



เซนเซอร์แจ้งเตือนระยะปลอดภัยสำหรับสายตาในการทำงานกับคอมพิวเตอร์

โดย

นาย สานิต	ยาหอม	B5613755
นางสาวนารีรัตน์	พลจันทัก	B5613830
นางสาวนุชจรี	เล็งศรี	B5618910

รายงานเล่มนี้เป็นส่วนหนึ่งของการศึกษาวิชา 527499 โครงการวิศวกรรมโทรคมนาคม
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง 2557
สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี
ประจำปีการศึกษาที่ 1 ปีการศึกษา 2559

เซนเซอร์แจ้งเตือนระยะปลอดภัยสำหรับสายตาในการทำงานกับคอมพิวเตอร์

คณะกรรมการสอบโครงการ

(รองศาสตราจารย์ ดร. มนต์ทิพย์ภา อุฑารสกุล)
อาจารย์ที่ปรึกษาโครงการ

(รองศาสตราจารย์ ดร. พีระพงษ์ อุฑารสกุล)
กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. ปิยาภรณ์ มีสวัสดิ์)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงการฉบับนี้ เป็นส่วนหนึ่งของ
วิชาการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม รายวิชา 527499 โครงการ
วิศวกรรมโทรคมนาคม ประจำปีการศึกษา 2559

อาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร.มนต์ทิพย์ภา อูหารสกุล
สาขาวิชา วิศวกรรมโทรคมนาคม
ภาคการศึกษาที่ 1/2559

ปัจจุบันการใช้งานทางด้านคอมพิวเตอร์พกพาอย่างแพร่หลายและเป็นสิ่งจำเป็นสำหรับผู้คนจำนวนมาก ซึ่งการใช้งานจากคอมพิวเตอร์นั้นก็ได้มีเพียงข้อดีสำหรับผู้ใช้อย่างเดียว แต่ยังคงส่งผลเสียในด้านสุขภาพตาในระยะยาว ดังนั้นคณะผู้จัดทำจึงเล็งเห็นความสำคัญของเรื่องสุขภาพตา โครงการนี้จึงได้นำเสนอการออกแบบอุปกรณ์เพื่อลดอาการสายตา โดยอาศัยการทำงานของคลื่นอัลตราโซนิกเป็นตัวช่วยในการคำนวณระยะทางแล้วนำค่ามาประมวลผล เพื่อแจ้งเตือนระยะการใช้งานที่เหมาะสมสำหรับการใช้งานคอมพิวเตอร์ เพื่อลดความเสี่ยงของสุขภาพตา

กิตติกรรมประกาศ

คุณความดีอันใดที่เกิดจากโครงการนี้ ขอมอบแด่บิดา มารดาของข้าพเจ้า ผู้ซึ่งคอยห่วงใยดูแล ให้โอกาส กำลังใจ และให้ความสนับสนุนทางการศึกษาแก่ข้าพเจ้ามาโดยตลอด

โครงการนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาจากอาจารย์ที่ปรึกษาโครงการ รองศาสตราจารย์ ดร. มนต์ทิพย์ภา อุฑารสกุล ที่ช่วยให้ความช่วยเหลือในด้านแนวคิดของโครงการ การดูแลเอาใจใส่ในการติดตามความคืบหน้าของโครงการ รวมทั้งได้ชี้แนะข้อบกพร่องของโครงการที่ข้าพเจ้ามองข้าม ตลอดจนฝึกฝนและสนับสนุนให้ข้าพเจ้ามีความสามารถในการทำโครงการสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบคุณนายสุวรรณ เอกรัมย์ ที่ช่วยเป็นที่ปรึกษาด้านการเขียนโปรแกรม



นายสาธิต

ยาหอม

นางสาวนริรัตน์

พลจันทิก

นางสาวนุชจิ

เล็งศรี

สารบัญ

หน้า

บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูป	ณ
สารบัญตาราง	ช

บทที่ 1 บทนำ

1.1 ความเป็นมา	1
1.2 หลักการและเหตุผล	1
1.3 วัตถุประสงค์	2
1.4 ขอบเขตการทำงาน	2
1.5 ขั้นตอนการดำเนินงาน	2
1.6 ผลที่คาดว่าจะได้รับ	3

บทที่ 2 ทฤษฎีและหลักการ

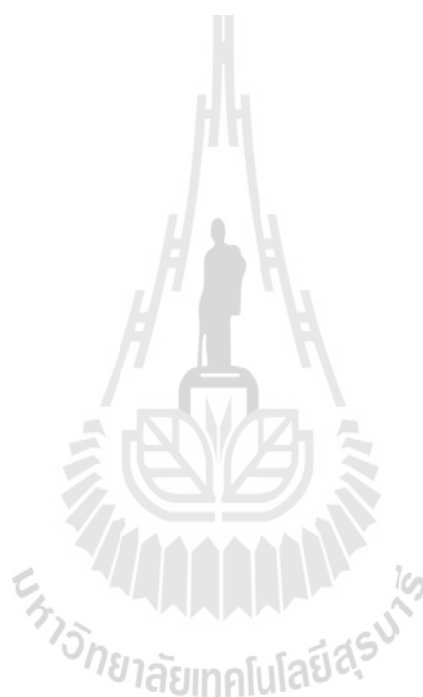
2.1 กล่าวนำ	4
2.2 สายตากับการใช้งานคอมพิวเตอร์พกพา	4
2.2.1 ลักษณะการใช้งานที่เหมาะสม	5
2.2.2 ผลกระทบที่อาจส่งผลต่อผู้ใช้งาน	8
2.3 ไมโครคอนโทรลเลอร์	11
2.3.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล AVR	11
2.3.2 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล AVR	12
2.3.3 การจัดการขาของไมโครคอนโทรลเลอร์ตระกูล AVR	13
2.4 เซนเซอร์วัดระยะทาง รุ่น US-015	15
2.4.1 คลื่นอัลตราโซนิก	16
2.4.2 การต่อใช้งาน US-015	18
2.4.3 หลักการทำงานของอัลตราโซนิกเซนเซอร์	18

สารบัญ (ต่อ)

	หน้า
2.5 ตัวเลขแสดงผล 7-Segment	22
2.5.1 โครงสร้างภายในของ 7-Segment	22
2.6 ลำโพงแจ้งเตือนและไฟแจ้งเตือน	26
2.6.1 คุณสมบัติของลำโพงแจ้งเตือน	26
2.6.2 การต่อขาเข้ากับ Arduino	27
2.6.3 หลอด LED	27
2.7 สรุปท้ายบท	29
บทที่ 3 การออกแบบระบบต้นแบบ	
3.1 กล่าวนำ	30
3.2 ส่วนประกอบของอุปกรณ์ต้นแบบ	30
3.3 หลักการทำงานของอุปกรณ์ต้นแบบ	34
3.3.1 แผนผังการทำงาน	35
3.4 การใช้โปรแกรม Arduino IDE	36
3.4.1 วิธีการดาวน์โหลด Arduino IDE	36
3.4.2 การอัปโหลดไฟล์ลงบอร์ด Arduino MEGA2560 R3	42
3.5 การสร้างอุปกรณ์ต้นแบบ	46
3.6 สรุปท้ายบท	49
บทที่ 4 การทดลองอุปกรณ์ต้นแบบ	
4.1 กล่าวนำ	50
4.2 การทดลองอุปกรณ์ต้นแบบ	50
4.3 สรุปผลการทดลอง	63

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปและข้อเสนอแนะ	
5.1 กล่าวนำ	65
5.2 ปัญหาที่พบบ่อยระหว่างการทำโครงการ	65
5.3 แนวทางการพัฒนาต่อไป	65
5.4 ประโยชน์ที่ได้จากโครงการ	65
บรรณานุกรม	66
ภาคผนวก	67
ประวัติผู้จัดทำ	90



สารบัญรูป

	หน้า
รูปที่ 2.1 การตั้ง Desktops	6
รูปที่ 2.2 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ตระกูล	12
รูปที่ 2.3 การจัดขาของ	13
รูปที่ 2.4 หลักการตรวจจับวัตถุโดยใช้สัญญาณความถี่เหนือเสียง	15
รูปที่ 2.5 เซนเซอร์วัดระยะทาง รุ่น US-015	16
รูปที่ 2.6 สัญลักษณ์ของอัลตราโซนิก	17
รูปที่ 2.7 ไดอะแกรมภายในอัลตราโซนิกเซนเซอร์	19
รูปที่ 2.8 การเว้นระยะห่างของเซนเซอร์	21
รูปที่ 2.9 การจัดวางแนวระนาบของสิ่งของ	21
รูปที่ 2.10 การตรวจจับวัตถุที่มีรูปร่างไม่แน่นอน	22
รูปที่ 2.11 ตำแหน่ง a – g ภายในและสัญลักษณ์ 7-Segment	23
รูปที่ 2.12 การต่อโครงสร้างภายในแบบคอมมอนแอนอด(A)	23
รูปที่ 2.13 การต่อโครงสร้างภายในแบบคอมมอนคาโทด(K)	24
รูปที่ 2.14 Buzzer แบบ diaphragm –Peizo	26
รูปที่ 2.15 การต่อขาเข้ากับ Arduino	27
รูปที่ 2.16 ส่วนประกอบของ LED	28
รูปที่ 2.17 สัญลักษณ์ของ LED ในวงจรไฟฟ้า	28
รูปที่ 3.1 ส่วนประกอบของอุปกรณ์ต้นแบบ	30
รูปที่ 3.2 การเชื่อมต่ออุปกรณ์ต้นแบบ	31
รูปที่ 3.3 จำลองการใช้งานของอุปกรณ์ต้นแบบ	34
รูปที่ 3.4 โฟลว์ชาร์ตการทำงานของโปรแกรม	35
รูปที่ 3.5 หน้าเว็บสำหรับดาวน์โหลด	37
รูปที่ 3.6 การเลือก Arduinoซอฟต์แวร์	37
รูปที่ 3.7 การเชื่อมต่อของบอร์ดไม่สำเร็จ	38
รูปที่ 3.8 การเลือก Update Driver Software	38
รูปที่ 3.9 การเลือกติดตั้งซอฟต์แวร์	39
รูปที่ 3.10 การเลือกตำแหน่งจัดเก็บไฟล์โปรแกรม	39

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.11 การเลือกโฟลเตอร์ไดรเวอร์	40
รูปที่ 3.12 การเลือก Install this driver software anyway	40
รูปที่ 3.13 การ Install Driver สำเร็จ	41
รูปที่ 3.14 การ set up Arduino Software	41
รูปที่ 3.15 การเลือกรุ่นของบอร์ด Arduino	42
รูปที่ 3.16 เปิดโปรแกรม Arduino	42
รูปที่ 3.17 code สำหรับอุปกรณ์ต้นแบบ	43
รูปที่ 3.18 การเลือก Port COM	43
รูปที่ 3.19 การเลือกประเภทของบอร์ด Arduino	44
รูปที่ 3.20 การเลือก Processor	44
รูปที่ 3.21 การตรวจสอบ Code	45
รูปที่ 3.22 อัปโหลดโปรแกรมลงบอร์ด	45
รูปที่ 3.23 ลายวงจร	46
รูปที่ 3.24 กัดลายวงจร	47
รูปที่ 3.25 บัดกรีอุปกรณ์ลงแผ่นปริ้น	47
รูปที่ 3.26 การเชื่อมสายระหว่างบอร์ดและลายวงจร	48
รูปที่ 3.27 ติดตั้งอุปกรณ์ลงกล่องอะคริลิก	48
รูปที่ 3.28 อุปกรณ์ต้นแบบที่สมบูรณ์	48
รูปที่ 4.1 ทดสอบวัฏระยะของเซนเซอร์อัลตราโซนิก	51
รูปที่ 4.2 การทดสอบการแจ้งเตือนของอุปกรณ์	60

สารบัญตาราง

	หน้า
ตารางที่ 2.1 รหัสการแสดงผลสำหรับ 7-Segment Common Anode เป็นตัวเลข 0-9	25
ตารางที่ 2.2 รหัสการแสดงผลสำหรับ 7-Segment Common Cathode เป็นตัวเลข 0-9	25
ตารางที่ 2.3 คุณสมบัติของบัชเชอร์	26
ตารางที่ 4.1 ผลการทดลองการวัดระยะทางของเซนเซอร์อัลตราโซนิก	52
ตารางที่ 4.2 ทดสอบวัดระยะผู้ใช้งานคนที่ 1	55
ตารางที่ 4.2 ทดสอบวัดระยะผู้ใช้งานคนที่ 2	56
ตารางที่ 4.2 ทดสอบวัดระยะผู้ใช้งานคนที่ 3	57
ตารางที่ 4.5 ช่วงที่ 1 (ความเร็วเสียง 1)	61
ตารางที่ 4.6 ช่วงที่ 2 (ความเร็วเสียง 2)	61
ตารางที่ 4.7 ช่วงที่ 3 (ความเร็วเสียง 3)	61
ตารางที่ 4.8 ช่วงที่ 4 (ความเร็วเสียง 4)	62
ตารางที่ 4.9 ช่วงที่ 5 (ความเร็วเสียง 5)	62



บทที่ 1

บทนำ

1.1 ความเป็นมา

เนื่องจากในปัจจุบันมีการใช้คอมพิวเตอร์พกพาเป็นหนึ่งในเทคโนโลยีที่ใช้กันอย่างแพร่หลายตั้งแต่วัยเด็ก วัยรุ่น วัยผู้ใหญ่ จนถึงวัยชรา วัตถุประสงค์ของการใช้งานก็แตกต่างกันไป ไม่ว่าจะเป็นการใช้งานเพื่อความบันเทิง ใช้งานในการทำงานหาข้อมูลต่างๆ หรือในอาชีพของแต่ละบุคคล ซึ่งระยะเวลาการใช้งาน ความสว่างของพื้นที่ที่ใช้งาน ลักษณะท่าทาง รวมไปถึงการอยู่ใกล้จอเกินไปนั้นย่อมส่งผลเสียต่อสายตาของผู้ใช้งานนอกจากนี้แล้ว จอภาพคอมพิวเตอร์สามารถแผ่รังสีได้เช่นเดียวกับอุปกรณ์ไฟฟ้าทั่วไปในรูปของคลื่นแม่เหล็กและคลื่นไฟฟ้า ได้แก่ ไฟฟ้าสถิต คลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่ำยิ่งยวด (Extremely Low Frequency :ELF) และคลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่ำมาก (Very Low Frequency : VLF) และนอกจากนี้ ยังมีรังสีไมโครเวฟ รังสีเหนือม่วง ซึ่งอาจจะมีอันตราย เนื่องจากลักษณะการใช้งานดังกล่าวนี้ อาจส่งผลทำให้เกิดกล้ามเนื้อตาอ่อนล้า ตาพร่ามัว เคืองตา แสบตา ตาแห้ง ในระยะยาวนั้นอาจส่งผลให้เกิดโรคทางสายตาไม่ว่าจะเป็น ต้อหินเรื้อรัง ม่านตาอักเสบ เยื่อบุตาอักเสบเรื้อรัง หรืออาจตาบอดได้ในที่สุด โครงการนี้จึงเล็งเห็นปัญหาในด้านผลเสียทางสายตาต่อผู้ใช้งานซึ่งผู้ใช้งานนั้นไม่สามารถทราบถึงระยะที่เข้าใกล้จอภาพเกินไปได้ ทั้งนี้ซึ่งเป็นระยะที่มีความเสี่ยงต่อสายตาเราและจากการศึกษาระดับรังสีคลื่นแม่เหล็กไฟฟ้าที่แผ่ออกมาจากจอภาพคอมพิวเตอร์จะมีปริมาณลดลงตามระยะทางที่ห่างออกไป และควรอยู่ในระยะการมองระหว่าง 50-70 เซนติเมตร เพื่อให้รังสีที่แผ่ออกมาจากจอคอมพิวเตอร์เป็นอันตรายต่อสุขภาพน้อยลง โครงการนี้จึงเป็นแนวทางหนึ่งเท่านั้นที่จะช่วยในเรื่องการถนอมสายตาของเราต่อการใช้งานคอมพิวเตอร์พกพา

1.2 หลักการและเหตุผล

ในปัจจุบันนี้มีเทคโนโลยีเข้ามาในชีวิตประจำวันมากมายทั้งความบันเทิงจนถึงการทำงานในแต่ละวัน อาทิเช่น การใช้คอมพิวเตอร์ทำงานอยู่หน้าจอคอมพิวเตอร์ติดต่อกันเป็นเวลานานๆ เช่น เกินสองชั่วโมงและใกล้จอมากเกินไป มักจะมีอาการปวดตา แสบตา ตามัว และบ่อยครั้งที่จะมีอาการปวดศีรษะปวดเมื่อย ปวดหลัง ปวดไหล่ ปวดต้นคอร่วมด้วยอาการเหล่านี้เกิดขึ้นโดยเราไม่ทันสังเกตหรือระวังในระยะยาวอาจส่งผลให้เรา เกิดโรคทางสายตาตามมาไม่ว่าจะเป็น ต้อหินเรื้อรัง ม่านตาอักเสบ เยื่อบุตาอักเสบเรื้อรังหรืออาจตาบอดได้ในที่สุด

เมื่อคอมพิวเตอร์ยังเป็นสิ่งจำเป็นต่อการใช้งานเพื่อประโยชน์ต่างๆอยู่ จึงได้คิดวิธีการเพื่อลดความเสี่ยงของผลเสียที่จะตามโดยใช้หลักการการทำงานของบอร์ด Arduino MEGA2560 R3 ร่วมกับเซนเซอร์อัลตราโซนิกโดยใช้คลื่นเสียงความถี่ต่ำ 40 kHz ส่งคลื่นไปกระทบผู้ใช้งานแล้วสะท้อนกลับมาเพื่อคำนวณระยะห่างหากอยู่ในระยะที่ไม่ปลอดภัยจะมีการแจ้งเตือนพร้อมสัญญาณไฟและค่าระยะห่างจะแสดงที่ 7-Segment

โครงการนี้จะช่วยแจ้งเตือนให้ผู้ใช้งานไม่ให้เข้าไปใกล้จนเกินไป เพื่อลดความเสี่ยงของสายตาและโรคทางสายตาที่จะตามมาในระยะยาว

1.3 วัตถุประสงค์

เพื่อสร้างอุปกรณ์ต้นแบบเพื่อเป็นทางเลือกการถนอมสายตา

1.4 ขอบเขตการทำงาน

- 1.4.1 ศึกษา ค้นหาข้อมูลเกี่ยวกับทฤษฎีคลื่นอัลตราโซนิกและการทำงานของบอร์ด Arduino MEGA2560 R3
- 1.4.2 เขียนโปรแกรมควบคุมการทำงานของเซนเซอร์อัลตราโซนิก โดยทำงานร่วมกับบอร์ด Arduino MEGA2560 R3
- 1.4.3 ทำการทดสอบค่าระยะที่วัดได้จากเซนเซอร์อัลตราโซนิก
- 1.4.4 ออกแบบวงจรเชื่อมต่อระหว่างเซนเซอร์อัลตราโซนิกและระบบไฟสัญญาณเตือนระบบ 7-segment บอกระยะและระบบ Buzzer กริ่งเสียงแจ้งเตือน
- 1.4.1 สร้างอุปกรณ์ต้นแบบทั้งหมด และทดสอบเพื่อให้ได้ตามวัตถุประสงค์

1.5 ขั้นตอนการดำเนินงาน

- 1.5.1 ศึกษา ค้นหาเกี่ยวกับการทำงานของบอร์ด Arduino MEGA2560 R3 เซนเซอร์อัลตราโซนิก 7-segment ลำโพงแจ้งเตือนและหลอดไฟ LED
- 1.5.2 เขียนโปรแกรมคำสั่งควบคุมการทำงานของเซนเซอร์อัลตราโซนิก (Ultrasonic sensor) เพื่อแจ้งเตือนพร้อมสัญญาณไฟ และแสดงค่าที่ 7-segment
- 1.5.3 สร้างและออกแบบอุปกรณ์ให้สามารถใช้งานได้สะดวกขึ้น
- 1.5.4 ทำการทดลองจริงกับผู้ใช้งาน

1.6 ผลที่คาดว่าจะได้รับ

- 1.6.1 สามารถช่วยทางด้านสุขภาพตาที่ได้รับผลกระทบจากจอคอมพิวเตอร์ได้
- 1.6.2สามารถนำความรู้ที่ได้มาประกอบอาชีพได้
- 1.6.3สามารถทำอุปกรณ์ต้นแบบได้บรรลุตามวัตถุประสงค์
- 1.6.4ได้เรียนรู้การทำงานเป็นทีม การรู้จักรับฟังความคิดเห็นของเพื่อนร่วมงาน



บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

เนื้อหาในบทนี้จะกล่าวถึงทฤษฎีและหลักการของอุปกรณ์ต่างๆที่ใช้อ้างอิงในการทำอุปกรณ์ต้นแบบ โดยเนื้อหาในบทนี้จะนำไปใช้ในการออกแบบและสร้างอุปกรณ์ต้นแบบในบทที่ 3 จะประกอบด้วย หนึ่งสายตากับการใช้งานคอมพิวเตอร์เป็นเรื่องของลักษณะของการใช้งานที่เหมาะสมและผลกระทบที่อาจส่งผลกระทบต่อสายตาในระยะยาว สองไมโครคอนโทรลเลอร์จะเป็นการอธิบายการจัดขาและหน้าที่การทำงานของแต่ละขา สามเซนเซอร์อัลตราโซนิกจะอธิบายหลักการทำงานของเซนเซอร์ สี่ตัวเลขแสดงผล 7-Segment จะมีสองแบบ แบบแรกจะเป็นแบบแอโอค แบบที่สองจะเป็นแบบแคโทด ห้าลำโพงแฉ่งเตือนและเอลอีดีก็จะอธิบายหลักการการทำงาน ซึ่งที่กล่าวมานั้นจะสามารถเชื่อมโยงกับหลักการทำงานของอุปกรณ์ต้นแบบได้

2.2 สายตากับการใช้งานคอมพิวเตอร์

ในปัจจุบันนี้ เทคโนโลยีคอมพิวเตอร์ได้มีการพัฒนาไปอย่างรวดเร็ว มีการนำคอมพิวเตอร์และคอมพิวเตอร์พกพา เข้ามาใช้กันอย่างแพร่หลายในทุกวงการ ซึ่งอาจจะเพราะความสะดวกสบายในการใช้งานต่างๆ อย่างครบวงจรในทุก ๆ ด้าน มีทั้งระบบมัลติมีเดียที่มีความหลากหลาย ซึ่งเป็นไปตามความต้องการของผู้ใช้งานรวมทั้งมีโปรแกรมซอฟต์แวร์ที่หลากหลายในการใช้งานระดับผู้ใช้งานตั้งแต่ระหว่างบุคคลไปจนถึงองค์กรต่างๆ มากมายในปัจจุบัน เมื่อมีการใช้งานอยู่เป็นประจำและในแต่ละครั้งมีการใช้งานเป็นระยะเวลานานๆหลายชั่วโมงติดต่อกัน ย่อมส่งผลกระทบต่อสุขภาพตามมามากทั้งในด้านสายตาและสุขภาพร่างกาย

นอกจากนี้คนที่ใช้สายตาในระยะใกล้ จะมีอาการปวดตา ไม่สบายตา ทั้งนี้เนื่องจากการทำงานที่ใช้สายตาในระยะใกล้ จำเป็นต้องอาศัยการหดเกร็งของกล้ามเนื้อตา เมื่อมีการใช้กล้ามเนื้อเป็นเวลานานก็มีการเมื่อยล้าของกล้ามเนื้อตา เช่นเดียวกับการเมื่อยล้าของกล้ามเนื้อแขนขาเวลาที่เราวางาน ใดๆ การเมื่อยล้าของกล้ามเนื้อตาในเบื้องต้นจะแสดงออกมาในรูปของการปวดตา เมื่อยตา ตาแห้ง ตาแดง เคืองตา มองเห็นภาพไม่ชัดเจนพร่ามัว เห็นสีผิดเพี้ยนไปจากปกติ นอกจากนี้ ในคนปกติจะมีการกะพริบตาเป็นระยะ ๆ เพื่อให้หน้าตากระจายหล่อเลี้ยงทั่วลูกตา แต่ในเวลาที่เราทำงาน เช่น อ่านหนังสือ หรือดูจอคอมพิวเตอร์ อัตราการกะพริบตานั้นจะน้อยลง ทำให้น้ำตาระเหยออกไปมากกว่าปกติ ลูกตามีน้ำหล่อเลี้ยงตาน้อยกว่าปกติทำให้กระจกตาแห้ง เราจึงรู้สึกแสบตา รู้สึกว่าตาแห้งหลังตาได้ลำบาก หรือในรายที่เป็นมาก ๆ อาจเกิดแผลขนาดเล็ก ๆ ที่กระจกตาได้

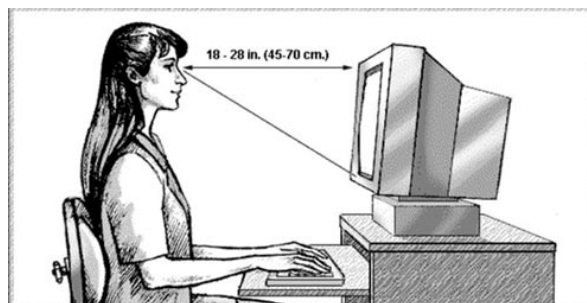
ในระยะยาวย่อมอาจส่งผลต่อการเป็นโรคทางสายตาได้ เช่น ต้อหินเรื้อรัง ม่านตาอักเสบ เยื่อบุตาอักเสบเรื้อรัง รวมไปถึงสมรรถนะทางสายตาเสื่อมประสิทธิภาพไวขึ้นและการใช้สายตาในระยะใกล้จอคอมพิวเตอร์เป็นเวลานานๆ ไม่ได้มีผลต่อเรื่องสายตาเพียงอย่างเดียว แต่ยังมีผลต่อส่วนอื่นๆ ของร่างกายด้วย เช่น กล้ามเนื้อคอ ไหล่ หลัง หากท่าทางที่ใช้นั่งทำงานไม่ถูกต้อง เช่น นั่งหลังงอ เอียงคอ หรือเอียงไหล่ ก็ทำให้มีอาการได้ง่ายขึ้น และมากกว่าที่ควรจะเป็น ทำให้มีอาการปวดคอ ปวดหลัง ปวดไหล่ หรือแม้แต่ปวดศีรษะได้

จอภาพคอมพิวเตอร์สามารถแผ่รังสีได้เช่นเดียวกับอุปกรณ์ไฟฟ้าทั่วไปในรูปแบบของคลื่นแม่เหล็กและคลื่นไฟฟ้า ได้แก่ ไฟฟ้าสถิตย์ คลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่ำยิ่งยวด และคลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่ำมากนอกจากนี้แล้ว ยังมีรังสีไมโครเวฟ รังสีเหนือม่วง ซึ่งอาจจะมีอันตรายต่อสุขภาพ เช่น การมองจอภาพที่มีแสงจ้า หรือแสงสะท้อนเป็นระยะเวลานาน อาจจะทำให้เกิดปัญหาอาการปวดตา สายตาล้า อาการปวดศีรษะ ซึ่งเกิดมาจากการบีบของกล้ามเนื้อตา จากการศึกษาระดับรังสีคลื่นแม่เหล็กไฟฟ้าที่แผ่ออกมาจากจอภาพคอมพิวเตอร์จะมีปริมาณลดลงตามระยะทางที่ห่างออกไป และควรอยู่ในระยะการมองระหว่าง 50-70 เซนติเมตร เพื่อให้รังสีที่แผ่ออกมาจากจอคอมพิวเตอร์ไม่เป็นอันตรายต่อสุขภาพ[1]

2.2.1 ลักษณะการใช้งานที่เหมาะสม[2]

ในการใช้คอมพิวเตอร์ระยะเวลานานๆ อาจส่งผลกระทบต่อร่างกาย ไม่ว่าจะเป็นความเจ็บปวดตามอวัยวะต่างๆ เช่น อาการปวดตา น้ำตาไหลเมื่อใช้คอมพิวเตอร์ในระยะเวลานานๆ อาการปวดหลัง ปวดไหล่ ปวดข้อมือ ปวดนิ้วมือ เป็นต้น ดังนั้นการเอาใจใส่ต่อสภาพแวดล้อมการทำงานให้ดีต่อสุขภาพ จึงเป็นสิ่งที่น่าสนใจและไม่ควรมองข้าม มีคนจำนวนมากในปัจจุบันที่จำเป็นต้องใช้งานคอมพิวเตอร์ ซึ่งหากใช้งานคอมพิวเตอร์ในระยะเวลาไม่นานก็คงไม่มีปัญหาอะไรที่จะกระทบต่อร่างกายมากนัก หากการใช้งานคอมพิวเตอร์ในระยะเวลาไม่นานก็คงไม่มีปัญหาอะไรที่จะกระทบต่อร่างกายมากนัก หากการใช้งานคอมพิวเตอร์ที่มากกว่าหนึ่งชั่วโมงหรือใช้งานต่อเนื่องเป็นระยะเวลานานๆ ก็ควรหันมาสนใจกับการใช้งานอย่างถูกลักษณะใส่ใจกับสภาพแวดล้อมการทำงาน ไม่ว่าจะเป็นโต๊ะคอมพิวเตอร์ เก้าอี้ การวางตำแหน่งอุปกรณ์คอมพิวเตอร์ การนั่ง การใช้คีย์บอร์ดและเมาส์อย่างถูกวิธี เป็นต้นสิ่งเหล่านี้จะทำให้ผู้ใช้งานคอมพิวเตอร์สามารถใช้งานคอมพิวเตอร์ได้ยาวนานขึ้น รวมถึงส่งผลกระทบต่อสุขภาพตามอวัยวะต่างๆ ของร่างกายให้น้อยที่สุดเท่าที่จะเป็นไปได้ และหากใช้งานไปสักระยะหนึ่งและเกิดความเมื่อยล้า ควรพักสายตาหรือยืดเส้นยืดสายด้วยการขยับหรือสลับนิ้วมือ แขน พักคืมน้ำ หรือเดินเล่นสักพักหนึ่งก็จะทำให้รู้สึกดีขึ้น สำหรับแนวทางการใช้งานอย่างถูกสุขลักษณะ สามารถพิจารณาจากตัวอย่างต่อไปนี้

1. การตั้ง Desktops



รูปที่ 2.1 การตั้ง Desktops [2]

- โต๊ะที่วางคอมพิวเตอร์ ควรจะสูง 18-28 นิ้ว ขึ้นกับส่วนสูงของผู้ใช้
- ปรับเก้าอี้ให้แขนขนานกับพื้นขณะทำงานและแขนควรอยู่สูงจากต้นขา 2 นิ้ว
- โต๊ะควรเป็นพื้นที่ว่างเปล่าเพื่อยืดเท้า
- ขงที่ไข้อยควรอยู่ใกล้มือ
- ถ้าใช้ที่หนีบกระดาษควรอยู่ระดับเดียวกับจอภาพ

2. การปรับเก้าอี้

การนั่งนานๆ จะทำให้กระดูกหลัง หมอนรองกระดูกได้รับน้ำหนักอยู่ตลอดเวลา นอกจากหลังแล้วยังมีผลต่อเท้า เนื่องจากเลือดจะไปกองที่เท้า ข้อเสนอแนะนี้จะช่วยให้ทำงานกับคอมพิวเตอร์อย่างสบาย

- อย่าอยู่ในท่าใดท่าหนึ่งนานๆ
- เปลี่ยนท่านั้นและยืนสลับกัน
- ปรับเบาะพิงหลังให้รองรับบริเวณเอวอาจจะใช้หมอนหรือผ้าหุนด้วยก็ได้ ท่านั่งที่ดีควรจะเป็นท่าที่ขาตั้งฉากกับลำตัว
- ปรับความสูงของเก้าอี้เพื่อให้เท้าวางบนพื้น
- นั่งหลังพิงพนักพิง
- ต้นขาขนานกับพื้น เข่าอยู่แนวระนาบเดียวกับข้อสะโพก
- เข่าควรอยู่ห่างจากเบาะ 2-3 นิ้ว
- อย่างนั่งหลังโก่ง
- ปรับความสูงของที่พักแขนให้แขนและไหล่ได้พักขณะทำงาน

3. การจัดแสงสว่าง

การจัดแสงสว่างที่ถูกต้อง จะช่วยให้การทำงานมีประสิทธิภาพ และสะดวกสบาย สภาพของแสงควรเหมาะสมกับงานที่ทำอยู่เป็นประจำ ในกรณีที่ต้องทำงานหน้าเครื่องประจำ สภาพของแสงควรมีลักษณะดังนี้

- ควรใช้ผ้าม่านปิดหน้าต่างๆ เพื่อควบคุมปริมาณแสงในห้อง
- ให้แสงจากหน้าต่างเข้าทางข้าง และจัดวางให้อยู่ระหว่างแถวไฟของเพดาน เพื่อป้องกันการสะท้อนของแสง
- จอไม่ควรได้รับแสงโดยตรงเพราะจะทำให้เกิดการสะท้อนของแสงหรืออาจจะใส่แผ่นกรองแสง

- ขอบบนของจอภาพ ควรอยู่ระดับเดียวกับตา

- ควรทำงานภายใต้แสงไฟทั่วไป และไฟสำหรับทำงานโดยเฉพาะ แสงที่ใช้ไม่ควรเกิน 18-46 แรงเทียน

- แสงควรเข้าทางด้านข้าง เพื่อป้องกันการสะท้อนของแสง
- วางจอภาพโดยหันด้านข้างของจอไปหาหน้าต่าง
- ไม่ควรใช้คอมไฟ
- ผนังด้านหลังจอภาพควรทาสีทึบเพื่อป้องกันสะท้อนของแสง
- ใช้แผ่นกันการสะท้อนของแสง
- ไม่ควรให้มีแสงไปตกกระทบโดยตรง เพื่อป้องกันไม่ให้เกิดจุดสว่างบนจอภาพ

4. การปรับคีย์บอร์ดและเมาส์

การทำงานกับคอมพิวเตอร์นานจะมีปัญหาเกี่ยวกับแขน ข้อมือ และมือ โดยมีคำแนะนำเพื่อการทำงานที่สะดวกสบายขึ้น

- ปรับความสูงของคีย์บอร์ด เพื่อให้หัวไหล่ได้พัก แขนแนบลำตัว คีย์บอร์ดอยู่ตรงหน้าไม่เอียงซ้ายหรือขวา
- คีย์บอร์ดควรอยู่ใกล้กับผู้ใช้เพื่อที่จะไม่ต้องเอื้อมมือ
- ข้อศอกควรได้ฉาก 90 องศา แขนส่วนปลายจะขนานกับพื้น
- เมาส์ควรวางข้างคีย์บอร์ด
- ขณะที่ไม่ได้ทำงานควรพักแขนไว้บนท้องไม่ควรพักแขนไว้บนคีย์บอร์ด หรือเมาส์

5. การวาง และเคลื่อนไหวมือ แขน

ขณะที่ใช้เครื่องคอมพิวเตอร์ทำงาน การเคลื่อนไหวมือและแขนอย่างสบายๆ เป็นธรรมชาติ จะช่วยป้องกันการเมื่อยล้าของมือ และแขน และควรปฏิบัติตามคำแนะนำในขณะที่ใช้เครื่องคอมพิวเตอร์

- เมื่อพิมพ์ข้อความควรให้ปลายแขนและข้อมืออยู่ในระดับเดียวกันและขนานกับพื้น
- วางข้อศอกในท่าที่สบายๆ ข้างลำตัว อาจแนบข้อศอกชิดลำตัว แต่ระวังอย่าชิดมากเกินไปจนข้อศอกกดไปที่ลำตัว
- ขณะที่แป้นพิมพ์ เมาส์ หรืออุปกรณ์ป้อนข้อมูลอื่นๆ ควรให้ข้อมืออยู่ในท่าตามธรรมชาติ ไม่ควรโค้งงอ หรือหักข้อมือ
- หากมีที่พักมือ ควรใช้เพื่อการพักมือเท่านั้น ไม่ควรวางมือบนที่พัก หรือบนโต๊ะขณะกำลังพิมพ์
- ปรับให้นิ้วหัวแม่มือและนิ้วอื่นๆ อยู่ในท่าที่ผ่อนคลายเป็นธรรมชาติ
- ใช้แรงให้น้อยที่สุดในการกดแป้นพิมพ์ หลีกเลี่ยงการกระแทกนิ้วลงบนแป้นพิมพ์แรงๆ จับเมาส์หลวมๆ อย่าจับแน่นเกินไป
- ใช้แรงจากแขนในการเลื่อนเมาส์

2.2.2 ผลกระทบที่อาจส่งผลต่อผู้ใช้งาน[3]

ภัยที่เกิดจากเทคโนโลยีสารสนเทศนั้น ไม่ได้มีเพียงแค่ภัยคุกคามหรือภัยแฝงจากสื่ออินเทอร์เน็ตเท่านั้น แต่ผลกระทบโดยตรงที่ผู้ใช้งานเทคโนโลยีสารสนเทศได้รับคือผลกระทบต่อสุขภาพ ทั้งกาย ใจ และผลกระทบต่อสังคมรอบข้างซึ่งโดยส่วนใหญ่แล้ว ผู้ใช้เทคโนโลยีสารสนเทศเองนั้นมักไม่รู้ตัวจนกว่าจะเกิดปัญหาเหล่านั้นขึ้น ประชาชนผู้ใช้งานคอมพิวเตอร์ อินเทอร์เน็ต รวมทั้งโทรศัพท์มือถือ แท็บเล็ต จึงควรระมัดระวังในการใช้อุปกรณ์เหล่านี้อย่างเหมาะสม ถูกวิธี เพื่อป้องกันการเกิดผลเสียต่อสุขภาพซึ่งอาจร้ายแรงจนเกินกว่าจะรักษาได้ ไม่ว่าจะเป็นผลกระทบที่ส่งผลกระทบต่อร่างกายโดยตรง เช่น ดวงตาหรือระบบประสาท เกิดโรคที่เกิดจากท่านั่ง เช่น กลุ่มอาการปวดข้อต่างๆ หรือโรคที่เกิดจากเชื้อโรคที่มีอยู่ในคอมพิวเตอร์ เช่น โรคภูมิแพ้ เป็นต้น

1. ผลกระทบต่อร่างกายโดยตรง

ดวงตา กล้ามเนื้อและระบบประสาท ทำให้เกิดอาการเมื่อยตา สายตาเสื่อม ปวดกล้ามเนื้อ และปวดศีรษะ คลื่นไส้ เป็นต้น โดยอาการที่พบบ่อยที่สุดของโรคนี้คือ ปวดตา เมื่อยตา ตาแห้ง ถ้าอาการเป็นมากยังอาจก่อให้เกิดปัญหาสายตาเสื่อมลงด้วย เนื่องจากขณะใช้คอมพิวเตอร์ดวงตาต้องจ้องมองหน้าจอที่มีตัวหนังสือหรือ ภาพกระพริบตลอดเวลา ทำให้กลไกตามธรรมชาติของการ

กระทบตาคือน้อยลงจนเราไม่สังเกต เป็นเหตุผลสำคัญที่ทำให้ตาแห้ง และหากดวงตาอยู่ในสภาพที่เหน็ดเหนื่อยหรือตาแห้ง ก็จะทำให้สายตาสีมลง

ระบบประสาท จากคลื่นแม่เหล็กไฟฟ้าของคอมพิวเตอร์ แม้ว่ารังสีชนิดต่างๆจากหน้าจอคอมพิวเตอร์จะมีความปลอดภัยก็ตาม แต่การรับการแผ่รังสีเป็นเวลานานก็อาจจะส่งผลกระทบต่อระบบประสาทของมนุษย์ ได้เช่นกัน จะทำให้เกิดอาการปวดศีรษะ คลื่นไส้ อึดอัด และนอนไม่หลับ เป็นต้น

2. โรคที่เกิดจากท่านั่งหรือการทำงานซ้ำซาก

โรค Cumulative Trauma Disorders (ความ ผิดปกติจากอุบัติเหตุสะสม) อาการของโรคจะค่อยเป็นค่อยไป จะมีอาการปวดคอ ไหล่ ข้อมือ และหลัง ผู้ที่เป็นมากๆ อาจมีอาการอื่นร่วมด้วย เช่น อาการชาที่มือ อาการของโรคพวกนี้แบ่งเป็น 3 ระยะ คือ ระยะแรกเป็นแล้วหายเมื่อได้พัก ระยะสองคือ มีอาการต่อเนื่องถึงกลางคืน และหายเมื่อได้พัก ระยะสามคือ เป็นตลอดเวลาไม่หาย

อาการ Repetitive Strain Injury หรือ RSI สามารถ เป็นได้กับทุกส่วนของร่างกายจากการนั่งทำงานหน้าเครื่อง คอมพิวเตอร์แบบไม่ถูกสุขลักษณะตั้งแต่แขน ข้อมือ ข้อมือ แขน หลัง ต้นคอ หัวไหล่ และส่ายตา เนื่องจากอวัยวะส่วนที่มีปัญหาถูกวางค้าง ถูกทิ้งน้ำหนัก หรือกดทับนานๆ จนอักเสบ หากปล่อยไว้นานๆ อาจต้องผ่าตัดเอ็น

กลุ่มอาการปวดข้อ (Carpal Tunnel Syndrome: CTS) เป็น กลุ่มอาการของผู้ที่ใช้คอมพิวเตอร์เป็นเวลานานๆ ทำให้เกิดอาการของโรคกระดูกข้อมือเจ็บปวด ข้อกระดูกนิ้วมือเสื่อม และชา สาเหตุ เกิดจากการกดแป้นพิมพ์ และการใช้เมาส์ต่อเนื่องเป็นเวลานาน การจับเมาส์โดยมีข้อมือเป็นจุดหมุน อาจเกิดพังผืดบริเวณข้อมือ หากปล่อยทิ้งไว้เป็นเวลานานจะทำให้เกิดอาการชาจนไม่สามารถหยิบของได้

การ รักษาโรคที่กล่าวมาข้างต้น หากเริ่มมีอาการอาจต้องรับประทานยาแก้ปวดและหยุดการเคลื่อนไหวโดยการพักข้อ มือ อาการก็อาจทุเลาลงได้ อาการปวดจะหายไปในที่สุด หากปวดบวม ให้รับประทานยาบรรเทาปวดและอาจต้องสวมอุปกรณ์ประคองมือ เพื่อลดการเคลื่อนไหวของข้อมือ หรือนิยดา กลุ่มสเตียรอยด์เข้าบริเวณข้อมือ เพื่อลดการอักเสบโดยตรง ส่วนในรายที่เป็นมานานอาจจำเป็นต้องผ่าตัดจึงจะได้ผลดี

3. โรคที่เกิดจากเชื้อโรคที่มีอยู่ในคอมพิวเตอร์

โรคมูมิแพ้ มีการวิจัยค้นพบว่าสารเคมีจากจอคอมพิวเตอร์ก่อให้เกิดโรคมูมิแพ้ได้ สารนี้มีชื่อว่า Triphenyl Phosphate ที่ ใช้กันอย่างแพร่หลายทั้งในจอวิดีโอและคอมพิวเตอร์ สามารถก่อให้เกิดปฏิกิริยาภูมิแพ้ เช่น คัน คัดจมูก และปวดศีรษะ ผลวิจัยพบว่า เมื่อจอคอมพิวเตอร์ร้อนขึ้น

จะปล่อยสารเคมีดังกล่าวออกมา โดยเฉพาะหากสภาพภายในห้องทำงานที่มีเนื้อที่จำกัด เครื่องคอมพิวเตอร์อาจจะเป็นสาเหตุสำคัญที่ก่อให้เกิดโรคมะเร็งได้

4.โรคที่เกิดจากการใช้งาน

โรคทอร์นไนด์ (Hurry Sickness) มัก จะเกิดกับผู้ที่เล่นอินเทอร์เน็ต ที่ทำให้กลายเป็นคนใจ เบื่อ หงุดหงิดง่าย ใจร้อน เครียดง่าย เช่น ทนรอเครื่องดาวน์โหลดนานๆ ไม่ได้ กระวนกระวาย หาก มีอาการมากๆ ก็จะทำให้เข้าข่ายโรคประสาทได้ จึงควรปรับเปลี่ยนลักษณะงานและพยายามควบคุม อารมณ์ตนเอง

การใช้คอมพิวเตอร์อย่างปลอดภัย

ผู้มีอาชีพที่จะต้องนั่งอยู่กับคอมพิวเตอร์ทั้งวัน มักจะเกิดปัญหากับสุขภาพหลายอย่าง ทั้ง สายตา ปวดคอ ปวดหลัง ปัญหาทางสายตาเป็นเรื่องใหญ่ ใครที่นั่งจ้องจอคอมพิวเตอร์ นานกว่า 3 ชั่วโมงติดต่อกัน จะทำให้กล้ามเนื้อตาอ่อนล้า สายตาจะพร่า อาจปวดกระบอกตา แสบตา ตาแดง น้ำตาไหล มีวิธีป้องกันและแก้ไขดังนี้

ปัญหาการปวดตา แสบตา น้ำตาไหล

1. ควรตั้งจอคอมพิวเตอร์ให้ห่างอย่างน้อย 2 ฟุต ในระดับสายตาตรงหน้าพอดี
2. เลือกจอภาพที่มีการกระจายรังสีต่ำ รู้ได้โดยเวลาปิดเครื่องไฟฟ้าสถิตจะมีน้อย (ถ้ามีมาก ให้เอามือไปอังใกล้ๆ หน้าจอจะเกิดอาการช่นลุก)
3. ปรับแสงให้พอรู้สึกสบายตา อาจใช้แผ่นกรองแสงสวมหน้าจอจะช่วยได้ ไฟแสงสว่าง ด้านหลังอาจทำให้เกิดภาพสะท้อนที่จอทำให้สายตาเสียได้
4. ทำความสะอาดจอภาพของคอมพิวเตอร์เสมอ
5. ควรพักสายตาบ้าง ไม่ควรทำคอมพิวเตอร์ติดต่อกันนานเกิน 1-2 ชม. ควรพักสายตาสัก 15 นาที หรืออาจใช้ผ้าชุบน้ำเย็นหมาด ๆ ปิดตาไว้ 2-3 นาที จะช่วยได้มาก
6. ผู้ที่ใช้คอนแทกเลนส์ ควรหยอดน้ำตาเทียมบ่อย

ปัญหาปวดคอ ปวดบ่า ปวดไหล่

1. ตั้งจอตรงหน้าพอดีไม่สูง ไม่ต่ำ ไม่เอียงซ้าย หรือขวา
2. คีย์บอร์ดและเมาส์ควรอยู่ระดับเอวหรือระดับหน้าตักพอดี เพราะถ้าอยู่สูงกว่านี้เวลาใช้ คีย์บอร์ดและเมาส์นาน ๆ ไหลจะค่อย ๆ ยกสูงขึ้นโดยอัตโนมัติ เพื่อให้แขนและมือจะได้ทำงาน ถนัด แต่การยกไหล่ขึ้นนาน ๆ กล้ามเนื้อที่ยกไหล่จะล้า ปวดเมื่อยได้ ปวดตั้งแต่ไหล่ บ่า ถึงคอ
3. ต้องพักการทำคอมพิวเตอร์ ทุก 1-2 ชม.

ปัญหาปวดหลัง

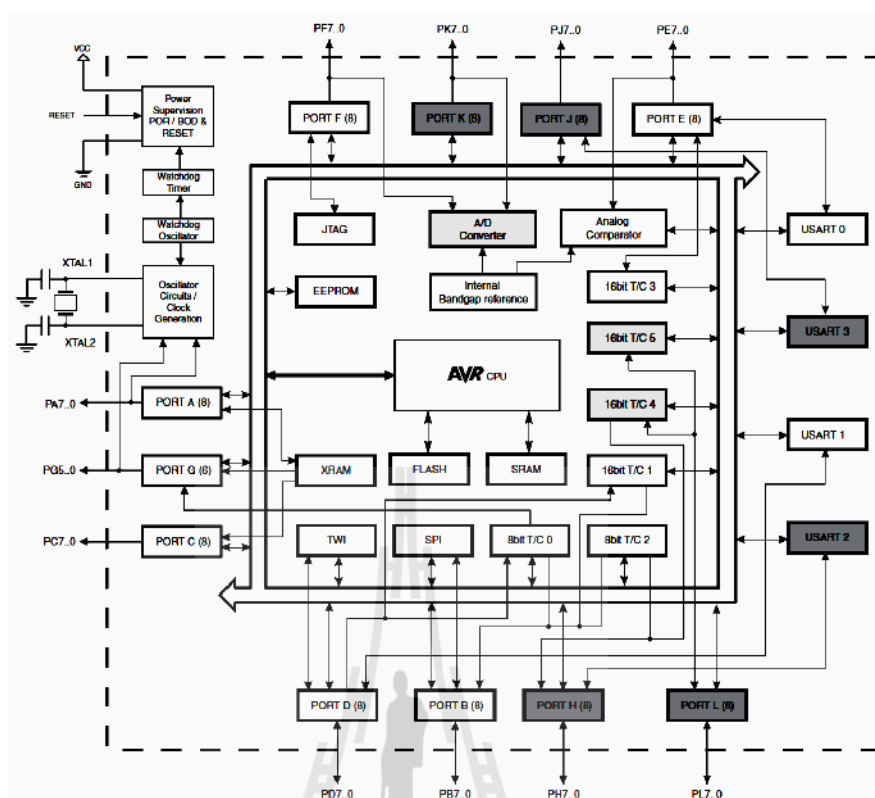
1. ขณะนั่งทำคอมพิวเตอร์ควรงั่งเก้าอี้ที่สูงพอดี เท้าวางบนพื้นได้เต็มเท้า ถ้าสูงเกินไปจนเท้าลอย หรือถ้าต่ำเกินไป ก้นจะจ่อมอยู่บนที่นั่ง ทำให้เมื่อยบริเวณก้นได้
2. เวลานั้นต้องเลื่อนตัวให้นั่งชิดพนักพิง ไม่ให้นั่งอยู่แค่ครึ่งที่นั่งของเก้าอี้
3. หลังจะต้องพิงพนักเก้าอี้อยู่ตลอดเวลา โดยพนักพิงทำมุมกับที่นั่งไม่เกิน 100 องศา
4. ต้องพักการใช้คอมพิวเตอร์ ทุก 1-2 ชม.

2.3 ไมโครคอนโทรลเลอร์ตระกูล AVR

ไมโครคอนโทรลเลอร์ตระกูล AVR เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต เช่น เบอร์ AT90S2313 AT90S2343 และ AT90S1200-12PC เป็นต้น ต่อมาบริษัท Atmel Corporation ได้พัฒนาปรับปรุงสถาปัตยกรรมจากอนุกรม AT90S มาสู่ Attiny สำหรับ AVR ในรุ่นเล็ก เช่น รุ่น Attiny2313 ขนาด 20 ขา และรุ่น ATmega สำหรับรุ่นใหญ่ เช่น เบอร์ ATmega16 ATmega8535 ขนาด 40 ขา ATmega128 ขนาด 64 ขา โดยใช้ซอฟต์แวร์ชื่อว่า AVR Studio เป็นชุดพัฒนาโปรแกรมที่รองรับทั้งภาษาแอสเซมบลีและภาษาซี

2.3.1 โครงสร้างภายในไมโครคอนโทรลเลอร์ตระกูล AVR

ไมโครคอนโทรลเลอร์ตระกูล AVR ใช้เทคโนโลยีการผลิตแบบ CMOS ที่ใช้พลังงานต่ำบนพื้นฐานของ AVR และยังเพิ่มสถาปัตยกรรม RISC ซึ่งภายในได้รวมวงจรต่างๆไว้อย่างครบถ้วนพร้อมที่จะทำงานได้เมื่อจ่ายไฟเลี้ยงและสัญญาณนาฬิกา (clock) ไมโครคอนโทรลเลอร์ตระกูล AVR ได้ถูกผลิตออกมามากหลายเบอร์โดยบริษัทต่างๆไม่ว่าจะเป็นเบอร์อะไรก็ตาม ถ้าเป็นไมโครคอนโทรลเลอร์ตระกูล AVR แล้วจะมีโครงสร้างต่างๆคล้ายกัน จะต่างกันออกไปในส่วนของคุณสมบัติพิเศษในแต่ละเบอร์ โดยที่โครงสร้างต่างๆภายในของไมโครคอนโทรลเลอร์ตระกูล AVR ได้แสดงไว้ดังรูปที่ 2.2



รูปที่ 2.2 บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์ตระกูล AVR [4]

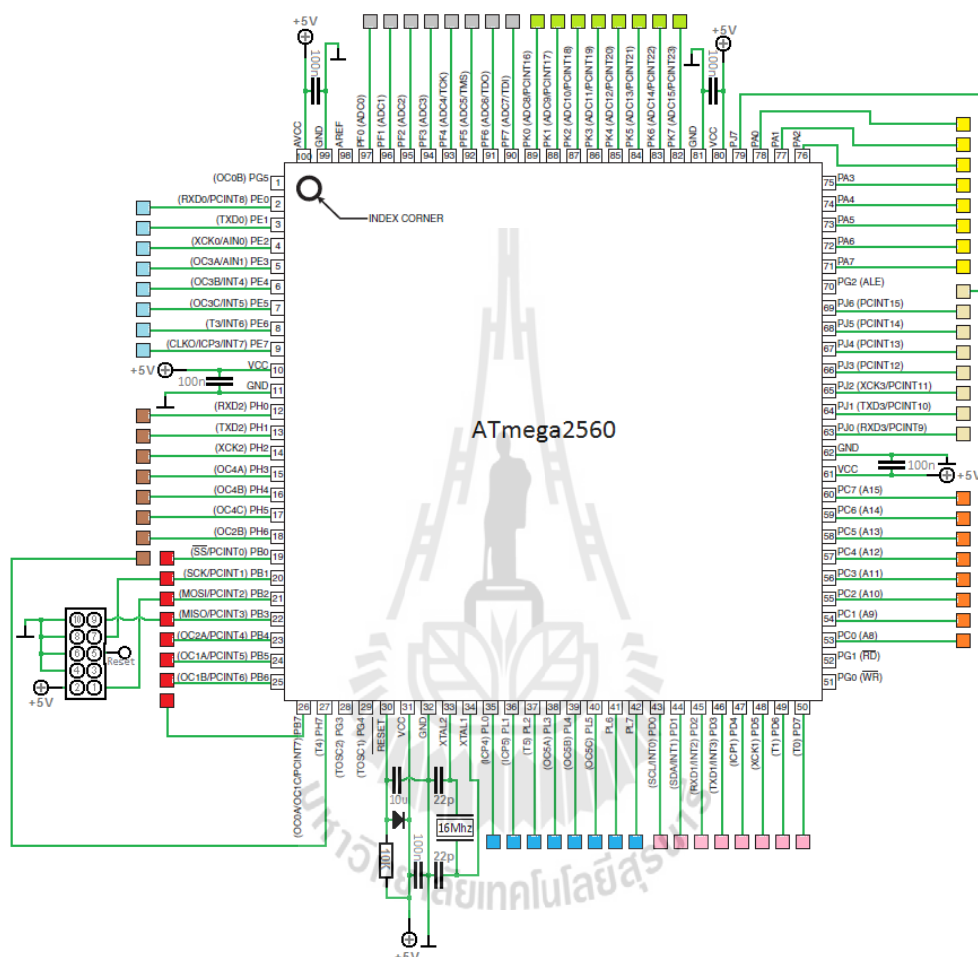
2.3.2 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล AVR

เนื่องจากคุณสมบัติไมโครคอนโทรลเลอร์แต่ละเบอร์นั้นมีความสารที่แตกต่างกันออกไป ในรายละเอียดปลีกย่อย ดังนั้นจะขออ้างอิงเบอร์ ATmega2560 ของบริษัท Atmel ซึ่งมีคุณสมบัติดังนี้

- มีหน่วยความจำโปรแกรมแบบแฟลช (Flash memory) ขนาด 256 กิโลไบต์ (kbyte)
- มีหน่วยความจำแบบ EEPROM ขนาด 4 กิโลไบต์ (kbyte)
- แหล่งจ่ายไฟกระแสตรงขนาด 5 โวลต์ (ทำงานในช่วง 4.5-5.5 โวลต์)
- ทำงานได้ด้วยสัญญาณนาฬิกาตั้งแต่ 0-16 เมกกะเฮิร์ต (MHz)
- สามารถสื่อสารข้อมูลแบบอนุกรมได้ด้วย UART Channel
- มีหน่วยความจำข้อมูลแบบสแตติกแรม (SRAM) 8 กิโลไบต์
- ความละเอียดของตัวแปลงอนาล็อกเป็นดิจิทัล (ADC) 10 บิต
- มีจำนวนตัวจับเวลาและตัวนับ 6 ไทเมอร์

2.3.3 การจัดขาของไมโครคอนโทรลเลอร์ตระกูล AVR

ไมโครคอนโทรลเลอร์ในตระกูล AVR จะมีโครงสร้างของการจัดเรียงขาคล้ายกันได้แสดงไว้ดังรูปที่ 2.3 และมีส่วนประกอบของขาต่อไปนี้



รูปที่ 2.3 การจัดขาของ AVR [4]

VCC ต่อไฟเลี้ยง

GND ต่อกราวด์

Port A (PA0-PA7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน สามารถทำงานได้สองหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port B (PB0-PB7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน สามารถทำงานได้สองหน้าที่เป็นพอร์ตอินพุตและเอาต์พุต

ทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอกเป็นพอร์ตที่อ่านและบันทึกข้อมูลได้ดีกว่าพอร์ตอื่นๆ

Port C (PC0-PC7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน สามารถทำงานได้สองหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port D (PD0-PD7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port E (PE0-PE7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port F (PF0-PF7) พอร์ต F ทำหน้าที่เป็นอนาล็อก A/D convertor ถ้า A/D convertor ไม่ได้ใช้จะทำหน้าที่เป็นพอร์ต 8 บิตแบบสองทิศทาง I/O มีการต่อตัวต้านทานพูลอัพไว้ภายในและเป็นพอร์ตที่สามารถอ่านและบันทึกข้อมูลได้ ถ้า JTAG อินเตอร์เฟซเปิดใช้งานจะถูกเปิดพูลอัพ ถึงแม้ PF7 (TDI) PF5 (TMS) และ PF4 (TCK) ไม่มีการตั้งค่า

Port G (PG0-PG5) เป็นพอร์ตแบบสองทิศทางขนาด 6 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port H (PH0-PH7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port J (PJ0-PJ7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port K (PK0-PK7) พอร์ต F ทำหน้าที่เป็นอนาล็อก A/D convertor เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

Port L (PL0-PL7) เป็นพอร์ตแบบสองทิศทางขนาด 8 บิต มีการต่อตัวต้านทานพูลอัพ (pull-up register) ไว้ภายใน ทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตทั่วไป นอกจากนี้ยังใช้เป็นพอร์ตสำหรับติดต่อกับหน่วยความจำภายนอก

RESET รีเซตการป้อนข้อมูล

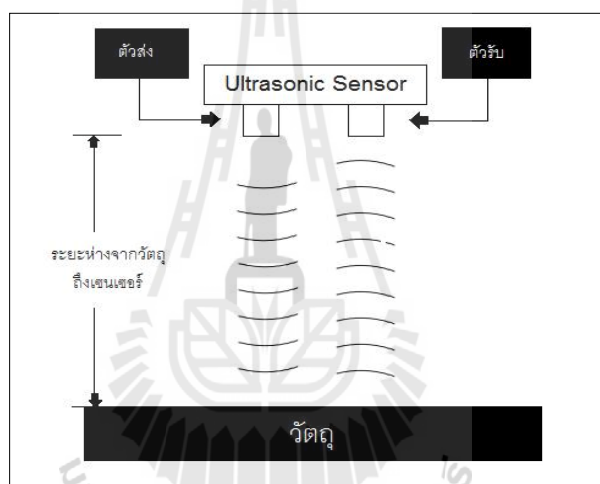
XTAL1 อินพุตที่ Inverting oscillator amplifier และอินพุตที่วงจรการทำ Clock ภายใน

XTAL2 เอาต์พุตจาก Inverting oscillator amplifier

AVCC ขาแรงดันสำหรับพอร์ต F และ A/D convertor ควรจะเชื่อมต่อ VCC ภายนอก แม้ว่า ADC ไม่ได้ใช้ หาก ADC จะใช้ก็ควรจะมีการใช้ VCC ผ่าน low-pass filter

Aref ขาอ้างอิงสำหรับอนาล็อก A/D convertor

2.4 เซนเซอร์วัดระยะทาง รุ่น US-015



รูปที่ 2.4 หลักการตรวจจับวัตถุโดยใช้สัญญาณความถี่เหนือเสียง [5]

เครื่องตรวจจับและวัดระยะทางด้วยอัลตราโซนิกเป็นแผงวงจรวัดตรวจจับระยะทางด้วยอัลตราโซนิกที่มีความแม่นยำและเที่ยงตรงสูง โดยสามารถวัดระยะทางได้ตั้งแต่ 1 เซนติเมตรไปจนถึง 4 เมตร โดยเครื่องตรวจจับและวัดระยะทางด้วยอัลตราโซนิกถูกออกแบบมาให้ใช้งานกับไมโครคอนโทรลเลอร์ได้ง่ายโดยใช้ขาเชื่อมต่อเพียง 1 หรือ 2 ขา โดยจะทำการส่งสัญญาณคลื่นอัลตราโซนิกออกไปแล้ววัดระยะที่มีสัญญาณสะท้อนตอบกลับมา โดยเอาต์พุตที่ได้จะอยู่ในรูปของความกว้างพัลส์ซึ่งสัมพันธ์กับระยะทางของวัตถุที่ตรวจจับได้ โดยความถี่อัลตราโซนิกที่ถูกส่งออกไปในอากาศด้วยความเร็ว 1.125 พุดต่อมิลลิวินาที (ประมาณ 346 เมตรต่อวินาที) ดังนั้นเมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น เวลาเริ่มส่งคลื่นและเวลาที่รับเสียงสะท้อนกลับมาจึงสามารถคำนวณค่าของระยะทางได้ แสดงหลักการตรวจจับในรูปที่ 2.4

ระยะทางที่ได้นั้นจะต้องมีการคำนวณค่ากลับทางคณิตศาสตร์เมื่อใช้กับไมโครคอนโทรลเลอร์แล้วถือว่าเป็นเรื่องที่ยากพอสมควร ดังนั้นเครื่องตรวจจับและวัดระยะทางด้วยอัลตราโซนิกจึงประมวลผลค่ากลับทางคณิตศาสตร์ต่างๆไว้เรียบร้อยแล้ว จากนั้นส่งผลลัพธ์ที่วัดได้ออกมาเป็นพัลส์ที่มีความกว้างสัมพันธ์กับระยะทางที่วัดได้ จะใช้สายสัญญาณเพียงเส้นเดียวจึงทำให้สะดวกมากในการนำมาเชื่อมต่อไมโครคอนโทรลเลอร์



รูปที่ 2.5 เซนเซอร์วัดระยะทางรุ่น US-015 [6]

2.4.1 คลื่นอัลตราโซนิก (Ultrasonic)

ระบบอัลตราโซนิก หมายถึง คลื่นเสียงที่มีความถี่สูงเกินกว่าที่มนุษย์จะได้ยิน โดยทั่วไปแล้วหูของมนุษย์โดยเฉลี่ยจะได้ยินเสียงสูงถึงเพียงแค่ประมาณ 15 KHz เท่านั้น ดังนั้นโดยปกติแล้วคำว่าอัลตราโซนิกหมายถึงคลื่นเสียงที่มีความสูงกว่า 20 KHz ขึ้นไป สาเหตุที่มีการนำเอาคลื่นย่านอัลตราโซนิกมาใช้ก็เพราะว่าเป็นคลื่นที่สามารถกำหนดการส่งคลื่นไปยังเป้าหมายที่ต้องการได้โดยเจาะจง เรื่องนี้เป็นคุณสมบัติของคลื่นอย่างหนึ่ง ยิ่งคลื่นมีความถี่สูงขึ้นความยาวคลื่นก็จะยิ่งสั้นลง ถ้าความยาวคลื่นยาวกว่าช่องเปิด(ที่ให้เสียงนั้นออกมา)ของตัวกำเนิดเสียงความถี่นั้น เช่น คลื่นความถี่ 300 Hz ในอากาศจะมีความยาวถึงประมาณ 1000 กิโลเมตร ซึ่งจะยาวกว่าช่องที่ให้คลื่นเสียงออกมาจากตัวกำเนิดเสียงโดยทั่วไปมากมายคลื่นจะหักเหที่ขอบด้านนอกของตัวกำเนิดเสียงทำให้เกิดการกระจายทิศทางคลื่นแต่ถ้าความถี่สูงขึ้นมาอยู่ในย่านอัลตราโซนิก อย่างเช่น 40 KHz จะมีความยาวคลื่นในอากาศเพียง 8 มม.เท่านั้นซึ่งเล็กกว่ารูเปิดของตัวที่ให้กำเนิดเสียงความถี่นี้มากคลื่นเสียงจะไม่มีการเลี้ยวเบนที่ขอบจึงพุ่งออกมาเป็นลำแคบๆหรือที่เรียกว่ามีทิศทาง

การมีทิศทางของคลื่นเสียงย่านอัลตราโซนิกทำให้นำไปใช้งานได้หลายอย่าง เช่น นำไปใช้ในเครื่องควบคุมระยะไกล (Ultrasonic remote control) โดยให้น้ำหนักความถี่สูงเครื่องวัดความหนาของวัตถุโดยตั้งเกดระยะเวลาที่คลื่นสะท้อนกลับมา เครื่องวัดความลึกและทำแผนที่ใต้ท้องทะเล เป็นต้น โดยความถี่ที่ใช้ขึ้นอยู่กับการใช้งาน เช่น คลื่นเสียงต้องเดินทางผ่านอากาศแล้ว ความถี่ที่ใช้ก็มักจะจำกัดอยู่เพียงไม่เกิน 50 KHz เพราะที่ความถี่สูงเกินกว่านี้อากาศจะดูดกลืนคลื่นเสียงเพิ่มขึ้นมาก ทำให้ระดับความแรงของคลื่นเสียงที่ระยะห่างออกไปลดลงอย่างรวดเร็ว ส่วนการใช้งานด้านการแพทย์ซึ่งต้องการรัศมีทำการสั้นๆก็อาจใช้ความถี่ในช่วง 1 MHz ถึง

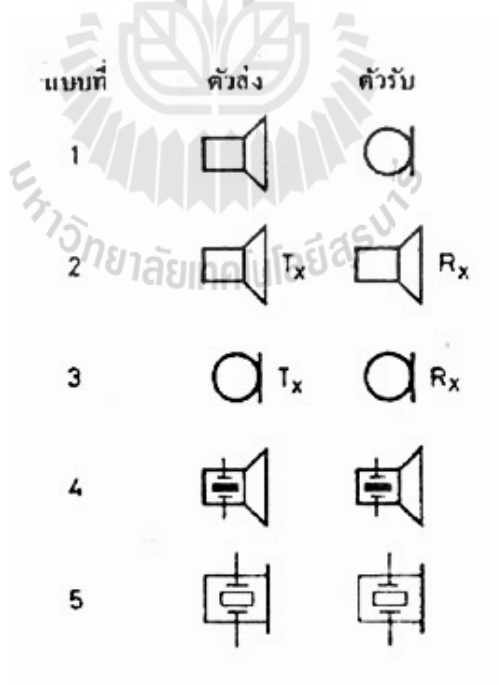
10 MHz ขณะที่ความถี่เป็น GHz ก็มีใช้กันในหลายๆการใช้งานที่ตัวกลางที่คลื่นเสียงเดินทางผ่านไม่ใช่อากาศอุปกรณ์ที่สามารถแปลงพลังงานในรูปอื่นให้มาเป็นพลังงานทางกล โดยการสั่นไปมาซึ่งทำให้เกิดคลื่นเสียงย่านอัลตราโซนิกกระจายไปในอากาศได้หรือแปลงพลังงานทางกลให้มาเป็นพลังงานในรูปอื่นได้นั้น มีชื่อเรียกว่า อัลตราโซนิกทรานสดิวเซอร์ (Ultrasonic Transducer) ในปัจจุบันอัลตราโซนิกทรานสดิวเซอร์มีหลายแบบขึ้นอยู่กับหลักการที่ใช้ แบบที่นิยมใช้กันมากได้แก่

-แบบเพียโซอิเล็กทริก(Piezo-electric Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าและพลังงานทางกล โดยมีความถี่เรโซแนนซ์คงที่อยู่มาก

-แบบแมกนีโตสตริกทีฟ (Magnetostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้าในขดลวดกับตำแหน่งความยาวของแกนเหล็กที่สวมขดลวดนั้นอยู่

-แบบอิเล็กโตรสตริกทีฟ(Electrostrictive Transducer) ซึ่งแปลงไปมาระหว่างพลังงานไฟฟ้ากับพลังงานทาง-กล

โดยทั่วไป ลักษณะปกติของตัวส่งอัลตราโซนิก จะเปรียบเสมือนลำโพงส่งสัญญาณเสียง และตัวรับจะมีลักษณะเหมือนไมโครโฟน ดังนั้นสัญลักษณ์ที่ใช้จึงมีลักษณะคล้ายๆกัน ซึ่งก็มีหลายแบบ คือ



รูปที่ 2.6 สัญลักษณ์ของอัลตราโซนิก [7]

ตัวส่ง คือ อัลตราโซนิกที่ถูกออกแบบเจาะจงมาให้แปลงสัญญาณไฟฟ้าที่ให้แก่ตัวมัน ให้ออกมาเป็นคลื่นเสียงย่านอัลตราโซนิก หน้าที่ของตัวส่งจึงคล้ายๆกับเป็นลำโพง

ตัวรับ คือ อัลตราโซนิกที่ถูกออกแบบเจาะจงมาให้แปลงคลื่นเสียงย่านอัลตราโซนิกที่มาจากกระทบตัวมัน ให้ออกมาเป็นสัญญาณไฟฟ้าหน้าที่ของตัวรับจึงคล้ายๆกับเป็นไมโครโฟน ด้วยเหตุนี้เวลาเขียนสัญลักษณ์ของอัลตราโซนิกจึงนิยมเขียนตามหน้าที่ของมันคือถ้าเป็นตัวส่งก็เขียนสัญลักษณ์เป็นลำโพง ถ้าเป็นตัวรับก็เขียนสัญลักษณ์เป็นไมโครโฟน

2.4.2 การต่อใช้งานของ US-015

มีจุดต่อสำหรับใช้งานอยู่ทั้งหมด 4 จุด

1.ขาไฟเลี้ยง(+5V) สำหรับต่อไฟเลี้ยงแรงดัน +5V

2.ขาEcho Pulse Output(ECHO) เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจากUS-015 ซึ่งการใช้งานจะนำขานี้ไปต่อเข้ากับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ เพื่อตรวจจับความกว้างของสัญญาณพัลส์ที่ส่งออกมาเพื่อแปลความหมายออกมาเป็นระยะทางอีกครั้งหนึ่ง

3.ขาTrigger Pulse Input (TRIGGER) เป็นขาอินพุตรับสัญญาณพัลส์ที่มีความกว้างอย่างน้อย10ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตราโซนิกความถี่40kHzออกสู่อากาศจากตัวส่ง ดังนั้นเมื่อคลื่นความถี่ดังกล่าวนี้เคลื่อนที่กระทบสิ่งกีดขวางที่อยู่เบื้องหน้าก็จะเกิดการสะท้อนกลับเข้ามายังตัวรับและถูกแปลงออกมาเป็นความกว้างของสัญญาณพัลส์ที่จะส่งออกไปทางขา Echo Pulse Output นอกจากนี้ในโหมด1สัญญาณ จะใช้จุดนี้เป็นจุดสื่อสารข้อมูลอนุกรมเพื่อรับส่งค่าการวัดกับไมโครคอนโทรลเลอร์

4.ขา GND สำหรับต่อกราวด์

2.4.3 หลักการทำงานของอัลตราโซนิกเซนเซอร์

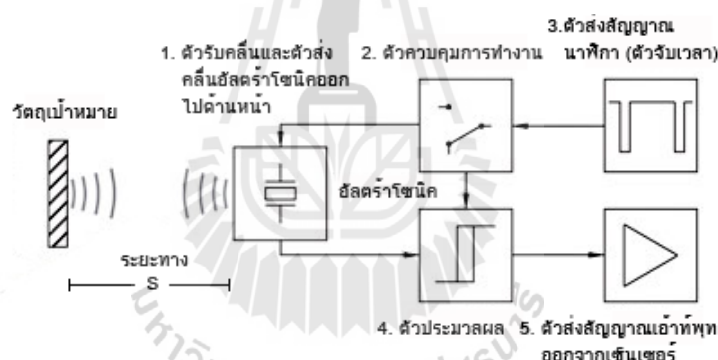
- US-015 เซ็นเซอร์วัดระยะทางอัลตราโซนิกโมดูล
- แรงดันไฟฟ้าในการทำงาน:DC 5 V
- กระแสคงที่:2.2 มิลลิแอมป์
- อุณหภูมิในการทำงาน:0 ~ + 70 °
- วิธีเอาต์พุต:GPIO
- การเหนี่ยวนำมุม: น้อยกว่า 15 องศา
- ช่วงการตรวจสอบ:2 ซม. ถึง 400 ซม.
- ความแม่นยำ:0.3 ซม. + 1%

- ขนาดเซนเซอร์: 45 x 20 x 1.6mm
- เส้นผ่าศูนย์กลางรู:1mm

$$\text{ระยะทาง (cm)} = \text{ระยะเวลาในหน่วยไมโครวินาที} / 29 \text{ (us/cm)} / 2 \quad (2.1)$$

อัลตราโซนิกเซนเซอร์เป็นเซนเซอร์ที่ใช้คลื่นเสียงในการตรวจจับตำแหน่งของวัตถุโดยส่วนประกอบของตัวเซนเซอร์จะประกอบด้วย

- 1.ตัวส่งคลื่นอัลตราโซนิกและตัวรับคลื่นอัลตราโซนิก(อัลตราโซนิกทรานสดิวเซอร์)
- 2.ตัวควบคุมการทำงาน
- 3.ตัวส่งสัญญาณนาฬิกา (ตัวรับเวลา)
- 4.ตัวประมวลผล
- 5.ตัวส่งสัญญาณเอาต์พุต



รูปที่ 2.7 โค้ดโปรแกรมภายในอัลตราโซนิกเซนเซอร์ [8]

โดยตัวเซนเซอร์จะทำงานโดย ตัวส่งสัญญาณจะส่งสัญญาณนาฬิกาไปที่ตัวคอนโทรลเลอร์เพื่อควบคุมการแปลงสัญญาณ แล้วส่งไปที่ตัวอัลตราโซนิกทรานสดิวเซอร์ซึ่งแบ่งเป็นสองส่วนคือ ตัวส่งและตัวรับ ตัวส่งจะสร้างคลื่นเสียงอัลตราโซนิก จากสัญญาณไฟฟ้าแล้วส่งคลื่นเสียงความถี่สูงหรืออัลตราโซนิกออกไปเป็นแนวตรง และเมื่อคลื่นเสียงอัลตราโซนิกไปกระทบกับวัตถุใดๆ ตามหลักการของคลื่นเสียง คือ มุมตกกระทบเท่ากับมุมสะท้อน คลื่นเสียงจะถูกสะท้อนกลับมาที่ตัวรับคลื่นเสียงอัลตราโซนิก เมื่อตัวรับได้รับคลื่นเสียงที่ถูกสะท้อนกลับมาแล้ว ตัวรับจะแปลงคลื่นเสียงอัลตราโซนิกนั้นเป็นสัญญาณไฟฟ้าแล้วส่งต่อ ให้ตัวประมวลผล ตัวประมวลผลจะ

ทำการคำนวณค่าระยะห่างจากระยะทางที่คลื่นเสียงเดินทางไปและเดินทางกลับอย่างแม่นยำ และส่งค่าที่คำนวณได้ไปให้ตัวส่งสัญญาณเอ้าท์พุท เพื่อส่งสัญญาณเอ้าท์พุทไปให้อุปกรณ์อื่นต่อไป

หลักการคำนวณ จะเป็นไปตามสูตรการเคลื่อนที่ในแนวราบ $S = VT$ โดย S = ระยะทาง, V = ความเร็วของคลื่นเสียง, T = ระยะเวลาที่คลื่นเสียงเดินทางทั้งหมด ดังตัวอย่างต่อไปนี้

ตัวอย่างการคำนวณเซนเซอร์ส่งคลื่นเสียงออกไปในอากาศที่ความเร็ว 343 M/S และได้รับคลื่นเสียงที่ถูกสะท้อนกลับมากในระยะเวลา 4 วินาทีจะได้ว่า

จากสูตรการเคลื่อนที่ในแนวตรง $S = VT$

$S = V(T/2)$; $(T/2)$ เนื่องจาก เวลาที่นับเป็นเวลาที่เสียงเดินทางทั้งขาไป

$S = 343(4/2)$ และหากกลับ เราต้องการแค่ระยะห่าง จึงต้องนำไปหาร 2

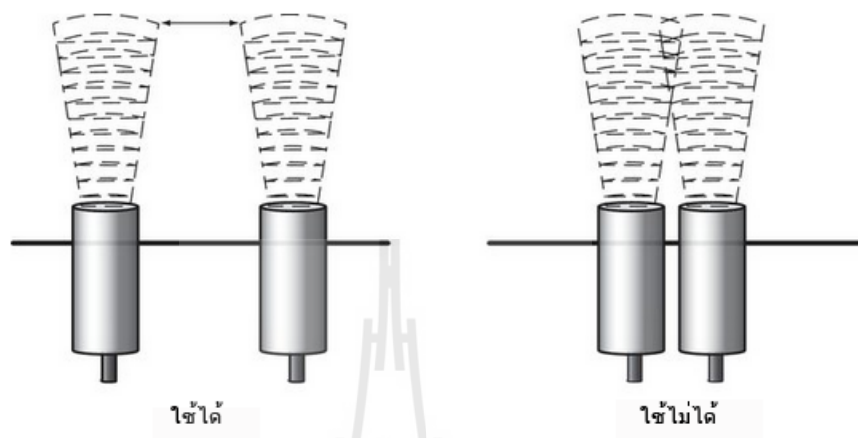
$S = 686 \text{ M.}$

จะได้ว่าระยะห่างของเซนเซอร์กับตัววัตถุมีระยะห่างกันเท่ากับ 686 เมตร

ซึ่งหลักการวัดระยะห่างของเซนเซอร์ชนิดอัลตราโซนิกนี้ มีประโยชน์เป็นอย่างมาก เนื่องจากสามารถนำไปตรวจจับวัตถุได้เกือบทุกประเภท เหมาะสำหรับการวัดระยะสิ่งของที่อยู่ระยะไกลมากๆ ในสภาวะอากาศที่เลวร้าย มีความสกปรกมากหรือมีฝุ่นมาก และยังสามารถใช้กับวัตถุที่เป็นของเหลววัตถุที่มีพื้นผิววัตถุเป็นแบบมันวาว โปร่งแสงหรือโปร่งใส ซึ่งเซนเซอร์ชนิดอื่นจะทำได้ไม่ดีเทียบเท่ากับอัลตราโซนิกเซนเซอร์ เนื่องจากการใช้คลื่นเสียงในการทำงาน ทำให้ไม่ถูกรบกวนด้วยสิ่งต่างๆที่กล่าวมาข้างต้น แต่ก็มีวัตถุบางประเภทที่ไม่เหมาะจะนำอัลตราโซนิกเซนเซอร์ไปใช้จับระยะทาง เช่น 1. วัตถุที่สามารถดูดซับเสียงได้เช่น ผ้า หรือโฟมต่างๆที่มีคุณสมบัติสามารถดูดซับเสียงได้เป็นอย่างดี 2. ไม่เหมาะกับการนำไปใช้กับวัตถุขนาดเล็กมากจนเกินไป เนื่องจากหน้าสัมผัสของวัตถุที่มีน้อย จึงสะท้อนคลื่นเสียงกลับมได้น้อย ทำให้การคำนวณระยะทางหรือตำแหน่งอาจจะไม่แม่นยำเท่าที่ควร ซึ่งวัตถุที่มีขนาดเล็กนั้นแนะนำให้ใช้ โฟโตอิเล็กทริกเซนเซอร์ (Photoelectric Sensors)

การจัดวางตำแหน่งของเซนเซอร์

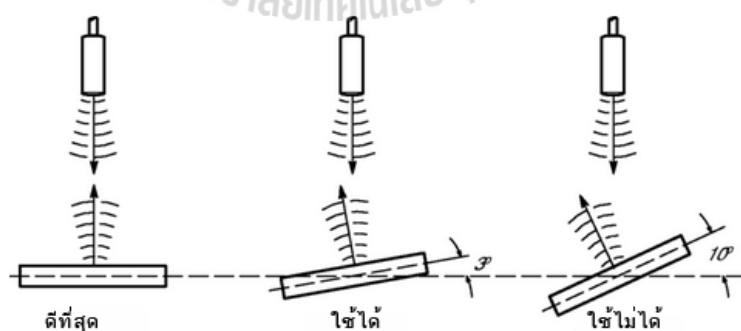
การเว้นระยะห่างของตัวเซนเซอร์ อัลตราโซนิกเซนเซอร์เป็นเซนเซอร์ที่ใช้การส่งคลื่นเสียงในการทำงานทำให้ต้องมีการเว้นระยะห่างของตัวเซนเซอร์ ดังรูปที่ 2.8



รูปที่ 2.8 การเว้นระยะห่างของเซนเซอร์ [8]

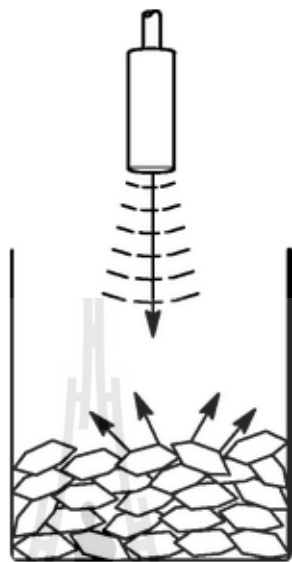
จะเห็นว่าเราควรเว้นระยะห่างของเซนเซอร์ เพื่อไม่ให้คลื่นเสียงที่ส่งออกไปมีการรบกวนกัน เพื่อประสิทธิภาพและความแม่นยำในการตรวจจับของตัวเซนเซอร์

การวางสิ่งของที่จะทำการตรวจจับ ควรจัดวางสิ่งของที่จะทำการตรวจจับให้มีขนาดที่สามารถสะท้อนคลื่นเสียงที่ส่งไปกลับมาได้ ตามคุณสมบัติของคลื่นเสียงที่ว่า มุมตกกระทบเท่ากับมุมสะท้อน



รูปที่ 2.9 การจัดวางแนวระนาบของสิ่งของ [8]

วัตถุที่จะทำการสะท้อนนั้น ควรจะมีลักษณะเป็นแนวราบ เพื่อให้ได้ประสิทธิภาพในการตรวจจับดีที่สุด ถ้าวัตถุที่ทำการตรวจจับเป็นชิ้นเล็กๆหรือมีรูปร่างที่ไม่แน่นอน จะทำให้ความสามารถในการตรวจจับและความแม่นยำลดลง



รูปที่ 2.10 การตรวจจับวัตถุที่มีรูปร่างไม่แน่นอน [8]

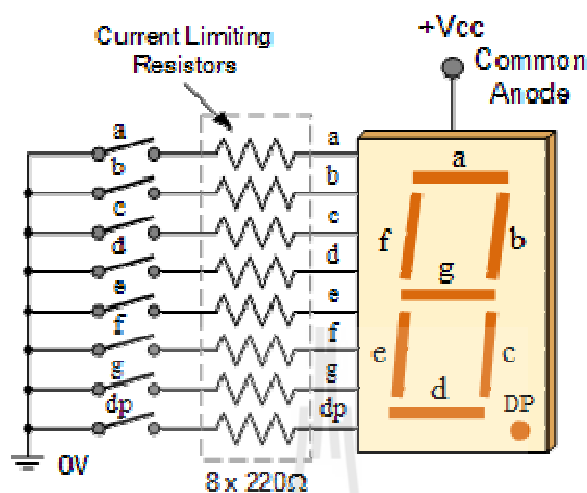
2.5 ตัวเลขแสดงผล 7-segment

วงจร LED 7-Segment เป็นรูปแบบของอุปกรณ์การแสดงผลชนิดหนึ่ง ซึ่งเป็นอีกทางเลือกนอกเหนือจากวงจรแสดงผลแบบ Dot-matrix วงจรแสดงผลแบบ LED หัวไป 7-Segment ถูกใช้อย่างแพร่หลายในวงการอุตสาหกรรมอิเล็กทรอนิกส์เพื่อใช้แสดงผลตัวเลขฐานสิบของกระบวนการทำงานภายในของอุปกรณ์ต่างๆ

2.5.1 โครงสร้างภายในของ 7-Segment

จอแสดงผลแบบ 7-Segments นั้นภายในจะประกอบไปด้วยหลอด LED ที่นำมาต่อกัน 7 ตัว ซึ่งจะถูกทำการเปิดหรือปิดเพื่อแสดงรูปแบบของตัวเลขฐานสิบ จะสามารถแสดงตัวเลขได้ตั้งแต่เลข 0 ถึงเลข 9 (บางรุ่นจะมี จุด แสดงด้วย) ในการใช้งาน 7-Segments นั้นใช้หลักการเขียนโปรแกรมควบคุมเหมือนกับการเขียนโปรแกรม ควบคุม LED หากเรารู้ตำแหน่งขาที่ต้องการก็สามารถจะเขียนแสดงเลขต่างๆได้ การต่อวงจรภายในของ 7-Segments จะมีการต่อได้ 2 แบบ คือ การต่อแบบ คอมมอนแอนอด(A) และ การต่อแบบคอมมอนคาโทด (K) ซึ่งการใช้งานของทั้ง 2 แบบนั้นจะแตกต่างกันตรงที่การป้อนสัญญาณ เพื่อให้ 7-Segments แสดงผลจะตรงกันข้ามกัน โดยที่

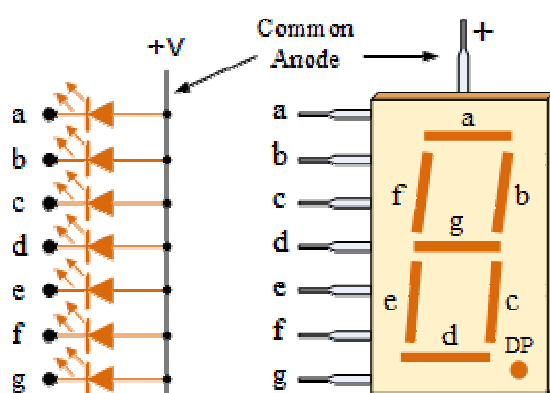
ตำแหน่งของ LED ภายใน 7-Segments จะถูกอ้างอิงจากตัวอักษรที่มีชื่อเรียกที่ใช้กันทั่วไป คือ a , b , c , d , e , f , g ได้แสดงไว้ดังรูปที่ 2.11



รูปที่ 2.11 ตำแหน่ง a – g ภายในและสัญลักษณ์ 7-Segment[9]

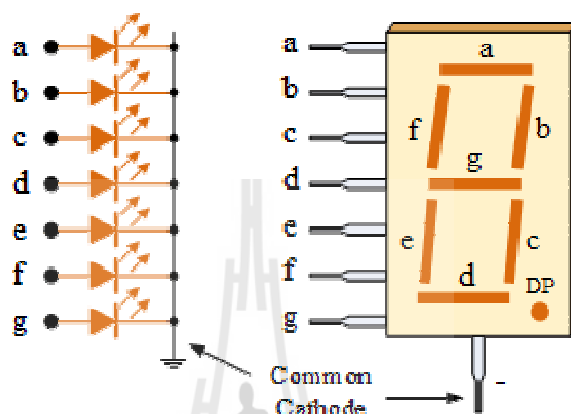
หลอดแสดงผล LED 7-Segment สามารถแบ่งตามลักษณะการต่อหลอดแสดงผล LED ทั้ง 7 หลอด ได้ 2 ชนิด ดังนี้

1. ชนิดต่อแบบแอโนดร่วม หรือ คอมมอนแอโนด (Common Anode) เป็นการต่อหลอดแสดงผล LED 7-Segment แบบแอโนดร่วม โดยต้องป้อนไฟบวกที่ขาาร่วมที่หรือขาคอมมอน (Common) และถ้าต้องการให้หลอดแสดงผล LED 7-Segment แสดงผลหรือให้สว่าง ต้องป้อนไฟลบหรือส่งลอจิก "0" มาที่ขาแคโทดได้แสดงไว้ดังรูปที่ 2.12



รูปที่ 2.12 การต่อโครงสร้างภายในแบบคอมมอนแอโนด(A) [9]

2. ชนิดต่อแบบแคโทดร่วม หรือ คอมมอนแคโทด (Common Cathode) เป็นการต่อหลอดแสดงผล LED 7-Segment แบบแคโทดร่วม โดยต้องป้อนไฟลบหรือกราวด์ที่ขาร่วมหรือขาคอมมอน และถ้าต้องการให้หลอดแสดงผล LED 7-Segment แสดงผลหรือสว่างต้องป้อนไฟบวกหรือส่งลอจิก "1" มาที่ขาแอนโนด ได้แสดงไว้ดังรูปที่ 1.13



รูปที่ 2.13 การต่อโครงสร้างภายในแบบคอมมอนคาโทด(K)[9]

การที่ตัวแสดงผล 7-Segment จะสามารถสื่อความหมายเป็นตัวเลขได้ เราจำเป็นต้องกำหนดรูปแบบควบคุมการติดสว่างดับ ในแต่ละบิตของ 7-Segment โดยจะต้องดูจากโครงสร้างการต่อวงจร ทั้งแบบคอมมอนแอนโนด (Common Anode) และคอมมอนแคโทด (Common Cathode) ต่อเข้ากับบอร์ดทดลอง ดังนั้นถ้าต้องการให้บิตใดของ 7-Segment แสดงผล ก็จะต้องป้อนลอจิก "1" และในทางกลับกันถ้าไม่ต้องการให้บิตใดแสดงผล จะต้องป้อนลอจิก "0" ให้กับ 7-Segment ซึ่งสามารถสรุปรูปแบบการแสดงผลได้ดังตารางที่ 2.1 และตารางที่ 2.2 ตามลำดับ

ตารางที่ 2.1 รหัสการแสดงผลสำหรับ 7-Segment Common Anode เป็นตัวเลข 0-9 [10]

ค่าข้อมูล(เลขฐานสอง)							ค่าข้อมูลส่งออกพอร์ต (เลขฐานสิบหก)	แสดงผลตัวเลข
g	f	e	d	c	b	a		
1	0	0	0	0	0	0	0x40	0
1	1	1	1	0	0	1	0x79	1
0	1	0	0	1	0	0	0x24	2
0	1	1	0	0	0	0	0x30	3
0	0	1	1	0	0	1	0x19	4
0	0	1	0	0	1	0	0x12	5
0	0	0	0	0	1	0	0x02	6
1	1	1	1	0	0	0	0x78	7
0	0	0	0	0	0	0	0x00	8
0	0	1	0	0	0	0	0x10	9

แสดงข้อมูลที่ส่งออกพอร์ตของหลอดแสดงผล LED 7-Segmentแบบต่อแอโนดร่วม (Common Anode)

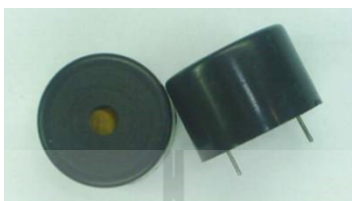
ตารางที่ 2.2 รหัสการแสดงผลสำหรับ 7-Segment Common Cathode เป็นตัวเลข 0-9 [10]

ค่าข้อมูล(เลขฐานสอง)							ค่าข้อมูลส่งออกพอร์ต (เลขฐานสิบหก)	แสดงผลตัวเลข
g	f	e	d	c	b	a		
0	1	1	1	1	1	1	0x3F	0
0	0	0	0	1	1	0	0x06	1
1	0	1	1	0	1	1	0x5B	2
1	0	0	1	1	1	1	0x4F	3
1	1	0	0	1	1	0	0x66	4
1	1	0	1	1	0	1	0x6D	5
1	1	1	1	1	0	1	0x7D	6
0	0	0	0	1	1	1	0x07	7
1	1	1	1	1	1	1	0x7F	8
1	1	0	1	1	1	1	0x6F	9

แสดงข้อมูลที่ส่งออกพอร์ตของหลอดแสดงผล LED 7-Segmentแบบต่อแคโทดร่วม (Common Cathode)

2.6 ลำโพงแฉ่งเตื่อนและไฟแฉ่งเตื่อน

เป็นลำโพงแบบแม่เหล็กหรือ แบบเพียโซที่มีวงจรกำเนิดความถี่ (Oscillator) อยู่ภายในตัว เมื่อป้อนแรงดันสามารถกำเนิดเสียงได้ด้วยตัวเอง แต่ไม่สามารถเปลี่ยนความถี่ของเสียงได้ Buzzer สามารถส่งเสียงได้โดยการสั่นสะเทือนที่เป็นจังหวะความถี่ เกิดเป็นคลื่นเสียง ปกติเสียงที่มนุษย์ได้ยินนั้นจะอยู่ที่ย่านความถี่ 20 Hz ถึง 20 KHz ได้แสดงไว้ดังรูปที่ 2.14



รูปที่ 2.14 Buzzer แบบ diaphragm –Peizo [11]

2.6.1 คุณสมบัติของลำโพงแฉ่งเตื่อน

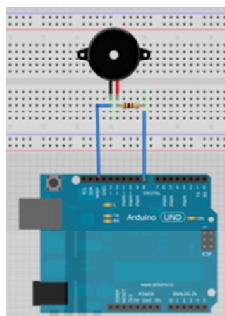
การแสดงสถานะในรูปแบบไฟสัญญาณเป็นระดับเสียงแฉ่งเตื่อนของระบบความปลอดภัย ในอุปกรณ์ที่มีการทำงานอยู่ ทำให้ผู้ที่ปฏิบัติทราบถึงระบบที่ทำงานอยู่ว่ามีความปลอดภัย ดังนั้น เพื่อในการเลือกใช้งานจึงต้องทราบถึงคุณสมบัติของบัสเซอร์ที่ได้แสดงในตารางต่อไปนี้

ตารางที่ 2.3 คุณสมบัติของบัสเซอร์[14]

Spec	Min	Typical values	Max	Unit
Power supply voltage VCC	4	5	8	V
Current	-	-	32	mA
High input	3	-	5	V
High (@ 5V) input current	-	-	0.44	mA
Low-level input	-0.3	-	+0.7	V
Volume	85	-	-	dB
Sound frequencies	2300 ± 300			Hz
Size	24X24X11			mm
Weight	4			g

2.6.2 การต่อขาเข้ากับ Arduino

การทำงานของบัสเซอร์ ทำงานเมื่อได้รับแรงดัน 5 V ที่ขา SIG โดยให้เสียงสูง ใช้งานได้กับ Microcontroller ทุกชนิด และ RPi ได้แสดงดังรูป 2.15

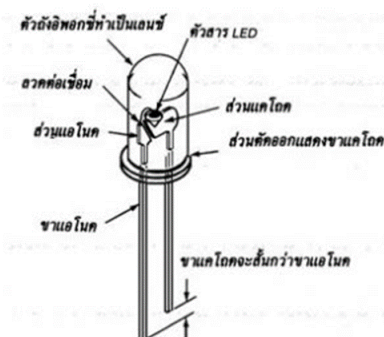


รูปที่ 2.15 การต่อขาเข้ากับ Arduino [12]

ขา ลบ (-) ของ buzzer ต่อกับ GND ของ Arduino ขา บวก (+) ของ buzzer ต่อกับ R 100 ohm และขาของ R อีกข้างต่อเข้ากับ pin 8 ของ Arduino การที่ขาบวกต่อเข้ากับตัวต้านทานนั้นก็เพื่อเป็นการลดปริมาณกระแสที่ไหลเข้าไปเพื่อมิให้บัสเซอร์เกิดความเสียหาย

2.7 หลอด LED

ไดโอดเปล่งแสง (อังกฤษ: Light-Emitting Diode หรือย่อว่า LED) เป็นอุปกรณ์สารกึ่งตัวนำอย่างหนึ่ง จัดอยู่ในจำพวกไดโอด ที่สามารถเปล่งแสงในช่วงสเปกตรัมแคบ เมื่อถูกไบอัสทางไฟฟ้าในทิศทางไปข้างหน้า ปรากฏการณ์นี้อยู่ในรูปของ Electroluminescence ซึ่งเป็นปรากฏการณ์ที่เปลี่ยนอิเล็กตรอนกลายเป็นโฟตอนโดยตรงและปราศจากความร้อน แกนกลางของ LED จะประกอบด้วยขั้วสองขั้วคือขั้ว P (Positive) และขั้ว N (Negative) และมี Junction คั่นอยู่ระหว่างขั้วทั้งสองเมื่อ LED ได้รับแรงดันไฟฟ้าที่สูงพออิเล็กตรอนจะกระโดดจากขั้ว P ไป N และปลดปล่อยโฟตรอนออกมา ทำให้เกิดการเปล่งแสงออกมา สีของแสงที่เปล่งออกมานั้นขึ้นอยู่กับองค์ประกอบทางเคมีของวัสดุที่นำมาใช้ และเปล่งแสงได้ใกล้ช่วงอัลตราไวโอเล็ต ช่วงแสงที่มองเห็น และช่วงอินฟราเรดที่ตาคนมองไม่เห็นต้องใช้ทรานซิสเตอร์มาเป็นตัวรับแสงแทน ส่วนประกอบของหลอด LED ได้แสดงไว้ดังรูปที่ 2.16



รูปที่ 2.16 ส่วนประกอบของ LED [13]



รูปที่ 2.17 สัญลักษณ์ของ LED ในวงจรไฟฟ้า [13]

LED มีหลายขนาดแต่ที่พบได้บ่อยจะเป็นขนาด 3 และ 5 มิลลิเมตร แกนของ LED มีลักษณะเป็นลูกบาศก์ ประกอบด้วยสารกึ่งตัวนำที่เป็นขั้วบวกและลบ (P N junction) ที่ทำหน้าที่เปล่งแสงออกมา ชนิดของสารกึ่งตัวนำจะเป็นตัวกำหนดสี และทั้งหมดจะอยู่บนถ้วยที่ทำหน้าที่สะท้อนแสงให้พุ่งไปที่โคมที่ทำจากอีพอกซีใสที่ใช้เป็นเลนส์ โคมจะทำหน้าที่เป็นเลนส์ โดยใช้ระยะจากถ้วยถึงโคมในการกำหนดระยะโฟกัส จากสัญลักษณ์ของ LED ในวงจรไฟฟ้า ขา A หรือที่เรียกว่าขา อนโอด โดยขานี้จะเป็นไฟบวก (+) ขา K หรือที่เรียกว่า ขา แคโทด โดยขานี้จะเป็นไฟลบ (-) แรงดันที่เราจะใช้ให้ LED เปล่งแสงได้จะอยู่ที่ประมาณ 1.5 - 3 โวลต์ โดยอาจจะขึ้นอยู่กับสีและคุณสมบัติเฉพาะตัวนั้นๆ โดยทั่วไปจะใช้ที่ 2.5 - 3 โวลต์ และ LED จะมีกระแสไหลผ่าน(กระแสไบอัสตรง) ได้ประมาณ 20 mA (มิลิแอมป์)

การนำ LED มาใช้งาน

LED นั้นต่างจากหลอดไฟชนิดอื่น โดยหลอด LED จะทำงานด้วยกระแสซึ่งจะกำหนดแรงดันเริ่มต้น และแรงดันที่ทนได้สูงสุดมาให้หากให้แรงดันเกินกว่ากำหนด จะทำให้เกิดความ

เสียหายแก่อุปกรณ์ LED ได้ จึงต้องใช้ค่าความต้านทานที่เหมาะสมเพื่อลดแรงดันที่จะทำให้ LED เสียหาย การคำนวณค่าความต้านทานที่เหมาะสมสำหรับวงจรอนุกรม

$$R_{\text{series}} = (V - V_f) / I_f \quad (2.2)$$

เมื่อ R_{series} เป็นค่าของตัวต้านทานหน่วยเป็น ohms
 V แรงดันจากแหล่งจ่ายไฟ
 V_f แรงดันตกคร่อม LED
 I_f เป็นกระแสของ LED

ตัวอย่างการคำนวณพื้นฐาน ในที่นี้เราจะให้ LED มีแรงดันตกคร่อม 2V และมีกระแสไหลผ่านตัวมันได้ 20 mA การคำนวณค่าตัวต้านทานที่มาต่อกับ จะได้ว่า ค่าความต้านทาน = (แรงดันแหล่งจ่าย - แรงดันตกคร่อม LED) / 0.02 (0.02 คือ 20mA)

เมื่อแหล่งจ่าย 5 V จะได้ว่า $R = (5 - 2) / 0.02 = 150$ คือใช้ ตัวต้านทาน 150 โอห์ม

เมื่อแหล่งจ่าย 9 V จะได้ว่า $R = (9 - 2) / 0.02 = 350$ คือใช้ ตัวต้านทาน 350 โอห์ม

เมื่อแหล่งจ่าย 12 V จะได้ว่า $R = (12 - 2) / 0.02 = 500$ คือใช้ ตัวต้านทาน 500 โอห์ม

แหล่งจ่าย ค่าความต้านทาน (โอห์ม)

3 V 100 - 200 Ω

5 V 150 - 250 Ω

9 V 350 - 450 Ω

12 V 500 - 1 K Ω

2.8 สรุปท้ายบท

การใช้งานคอมพิวเตอร์และคอมพิวเตอร์พกพาในปัจจุบันมีการใช้งานกันอย่างแพร่หลายในทุกวงการทั้งเด็กและผู้ใหญ่ เมื่อมีการใช้งานติดต่อกันเป็นระยะเวลานานย่อมส่งผลกระทบต่อสายตา เพื่อเป็นทางเลือกหนึ่งในการถนอมสายตาจึงได้สร้างอุปกรณ์ต้นแบบคือ เซนเซอร์แจ้งเตือนระยะปลอดภัยสำหรับสายตาในการทำงานกับคอมพิวเตอร์ ซึ่งมีองค์ประกอบสำคัญคือ ไมโครคอนโทรลเลอร์ เซนเซอร์วัดระยะทาง Ultrasonic distance detector module รุ่น US-015 ตัวเลขแสดงผล 7-segment ลำโพงแจ้งเตือน (Buzzer) และ LED

บทที่ 3

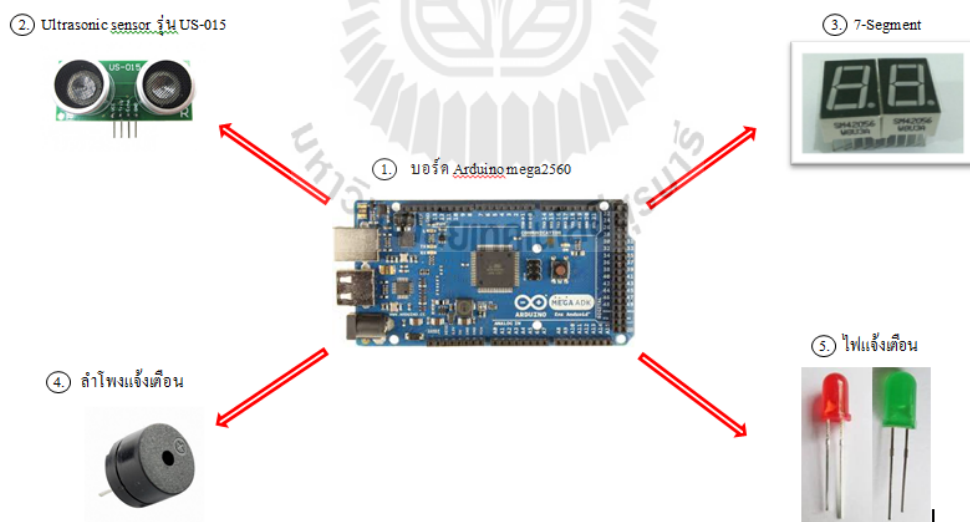
การออกแบบระบบต้นแบบ

3.1 กล่าวนำ

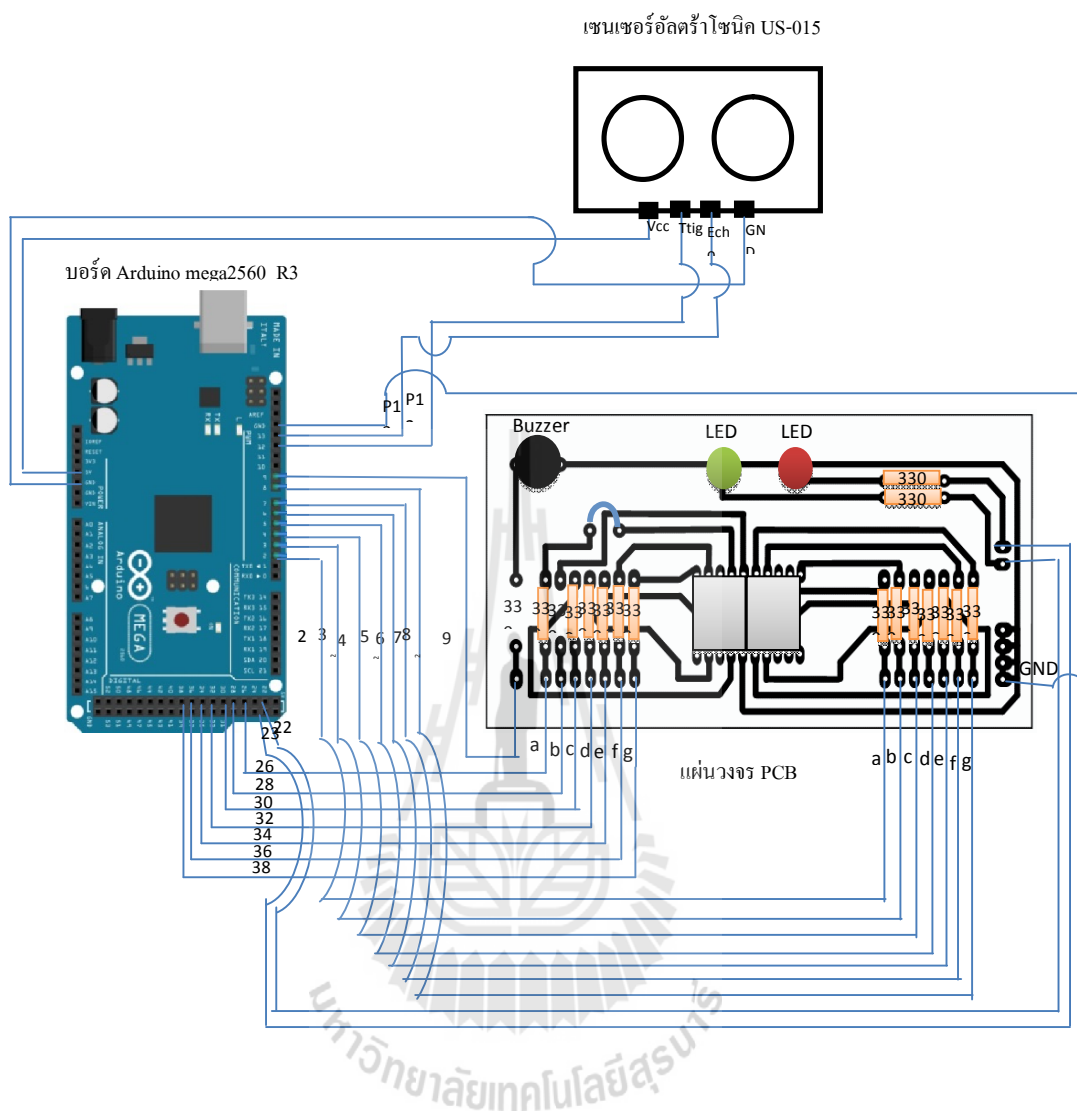
เนื้อหาในบทนี้จะกล่าวถึงการออกแบบและการนำอุปกรณ์แต่ละตัวมาประกอบเป็นอุปกรณ์วัดระยะแจ้งเตือน ซึ่งจะอัปโหลดชุดโปรแกรมลงบอร์ด Arduino MEGA2560 R3 เพื่อให้อุปกรณ์ทำงานตามต้องการ อุปกรณ์โดยรวมระบบของเซนเซอร์แจ้งเตือนระยะปลอดภัยสำหรับสายตาในการใช้งานกับคอมพิวเตอร์ มีอยู่ทั้งหมด 5 ส่วนด้วยกัน คือ เซนเซอร์วัดระยะทาง บอร์ด Arduino MEGA2560 R3 อุปกรณ์ 7-Segment ลำโพงแจ้งเตือนและไฟแจ้งเตือน

3.2 ส่วนประกอบของอุปกรณ์ต้นแบบ

สำหรับส่วนประกอบของอุปกรณ์ต้นแบบที่เลือกใช้จะแสดง ดังรูปที่ 3.1 ซึ่งอุปกรณ์ต้นแบบเป็นการทำงานร่วมกันของ บอร์ด Arduino MEGA2560 R3 เซนเซอร์อัลตราโซนิก 7-segment ลำโพงแจ้งเตือนและหลอดไฟแอลอีดี โดยมีการเชื่อมต่ออุปกรณ์ดังรูปที่ 3.2



รูปที่ 3.1 ส่วนประกอบของอุปกรณ์ต้นแบบ



รูปที่ 3.2 การเชื่อมต่ออุปกรณ์ต้นแบบ

การเชื่อมต่ออุปกรณ์ต้นแบบ

1. เซนเซอร์อัลตราโซนิกกับบอร์ด Arduino MEGA2560 R3

- ขา Trig ของเซนเซอร์อัลตราโซนิก ต่อกับ ขา digital 12 ของบอร์ด Arduino MEGA2560 R3

- ขา Echo ของเซนเซอร์อัลตราโซนิก ต่อกับ ขา digital 13 ของบอร์ด Arduino MEGA2560 R3

- ขา Vcc ของเซนเซอร์อัลตราโซนิก ต่อกับ ขา 5v ของบอร์ด Arduino MEGA2560 R3

- ขา GND ของเซนเซอร์อัลตราโซนิก ต่อกับ ขา GND ของบอร์ด Arduino MEGA2560 R3

2. แผ่น PCB กับบอร์ด Arduino MEGA2560 R3

● 7-segment กับบอร์ด Arduino MEGA2560 R3

-ขา a หลักสิบของ 7-segment ต่อกับ ขา digital 26 ของบอร์ด Arduino MEGA2560 R3

- ขา b หลักสิบของ 7-segment ต่อกับ ขา digital 28 ของบอร์ด Arduino MEGA2560 R3

- ขา c หลักสิบของ 7-segment ต่อกับ ขา digital 30 ของบอร์ด Arduino MEGA2560 R3

- ขา d หลักสิบของ 7-segment ต่อกับ ขา digital 32 ของบอร์ด Arduino MEGA2560 R3

- ขา e หลักสิบของ 7-segment ต่อกับ ขา digital 34 ของบอร์ด Arduino MEGA2560 R3

- ขา f หลักสิบของ 7-segment ต่อกับ ขา digital 36 ของบอร์ด Arduino MEGA2560 R3

- ขา g หลักสิบของ 7-segment ต่อกับ ขา digital 38 ของบอร์ด Arduino MEGA2560 R3

- ขา a หลักหน่วยของ 7-segment ต่อกับ ขา digital 2 ของบอร์ด Arduino MEGA2560 R3

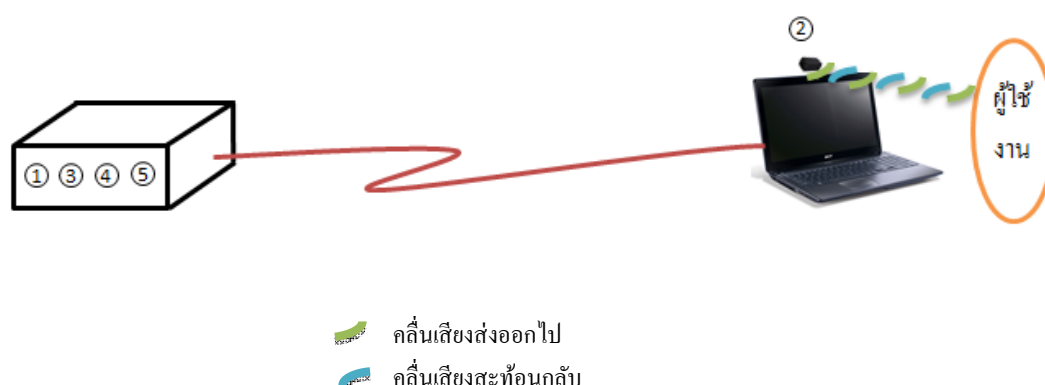
- ขา b หลักหน่วยของ 7-segment ต่อกับ ขา digital 3 ของบอร์ด Arduino MEGA2560 R3

- ขา cหลักหน่วยของ 7-segment ต่อกับ ขา digital 4ของบอร์ด Arduino MEGA2560 R3
- ขา dหลักหน่วยของ 7-segment ต่อกับ ขา digital 5ของบอร์ด Arduino MEGA2560 R3
- ขา eหลักหน่วยของ 7-segment ต่อกับ ขา digital 6ของบอร์ด Arduino MEGA2560 R3
- ขา fหลักหน่วยของ 7-segment ต่อกับ ขา digital 7ของบอร์ด Arduino MEGA2560 R3
- ขา gหลักหน่วยของ 7-segment ต่อกับ ขา digital 8ของบอร์ด Arduino MEGA2560 R3
- ลำโพงแฉ่งเตือนกับบอร์ด Arduino MEGA2560 R3
 - ขา บวกต่อกับ ขา digital 9ของบอร์ด Arduino MEGA2560 R3
 - ขา ลบ ต่อกับ ขา GNDของบอร์ด Arduino MEGA2560 R3 (แผ่น PCB ทำการเชื่อมต่อ GND เป็นจุดเดียวกันแล้ว)
- หลอดไฟ LED กับบอร์ด Arduino MEGA2560 R3
 - ขา บวก ของหลอดไฟ LED สีเขียว ต่อกับขา digital 22 ของบอร์ด Arduino MEGA2560 R3
 - ขา บวก ของหลอดไฟ LED สีแดง ต่อกับขา digital 23 ของบอร์ด Arduino MEGA2560 R3
 - ขา GND ของหลอดไฟ LED ทั้งสองได้เชื่อมต่อเป็นจุดเดียวเพื่อต่อกับขา GND ของบอร์ด Arduino MEGA2560 R3

3.3 หลักการทำงานของอุปกรณ์ต้นแบบ

การทำงานของอุปกรณ์ต้นแบบนี้จะเห็นได้จากรูปที่ 3.3 ที่แสดงการจำลองการใช้งานอุปกรณ์ต้นแบบโดยใช้บอร์ด Arduino MEGA2560 R3 บอร์ดนี้ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATmega2560 ในการควบคุมชุดฮาร์ดแวร์ (Hardware) โดยมีส่วนประกอบของอุปกรณ์ต้นแบบตามรูปที่ 3.1 การทำงานของอุปกรณ์ต้นแบบที่อัปโหลดชุดคำสั่งลงบอร์ดเรียบร้อยแล้ว (ขั้นตอนการติดตั้ง อัปโหลดโปรแกรมและชุดโปรแกรมคำสั่งสามารถดูได้จากผนวก) โดยหมายเลข 2 ทำการวัดค่าระยะทางส่งไปหมายเลข 1 เพื่อทำการประมวลผลควบคุมการทำงานของอุปกรณ์ต้นแบบ โดยมีรายละเอียดดังนี้

เมื่อเชื่อมต่อสาย USB ที่พอร์ตอนุกรมของคอมพิวเตอร์พกพาหรือแหล่งจ่ายภายนอกที่ให้ไฟเลี้ยงให้กับบอร์ด Arduino ที่ 5v บอร์ด Arduino จะสามารถทำงานได้พร้อมมีสัญญาณไฟแจ้งสีเขียวติดตลอดที่บอร์ดกำลังทำงานอยู่ (ซึ่งสัญญาณไฟแจ้งเตือนสามารถบ่งบอกว่าวงจรทำงานอยู่) และเซนเซอร์จะทำการส่งคลื่นเสียงออกมาตลอดเวลา เมื่อมีผู้ใช้งานคอมพิวเตอร์คลื่นเสียงก็จะกระทบกับผู้ใช้จนแล้วสะท้อนกลับ เพื่อไปประมวลผลระยะทางที่ได้แล้วส่งไปแสดงผลที่ 7-Segment เมื่อผู้ใช้งานเริ่มเข้าใกล้หน้าจอคอมพิวเตอร์ที่มีระยะทางน้อยกว่า 50 เซนติเมตร ลำโพงแจ้งเตือนจะส่งเสียงดังพร้อมสัญญาณไฟแจ้งเตือนสีแดงที่จะกระพริบพร้อมเสียงแจ้งเตือน โดยมีช่วงระยะการส่งเสียงคือ เสียงที่ 1 ช่วงระยะทาง 40-49 เซนติเมตร เสียงที่ 2 ช่วงระยะทาง 30-39 เซนติเมตร เสียงที่ 3 ช่วงระยะทาง 20-29 เซนติเมตร เสียงที่ 4 ช่วงระยะทาง 10-19 เซนติเมตร เสียงที่ 5 ช่วงระยะทางที่น้อยกว่า 10 เมื่อผู้ใช้งานถอยห่างจากระยะหน้าจอของคอมพิวเตอร์ซึ่งมีระยะทางมากกว่า 50 เซนติเมตร ลำโพงแจ้งเตือนจะไม่ส่งเสียงดังและไฟสัญญาณแจ้งเตือนจะดับได้แสดงแบบจำลองการใช้งานของอุปกรณ์ต้นแบบดังรูปที่ 3.3

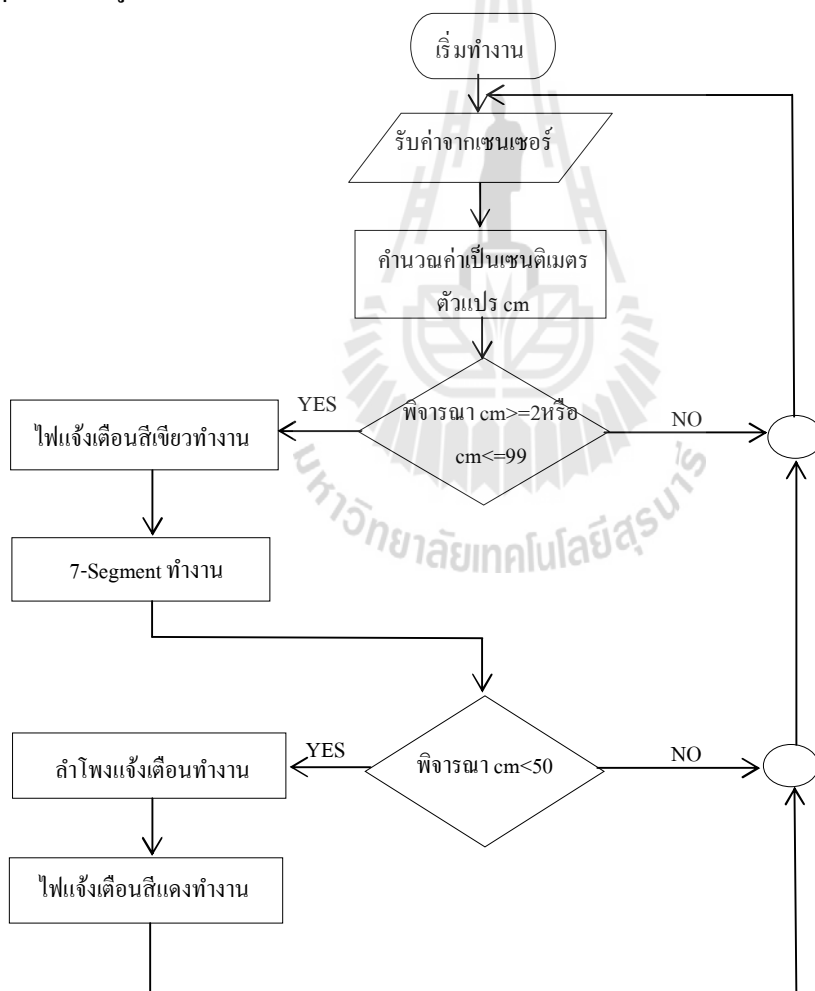


รูปที่ 3.3 จำลองการใช้งานของอุปกรณ์ต้นแบบ

3.3.1 แผนผังการทำงาน

1. เริ่มการทำงาน
2. รับค่าจากเซนเซอร์ เพื่อนำไปคำนวณ
3. คำนวณค่าที่ได้รับได้จากเซนเซอร์เป็นหน่วย เซนติเมตร
4. ประมวลผลระยะทาง โดยกำหนดให้ระยะทางที่รับได้อยู่ในช่วง 0-99เซนติเมตร เพื่อให้สัญญาณไฟแจ้งเตือนสีเขียวทำงาน ถ้าไม่อยู่ในช่วงนี้กลับไปทำข้อที่ 2
5. แสดงผลที่ได้รับจากการประมวลผลที่ 7-Segment และนำค่าที่ได้ไปพิจารณาถ้าระยะทางน้อยกว่า 50 เซนติเมตร สั่งให้ลำโพงแจ้งเตือนพร้อมกับสัญญาณไฟแจ้งเตือนสีแดงทำงาน
6. กลับไปทำข้อที่ 2

จะสามารถเขียนเป็นแผนภาพแสดงความคิดในการเขียนชุดโปรแกรมสำหรับควบคุมอุปกรณ์ดังรูปที่ 3.4



รูปที่ 3.4 โฟลว์ชาร์ตการทำงานของโปรแกรม

3.4 การใช้โปรแกรม Arduino IDE

Arduino คือ โครงการที่นำชิปไอซีไมโครคอนโทรลเลอร์ตระกูลต่างๆมาใช้ร่วมกันในภาษา C ซึ่งภาษา C นี้เป็นลักษณะเฉพาะ คือ มีการเขียนไบบารีของ Arduino ขึ้นมาเพื่อสั่งงานไมโครคอนโทรลเลอร์ที่ต่างกัน สามารถใช้งานโค้ดตัวเดียวกันได้ โดยตัวโครงการได้ออกแบบบอร์ดทดลองมาหลายรูปแบบ เพื่อใช้งานกับ IDE ของตนเอง สาเหตุหลักที่ทำให้ Arduino เป็นที่นิยมมากเป็นเพราะซอฟต์แวร์ที่ใช้งานร่วมกันสามารถโหลดได้ฟรี จากที่กล่าวไปแล้ว Arduino นั้นได้ใช้ชิป AVR เป็นหลักใน Arduino แทบทุกรุ่น สาเหตุมาจากไมโครคอนโทรลเลอร์ของตระกูล AVR นั้นมีความทันสมัยในชิปบางตัวสามารถเชื่อมต่อผ่าน USB ได้โดยตรง สามารถใช้กับคอมพิวเตอร์สมัยใหม่ได้อย่างดีและในไมโครคอนโทรลเลอร์ตระกูล AVR ยังมีส่วนของโปรแกรมพิเศษที่เรียกว่า Bootloader อยู่ในระดับล่างกว่าปกติ ซึ่งจะเป็นส่วนโปรแกรมที่จะถูกเรียกขึ้นมา ก่อนการเรียกโปรแกรมปกติ ทำให้สามารถเขียนสั่งให้ทำงานใดๆก็ได้ ก่อนการเรียกโปรแกรมปกติ ทำให้ Arduino นั้นอาศัยส่วนโปรแกรมพิเศษนี้ในการทำให้ชิปสามารถโปรแกรมผ่านพอร์ตอนุกรมชนิด UART ได้ จึงทำให้การเขียนโปรแกรมลงไปในชิปใช้เพียง USB to UART ก็เพียงพอแล้ว แต่การโปรแกรมด้วยการใช้โปรโตคอล UART ก็มีข้อเสียตรงที่ใช้เวลานานในการบูตเข้าโปรแกรมปกติประมาณ 1-2 วินาที

3.4.1 วิธีการดาวน์โหลด Arduino IDE

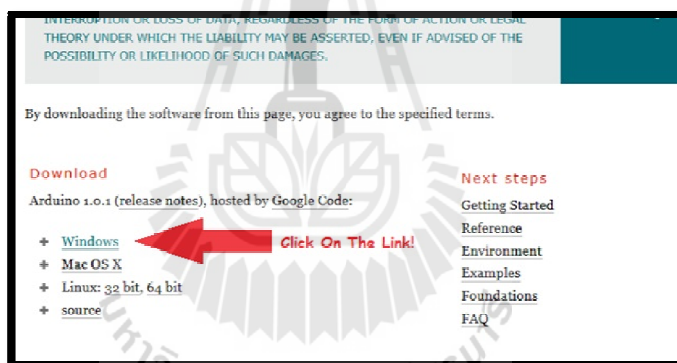
สำหรับวิธีการดาวน์โหลด MCU ของ Atmel ในกลุ่มเบอร์ที่ใช้วิธีการดาวน์โหลดโปรแกรมแบบ ISP เช่น ATmega328 ATmega32u4 AT91SA3X8E ATmega32v4 และ ATmega2560 เป็นต้น เบอร์ที่กล่าวมานี้ใช้โปรแกรม Arduino ได้ ในการดาวน์โหลดจะมี Driver สำหรับลง Max OS X /Linux 32 bit /Linux 64 bit และ Windows ในที่นี้ขออ้างการดาวน์โหลด Driver สำหรับ Windows ดังนี้

1. ดาวน์โหลดไปที่ [12] (เว็บสำหรับดาวน์โหลด) แล้วกดคลิกที่ Download เพื่อเข้าสู่หน้าดาวน์โหลดดังรูปที่ 3.5



รูปที่ 3.5 หน้าเว็บสำหรับดาวน์โหลด [15]

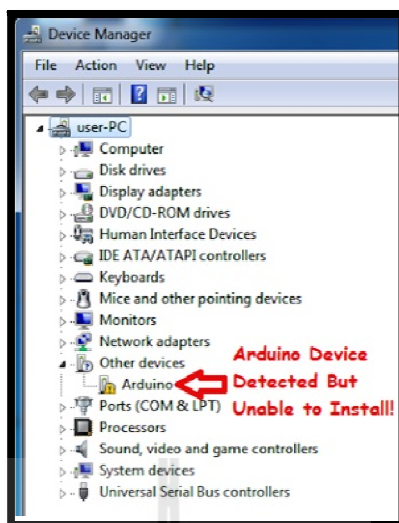
2. เมื่อเข้ามาหน้าดาวน์โหลดให้คลิกที่ Windows link เพื่อทำการดาวน์โหลด Arduino ซอฟต์แวร์สำหรับระบบปฏิบัติการ Windows ตามรูปที่ 3.6



รูปที่ 3.6 การเลือก Arduino ซอฟต์แวร์ [15]

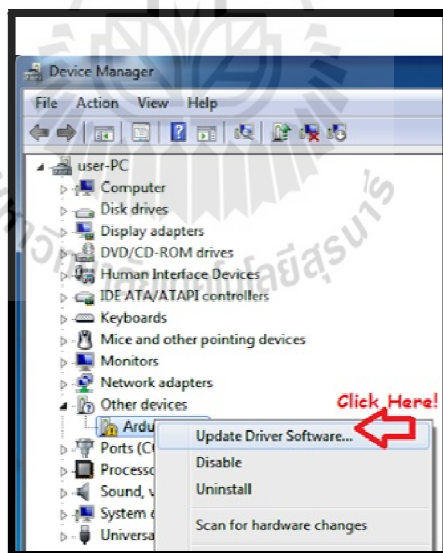
4. ติดตั้งไดรเวอร์ของ Windows Arduino โดยทำการเสียบบอร์ด Arduino MEGA2560 R3 โดยใช้สาย USB เชื่อมต่อที่ UART ของคอมพิวเตอร์ Windows จะพยายาม Install driver แต่สุดท้ายไม่สำเร็จ

5. คลิกที่ Start → Control panel → Device Manager จะแสดง Arduino Device ที่ทำการ connect ไว้ (ขึ้นอยู่กับว่าใช้บอร์ดชนิดไหนชื่อก็จะแสดงขึ้นมาให้เห็น) จะเป็นเครื่องหมายตกใจสีเหลือง ซึ่งแสดงว่าอุปกรณ์ไม่สามารถทำการ Install ได้ ดังรูปที่ 3.7



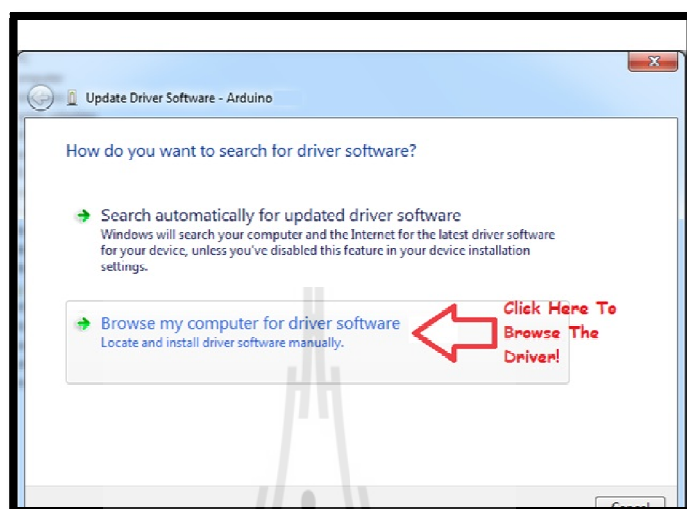
รูปที่ 3.7 การเชื่อมต่อของบอร์ดไม่สำเร็จ [15]

6. ทำการติดตั้งไดรเวอร์ในหน้าต่างของ Device Manager ให้คลิกขวาที่ Arduino board แล้วคลิก Update Driver Software บน pop-up ดังรูปที่ 3.8



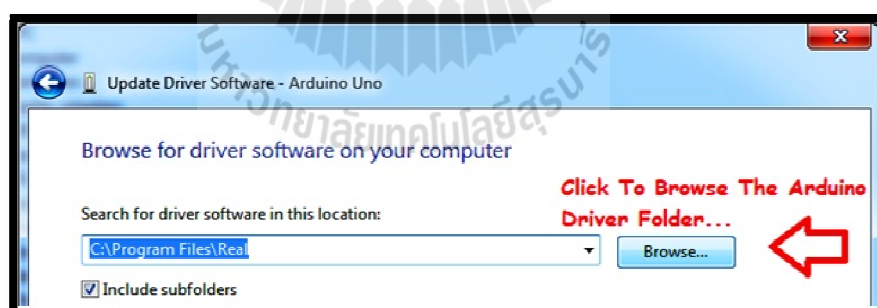
รูปที่ 3.8 การเลือก Update Driver Software [15]

7. หลังจากนั้นจะมีกล่องขึ้น pop-up มาโชว์ว่า Update Driver Software ให้คลิกเลือกที่ Browse my computer for driver software เพื่อจะติดตั้งซอฟต์แวร์ ดังรูปที่ 3.9



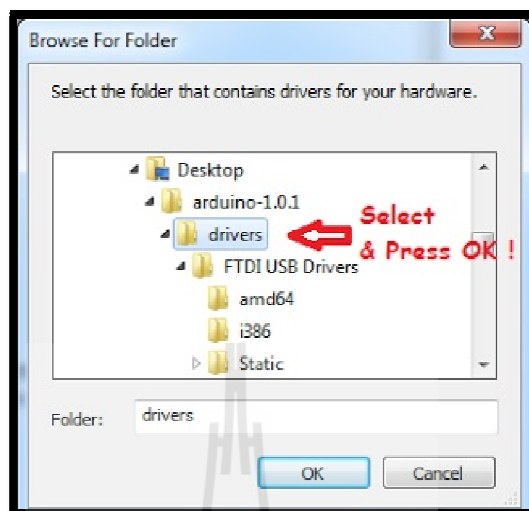
รูปที่ 3.9 การเลือกติดตั้งซอฟต์แวร์[15]

8. คลิก Browse... เพื่อเลือกตำแหน่งจัดเก็บ Program File ดังรูปที่ 3.10



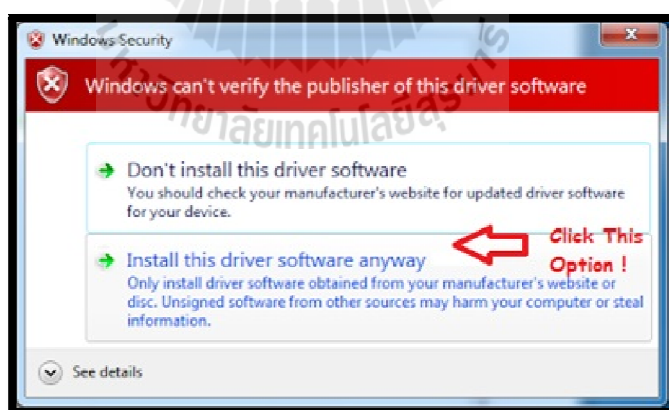
รูปที่ 3.10 การเลือกตำแหน่งจัดเก็บไฟล์โปรแกรม[15]

9. เลือกที่ drivers folder ใน Arduino folder ที่ดาวน์โหลดมา คลิก OK จากนั้นคลิก Next ดังรูปที่ 3.11



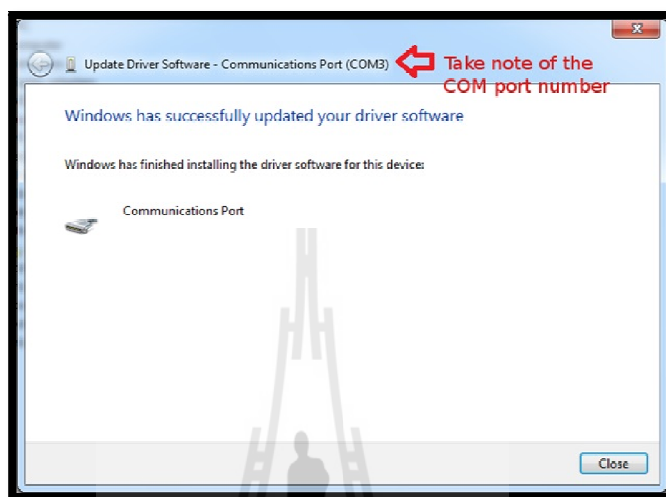
รูปที่ 3.11 การเลือกโฟลเดอร์ไดรเวอร์ [15]

10. จะมีกล่อง pop-up ขึ้นมาให้คลิก Install this software anyway เพื่อดำเนินการไดรเวอร์ ต่อ ดังรูปที่ 3.12



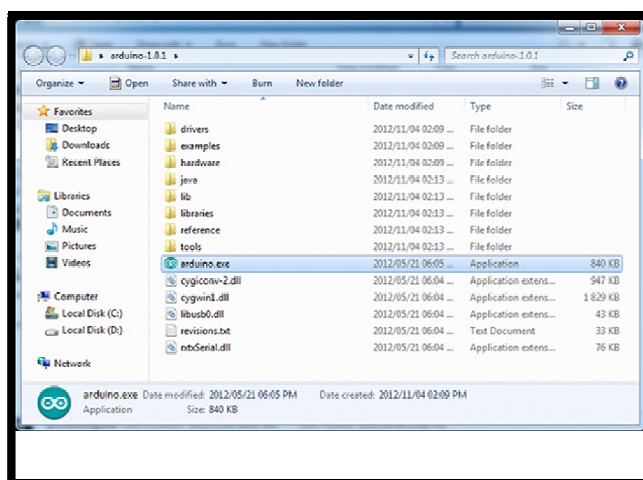
รูปที่ 3.12 การเลือก Install this driver software anyway [15]

11. เมื่อทำการ Install Driver เป็นที่เรียบร้อยแล้วจะมี dialog box ตามรูปที่ 3.13ด้านล่าง เราต้องทราบ port number ที่บอร์ด Arduino ทำการเชื่อมต่ออยู่ในรูปที่ 3.13นี้คือ port COM3 แต่ port ที่ใช้งานจริงๆอาจจะเป็น port COM อื่นๆก็ได้ขึ้นกับการเชื่อมต่อพอร์ตอนุกรมจากนั้นคลิก close



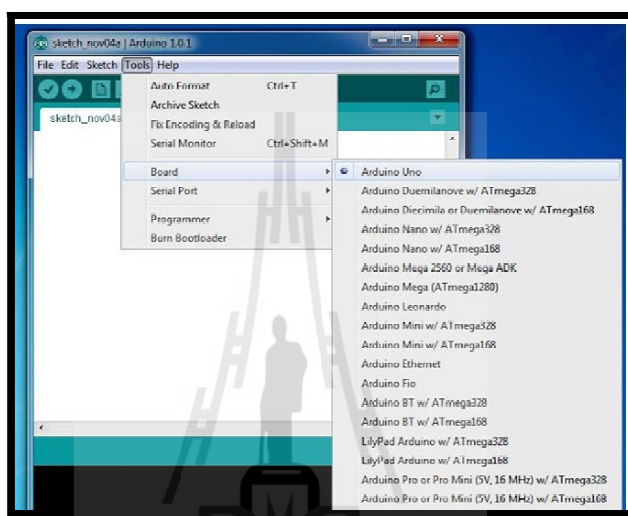
รูปที่ 3.13การ Install Driver สำเร็จ[15]

12. ทำการ set up Arduino Software หลังจาก Installation แล้วการ set นี้ ทำได้แค่ครั้งเดียว นอกจากต้องการเปลี่ยนรุ่นของบอร์ดหรือเปลี่ยน port ที่ทำการเชื่อมต่อกับบอร์ดเท่านั้น เลือกไปที่โฟลเดอร์ที่จัดเก็บไว้เลือก Arduino IDE ไว้แล้วทำการ startSoftwareของ Arduino IDE โดยทำการดับเบิ้ลคลิกไปที่ Arduino application ดังรูปที่ 3.14



รูปที่ 3.14 การ set up Arduino Software [15]

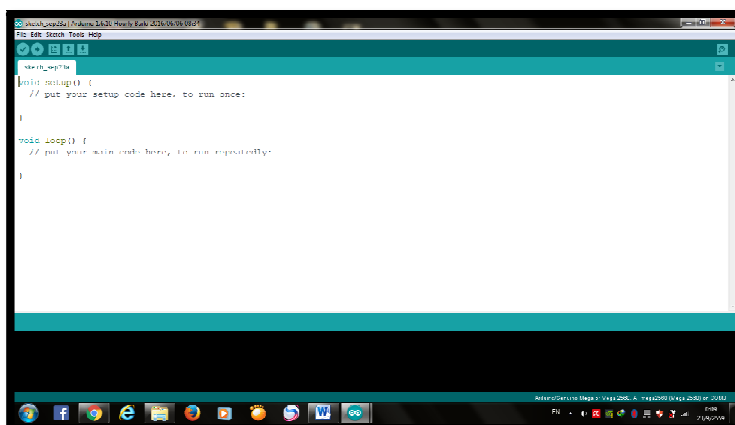
13. เลือกรุ่นของบอร์ด Arduino ที่ทำการเชื่อมต่อให้ถูกต้องใน list ในรูปเลือกบอร์ด Arduino UNO และทำการเลือก serial port ที่ต่อให้ถูกต้องเลือกที่ computer serial port ที่ Arduino ทำการเชื่อมต่ออยู่ดังรูปที่ 3.15



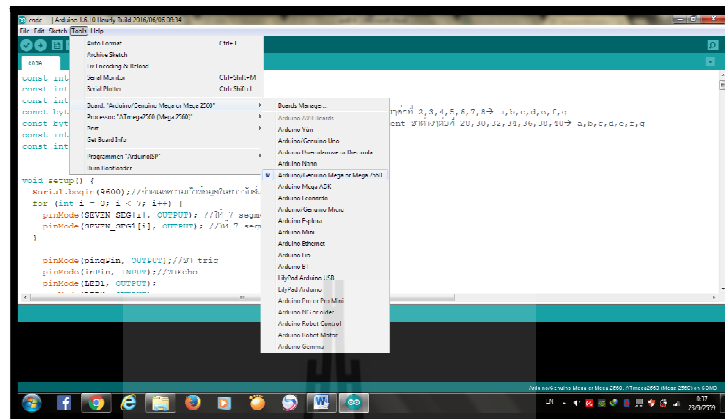
รูปที่ 3.15 การเลือกรุ่นของบอร์ด Arduino [15]

3.4.2 การอัปโหลดไฟลงบอร์ด Arduino MEGA2560 R3

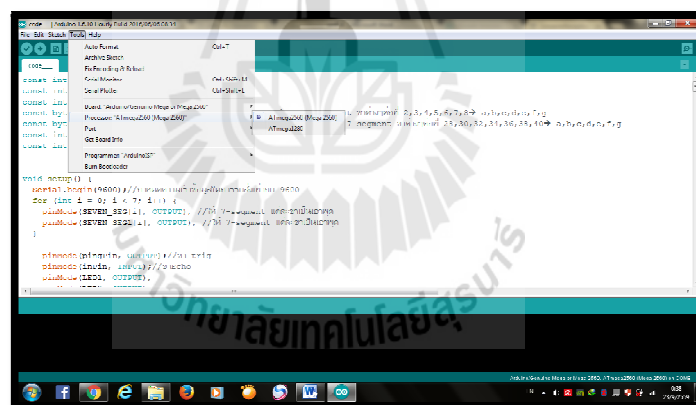
1. เปิดโปรแกรม Arduino จะได้หน้าต่างดังรูปที่ 3.16



5. เลือกประเภทของบอร์ด Arduino คลิก Tools Board Arduino/Genuino Mega or Mega 2560 ดังรูปที่ 3.19 จากนั้นเลือก Processor ตามรูปที่ 3.20

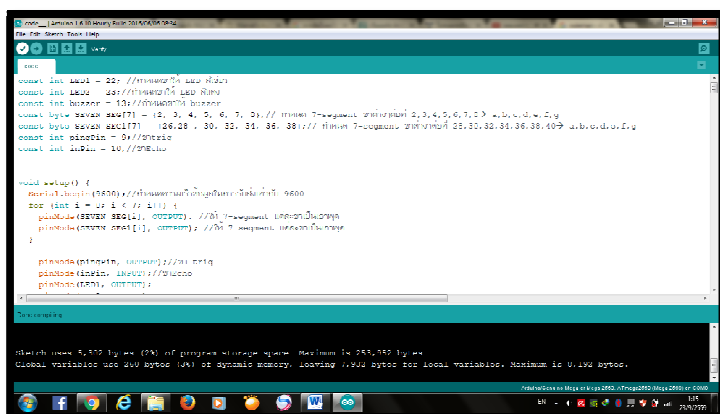


รูปที่ 3.19 การเลือกประเภทของบอร์ด Arduino



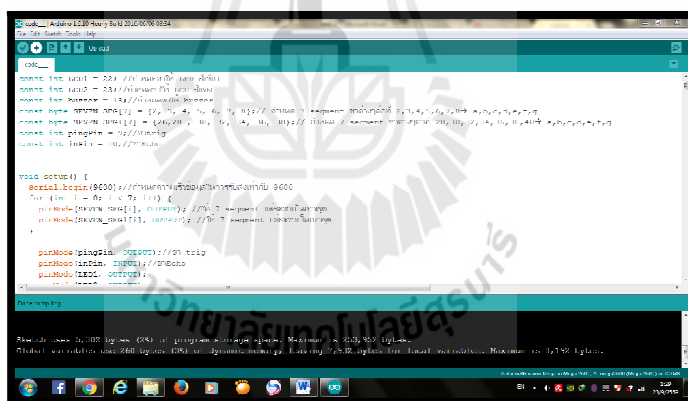
รูปที่ 3.20 การเลือก Processor

6. ทำการตรวจสอบ Code คลิกที่  Verify โปรแกรมทำการ Compiling ของ Code หากโค้ดไม่มีบรรทัดไหนพิมพ์ผิดที่ฝั่งซ้ายและด้านล่างของโปรแกรมจะขึ้น Done compiling ดังรูปที่ 3.21



รูปที่ 3.21 การตรวจสอบ Code

7. อัปโหลดชุดโปรแกรมลงบอร์ด Arduino MEGA2560 R3 คลิกที่  Upload ดังรูปที่ 3.22



รูปที่ 3.22 อัปโหลดโปรแกรมลงบอร์ด

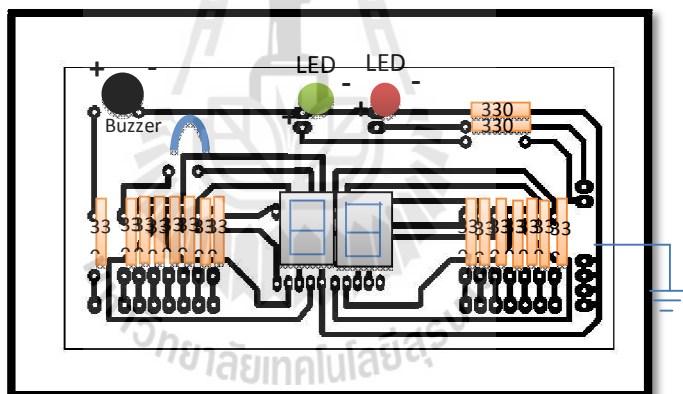
3.5 การสร้างอุปกรณ์ต้นแบบ

1. เตรียมอุปกรณ์หลักที่ใช้ในการสร้างอุปกรณ์ต้นแบบมีดังนี้

- บอร์ด Arduino MEGA2560 R3	1	บอร์ด
- เซนเซอร์อัลตราโซนิก รุ่น US-015	1	ตัว
- ลำโพงแฉ่งเตือน	1	ตัว
- หลอดไฟ LED สีเขียว	1	ตัว
- หลอดไฟ LED สีแดง	1	ตัว

2. ขั้นตอนการสร้างอุปกรณ์ต้นแบบ

- ออกแบบลายวงจรให้ได้ตามต้องการโดยใช้โปรแกรม Circuit Wizard ซึ่งโปรแกรมนี้มีความง่ายเหมาะสำหรับผู้เริ่มทำการศึกษาและแผ่นปริ้นสามารถลดจำนวนของสายไฟลงด้วย ซึ่งลายวงจรที่ใช้สำหรับประกอบอุปกรณ์บางส่วนของอุปกรณ์ต้นแบบแสดงดังรูปที่ 3.23



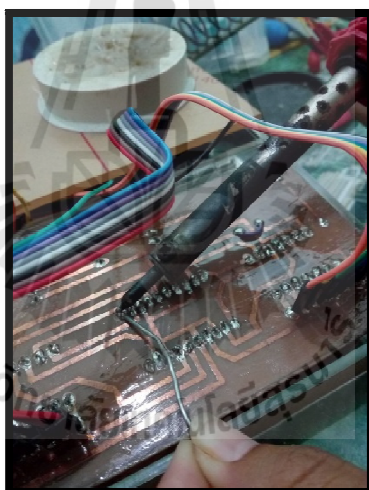
รูปที่ 3.23 ลายวงจร

- การกัดแผ่นปริ้นให้ได้ลายวงจรที่ต้องการต้องนำแผ่นปริ้นที่วาดลายวงจรด้วยปากกามีกสำหรับกัดแผ่นปริ้นหรือจะใช้วิธีการปริ้นลายวงจรแล้วใช้เตารีดรีดประมาณ 10 นาที ให้เส้นลายวงจรติดกับแผ่น PCB จากนั้นทิ้งไว้จนแผ่น PCB ไม่มีความร้อนจึงนำไปแช่น้ำยากัดปริ้น ใช้เวลาประมาณ 30 นาที เพื่อให้ได้แผ่นวงจรที่สมบูรณ์ดังรูปที่ 3.24



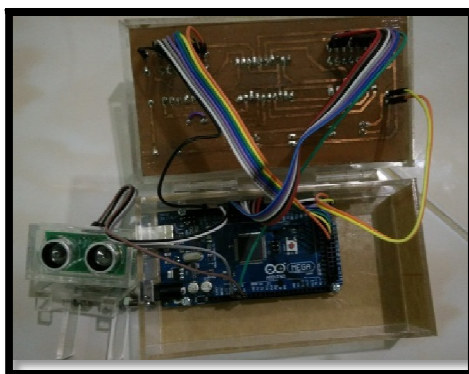
รูปที่ 3.24 กัดลายวงจร

- นำแผ่นปริ๊นที่กัดมาล้างด้วยน้ำสะอาดและอุปกรณ์ต่างๆ มาบัดกรีเพื่อให้อุปกรณ์ยึดติดกับแผ่น PCB มั่นคงและป้องกันการหลุดของอุปกรณ์ดังรูปที่ 3.25



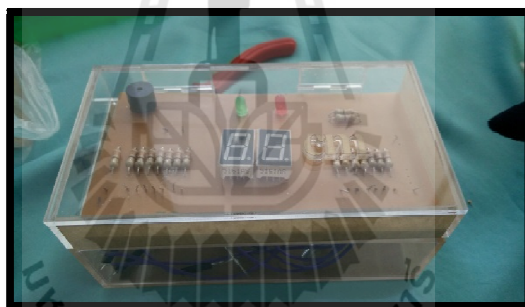
รูปที่ 3.25 บัดกรีอุปกรณ์ลงแผ่นปริ๊น

- นำสายไฟมาเชื่อมต่อระหว่างบอร์ด Arduino MEGA2560 R3 กับ แผ่นปริ๊นที่ประกอบอุปกรณ์เรียบร้อยแล้ว ดังรูปที่ 3.26



รูปที่ 3.26 การเชื่อมต่อสายระหว่างบอร์ดและลายวงจร

- นำอุปกรณ์ที่ประกอบเสร็จเรียบร้อยแล้วไปติดตั้งในกล่องอะคริลิก เพื่อความสะดวกในการใช้งานแสดงให้เห็นดังรูปที่ 3.27



รูปที่ 3.27 ติดตั้งอุปกรณ์ลงในกล่องอะคริลิก

- อุปกรณ์ต้นแบบที่สมบูรณ์แสดงดังรูปที่ 3.28



รูปที่ 3.28 อุปกรณ์ต้นแบบที่สมบูรณ์

3.6 สรุปท้ายบท

การออกแบบและสร้างอุปกรณ์ต้นแบบ จะนำแต่ละอุปกรณ์เข้ามาประกอบกันด้วยวิธีหรือเทคนิคต่างๆตามผู้ออกแบบถนัด โดยทำเป็นขั้นตอนเพื่อให้อุปกรณ์ต้นแบบมีความสมบูรณ์ถูกต้องตามที่ออกแบบไว้ จากนั้นทำการออกแบบชุดโปรแกรมสำหรับการทำงานของอุปกรณ์ต้นแบบ เพื่อใช้ในการอัปโหลดชุดโปรแกรมลงบอร์ดโดยมีเนื้อหาในบทถัดไป



บทที่ 4

การทดลองอุปกรณ์ต้นแบบ

4.1 กล่าวนำ

ในบทนี้อุปกรณ์ที่ได้ออกแบบและสร้างขึ้นมาประกอบเข้าเป็นอุปกรณ์ต้นแบบ เพื่อแจ้งเตือนระยะการใช้คอมพิวเตอร์โดยอาศัยคลื่นอัลตราโซนิก โดยทำการทดลองวัดระยะทางของเซนเซอร์อัลตราโซนิก US-015 การทดลองวัดระยะใช้งานอุปกรณ์ต้นแบบโดยผู้ใช้งานและการทดลองการแจ้งเตือนของอุปกรณ์ต้นแบบ

4.2 การทดลองอุปกรณ์ต้นแบบ

การทดสอบการทำงานของอุปกรณ์เซนเซอร์แจ้งเตือนระยะปลอดภัยสำหรับสายตาในการทำงานกับคอมพิวเตอร์ เนื่องจากต้องการทราบว่าอุปกรณ์สามารถทำงานได้มีประสิทธิภาพหรือไม่ และระยะทางมีผลต่อเซนเซอร์อย่างไร จึงได้แบ่งการทดลองอุปกรณ์เป็น 3 การทดลองได้แก่

1. ทดสอบวัดระยะของเซนเซอร์อัลตราโซนิก US-015
2. ทดสอบวัดระยะใช้งานอุปกรณ์ต้นแบบโดยผู้ใช้งานจริง
3. ทดสอบเสียงแจ้งเตือนของอุปกรณ์ต้นแบบโดยผู้ใช้งานจริง

การทดลองที่ 1

: ทดสอบวัดระยะทางของเซนเซอร์อัลตราโซนิก US-015

วัตถุประสงค์

: เพื่อทดสอบเซนเซอร์อัลตราโซนิก US-015 มีความคลาดเคลื่อนมากน้อยเพียงใด

วัสดุและอุปกรณ์

1. ชุดอุปกรณ์ต้นแบบ
2. ตลับเมตร
3. ไม้ฉาก
4. ฟิวเจอร์บอร์ดขนาด 50×65 เซนติเมตร

ขั้นตอนการทดลอง

1. นำกล่องชุดอุปกรณ์ต้นแบบต่อเข้ากับ USB ของคอมพิวเตอร์ แล้วอัปโหลดโปรแกรมสำหรับเซนเซอร์ลงบอร์ด Arduino MEGA2560 R3 (ชุดโปรแกรมแสดงในภาคผนวกข้อที่ 1.1 หน้า ที่ 68)
2. ติดตั้งเซนเซอร์ที่ขอบของหน้าจอ พร้อมกับใช้แผ่นฟิวเจอร์บอร์ดคัลเล่อร์และไม้ฉากเพื่อให้ฟิวเจอร์บอร์ดตั้งฉากกับพื้น เพื่อไม่เกิดความผิดพลาดจากผู้ทำการวัดทดสอบการวัดระยะทางแสดงดังรูปที่ 4.1



รูปที่ 4.1ทดสอบวัดระยะของเซนเซอร์อัลตราโซนิก

3. อ่านค่าจาก Serial monitor ที่โปรแกรม Arduino และบันทึกผลการทดลอง

บันทึกผลการทดลอง

จากการทดสอบวัดระยะทางของเซนเซอร์ในแต่ละครั้ง ผลการทดลองการวัดระยะทางแสดงดังตารางที่ 4.1

ตารางผลการทดลองที่ 4.1 ผลการทดลองการวัดระยะทางของเซนเซอร์อัลตราโซนิก

ระยะทาง จริง(cm)	ระยะที่วัด ได้ครั้งที่ 1 (cm)	ระยะที่วัด ได้ครั้งที่ 2(cm)	ระยะที่วัด ได้ครั้งที่ 3(cm)	ระยะที่วัด ได้ครั้งที่ 4(cm)	ระยะที่วัด ได้ครั้งที่ 5(cm)	ค่าเฉลี่ย (cm)	ค่าความ ผิดพลาด (%error)
5	5	5	5	5	5	5	0
10	10	10	10	10	10	10	0
20	20	20	20	20	20	20	0
30	30	30	30	30	30	30	0
40	40	40	40	40	40	40	0
50	50	50	50	50	50	50	0
60	60	60	60	59	60	59.8	0.33
70	70	69	69	70	69	69.4	0.85
80	79	79	79	79	78	79	1.25
90	89	89	89	89	89	89	1.11
100	99	99	99	99	99	99	1
110	109	109	109	109	109	109	0.91
120	119	119	119	119	119	119	0.83
130	128	128	127	128	128	127.8	1.69
140	137	137	137	137	137	137	2.14
150	147	147	147	147	146	146.6	2.26
160	157	156	156	157	156	156.4	2.25
170	167	167	166	166	167	166.6	2
180	177	176	176	177	176	176.4	2
190	186	187	186	186	187	186.4	1.89
200	196	196	197	196	196	196.2	1.9
300	295	296	295	295	295	295.2	1.6
400	393	393	393	393	393	393	1.75

ค่าความผิดพลาดเฉลี่ยรวม = 1.11%

วิเคราะห์ผลการทดลอง

จากการทดลองการทดสอบความแม่นยำในการตรวจจับของเซนเซอร์อัลตราโซนิกโดยการวัดระยะทางซึ่งจะมีผลถึงระยะปลอดภัยที่ได้กำหนดไว้จากทฤษฎีในบทที่ 2

จากตารางที่ 4.1 ผลการทดลองในการวัดระยะทางโดยอาศัยคลื่นอัลตราโซนิกในแนวราบขนานกับพื้น ทำการทดลองวัดค่าจำนวน 5 ครั้ง เพื่อให้ได้ทราบค่าที่เที่ยงตรงที่สุดเมื่อเทียบกับค่าจริงโดยเริ่มทำการวัดที่ระยะ 5 เซนติเมตรจากเซนเซอร์ไปกระทบกับแผ่นฟิวเจอร์บอร์ดที่เตรียมไว้ จากนั้นเปลี่ยนระยะเป็น 10,20,30...,190 เซนติเมตรเมื่อทำการวัดครบตามจำนวนที่ต้องการแล้วนำค่าที่ได้ไปหาค่าเฉลี่ยของแต่ละระยะ สามารถคำนวณได้จากสมการดังต่อไปนี้

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (4.1)$$

ตัวอย่างการคำนวณที่ระยะ 60 เซนติเมตรจะได้ $\bar{X} = \frac{60+60+60+59+60}{5} = 59.8$

เมื่อได้ค่าเฉลี่ยของแต่ละระยะจึงนำมาหาค่าความผิดพลาดซึ่งคิดจากสมการดังต่อไปนี้
เปอร์เซ็นต์ความคลาดเคลื่อนสัมพัทธ์

$$(\text{percent relative error}) = \left| \frac{(\text{ค่าที่ทดลองได้} - \text{ค่าจริง}) \times 100\%}{\text{ค่าจริง}} \right| \quad (4.2)$$

ตัวอย่างการคำนวณที่ระยะ 60 เซนติเมตรจะได้ $\text{relative error} = \left| \frac{(59.8 - 60) \times 100\%}{60} \right| = 0.33\%$

เมื่อได้ค่าความผิดพลาดของแต่ละระยะแล้วจึงทำการหาค่าเฉลี่ยรวมของ %error ในที่นี้คำนวณเปอร์เซ็นต์ความผิดพลาดเฉลี่ยรวมได้ 1.11% ซึ่งอาจเป็นผลมาจากตัวอุปกรณ์เพื่อแสดงให้เห็นถึงข้อเท็จจริงในการวัดระยะทางของเซนเซอร์อัลตราโซนิก แม้จะมีความคลาดเคลื่อนที่เกิดขึ้นทั้งจากเซนเซอร์อัลตราโซนิกตัวนี้ดังที่กล่าวในทฤษฎีบทที่ 2 และเกิดจากความคลาดเคลื่อนเชิงสถิติซึ่งเป็นธรรมชาติของปรากฏการณ์นั้นๆ มีสาเหตุจากสถานะของตัวแปรภายนอก เช่น อุณหภูมิ ความดัน ฯลฯ หรือตัวแปรภายในที่ไม่สามารถควบคุมได้

สรุปผลการทดลอง

จากผลการทดลองการวัดระยะทางโดยอาศัยคลื่นอัลตราโซนิกในแนวราบกับพื้นจะเห็นว่าในช่วงแรกๆตั้งแต่ระยะที่ 5 เซนติเมตร ถึง 50 เซนติเมตรเซนเซอร์อัลตราโซนิกมีการทำงานที่เที่ยงตรงโดยดูจากค่าความผิดพลาดที่คำนวณได้ เมื่อระยะทางในการทดลองเริ่มไกลขึ้นค่าความผิดพลาดก็จะมีแนวโน้มเพิ่มขึ้นเกิดจากความสามารถในการวัดของตัวเซนเซอร์เนื่องจากเกิดความคลาดเคลื่อนในเชิงสถิติซึ่งเป็นตัวแปรที่ไม่สามารถควบคุมได้และความคลาดเคลื่อนที่เกิดจาก

เซนเซอร์อัลตราโซนิกตัวนี้ซึ่งสามารถวัดระยะได้ 2-400 เซนติเมตรบวกลบ 1.11% ที่ได้จากการทดลอง

การทดลองที่ 2 : ทดสอบวัดระยะใช้งานอุปกรณ์ต้นแบบโดยทดลองกับผู้ใช้งานจริง
วัตถุประสงค์ : เพื่อทดสอบการทำงานของอุปกรณ์ต้นแบบว่ามีประสิทธิภาพความแม่นยำมากน้อยเพียงใดเมื่อทดสอบจากผู้ใช้งานจริง

วัสดุและอุปกรณ์

1. ชุดอุปกรณ์ต้นแบบ
2. ตลับเมตร

ขั้นตอนการทดลอง

1. นำอุปกรณ์ต้นแบบต่อกับ USB ของคอมพิวเตอร์ อัปโหลดโปรแกรมสำหรับการควบคุมอุปกรณ์ต้นแบบแสดงในภาคผนวกข้อที่ 1.4 หน้าที่ 78
2. ทำการทดสอบโดยผู้ใช้งานนั่งอยู่หน้าคอมพิวเตอร์ ปรับตัวเซนเซอร์ให้อยู่ระดับสายตาพร้อมใช้ตลับเมตรวัดระยะห่าง
3. อ่านค่าจาก Serial monitor ที่โปรแกรม Arduino และบันทึกผลการทดลอง

บันทึกผลการทดลอง

จากการทดลองวัดค่าระยะห่างจากหน้าจอคอมพิวเตอร์กับผู้ใช้งานโดยใช้เซนเซอร์อัลตราโซนิกเป็นตัววัดระยะทาง ค่าผลการทดลองแสดงดังตารางผลการทดลองที่ 4.2, 4.3 และ 4.4

ตารางผลการทดลองที่ 4.2ทดสอบวัดระยะผู้ใช้งานคนที่ 1

ระยะจริง (cm)	ระยะที่วัดได้ ครั้งที่ 1 (cm)	ระยะที่วัดได้ ครั้งที่ 2 (cm)	ระยะที่วัดได้ ครั้งที่ 3 (cm)	ค่าเฉลี่ย (cm)	ค่าความผิดพลาด (%error)
5	5	5	5	5	0
10	10	10	10	10	0
15	15	16	15	15.33	2.2
20	21	20	20	20.33	1.65
25	25	26	25	25.33	1.32
30	30	30	32	30.66	2.2
35	35	34	36	35	0
40	41	41	43	41.66	4.15
45	46	45	46	45.66	1.46
50	52	50	52	51.33	2.66
55	56	55	56	55.66	1.2
60	61	60	62	61	1.66
65	66	66	67	66.33	2.04
70	71	72	71	71.33	1.9
75	75	76	76	77.33	0.88
80	80	80	81	80.33	0.41
85	85	85	86	85.33	0.38
90	90	91	90	90.33	0.36
95	96	96	95	95.66	0.69
99	98	99	98	98.33	0.67

ค่าความผิดพลาดเฉลี่ยรวม = 1.29%

ตารางผลการทดลองที่ 4.3ทดสอบวัดระยะผู้ใช้งานคนที่ 2

ระยะจริง (cm)	ระยะที่วัดได้ ครั้งที่ 1 (cm)	ระยะที่วัดได้ ครั้งที่ 2 (cm)	ระยะที่วัดได้ ครั้งที่ 3 (cm)	ค่าเฉลี่ย (cm)	ค่าความผิดพลาด (%error)
5	5	5	5	5	0
10	10	10	10	10	0
15	15	15	15	15	0
20	20	20	20	20	0
25	25	26	26	25.67	2.6
30	30	30	30	30	0
35	35	35	37	35.7	2
40	41	40	41	40.67	1.67
45	45	45	45	45	0
50	51	50	50	50.33	0.66
55	55	55	55	55	0
60	62	61	61	61.33	2.21
65	65	65	65	65.33	0.50
70	71	72	70	71	1.42
75	77	77	76	76.67	2.22
80	81	81	81	81	1.25
85	86	86	85	85.67	0.78
90	92	92	91	91.67	1.85
95	96	96	97	96.33	1.4
99	99	99	100	99.33	0.33

ค่าความผิดพลาดเฉลี่ย = 0.96%

ตารางผลการทดลองที่ 4.4ทดสอบวัดระยะผู้ใช้งานคนที่ 3

ระยะจริง (cm)	ระยะที่วัดได้ ครั้งที่ 1 (cm)	ระยะที่วัดได้ ครั้งที่ 2 (cm)	ระยะที่วัดได้ ครั้งที่ 3 (cm)	ค่าเฉลี่ย (cm)	ค่าความผิดพลาด (%error)
5	5	5	5	5	0
10	10	10	10	10	0
15	15	15	15	15	0
20	20	21	21	20.67	3.35
25	25	26	25	25.33	1.32
30	31	30	31	30.67	2.23
35	36	37	36	36.33	0.94
40	41	40	40	40.33	0.82
45	46	46	46	46	2.22
50	52	51	53	52	4
55	55	57	55	55.67	1.21
60	63	61	62	62	3.33
65	67	67	66	66.66	1.01
70	72	71	70	71	1.42
75	76	76	75	75.67	0.89
80	80	81	80	80.33	0.41
85	86	86	86	86	1.17
90	92	91	92	91.66	1.84
95	96	96	96	96	1.05
99	102	101	102	101.67	2.69

ค่าความผิดพลาดเฉลี่ย = 1.52%

ค่าความผิดพลาดเฉลี่ยรวม 3 คน = 1.25%

วิเคราะห์ผลการทดลอง

จากการทดสอบวัดระยะใช้งานอุปกรณ์ต้นแบบโดยทดลองกับผู้ใช้งานจริง เพื่อทดสอบการทำงานของอุปกรณ์ต้นแบบว่ามีประสิทธิภาพความแม่นยำมากน้อยเพียงใดเมื่อทดสอบจากผู้ใช้งานจริง ทำการทดลองวัดค่า 3 ครั้ง จำนวน 20 ระยะ โดยเริ่มจากระยะ 5 เซนติเมตรจนถึง 99 เซนติเมตร โดยแต่ละระยะที่วัดห่างกัน 5 เซนติเมตร พบว่า

จากตารางที่ 4.2 เมื่อได้ค่าที่วัดระยะแล้วนำมาคำนวณหาค่าเฉลี่ยและค่าความผิดพลาดในแต่ละระยะ ซึ่งสามารถนำมาคำนวณได้จากสมการที่ 1 และสมการที่ 2 จากการในการทดลองที่ 1 ตามลำดับ จะสังเกตเห็นว่าค่าความผิดพลาดของการวัดนี้จะอยู่ที่ช่วง 0-4.15% ของระยะที่วัดนั้นๆ ซึ่งเป็นค่าที่ค่อนข้างมีความผิดพลาดน้อยเมื่อนำมาคิดเป็นหน่วยเซนติเมตร

โดยวิธีคิดดังต่อไปนี้ เช่น ที่ระยะ 15 เซนติเมตร จำนวนออกมาแล้วมีความผิดพลาด 2.2%

แสดงว่า 2.2% ของ 5 เซนติเมตร คือ $(2.2 \times 15) / 100 = 0.33$ เซนติเมตร

และมีค่าความผิดพลาดเฉลี่ยรวม 1.29% โดยหาจากสมการที่ 1 ในการทดลองที่ 1 ได้เลย ซึ่งสามารถคิดเป็นเซนติเมตรดังต่อไปนี้ $(1.29 \times 99) / 100 = 1.27$ เซนติเมตร

จากตารางที่ 4.3 เช่นเดียวกับตารางที่ 4.2 เมื่อได้ค่าที่วัดระยะแล้วนำมาคำนวณหาค่าเฉลี่ยและค่าความผิดพลาดในแต่ละระยะ ซึ่งสามารถนำมาคำนวณได้จากสมการที่ 1 และสมการที่ 2 จากการในการทดลองที่ 1 ตามลำดับ จะสังเกตเห็นว่าค่าความผิดพลาดของการวัดนี้จะอยู่ที่ช่วง 0-2.6% ของระยะที่วัดนั้นๆ เมื่อนำมาคำนวณแล้วพบว่ามีค่าความผิดพลาดเฉลี่ยรวม 0.94% โดยหาจากสมการที่ 1 ในการทดลองที่ 1 สามารถคิดเป็นเซนติเมตรดังต่อไปนี้ $(0.94 \times 99) / 100 = 0.93$ เซนติเมตร ซึ่งเมื่อเทียบกับผลของการทดลองในตารางที่ 4.2 แล้วพบว่ามีค่าความคลาดเคลื่อนน้อยกว่า

จากตารางที่ 4.4 เช่นเดียวกับตารางที่ 4.3 และตารางที่ 4.2 เมื่อได้ค่าที่วัดระยะแล้วนำมาคำนวณหาค่าเฉลี่ยและค่าความผิดพลาดในแต่ละระยะ ซึ่งสามารถนำมาคำนวณได้จากสมการที่ 1 และสมการที่ 2 จากการในการทดลองที่ 1 ตามลำดับ จะสังเกตเห็นว่าค่าความผิดพลาดของการวัดนี้จะอยู่ที่ช่วง 0-3.35% ของระยะที่วัดนั้นๆ เมื่อนำมาคำนวณแล้วพบว่ามีค่าความผิดพลาดเฉลี่ยรวม 1.52% โดยหาจากสมการที่ 1 ในการทดลองที่ 1 สามารถคิดเป็นเซนติเมตรดังต่อไปนี้ $(1.52 \times 99) / 100 = 1.50$ เซนติเมตร ซึ่งเมื่อเทียบกับผลของการทดลองในตารางที่ 4.1, 4.2 และ 4.3 ตามลำดับแล้วพบว่ามีค่าความคลาดเคลื่อนของตารางที่ 4.2 น้อยที่สุด

ค่าความคลาดเคลื่อนของค่าระยะจริงที่วัดในแต่ละตารางการทดลองมีค่ากระโดดไปมานั้นเกิดจากสูตรการหาค่าความคลาดเคลื่อนที่ยังระยะใกล้ๆ จะทำให้มีค่าความคลาดเคลื่อนมากจึงทำการหาค่าเฉลี่ยทั้งหมดของแต่ละตารางการทดลองและจึงเฉลี่ยค่าความคลาดเคลื่อนที่เกิดขึ้นทั้งสามตาราง

การทดลองหาค่าได้จากสมการที่ 1 ในการทดลองที่ 1 จึงได้ค่าความผิดพลาดเฉลี่ยรวม 1.25% สามารถคิดเป็นเซนติเมตรดังต่อไปนี้ $(1.25 \times 99)/100 = 1.23$ เซนติเมตร มีผลมาจากท่าทางในการนั่งใช้งานของผู้ใช้งานคอมพิวเตอร์ การขยับตัวเพียงเล็กน้อย หรือแม้แต่ลักษณะใบหน้าและความสูงของผู้ใช้งานก็อาจมีส่วนในการส่งผลให้เกิดความคลาดเคลื่อนดังกล่าว รวมทั้งจากการวัดระยะอันเนื่องมาจากเครื่องมือที่ใช้วัดและตัวเซนเซอร์เองซึ่งค่าความคลาดเคลื่อนในตัวมันเองอยู่แล้วแต่น้อยมาก

สรุปผลการทดลอง

จากการทดสอบวัดระยะใช้งานอุปกรณ์ต้นแบบโดยทดลองกับผู้ใช้งานจริง เพื่อทดสอบการทำงานของอุปกรณ์ต้นแบบว่ามีประสิทธิภาพความแม่นยำมากน้อยเพียงใดเมื่อทดสอบจากผู้ใช้งานจริงของทั้ง 3 บุคคลดังกล่าว จะเห็นได้ว่ามีค่าไม่คงที่ ในบางครั้งของการวัดระยะมีความคลาดเคลื่อน บางระยะไม่มีความคลาดเคลื่อนเลย ก็มีผลมาจากหลายเหตุผลปัจจัยดังที่กล่าวไว้ในวิเคราะห์ผลการทดลอง ข้างต้น เมื่อนำค่าความคลาดเคลื่อนมาคำนวณให้อยู่ในหน่วยเซนติเมตร แล้วคนที่ 1 มีค่าความคลาดเคลื่อน 1.27 เซนติเมตร คนที่ 2 มีค่าความคลาดเคลื่อน 0.93 เซนติเมตร คนที่ 3 มีค่าความคลาดเคลื่อน 1.50 เซนติเมตร เมื่อนำค่าความคลาดเคลื่อนทั้งสามมาเฉลี่ยได้ 1.23 เซนติเมตร จะได้ว่าต้องทำการปรับโค้ดในการรับค่าและแจ้งเตือนให้มีค่าเพิ่มขึ้นและลดลงจากทฤษฎี ± 1 เซนติเมตร นั่นก็ต้องปรับโค้ดการแจ้งเตือนให้อยู่ในช่วงตั้งแต่ 2-49 เซนติเมตรจากเดิมมีการแจ้งเตือนอยู่ในช่วง 2-50 เซนติเมตร ทั้งนี้ความคลาดเคลื่อนที่เกิดดังผลการทดลองนั้นถือเป็นความคลาดเคลื่อนที่น้อยมากเมื่อเทียบกับการส่งผลต่อเสียต่อผู้ใช้งานและการใช้งานของอุปกรณ์ต้นแบบ ดังนั้นอุปกรณ์ต้นแบบจึงค่อนข้างมีความแม่นยำและมีประสิทธิภาพในการวัดระยะของผู้ใช้งานจริงอยู่มาก

การทดลองที่ 3

:ทดสอบเสียงแจ้งเตือนของอุปกรณ์ต้นแบบโดยทดสอบกับผู้ใช้งานจริง

วัตถุประสงค์

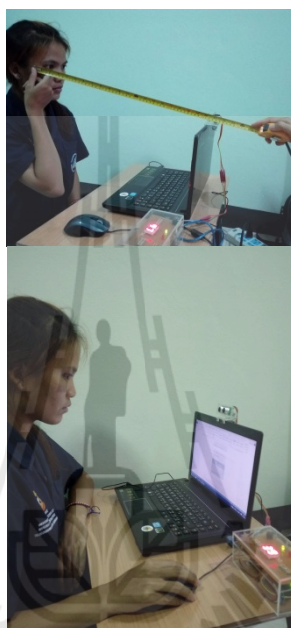
:เพื่อทดสอบอุปกรณ์ว่าสามารถแจ้งเตือนโดยเสียงแจ้งเตือนในแต่ละช่วงระยะทางนั้นจะมีความถี่เสียงการแจ้งเตือนต่างกัน และอุปกรณ์สามารถแจ้งเตือนได้จริงหรือไม่

วัสดุและอุปกรณ์

1. ชุดอุปกรณ์ต้นแบบ
2. ตลับเมตร

ขั้นตอนการทดลอง

1. ต่อชุดอุปกรณ์ต้นแบบเข้ากับ USB ของคอมพิวเตอร์เพื่อใช้เป็นแหล่งจ่ายไฟให้อุปกรณ์ทำงานพร้อมติดตั้งตัวเซนเซอร์ที่จอคอมพิวเตอร์ จึงเริ่มทำการทดลองโดยใช้คลิป์เมตรวัดระยะห่างต่างๆของผู้ใช้งานแสดงดังรูปที่ 4.2



รูปที่ 4.2 การทดสอบการแจ้งเตือนของอุปกรณ์

2. อ่านค่าจาก 7-Segment และบันทึกผลการทดลอง

บันทึกผลการทดลอง

จากการทดลองการแจ้งเตือนของอุปกรณ์โดยทดลองกับผู้ใช้งานจริง ซึ่งผลการทดสอบของแต่ละช่วงระยะทางการแจ้งเตือนจะมีความถี่เสี่ยงในการแจ้งเตือนต่างกัน แบ่งช่วงความถี่ของเสี่ยงแจ้งเตือนเป็น 5 ช่วง จะแสดงในตารางผลการทดลองที่ 4.5, 4.6, 4.7, 4.8 และ 4.9

ตารางผลการทดลองที่ 4.5 ช่วงที่ 1 (ความเร็วเสียง 1)

ระยะทาง (cm)	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3	
	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง
2	2	เสียงดัง	2	เสียงดัง	2	เสียงดัง
4	4	เสียงดัง	4	เสียงดัง	4	เสียงดัง
6	6	เสียงดัง	6	เสียงดัง	6	เสียงดัง
8	8	เสียงดัง	8	เสียงดัง	8	เสียงดัง

ตารางผลการทดลองที่ 4.6 ช่วงที่ 2 (ความเร็วเสียง 2)

ระยะทาง (cm)	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3	
	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง
10	10	เสียงดัง	10	เสียงดัง	10	เสียงดัง
12	12	เสียงดัง	12	เสียงดัง	12	เสียงดัง
14	14	เสียงดัง	14	เสียงดัง	14	เสียงดัง
16	16	เสียงดัง	16	เสียงดัง	17	เสียงดัง
18	18	เสียงดัง	19	เสียงดัง	18	เสียงดัง

ตารางผลการทดลองที่ 4.7 ช่วงที่ 3 (ความเร็วเสียง 3)

ระยะทาง (cm)	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3	
	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง
20	20	เสียงดัง	21	เสียงดัง	20	เสียงดัง
22	22	เสียงดัง	22	เสียงดัง	23	เสียงดัง
24	24	เสียงดัง	25	เสียงดัง	25	เสียงดัง
26	26	เสียงดัง	27	เสียงดัง	27	เสียงดัง
28	28	เสียงดัง	29	เสียงดัง	28	เสียงดัง

ตารางผลการทดลองที่ 4.8 ช่วงที่ 4 (ความเร็วเสียง 4)

ระยะทาง (cm)	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3	
	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง
30	30	เสียงดัง	31	เสียงดัง	31	เสียงดัง
32	33	เสียงดัง	32	เสียงดัง	32	เสียงดัง
34	34	เสียงดัง	35	เสียงดัง	35	เสียงดัง
36	36	เสียงดัง	36	เสียงดัง	37	เสียงดัง
38	39	เสียงดัง	38	เสียงดัง	38	เสียงดัง

ตารางผลการทดลองที่ 4.9 ช่วงที่ 5 (ความเร็วเสียง 5)

ระยะทาง (cm)	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3	
	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง	ระยะจริง (cm)	เสียง
40	41	เสียงดัง	41	เสียงดัง	40	เสียงดัง
42	42	เสียงดัง	42	เสียงดัง	43	เสียงดัง
44	45	เสียงดัง	44	เสียงดัง	44	เสียงดัง
46	46	เสียงดัง	45	เสียงดัง	47	เสียงดัง
48	47	เสียงดัง	48	เสียงดัง	48	เสียงดัง

วิเคราะห์ผลการทดลอง

จากการทดสอบเสียงแฉ่งเต็อน เพื่อทดสอบการทำงานของอุปกรณ์ต้นแบบในการแฉ่งเต็อนแต่ละช่วงระยะทาง พบว่าจากตารางการทดลองที่ 4.5 แสดงค่าระยะทางในช่วง 2-9 เซนติเมตร จะเป็นช่วงที่ใกล้หน้าจอกอมพิวเตอรืมากที่สุด ซึ่งระยะทางในช่วงนี้เสียงบัสเซอร์จะแฉ่งเต็อนด้วยความเร็วเสียงมากที่สุดเมื่อเทียบกับช่วงอื่นๆ และเพื่อเตือนให้ผู้ใช้งานถอยออกจากระยะประชิดกับหน้าจอกอมพิวเตอรื ตารางผลการทดลองที่ 4.6 ช่วงที่ 2 ค่าระยะทางช่วง 10-19 เซนติเมตร ความเร็วของเสียงการแฉ่งเต็อนจะช้ากว่าเมื่อเทียบกับช่วงแรก ตารางการทดลองที่ 4.7 ช่วงที่ 3 ค่าระยะทางช่วง 20-29 เซนติเมตร ความเร็วเสียงของการแฉ่งเต็อนจะช้ากว่าเมื่อเทียบกับช่วงที่สอง

ตารางการทดลองที่ 4.8 ช่วงที่ 4 ค่าระยะทางระหว่าง 30-39 เซนติเมตร ความเร็วเสียงของการแฉ่งเดือนจะช้ากว่าเมื่อเทียบกับช่วงที่สาม ตารางผลการทดลองที่ 4.9 ค่าระยะทางระหว่าง 40-49 เซนติเมตร ความเร็วเสียงของการแฉ่งเดือนจะช้ากว่าเมื่อเทียบกับช่วงที่สี่ แต่ค่าที่เซนเซอร์ทำการวัดได้ไม่คงที่อาจจะเป็นที่การเปลี่ยนอิริยาบถของผู้ใช้งาน ซึ่งเซนเซอร์จะตรวจวัดระยะที่ใบหน้าที่มีส่วนเว้าส่วนโค้งต่างกันค่าระยะทางที่ได้จึงไม่คงที่เมื่อเทียบกับการทดลองที่ 1 ที่ทำการวัดโดยใช้แผ่นฟิวเจอร์บอร์ด เมื่อผู้ใช้งานอยู่ห่างจากหน้าจอกอมพิวเตอร์เป็นระยะทางมากกว่า 50 เซนติเมตร เป็นต้นไป จะไม่ได้ยินเสียงแฉ่งเดือนของอุปกรณ์ต้นแบบ

สรุปผลการทดลอง

จากการทดสอบเสียงแฉ่งเดือน ระยะทางทั้ง 5 ช่วง เริ่มทำการทดลองที่ระยะทาง 2 เซนติเมตร เพิ่มขึ้นครั้งละ 2 เซนติเมตร จนถึง ระยะทาง 48 เซนติเมตร จะเห็นได้ว่าเมื่อทำการทดลองกับผู้ใช้งานจริงในระยะทางต่างๆ เซนเซอร์จะตรวจวัดระยะทางแล้วส่งต่อไปยังบอร์ด เพื่อทำการประมวลผลควบคุมให้ลำโพงแฉ่งเดือน สามารถทำการแฉ่งเดือนตามระยะทางในแต่ละช่วง ตามตารางการทดลองทั้ง 5 ช่วง ซึ่งการทำงานของบัสเซอร์ก็ขึ้นกับการวัดค่าระยะทางของเซนเซอร์ โดยเราจะมุ่งไปที่อุปกรณ์แฉ่งเดือนทำการแฉ่งเดือนในระดับความเร็วเสียงสัมพันธ์กับช่วงระยะทางที่เซนเซอร์อัลตราโซนิกวัดได้ ดังนั้นจะสรุปได้ว่าอุปกรณ์ต้นแบบสามารถทำการแฉ่งเดือนได้ตรงตามช่วงระยะทางต่างๆ

4.3 สรุปผลการทดลอง

จากผลการทดลองการทำงานของอุปกรณ์เตือนระยะปลอดภัยจากการใช้งานคอมพิวเตอร์ โน้ตบุ๊กสามารถแฉ่งเดือนได้จริงตามระยะที่กำหนดไว้ ซึ่งได้ทำการทดสอบการทำงานของตัวเซนเซอร์อัลตราโซนิกดังตารางการทดลองของการทดลองทั้งหมด 3 การทดลองที่ได้กล่าวมาแล้วข้างต้นจึงเห็นได้ว่าการทำงานของตัวเซนเซอร์อัลตราโซนิกมีความน่าเชื่อถือได้รวมถึงการทำงานของบอร์ด Arduino MEGA2560 R3 ลำโพงแฉ่งเดือน ไฟแฉ่งเดือนและการแสดงผลของ 7-Segment

จากการทำงานที่น่าเชื่อของตัวเซนเซอร์อัลตราโซนิกนี้โดยการรับค่ามาแล้วนำไปประมวลผลเพื่อแฉ่งเดือน ซึ่งการจัดสร้างอุปกรณ์แฉ่งเดือนระยะการใช้งานคอมพิวเตอร์โน้ตบุ๊ก มีวัตถุประสงค์เพื่อให้ผู้ใช้งานอยู่ในตำแหน่งของการใช้งานที่เหมาะสมซึ่งเป็นการถนอมสายตาของผู้ใช้งานที่ใช้คอมพิวเตอร์โน้ตบุ๊กอยู่เป็นประจำและลดการหดเกร็งของกล้ามเนื้อดวงตา (Ciliary muscle) เป็นผลทำให้สายตาดีขึ้นโดยปกติแล้วผู้ใช้งานจะไม่ทราบว่าเข้าใกล้หน้าจามากเกินไปจึง

เป็นทางเลือกหนึ่งในการลดความเสี่ยงที่มีส่งผลกระทบต่อสายตาเสื่อมก่อนวัย อีกทั้งเป็นการฝึกทักษะวิชาชีพและการแก้ปัญหา เพื่อสร้างอุปกรณ์ต้นแบบเตือนระยะเวลาการใช้งานคอมพิวเตอร์โน้ตบุ๊กซึ่งมีความสามารถในการทำงานและใช้งานได้จริง ซึ่งสามารถดำเนินการและจัดสร้างให้แล้วเสร็จได้ และจากการทดสอบการทำงานสามารถช่วยถนอมสายตาได้จริงโดยสบายตาขึ้น



บทที่ 5

สรุปและข้อเสนอแนะ

5.1 กล่าวนำ

ในบทที่ 3 ได้กล่าวถึงการออกแบบและการสร้างอุปกรณ์ต้นแบบ เพื่อใช้ในเป็นอุปกรณ์แจ้งเตือนระยะห่างที่เหมาะสมสำหรับการใช้งานคอมพิวเตอร์พกพา โดยใช้ตัวเซนเซอร์อัลตราโซนิกเป็นตัววัดระยะทาง ซึ่งได้ออกแบบให้มีค่าระยะทางมากกว่า 50 เซนติเมตร เป็นระยะทางที่มีความเสี่ยงน้อยที่ได้รับผลกระทบจากหน้าจอคอมพิวเตอร์ ส่วนในบทที่ 5 นี้กล่าวถึงปัญหาที่พบระหว่างการทำโครงการ ข้อเสนอแนะ และแนวทางการพัฒนาต่อไป

5.2 ปัญหาที่พบระหว่างการทำโครงการ

มีความรู้ในการเขียนโปรแกรม Arduino น้อย จึงต้องใช้เวลาในการศึกษาและทำความเข้าใจมาก อีกทั้งมีความรู้ในการออกแบบลายแผ่นวงจรพิมพ์และบัดกรีอุปกรณ์ลงแผ่น PCB ทำให้วงจรมีความเสียหายบ่อยและใช้ไม่ได้ ทำให้ต้องสร้างวงจรหลายครั้งจึงจะสามารถนำมาใช้งานได้

5.3 แนวทางการพัฒนาต่อไป

ในการพัฒนาการสร้างเซนเซอร์แจ้งเตือนระยะการใช้งานระหว่างผู้ใช้งานกับคอมพิวเตอร์ในครั้งต่อไป อาจจะเพิ่มเซนเซอร์วัดความสว่างและเขียนโปรแกรมเพิ่มฟังก์ชันต่างๆ ให้แก่อุปกรณ์ต้นแบบเพื่อที่จะสามารถปรับความสว่างของหน้าจอคอมพิวเตอร์แบบอัตโนมัติ

5.4 ประโยชน์ที่ได้จากโครงการ

ในปัจจุบัน คอมพิวเตอร์พกพาเป็นเทคโนโลยีที่ถูกนำมาใช้งานมากขึ้น ซึ่งส่วนใหญ่เกี่ยวกับ นักเรียน วิทยาลัยและผู้ใช้งานอื่นๆ ดังนั้น ประโยชน์ที่ได้รับโดยตรงจากโครงการนี้ คือ ความรู้ในเรื่องการออกแบบวงจร และความรู้ในเรื่องการใช้โปรแกรม Arduino เพื่อประยุกต์ใช้งานให้สามารถใช้งานตามที่ออกแบบไว้ได้ นอกจากนั้นความรู้ที่ได้จากการศึกษาแล้ว ในการทดลองยังให้ประสบการณ์ ความเข้าใจ และความคิดในการแก้ไขปัญหาเฉพาะหน้าอีกด้วย เนื่องจากการกัดแผ่นวงจรพิมพ์ให้ได้ขนาดตามที่ออกแบบไว้ ต้องทำด้วยมือทั้งหมดทำให้ได้ความชำนาญในเรื่องของการกัดลายแผ่นวงจรพิมพ์ด้วย นอกจากนั้นอุปกรณ์ต้นแบบยังสามารถลดอันตรายจากผู้ใช้งานที่อยู่หน้าจอคอมพิวเตอร์นานๆ

บรรณานุกรม

- [1] <http://www.ku.ac.th/e-magazine/june48/it/computer.html>
- [2] <http://prosoftfamily.softbankthai.com/Article/Detail/24917>
- [3] <http://www.reo14.moe.go.th/index.php/ict/securityinfo/1051-effecticttohealth>
- [4] http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- [5] <https://iamzxlee.wordpress.com/2013/07/06/ultrasonic-sensor-hc-sr04/>
- [6] <http://www.thaieasyelec.com/products/sensors/infrared-ultrasonic/us-015-ultrasonic-module-detail.html>
- [7] <http://tonanasia.com/wordpress/techniques/technical-library>
- [8] <http://www.supremelines.co.th>
- [9] <http://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>
- [10] <http://engineeringkowlge.blogspot.com/2013/03/led-7-segment-micro-controller-pic-pic.html>
- [11] <http://www.thaimicrotron.com/CCS-628/Reference/Speaker.htm>
- [12] <http://arduino.cc/en/Tutorial/Tone>
- [13] <http://www.thaigundam.com/forum/index.php?topic=12893.0>
- [14] <http://html.alldatasheet.com/html-pdf/238279/CUI/CEM-1205C/56/1/CEM-1205C.html>
- [15] <https://www.arduitronics.com/article/23/installing-driver-arduino-on-window>



1. ชุดโปรแกรม

1.1 ชุดโปรแกรมสำหรับเซนเซอร์วัดระยะทาง

```

const int pingPin = 13;
const int inPin = 12;

void setup() {
    Serial.begin(9600);
}

void loop() {
    long duration, cm;
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    pinMode(inPin, INPUT);
    duration = pulseIn(inPin, HIGH);
    cm = microsecondsToCentimeters(duration);
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    delay(500);
}

long microsecondsToCentimeters(long microseconds) {
    // The speed of sound is 340 m/s or 29 microseconds per centimeter.
    // The ping travels out and back, so to find the distance of the
    // object we take half of the distance travelled.
    return microseconds / 29 / 2;
}

```

1.2 ชุดโปรแกรมสำหรับ 7-Segment

```

const byte SEVEN_SEG[7] = {2, 3, 4, 5, 6, 7, 8};
const byte SEVEN_SEG1[7] = {26,28 , 30, 32, 34, 36, 38};

void setup() {
    Serial.begin(9600);/
    for (inti = 0; i < 7; i++) {
        pinMode(SEVEN_SEG[i], OUTPUT);
        pinMode(SEVEN_SEG1[i], OUTPUT);
    }
}

int in;
int a;
int b;

void loop() {
    if(Serial.available()>0){
        in = Serial.parseInt();
        if(in<0||in>99){
            Serial.println("Plese input number between 00 - 99");
        }
        else{
            a = in/10;
            b = in%10;
            if(a==0){
                zero1();}
            else if(a==1){
                one1();}
            else if(a==2){
                two1();}
            else if(a==3){

```

```
        three1();}
else if(a==4){
        four1();}
else if(a==5){
        five1();}
else if(a==6){
        six1();}
else if(a==7){
        seven1();}
else if(a==8){
        eight1();}
else{
        nine1();}
delay(200);
if(b==0){
        zero();}
else if(b==1){
        one();}
else if(b==2){
        two();}

else if(b==3){
        three();}
else if(b==4){
        four();}
else if(b==5){
        five();}
else if(b==6){
        six();}
else if(b==7){
        seven();}
```

```

        else if(b==8){
            eight();}

        else{
            nine(); }

    }

}

void zero() {
    digitalWrite(SEVEN_SEG[0], HIGH);
    digitalWrite(SEVEN_SEG[1], HIGH);
    digitalWrite(SEVEN_SEG[2], HIGH);
    digitalWrite(SEVEN_SEG[3], HIGH);
    digitalWrite(SEVEN_SEG[4], HIGH);
    digitalWrite(SEVEN_SEG[5], HIGH);
    digitalWrite(SEVEN_SEG[6], LOW);
}

void one() {
    digitalWrite(SEVEN_SEG[0], LOW);
    digitalWrite(SEVEN_SEG[1], HIGH);
    digitalWrite(SEVEN_SEG[2], HIGH);
    digitalWrite(SEVEN_SEG[3], LOW);
    digitalWrite(SEVEN_SEG[4], LOW);
    digitalWrite(SEVEN_SEG[5], LOW);
    digitalWrite(SEVEN_SEG[6], LOW);
}

void two() {
    digitalWrite(SEVEN_SEG[0], HIGH);
    digitalWrite(SEVEN_SEG[1], HIGH);
    digitalWrite(SEVEN_SEG[2], LOW);
    digitalWrite(SEVEN_SEG[3], HIGH);
    digitalWrite(SEVEN_SEG[4], HIGH);

```

```
        digitalWrite(SEVEN_SEG[5], LOW);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }

    void three() {
        digitalWrite(SEVEN_SEG[0], HIGH);
        digitalWrite(SEVEN_SEG[1], HIGH);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], HIGH);
        digitalWrite(SEVEN_SEG[4], LOW);
        digitalWrite(SEVEN_SEG[5], LOW);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }

    void four() {
        digitalWrite(SEVEN_SEG[0], LOW);
        digitalWrite(SEVEN_SEG[1], HIGH);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], LOW);
        digitalWrite(SEVEN_SEG[4], LOW);
        digitalWrite(SEVEN_SEG[5], HIGH);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }

    void five() {
        digitalWrite(SEVEN_SEG[0], HIGH);
        digitalWrite(SEVEN_SEG[1], LOW);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], HIGH);
        digitalWrite(SEVEN_SEG[4], LOW);
        digitalWrite(SEVEN_SEG[5], HIGH);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }
```

```
void six() {  
    digitalWrite(SEVEN_SEG[0], HIGH);  
    digitalWrite(SEVEN_SEG[1], LOW);  
    digitalWrite(SEVEN_SEG[2], HIGH);  
    digitalWrite(SEVEN_SEG[3], HIGH);  
    digitalWrite(SEVEN_SEG[4], HIGH);  
    digitalWrite(SEVEN_SEG[5], HIGH);  
    digitalWrite(SEVEN_SEG[6], HIGH);  
}  
  
void seven() {  
    digitalWrite(SEVEN_SEG[0], HIGH);  
    digitalWrite(SEVEN_SEG[1], HIGH);  
    digitalWrite(SEVEN_SEG[2], HIGH);  
    digitalWrite(SEVEN_SEG[3], LOW);  
    digitalWrite(SEVEN_SEG[4], LOW);  
    digitalWrite(SEVEN_SEG[5], LOW);  
    digitalWrite(SEVEN_SEG[6], LOW);  
}  
  
void eight() {  
    digitalWrite(SEVEN_SEG[0], HIGH);  
    digitalWrite(SEVEN_SEG[1], HIGH);  
    digitalWrite(SEVEN_SEG[2], HIGH);  
    digitalWrite(SEVEN_SEG[3], HIGH);  
    digitalWrite(SEVEN_SEG[4], HIGH);  
    digitalWrite(SEVEN_SEG[5], HIGH);  
    digitalWrite(SEVEN_SEG[6], HIGH);  
}  
  
void nine() {  
    digitalWrite(SEVEN_SEG[0], HIGH);  
    digitalWrite(SEVEN_SEG[1], HIGH);
```

```
digitalWrite(SEVEN_SEG[2], HIGH);
digitalWrite(SEVEN_SEG[3], HIGH);
digitalWrite(SEVEN_SEG[4], LOW);
digitalWrite(SEVEN_SEG[5], HIGH);
digitalWrite(SEVEN_SEG[6], HIGH);
}

void zero1() {
    digitalWrite(SEVEN_SEG1[0], HIGH);
    digitalWrite(SEVEN_SEG1[1], HIGH);
    digitalWrite(SEVEN_SEG1[2], HIGH);
    digitalWrite(SEVEN_SEG1[3], HIGH);
    digitalWrite(SEVEN_SEG1[4], HIGH);
    digitalWrite(SEVEN_SEG1[5], HIGH);
    digitalWrite(SEVEN_SEG1[6], LOW);
}

void one1() {
    digitalWrite(SEVEN_SEG1[0], LOW);
    digitalWrite(SEVEN_SEG1[1], HIGH);
    digitalWrite(SEVEN_SEG1[2], HIGH);
    digitalWrite(SEVEN_SEG1[3], LOW);
    digitalWrite(SEVEN_SEG1[4], LOW);
    digitalWrite(SEVEN_SEG1[5], LOW);
    digitalWrite(SEVEN_SEG1[6], LOW);
}

void two1() {
    digitalWrite(SEVEN_SEG1[0], HIGH);
    digitalWrite(SEVEN_SEG1[1], HIGH);
    digitalWrite(SEVEN_SEG1[2], LOW);
    digitalWrite(SEVEN_SEG1[3], HIGH);
    digitalWrite(SEVEN_SEG1[4], HIGH);
```



```
        digitalWrite(SEVEN_SEG1[5], LOW);
        digitalWrite(SEVEN_SEG1[6], HIGH);
    }

    void three1() {
        digitalWrite(SEVEN_SEG[0], HIGH);
        digitalWrite(SEVEN_SEG[1], HIGH);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], HIGH);
        digitalWrite(SEVEN_SEG[4], LOW);
        digitalWrite(SEVEN_SEG[5], LOW);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }

    void four1() {
        digitalWrite(SEVEN_SEG1[0], LOW);
        digitalWrite(SEVEN_SEG1[1], HIGH);
        digitalWrite(SEVEN_SEG1[2], HIGH);
        digitalWrite(SEVEN_SEG1[3], LOW);
        digitalWrite(SEVEN_SEG1[4], LOW);
        digitalWrite(SEVEN_SEG1[5], HIGH);
        digitalWrite(SEVEN_SEG1[6], HIGH);
    }

    void five1() {
        digitalWrite(SEVEN_SEG1[0], HIGH);
        digitalWrite(SEVEN_SEG1[1], LOW);
        digitalWrite(SEVEN_SEG1[2], HIGH);
        digitalWrite(SEVEN_SEG1[3], HIGH);
        digitalWrite(SEVEN_SEG1[4], LOW);
        digitalWrite(SEVEN_SEG1[5], HIGH);
        digitalWrite(SEVEN_SEG1[6], HIGH);
    }
```

```
void six1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);  
    digitalWrite(SEVEN_SEG1[1], LOW);  
    digitalWrite(SEVEN_SEG1[2], HIGH);  
    digitalWrite(SEVEN_SEG1[3], HIGH);  
    digitalWrite(SEVEN_SEG1[4], HIGH);  
    digitalWrite(SEVEN_SEG1[5], HIGH);  
    digitalWrite(SEVEN_SEG1[6], HIGH);  
}
```

```
void seven1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);  
    digitalWrite(SEVEN_SEG1[1], HIGH);  
    digitalWrite(SEVEN_SEG1[2], HIGH);  
    digitalWrite(SEVEN_SEG1[3], LOW);  
    digitalWrite(SEVEN_SEG1[4], LOW);  
    digitalWrite(SEVEN_SEG1[5], LOW);  
    digitalWrite(SEVEN_SEG1[6], LOW);  
}
```

```
void eight1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);  
    digitalWrite(SEVEN_SEG1[1], HIGH);  
    digitalWrite(SEVEN_SEG1[2], HIGH);  
    digitalWrite(SEVEN_SEG1[3], HIGH);  
    digitalWrite(SEVEN_SEG1[4], HIGH);  
    digitalWrite(SEVEN_SEG1[5], HIGH);  
    digitalWrite(SEVEN_SEG1[6], HIGH);  
}
```

```
void nine1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);  
    digitalWrite(SEVEN_SEG1[1], HIGH);
```

```

digitalWrite(SEVEN_SEG1[2], HIGH);
digitalWrite(SEVEN_SEG1[3], HIGH);
digitalWrite(SEVEN_SEG1[4], LOW);
digitalWrite(SEVEN_SEG1[5], HIGH);
digitalWrite(SEVEN_SEG1[6], HIGH);
}

```

1.3 ชุดโปรแกรมสำหรับลำโพงแจ้งเตือนและไฟแจ้งเตือน

1. ลำโพงแจ้งเตือน

```

constint buzzer = 13;
void setup() {
    Serial.begin(9600);
    pinMode(buzzer, OUTPUT);
}
void loop() {
    digitalWrite(buzzer,HIGH)
    delay(500);
    digitalWrite(buzzer,LOW);
}

```

2. ไฟแจ้งเตือน

```

int led = 22;
int led2 = 23;
void setup(){
    Serial.begin(9600);
    pinMode(led, OUTPUT);
    pinMode(led2, OUTPUT);
}
char in;
void loop(){

```

```

        if(Serial.available()>0){
            in = Serial.read();
            if(in == '1'){
                digitalWrite(led, HIGH);
                digitalWrite(led2, LOW);
            }
            else if(in == '0'){
                digitalWrite(led, LOW);
                digitalWrite(led2, HIGH);
            }
        }
    }
}

```

1.4 ชุดโปรแกรมสำหรับรวมอุปกรณ์ต้นแบบ

```

constint LED1 = 22;
constint LED2 = 23;
constint buzzer = 13;
const byte SEVEN_SEG[7] = {2, 3, 4, 5, 6, 7, 8};
const byte SEVEN_SEG1[7] = {26,28 , 30, 32, 34, 36, 38};};
constintpingPin = 9;
constintinPin = 10;
void setup() {
    Serial.begin(9600);
    for (inti = 0; i < 7; i++) {
        pinMode(SEVEN_SEG[i], OUTPUT);
        pinMode(SEVEN_SEG1[i], OUTPUT);
    }
    pinMode(pingPin, OUTPUT);
    pinMode(inPin, INPUT);
    pinMode(LED1, OUTPUT);

```

```

        pinMode(LED2, OUTPUT);
        pinMode(buzzer, OUTPUT);
    }
    int a;
    int b;
    int duration, cm;
    void loop() {
        digitalWrite(pingPin, LOW);
        delayMicroseconds(2);
        digitalWrite(pingPin, HIGH);
        delayMicroseconds(5);
        digitalWrite(pingPin, LOW);
        duration = pulseIn(inPin, HIGH);
        cm = (duration/ 29 / 2);
        Serial.print(cm);
        Serial.print("cm");
        Serial.println();
        if(cm<0||cm>99){
            Serial.println("Plese input number between 00 - 99");
        }
        else{
            digitalWrite(LED1,HIGH);
            a = cm/10;
            b = cm%10;
            if(cm>=0&&cm<50){
                if(a==0){
                    zero1();}
                else if(a==1){
                    one1();}
                else if(a==2){

```

```
two1();}
else if(a==3){
three1();}
else if(a==4){
four1();}
else if(a==5){
five1();}
else if(a==6){
six1();}
else if(a==7){
seven1();}
else if(a==8){
eight1();}
else{
nine1();
}
delay(500);
if(b==0){
zero();}
else if(b==1){
one();}
else if(b==2){
two();}
else if(b==3){
three();}
else if(b==4){
four();}
else if(b==5){
five();}
else if(b==6){
```

```

        six();}
else if(b==7){
        seven();}
else if(b==8){
        eight();}
else{
        nine();
}
if(cm>=40&&cm<50){
        digitalWrite(buzzer,HIGH);
        digitalWrite(LED2,HIGH);
        delay(700);
        digitalWrite(buzzer,LOW);
        digitalWrite(LED2,LOW);
}
else if(cm>=30&&cm<40){
        digitalWrite(buzzer,HIGH);
        digitalWrite(LED2,HIGH);
        delay(500);
        digitalWrite(buzzer,LOW);
        digitalWrite(LED2,LOW);
}
else if(cm>=20&&cm<30){
        digitalWrite(buzzer,HIGH);
        digitalWrite(LED2,HIGH);
        delay(190);
        digitalWrite(buzzer,LOW);
        digitalWrite(LED2,LOW);
}
else if(cm>=10&&cm<20){

```

```

        digitalWrite(buzzer,HIGH);
        digitalWrite(LED2,HIGH);
        delay(70);
        digitalWrite(buzzer,LOW);
        digitalWrite(LED2,LOW);
    }
    else{
        digitalWrite(buzzer,HIGH);
        digitalWrite(LED2,HIGH);
        delay(15);
        digitalWrite(buzzer,LOW);
        digitalWrite(LED2,LOW);
    }
}
else{
    if(a==0){
        zero1();}
    else if(a==1){
        one1();}
    else if(a==2){
        two1();}
    else if(a==3){
        three1();}
    else if(a==4){
        four1();}
    else if(a==5){
        five1();}
    else if(a==6){
        six1();}
    else if(a==7){

```



```
    seven1();}
    else if(a==8){
        eight1();}
    else{
        nine1();
    }

    delay(500);
    if(b==0){
        zero();}
    else if(b==1){
        one();}
    else if(b==2){
        two();}
    else if(b==3){
        three();}
    else if(b==4){
        four();}
    else if(b==5){
        five();}
    else if(b==6){
        six();}
    else if(b==7){
        seven();}
    else if(b==8){
        eight();}
    else{
        nine();
    }
}
```

```
}  
void zero() {  
    digitalWrite(SEVEN_SEG[0], HIGH);  
    digitalWrite(SEVEN_SEG[1], HIGH);  
    digitalWrite(SEVEN_SEG[2], HIGH);  
    digitalWrite(SEVEN_SEG[3], HIGH);  
    digitalWrite(SEVEN_SEG[4], HIGH);  
    digitalWrite(SEVEN_SEG[5], HIGH);  
    digitalWrite(SEVEN_SEG[6], LOW);  
}  
void one() {  
    digitalWrite(SEVEN_SEG[0], LOW);  
    digitalWrite(SEVEN_SEG[1], HIGH);  
    digitalWrite(SEVEN_SEG[2], HIGH);  
    digitalWrite(SEVEN_SEG[3], LOW);  
    digitalWrite(SEVEN_SEG[4], LOW);  
    digitalWrite(SEVEN_SEG[5], LOW);  
    digitalWrite(SEVEN_SEG[6], LOW);  
}  
void two() {  
    digitalWrite(SEVEN_SEG[0], HIGH);  
    digitalWrite(SEVEN_SEG[1], HIGH);  
    digitalWrite(SEVEN_SEG[2], LOW);  
    digitalWrite(SEVEN_SEG[3], HIGH);  
    digitalWrite(SEVEN_SEG[4], HIGH);  
    digitalWrite(SEVEN_SEG[5], LOW);  
    digitalWrite(SEVEN_SEG[6], HIGH);  
}  
void three() {  
    digitalWrite(SEVEN_SEG[0], HIGH);
```

```
digitalWrite(SEVEN_SEG[1], HIGH);
digitalWrite(SEVEN_SEG[2], HIGH);
digitalWrite(SEVEN_SEG[3], HIGH);
digitalWrite(SEVEN_SEG[4], LOW);
digitalWrite(SEVEN_SEG[5], LOW);
digitalWrite(SEVEN_SEG[6], HIGH);
}

void four() {
    digitalWrite(SEVEN_SEG[0], LOW);
    digitalWrite(SEVEN_SEG[1], HIGH);
    digitalWrite(SEVEN_SEG[2], HIGH);
    digitalWrite(SEVEN_SEG[3], LOW);
    digitalWrite(SEVEN_SEG[4], LOW);
    digitalWrite(SEVEN_SEG[5], HIGH);
    digitalWrite(SEVEN_SEG[6], HIGH);
}

void five() {
    digitalWrite(SEVEN_SEG[0], HIGH);
    digitalWrite(SEVEN_SEG[1], LOW);
    digitalWrite(SEVEN_SEG[2], HIGH);
    digitalWrite(SEVEN_SEG[3], HIGH);
    digitalWrite(SEVEN_SEG[4], LOW);
    digitalWrite(SEVEN_SEG[5], HIGH);
    digitalWrite(SEVEN_SEG[6], HIGH);
}

void six() {
    digitalWrite(SEVEN_SEG[0], HIGH);
    digitalWrite(SEVEN_SEG[1], LOW);
    digitalWrite(SEVEN_SEG[2], HIGH);
    digitalWrite(SEVEN_SEG[3], HIGH);
```

```
        digitalWrite(SEVEN_SEG[4], HIGH);
        digitalWrite(SEVEN_SEG[5], HIGH);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }
    void seven() {
        digitalWrite(SEVEN_SEG[0], HIGH);
        digitalWrite(SEVEN_SEG[1], HIGH);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], LOW);
        digitalWrite(SEVEN_SEG[4], LOW);
        digitalWrite(SEVEN_SEG[5], LOW);
        digitalWrite(SEVEN_SEG[6], LOW);
    }
    void eight() {
        digitalWrite(SEVEN_SEG[0], HIGH);
        digitalWrite(SEVEN_SEG[1], HIGH);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], HIGH);
        digitalWrite(SEVEN_SEG[4], HIGH);
        digitalWrite(SEVEN_SEG[5], HIGH);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }
    void nine() {
        digitalWrite(SEVEN_SEG[0], HIGH);
        digitalWrite(SEVEN_SEG[1], HIGH);
        digitalWrite(SEVEN_SEG[2], HIGH);
        digitalWrite(SEVEN_SEG[3], HIGH);
        digitalWrite(SEVEN_SEG[4], LOW);
        digitalWrite(SEVEN_SEG[5], HIGH);
        digitalWrite(SEVEN_SEG[6], HIGH);
    }
```

```
}  
void zero1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);  
    digitalWrite(SEVEN_SEG1[1], HIGH);  
    digitalWrite(SEVEN_SEG1[2], HIGH);  
    digitalWrite(SEVEN_SEG1[3], HIGH);  
    digitalWrite(SEVEN_SEG1[4], HIGH);  
    digitalWrite(SEVEN_SEG1[5], HIGH);  
    digitalWrite(SEVEN_SEG1[6], LOW);  
}  
void one1() {  
    digitalWrite(SEVEN_SEG1[0], LOW);  
    digitalWrite(SEVEN_SEG1[1], HIGH);  
    digitalWrite(SEVEN_SEG1[2], HIGH);  
    digitalWrite(SEVEN_SEG1[3], LOW);  
    digitalWrite(SEVEN_SEG1[4], LOW);  
    digitalWrite(SEVEN_SEG1[5], LOW);  
    digitalWrite(SEVEN_SEG1[6], LOW);  
}  
void two1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);  
    digitalWrite(SEVEN_SEG1[1], HIGH);  
    digitalWrite(SEVEN_SEG1[2], LOW);  
    digitalWrite(SEVEN_SEG1[3], HIGH);  
    digitalWrite(SEVEN_SEG1[4], HIGH);  
    digitalWrite(SEVEN_SEG1[5], LOW);  
    digitalWrite(SEVEN_SEG1[6], HIGH);  
}  
void three1() {  
    digitalWrite(SEVEN_SEG1[0], HIGH);
```

```
digitalWrite(SEVEN_SEG1[1], HIGH);
digitalWrite(SEVEN_SEG1[2], HIGH);
digitalWrite(SEVEN_SEG1[3], HIGH);
digitalWrite(SEVEN_SEG1[4], LOW);
digitalWrite(SEVEN_SEG1[5], LOW);
digitalWrite(SEVEN_SEG1[6], HIGH);
}

void four1() {
    digitalWrite(SEVEN_SEG1[0], LOW);
    digitalWrite(SEVEN_SEG1[1], HIGH);
    digitalWrite(SEVEN_SEG1[2], HIGH);
    digitalWrite(SEVEN_SEG1[3], LOW);
    digitalWrite(SEVEN_SEG1[4], LOW);
    digitalWrite(SEVEN_SEG1[5], HIGH);
    digitalWrite(SEVEN_SEG1[6], HIGH);
}

void five1() {
    digitalWrite(SEVEN_SEG1[0], HIGH);
    digitalWrite(SEVEN_SEG1[1], LOW);
    digitalWrite(SEVEN_SEG1[2], HIGH);
    digitalWrite(SEVEN_SEG1[3], HIGH);
    digitalWrite(SEVEN_SEG1[4], LOW);
    digitalWrite(SEVEN_SEG1[5], HIGH);
    digitalWrite(SEVEN_SEG1[6], HIGH);
}

void six1() {
    digitalWrite(SEVEN_SEG1[0], HIGH);
    digitalWrite(SEVEN_SEG1[1], LOW);
    digitalWrite(SEVEN_SEG1[2], HIGH);
    digitalWrite(SEVEN_SEG1[3], HIGH);
```

```

        digitalWrite(SEVEN_SEG1[4], HIGH);
        digitalWrite(SEVEN_SEG1[5], HIGH);
        digitalWrite(SEVEN_SEG1[6], HIGH);
    }

    void seven1() {
        digitalWrite(SEVEN_SEG1[0], HIGH);
        digitalWrite(SEVEN_SEG1[1], HIGH);
        digitalWrite(SEVEN_SEG1[2], HIGH);
        digitalWrite(SEVEN_SEG1[3], LOW);
        digitalWrite(SEVEN_SEG1[4], LOW);
        digitalWrite(SEVEN_SEG1[5], LOW);
        digitalWrite(SEVEN_SEG1[6], LOW);}

    void eight1() {
        digitalWrite(SEVEN_SEG1[0], HIGH);
        digitalWrite(SEVEN_SEG1[1], HIGH);
        digitalWrite(SEVEN_SEG1[2], HIGH);
        digitalWrite(SEVEN_SEG1[3], HIGH);
        digitalWrite(SEVEN_SEG1[4], HIGH);
        digitalWrite(SEVEN_SEG1[5], HIGH);
        digitalWrite(SEVEN_SEG1[6], HIGH);}

    void nine1() {
        digitalWrite(SEVEN_SEG1[0], HIGH);
        digitalWrite(SEVEN_SEG1[1], HIGH);
        digitalWrite(SEVEN_SEG1[2], HIGH);
        digitalWrite(SEVEN_SEG1[3], HIGH);
        digitalWrite(SEVEN_SEG1[4], LOW);
        digitalWrite(SEVEN_SEG1[5], HIGH);
        digitalWrite(SEVEN_SEG1[6], HIGH); }

```

ประวัติผู้จัดทำ

นายสาธิต ยาหอม เกิดเมื่อวันที่ 9 ตุลาคม 2537 ที่อยู่ตามทะเบียนบ้าน บ้านเลขที่ 8/1 หมู่ 3 ตำบลหนองกระเจา อำเภอยางชุมน้อย จังหวัดนครราชสีมา สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนชุมแสงชนูทิศ อำเภอยางชุมน้อย จังหวัดนครราชสีมา ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นางสาวนริรัตน์ พลจันทร์ เกิดเมื่อวันที่ 31 ตุลาคม 2537 ที่อยู่ตามทะเบียนบ้าน บ้านเลขที่ 171/4 ตำบลลาดบัวขาว อำเภอสว่าง จังหวัดนครราชสีมา สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนสีคิ้ว"สวัสดีคุณวิทยา" อำเภอสว่าง จังหวัดนครราชสีมา ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นางสาวนุชจรี เล็งศรี เกิดเมื่อวันที่ 18 กุมภาพันธ์ 2538 ที่อยู่ตามทะเบียนบ้าน บ้านเลขที่ 46 หมู่ 10 ตำบลหินโคน อำเภอลำปลายมาศ จังหวัดบุรีรัมย์ สำเร็จการศึกษาระดับชั้นมัธยมศึกษาตอนปลายจากโรงเรียนธารทองพิทยาคม อำเภอลำปลายมาศ จังหวัดบุรีรัมย์ ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

