C,0x0000700E,
0x0000E007,0x0000C003,0x00000003,0x00000003,0x00000007,0x0000000E,0x0000001C,0x
00000FF8,0x00001FF0,0x00003800,0x00007000,0x0000E000,0x0000C003,0x0000C007,0x00
00E00E,0x0000701C,0x00003FF8,0x00001FF0};//s

```
PROGMEM prog_uint32_t FONT_084[33] =
{17,0x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x0000018
0,0x00000180,0x00000180,0x00000180,0x0000FFFF,0x0000FFFF,0x00000180,0x00000180,0
x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x0
0000180,0x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x00000180,0x000
00180,0x00000180,0x00000180,0x00000180};//t

PROGMEM prog_uint32_t FONT_085[33] =
{13,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000C03,0x00000C03,0x00000C03,0x00000C03,
0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0
x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x00000C03,0x
00000C03,0x00000C03,0x00000FFF,0x00000FFF};//u

PROGMEM prog_uint32_t FONT_086[33] =
{17,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x0000C003,0x0000C003,0x0000C003,0x00006006,
0x00006006,0x00006006,0x0000300C,0x0000300C,0x0000300C,0x00001818,0x00001818,0x
00001818,0x00000C30,0x00000C30,0x00000C30,0x00000C30,0x00000660,0x00000660,0x0
0000660,0x00000660,0x000003C0,0x00000180};//v

PROGMEM prog_uint32_t FONT_087[33] =
{25,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00C01803,0x00C01803,0x00C01803,0x00C01803,
0x00C01803,0x00C01803,0x00C01803,0x00C01803,0x00C01803,0x00C01803,0x00C01803,0
x00C01803,0x00C01803,0x00C01803,0x00C01803,0x00C01803,0x00C03C03,0x00E07E07,0x
0070E70E,0x0039C39C,0x001F81F8,0x000F00F0};//w

PROGMEM prog_uint32_t FONT_088[33] =
{13,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000C03,0x00000C03,0x00000606,0x00000606,
0x0000030C,0x0000030C,0x00000198,0x00000198,0x000000F0,0x000000F0,0x00000060,0x
```

00000060,0x000000F0,0x000000F0,0x00000198,0x00000198,0x0000030C,0x0000030C,0x00000606,0x00000606,0x00000C03,0x00000C03};//x

PROGMEM prog_uint32_t FONT_089[33] = {13,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000C03,0x00000C03,0x00000606,0x00000606,0x0000030C,0x0000030C,0x00000198,0x00000198,0x000000F0,0x000000F0,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060,0x00000060};//y

PROGMEM prog_uint32_t FONT_090[33] = {17,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x0000FFFF,0x0000FFFF,0x0000C000,0x0000C000,0x0000C000,0x0000E000,0x00007000,0x00003800,0x00001C00,0x00000E00,0x00000700,0x00000380,0x000001C0,0x000000E0,0x00000070,0x00000038,0x0000001C,0x0000000E,0x00000007,0x00000003,0x0000FFFF,0x0000FFFF};//z

PROGMEM prog_uint32_t FONT_091[33] = {9 ,0x000000E0,0x00000030,0x00000018,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x00000006,0x00000003,0x00000006,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x0000000C,0x00000018,0x00000030,0x000000E0};//{

PROGMEM prog_uint32_t FONT_092[33] = {3 ,0x00000003};//|

PROGMEM prog_uint32_t FONT_093[33] = {9 ,0x00000007,0x0000000C,0x00000018,0x00000030,0x00000030,0x00000030,0x00000030,0

x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x0
0000060,0x000000C0,0x00000060,0x00000030,0x00000030,0x00000030,0x00000030,0x000
00030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000030,0x00000
030,0x00000018,0x0000000C,0x00000007};//}

PROGMEM prog_uint32_t FONT_094[33] =
{22,0x001C03F0,0x001C03F0,0x00070E1C,0x00070E1C,0x0001F807,0x0001F807,0x0000000
0,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,
0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x
00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00
000000,0x00000000,0x00000000,0x00000000};//~

//

PROGMEM uint32_t *FONTS_Table[] =

{

FONT_000,FONT_001,FONT_002,FONT_003,FONT_004,FONT_005,FONT_006,FONT_007,FO
NT_008,FONT_009,FONT_010,FONT_011,FONT_012,FONT_013,FONT_014,FONT_015,

FONT_016,FONT_017,FONT_018,FONT_019,FONT_020,FONT_021,FONT_022,FONT_023,FO
NT_024,FONT_025,FONT_026,FONT_027,FONT_028,FONT_029,FONT_030,FONT_031,

FONT_032,FONT_033,FONT_034,FONT_035,FONT_036,FONT_037,FONT_038,FONT_039,FO
NT_040,FONT_041,FONT_042,FONT_043,FONT_044,FONT_045,FONT_046,FONT_047,

FONT_048,FONT_049,FONT_050,FONT_051,FONT_052,FONT_053,FONT_054,FONT_055,FO
NT_056,FONT_057,FONT_058,FONT_059,FONT_060,FONT_061,FONT_062,FONT_063,

```
FONT_064,FONT_065,FONT_066,FONT_067,FONT_068,FONT_069,FONT_070,FONT_071,FO
NT_072,FONT_073,FONT_074,FONT_075,FONT_076,FONT_077,FONT_078,FONT_079,

FONT_080,FONT_081,FONT_082,FONT_083,FONT_084,FONT_085,FONT_086,FONT_087,FO
NT_088,FONT_089,FONT_090,FONT_091,FONT_092,FONT_093,FONT_094,
};
//
uint32_t BUFDIGIT_AC[MAXDIGIT+1];
uint32_t BUFDIGIT_BD[MAXDIGIT+1];

uint32_t BUFDISPLAY[32][400];

String textFont;

unsigned int SIZE_MESSAGES;
//
////////////////////////////////////////////////////////////////////////////////
void delayClock()
{
  __asm__("nop\n\t");__asm__("nop\n\t");__asm__("nop\n\t");
  __asm__("nop\n\t");__asm__("nop\n\t");__asm__("nop\n\t");
```

```
}
////////////////////////////////////////////////////////////////////////////////
void storageRegisterClock() //StorageRegisterClock();
{
  PIOD->PIO_CODR = PIO_PD6; // set bit LOW    STR
  delayClock();
  PIOD->PIO_SODR = PIO_PD6; // set bit HIGH   STR
}
////////////////////////////////////////////////////////////////////////////////
void shiftRegisterClockInput()
{
  PIOD->PIO_CODR = PIO_PD9; // set bit LOW    CLK
  delayClock();
  PIOD->PIO_SODR = PIO_PD9; // set bit HIGH   CLK
}
////////////////////////////////////////////////////////////////////////////////
void set_bit_inac(boolean s)
{
  if(s) PIOD->PIO_SODR = PIO_PD2;
  else  PIOD->PIO_CODR = PIO_PD2;
}
```

```
////////////////////////////////////////////////////////////////////////////////////////

void set_bit_inbd(boolean s)

{

  if(s) PIOD->PIO_SODR = PIO_PD3;

  else  PIOD->PIO_CODR = PIO_PD3;

}

////////////////////////////////////////////////////////////////////////////////////////

void putByteShiftRegister() // ฟังชันก์ ย่อย ส่งข้อมูล ด้าน คอลัมน

{

  uint32_t iac,ibd;

  boolean biac,bibd;

  for(int g = 0;g < MAXDIGIT;g++)

  {

    iac = BUFDIGIT_AC[g];

    ibd = BUFDIGIT_BD[g];

    for(int b = 0;b < 8;b++)

    {

      biac = (iac << b) & 0x80;

      bibd = (ibd << b) & 0x80;


      set_bit_inac(biac);
```

```
  set_bit_inbd(bibd);


  shiftRegisterClockInput();
}
for(int b = 0;b < 8;b++)
{
  biac = (iac << b) & 0x8000;
  bibd = (ibd << b) & 0x8000;


  set_bit_inac(biac);
  set_bit_inbd(bibd);;


  shiftRegisterClockInput();
}
for(int b = 0;b < 8;b++)
{
  biac = (iac << b) & 0x800000;
  bibd = (ibd << b) & 0x800000;


  set_bit_inac(biac);
  set_bit_inbd(bibd);
```

```
      shiftRegisterClockInput();
    }
    for(int b = 0;b < 8;b++)
    {
      biac = (iac << b) & 0x80000000;
      bibd = (ibd << b) & 0x80000000;

      set_bit_inac(biac);
      set_bit_inbd(bibd);

      shiftRegisterClockInput();
    }
  }
  storageRegisterClock();
}
////////////////////////////////////////////////////////////////////////////////
void offRow() // ฟังชันก์ ปิด การแสกน ROW
{
  digitalWrite(D1,LOW);
  digitalWrite(D2,LOW);
```

```
}
////////////////////////////////////////////////////////////////////////////////
void scanRow(byte r) // ฟังชันก์ แสกน ROW 1 - 16
{
  if(r < 8)
  {
    r = r - 0;
    if(r & 0x01) digitalWrite(S0,HIGH); // set bit (hi)   // S0
    else         digitalWrite(S0, LOW); // set bit (low) // S0
    if(r & 0x02) digitalWrite(S1,HIGH); // set bit (hi)   // S1
    else         digitalWrite(S1, LOW); // set bit (low) // S1
    if(r & 0x04) digitalWrite(S2,HIGH); // set bit (hi)   // S2
    else         digitalWrite(S2, LOW); // set bit (low) // S2
    digitalWrite(D1,HIGH);  // set bit (hi)              // D1
  }
  else if(r < 16)
  {
    r = r - 8;
    if(r & 0x01) digitalWrite(S0,HIGH); // set bit (hi)    // S0
    else         digitalWrite(S0, LOW); // set bit (low)  // S0
    if(r & 0x02) digitalWrite(S1,HIGH); // set bit (hi)    // S1
```

```
    else        digitalWrite(S1, LOW); // set bit (low)  // S1
    if(r & 0x04) digitalWrite(S2,HIGH); // set bit (hi)    // S2
    else        digitalWrite(S2, LOW); // set bit (low)  // S2
    digitalWrite(D2,HIGH); // set bit (hi)                // D2
  }
  else return;
  delayMicroseconds(990);
}
////////////////////////////////////////////////////////////////////////////////
void clearBuf()
{
  for(int r = 0;r < 32;r++)
  {
    for(int c = 0;c < 400;c++)
   {
      BUFDISPLAY[r][c] = 0;
   }
  }
}
////////////////////////////////////////////////////////////////////////////////
void loadFont(String f)
```

```
{
  uint32_t data;
  unsigned int s;
  unsigned int count_bit_s;
  unsigned int count_bit_t;
  unsigned int row_data_buf;

  unsigned int len = f.length();

  clearBuf();

  for(unsigned int r = 0;r < 32;r++)
  {
    count_bit_s = 0;
    count_bit_t = 0;
    row_data_buf = 5;

    SIZE_MESSAGES = 0;

    for(unsigned int i = 0;i < len;i++)
    {
```

```
byte index = (byte) f[i];


index = index - 32;


count_bit_s  = count_bit_t;
          s = pgm_read_dword_near(FONTS_Table[index] + 0);
data        = pgm_read_dword_near(FONTS_Table[index] + r + 1);


count_bit_t += s;
SIZE_MESSAGES += s;


if(count_bit_t >= 32)
{
  BUFDISPLAY[r][row_data_buf] = BUFDISPLAY[r][row_data_buf] | (data <<
count_bit_s);
  row_data_buf++;


  count_bit_s = 32 - count_bit_s;


  count_bit_t = count_bit_t - 32;
```

```
      BUFDISPLAY[r][row_data_buf] = BUFDISPLAY[r][row_data_buf] | (data >>
count_bit_s);
    }
    else
    {
      BUFDISPLAY[r][row_data_buf] = BUFDISPLAY[r][row_data_buf]  | (data <<
count_bit_s);
    }

  }
 }
 SIZE_MESSAGES = (SIZE_MESSAGES / 32) + 6;
}
////////////////////////////////////////////////////////////////////////////////
void runDisplay()
{
 for(int f = 0;f < SIZE_MESSAGES ;f++)
 {
  for(int m = 0;m < 32;m++)
  {
   for(int s = 0;s < 5;s++)
   {
```

```
for(int r = 0;r < 16;r++)
{
  if(r < 8)
  {
    for(int g = 0;g < 5;g++)
    {
      uint32_t d1 = BUFDISPLAY[r][f + 0 + g];
      uint32_t d2 = BUFDISPLAY[r][f + 1 + g];

      d1 = d1 >> m;
      d2 = d2 << (32 - m);
      d1 = d1 | d2;

      BUFDIGIT_AC[g] =  d1;

      uint32_t d3 = BUFDISPLAY[r + 8][f + 0 + g];
      uint32_t d4 = BUFDISPLAY[r + 8][f + 1 + g];

      d3 = d3 >> m;
      d4 = d4 << (32 - m);
```

```
    d3 = d3 | d4;


    BUFDIGIT_BD[g] =  d3;
  }
}
else
{
  for(int g = 0;g < 5;g++)
  {
    uint32_t d1 = BUFDISPLAY[r + 8][f + 0 + g];
    uint32_t d2 = BUFDISPLAY[r + 8][f + 1 + g];


    d1 = d1 >> m;
    d2 = d2 << (32 - m);
    d1 = d1 | d2;


    BUFDIGIT_AC[g] =  d1;



    uint32_t d3 = BUFDISPLAY[r + 16][f + 0 + g];
    uint32_t d4 = BUFDISPLAY[r + 16][f + 1 + g];
```

```
            d3 = d3 >> m;
            d4 = d4 << (32 - m);
            d3 = d3 | d4;

            BUFDIGIT_BD[g] =  d3;
          }
        }
        offRow();
        putByteShiftRegister();
        scanRow(r);
        if(getDataText() == true) return;
      }
    }
  }
}
////////////////////////////////////////////////////////////////////////////////////////
void textCol()
{
  uint32_t c = 0x00000001;
```

```
for(int d = 0;d < 32;d++)
{
  c = c << 1;
  for(int s = 0;s < 40;s++)
  {
     for(int r = 0;r < 16;r++)
     {
        BUFDIGIT_AC[0] = c;
        BUFDIGIT_AC[1] = c;
        BUFDIGIT_AC[2] = c;
        BUFDIGIT_AC[3] = c;
        BUFDIGIT_AC[4] = c;

        BUFDIGIT_BD[0] = c;
        BUFDIGIT_BD[1] = c;
        BUFDIGIT_BD[2] = c;
        BUFDIGIT_BD[3] = c;
        BUFDIGIT_BD[4] = c;

      offRow();
      putByteShiftRegister();
```

```
        storageRegisterClock();
        scanRow(r);
        delayMicroseconds(1000);
      }
    }
  }
}
////////////////////////////////////////////////////////////////////////////////////////
void testRow()
{
  for(int d = 0;d < 16;d++)
  {
    for(int s = 0;s < 40;s++)
    {
      for(int r = 0;r < 16;r++)
      {

        if(r == d)
        {
          BUFDIGIT_AC[0] =  0xFFFFFFFF;
          BUFDIGIT_AC[1] =  0xFFFFFFFF;
```

```
  BUFDIGIT_AC[2] = 0xFFFFFFFF;
  BUFDIGIT_AC[3] = 0xFFFFFFFF;
  BUFDIGIT_AC[4] = 0xFFFFFFFF;

  BUFDIGIT_BD[0] = 0xFFFFFFFF;
  BUFDIGIT_BD[1] = 0xFFFFFFFF;
  BUFDIGIT_BD[2] = 0xFFFFFFFF;
  BUFDIGIT_BD[3] = 0xFFFFFFFF;
  BUFDIGIT_BD[4] = 0xFFFFFFFF;
}
else
{
  BUFDIGIT_AC[0] = 0;
  BUFDIGIT_AC[1] = 0;
  BUFDIGIT_AC[2] = 0;
  BUFDIGIT_AC[3] = 0;
  BUFDIGIT_AC[4] = 0;

  BUFDIGIT_BD[0] = 0;
  BUFDIGIT_BD[1] = 0;
  BUFDIGIT_BD[2] = 0;
```

```
          BUFDIGIT_BD[3] =  0;
          BUFDIGIT_BD[4] =  0;
        }

        offRow();
        putByteShiftRegister();
        storageRegisterClock();
        scanRow(r);
        delayMicroseconds(1000);
      }
    }
  }
}
////////////////////////////////////////////////////////////////////////////////
boolean getDataText()
{
  if (Serial1.available())
  {
    char d = Serial1.read();

    if(d == '\r')
```

```
    {
      Serial1.println("OK");
      loadFont(textFont);
      textFont = "";
      return true;
    }
    else if(d >= 0x20 && d <= 0x7f)
    {
      textFont += d;
      offRow();
    }
  }
  return false;
}
////////////////////////////////////////////////////////////////////////////////
void setup()
{
  Serial1.begin(115200);

  pinMode(D1 , OUTPUT);
  pinMode(D2 , OUTPUT);
```

```
pinMode(S0 , OUTPUT);

pinMode(S1 , OUTPUT);

pinMode(S2 , OUTPUT);


pinMode(INAC, OUTPUT);

pinMode(INBD, OUTPUT);

pinMode(STR , OUTPUT);

pinMode(CLK , OUTPUT);


digitalWrite(D1,LOW);

digitalWrite(D2,LOW);

digitalWrite(S0,HIGH);

digitalWrite(S1,HIGH);

digitalWrite(S2,HIGH);

digitalWrite(INAC,HIGH);

digitalWrite(INBD,HIGH);

digitalWrite(CLK ,HIGH);

digitalWrite(STR ,HIGH);

textFont = "";

SIZE_MESSAGES = 0;

clearBuf();
```

```
  offRow();
}
////////////////////////////////////////////////////////////////////////////////
void loop()
{
  getDataText();
  runDisplay();
}
```