

ระบบควบคุมแอลอีดีดอทแมทริกดิสเพลแบบไร้สาย
Wireless System Control LED Dot Matrix Display

ณัฐชนัน กมลสุวรรณ
Nutchanun Kmosuwan

ณัฐพงศ์ ธนะสินธนา
Nutthaphong Thanasinthana

บรรณวิษณุ นิลกนิษฐ
Bannawit Nilakanis

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2557

ระบบควบคุมแอลอีดีดอทแมทริกดิสเพลแบบไร้สาย

Wireless System Control LED Dot Matrix Display

โดย

ณัฐชนัน กมลสุวรรณ

ณัฐพงศ์ ธนะสินธนา

บรรณวิษณุ นิลกนิษฐ

อาจารย์ที่ปรึกษา

ผศ.ดร.แสงระวี บัวแก้ว

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2557

หัวข้อปริญญานิพนธ์	ระบบควบคุมแอลอีดีติดต่อทแมทริกดิสเพลแบบไร้สาย	
นักศึกษา	นาย ณัฐชนัน กมลสุวรรณ	รหัสประจำตัว 54010397
	นาย ณัฐพงศ์ ณะสินธนา	รหัสประจำตัว 54010416
	นาย บรรณวิษุ์ นิลกนิษฐ	รหัสประจำตัว 54010727
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2557	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	ผศ.ดร.แสงระวี บัวแก้ว	

บทคัดย่อ

โครงการนี้นำเสนอการควบคุมแอลอีดีติดต่อทแมทริกดิสเพลผ่านสัญญาณ Bluetooth 2.4 GHz เพื่อความสะดวกในการส่งข้อมูลโดยไม่ต้องใช้สายเคเบิลซึ่งในโครงการนี้จะทำการส่งข้อมูลผ่านแอปพลิเคชันบน SMARTPHONE ระบบปฏิบัติการแอนดรอยด์ หลักการประกอบไปด้วย 2 ส่วนหลัก คือ ส่วนส่งการจากโทรศัพท์ และ ส่วนประมวลผลสัญญาณ Bluetooth เมื่อผู้ใช้งานเลือกอุปกรณ์ที่ต้องการควบคุมจากแอปพลิเคชันบนโทรศัพท์แล้ว คำสั่งจะถูกส่งผ่านสัญญาณ Bluetooth ไปยังส่วนประมวลผลแปลงเป็นคำสั่งให้ LED Dot Matrix ทำงาน

Thesis Title	Wireless system control LED Dot Matrix Display	
Student	Mr. Nutchanut Kmosuwan	Student ID 54010397
	Mr. Nutthaphong Thanasinthana	Student ID 54010416
	Mr. Bannawit Nilakanis	Student ID 54010727
Degree	Bachelor of Engineering	
Program	Electronics Engineering	
Year	2014	
Thesis Advisor	Asst.Prof.Dr . Seangrawee Buakeaw	

Abstract

This project presents system control LED Dot Matrix Display data transmission by Bluetooth 2.4 GHz, for easiness to transmission data without cable-line. Our system will send data on application in smartphone (system android). The main system consists of two parts. First part is command system; Second part is processing system. When users picked electronics on smartphone, command is sent to processing system. Processing system will analyze command and Controlling LED Dot Matrix.

กิตติกรรมประกาศ

การทำโครงการระบบควบคุมการส่งข้อมูลไร้สายด้วยความถี่ 2.4 GHz นี้ได้รับการสนับสนุนและช่วยเหลือให้คำแนะนำจากอาจารย์ภาควิชาอิเล็กทรอนิกส์ และได้รับความอนุเคราะห์ในด้านอุปกรณ์และเครื่องมือจากภาควิชาอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่ง ดร.แสงระวี บัวแก้ว ที่ให้แนวคิดและคำปรึกษา การแก้ไขปัญหาและการสนับสนุนในการทำวิจัย รวมไปถึงบิดาแลมารดาของผู้ทำโครงการ ที่ให้การสนับสนุนในด้านทุนทรัพย์เพื่อทำการจัดหาอุปกรณ์บางส่วน ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาของท่านอย่างที่สุดและขอกราบขอบพระคุณเป็นอย่างสูง

ณัฐชนัน กมลสุวรรณ
ณัฐพงศ์ ธนะสินธนา
บรรณวิชญ์ นิลกนิษฐ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตในโครงการ.....	1
1.4 ประโยชน์ที่จะได้รับ.....	1
1.5 รายละเอียดโดยย่อ.....	2
บทที่ 2 ทฤษฎีและหลักการ.....	3
2.1 Bluetooth.....	3
2.2 HC-05.....	5
2.3 แอนดรอยด์ (Android)	7
2.4 Arduino.....	10
2.5 LED Dot Matrix.....	13
2.6 MAX7219.....	16
บทที่ 3 วงจรและโปรแกรมที่ใช้.....	19
3.1 การออกแบบ.....	19
3.1.1 การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่	19
3.1.2 การออกแบบวงจร Wireless System Control LED Dot Matrix Display.....	20
3.1.2.1 วงจร Microcontroller.....	20
3.1.2.2 วงจร LED Dot Matrix.....	21
3.1.2.3 วงจร Bluetooth Module.....	22
3.2 โปรแกรมไมโครคอนโทรลเลอร์ใช้.....	22
3.3 หลักการทำงานของแอปพลิเคชัน.....	27
บทที่ 4 การทดลองและผลการทดลอง.....	28
4.1 ขั้นตอนการใช้งานแอปพลิเคชัน.....	28
4.2 การส่งงานงานแอปพลิเคชันไปยัง LED Dot Matrix.....	31

สารบัญ(ต่อ)

	หน้า
4.3 สัญญาณพัลส์ของ วงจร LED Dot Matrix.....	33
4.4 ตารางระยะทางที่บลูทูธสามารถทำงานได้.....	34
บทที่ 5 สรุปผลการทดลอง.....	35
ภาคผนวก.....	36
หนังสืออ้างอิง.....	37

สารบัญตาราง

ตารางที่	หน้า
2.1 เปรียบเทียบเทคโนโลยี Bluetooth กับ IEEE.....	5
2.2 แสดงตำแหน่งของบิตในรีจิสเตอร์สำหรับคำสั่งและข้อมูลขนาด 16 บิต.....	17
2.3 แสดงเลขที่อยู่ของรีจิสเตอร์ภายในของไอซี MAX7219.....	17
4.4 ตารางระยะทางที่บลูทูธสามารถทำงานได้.....	34

สารบัญรูป

รูปที่	หน้า
2.1 บล็อกไดอะแกรม.....	3
2.2 โมดูล HC 05.....	6
2.3 สถาปัตยกรรมของ HC 05.....	6
2.4 สถาปัตยกรรมของแอนดรอยด์.....	7
2.5 ตัวอย่างโปรแกรม 2 MIT App inventor.....	9
2.6 การเชื่อมต่อโมดูลกับ บอร์ด Arduino.....	10
2.7 PIN ต่างๆของบอร์ด Arduino.....	11
2.8 เลือกหุ่นบอร์ด Arduino ที่ต้องการ Upload.....	12
2.9 เลือกหมายเลข Comport ของบอร์ด.....	12
2.10 การ Compiler และ Up Load Code.....	13
2.11 วงจร LED Dot Matrix 8x8.....	14
2.12 การต่อ MAX7219 กับ LED Dot Matrix 8x8.....	15
2.13 แสดงตำแหน่งขาของไอซี MAX7219 ตัวถังแบบ PDIP-24.....	16
2.14 รูปแสดงตัวอย่างการต่อวงจรใช้งานไอซี MAX7219 สำหรับ 7-Segment แบบหลายหลัก....	16
2.16 รูปแสดงตำแหน่งจุดเชื่อมต่อสัญญาณของโมดูล MAX7219.....	18
3.1 บล็อกไดอะแกรม.....	19
3.2 ขั้นตอนการทำงานของแอปพลิเคชัน.....	19
3.3 วงจร Wireless System Control LED Dot Matrix Display.....	20
3.4 วงจร Microcontroller.....	20
3.5 วงจร LED Dot Matrix.....	21
3.6 วงจร Bluetooth Module.....	22
3.6 ทำงานของแอปพลิเคชัน.....	27
4.1 หน้าจอการเปิดบลูทูธ.....	28
4.2 หน้าจอไอคอนแอปพลิเคชัน.....	28
4.3 ปุ่มเชื่อมต่อหน้าจอแอปพลิเคชัน.....	29
4.4 ปุ่มเชื่อมต่อหน้าจอแอปพลิเคชัน.....	29
4.5 แถบแสดงสถานะแอปพลิเคชัน.....	30
4.6 การพิมพ์ข้อความที่จะส่ง.....	30
4.7 ตัวอย่างการส่งตัวอักษร.....	31
4.8 แสดงผลการส่งตัวอักษร.....	31
4.9 ตัวอย่างการส่งประโยค.....	31
4.10 แสดงผลส่งประโยค.....	31
4.11 ตัวอย่างการส่งตัวเลข.....	32
4.12 แสดงผลการส่งตัวเลข.....	32
4.13 ตัวอย่างการส่งตัวสัญลักษณ์.....	32

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.14 แสดงผลการส่งตัวสัญลักษณ์.....	32
4.15 พัลส์ขา DIN ของ MAX7219.....	33
4.16 พัลส์ขา CLK ของ MAX7219.....	33
4.17 พัลส์ขา LOAD ของ MAX7219.....	33
5.1 ระบบควบคุมการส่งข้อมูลไร้สายด้วยความถี่ 2.4 GHz.....	36
5.2 หน้าจอแอปพลิเคชัน DTS.....	36

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

เนื่องจากในปัจจุบันเทคโนโลยีการสื่อสารได้พัฒนาไปเร็วมากและได้เข้ามาช่วยอำนวยความสะดวกให้แก่มนุษย์มากขึ้นเทคโนโลยีเหล่านี้พบได้ทั่วไปเช่นรีโมทคอนโทรลซึ่งเป็นอุปกรณ์ที่ใช้ควบคุมอุปกรณ์ไฟฟ้าในระยะไกลโดยไม่ใช้สายไฟเพิ่มความสะดวกรวดสบาย และลดค่าสายไฟ ถ้าเราต้องการควบคุมเครื่องใช้ไฟฟ้าในระยะหลายสิบบเมตร เราต้องต่อสายไฟจากตัวควบคุมไปถึงอุปกรณ์ไฟฟ้า (หน้าจอแสดงผล dotmatrix)และสายไฟต้องมีคุณภาพดีจึงทำให้ข้อมูลที่ส่งไม่ผิดเพี้ยน ทำให้เกิดระบบควบคุมการส่งข้อมูลไร้สายด้วยความถี่ 2.4 GHz

1.2 วัตถุประสงค์ของโครงการ

โครงการนี้มีวัตถุประสงค์ดังต่อไปนี้

- 1.2.1. ศึกษาเกี่ยวกับการส่งข้อมูลไร้สายเพื่อให้สามารถนำไปใช้ประยุกต์ใช้ในงานต่างๆได้
- 1.2.2. ศึกษาการเขียนแอปพลิเคชันในระบบแอนดรอย
- 1.2.3. ศึกษาการโปรแกรม arduino ในการใช้ MCU

1.3 ขอบเขตในโครงการ

ศึกษา ทำความเข้าใจเกี่ยวกับระบบส่งข้อมูลไร้สาย คำสั่งที่ใช้ในการควบคุมอุปกรณ์ไมโครคอนโทรลเลอร์ และ การสร้างแอปพลิเคชันบนระบบแอนดรอย ที่สามารถทำงานตามที่ต้องการได้ คือใช้แอปพลิเคชันในระบบแอนดรอยส่งข้อมูลไปยังคอนโทรลเลอร์ระยะไกลโดยผ่านย่านความถี่ 2.4GHz และวงจรและการแสดงผลของ led dot matrix display และการสร้าง application บน smartphone

1.4 ประโยชน์ที่จะได้รับ

ได้ความรู้และความเข้าใจเกี่ยวกับ โครงสร้างการใช้งาน และคำสั่งต่างๆของอุปกรณ์ไมโครคอนโทรลเลอร์ และสามารถนำไปประยุกต์ใช้งานต่างๆได้(ใช้ควบคุมจอแสดงผล dot matrix และเข้าใจการส่งข้อมูลแบบไร้สายในย่านความถี่ต่างๆและสามารถส่งข้อมูลได้จริง สามารถสร้างแอปพลิเคชันบนสมาร์ตโฟนในระบบแอนดรอยได้และ เรียนรู้วงจรและการแสดงผลของ led dot matrix display

1.5 รายละเอียดโดยย่อ

บทที่ 1 บทนำ เป็นการกล่าวถึงรายละเอียดความเป็นมาและวัตถุประสงค์ของโครงการ เพื่อให้เห็นภาพรวมก่อนที่จะเข้าสู่บทฤษฎีต่อไป

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึงรายละเอียด ทฤษฎีเบื้องต้นของอุปกรณ์ต่างๆที่ใช้ในวงจรอย่างคร่าวๆ

บทที่ 3 วงจร และโปรแกรมที่ใช้ กล่าวถึงรายละเอียดของวงจรในส่วนต่างๆ

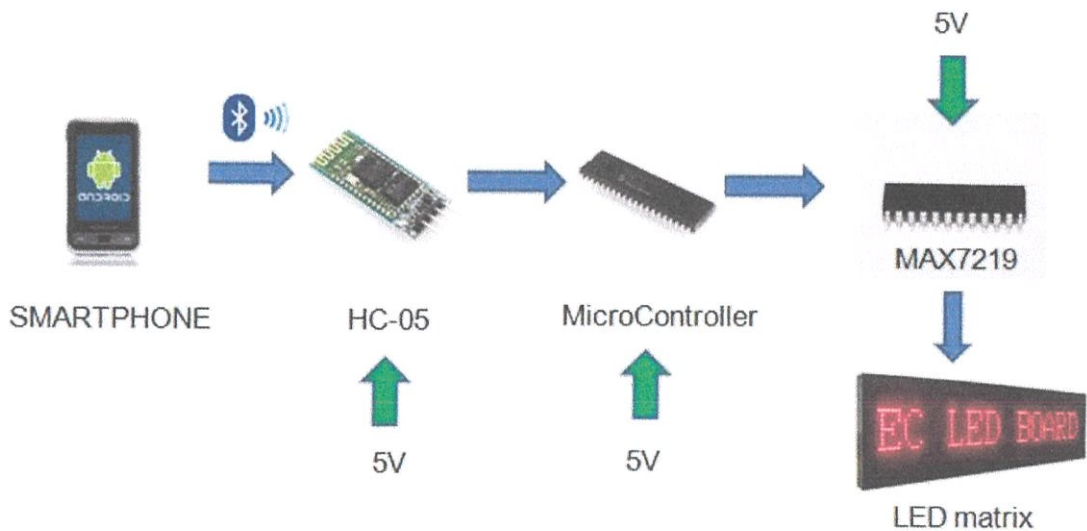
บทที่ 4 การทดลอง และผลการทดลอง จะกล่าวถึงวิธีการทดสอบ และผลการทดสอบหลังการนำวงจรและโปรแกรมควบคุมที่ได้สร้างจากบทที่ 3 แล้วทำการวิเคราะห์ผลที่เกิดขึ้น

บทที่ 5 สรุปผลการทดลอง จะสรุปผลลัพธ์ที่ได้จากการปฏิบัติโครงการ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

โครงการชิ้นนี้สามารถแสดงบล็อกไดอะแกรมการทำงานได้ดังรูปที่ 2.1 จึงมีทฤษฎีและหลักการที่เกี่ยวข้องด้วยกัน 4 หัวข้อหลักคือ Bluetooth , Android , Arduino , LED Dot Matrix



รูปที่ 2.1 บล็อกไดอะแกรม

2.1 Bluetooth

Bluetooth เป็นเทคโนโลยีของอินเทอร์เน็ตเฟสทางคลื่นวิทยุ (คือ การทำงานที่ใช้ไมโครชิพขนาด 9 มม. X 9 มม. ซึ่งทำงานเป็นตัวเชื่อมที่ใช้สัญญาณวิทยุขนาดระยะสั้นและมีราคาถูก) ใช้ในการเชื่อมโยงสื่อสารสายในแถบความถี่ 2.45GHz ทำให้อุปกรณ์อิเล็กทรอนิกส์ที่ถือเคลื่อนย้ายได้สามารถติดต่อเชื่อมโยงสื่อสารแบบไร้สายระหว่างกันในช่วงระยะห่างสั้น ๆ ได้

เทคโนโลยีการเชื่อมโยงหรือการสื่อสารแบบใหม่ที่ถูกคิดค้นขึ้น เป็นเทคโนโลยีของอินเทอร์เน็ตเฟสทางคลื่นวิทยุ ตั้งอยู่บนพื้นฐานของการสื่อสารระยะใกล้ที่ปลอดภัยผ่านช่องสัญญาณความถี่ 2.4 Ghz โดยที่ถูกพัฒนาขึ้นเพื่อลดข้อจำกัดของการใช้สายเคเบิลในการเชื่อมโยงโดยมีความเร็วในการเชื่อมโยงสูงสุดที่ 1 mbp ระยะครอบคลุม 10 เมตร เทคโนโลยีการส่งคลื่นวิทยุของบลูทูธจะใช้การกระโดดเปลี่ยนความถี่ (Frequency hop) เพราะเทคโนโลยีนี้เหมาะที่จะใช้กับการส่งคลื่นวิทยุที่มีกำลังส่งต่ำและราคาถูก โดยจะแบ่งออกเป็นหลายช่องความถี่ขนาดเล็ก ในระหว่างที่มีการเปลี่ยนช่องความถี่ที่ไม่แน่นอนทำให้สามารถหลีกเลี่ยงสัญญาณรบกวนที่เข้ามาแทรกแซงได้ ซึ่งอุปกรณ์ที่จะได้รับการยอมรับว่าเป็นเทคโนโลยีบลูทูธ ต้องผ่านการทดสอบจาก Bluetooth SIG (Special Interest Group) เสียก่อนเพื่อยืนยันว่ามันสามารถที่จะทำงานร่วมกับอุปกรณ์บลูทูธตัวอื่น ๆ และอินเทอร์เน็ตได้

เป้าหมายของ Bluetooth

เทคโนโลยีบลูทูธพัฒนาขึ้นมา โดยมีเป้าหมายคือ 4.1 Low cost implementation พัฒนาให้มีราคาต่ำ ที่สามารถให้คนทั่วไปใช้ได้ 4.2 Small implementation size ทำให้บลูทูธมีขนาดเล็กที่สุด เพื่อให้ใช้งานได้ง่าย 4.3 Low power consumption ให้บลูทูธใช้พลังงานในการทำงาน

น้อย เพื่อให้สามารถติดต่อกันได้โดยไร้ข้อจำกัด 4.4 Robust, high quality data & voice transfer พัฒนาให้บลูทูธมีความทนทานในการใช้งานและสามารถส่งทั้งข้อมูลและเสียงได้อย่างมีประสิทธิภาพ

4.5 Open global standard เป็นมาตรฐานเปิด คือให้ผู้ที่สนใจสามารถนำไปพัฒนาต่อได้ ทำให้เทคโนโลยีพัฒนาได้อย่างรวดเร็ว (Bluetooth 2001)

ลักษณะการทำงานของ Bluetooth

การอธิบายโดยย่อ คือ บลูทูธเป็นเทคโนโลยีของอินเทอร์เน็ตไร้สายทางคลื่นวิทยุ ใช้ในการเชื่อมโยงสื่อสารไร้สายในแถบความถี่ 2.45GHz ทำให้อุปกรณ์อิเล็กทรอนิกส์ที่เคลื่อนย้ายได้สามารถติดต่อสื่อสารกันแบบไร้สายระหว่างกันในระยะห่างสั้นๆ ได้ อุปกรณ์แต่ละตัวสามารถติดต่อสื่อสารกับอุปกรณ์อื่นๆ ได้สูงสุดถึง 7 ตัวพร้อมกัน เราเรียกเครือข่ายการติดต่อนี้ว่า Piconet ยิ่งไปกว่านั้น อุปกรณ์แต่ละตัวยังสามารถสังกัดอยู่กับเครือข่าย Piconet ได้หลายเครือข่ายพร้อมกันอีกด้วย

เทคโนโลยีการส่งคลื่นวิทยุของบลูทูธจะใช้การกระโดดเปลี่ยนความถี่ (Frequency hop) เพราะว่าเทคโนโลยีนี้เหมาะที่จะใช้กับการส่งคลื่นวิทยุที่มีกำลังส่งต่ำและราคาถูก โดยจะแบ่งออกเป็นหลายช่องความถี่ขนาดเล็ก ในระหว่างที่มีการเปลี่ยนช่องความถี่ที่ไม่แน่นอนทำให้สามารถหลีกเลี่ยงสัญญาณรบกวนที่เข้ามาแทรกแซงได้ Bluetooth จะใช้คลื่นความถี่ในการติดต่อในย่าน ISM (Industrial, Scientific, Medical) ที่มีความถี่ 2.4 กิกะเฮิร์ต และใช้พลังงานต่ำ โดยทางปฏิบัติแล้ว อุปกรณ์ของ Bluetooth นั้นจะมีพื้นที่การใช้งานไม่เกิน 10 เมตร โดยการติดต่อผ่านทางช่องสัญญาณที่สนับสนุนทั้งข้อมูล (อะซิงโครนัส) และเสียง (ซิงโครนัส) ที่ความเร็ว 741 Kbps

Wi-Fi vs. Bluetooth

ผู้ใช้มือใหม่มักจะสับสนกับเทคโนโลยี 2 ตัวนี้เสมอ ทั้งๆ ที่ความเป็นจริงทั้งสองเทคโนโลยีจะได้รับการออกแบบให้มีการใช้งานที่ต่างกันมาก แม้ว่าทั้งคู่จะใช้ความถี่ในแบนด์เดียวกันคือ 2.4GHz เอาเป็นว่า ตอนนี้ เรามาทำความรู้จักแต่ละเทคโนโลยีกันก่อนดีกว่า

Wi-Fi (หรือ 802.11 a/b/g) เป็นมาตรฐานที่ช่วยให้คุณสามารถเชื่อมต่ออินเทอร์เน็ตแบบไร้สายภายในบ้านหรือสำนักงานได้ โดยใช้อุปกรณ์อย่างเช่น Wi-Fi router เป็นต้น (นอกจากนี้ ปัจจุบันมีการเปิดให้บริการเชื่อมต่อไร้สายสาธารณะ หรือที่เรียกว่า Hot Spot ตามสถานที่ต่างๆ อย่างแพร่หลาย) การเชื่อมต่อไร้สายด้วยเทคโนโลยี Wi-Fi จะมีรัศมีการทำงานครอบคลุมสูงสุดถึง 100 ฟุต (ประมาณ 30 เมตร) จากตัวอุปกรณ์ โดยในส่วนของความเร็วที่ได้จะแตกต่างกันขึ้นอยู่กับมาตรฐานที่เลือกใช้ ซึ่งถ้าเป็น 802.11b จะได้สูงสุด 11Mbps ในขณะที่ถ้าเป็นมาตรฐาน 802.11g จะสามารถเชื่อมต่อที่ความเร็วได้สูงสุดถึง 125Mbps ซึ่งด้วยความเร็วระดับนี้จะเทียบเท่ากับการเชื่อมต่อด้วย Ethernet 100Mbps ที่ใช้กับเครือข่ายสำนักงานเลยทีเดียว

Bluetooth จะเป็นเทคโนโลยีสำหรับเครือข่ายส่วนบุคคล โดยจะมีรัศมีการทำงานค่อนข้างใกล้ เช่นแค่ภายในห้องเดียวกัน อีกทั้งยังมีความเร็วในการเชื่อมต่อที่ต่ำกว่าอีกด้วย (771 Kbps สำหรับ Bluetooth 1.x และ 2.1Mbps สำหรับ Bluetooth 2.0) มันได้รับการออกแบบมาเพื่อใช้เชื่อมต่อการทำงานแบบไร้สายระหว่างอุปกรณ์อย่างเช่น คีย์บอร์ด เมาส์ หูฟัง และเครื่องพิมพ์ กับคอมพิวเตอร์ พ็อกเก็ตพีซี และมือถือ ดังนั้นการเชื่อมต่อด้วย Bluetooth จึงไม่มีความจำเป็นต้องใช้งานที่ความเร็วระดับบรอดแบนด์ นอกจากนี้มันไม่ต้องการเราท์เตอร์อีกด้วย เนื่องจากเป็นการเชื่อมต่อระหว่างอุปกรณ์กับอุปกรณ์ โดยสนับสนุนการเชื่อมต่อพร้อมกันได้สูงสุด 8 ตัวด้วยกัน อย่างไรก็ตาม Wi-Fi จะสนับสนุนการเชื่อมต่อระหว่างคอมพิวเตอร์ด้วยกันเองเหมือนกัน (Peer-to-Peer ad-hoc network) แต่ส่วนใหญ่แล้วจะเป็นการเชื่อมต่อผ่านเราท์เตอร์ 5

ตารางที่ 2.1 เปรียบเทียบเทคโนโลยี Bluetooth กับ IEEE

802.11b	IEEE	Bluetooth
802.11b		
พื้นที่การใช้งาน	ภายในระยะประมาณ 100 เมตรบริเวณเปิด	ภายในระยะประมาณ 10 เมตรบริเวณเปิด
แบนด์วิดท์	11 Mbps และ สามารถใช้งาน WEP ได้ที่ 2-3 Mbps	721 Kbps
ผลต่อระดับสัญญาณ	คลื่นวิทยุ กำแพง สิ่งปลูกสร้าง อาคาร	คลื่นวิทยุ กำแพง สิ่งปลูกสร้าง อาคาร
ปริมาณกระแสไฟฟ้าที่ใช้	300mA	60mA
ความรู้ที่จำเป็นในการใช้งาน	ความรู้ในการคอนฟิก เน็ตเวิร์ก	ความรู้ในการเชื่อมต่อพีกับแอ็คเซสพอยต์
โปรโตคอลที่สนับสนุน	TCP/IP	TCP/IP, OBEX
จำนวนอุปกรณ์ที่สามารถใช้งานได้พร้อมกัน	อุปกรณ์เป็นจำนวนมาก	อุปกรณ์ 8 ชิ้น

2.2 HC-05

HC-05 Bluetooth Module เป็นโมดูลไร้สายที่ใช้สื่อสารกันด้วย Bluetooth SPP (Serial Port Protocol) โดย Serial Port เป็น Bluetooth V2.0+EDR (Enhance Data Rate) 3Mbps Modulation พร้อมกับความถี่ใช้งาน 2.4 GHz สามารถเชื่อมต่อกับอุปกรณ์บลูทูธอื่นๆได้ รวมทั้งเชื่อมต่อด้วย Serial Interface ระหว่างคอมพิวเตอร์ด้วยวงจร RS232 เพื่อเข้าสู่ AT Command ในการปรับแต่งค่าต่างๆของโมดูล

โมดูลรุ่นนี้สามารถตั้งค่าให้เป็นได้ทั้ง Master และ Slave Mode ซึ่งรายละเอียดจะอยู่ใน ส่วนของ AT Command

คุณสมบัติต่างๆ

- ความตอบสนองสัญญาณ -80dBm
- กำลังส่งคลื่นวิทยุเพิ่มได้ถึง +4dbm
- แรงดัน 1.8-3.6v
- PIO Control
- UART Interface พร้อมกับสามารถปรับแต่ง baud rate ได้
- เสอาอากาศในตัว

2.3 แอนดรอยด์ (Android)

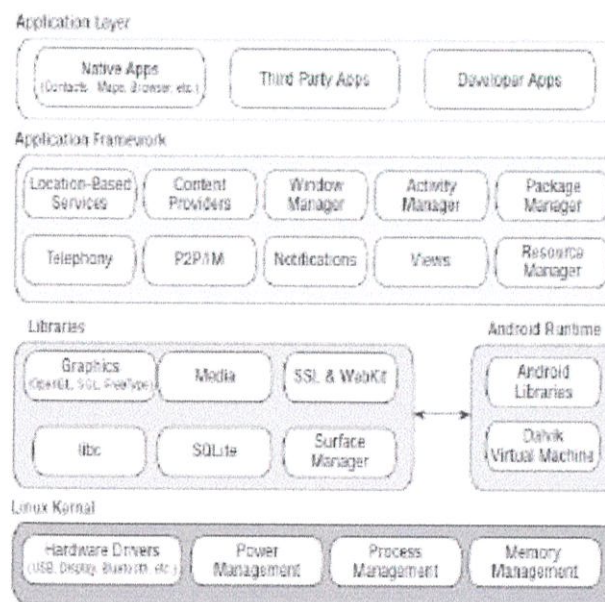
ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ (เรียกอีกอย่างว่า สมาร์ทโฟน-Smartphone) และคอมพิวเตอร์แบบแท็บเล็ต (Tablet Computer หรือ Tablet PC) เป็นต้น แอนดรอยด์นั้นพัฒนามาจากระบบปฏิบัติการลินุกซ์ (Linux) อีกทีหนึ่งแรกเริ่มนั้นแอนดรอยด์ถูกพัฒนาโดยบริษัท Android Inc. ซึ่งก่อตั้งในปี ค.ศ.2003ต่อมาในปี ค.ศ.2005 กูเกิลได้ซื้อกิจการของบริษัทนี้ จากนั้นได้ร่วมมือกับกลุ่มบริษัททางด้านฮาร์ดแวร์ ซอฟต์แวร์ และการสื่อสาร เช่น Intel, HTC, LG, Texas Instruments จัดตั้งองค์กรความร่วมมือที่มีชื่อว่า Open Handset Alliance ขึ้นในปี ค.ศ.2007 โดยมีวัตถุประสงค์ในการสร้างแพลตฟอร์ม (Platform) สำหรับอุปกรณ์พกพาที่มีพื้นฐานอยู่บนมาตรฐานเปิด (Open Standard) นอกจากนี้การที่แอนดรอยด์เป็นระบบเปิดหรือโอเพ่นซอร์ส (Open Source) ก็เปิดโอกาสให้ผู้ผลิตสามารถปรับแต่งแอนดรอยด์ให้เหมาะกับฮาร์ดแวร์ของตนเอง ได้การพัฒนาแอปพลิเคชันที่รันบนแอนดรอยด์จะใช้ภาษาจาวา โดยการเข้าถึงความสามารถต่างๆ ของแอนดรอยด์จะสามารถกระทำผ่าน Java Library ที่กูเกิลได้จัดเตรียมไว้ใน Android SDK หรือก็คือชุดพัฒนาซอฟต์แวร์สำหรับแอนดรอยด์นั่นเอง

2.3.1 สถาปัตยกรรมของแอนดรอยด์

ส่วนประกอบหลักๆ ของระบบปฏิบัติการแอนดรอยด์ถูกแสดงดังรูปที่ 2.11 โดยส่วนประกอบในแต่ละชั้นจะเรียกใช้บริการของชั้นที่อยู่ถัดลงไป

2.3.1.1 Linux Kernel

ส่วนที่เป็นแกนหลักหรือเคอร์เนล (Kernel) ของแอนดรอยด์นั้นความจริงก็คือเคอร์เนลของลินุกซ์ ซึ่งเป็นระบบปฏิบัติการที่สร้างโดย Linus Torvalds ในปี ค.ศ.1991 จะทำหน้าที่เป็น Hardware Abstraction Layer กล่าวคือเป็นตัวกลางระหว่างฮาร์ดแวร์กับซอฟต์แวร์ที่อยู่ถัดขึ้นไป และทำหน้าที่บริหารจัดการทรัพยากรต่างๆ ของเครื่อง เช่น การจัดการหน่วยความจำ การจัดการโพเซส ฯลฯ ผู้ผลิตอุปกรณ์สามารถ “พอร์ต” (port) แอนดรอยด์ให้ไปรันบนฮาร์ดแวร์แบบต่างๆ ได้ โดยเปลี่ยนแปลงในส่วนนี้



รูปที่ 2.4 สถาปัตยกรรมของแอนดรอยด์

2.3.1.2 Libraries

ถัดขึ้นมาจาก Linux Kernel ก็คือส่วนที่เป็นไลบรารีของแอนดรอยด์ ซึ่งทั้งหมดเขียนด้วยภาษา C หรือ C++ และถูกคอมไพล์มาสำหรับฮาร์ดแวร์ของอุปกรณ์แต่ละรุ่น ไลบรารีที่น่าสนใจมีดังนี้

- 1.) **Surface Manager** คือ ไลบรารีจัดการส่วนแสดงผลที่มีความสามารถในการผสมกราฟฟิกทั้ง 2 มิติและ 3 มิติจากแอปพลิเคชันต่างๆ เข้าด้วยกัน ทำให้สามารถสร้างเอฟเฟกต์ เช่น วินโดว์ที่มองทะลุไปข้างหลังได้และ Transition ในรูปแบบต่างๆ
- 2.) **Media Libraries** คือไลบรารีที่จัดเตรียมบริการในการเล่นและบันทึกเสียง วิดีโอ และรูปภาพในฟอร์แมตต่างๆ เช่น MPEG4, H.264, MP3, AAC, AMR, JPG และ PNG
- 3.) **SQLite** คือ Database Engine ที่มีประสิทธิภาพและมีขนาดเล็ก เพื่อให้เราจัดเก็บข้อมูลของแอปพลิเคชันไว้ในรูปแบบของฐานข้อมูลเชิงสัมพันธ์(Relational Database)
- 4.) **WebKit** คือไลบรารีที่ใช้แสดงเนื้อหาเว็บเพจ ซึ่งเป็นตัวเดียวกับที่ใช้ใน Google Chrome และ Apple Safari รวมถึงเว็บเบราว์เซอร์ในมือถือ iPhone และมือถือตระกูล S60 ของโนเกียด้วย

2.3.1.3 Android Runtime

- 1.) **Dalvik VM (Virtual Machine)** ส่วนนี้ถูกเขียนด้วยภาษา Java เพื่อใช้เฉพาะการใช้งานในอุปกรณ์เคลื่อนที่ Dalvik VM จะแตกต่างจาก Java VM(Virtual Machine) คือ Dalvik VM จะรันไฟล์ .dex ที่คอมไพล์มาจากไฟล์ .class และ .jar โดยมี tool ที่ชื่อว่า dx ทาหน้าที่ในการบีบอัดคลาส Java ทั้งนี้ไฟล์ .dex จะมีขนาดกะทัดรัดและเหมาะสมกับอุปกรณ์เคลื่อนที่มากกว่า .class เพื่อต้องการใช้พลังงานจากแบตเตอรี่อย่างมีประสิทธิภาพสูงสุด
- 2.) **Core Java Library** ส่วนนี้เป็นไลบรารีมาตรฐาน แต่ก็มีแตกต่างจากไลบรารีของ Java SE (Java Standard Edition) และ Java ME (Java Mobile Edition)

2.3.1.4 Application Framework

ในขั้นนี้จะอนุญาตให้นักพัฒนาสามารถเข้าเรียกใช้งาน โดยผ่าน API (Application Programming Interface) ซึ่ง Android ได้ออกแบบไว้เพื่อลดความซ้ำซ้อนในการใช้งาน application component โดยในขั้นนี้ประกอบด้วยแอปพลิเคชันเฟรมเวิร์คดังนี้

- 1.) **View System** เป็นส่วนที่ใช้ในการควบคุมการทำงานสำหรับการสร้างแอปพลิเคชัน เช่น lists, grids, text boxes, buttons และ embeddable web browser
- 2.) **Location Manager** เป็นส่วนที่จัดการเกี่ยวกับตำแหน่งของเครื่องอุปกรณ์พกพาเคลื่อนที่
- 3.) **Content Provider** เป็นส่วนที่ใช้ควบคุมการเข้าถึงข้อมูลที่มีการใช้งานร่วมกัน (Share data) ระหว่างแอปพลิเคชันที่แตกต่างกัน เช่น ข้อมูลผู้ติดต่อ(Contact)
- 4.) **Resource Manager** เป็นส่วนที่จัดการข้อมูลต่างๆ ที่ไม่ใช่ส่วนของโค้ดโปรแกรม เช่น รูปภาพ, localized strings, layout ซึ่งจะอยู่ในไดเรกทอรี res/
- 5.) **Notification Manager** เป็นส่วนที่ควบคุมอีเวนต์(Event) ต่างๆ ที่แสดงบนแถบสถานะ (Status bar) เช่น ในกรณีที่ได้รับข้อความหรือสายที่ไม่ได้รับและการแจ้งเตือนอื่นๆ เป็นต้น
- 6.) **Activity Manager** เป็นส่วนควบคุม Life Cycle ของแอปพลิเคชัน

2.3.1.5 Application

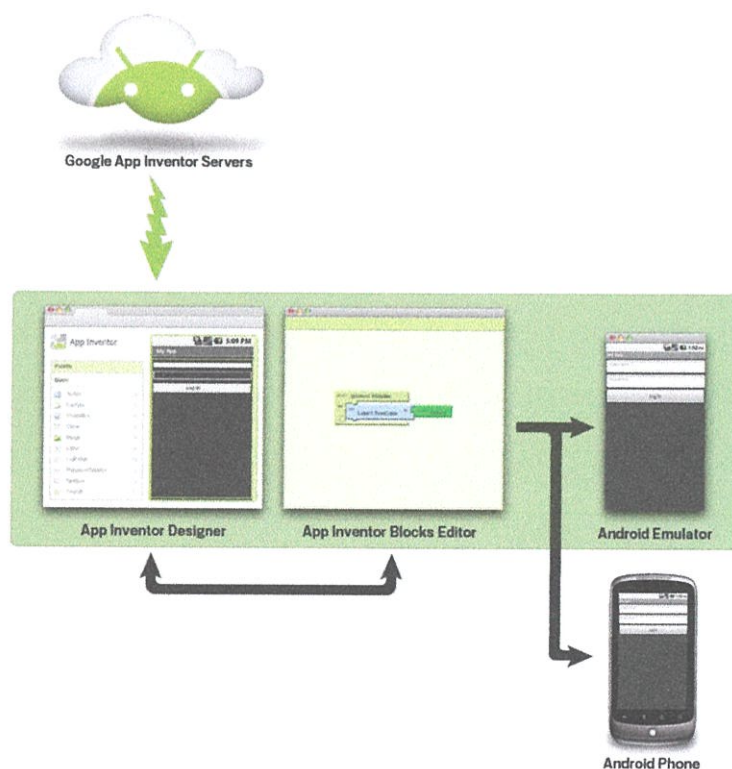
ส่วนบนสุดของสถาปัตยกรรมแอนดรอยด์ ก็คือแอปพลิเคชันต่างๆ ที่ติดตั้งมากับเครื่องอยู่แล้ว (Core Application) เช่น Phone dialer, Email, Contacts, Web browser และ Google Play (Android Market เดิม) เป็นต้น รวมถึงแอปพลิเคชันที่เราสร้างขึ้นด้วย ซึ่งแอปพลิเคชันทั้งหมดในส่วนนี้จะเขียนด้วยภาษาจาวา อยู่ในรูปแบบของไฟล์ .apk โดยทั่วไปแล้วจะอยู่ในไดเรกทอรี data/app

2.3.2 MIT App inventor

App Inventor เป็นเครื่องมือที่ใช้สำหรับสร้างแอปพลิเคชันสำหรับสมาร์ทโฟนและแท็บเล็ตที่เป็นระบบปฏิบัติการ Android ซึ่งบริษัท Google ร่วมมือกับ MIT พัฒนาโปรแกรม App inventor ขึ้น ต่อมา Google ถอนตัวออกมาและยกให้ MIT พัฒนาต่อเองในนาม MIT App inventor

App inventor ใช้หลักการคล้ายๆ กับ Scratch แต่ซับซ้อนกว่า โดยลักษณะการเขียนโปรแกรมแบบ Visual Programming คือ เขียนโปรแกรมด้วยการต่อบล็อกคำสั่ง เน้นการออกแบบเพื่อแก้ปัญหา (problem solving) ด้วยการสร้างโปรแกรมที่ผู้เรียนสนใจ บนโทรศัพท์มือถือสมาร์ทโฟน (สมัยนี้สมาร์ทโฟนใช้กันทั่วไปอยู่แล้ว โดยเฉพาะเด็กวัยรุ่น)

App inventor จึงเป็นอีกโปรแกรมหนึ่ง ที่เหมาะสำหรับใช้ในการสอนเขียนโปรแกรมให้นักเรียนในระดับมัธยมปลาย หรือระดับมหาวิทยาลัย โดยเฉพาะผู้ที่ไม่เคยเขียนโปรแกรมมาก่อนหรือไม่ได้เรียนอยู่ในสาย คอมพิวเตอร์ การเขียนโปรแกรมบนสมาร์ทโฟนและแท็บเล็ต Android ด้วย App inventor ในภาพรวมแสดงได้ตามรูปด้านล่างนี้



รูปที่ 2.5 ตัวอย่าง โปรแกรม 2 MIT App inventor

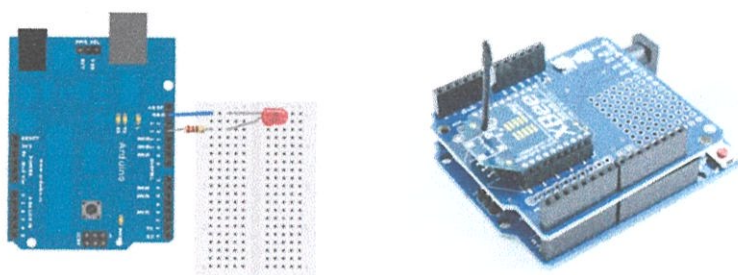
App Inventor servers เป็นเครื่องที่ให้บริการและเก็บงานโปรเจกต์ต่างๆ ที่ผู้ใช้สร้างขึ้นมา ผู้ใช้พัฒนาโปรแกรมมือถือ Android โดยสร้างโปรเจกต์และเขียนโปรแกรมบนเว็บเบราว์เซอร์ที่เชื่อมต่อไปยัง App Inventor servers เมื่อได้โปรแกรมมา ก็สามารถทดสอบกับโปรแกรมมือถือจำลอง (Android emulator) หรือโทรศัพท์มือถือ Android จริงๆ ก็ได้

ขั้นตอนการสร้างโปรแกรม (ตามภาพ) เริ่มจากออกแบบหน้าตาโปรแกรมบนมือถือ ด้วยโปรแกรม App Inventor Designer ซึ่งใช้สำหรับสร้างส่วนโปรแกรมต่างๆ (components) เพื่อใช้งานในโปรแกรมมือถือที่จะสร้างขึ้นจากนั้นเขียนโปรแกรมให้แต่ละส่วนโปรแกรม ด้วยโปรแกรม App Inventor Blocks Editor ซึ่งใช้วิธีการต่อบล็อกคำสั่ง เพื่อให้ส่วนโปรแกรมนั้นๆ ทำหน้าที่ของมันตามทีออกแบบเอาไว้ ระหว่างเขียนโปรแกรม อาจมีการแก้ไข เพิ่มเติม หรือลบบางส่วนโปรแกรมออกไป ทำให้ต้องแก้ไขโปรแกรม (debug) จนกว่าจะได้โปรแกรมตามที่ออกแบบไว้ เมื่อทุกส่วนโปรแกรมถูกสร้างเสร็จแล้ว ก็ได้เวลาทดสอบการใช้งาน โดยการติดตั้งโปรแกรมลงบนมือถือ Android แล้วทดสอบการใช้งานผ่านมือถือจริงๆ แต่ถ้าไม่มีมือถือ ก็ยังสามารถทดสอบได้ ผ่านโปรแกรมมือถือจำลอง (Android emulator) ในคอมพิวเตอร์แทน

2.4 Arduino

Arduino อ่านว่า (อา-ดู-อี-โน้ หรือ อาดูยโน้) เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ตัวบอร์ดถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้น ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลงเพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ความง่ายของบอร์ด Arduino ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด (ดูตัวอย่างรูปที่ 1) หรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ (ดูตัวอย่างรูปที่ 2) เช่น XBee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาเปรียบกับบอร์ดบนบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

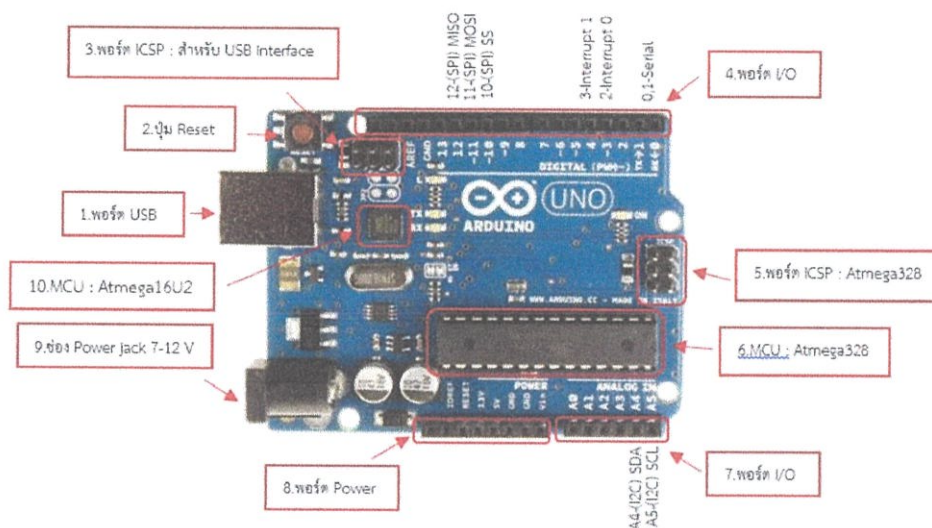


รูปที่ 2.6 การเชื่อมต่อโมดูลกับ บอร์ด Arduino

จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

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

Layout & Pin out Arduino Board (Model: Arduino UNO R3)



รูปที่ 2.7 PIN ต่างๆของบอร์ด Arduino

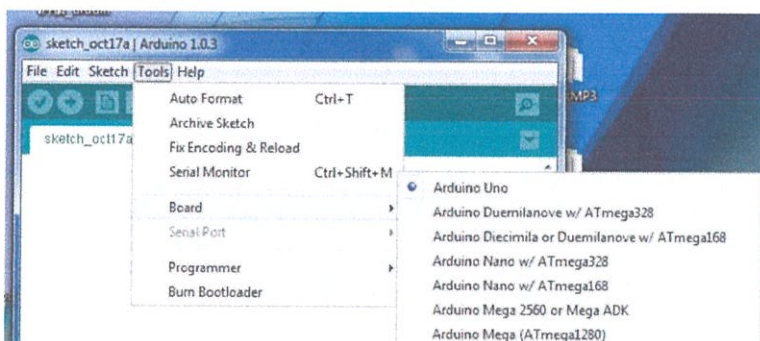
1. USB Port: ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button: เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2
4. I/O Port: Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin 0,1 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/O Port: นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
8. Power Port: ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, V_{in}
9. Power Jack: รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

จุดเด่นที่ทำให้บอร์ด Arduino เป็นที่นิยม

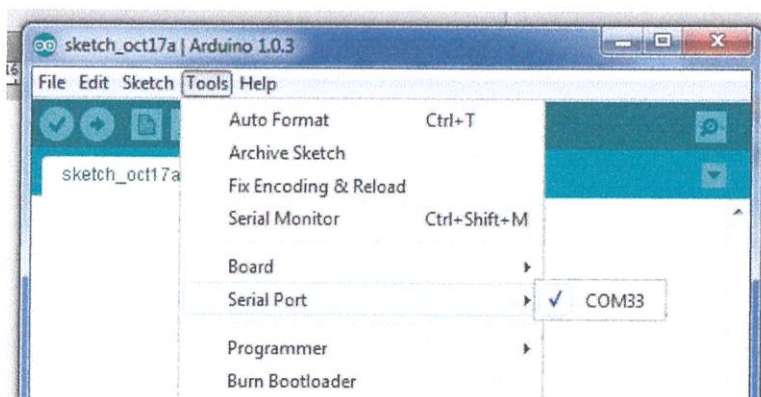
- ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
- มี Arduino Community กลุ่มคนที่ร่วมกันพัฒนาที่แข็งแกร่ง
- Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
- ราคาไม่แพง
- Cross Platform สามารถพัฒนาโปรแกรมบน OS ใดก็ได้

รูปแบบการเขียนโปรแกรมบน Arduino

1. เขียนโปรแกรมบนคอมพิวเตอร์ ผ่านทางโปรแกรม ArduinoIDE
2. หลังจากเขียนโค้ดโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกรุ่นบอร์ด Arduino ที่ใช้และหมายเลข Com port

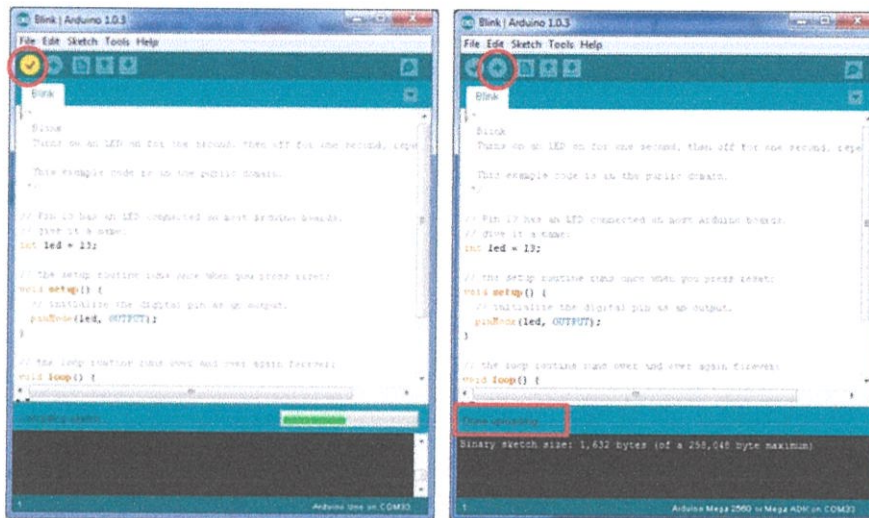


รูปที่ 2.8 เลือกรุ่นบอร์ด Arduino ที่ต้องการ upload



รูปที่ 2.9 เลือกหมายเลข Comport ของบอร์ด

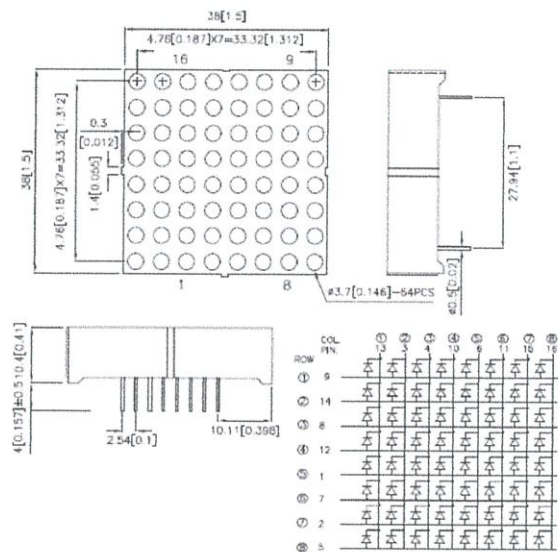
3. กดปุ่ม Verify เพื่อตรวจสอบความถูกต้องและ Compile โค้ดโปรแกรม จากนั้นกดปุ่ม Upload โค้ด โปรแกรมไปยังบอร์ด Arduino ผ่านทางสาย USB เมื่ออัปโหลดเรียบร้อยแล้ว จะแสดงข้อความ แถบข้างล่าง “Done uploading” และบอร์ดจะเริ่มทำงานตามที่เขียนโปรแกรมไว้ได้ทันที



รูปที่ 2.10 การ Compiler และ Up Load Code

2.5 LED Dot Matrix

การควบคุมหลอดไฟ LED Matrix (Dot Matrix) ขนาดต่างๆ เช่น 8x8, 8x16, 8x32, 16x16, 16x32 เป็นต้น เป็นอีกหนึ่งอุปกรณ์ที่สามารถใช้ในการฝึกเขียนโปรแกรม ไมโครคอนโทรลเลอร์ โดยปกติการเขียนโปรแกรมควบคุม LED Matrix 8x8 ต้องใช้ขาสัญญาณ ทั้งหมด 16 ขาด้วยกันแบ่งออกเป็น 8 ขาสัญญาณในแนวนอน (Rows) และ 8 ขาสัญญาณในแนวตั้ง (Columns) แต่ในปัจจุบันมีการพัฒนาโดยนำ IC เข้ามาต่อเพิ่ม เพื่อลดจำนวนขาที่ใช้ในการต่อหรือ การควบคุมให้น้อยลงและเขียนโปรแกรมให้ง่าย ขึ้น IC นี้คือ MAX7219CNGใช้กันก็มี Schematic แบบภาพด้านล่างนี้มี 8 Pin คุม Column และ 8 Pin คุม Row 64 LEDแล้วใช้16 Pin

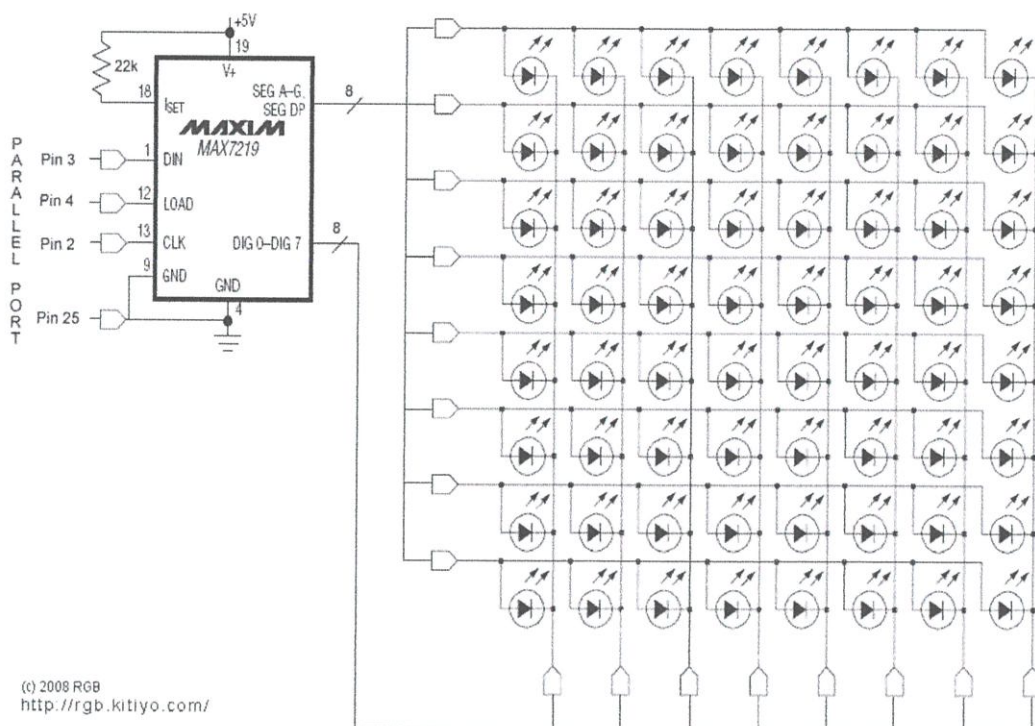


รูปที่ 2.11 วงจร LED Dot Matrix 8x8

เทคนิค การเปลี่ยนแปลงอะไรก็ตามที่เร็วกว่า 20 ms ตาของมนุษย์จะไม่สามารถแยกแยะได้ เช่น ถ้าหลอด LED กระพริบเร็วโดยมีช่วงเวลาระหว่างที่ติดและดับสั้นกว่า 20 ms แล้วสายตาของคนจะไม่เห็นว่ามันกระพริบ เราเรียกปรากฏการณ์นี้ว่า "Persistence of Vision" ดังนั้นเราจึง Drive LED 8x8 Matrix นี้โดยการควบคุมมันทีละแถว จากนั้นเปลี่ยนแถวไปเรื่อยๆ จนครบถึงแถวที่ 8 แล้วเริ่มบังคับใหม่ที่แถวแรก เป็นอย่างนี้เรื่อยๆไป โดยการควบคุมที่ว่าจะต้องกวาดไปให้ครบทุกแถวภายใน 20 ms ทำแบบนี้เราจะเห็นภาพเต็มทั้ง 64 ดวง ตามที่บังคับ โดยไม่ต้องควบคุมพร้อมๆกันทุกดวง

วิธีการก็คือ

การส่งข้อมูลเข้าไปทีละ 1 บิต จนครบแล้วค่อยให้ไปเรียงออกที่ขาทั้ง 16 ของ LED Matrix พร้อมๆกัน ซึ่งจะได้การควบคุม 1 Row การทำงานแบบนี้เราต้องใช้ IC ที่เรียกว่า "Shift Register" ถ้าใครเรียน Digital Design หรือ Logic Design ก็คงจะเคยได้ยินกันบ้าง ประกอบการส่งข้อมูลแบบ SPI (Serial Peripheral Interface) รวมร่างใน IC ตัวเดียว กลายเป็นว่าเราจะใช้สายเหลือ 5 เส้นต่อแผง (VCC GND DIN CS และ CLK) จากนั้นก็ใช้ Library บน Arduino เราจะใช้ของ MAX7219CNG ตัวนี้เราสามารถใช้งานได้กับทั้ง 7 segment LED 2 digit ก็ได้ หรือ จะเอามาใช้กับ LED 8x8 Matrix

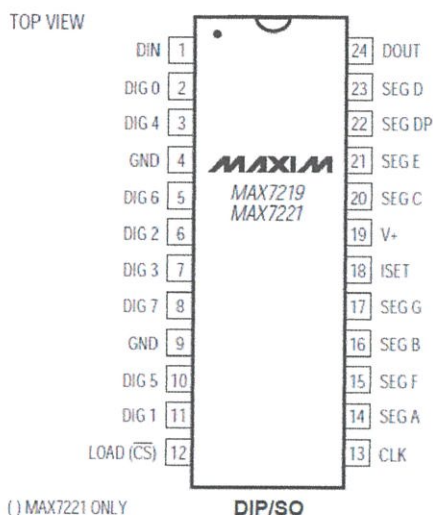


รูปที่ 2.12 การต่อ MAX7219 กับ LED Dot Matrix 8x8

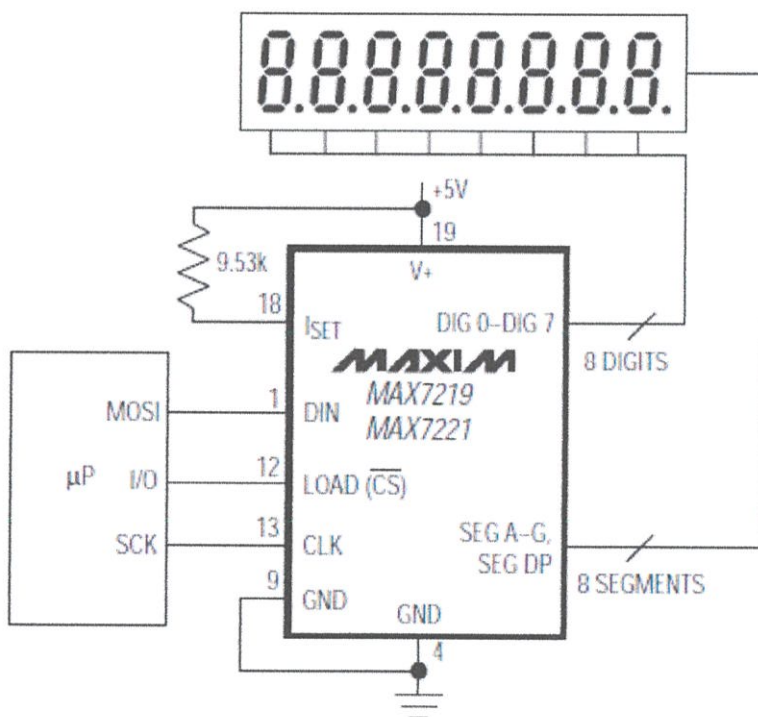
แผง LED แบบ Common Cathode (CC) ขนาด 8x8 ก็ได้ และสามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านบัส SPI ในทลวงนี้ได้เลือกใช้โมดูลที่มีไอซี MAX7219 และแผง LED ขนาด 8x8 พร้อมใช้งาน ทำให้ไม่ต้องเสียเวลาในการต่อวงจร

ข้อมูลเชิงเทคนิคของไอซี MAX7219 ที่สำคัญมีดังนี้ (อ้างอิงจาก MAX7219 datasheet)

- ใช้แรงดันไฟเลี้ยงได้ในช่วง 4V ถึง 5.5V
- ขา LOAD (/CS), CLK, DIN, DOUT เป็นขาสำหรับการเลื่อนบิตข้อมูลตามจังหวะสัญญาณ CLK เข้าที่ขา DIN และมีข้อมูลออกที่ขา DOUT
- ขา DIG0, ..., DIG7 สามารถรับกระแสได้สูง (Current Sink) และนำไปใช้ต่อกับขา Common Cathode เพื่อเลือก DIGIT (เลือกให้หลักใดแสดงผล หรือ "active")
- ขา SEG_A, ..., SEG_G, SEG_DP เป็นขาสำหรับกำหนดสถานะติด-ดับของ LED ที่ตรงกับแต่ละ Segment
- ขา ISET ให้ต่อตัวต้านทาน (เช่น 10k โอห์ม) ไปยัง VCC ใช้กำหนดกระแสสูงสุดของแต่ละ Segment (LED)



รูปที่ 2.13 แสดงตำแหน่งขาของไอซี MAX7219 ตัวถังแบบ PDIP-24



รูปที่ 2.14 รูปแสดงตัวอย่างการต่อวงจรใช้งานไอซี MAX7219 สำหรับ

7-Segment แบบหลายหลัก

2.6 MAX7219

การส่งข้อมูลเข้าไปในไอซี MAX7219 เพื่อเขียนลงในรีจิสเตอร์แต่ละตัว จะใช้วิธีเลื่อนบิตตามจังหวะของสัญญาณ CLK โดยส่งคร่าวละ 16 บิต และส่งบิต D15 เข้าไปก่อนตามลำดับ ซึ่งรูปแบบการส่งข้อมูลแบบนี้สอดคล้องกับการทำงานของ SPI ในโหมด 0

ตารางที่ 2.2 แสดงตำแหน่งของบิตในรีจิสเตอร์สำหรับคำสั่งและข้อมูลขนาด 16 บิต

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	ADDRESS				MSB		DATA						LSB

ตารางที่ 2.3 แสดงเลขที่อยู่ของรีจิสเตอร์ภายในของไอซี MAX7219

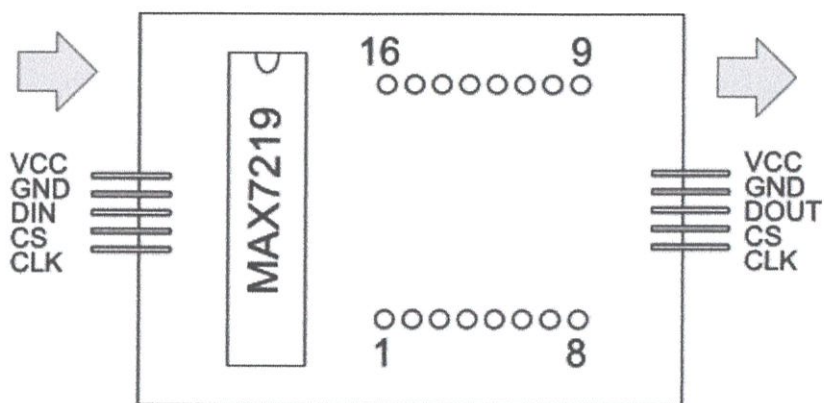
REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	X0
Digit 0	X	0	0	0	1	X1
Digit 1	X	0	0	1	0	X2
Digit 2	X	0	0	1	1	X3
Digit 3	X	0	1	0	0	X4
Digit 4	X	0	1	0	1	X5
Digit 5	X	0	1	1	0	X6
Digit 6	X	0	1	1	1	X7
Digit 7	X	1	0	0	0	X8
Decode Mode	X	1	0	0	1	X9
Intensity	X	1	0	1	0	XA
Scan Limit	X	1	0	1	1	XB
Shutdown	X	1	1	0	0	XC
Display Test	X	1	1	1	1	XF

ขั้นตอนการใช้งานสำหรับแผง LED 8x8 นั้นก็ทำได้ง่าย โดยกำหนดค่าสำหรับรีจิสเตอร์ดังนี้

- รีจิสเตอร์ "Decode Mode" (0x9): 0xFF ซึ่งหมายถึง Decode for digits 7-0
- รีจิสเตอร์ "Shutdown" (0xC) : 0x01 ซึ่งหมายถึง Normal operation
- รีจิสเตอร์ "Intensity" (0xA) : 0x00 .. 0x0F ซึ่งหมายถึง ระดับความสว่างของ LED ซึ่งสามารถเลือกได้ 16 ระดับ 0x00 = ต่ำสุด และ 0x0F = สูงสุด
- รีจิสเตอร์ "Scan Limit" (0xB) : 0x07 ซึ่งหมายถึง ใช้ขา DIG0 ถึง DIG7 ในการ "สแกน" (scan) โดยอัตโนมัติ

- รีจิสเตอร์ "Digit 0" .. "Digit 7" : ให้กำหนดค่าสำหรับสถานะติด-ดับของ LED จำนวน 8 ดวง ของหลักที่1ถึง8

ไอซี MAX7219 จะทำขั้นตอน "scan" โดยอัตโนมัติ โดยเลือกแถวแนวนอนทีละแถวตามลำดับ และส่งข้อมูลที่อยู่ในรีจิสเตอร์ ออกที่ขา SEG_A,.. , SEG_G, SEG_DP สำหรับแถวแนวนอนที่ได้เลือก ในแต่ละช่วงเวลา

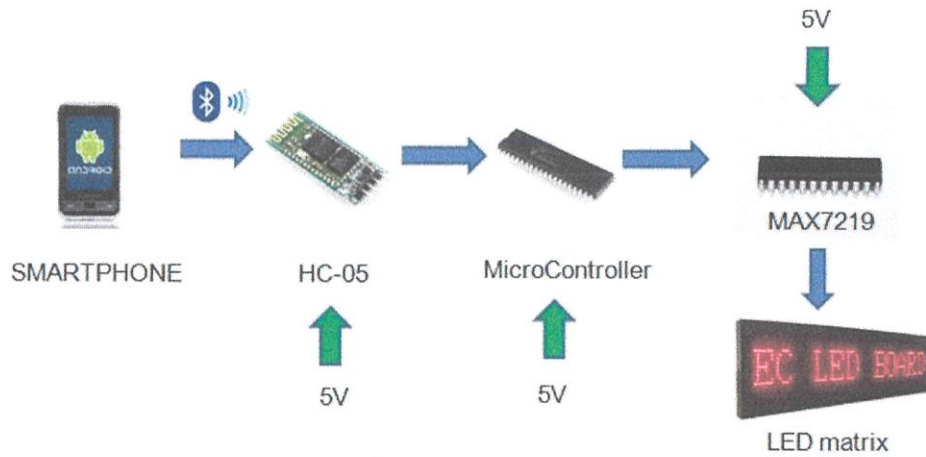


รูปที่ 2.16 รูปแสดงตำแหน่งจุดเชื่อมต่อสัญญาณของโมดูล MAX7219

ขา 1-8 แถวล่าง จากซ้ายไปขวา: 1=DIG4, 2=DIG6, 3=D6, 4=D5, 5=DIG7, 6=D3, 7=DIG5, 8=DIG2

ขา 9-16 แถวบน จากขวาไปซ้าย: 9=DIG0, 10=D4, 11=D2, 12=DIG3, 13=D7, 14=DIG1, 15=D1, 16=D0

บทที่ 3 วงจรและโปรแกรมที่ใช้



รูปที่ 3.1 บล็อกไดอะแกรม

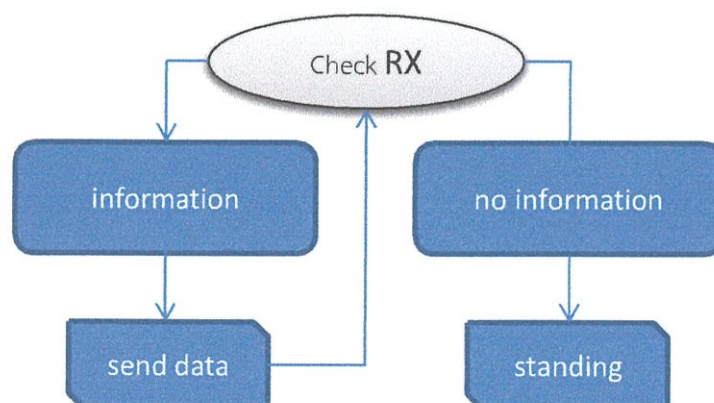
การทำงานของระบบเริ่มต้นจากการกดปุ่มบนหน้าจอแอปพลิเคชันของโทรศัพท์เคลื่อนที่ ระบบปฏิบัติการแอนดรอยด์ จะทำการส่งข้อมูลที่รับมาไปยัง ตัวบลูทูธโมดูลที่เชื่อมต่อกับ ไมโครคอนโทรลเลอร์ ตัวไมโครคอนโทรลเลอร์จะทำการวิเคราะห์ข้อมูลที่รับมาแล้วเปลี่ยนเป็นคำสั่งในการควบคุม LED Dot Matrix

จากระบบข้างต้น จะสามารถแบ่งงานออกเป็น 4 ส่วนหลักๆ คือ การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์, การสื่อสารข้อมูลผ่าน Bluetooth, การวิเคราะห์ข้อมูลของไมโครคอนโทรลเลอร์ และควบคุม LED Dot Matrix

3.1 การออกแบบ

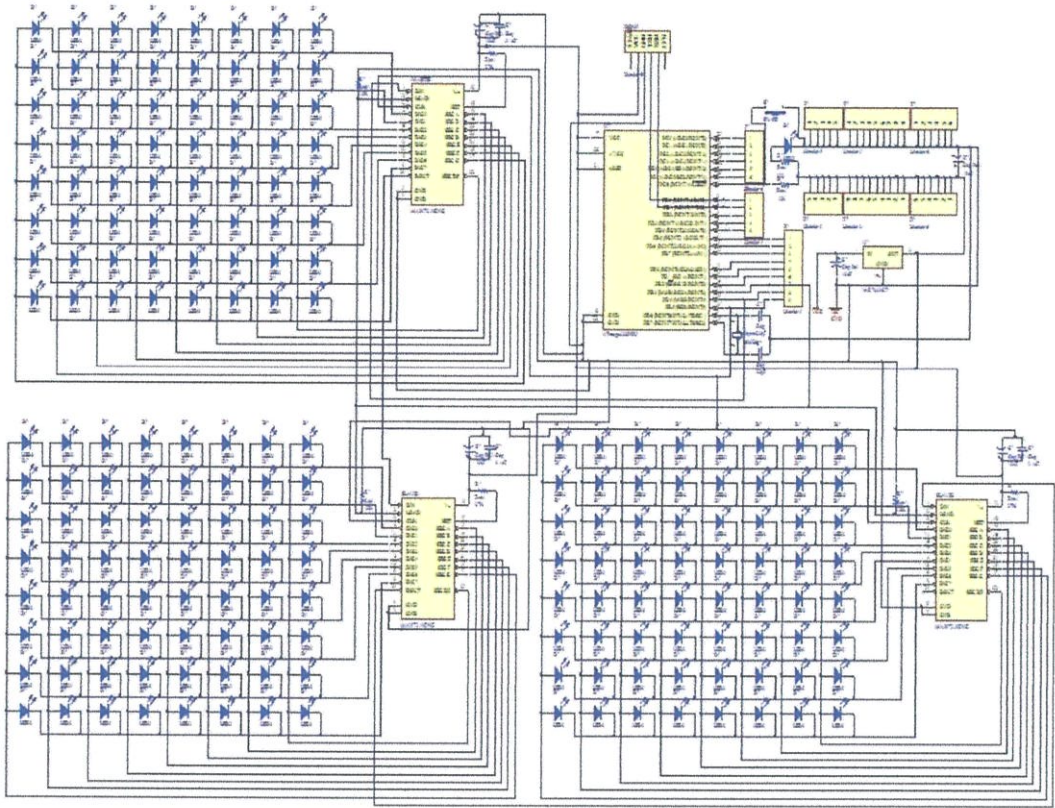
3.1.1 การสร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์

เขียนแผนผังแสดงขั้นตอนการทำงานของแอปพลิเคชันดังรูป



รูปที่ 3.2 ขั้นตอนการทำงานของแอปพลิเคชัน

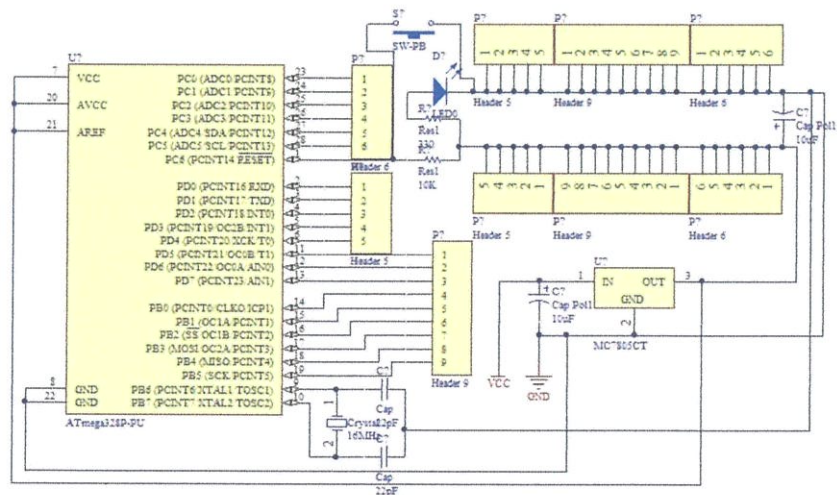
3.1.2 การออกแบบวงจร Wireless System Control LED Dot Matrix Display



รูปที่ 3.3 วงจร Wireless System Control LED Dot Matrix Display

ส่วนต่างๆของวงจร

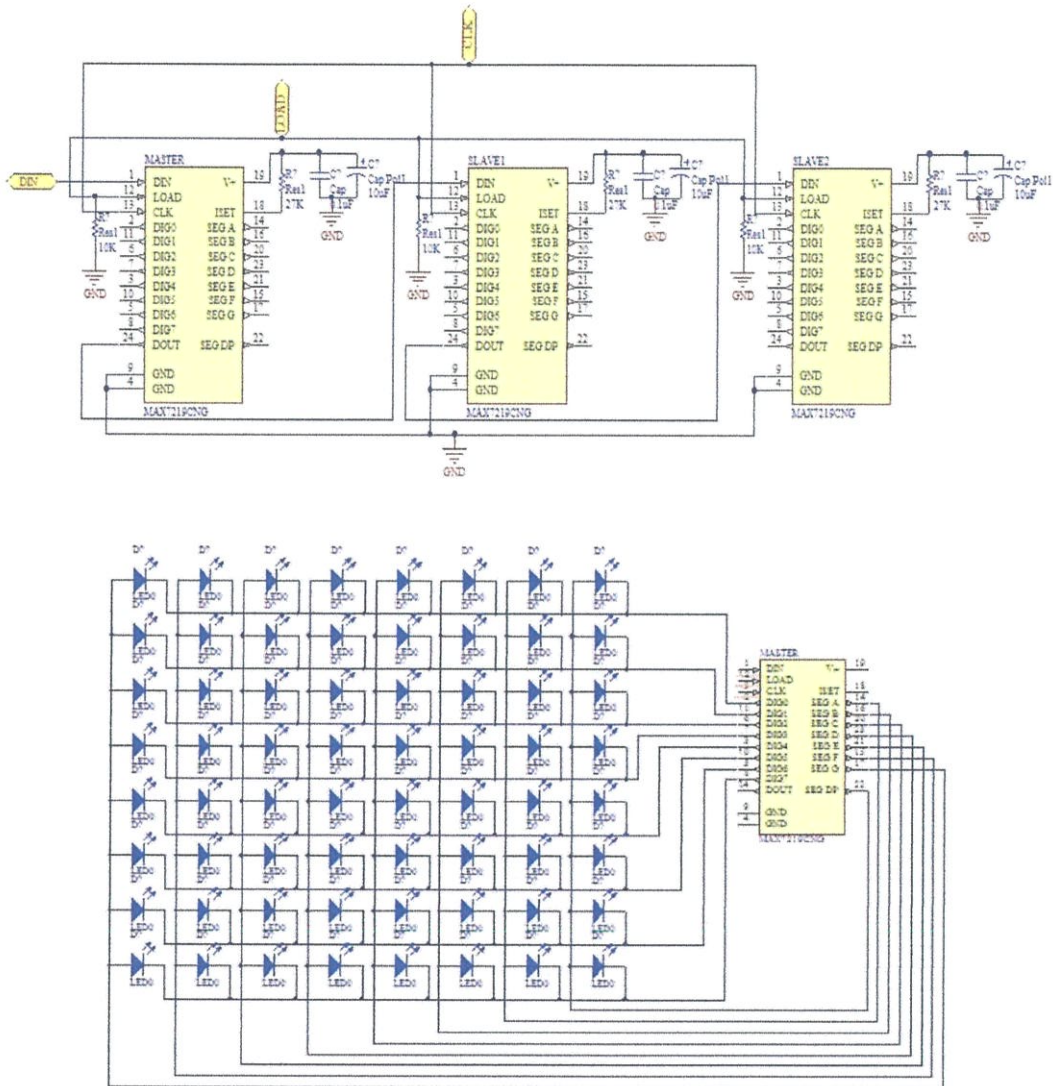
3.1.2.1 วงจร Microcontroller



รูปที่ 3.4 วงจร Microcontroller

วงจร Microcontroller ใช้ ATmega328P-PU เป็นส่วนประกอบหลักของวงจร ทำหน้าที่ประมวลผลต่างๆ ให้ไฟเลี้ยงผ่านถ่านไฟฉาย 9V 2 ก้อน ขาที่ใช้มี 9 ขาคือ GND 2 ขา, VCC 2 ขา, Tx, Rx, LOAD, DIN, CLK

3.1.2.2. วงจร LED Dot Matrix



รูปที่ 3.5 วงจร LED Dot Matrix

วงจร LED Dot Matrix ใช้ MAX7219 รับข้อมูลที่ได้จากไมโครคอนโทรลเลอร์มาทำวิเคราะห์แล้วเปลี่ยนเป็นคำสั่งในการควบคุม LED Dot Matrix ขาที่ใช้ทั้งหมด 16 ขา โดยขาบวกของ LED Dot Matrix ต่อกับขา SEG DP – SEG G และขาลบ ต่อกับขา DIG 0 – DIG 7

2, 8, B10110000, B01110000, B00000000, B00000000, B00000000, // ,
4, 8, B00001000, B00001000, B00001000, B00001000, B00000000, // -
2, 8, B01100000, B01100000, B00000000, B00000000, B00000000, // .
4, 8, B01100000, B00011000, B00000110, B00000001, B00000000, // /
4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // 0
3, 8, B01000010, B01111111, B01000000, B00000000, B00000000, // 1
4, 8, B01100010, B01010001, B01001001, B01000110, B00000000, // 2
4, 8, B00100010, B01000001, B01001001, B00110110, B00000000, // 3
4, 8, B00011000, B00010100, B00010010, B01111111, B00000000, // 4
4, 8, B00100111, B01000101, B01000101, B00111001, B00000000, // 5
4, 8, B00111110, B01001001, B01001001, B00110000, B00000000, // 6
4, 8, B01100001, B00010001, B00001001, B00000111, B00000000, // 7
4, 8, B00110110, B01001001, B01001001, B00110110, B00000000, // 8
4, 8, B00000110, B01001001, B01001001, B00111110, B00000000, // 9
2, 8, B01010000, B00000000, B00000000, B00000000, B00000000, // :
2, 8, B10000000, B01010000, B00000000, B00000000, B00000000, // ;
3, 8, B00010000, B00101000, B01000100, B00000000, B00000000, // <
3, 8, B00010100, B00010100, B00010100, B00000000, B00000000, // =
3, 8, B01000100, B00101000, B00010000, B00000000, B00000000, // >
4, 8, B00000010, B01011001, B00001001, B00000110, B00000000, // ?
5, 8, B00111110, B01001001, B01010101, B01011101, B00001110, // @
4, 8, B01111110, B00010001, B00010001, B01111110, B00000000, // A
4, 8, B01111111, B01001001, B01001001, B00110110, B00000000, // B
4, 8, B00111110, B01000001, B01000001, B00100010, B00000000, // C
4, 8, B01111111, B01000001, B01000001, B00111110, B00000000, // D
4, 8, B01111111, B01001001, B01001001, B01000001, B00000000, // E
4, 8, B01111111, B00001001, B00001001, B00000001, B00000000, // F
4, 8, B00111110, B01000001, B01001001, B01111010, B00000000, // G
4, 8, B01111111, B00001000, B00001000, B01111111, B00000000, // H
3, 8, B01000001, B01111111, B01000001, B00000000, B00000000, // I
4, 8, B00110000, B01000000, B01000001, B00111111, B00000000, // J
4, 8, B01111111, B00001000, B00010100, B01100011, B00000000, // K
4, 8, B01111111, B01000000, B01000000, B01000000, B00000000, // L

5, 8, B01111111, B00000010, B00001100, B00000010, B01111111, // M
 5, 8, B01111111, B00000100, B00001000, B00010000, B01111111, // N
 4, 8, B00111110, B01000001, B01000001, B00111110, B00000000, // O
 4, 8, B01111111, B00001001, B00001001, B00000110, B00000000, // P
 4, 8, B00111110, B01000001, B01000001, B01011110, B00000000, // Q
 4, 8, B01111111, B00001001, B00001001, B01110110, B00000000, // R
 4, 8, B01000110, B01001001, B01001001, B00110010, B00000000, // S
 5, 8, B00000001, B00000001, B01111111, B00000001, B00000001, // T
 4, 8, B00111111, B01000000, B01000000, B00111111, B00000000, // U
 5, 8, B00001111, B00110000, B01000000, B00110000, B00001111, // V
 5, 8, B00111111, B01000000, B00111000, B01000000, B00111111, // W
 5, 8, B01100011, B00010100, B00001000, B00010100, B01100011, // X
 5, 8, B00000111, B00001000, B01110000, B00001000, B00000111, // Y
 4, 8, B01100001, B01010001, B01001001, B01000111, B00000000, // Z
 2, 8, B01111111, B01000001, B00000000, B00000000, B00000000, // [
 4, 8, B00000001, B00000110, B00011000, B01100000, B00000000, // \ backslash
 2, 8, B01000001, B01111111, B00000000, B00000000, B00000000, //]
 3, 8, B00000010, B00000001, B00000010, B00000000, B00000000, // hat
 4, 8, B01000000, B01000000, B01000000, B01000000, B00000000, // _
 2, 8, B00000001, B00000010, B00000000, B00000000, B00000000, // `
 4, 8, B00100000, B01010100, B01010100, B01111000, B00000000, // a
 4, 8, B01111111, B01000100, B01000100, B00111000, B00000000, // b
 4, 8, B00111000, B01000100, B01000100, B00101000, B00000000, // c
 4, 8, B00111000, B01000100, B01000100, B01111111, B00000000, // d
 4, 8, B00111000, B01010100, B01010100, B00011000, B00000000, // e
 3, 8, B00000100, B01111110, B00000101, B00000000, B00000000, // f
 4, 8, B10011000, B10100100, B10100100, B01111000, B00000000, // g
 4, 8, B01111111, B00000100, B00000100, B01111000, B00000000, // h
 3, 8, B01000100, B01111101, B01000000, B00000000, B00000000, // i
 4, 8, B01000000, B10000000, B10000100, B01111101, B00000000, // j
 4, 8, B01111111, B00010000, B00101000, B01000100, B00000000, // k
 3, 8, B01000001, B01111111, B01000000, B00000000, B00000000, // l
 5, 8, B01111100, B00000100, B01111100, B00000100, B01111000, // m

```

4, 8, B01111100, B00000100, B00000100, B01111000, B00000000, // n
4, 8, B00111000, B01000100, B01000100, B00111000, B00000000, // o
4, 8, B11111100, B00100100, B00100100, B00011000, B00000000, // p
4, 8, B00011000, B00100100, B00100100, B11111100, B00000000, // q
4, 8, B01111100, B00001000, B00000100, B00000100, B00000000, // r
4, 8, B01001000, B01010100, B01010100, B00100100, B00000000, // s
3, 8, B00000100, B00111111, B01000100, B00000000, B00000000, // t
4, 8, B00111100, B01000000, B01000000, B01111100, B00000000, // u
5, 8, B00011100, B00100000, B01000000, B00100000, B00011100, // v
5, 8, B00111100, B01000000, B00111100, B01000000, B00111100, // w
5, 8, B01000100, B00101000, B00010000, B00101000, B01000100, // x
4, 8, B10011100, B10100000, B10100000, B01111100, B00000000, // y
3, 8, B01100100, B01010100, B01001100, B00000000, B00000000, // z
3, 8, B00001000, B00110110, B01000001, B00000000, B00000000, // {
1, 8, B01111111, B00000000, B00000000, B00000000, B00000000, // |
3, 8, B01000001, B00110110, B00001000, B00000000, B00000000, // }
4, 8, B00001000, B00000100, B00001000, B00000100, B00000000, // ~
};

int data = 11; // 8, DIN pin of MAX7219 module
int load = 10; // 9, CS pin of MAX7219 module
int clock = 13; // 10, CLK pin of MAX7219 module
int maxInUse = 3; //จำนวนของ max7219
MaxMatrix m(data, load, clock, maxInUse); // define module
byte buffer[10];
char string1[] = " Please Enter text "; //ประโยคเริ่มต้น
void setup(){
    m.init(); // module initialize
    m.setIntensity(0); // ความเข้มของ dotmatrix 0-15
    Serial.begin(9600); // เริ่มต้นการสื่อสารแบบอนุกรม
}
void loop(){
    byte c;
    // โค้ดสำหรับการรับข้อมูลจากแอปพลิเคชัน

```

```

while (Serial.available() > 0){
    byte c = Serial.read();
    Serial.println(c, DEC);
    printCharWithShift(c, 100);
}
delay(100);
m.shiftLeft(false, true);
printStringWithShift(string1, 100);
}

void printCharWithShift(char c, int shift_speed){
    if (c < 32) return;
    c -= 32;
    memcpy_P(buffer, CH + 7*c, 7);
    m.writeSprite(maxInUse*8, 0, buffer);
    m.setColumn(maxInUse*8 + buffer[0], 0);
    for (int i=0; i<buffer[0]+1; i++)
    {
        delay(shift_speed);
        m.shiftLeft(false, false);
    }
}

void printStringWithShift(char* s, int shift_speed){
    while (*s != 0){
        printCharWithShift(*s, shift_speed);
        s++;
    }
}

void printString(char* s)
{
    int col = 0;
    while (*s != 0)
    {
        if (*s < 32) continue;

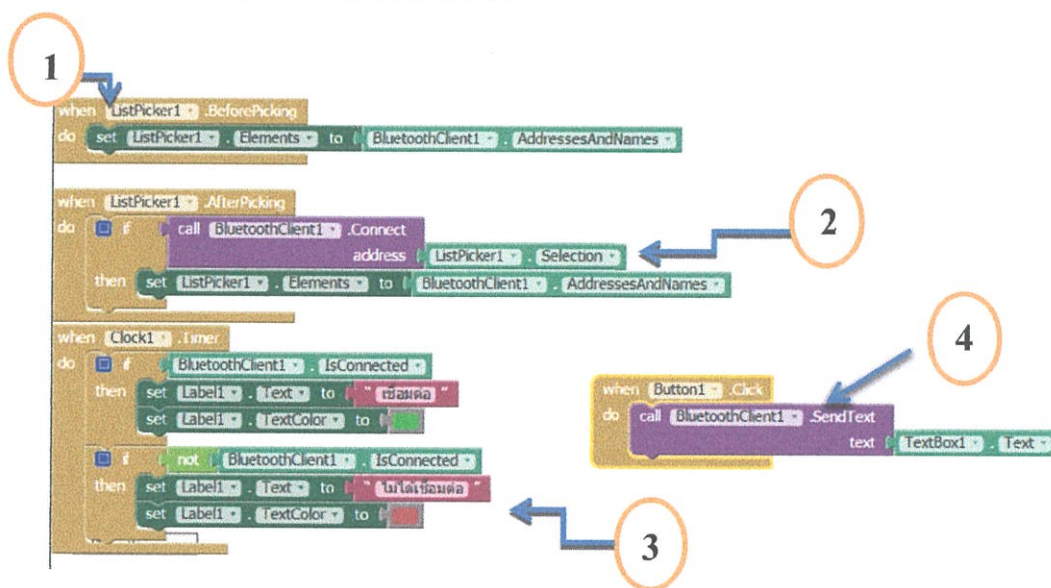
```

```

char c = *s - 32;
memcpy_P(buffer, CH + 7*c, 7);
m.writeSprite(col, 0, buffer);
m.setColumn(col + buffer[0], 0);
col += buffer[0] + 1;
s++;
}
}

```

3.3 หลักการทำงานของแอปพลิเคชัน



รูปที่ 3.6 ทำงานของแอปพลิเคชัน

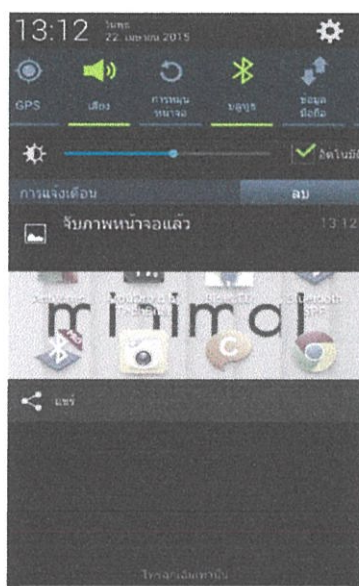
1. ก่อนกดปุ่มเชื่อมต่อทำการตรวจสอบแอดเดรสและชื่อของบลูทูธก่อน
2. หลังจากกดปุ่มเชื่อมต่อให้เรียกหน้าต่างรายชื่อและแอดเดรสของบลูทูธ
3. ถ้ากำลังเชื่อมต่อบลูทูธอยู่ ให้แถบแสดงสถานะแสดงคำว่า **เชื่อมต่อ** ด้วยตัวอักษรสีเขียว ถ้าไม่ ให้แสดงคำว่า **ไม่เชื่อมต่อ** ด้วยตัวอักษรสีแดง
4. เมื่อกดปุ่ม ส่งข้อความ ให้ส่งข้อความใน textbox

บทที่ 4

การทดลองและผลการทดลอง

4.1 ขั้นตอนการใช้งานแอปพลิเคชัน

4.1.1 เปิดใช้งานบลูทูธในสมาร์ทโฟน



รูปที่ 4.1 หน้าจอการเปิดบลูทูธ

4.1.2 เข้าแอปพลิเคชันส่งข้อมูล (DTS)



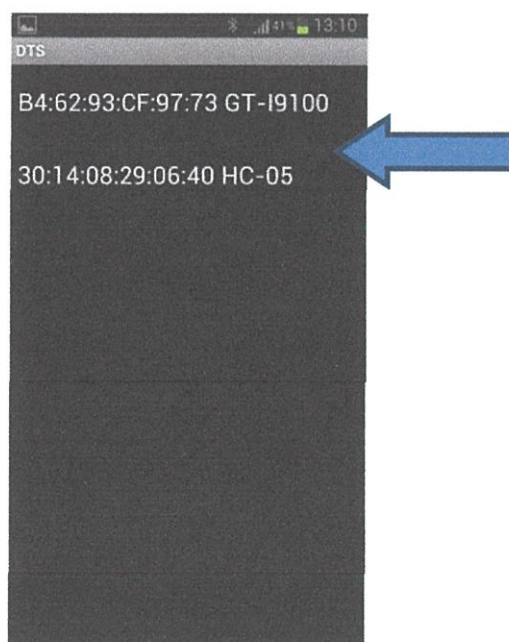
รูปที่ 4.2 หน้าจอไอคอนแอปพลิเคชัน

4.1.3 หน้าจอแอปพลิเคชันจากนั้น กดปุ่มเชื่อมต่อเพื่อเชื่อมต่อบลูทูธ



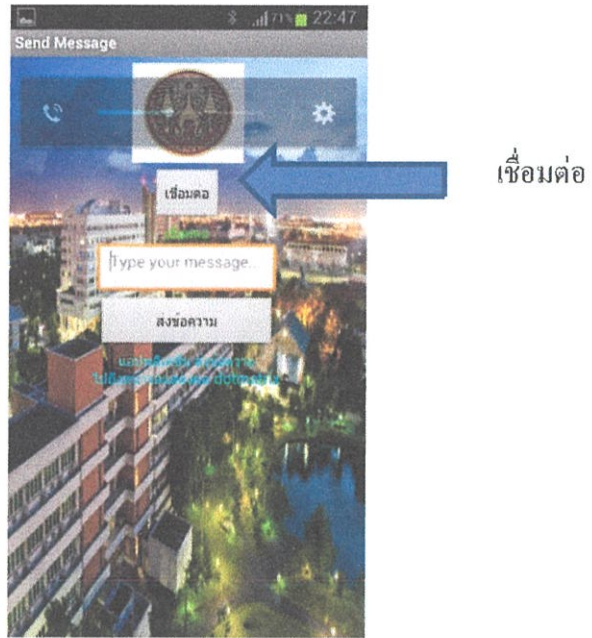
รูปที่ 4.3 ปุ่มเชื่อมต่อหน้าจอแอปพลิเคชัน

4.1.4 แอปพลิเคชันจะโชว์ชื่อบลูทูธ เลือกบลูทูธของหน้าจอแสดงผลดอทเมตริกซ์ (HC-05)



รูปที่ 4.4 ปุ่มเชื่อมต่อหน้าจอแอปพลิเคชัน

4.1.5 แทบแสดงสถานะเปลี่ยนเป็นสถานะ



รูปที่ 4.5 แทบแสดงสถานะแอปพลิเคชัน

4.1.6 พิมพ์ข้อความที่ต้องการให้โชว์บนหน้าจอแสดงผลในช่องพิมพ์ตัวอักษร แล้วกดปุ่มส่งข้อความ



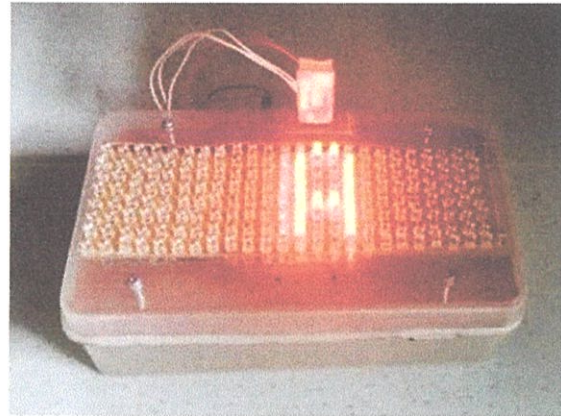
รูปที่ 4.6 การพิมพ์ข้อความที่จะส่ง

4.2 การส่งงานงานแอปพลิเคชันไปยัง LED Dot Matrix

4.2.1 ส่งตัวอักษร A จากแอปพลิเคชันไปยัง LED Dot Matrix

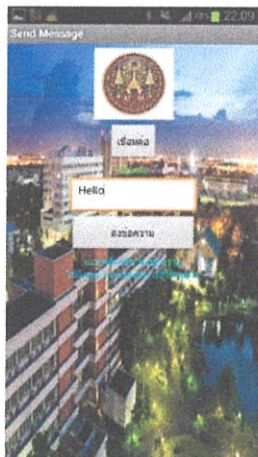


รูปที่ 4.7 ตัวอย่างการส่งตัวอักษร

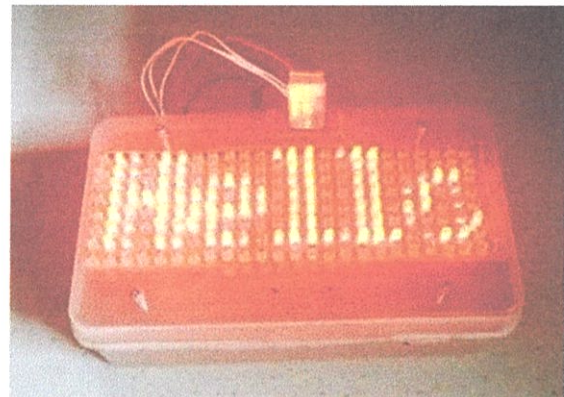


รูปที่ 4.8 แสดงผลการส่งตัวอักษร

4.2.2 ส่งตัวอักษร Hello จากแอปพลิเคชันไปยัง LED Dot Matrix



รูปที่ 4.9 ตัวอย่างการส่งประโยค

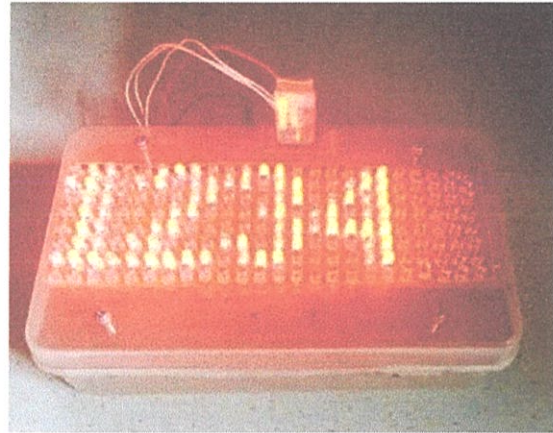


รูปที่ 4.10 แสดงผลส่งประโยค

4.2.1 ส่งตัวเลข 1234 จากแอปพลิเคชันไปยัง LED Dot Matrix



รูปที่ 4.11 ตัวอย่างการส่งตัวเลข

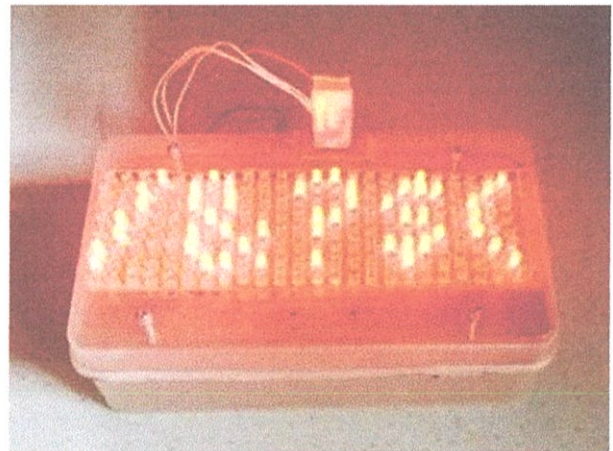


รูปที่ 4.12 แสดงผลการส่งตัวเลข

4.2.4 ส่งตัวสัญลักษณ์ /&?\$(จากแอปพลิเคชันไปยัง LED Dot Matrix

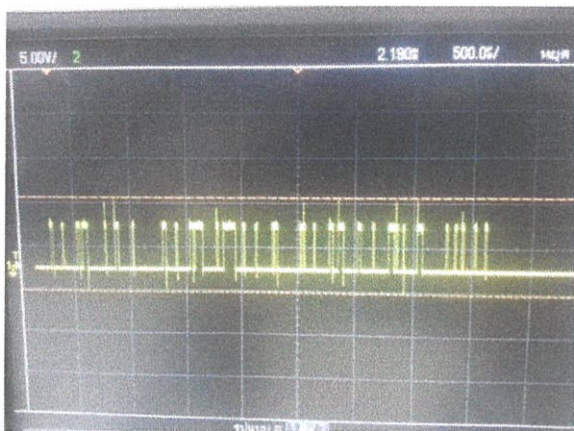


รูปที่ 4.13 ตัวอย่างการส่งตัวสัญลักษณ์



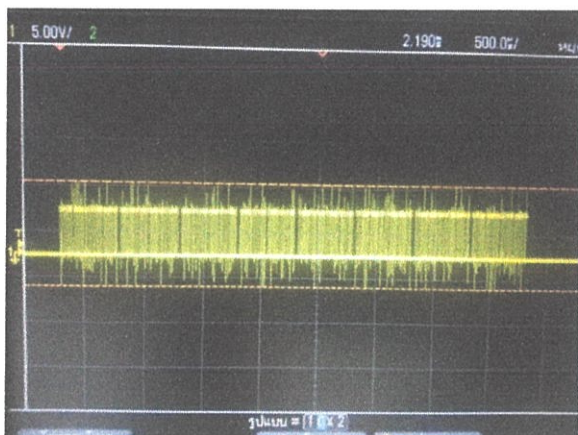
รูปที่ 4.14 แสดงผลการส่งตัวสัญลักษณ์

4.3 สัญญาณพัลส์ของ วงจร LED Dot Matrix



DIN
5 Volt/Div
Time/Div 500 us

รูปที่ 4.15 พัลส์ขา DIN ของ MAX7219



CLK
5 Volt/Div
Time/Div 500 us

รูปที่ 4.16 พัลส์ขา CLK ของ MAX7219



LOAD
5 Volt/Div
Time/Div 500 us

รูปที่ 4.17 พัลส์ขา LOAD ของ MAX7219

4.4 ตารางระยะทางที่ปลูทุอสามารถทำงานได้

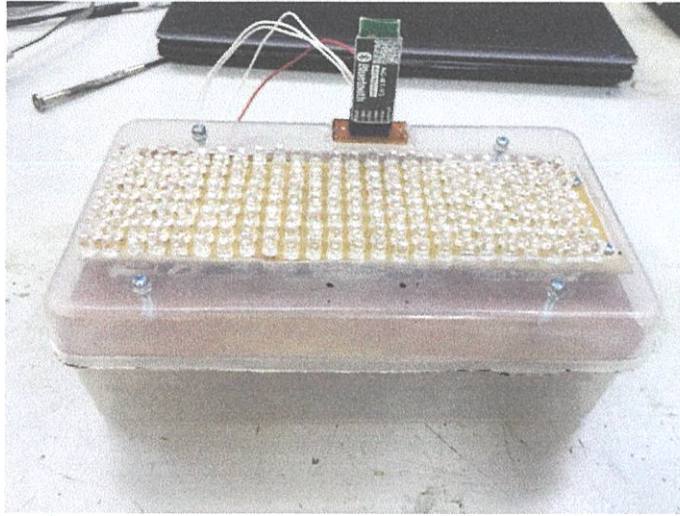
ระยะทาง(เมตร)	ไม่มีสิ่งกีดขวาง	มีสิ่งกีดขวาง
5	ทำงานได้	ทำงานได้
10	ทำงานได้	ทำงานได้
15	ทำงานได้	ทำงานได้
20	ทำงานได้	ทำงานไม่ได้
25	ทำงานได้	ทำงานไม่ได้
30	ทำงานได้	ทำงานไม่ได้
35	ทำงานได้	ทำงานไม่ได้
40	ทำงานไม่ได้	ทำงานไม่ได้

บทที่ 5

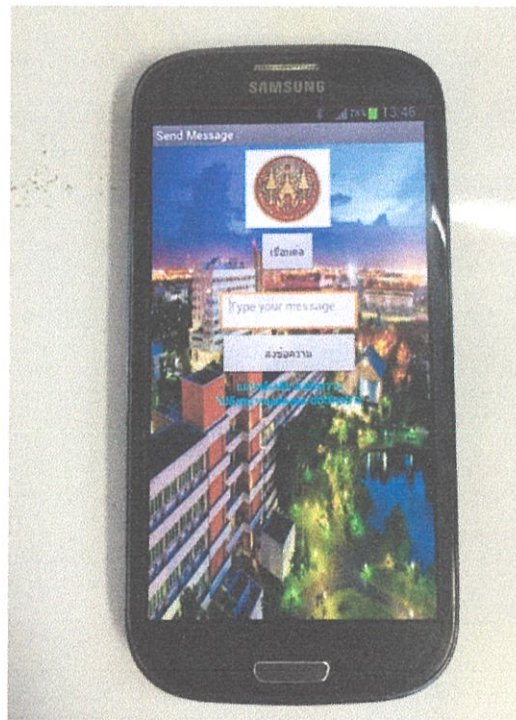
สรุปผลการทดลอง

จากจุดประสงค์ในการทำโครงการเพื่อศึกษาการส่งข้อมูลไร้สายผ่านบลูทูธเพื่อนการประหยัดค่าใช้จ่ายของสายไฟได้และสามารถนำไปใช้ในที่ไม่สามารถเดินสายไฟได้ โดยเราได้ใช้ แอปพลิเคชันในระบบ แอนดรอยด์เป็นตัวส่งการ และ ใช้โมดูลบลูทูธ hc-05 เป็นตัวรับข้อมูลและใช้หน้าจอแสดงผลตอทเมตริกซ์ เป็นตัวแสดงผล จากการทำงานเราใช้การส่งข้อมูลแบบSPI (Serial Peripheral Interface) ระหว่าง สมาร์ทโฟน กับ โมดูลบลูทูธ hc-05 โดยใช้คลื่นความถี่ 2.4G (Bluetooth) และ ระหว่าง Arduino กับ MAX7219CNG โดย กำหนด baud rate ที่ 9600 และเรายังใช้การส่งข้อมูลแบบ Shift Bit เพื่อลดจำนวน PIN ในการใช้งาน วิธีการก็คือการส่งข้อมูลเข้าไปทีละ 1 บิต จนครบแล้วค่อยให้ไปเรียงออกที่ขาทั้ง 16 ของ LED Matrix พร้อมๆกัน ซึ่งจะได้การควบคุม 1 Row การทำงานแบบนี้เราต้องใช้ IC ที่เรียกว่า Shift Register จากผลการทดลองเราสามารถส่งข้อมูลจากสมาร์โฟน ไปยังหน้าจอแสดงผล dotmatrix สามารถส่งในที่ไม่มีสิ่งกีดขวางได้ในระยะทาง 0-35 เมตร และสามารถส่งข้อมูลในที่ที่มีสิ่งกีดขวางได้ระยะทาง 0-15 เมตร จากการทำทดลองนี้ นำการส่งข้อมูลแบบไร้สายผ่านสัญญาณบลูทูธมาใช้แทนสายไฟ เราสามารถลดการใช้สายไฟได้ถึง 0-35 เมตร

ภาคผนวก



รูปที่ 5.1 ระบบควบคุมการส่งข้อมูลไร้สายด้วยความถี่ 2.4 GHz



รูปที่ 5.2 หน้าจอแอปพลิเคชัน DTS

หนังสืออ้างอิง

- [1] เอกชัย มะการ. เรียนรู้ เข้าใจ ใช้งานไมโครคอนโทรลเลอร์ตระกูลAVR ด้วย Arduino. กรุงเทพฯ. พ.ศ.2552
- [2] <http://www.atmel.com/devices/atmega328.aspx>
- [3] <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html>
- [4] http://cpre.kmutnb.ac.th/esl/learning/index.php?article=max7219_7seg_4digit
- [5] <http://www.maximintegrated.com/en/products/power/display-power-control/MAX7219.html>