

โปรแกรมตรวจสอบสถานะลานจอดรถยนต์

ชาติชาย ยุธยาตร์

เพื่อความสมบูรณ์แห่งปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์)
สาขาวิชาคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาวิทยาศาสตร์ประยุกต์
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครสวรรค์

พ.ศ. 2561

ปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
โปรแกรมตรวจสอบสถานะลานจอดรถยนต์

นามผู้จัดทำโครงการ นายชาติชาย ยุระยาตร์
ได้พิจารณาเห็นสมควรรับเป็นส่วนหนึ่งของรายวิชาการศึกษาเอกเทศด้านวิทยาการคอมพิวเตอร์
ตามหลักสูตรปริญญาตรีวิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ของมหาวิทยาลัยราชภัฏนครสวรรค์

ลงชื่อ.....อาจารย์ที่ปรึกษา
(อาจารย์วัฒนาพร วัฒนชัยธรรม)

วันที่..... เดือน.....พ.ศ.

ลงชื่อ.....กรรมการ
(อาจารย์ธนพัฒน์ วัฒนชัยธรรม)

วันที่..... เดือน.....พ.ศ.

ลงชื่อ.....กรรมการ
(อาจารย์เอกสิทธิ์ สิทธิสมาน)

วันที่..... เดือน.....พ.ศ.

ลงชื่อ.....
(รศ.ดร.กฤษดา ชันกลีกรรม)

หัวหน้าสาขาวิชาคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

วันที่..... เดือน.....พ.ศ.

กิตติกรรมประกาศ

ในโครงการนี้สำเร็จลุล่วงไปด้วยความเรียบร้อยและสมบูรณ์ ผู้จัดทำขอขอบพระคุณ สำหรับผู้ที่ให้ความช่วยเหลือในด้านต่าง ๆ ที่ให้ความสะดวกในการให้ข้อมูลต่าง ๆ สำหรับในการดำเนินงานโครงการในครั้งนี้ที่ทำให้ผู้จัดทำได้ดำเนินโครงการนี้สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณ อาจารย์วัฒนาพร วัฒนชัยธรรม อาจารย์ที่ปรึกษา และคณาจารย์จาก สาขาวิชาคอมพิวเตอร์และเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏนครสวรรค์ทุกท่าน ที่ให้การอบรมสั่งสอนให้ความรู้ความเข้าใจในศาสตร์ด้านวิทยาการคอมพิวเตอร์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. เทิดพันธุ์ ชูกร หัวหน้าสาขาไฟฟ้าอุตสาหกรรม คณะเทคโนโลยีการเกษตรและเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏนครสวรรค์ ที่ให้คำแนะนำให้ความรู้ความเข้าใจในศาสตร์ด้านไฟฟ้าอุตสาหกรรม และการต่อวงจรไฟฟ้าทั้งหมด

ขอขอบพระคุณ บิดา มารดา ที่เป็นกำลังใจ และขอขอบพระคุณทุกท่านที่มีส่วนเกี่ยวข้องและให้การสนับสนุนเป็นอย่างดีมาโดยตลอด ทำให้การค้นคว้าสำเร็จลุล่วงไปด้วยดี คุณค่าและประโยชน์อันพึงมีจากการค้นคว้าฉบับนี้ ผู้ศึกษาค้นคว้าขออุทิศเพื่อบูชา พระคุณบิดา มารดา อาจารย์ และผู้มีพระคุณทุก ๆ ท่าน

ชาติชาย ยุระยาตร์

บทคัดย่อ

เนื่องจากลานจอดรถยนต์ หรือระบบการให้บริการลานจอดรถยนต์ในปัจจุบันอาจมีความวุ่นวายในการดำเนินการ ด้วยสาเหตุต่าง ๆ เช่น การให้บริการที่จอดรถยนต์ไม่เพียงพอ การให้บริการที่ไม่สะดวก ไม่ว่าจะเป็นตามห้างร้าน หรือลานจอดรถยนต์สาธารณะ ด้วยระบบการให้บริการที่ล่าช้า ทำให้ผู้ใช้งานรู้สึกไม่พึงพอใจ หรือทำให้ผู้ใช้งานไม่สามารถตรวจสอบที่จอดรถได้ในขณะนั้น

การพัฒนาโปรแกรมตรวจสอบสถานะลานจอดรถยนต์นี้มีวัตถุประสงค์เพื่อให้เกิดความเป็นระบบ และระเบียบในการใช้งานสำหรับผู้ที่ใช้ใช้งานลานจอดรถยนต์ ทำให้เกิดความรวดเร็วในการหาที่จอดรถ และมีประสิทธิภาพมากยิ่งขึ้น

การพัฒนาโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ใช้โปรแกรม Visual Basic และโปรแกรม Arduino IDE พัฒนาโปรแกรมด้วยภาษา C ในการพัฒนาโปรแกรมเพื่อที่จะช่วยให้ผู้ที่ใช้งานลานจอดรถยนต์เกิดความสะดวก และทราบถึงสถานะที่จอดรถยนต์อย่างแม่นยำและมีประสิทธิภาพ โปรแกรมนี้แบ่งการทำงานออกเป็น 2 ส่วน 1) ส่วนโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ แสดงสถานะช่องจอดรถยนต์ในขณะนั้น 2) ส่วนของแบบจำลองลานจอดรถยนต์ แสดงข้อมูลช่องว่าง และไฟแสดงสถานะได้อย่างชัดเจน

ซึ่งโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ ช่วยอำนวยความสะดวกในการตรวจสอบสถานะลานจอดรถยนต์ โดยมีการบอกสถานะอย่างถูกต้อง พร้อมแสดงไฟสถานะตามช่องจอดรถ เพื่อให้เกิดความพึงพอใจของผู้ใช้งาน และเกิดความเป็นระบบระเบียบมากยิ่งขึ้น

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อ	ข
สารบัญ	ค
สารบัญภาพ	จ
สารบัญตาราง	ซ
บทที่ 1 บทนำ	1
ความเป็นมาและความสำคัญของปัญหา	1
วัตถุประสงค์ของระบบ	1
ขอบเขตของระบบงาน	2
เครื่องมือที่ใช้ในการพัฒนาระบบ	2
ประโยชน์ที่คาดว่าจะได้รับ	2
ระยะเวลาทำเนิงาน	3
บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง	4
ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์	4
ไมโครคอนโทรลเลอร์ Arduino	7
ประโยชน์ของไมโครคอนโทรลเลอร์	8
ข้อมูลของบอร์ด Arduino Mega	8
หลักการทำงานของ HY-SRF05 Ultrasonic Sensor	10
ภาษา C	12
ประวัติภาษา C	13
วิวัฒนาการของภาษา C	14
งานวิจัยที่เกี่ยวข้อง	15
บทที่ 3 วิธีดำเนินงาน	16
วิเคราะห์ความต้องการของระบบ	16

สารบัญ (ต่อ)

	หน้า
บทที่ 3 วิธีดำเนินงาน (ต่อ)	
การวิเคราะห์และออกแบบระบบ	16
เครื่องมือที่ใช้ในการพัฒนาและออกแบบระบบ	16
Flowchart ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์	17
Flowchart แสดงการทำงานของไฟสถานะ LED	18
Flowchart แสดงการทำงานการแสดงผลบนแบบจำลองลานจอดรถยนต์	19
Hardware Design ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์	20
Output Design ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์	22
บทที่ 4 การติดตั้งและการใช้งานโปรแกรม	23
ขั้นตอนการติดตั้งโปรแกรม Arduino IDE	24
ส่วนของการติดตั้งโปรแกรม	24
ส่วนของการติดตั้งโปรแกรมที่ใช้งาน	29
ส่วนของโปรแกรม	32
ส่วนของการติดตั้งฮาร์ดแวร์	34
บทที่ 5 บทสรุปและข้อเสนอแนะ	36
บทสรุป	36
ประโยชน์ที่ได้รับจากการศึกษาเอกเทศ	36
ปัญหาและอุปสรรคและแนวทางแก้ไข	37
ข้อเสนอแนะ	37
บรรณานุกรม	38
ภาคผนวก	39

สารบัญภาพ

	หน้า
บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง	
ภาพที่ 2.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์	6
ภาพที่ 2.2 Arduino MEGA 2560 R3	9
ภาพที่ 2.3 แสดงหลักการทำงานของตัวตรวจจับโดยใช้สัญญาณความถี่เหนือเสียง	11
ภาพที่ 2.4 แสดงขาสัญญาณของ SRF05 และการกำหนดโหมดทำงาน	12
บทที่ 3 วิธีการดำเนินงาน	
ภาพที่ 3.1 แสดง Flowchart การทำงานของโปรแกรมด้วย Visual Basic	17
ภาพที่ 3.2 แสดง Flowchart การทำงานของโปรแกรมด้วย Arduino	18
ภาพที่ 3.3 แสดง Flowchart การทำงานของโปรแกรมด้วย Arduino (ต่อ)	19
ภาพที่ 3.4 แสดงการออกแบบวงจรอัลตราโซนิก	20
ภาพที่ 3.5 การออกแบบการต่อวงจรหน้าจอแสดงผล	21
ภาพที่ 3.6 การออกแบบการต่อวงจรไฟแสดงผล	21
ภาพที่ 3.7 แสดงหน้าต่างการออกแบบส่วนของการแสดงสถานะ	22
ภาพที่ 3.8 แสดงหน้าต่างการออกแบบส่วนของการแสดงรายการสถานะ	22
ภาพที่ 3.9 แสดงหน้าต่างโปรแกรมการใช้งานหลัก	23
บทที่ 4 การติดตั้งและการใช้งานโปรแกรม	
ภาพที่ 4.1 แสดงหน้าเว็บไซต์ Arduino IDE	24
ภาพที่ 4.2 แสดง License Agreement	25
ภาพที่ 4.3 แสดง Installation Options	25
ภาพที่ 4.4 แสดง Installation Folder	26
ภาพที่ 4.5 แสดงหน้าต่างติดตั้งไดรเวอร์ Adafruit Industries LLC Ports	26
ภาพที่ 4.6 แสดงหน้าต่างติดตั้งไดรเวอร์ Arduino USB Driver srl	27
ภาพที่ 4.7 แสดงหน้าต่างติดตั้งไดรเวอร์ Arduino USB Driver LLC	27

สารบัญภาพ (ต่อ)

	หน้า
บทที่ 4 การติดตั้งและการใช้งานโปรแกรม (ต่อ)	
ภาพที่ 4.8 หน้าต่างการตั้งค่าบอร์ดที่ใช้งาน	28
ภาพที่ 4.9 หน้าต่างการตั้งค่าพอร์ต	28
ภาพที่ 4.10 แสดงหน้าเว็บไซต์ดาวน์โหลดโปรแกรม Smart Parking System	29
ภาพที่ 4.11 ไฟล์ติดตั้งโปรแกรม Setup.msi	29
ภาพที่ 4.12 แสดงหน้าต่างเริ่มขั้นตอนติดตั้งโปรแกรม	30
ภาพที่ 4.13 แสดงหน้าจอ Select Installation Folder	30
ภาพที่ 4.14 แสดงหน้าจอ Confirm Installation	31
ภาพที่ 4.15 แสดงหน้าจอ Installing Program	31
ภาพที่ 4.16 แสดงหน้าจอโปรแกรม	32
ภาพที่ 4.17 แสดงหน้าจอเลือก Serial Port	32
ภาพที่ 4.18 แสดงหน้าต่างจำนวนช่องที่ วาง-ไม่วาง	33
ภาพที่ 4.19 แสดงสถานะจำนวนช่องที่ วาง-ไม่วาง	33
ภาพที่ 4.20 แสดงภาพ Module I2C Interface	34
ภาพที่ 4.21 แสดงภาพแบบจำลองลานจอดรถยนต์	35
ภาคผนวก	39
ภาพที่ 1 แบบจำลองลานจอดรถยนต์	40
ภาพที่ 2 แบบจำลองลานจอดรถยนต์	40
ภาพที่ 3 แบบจำลองลานจอดรถยนต์	41
ภาพที่ 4 แบบจำลองลานจอดรถยนต์	41
ภาพที่ 5 แบบจำลองลานจอดรถยนต์	42
ภาพที่ 6 แบบจำลองลานจอดรถยนต์	42
ภาพที่ 7 แบบจำลองลานจอดรถยนต์	43
ภาพที่ 8 ทดสอบการต่อเซ็นเซอร์อัลตราโซนิก	43
ภาพที่ 9 การออกแบบ แบบจำลองลานจอดรถ	44

สารบัญภาพ(ต่อ)

	หน้า
ภาคผนวก (ต่อ)	
ภาพที่ 10 ทดสอบการต่อเซ็นเซอร์อัลตราโซนิก	44
ภาพที่ 11 ทดสอบการต่อเซ็นเซอร์อัลตราโซนิก ไฟแสดงผล และจอ LCD	45
ภาพที่ 12 ประกอบโครงสร้างแบบจำลองลานจอดรถยนต์	45

สารบัญตาราง

	หน้า
บทที่ 1 ทฤษฎีและเอกสารที่เกี่ยวข้อง	
ตารางที่ 1.1 ระยะเวลาการดำเนินงาน	4
บทที่ 2 ทฤษฎีและเอกสารที่เกี่ยวข้อง	
ตารางที่ 2.1 แสดงข้อมูลเกี่ยวกับบอร์ด Arduino MEGA 2560 R3	10

บทที่ 1

บทนำ

ชื่อระบบงาน	โปรแกรมตรวจสอบสถานะลานจอดรถยนต์	
ชื่อภาษาอังกฤษ	Smart Parking System	
ชื่ออาจารย์ที่ปรึกษา	อาจารย์วัฒนาพร วัฒนชัยธรรม	
ผู้จัดทำ	นายชาติชาย ยุระยาตร์	รหัสนักศึกษา 57113532064

ความเป็นมาและความสำคัญของปัญหา

การจัดการที่จอดรถให้มีความทันสมัยเพื่ออำนวยความสะดวก รวดเร็วด้วยการนำอุปกรณ์เทคโนโลยี เช่น บอร์ด Arduino MEGA R3 2016 เป็นตัวประมวลผลหลัก เพื่อตรวจสอบสถานะที่จอดรถ เข้ามาช่วยจัดการถือเป็นสิ่งจำเป็นอย่างยิ่งสำหรับอาคารลานจอดรถตามสถานที่ต่าง ๆ เพื่อช่วยให้ผู้ขับขี่ที่นำรถเข้ามาใช้บริการ ได้อย่างรวดเร็ว

ซึ่งปัจจุบันสถานที่จอดรถ หรือลานจอดรถหลายแห่งยังคงล้าสมัย โดยยังใช้พนักงานคอยตรวจสอบหรือบางแห่งอาจจะไม่มีพนักงานดูแล แต่ก็ยังไม่เพียงพอที่จะช่วยอำนวยความสะดวกให้ลูกค้าและพนักงาน ที่นำรถเข้ามาใช้บริการที่จอดรถได้ บางสถานที่จะเห็นรถติดเป็นแถวยาวบริเวณด้านทางเข้า และทางออก ขาดความปลอดภัย และความสะดวกสบายในการนำรถเข้า-ออก จากลานจอดรถ ทำให้ลูกค้าหรือพนักงาน รู้สึกไม่อยากจะมาใช้บริการ เนื่องจากเสียเวลา และเบียดเบียนในเรื่องของการนำรถเข้ามาจอด ปัญหาเช่นนี้ก่อให้เกิดความเสียหายอย่างที่ไม่ควรจะเป็น

ดังนั้น ผู้จัดทำจึงมีแนวคิดนำเอาเทคโนโลยีตรวจสอบที่จอดรถด้วยระบบเซ็นเซอร์วัดระยะทางด้วยคลื่นอัลตราโซนิกและใช้บอร์ด Arduino MEGA R3 2016 เป็นตัวประมวลผลนำมาประยุกต์ใช้งานให้เข้ากับการเลือกที่จอดรถ เพื่อให้ผู้ใช้สามารถตรวจสอบสถานะก่อนที่จะนำยานพาหนะเข้าจอดได้และยังมีหน้าตาโปรแกรมที่สามารถใช้งานได้ง่าย และไม่ยุ่งยาก ทั้งนี้เพื่อให้การดำเนินงานการตรวจสอบสถานะที่จอดรถยานพาหนะได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

วัตถุประสงค์

เพื่อพัฒนาโปรแกรมตรวจสอบสถานะที่จอดรถโดยมีวัตถุประสงค์ ดังนี้

1. เพื่อสร้างแบบจำลองลานจอดรถยนต์
2. เพื่อพัฒนาโปรแกรมตรวจสอบสถานะลานจอดรถ

ขอบเขตของโครงการ

1. ส่วนของหน้าต่างโปรแกรม

- 1.1 แสดงสถานะที่จอดรถ

- โปรแกรมสามารถแสดงสถานะที่จอดรถได้ตามเวลาจริง

- 1.2 แสดงจำนวนช่องว่าง

- โปรแกรมสามารถแสดงส่วนของหน้าต่าง จำนวนช่องที่ว่างได้

- 1.3 แสดงพอร์ตอนุกรมในการเชื่อมต่อ

- โปรแกรมสามารถเลือกพอร์ตในการเชื่อมต่อเข้ากับโมเดลได้

2. ส่วนของแบบจำลองลานจอดรถ

2.1 ไฟ LED แสดงสถานะในแต่ละช่องจอด โมเดลลานจอดรถมีสถานะไฟ LED เพื่อแสดงสถานะว่างหรือไม่ว่างได้

2.2 Monitor แสดงตัวเลขสถานะ โมเดลลานจอดรถมีแถบแสดงสถานะจำนวนช่องที่ว่าง

2.3 โมเดลลานจอดรถมีทั้งหมด 4 ช่องจอด

เครื่องมือที่ใช้ในการพัฒนาระบบ

1. ซอฟต์แวร์ที่ใช้ในการพัฒนาโปรแกรม

- 1.1 เครื่องคอมพิวเตอร์ 1 ชุด

- Microsoft Windows 7 (64bit)

- Intel Core i5

- Ram 4 Gb

- 1.2 พัฒนาโปรแกรมด้วย Visual Basic Version 2017 และ Arduino IDE Version 1.8.5

2. ฮาร์ดแวร์ที่ใช้ในการพัฒนาโปรแกรม

2.1 ตัวประมวลผลหลัก Arduino MEGA 2560 R3

2.2 Jump Wire

2.3 LED 4 ดวง

2.4 เซ็นเซอร์ตรวจจับวัตถุ HY-SRF05 Ultrasonic Sensor 4 ตัว

ประโยชน์ที่คาดว่าจะได้รับ

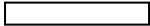
1. เพื่อช่วยในการตรวจสอบที่ว่างสำหรับที่จอดรถในขณะปัจจุบัน
2. เพื่อให้ผู้ใช้ลดระยะเวลาในการหาที่จอด

ระยะเวลาการดำเนินงาน

ตารางที่ 1.1 ระยะเวลาการดำเนินการ

ขั้นตอนการดำเนินงาน	พ.ศ. 2560												พ.ศ. 2561						
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.
1. ทำการศึกษา ระบบงานและรวบรวม ข้อมูล																			
2. นำเสนอเค้าโครงการงาน กับอาจารย์ที่ปรึกษา																			
3. วิเคราะห์และ ออกแบบระบบ																			
4. พัฒนาโปรแกรม																			
5. ทดสอบโปรแกรม และแก้ไขข้อผิดพลาด																			
6. สรุป อภิปรายผลและ จัดทำรูปเล่มรายงาน																			

หมายเหตุ

 แทนระยะเวลาที่วางแผนไว้
 แทนระยะเวลาที่ทำงานจริง

บทที่ 2

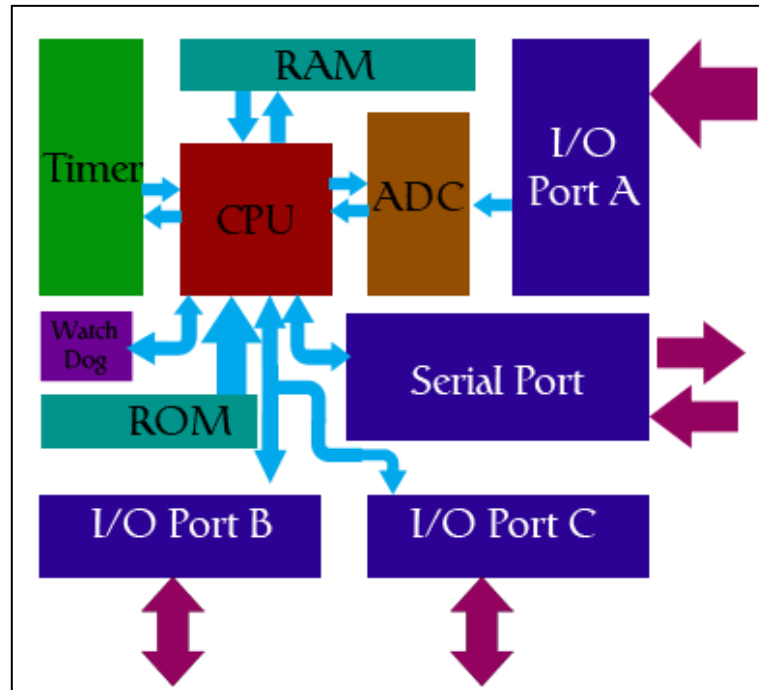
ทฤษฎีและเอกสารที่เกี่ยวข้อง

แนวคิดและทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับการพัฒนา โปรแกรมตรวจสอบสถานะลานจอดรถ ประกอบด้วยแนวคิดและหลักการ ดังต่อไปนี้

1. ความรู้เกี่ยวกับไมโครคอนโทรลเลอร์เบื้องต้น
2. ข้อมูลของบอร์ด Arduino MEGA
3. จุดเด่นของบอร์ด Arduino MEGA
4. หลักการทำงานของ HY-SRF05 Ultrasonic Sensor
5. ภาษา C
6. การออกแบบหน้าต่างโปรแกรม
7. งานวิจัยที่เกี่ยวข้อง

ความรู้เกี่ยวกับไมโครคอนโทรลเลอร์เบื้องต้น

ไมโครคอนโทรลเลอร์ (อังกฤษ: Microcontroller มักย่อว่า μC , uC หรือ MCU) คือ อุปกรณ์ ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวม เอาซีพียู หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดย ทำการบรรจุเข้าไว้ในตัวถังเดียวกัน ไมโครคอนโทรลเลอร์ถ้าแปลความหมายแบบตรงตัวก็คือ ระบบคอนโทรลขนาดเล็กเรียกอีกอย่าง หนึ่งคือเป็นระบบคอมพิวเตอร์ขนาดเล็ก ที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย โดยผ่านการออกแบบ วงจรให้เหมาะกับงานต่าง ๆ และยังสามารถโปรแกรมคำสั่งเพื่อควบคุมขา Input / Output เพื่อสั่งงานให้ไปควบคุมอุปกรณ์ต่าง ๆ ได้อีกด้วย ซึ่งก็นับว่าเป็นระบบที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย ทั้งทางด้าน Digital และ Analog ระบบไมโครคอนโทรลเลอร์ ในยุคปัจจุบันนั้นสามารถทำการเชื่อมต่อกับระบบ Network ของคอมพิวเตอร์ทั่วไปได้อีกด้วย ดังนั้นการสั่งงานจึงไม่ใช่แค่หน้าแผงวงจร แต่อาจจะเป็นการสั่งงานอยู่คนละ ทวีปหรือประเทศผ่านเครือข่ายอินเทอร์เน็ตก็ได้



ภาพที่ 2.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

ที่มา : <https://bit.ly/2UJkKLy>

ไมโครคอนโทรลเลอร์แบ่งออกเป็น 5 ส่วนดังนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งหมด โดยนำข้อมูลจากอุปกรณ์รับข้อมูลมาทำงาน ประมวลผลข้อมูลตามคำสั่งของโปรแกรม และส่งผลลัพธ์ออกไปหน่วยแสดงผล
2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของ เครื่องคอมพิวเตอร์ตั้งโต๊ะ คือ ข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดานทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมากใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณอาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุตเพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) , บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะ หากสัญญาณนาฬิกา มีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์จวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

ไมโครคอนโทรลเลอร์แบ่งออกเป็น 11 ประเภท ดังนี้

1. ไมโครคอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip)
2. ไมโครคอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Phillips)
3. ไมโครคอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโครคอนโทรลเลอร์ตระกูล ARM7,ARM9 (บริษัทผู้ผลิต Atmel, Phillips, Analog Device, Samsung, STMicroelectronics)
5. ไมโครคอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
6. ไมโครคอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
7. ไมโครคอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instruments)
8. ไมโครคอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
9. ไมโครคอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
10. ไมโครคอนโทรลเลอร์ตระกูล RABBIT (บริษัทผู้ผลิต RABBIT SEMICONDUCTOR)
11. ไมโครคอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

ภาษาที่ใช้เขียน โปรแกรมควบคุมไมโครคอนโทรลเลอร์มีทั้งหมด 5 ภาษา ดังนี้

1. ภาษา Assembly
2. ภาษา Basic
3. ภาษา C
4. ภาษา Pascal
5. ภาษา Java

โครงสร้างและส่วนประกอบ

1. CPU
2. Memory
3. Port
4. BUS
5. วงจรกำเนิดสัญญาณนาฬิกา
6. ส่วนประกอบอื่น ๆ

1.1 ไมโครคอนโทรลเลอร์ Arduino

Arduino คือ บอร์ดไมโครคอนโทรลเลอร์ในรูปแบบ open-source เน้นการใช้งานที่ง่ายทั้ง hardware และ software กว่าปีที่ Arduino ได้รับสมองของหลายพันโครงการจากวัตถุในชีวิตประจำวันได้ใช้เครื่องมือทางวิทยาศาสตร์ที่ซับซ้อนชุมชนทั่วโลกของผู้ผลิต นักเรียน มือสมัครเล่น ศิลปิน โปรแกรมเมอร์และผู้เชี่ยวชาญเฉพาะด้าน ได้รวมตัวกันรอบแพลตฟอร์มเปิดแหล่งที่มาที่ผลงานของพวกเขาได้ เพิ่มขึ้นเป็นจำนวนเงินที่เหลือเชื่อของความรู้ที่สามารถเข้าถึงได้ที่สามารถช่วยที่ดีของสามเณรและผู้เชี่ยวชาญเหมือนกัน

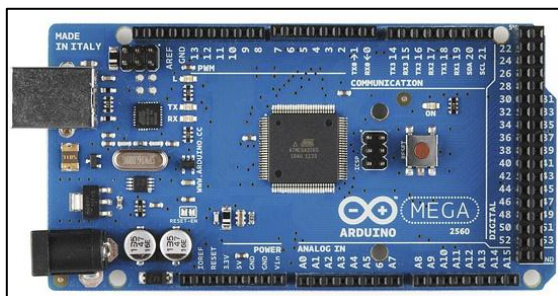
Arduino คือ โครงการที่นำชิปไอซีไมโครคอนโทรลเลอร์ตระกูลต่าง ๆ มาใช้ร่วมกันในภาษา C ซึ่งภาษา C นี้เป็นลักษณะเฉพาะ คือมีการเขียนไวยากรณ์ของ Arduino ขึ้นมาเพื่อให้การสั่งงานไมโครคอนโทรลเลอร์ที่แตกต่างกัน สามารถใช้งานโค้ดตัวเดียวกันได้ โดยตัวโครงการได้ออกบอร์ดทดลองมาหลายรูปแบบ เพื่อใช้งานกับ IDE ของตนเอง สาเหตุหลักที่ทำให้ Arduino เป็นนิยมากเป็นเพราะซอฟต์แวร์ที่ใช้งานร่วมกันสามารถโหลดได้ฟรี และตัวบอร์ดทดลองยังถูกแจกแถม ทำให้ผู้ผลิตจึ้นนำไปผลิตและขายออกตลาดมาในราคาที่ถูกลงมาก ๆ โดยบอร์ดที่ถูกที่สุดในตอนนี้คือบอร์ด Arduino ที่มีราคาเพียง 120 – 150 บาทเท่านั้น

1.2 ประโยชน์ของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ควบคุมอุปกรณ์ไฟฟ้าหรือระบบอิเล็กทรอนิกส์ต่างๆ ไมโครคอนโทรลเลอร์นั้นเปรียบเสมือนคอมพิวเตอร์ขนาดเล็กอย่างไหนทำอะไรเราก็เขียนโปรแกรมที่เราต้องการยัดใส่ลงไป ไมโครคอนโทรลเลอร์ก็เลยเข้ามาเกี่ยวข้องเพื่อรองรับกับความอยากรู้ความเข้าใจที่พวกเราอยากได้โดยให้มีขนาดเล็กที่สุด แต่ว่าไม่ใช่เพียงขนาดเล็กแค่นั้น มันยังสามารถป้อนชุดคำสั่งให้สามารถปฏิบัติงานได้อย่างอัตโนมัติ ด้วยแบบการเขียนโปรแกรมภาษาต่าง ๆ ตามความชำนาญ นัก ออกแบบ พัฒนาผลิตภัณฑ์ ตลอดจนนักประดิษฐ์ทั้งหลาย ต่างหลีกเลี่ยงไม่ได้เลยที่จะต้องอาศัยวงจรอิเล็กทรอนิกส์ เข้าไปมีส่วนเกี่ยวข้องในการควบคุม แต่ครั้งวงจรอิเล็กทรอนิกส์ที่นำมาต่ออนุกรมเพื่อความสามารถที่เราต้องการนั้นก็ใหญ่โตเสียเหลือเกิน ดูเหมือนจะขัดแย้งกับความต้องการของผู้บริโภค และหลักการออกแบบผลิตภัณฑ์

สำหรับโปรแกรมตรวจสอบสถานะลานจอดรถยนต์นั้นได้นำไมโครคอนโทรลเลอร์ตระกูลอา두โน้ เพื่อนำมาพัฒนาระบบด้วยภาษา C โดยอาศัยหลักการทำงานของเซ็นเซอร์ตรวจจับวัตถุด้วยคลื่นอัลตราโซนิก เพื่อให้ทำงานร่วมกับไมโครคอนโทรลเลอร์ตระกูลอาดูโน้ ให้ได้อย่างมีประสิทธิภาพสูงสุด

ข้อมูลของบอร์ด Arduino MEGA



ภาพที่ 2.2 Arduino MEGA 2560 R3

ที่มา : <https://www.arduinoall.com>

เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่นงานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำ

แบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่าในความเร็วของ MCU ที่เท่ากัน

ตารางที่ 2.1 แสดงข้อมูลเกี่ยวกับบอร์ด Arduino MEGA 2560 R3

Microcontroller	ATmega2560
Operating	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output , 4 UART TTL
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of Which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

จุดเด่นของบอร์ด Arduino

1. ง่ายต่อการพัฒนา มีรูปแบบคำสั่งพื้นฐาน ไม่ซับซ้อนเหมาะสำหรับผู้เริ่มต้น
2. มี Arduino Community ที่ร่วมกันพัฒนา
3. Open Hardware ทำให้ผู้ใช้สามารถนำบอร์ดไปต่อยอดใช้งานได้หลายด้าน
4. ราคาไม่แพง
5. Cross Platform สามารถพัฒนาโปรแกรมบนระบบปฏิบัติการ ใดก็ได้

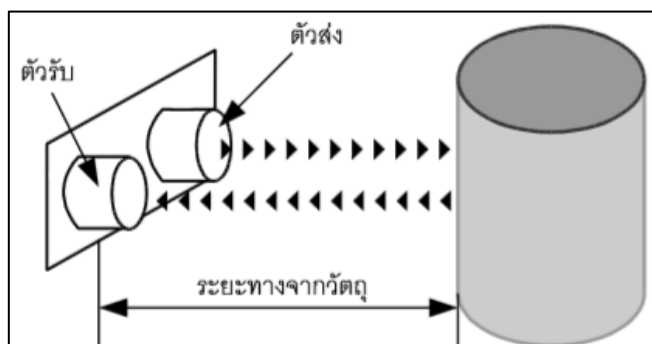
หลักการทำงานของ HY-SRF05 Ultrasonic Sensor

SRF05 เป็นแผงวงจรวัดตรวจจับและวัดระยะทางด้วยคลื่นอัลตราโซนิกที่มีความเที่ยงตรงสูง โดยสามารถวัดระยะได้ตั้งแต่ 1 เซนติเมตรไปจนถึง 4 เมตร SRF04 ถูกออกแบบมาให้ใช้งานกับ ไมโครคอนโทรลเลอร์ได้ง่ายโดยใช้ขาเชื่อมต่อเพียง 1 หรือ 2 ขา ขึ้นอยู่กับการกำหนดรูปแบบการทำงานทาง ฮาร์ดแวร์

SRF05 จะทำการส่งสัญญาณคลื่นอัลตราโซนิกออกไป แล้ววัดระยะเวลาที่สัญญาณสะท้อนตอบกลับมา เอาต์พุตที่ได้จาก SRF05 จะอยู่ในรูปของความกว้างพัลส์ซึ่งสัมพันธ์กับระยะทางของวัตถุที่ตรวจจับได้ ความถี่สัญญาณอัลตราโซนิกของ SRF05 คือ 40kHz ถูกส่งออกไปในอากาศด้วยความเร็ว 1.125 ฟุตต่อมิลลิวินาที (ประมาณ 346 เมตรต่อวินาที) ดังนั้นเมื่อทราบความเร็วในการเคลื่อนที่ของคลื่น, เวลาเริ่มต้นส่งคลื่น และเวลาที่รับเสียงสะท้อนกลับมา จึงสามารถคำนวณหาค่าของระยะทาง ได้ ดังแสดงหลักการทำงานของใน ภาพที่ 2.3

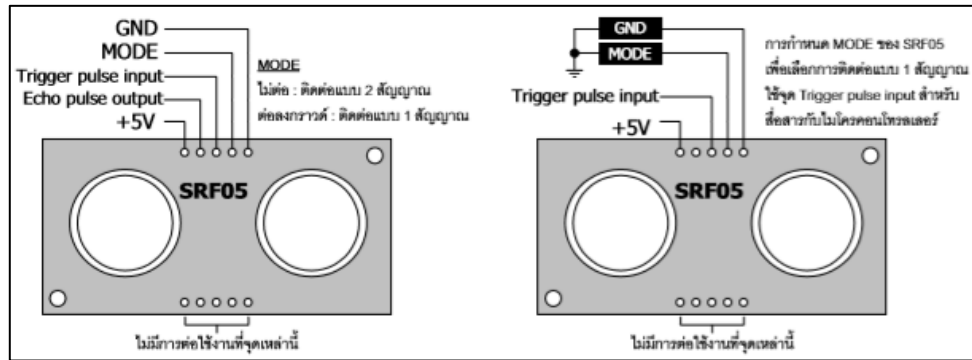
ระยะทางที่ได้นั้นจะต้องมีการคำนวณค่ากลับทางคณิตศาสตร์เมื่อใช้กับไมโครคอนโทรลเลอร์ แล้วถือว่าเป็นเรื่องยุ่งยากพอสมควร ดังนั้น SRF05 จึงประมวลผลค่าทางคณิตศาสตร์ต่าง ๆ เหล่านี้ไว้เรียบร้อยแล้ว จากนั้นส่งผลลัพธ์ที่วัดได้ออกมาเป็นพัลส์ที่มีความกว้างสัมพันธ์กับระยะทางที่วัดได้

การส่งผลลัพธ์ที่วัดได้ออกมาเป็นในเชิงความกว้างของสัญญาณพัลส์อาจจะดูว่ายากกว่าการส่งเป็นข้อมูลดิจิตอลออกมาแต่การส่งออกมาเป็นข้อมูลดิจิตอลจะต้องใช้สายสัญญาณจำนวนมาก ซึ่งทำให้ต้องใช้ขาพอร์ตในการเชื่อมต่อเป็นจำนวนมากตามไปด้วยดังนั้นหากส่งผลลัพธ์ออกมาในรูปของสัญญาณพัลส์จะใช้สายสัญญาณเพียงเส้นเดียวจึงทำให้สะดวกมากในการนำมาเชื่อมต่อกับไมโครคอนโทรลเลอร์



ภาพที่ 2.3 แสดงหลักการทำงานของวัตถุโดยใช้สัญญาณความถี่เหนือเสียงหรืออัลตราโซนิก

ที่มา : www.es.co.th/schematic/pdf/NX-SFR05.pdf



ภาพที่ 2.4 แสดงขาสัญญาณของ SRF05 และการกำหนดโหมดทำงาน

ที่มา : www.es.co.th/schematic/pdf/NX-SFR05.pdf

จุดต่อการใช้งานของ HY-SRF05 Ultrasonic Sensor มีจุดต่อสำหรับการใช้งาน 5 จุด ตามภาพที่ 2.4

1. ขาไฟเลี้ยง (+5V) สำหรับต่อไฟเลี้ยงแรงดัน +5V
2. ขา Echo Pulse Output (ECHO) เป็นขาเอาต์พุตสำหรับส่งสัญญาณพัลส์ออกจาก SRF05 ซึ่งการใช้งานจะนำขานี้ไปต่อเข้ากับพอร์ตอินพุตของไมโครคอนโทรลเลอร์เพื่อตรวจจับความกว้างของสัญญาณพัลส์ที่ส่งออกมาเพื่อแปลความหมายออกมาเป็นระยะทางอีกครั้งหนึ่ง
3. ขา Trigger Pulse Input (TRIGGER) เป็นขาอินพุตรับสัญญาณพัลส์ที่มีความกว้างอย่างน้อย 10 ไมโครวินาทีเพื่อกระตุ้นการสร้างคลื่นอัลตราโซนิกความถี่ 40kHz ออกสู่อากาศจากตัวส่ง ดังนั้นเมื่อคลื่นความถี่ดังกล่าวนี้เคลื่อนที่ไปกระทบสิ่งกีดขวางที่อยู่เบื้องหน้าก็จะเกิดการสะท้อนกลับเข้ามายังตัวรับ และถูกแปลงออกมาเป็นความกว้างของสัญญาณพัลส์ที่จะส่งออกไปทางขา Echo Pulse Output นอกจากในโหมด 1 สัญญาณ จะใช้จุดนี้เป็นจุดสื่อสารข้อมูลอนุกรมเพื่อรับส่งค่าการวัดกับไมโครคอนโทรลเลอร์
4. ขา MODE สำหรับเลือกรูปแบบการติดต่อกับ SRF05 ปลอยลอยไว้ : เลือกให้ติดต่อบน 2 สัญญาณ ผ่านจุดต่อ ECHO และ TRIGGER และ ต่อลงกราวด์ : เลือกให้ติดต่อบน 1 สัญญาณ ผ่านจุดต่อ TRIGGER
5. ขา GND สำหรับต่อกราวด์

สำหรับโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ใช้หลักการทำงานของเซ็นเซอร์ตรวจจับวัตถุด้วยคลื่นเสียงโดยตัวเซ็นเซอร์จะส่งคลื่นเสียงออกไปเพื่อให้ไปกระทบกับวัตถุแล้วส่งคลื่นเสียงกลับมา โดยเป็นคลื่นที่มีความถี่เหนือกว่าคลื่นความถี่เสียงความถี่ที่อัลตราโซนิกส่งออกมาจะเป็นความถี่ที่ 40 kHz

ภาษา C

ภาษาซี เป็นการเขียนโปรแกรมพื้นฐาน สามารถประยุกต์ใช้กับงานต่าง ๆ ได้มากมาย ระบบปฏิบัติการคอมพิวเตอร์ ทางคณิตศาสตร์ โปรแกรมทางไฟฟ้า อิเล็กทรอนิกส์ ไมโครคอนโทรลเลอร์ เช่น โปรแกรม MATLAB (The MathWorks - MATLAB and Simulink for Technical Computing) ซึ่งเวลาใช้สามารถพิมพ์ชุดคำสั่งภาษาซีเพิ่มเข้าไปในโปรแกรมคำนวณทางคณิตศาสตร์ ประมวลผลทางสัญญาณไฟฟ้า ทางไฟฟ้าสื่อสารก็ได้ ทำให้ประสิทธิภาพของงานที่ทำดียิ่งขึ้นครับ และยังมีโปรแกรมอื่น ๆ ที่มีภาษาซีประยุกต์ใช้กันอีกมากมาย ไม่สามารถนำมากล่าวได้หมด ถึงแม้ว่าภาษาซีอาจจะดูเก่าไปสำหรับคนอื่น แต่ผมว่าควรศึกษาภาษาซีที่เป็นรากฐานของภาษาอื่น ๆ เสียก่อน เพราะภาษา C++ จาวา (Java) ฯลฯ และ ระบบลินุกซ์ เป็นระบบที่ถูกพัฒนามาจากระบบยูนิกซ์ซึ่งก็เป็นที่รู้กันทั่วไปว่า ภาษาคู่บารมีของระบบปฏิบัติการตระกูลยูนิกซ์มีการพัฒนามาจากภาษาซีเช่นกัน

ภาษาซีเป็นภาษาที่บางคนเรียกว่าภาษาระดับกลาง คือไม่เป็นภาษาระดับต่ำแบบแอสเซมบลีหรือเป็นภาษาสูงแบบ เบสิก โคบอล ฟอรัทเรน หรือ ปาสคาล เนื่องจากคุณสามารถจะจัดการเกี่ยวกับเรื่องของพอยน์เตอร์ได้อย่างอิสระ และบางทีคุณก็สามารถควบคุมฮาร์ดแวร์ผ่านทาง ภาษาซีได้ราวกับคุณเขียนมันด้วยภาษาแอสเซมบลี ด้วยข้อดีเหล่านี้เองทำให้โปรแกรมที่ถูกเขียนด้วยภาษาซีมีความเร็วในการปฏิบัติงานสูงกว่าภาษาทั่ว ๆ ไป แต่ก็ต้องแลกกับการเรียนรู้และการฝึกฝนอย่างหนัก

1. ประวัติภาษาซี

ภาษาซีเป็นภาษาที่ถือว่าเป็นทั้งภาษาระดับสูงและระดับต่ำ ถูกพัฒนาโดยเดนนิส ริตชี (Dennis- Ritche) แห่งห้องทดลองเบลล์ (Bell Laboratories) ที่เมอร์ริลล์ มลรัฐนิวเจอร์ซีย์ โดยเดนนิส ได้ใช้หลักการของภาษา บีซีพีแอล (BCPL : Basic Combine Programming Language) ซึ่งพัฒนาขึ้นโดยเคน ทอมสัน (Ken Tomson) การออกแบบและพัฒนาภาษาซีของเดนนิส ริตชี มีจุดมุ่งหมายให้เป็นภาษาสำหรับใช้เขียนโปรแกรมปฏิบัติการระบบยูนิกซ์ และได้ตั้งชื่อว่า ซี (C) เพราะเห็นว่า ซี (C) เป็นตัวอักษรต่อจากบี (B) ของภาษา BCPL ภาษาซีถือว่าเป็นภาษาระดับสูงและภาษาระดับต่ำ ทั้งนี้เพราะ ภาษาซีมีวิธีใช้ข้อมูลและมีโครงสร้างการควบคุมการทำงานของโปรแกรมเป็นอย่างเดียวกับภาษาของโปรแกรมระดับสูงอื่น ๆ จึงถือว่าเป็นภาษาระดับสูง ในด้านที่ถือว่าภาษาซีเป็นภาษาระดับต่ำ เพราะภาษาซีมีวิธีการเข้าถึงในระดับต่ำที่สุดของฮาร์ดแวร์ ความสามารถทั้งสองด้านของภาษานี้เป็นสิ่งที่เกื้อหนุนซึ่งกันและกัน ความสามารถระดับต่ำทำให้ภาษาซีสามารถใช้เฉพาะเครื่องได้ และความสามารถระดับสูง ทำให้ภาษาซีเป็นอิสระจากฮาร์ดแวร์ ภาษาซีสามารถสร้างรหัสภาษาเครื่องซึ่งตรงกับชนิดของข้อมูลนั้นได้เอง ทำให้โปรแกรมที่เขียนด้วยภาษาซีที่เขียนบนเครื่องหนึ่ง สามารถนำไปใช้กับอีกเครื่องหนึ่งได้ ประกอบกับการใช้พอยน์เตอร์ในภาษาซี นับได้ว่าเป็นตัวอย่างที่ดีของการเป็นอิสระจากฮาร์ดแวร์

2. วิวัฒนาการของภาษาซี

- ค.ศ. 1970 มีการพัฒนาภาษา B โดย Ken Thompson ซึ่งทำงานบนเครื่อง DEC PDP-7 ซึ่งทำงานบนเครื่องไมโครคอมพิวเตอร์ไม่ได้ และยังมีข้อจำกัดในการใช้งานอยู่ (ภาษา B สืบทอดมาจาก ภาษา BCPL ซึ่งเขียนโดย Marth Richards)

- ค.ศ. 1972 Dennis M. Ritchie และ Ken Thompson ได้สร้างภาษา C เพื่อเพิ่มประสิทธิภาพ ภาษา B ให้ดียิ่งขึ้น ในระยะแรกภาษา C ไม่เป็นที่นิยมแก่นักโปรแกรมเมอร์

- ค.ศ. 1978 Brian W. Kernighan และ Dennis M. Ritchie ได้เขียนหนังสือเล่มหนึ่งชื่อว่า The C Programming Language และหนังสือเล่มนี้ทำให้บุคคลทั่วไปรู้จักและนิยมใช้ภาษา C ในการเขียน โปรแกรมมากขึ้น แต่เดิมภาษา C ใช้ Run บนเครื่องคอมพิวเตอร์ 8 bit ภายใต้ระบบปฏิบัติการ CP/M ของ IBM PC

- ค.ศ. 1981 เป็นช่วงของการพัฒนาเครื่องไมโครคอมพิวเตอร์ ภาษา C จึงมี บทบาทสำคัญ ในการนำมาใช้บนเครื่อง PC ตั้งแต่นั้นเป็นต้นมา และมีการพัฒนาต่อมาอีกหลาย ๆ ค่าย ดังนั้นเพื่อกำหนดทิศทางการใช้ภาษา C ให้เป็นไปแนวทางเดียวกัน ANSI (American National Standard Institute) ได้กำหนดข้อตกลงที่เรียกว่า 3J11 เพื่อสร้างภาษา C มาตรฐานขึ้นมา เรียกว่า ANSI C

- ค.ศ. 1983 Bjarne Stroustrup แห่งห้องปฏิบัติการเบล (Bell Laboratories) ได้พัฒนา ภาษา C++ ขึ้นรายละเอียดและความสามารถของ C++ มีส่วนขยายเพิ่มจาก C ที่สำคัญ ๆ ได้แก่ แนวความคิดของการเขียนโปรแกรมแบบกำหนดวัตถุเป้าหมายหรือแบบ OOP (Object Oriented Programming) ซึ่งเป็นแนวการเขียนโปรแกรมที่เหมาะสมกับการพัฒนาโปรแกรมขนาดใหญ่ที่มีความสลับซับซ้อนมาก มีข้อมูลที่ใช้ในโปรแกรมจำนวนมาก จึงนิยมใช้เทคนิคของการเขียนโปรแกรมแบบ OOP ในการพัฒนาโปรแกรมขนาดใหญ่ในปัจจุบันนี้

โปรแกรมตรวจสอบสถานะลานจอดรถยนต์ใช้ภาษา C เป็นพื้นฐานในการเขียนโปรแกรม ทางด้านฮาร์ดแวร์ทั้งหมด เพื่อให้ฮาร์ดแวร์ทำงานได้อย่างมีประสิทธิภาพสูงสุด เพื่อที่จะนำข้อมูลที่ได้ส่งไปประมวลผล

งานวิจัยที่เกี่ยวข้อง

อาทิตย์ อยู่เย็น และคณะ (2557) ศึกษาและปฏิบัติเกี่ยวกับโครงการระบบรักษาความปลอดภัยการเข้า - ออกสถานที่ ด้วยหลักการคลื่นความถี่วิทยุ และคลื่นความถี่อัลตราโซนิก เข้ากับการพัฒนางจรควบคุมด้วยอาคูโน้ และ RF ID ลักษณะโครงการสามารถควบคุมการเปิดและปิดประตูได้อย่างอัตโนมัติ สามารถบันทึกการเข้าและออกของรถแต่ละคันได้ และสามารถรายงานการเข้าและออกของรถแต่ละคันได้

เดชวิทย์ รัชมี และ ธงรบ อักษร (2560) ศึกษาและปฏิบัติเกี่ยวกับระบบการจัดการลานจอดรถด้วยไมโครคอนโทรลเลอร์ กระบวนการทำงาน รับส่ง-ส่งข้อมูล และออกแบบสร้างแบบจำลองลานจอดรถยนต์ โดยมีช่องจอดรถยนต์จำนวน 5 ช่องจอด โดยเชื่อมต่อด้วยไมโครคอนโทรลเลอร์ Arduino ESP8266 โดยส่งผ่านเครือข่ายไร้สาย (WIFI) และใช้เซ็นเซอร์ HY-SRF05 เพื่อวัดระยะทาง

บทที่ 3

วิธีการดำเนินงาน

การวิเคราะห์ความต้องการของระบบ (Requirement Analysis)

โปรแกรมตรวจสอบสถานะลานจอดรถยนต์ได้ถูกพัฒนาขึ้นมาเพื่อสำหรับผู้ใช้ ที่ต้องการความสะดวกและความรวดเร็วในการตรวจสอบลานจอดรถ เช่น สถานะว่างหรือไม่ว่างในแต่ละช่องนั้น และโปรแกรมถูกออกแบบมาเพื่อให้ผู้ใช้สามารถใช้งานได้ง่าย เป็นต้น

การวิเคราะห์และออกแบบระบบ (System Analysis and Design)

การวิเคราะห์และออกแบบระบบ ถือว่าเป็นขั้นตอนที่สำคัญอย่างมากในการพัฒนาโปรแกรม หากวิเคราะห์ระบบได้ออกมาดี โปรแกรมที่พัฒนาจะสามารถบรรลุถึงวัตถุประสงค์ของการพัฒนา โปรแกรมทำให้โปรแกรมมีประสิทธิภาพสูงสุด เช่นเดียวกับการออกแบบระบบ หากออกแบบมาตรงตามวัตถุประสงค์ โปรแกรมก็จะสามารถทำงานได้อย่างมีประสิทธิภาพ

ซึ่งในการวิเคราะห์และออกแบบโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ นั้นประกอบไปด้วย

1. Flowchart
2. Hardware Design
3. Output Design

เครื่องมือที่ใช้ในการออกแบบและพัฒนาระบบ

1. ด้านฮาร์ดแวร์ (hardware)
 - 1.1 คอมพิวเตอร์
 - 1.1.1 หน่วยประมวลผลกลาง CPU INTEL CORE I5-3317U 1.70 GHz
 - 1.1.2 หน่วยความจำ RAM 4 GB
 - 1.1.3 พื้นที่เก็บข้อมูล 1 TB
 - 1.2 ตัวประมวลผลหลัก Arduino MEGA 2560 R3
 - 1.3 Jump Wire
 - 1.4 LED 6 ดวง
 - 1.5 เซ็นเซอร์ตรวจจับวัตถุ HY-SRF05 Ultrasonic Sensor 4 ตัว

2. ด้านซอฟต์แวร์ (Software)

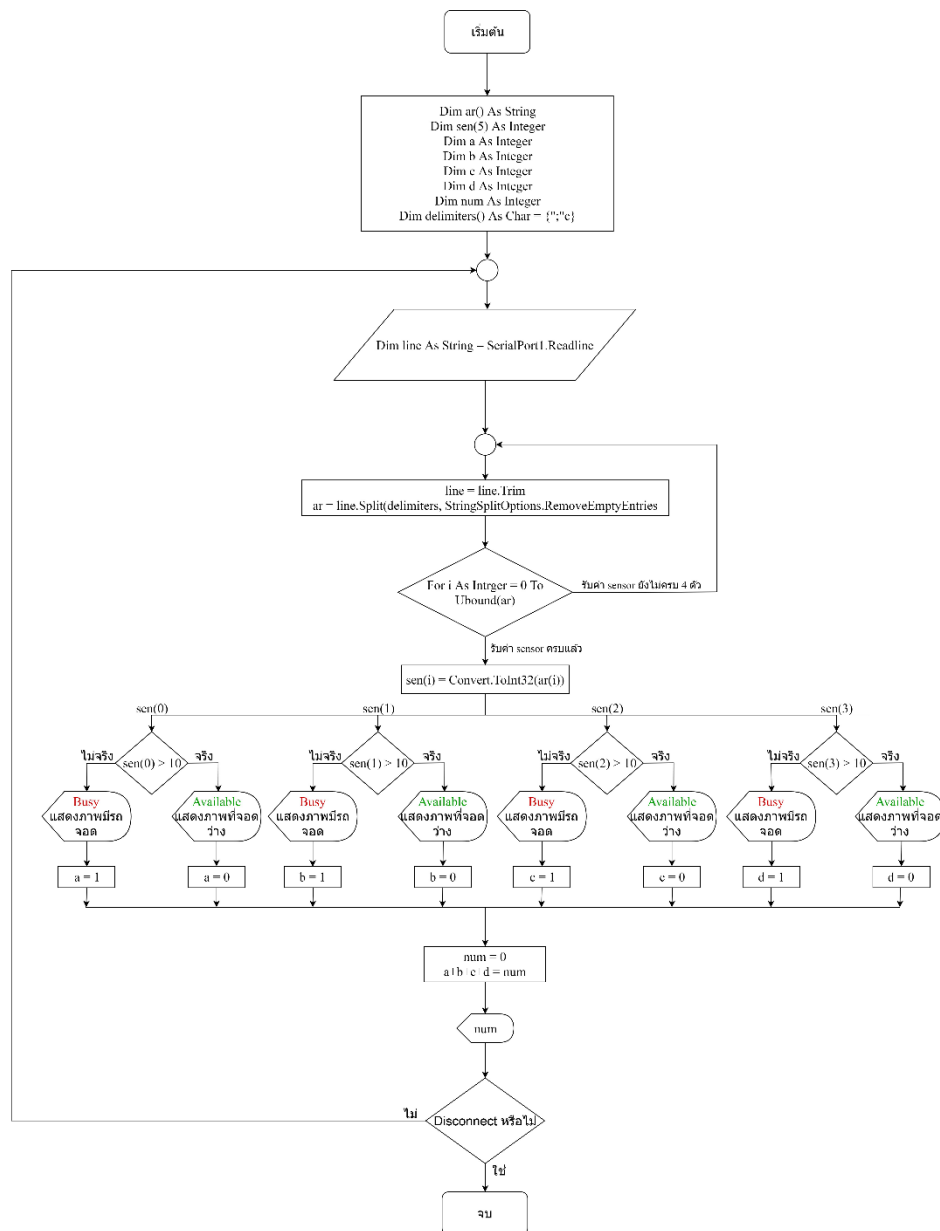
2.1 ระบบปฏิบัติการ Microsoft Windows 7

2.1.1 ใช้ภาษา C และภาษา Basic ในการพัฒนาโปรแกรม

2.1.2 ใช้โปรแกรม Visual Basic และโปรแกรม Arduino IDE ในการพัฒนาโปรแกรม

2.1.3 ใช้โปรแกรม Adobe Photoshop CC2018 ในการตกแต่งภาพ

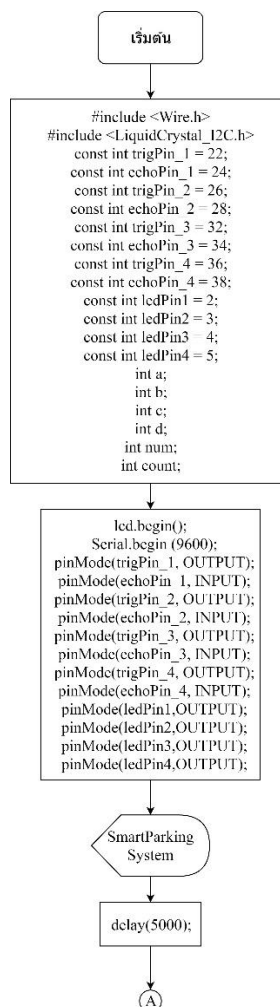
1. Flowchart ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ด้วย Visual Basic



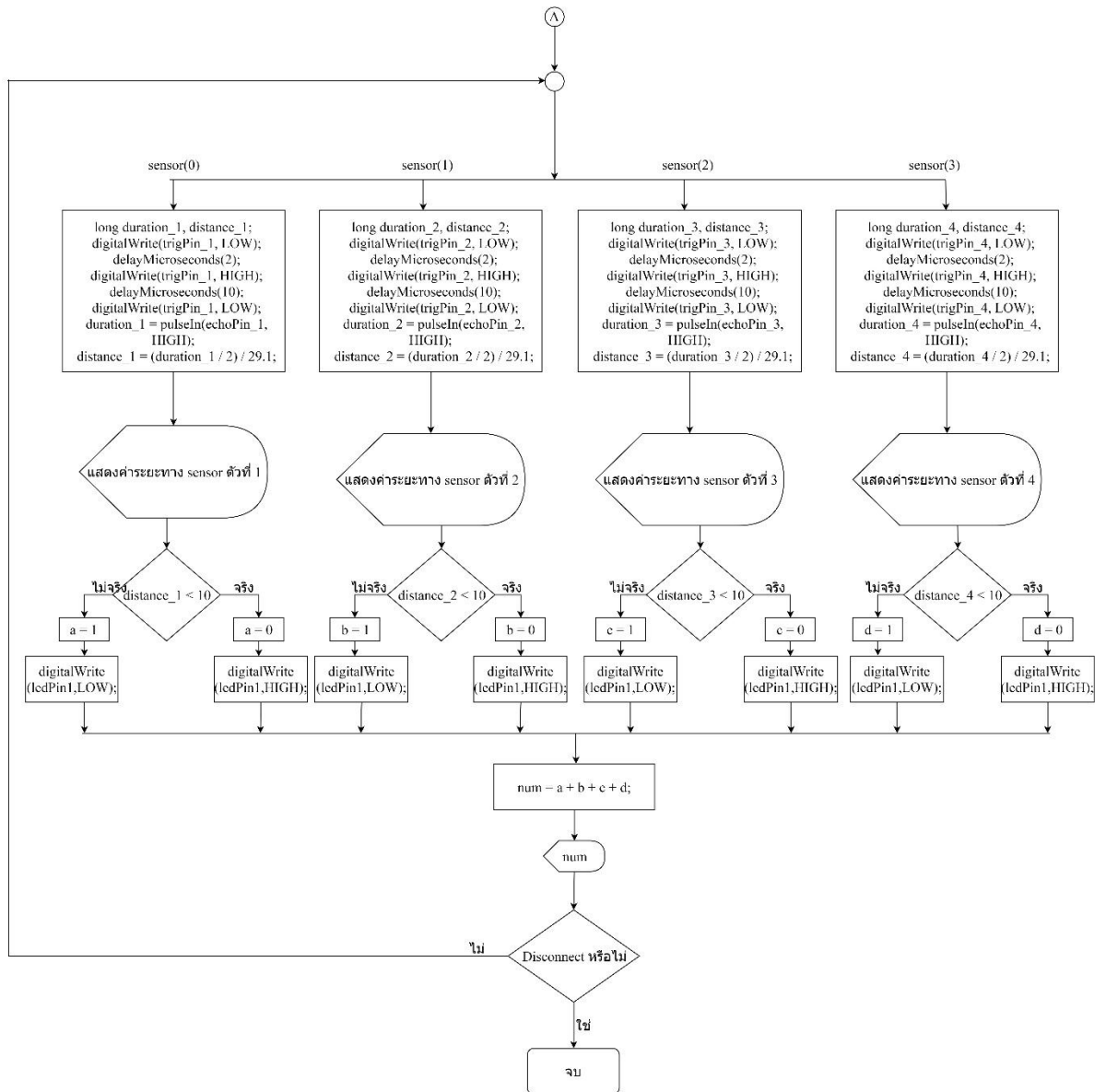
ภาพที่ 3.1 แสดง Flowchart การทำงานของโปรแกรมด้วย Visual Basic

จาก Flowchart ดังภาพที่ 3.1 จะแสดงการทำงานของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ โดยจะเริ่มต้น ด้วยการรับค่าเซ็นเซอร์ทั้ง 4 ตัวมาพร้อมกันในรูปแบบข้อความเดียว โดยมีสัญลักษณ์ ; เป็นตัวคั่นระหว่างเซ็นเซอร์แต่ละตัวเพื่อให้ทราบค่าทั้ง 4 ตัวอย่างถูกต้อง จากนั้นนำค่าที่ได้ทั้ง 4 ตัวมาใช้ฟังก์ชัน Split โดยจะนำค่าทั้ง 4 ตัวใส่เข้าไปในตัวแปรอาเรย์ให้ครบ 4 ตัว หลังจากนั้นวนลูปค่าที่ได้ทั้งหมดเพื่อไปแปลงค่าเป็น Int32 และนำค่าทั้ง 4 ตัวมาใส่ในรูปแบบ case โดยมีเงื่อนไข ถ้าเซ็นเซอร์มากกว่า 10 จริ่งหน้าจอแสดงผลจะแสดงคำว่า Available พร้อมทั้งแสดงภาพที่จอดรถว่าง ถ้าไม่จริ่งหน้าจอแสดงผลจะแสดงคำว่า Busy พร้อมทั้งแสดงภาพที่จอดรถไม่ว่าง โดยที่กำหนดให้สถานะ Available มีค่าเป็น 1 และสถานะ Busy มีสถานะเป็น 0 จากนั้น ค่าค่าที่ได้ทั้งหมดมาบวกกันทั้ง 4 ค่า แล้วแสดงออกทางหน้าจอจำนวนช่องจอดรถที่ว่าง เมื่อแสดงจำนวนเสร็จเรียบร้อยแล้วการทำงานต่อไปจะเป็นเงื่อนไขในการยกเลิกการเชื่อมต่อ โดยที่ถ้ายังไม่ต้องการยกเลิกการเชื่อมต่อจะทำการรับค่าเซ็นเซอร์ใหม่ ถ้าต้องการยกเลิกการเชื่อมต่อโปรแกรมจะสิ้นสุดทันที

2. Flowchart ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ด้วย Arduino



ภาพที่ 3.2 แสดง Flowchart การทำงานของโปรแกรมด้วย Arduino



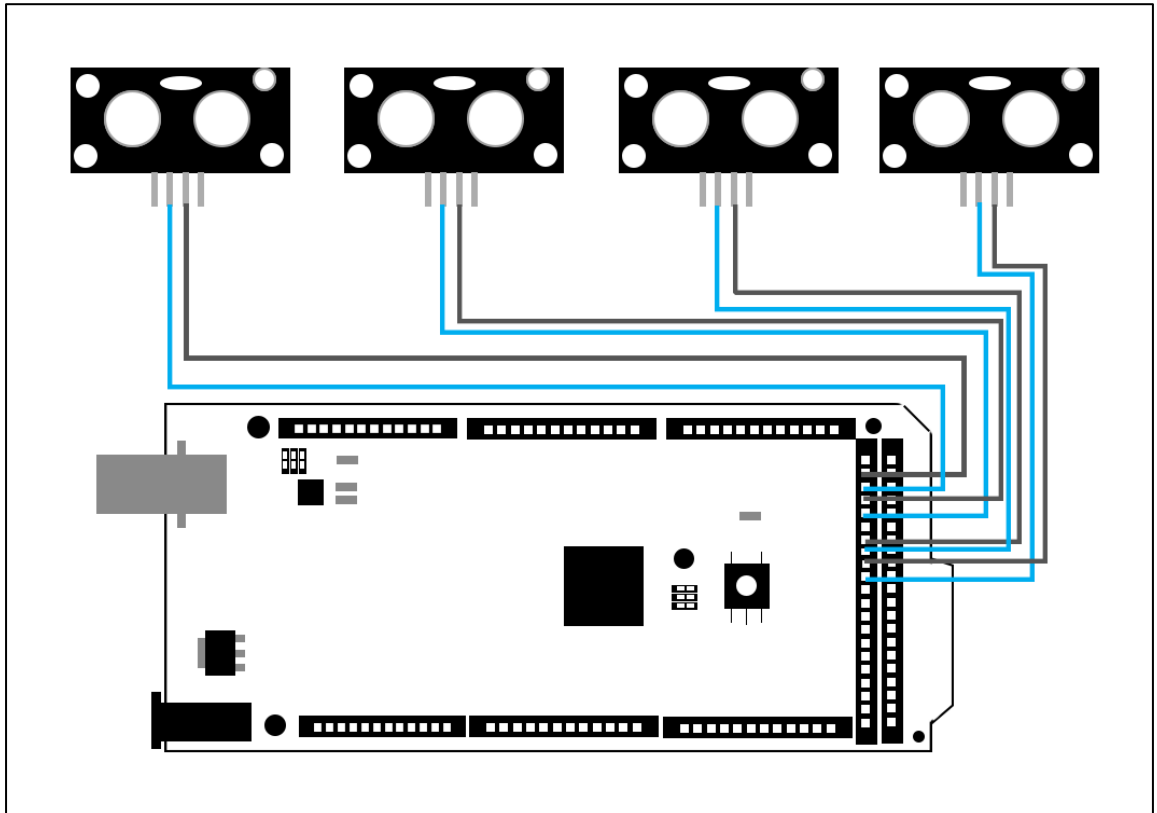
ภาพที่ 3.3 แสดง Flowchart การทำงานของโปรแกรมด้วย Arduino (ต่อ)

จาก Flowchart ดังภาพ 3.3 แสดงการทำงานของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ทางด้าน Arduino โดยจะเริ่มจากการสร้างตัวแปรและกำหนดขา Input และขา Output โปรแกรมจะแสดงผลคำว่า Smart Parking System ออกทางหน้าจอโดยกำหนดค่าดีเลย์เพื่อให้ข้อความดังกล่าวหายไป ระบบจะแยกเป็นรูปแบบ case เพื่อที่จะแยกรับค่าเซ็นเซอร์ทั้ง 4 ตัว แล้วนำค่าที่ได้มาแสดงผลออกทางหน้าจอ โดยกำหนดให้ถ้าระยะน้อยกว่า 10 จริงจะมีค่าเท่ากับ 0 ถ้าระยะทางมากกว่า 10 จะมีค่าเป็น 1 แล้วนำค่าทั้งหมดมาบวกกันเพื่อที่จะได้จำนวนช่องว่าง เมื่อแสดงจำนวนเสร็จเรียบร้อยแล้วการทำงานต่อไปจะเป็นเงื่อนไขในการยกเลิกการเชื่อมต่อ โดยที่ถ้ายังไม่

ต้องการยกเลิกการเชื่อมต่อจะทำการรับค่าเซ็นเซอร์ใหม่ ถ้าต้องการยกเลิกการเชื่อมต่อโปรแกรมจะสิ้นสุดทันที

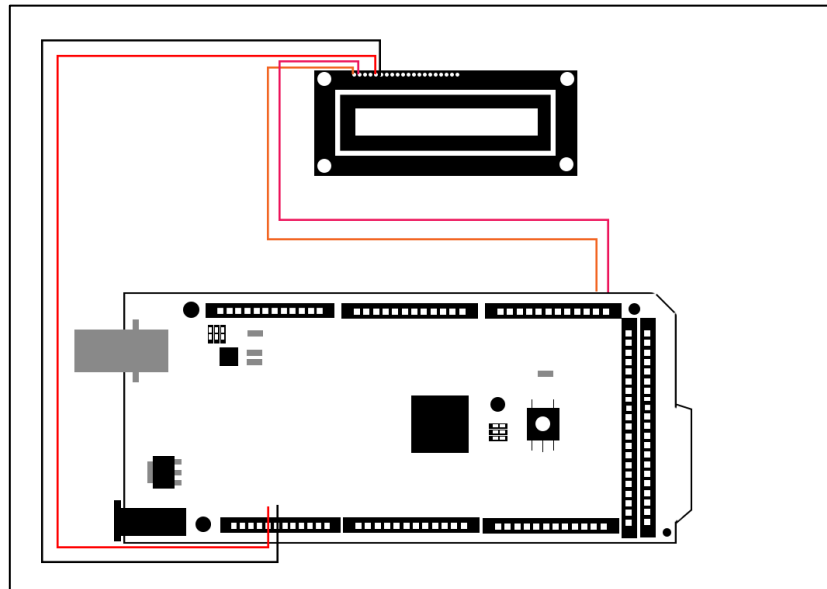
4. Hardware Design ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์

4.1 ส่วนของการออกแบบการต่อวงจรอัลตราโซนิก



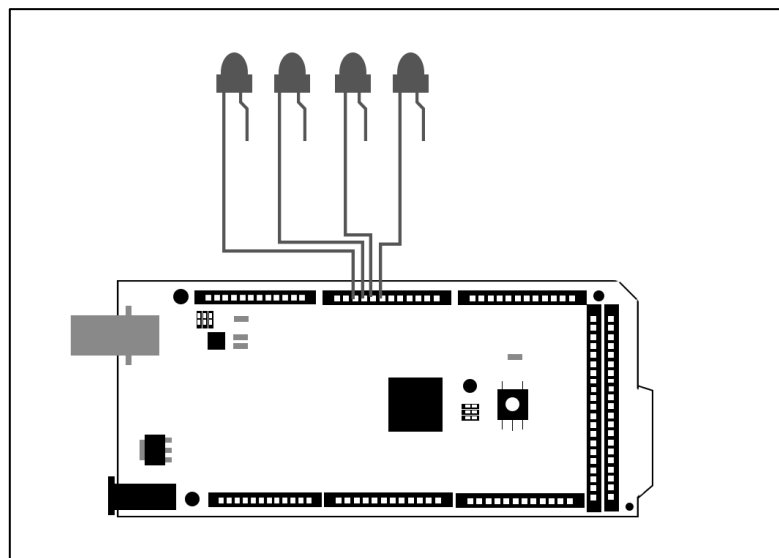
ภาพที่ 3.4 แสดงการออกแบบวงจรอัลตราโซนิก

4.2 ส่วนออกแบบจอแสดงผล



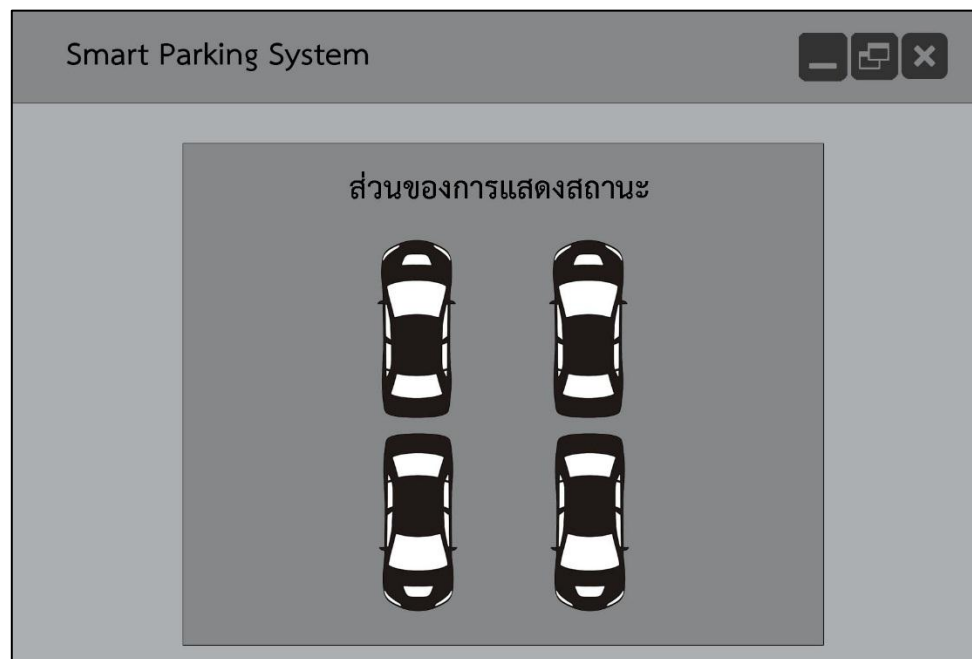
ภาพที่ 3.5 การออกแบบการต่อวงจรหน้าจอแสดงผล

4.3 ส่วนของการออกแบบการต่อวงจรไฟแสดงผล

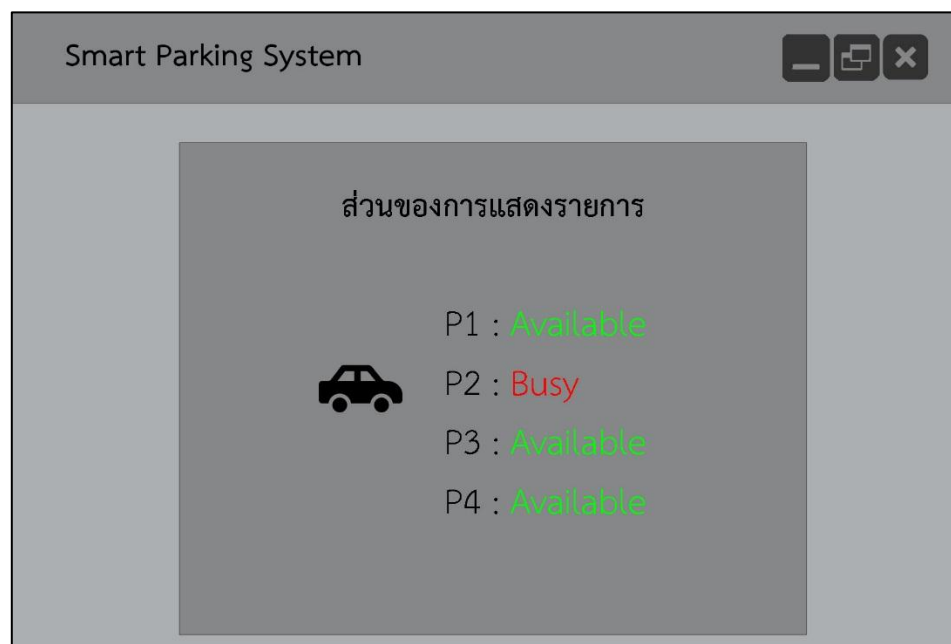


ภาพที่ 3.6 การออกแบบการต่อวงจรไฟแสดงผล

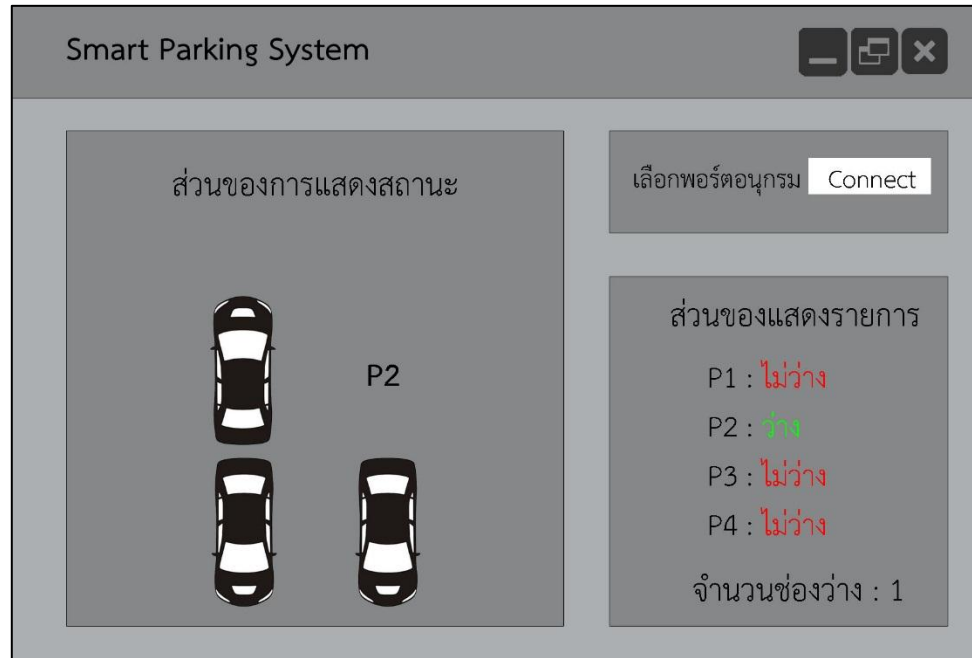
3. Output Design ของโปรแกรมตรวจสอบสถานะลานจอดรถยนต์



ภาพที่ 3.7 แสดงหน้าต่างการออกแบบส่วนของการแสดงสถานะ



ภาพที่ 3.8 แสดงหน้าต่างการออกแบบส่วนของการแสดงรายการสถานะ



ภาพที่ 3.9 แสดงหน้าต่างโปรแกรมการใช้งานหลัก

บทที่ 4

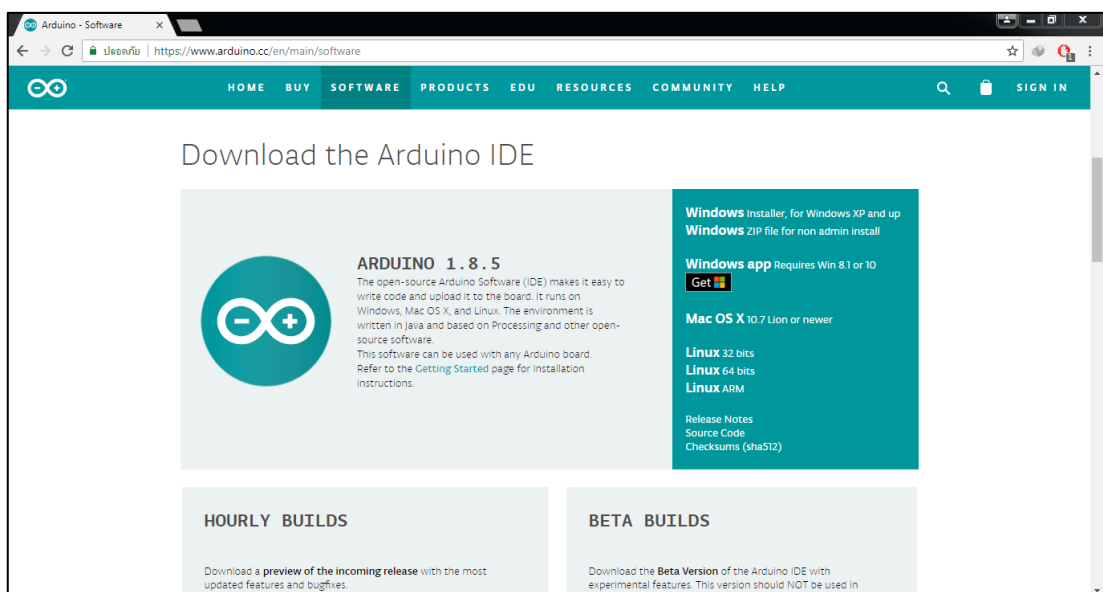
การติดตั้งและการทำงานของโปรแกรม

ขั้นตอนการติดตั้งโปรแกรม

โปรแกรมตรวจสอบสถานะที่จอตรยยนต์ ถูกพัฒนาโปรแกรมมาจาก Visual Basic โดยใช้ภาษา Basic ในการเขียนโปรแกรม และโปรแกรม Arduino IDE โดยใช้ภาษา C ซึ่งผู้พัฒนามีความต้องการให้ผู้ใช้งาน สามารถติดตั้งและใช้งานได้อย่างสะดวก สำหรับโปรแกรมตรวจสอบสถานะที่จอตรยยนต์จะใช้ Visual Basic เวอร์ชัน 2017 ซึ่งผู้ใช้งานสามารถดาวน์โหลดและติดตั้งโปรแกรมได้ดังต่อไปนี้

4.1 ขั้นตอนการติดตั้งโปรแกรม Arduino IDE

4.1.1 ดาวน์โหลดโปรแกรม Arduino IDE ได้จากเว็บไซต์ ดังภาพที่ 4.1

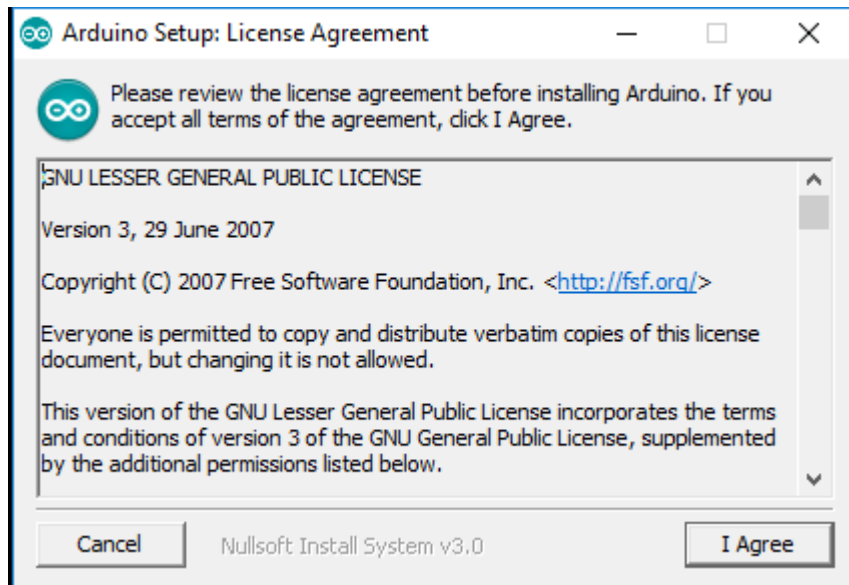


ภาพที่ 4.1 แสดงหน้าเว็บไซต์ Arduino IDE

4.1.2 เลือก Download สำหรับ Microsoft Windows, Mac OS หรือ Linux

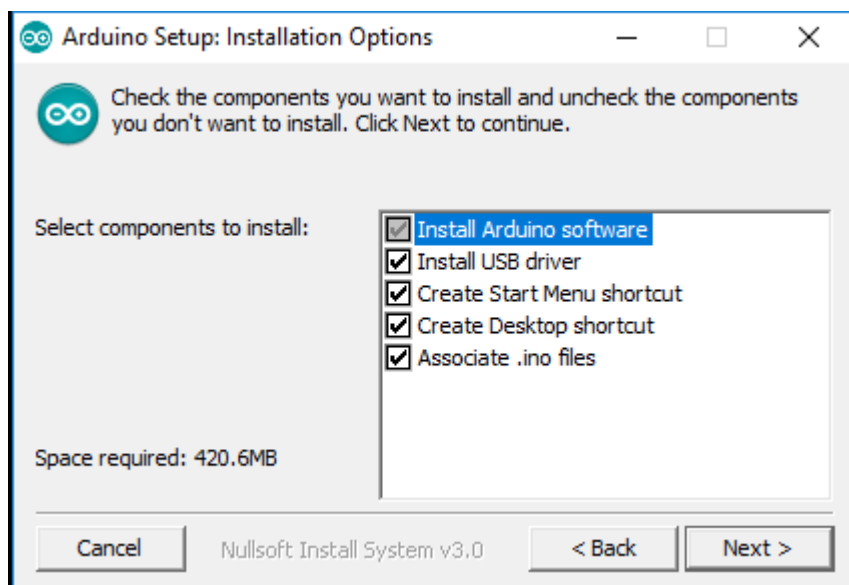
4.1.3 เมื่อเสร็จสิ้นการดาวน์โหลดจะได้ชื่อไฟล์ arduino-1.8.5-windows.exe

4.1.4 เมื่อเปิดไฟล์ arduino-1.8.5-windows.exe จะปรากฏ License ให้คลิก I Agree
 ดังภาพ 4.2



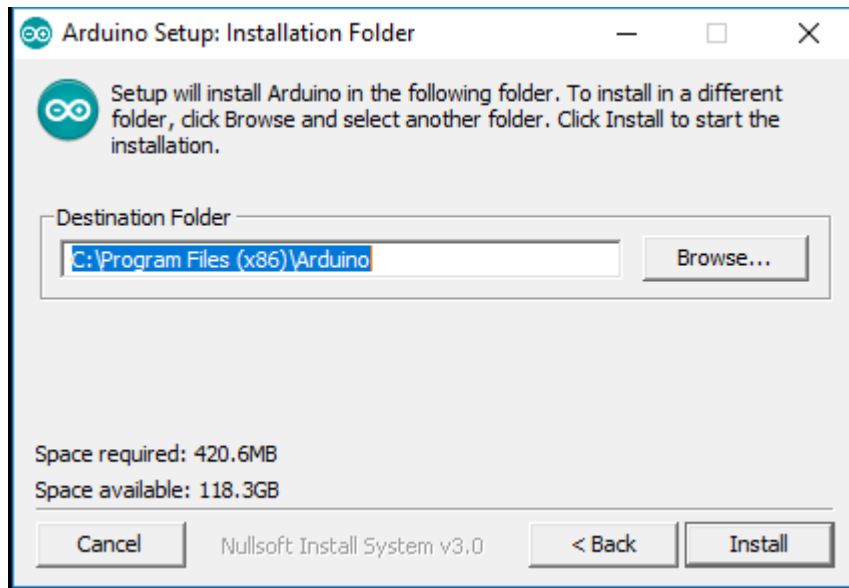
ภาพที่ 4.2 แสดง License Agreement

4.1.5 ทำการเลือก components ทั้งหมดเพื่อติดตั้งโปรแกรม ดังภาพที่ 4.3



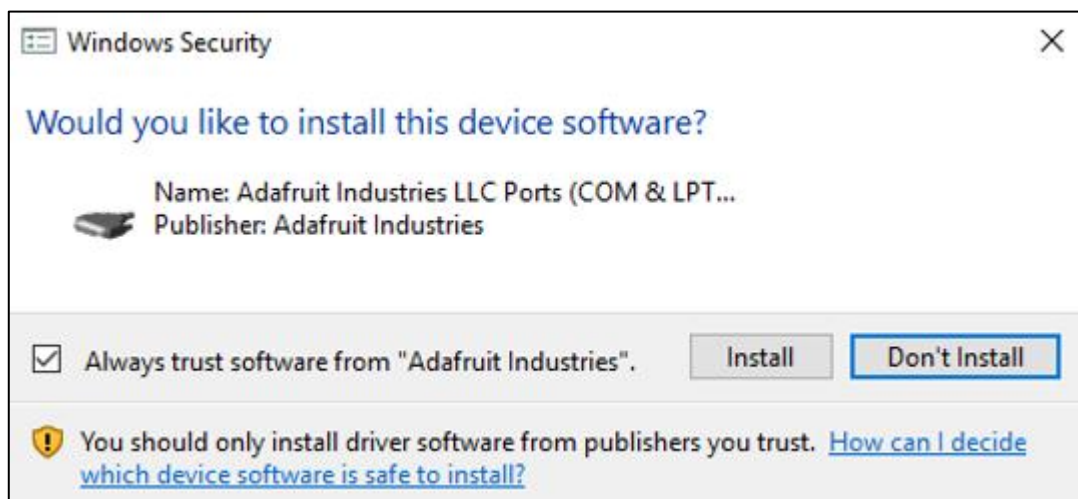
ภาพที่ 4.3 แสดง Installation Options

4.1.6 เลือกโฟลเดอร์ที่จะทำการติดตั้งโปรแกรม ดังภาพที่ 4.4

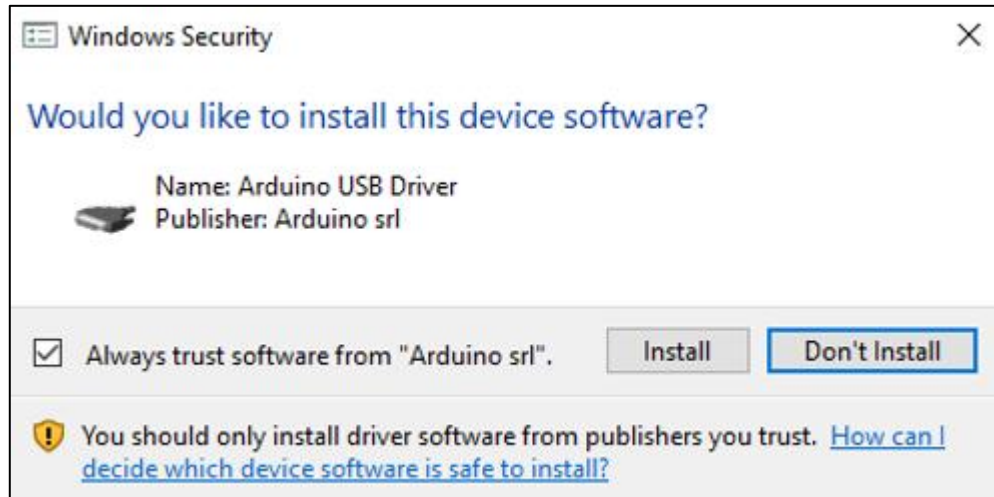


ภาพที่ 4.4 แสดง Installation Folder

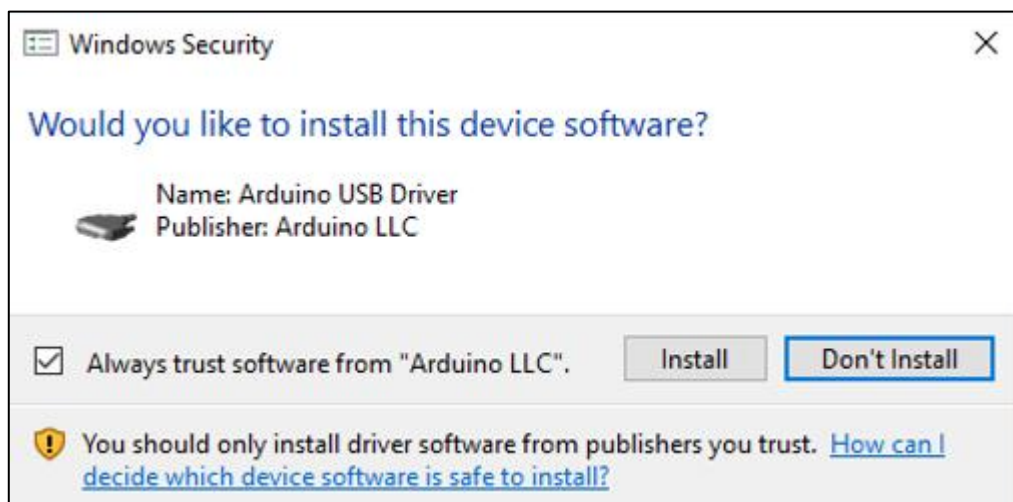
4.1.7 หลังจากกำลังติดตั้งโปรแกรม โปรแกรมจะทำการให้ติดตั้งไดรเวอร์ ดังภาพที่ 4.5 และภาพที่ 4.6 และภาพที่ 4.7



ภาพที่ 4.5 แสดงหน้าต่างติดตั้งไดรเวอร์ Adafruit Industries LLC Ports



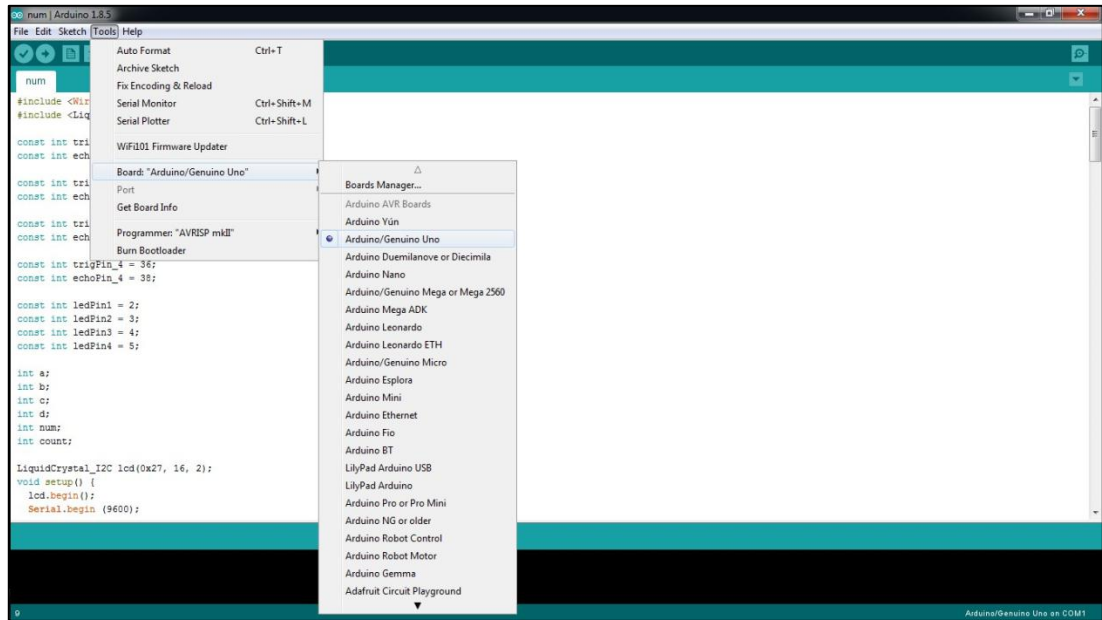
ภาพที่ 4.6 แสดงหน้าต่างติดตั้งไดรเวอร์ Arduino USB Driver srl



ภาพที่ 4.7 แสดงหน้าต่างติดตั้งไดรเวอร์ Arduino USB Driver LLC

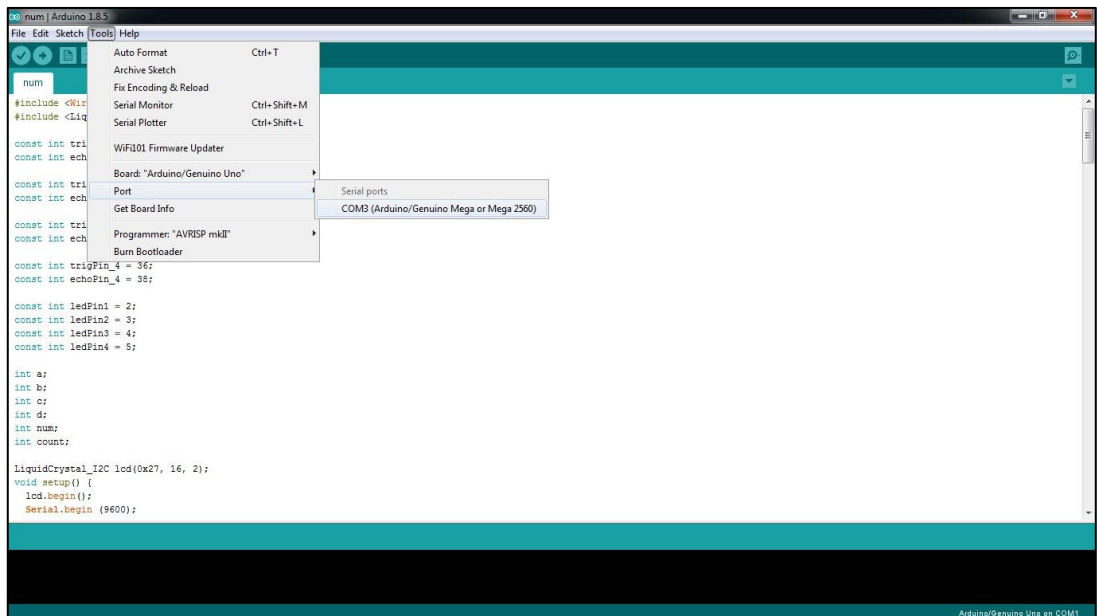
4.1.8 หลังจากติดตั้งโปรแกรม Arduino IDE เสร็จเรียบร้อยแล้วให้ทำการเปิดโปรแกรมเพื่อตั้งค่าต่าง ๆ

4.1.9 ตั้งค่าให้ตรงกับบอร์ดที่ใช้งาน ดังภาพที่ 4.8



ภาพที่ 4.8 หน้าต่างการตั้งค่าบอร์ดที่ใช้งาน

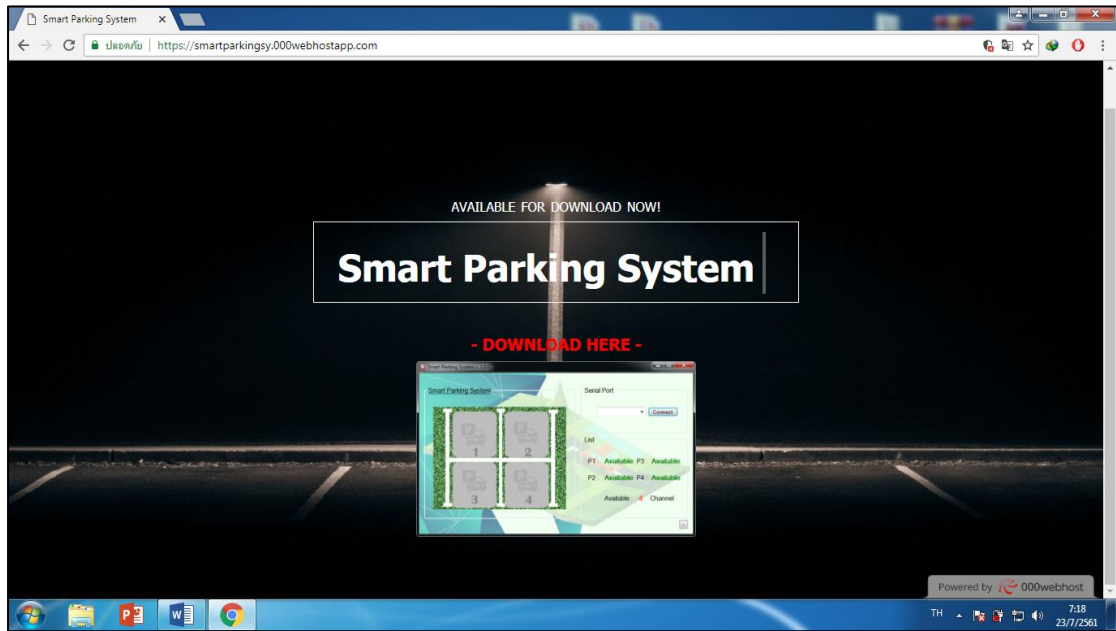
4.1.10 ตั้งค่าพอร์ตที่ใช้ในการรับส่งข้อมูล ดังภาพที่ 4.9



ภาพที่ 4.9 หน้าต่างการตั้งค่าพอร์ต

4.2 ส่วนของการติดตั้งโปรแกรมที่ใช้งาน

4.2.1 ดาวน์โหลดโปรแกรมจากเว็บไซต์ <http://bit.ly/2O54G5U> ดังภาพที่ 4.10



ภาพที่ 4.10 แสดงหน้าเว็บไซต์ดาวน์โหลดโปรแกรม Smart Parking System

4.2.2 คลิกปุ่ม Download Here สีแดงตรงกลางหน้าจอ

4.2.3 เมื่อเสร็จสิ้นการดาวน์โหลดจะได้ไฟล์ชื่อ Setup.msi ดังภาพที่ 4.11



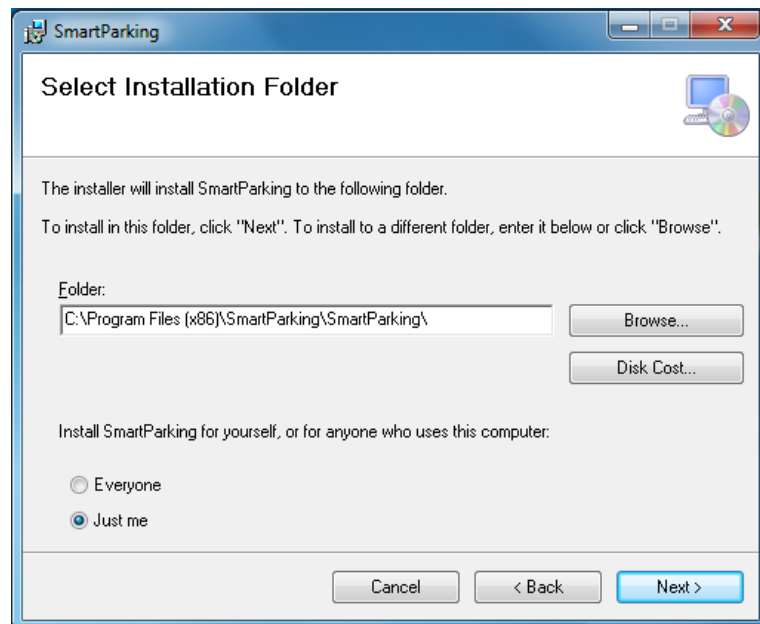
ภาพที่ 4.11 ไฟล์ติดตั้งโปรแกรม Setup.msi

4.2.4 เมื่อเปิดไฟล์ Setup.msi จะปรากฏขั้นตอนการติดตั้งให้คลิก Next เพื่อดำเนินการสู่ขั้นตอนต่อ ดังภาพที่ 4.12



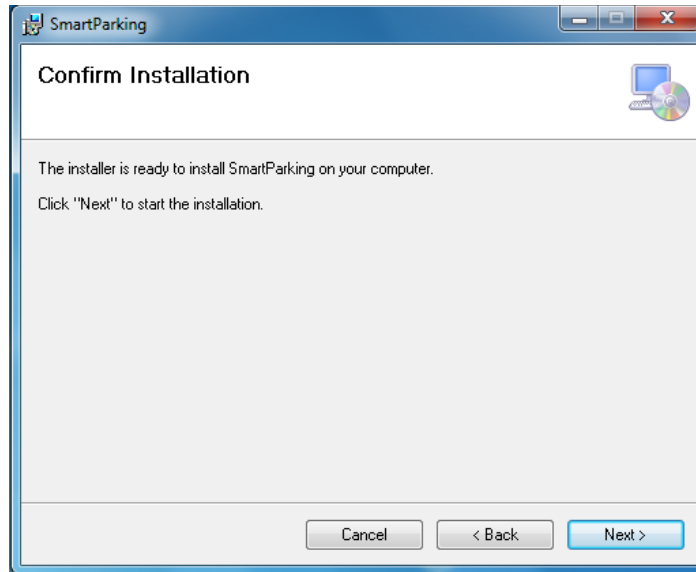
ภาพที่ 4.12 แสดงหน้าต่างเริ่มขั้นตอนติดตั้งโปรแกรม

4.2.5 หลังจากคลิก Next แล้วจะพบหน้าจอ Select Installation Folder ให้ผู้ใช้งานลงตาม Folder ดังภาพห้ามผู้ใช้งานเปลี่ยนแปลง เลือก Just me หรือ Everyone ตามต้องการ หลังจากนั้นคลิกปุ่ม Next ดังภาพที่ 4.13



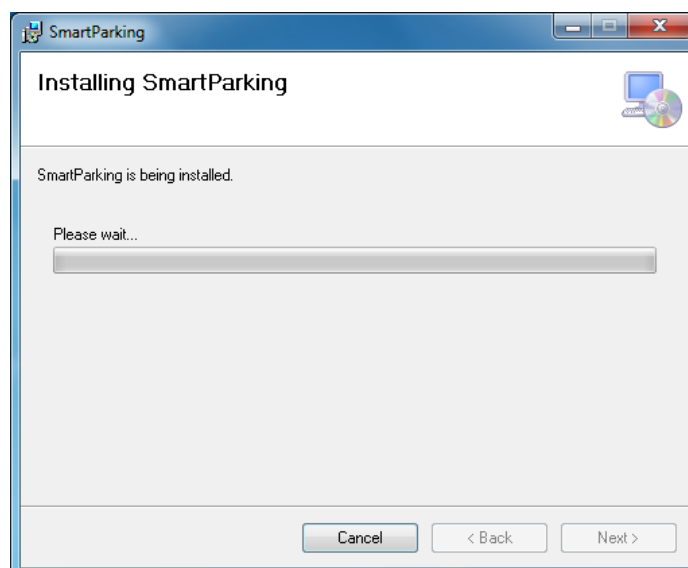
ภาพที่ 4.13 แสดงหน้าจอ Select Installation Folder

4.2.6 หลังจากคลิก Next แล้วจะพบหน้าต่าง Confirm Installation เพื่อให้ผู้ใช้ยืนยันการติดตั้งอีกครั้ง หลังจากนั้นคลิก Next ดังภาพที่ 4.14



ภาพที่ 4.14 แสดงหน้าจอ Confirm Installation

4.2.7 หลังจากผู้ใช้งาน Confirm Installation เรียบร้อยแล้ว จะพบหน้าต่าง Installing รอสักครู่เพื่อทำการติดตั้งโปรแกรม ดังภาพที่ 4.15



ภาพที่ 4.15 แสดงหน้าจอ Installing Program

4.2.8 หลังจากติดตั้งโปรแกรมเสร็จเรียบร้อยแล้วเปิดโปรแกรมจะได้หน้าต่าง ดังภาพที่ 4.16

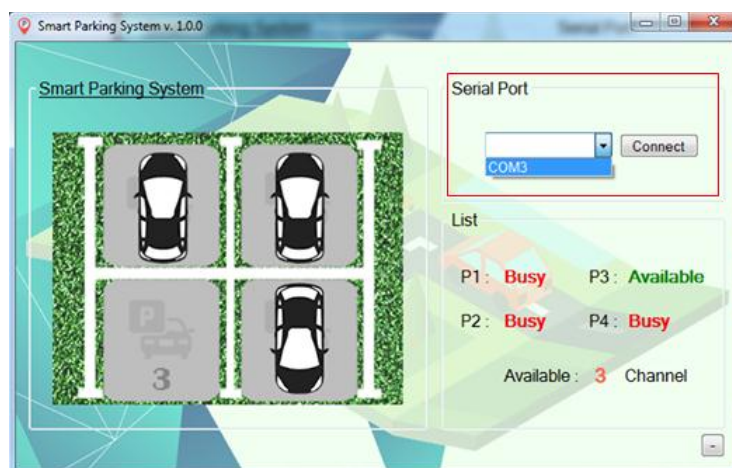


ภาพที่ 4.16 แสดงหน้าจอโปรแกรม

4.3 ส่วนของโปรแกรม

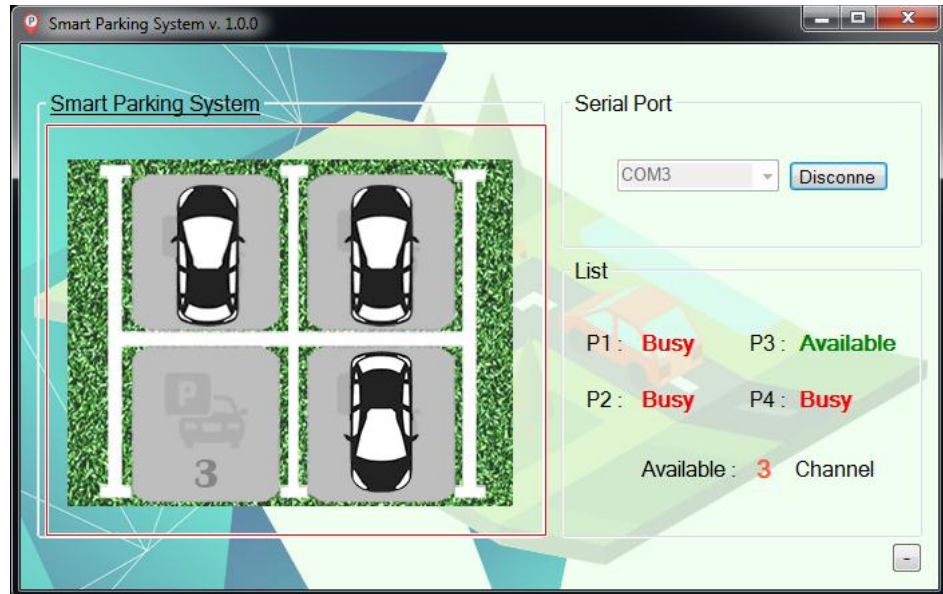
การพัฒนาโปรแกรมตรวจสอบสถานะที่จอดรถเพื่อให้สะดวกของผู้ใช้บริการหรือพนักงานดูแล เพื่อให้เกิดความรวดเร็วในการตรวจสอบช่องจอดรถ เป็นต้น

1. หน้าต่างของโปรแกรมจะประกอบไปด้วย ช่องเลือก Serial Port เพื่อเลือกเชื่อมต่อกับ Hardware ดังภาพที่ 4.17



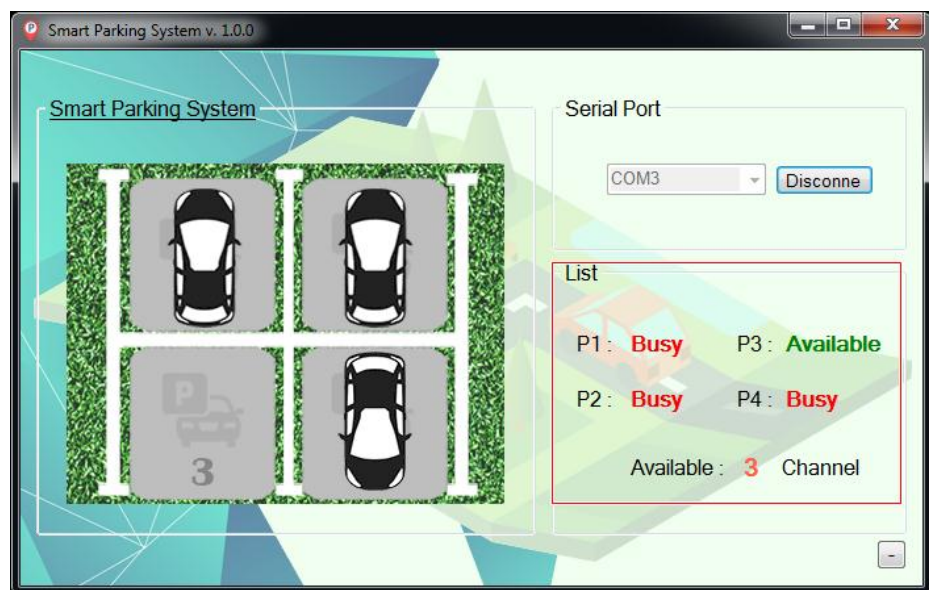
ภาพที่ 4.17 แสดงหน้าจอเลือก Serial Port

2. หน้าต่างโปรแกรมจะประกอบไปด้วยส่วนของแสดงภาพช่องจอดรถ เพื่อตรวจสอบสถานะว่าง-ไม่ว่าง ดังภาพที่ 4.18



ภาพที่ 4.18 แสดงหน้าต่างจำนวนช่องที่ว่าง-ไม่ว่าง

3. หน้าต่างโปรแกรมจะประกอบไปด้วยส่วนของตัวอักษร เพื่อแสดงสถานะ ดังภาพที่ 4.19



ภาพที่ 4.19 แสดงสถานะจำนวนช่องที่ว่าง-ไม่ว่าง

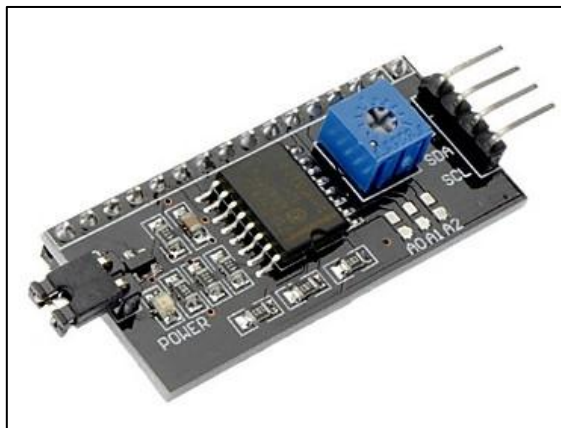
4.3 ส่วนของการติดตั้งฮาร์ดแวร์

4.3.1 ส่วนของการติดตั้งวงจรไฟฟ้าเซ็นเซอร์อัลตราโซนิก

1. ในส่วนของการต่อวงจรเซ็นเซอร์อัลตราโซนิกตัวที่ 1 ขา Trig ต่อเข้ากับ Digital ที่ 22 ขา Echo ต่อเข้ากับ Digital ที่ 24 ที่บอร์ดอาดูโน
2. ในส่วนของการต่อวงจรเซ็นเซอร์อัลตราโซนิกตัวที่ 2 ขา Trig ต่อเข้ากับ Digital ที่ 26 ขา Echo ต่อเข้ากับ Digital ที่ 28 ที่บอร์ดอาดูโน
3. ในส่วนของการต่อวงจรเซ็นเซอร์อัลตราโซนิกตัวที่ 3 ขา Trig ต่อเข้ากับ Digital ที่ 32 ขา Echo ต่อเข้ากับ Digital ที่ 34 ที่บอร์ดอาดูโน
4. ในส่วนของการต่อวงจรเซ็นเซอร์อัลตราโซนิกตัวที่ 4 ขา Trig ต่อเข้ากับ Digital ที่ 36 ขา Echo ต่อเข้ากับ Digital ที่ 38 ที่บอร์ดอาดูโน
5. ในส่วนของขา Vcc และ Gnd ได้ทำการแยกแหล่งจ่ายไฟให้กับบอร์ดออกจาก Adapter Output 5V เพื่อให้ไฟไปเลี้ยงบอร์ดอย่างเพียงพอ

4.3.1 ส่วนของการติดตั้งวงจรไฟฟ้าเซ็นเซอร์อัลตราโซนิก

1. การต่อโมดูลหน้าจอแสดงผล LCD ใช้โมดูล I2C Interface ช่วยเพื่อสามารถเชื่อมต่อจอต้วยสายไฟเพียงแค่ 2 เส้น เพื่อให้สะดวกในการใช้งาน ดังภาพที่ 4.20

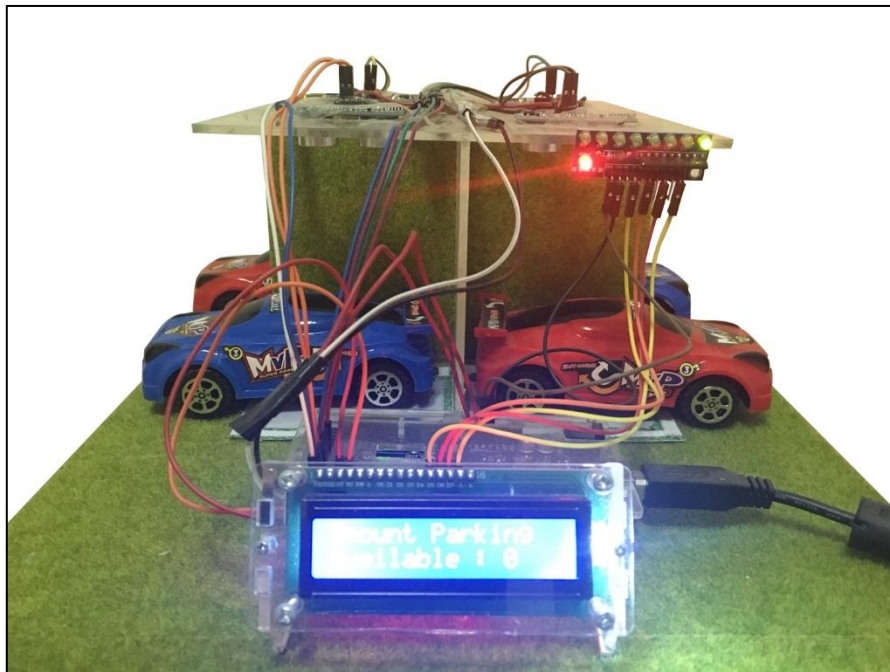


ภาพที่ 4.20 แสดงภาพ Module I2C Interface

2. ในส่วนของการต่อวงจรจอแสดงผล ขา Vcc ต่อเข้ากับ แหล่งจ่ายไฟ 5V และขา Gnd ต่อเข้ากับ Gnd ที่บอร์ด
3. ในส่วนของการต่อวงจรแสดงผล ขา SDA ต่อเข้ากับ SDA20 และขา SCL ต่อเข้ากับ SCL21 ที่บอร์ด

4.3.1 ส่วนของการติดตั้งวงจรไฟฟ้าหลอดไฟแสดงสถานะ

1. การต่อวงจรไฟแสดงผล ไฟแสดงผลดวงที่ 1 ต่อเข้ากับ PWM 2 ที่บอร์ด
2. การต่อวงจรไฟแสดงผล ไฟแสดงผลดวงที่ 2 ต่อเข้ากับ PWM 3 ที่บอร์ด
3. การต่อวงจรไฟแสดงผล ไฟแสดงผลดวงที่ 3 ต่อเข้ากับ PWM 4 ที่บอร์ด
4. การต่อวงจรไฟแสดงผล ไฟแสดงผลดวงที่ 4 ต่อเข้ากับ PWM 5 ที่บอร์ด



ภาพที่ 4.21 แสดงภาพแบบจำลองลานจอดรถยนต์

บทที่ 5

บทสรุปและข้อเสนอแนะ

อภิปรายและสรุปผลการดำเนินโครงการ

โปรแกรมตรวจสอบสถานะลานจอดรถยนต์ มีความสามารถในการแสดงข้อมูลจำนวนช่องจอดรถยนต์ ที่ว่างอยู่ ณ ขณะนั้นได้ และมีแบบจำลองลานจอดรถยนต์ที่มีจอแสดงจำนวนช่องจอดรถได้อย่างชัดเจนและถูกต้อง เพื่อให้ผู้ใช้สามารถดูจำนวนช่องว่างได้ และมีไฟแสดงสถานะแต่ละช่องได้อย่างชัดเจน เป็นการอำนวยความสะดวกในการเข้าใช้บริการ และทราบถึงจำนวนที่ว่างได้อย่างถูกต้อง จึงทำให้โปรแกรมตรวจสอบสถานะลานจอดรถยนต์มีประสิทธิภาพแก่ผู้ใช้ได้มากที่สุด

สรุปผลการดำเนินโครงการ

การพัฒนาโปรแกรมตรวจสอบสถานะลานจอดรถ เพื่อลดระยะเวลาการตรวจสอบหาที่จอดรถ และเพื่อการใช้งานโปรแกรมที่เรียบง่าย จากผลการพัฒนาโปรแกรมสามารถสรุปผลได้ดังนี้

1. เพื่อให้ผู้ใช้สามารถตรวจสอบจำนวนช่องว่างได้ผ่านจอแสดงผล
2. เพื่อให้ปรับปรุงการทำงานและนำคอมพิวเตอร์มาใช้ให้เกิดประโยชน์
3. สามารถแสดงไฟสถานะได้
4. เพื่อให้เกิดการเป็นระเบียบในการจอดรถยนต์

ประโยชน์ที่ได้รับจากการศึกษาเอกเทศ

1. ได้รับความรู้และความเข้าใจในการวิเคราะห์ออกแบบระบบ
2. เรียนรู้วิธีการเขียนโปรแกรมด้วยภาษา C และภาษา Basic
3. ได้รู้จักการทำงานที่มีการวางแผนและเป็นระบบมากขึ้น
4. ได้เรียนรู้การแก้ไขปัญหาด้วยตนเอง
5. ได้รับความรู้และความเข้าใจในเรื่องของไมโครคอนโทรลเลอร์
6. ได้เรียนรู้ในสิ่งใหม่ ๆ เช่น การจำลองแบบจำลองลานจอดรถยนต์
7. ได้เรียนรู้เกี่ยวกับการต่อวงจรไฟฟ้าเบื้องต้น

ปัญหาอุปสรรคและแนวทางแก้ไข

ปัญหาอุปสรรคและแนวทางแก้ไขจาก การวิเคราะห์ออกแบบโปรแกรมตรวจสอบสถานะลานจอดรถยนต์ ผู้ศึกษาพบว่า โปรแกรมตรวจสอบสถานะลานจอดรถยนต์สามารถทำงานได้อย่างมีประสิทธิภาพ แต่เนื่องจากผู้จัดทำยังประสบกับปัญหาและอุปสรรคบางประการซึ่งพอสรุปได้ดังนี้ เนื่องจากโปรแกรมตรวจสอบสถานะลานจอดรถยนต์เป็นการพัฒนาครั้งแรกแล้วยังไม่มีการใช้งานมาก่อน จึงทำให้เป็นปัญหากับผู้ใช้งาน และการดำเนินของผู้พัฒนาโปรแกรมยังไม่ดีมากนักเนื่องจากยังขาดประสบการณ์ในด้านการเขียนโปรแกรม

ข้อเสนอแนะและแนวทางการพัฒนา

1. ควรเพิ่มฐานข้อมูลและคำนวณเวลารถยนต์จอดในแต่ละช่อง ให้ทราบถึงระยะเวลาในการจอดรถยนต์
2. แบบจำลองลานจอดรถยนต์ควรที่จะส่งข้อมูลผ่านระบบเครือข่ายไร้สายเพื่อที่จะไม่ต้องเชื่อมต่อโดยพอร์ตอนุกรม

บรรณานุกรม

นายณัฐวัฒน์ เรืองทอง "ระบบจัดการลานจอดรถ" ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์, 2555

นายพงษ์ประภัทร ชูหิรัญญวัฒน์ และคณะ “การศึกษาและออกแบบ แบบจำลองลานจอดรถอัจฉริยะ” โครงการวิศวกรรมโทรคมนาคม หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี, 2557

เดชวิทย์รัศมี และ ธงรบ อักษร “ระบบการจัดการลานจอดรถยนต์ ด้วยไมโครคอนโทรลเลอร์” สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยีสารสนเทศ มหาวิทยาลัยราชภัฏพิบูลสงคราม, 2560

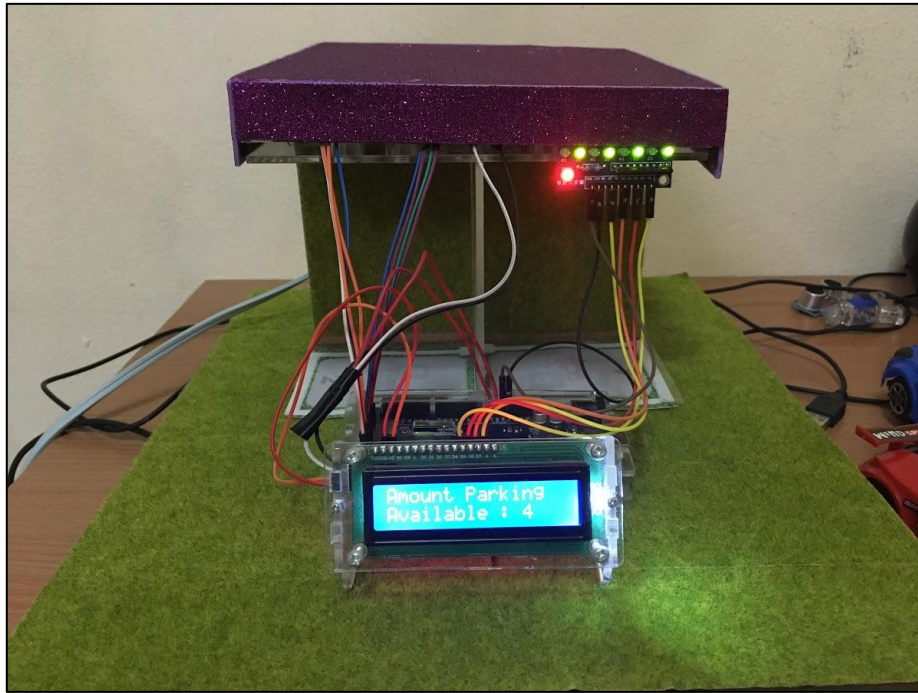
บทความตัวอย่างการใช้งาน เซนเซอร์ วัดระยะทาง HC-SR05 [ออนไลน์].

เข้าถึงได้จาก : <http://www.myarduino.net>

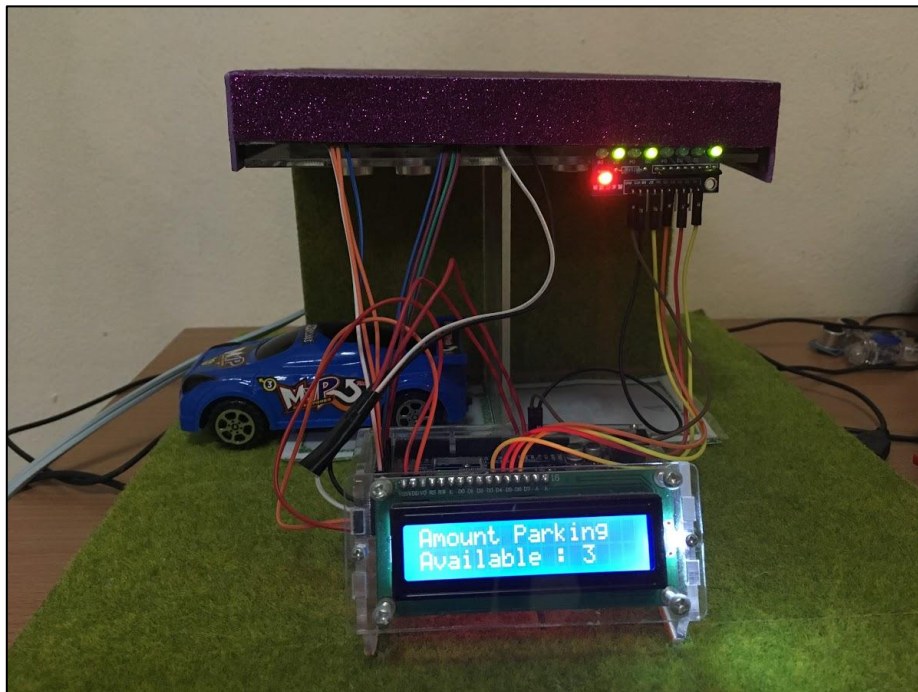
บทความทำความรู้จัก Arduino รุ่นต่าง ๆ [ออนไลน์].

เข้าถึงได้จาก : <http://bit.ly/2LGZHql>

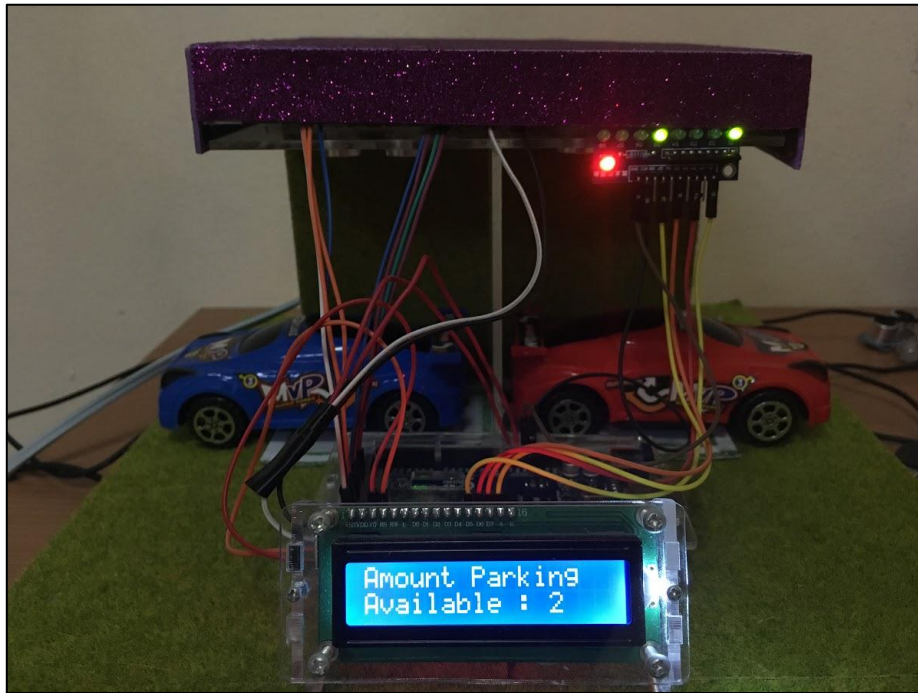
ภาคผนวก



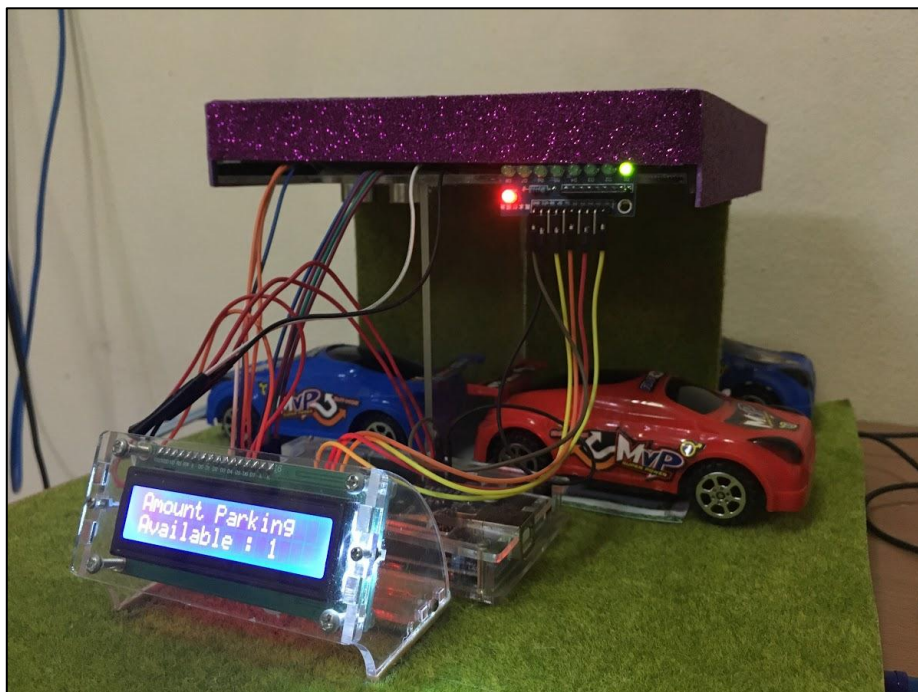
ภาพที่ 1 แบบจำลองลานจอดรถยนต์



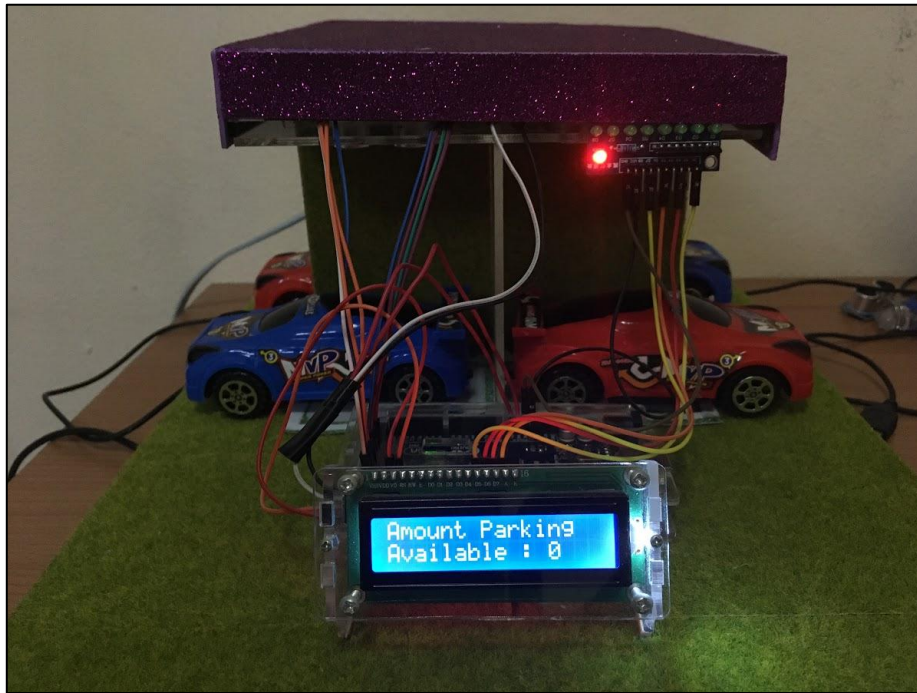
ภาพที่ 2 แบบจำลองลานจอดรถยนต์



ภาพที่ 3 แบบจำลองลานจอดรถยนต์



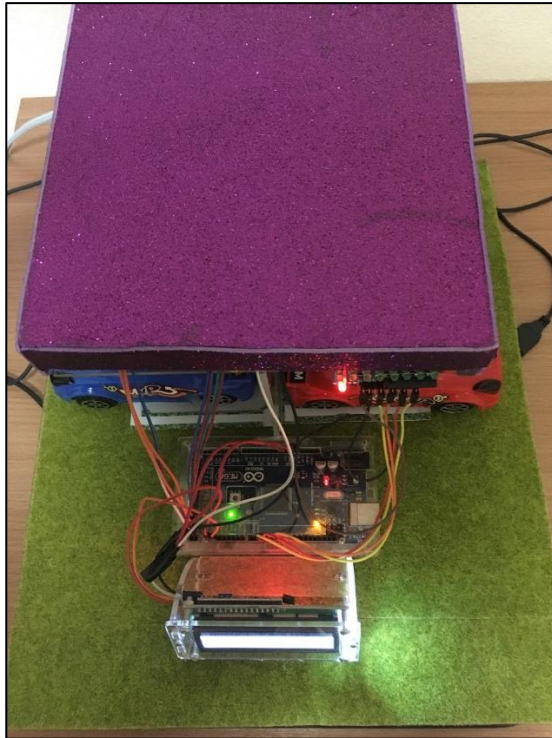
ภาพที่ 4 แบบจำลองลานจอดรถยนต์



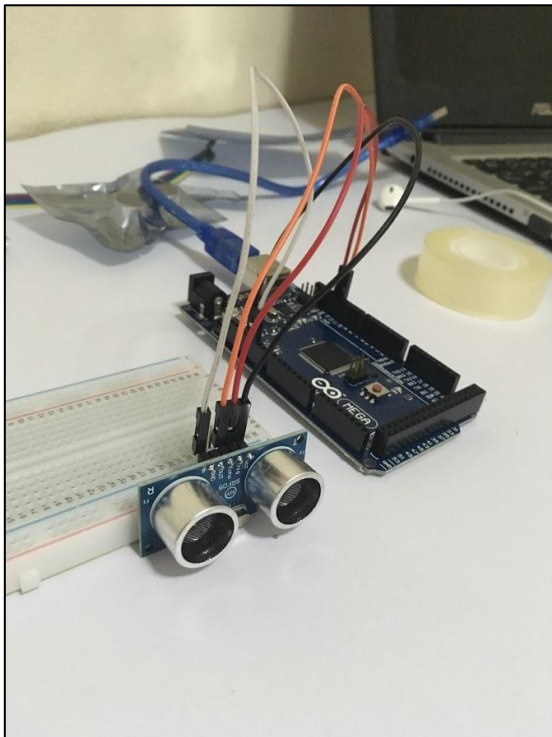
ภาพที่ 5 แบบจำลองลานจอดรถยนต์



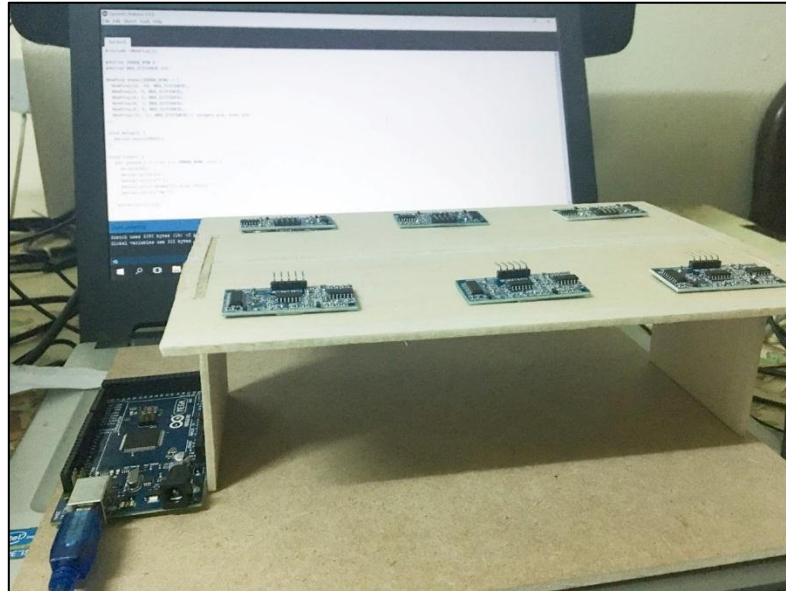
ภาพที่ 6 แบบจำลองลานจอดรถยนต์



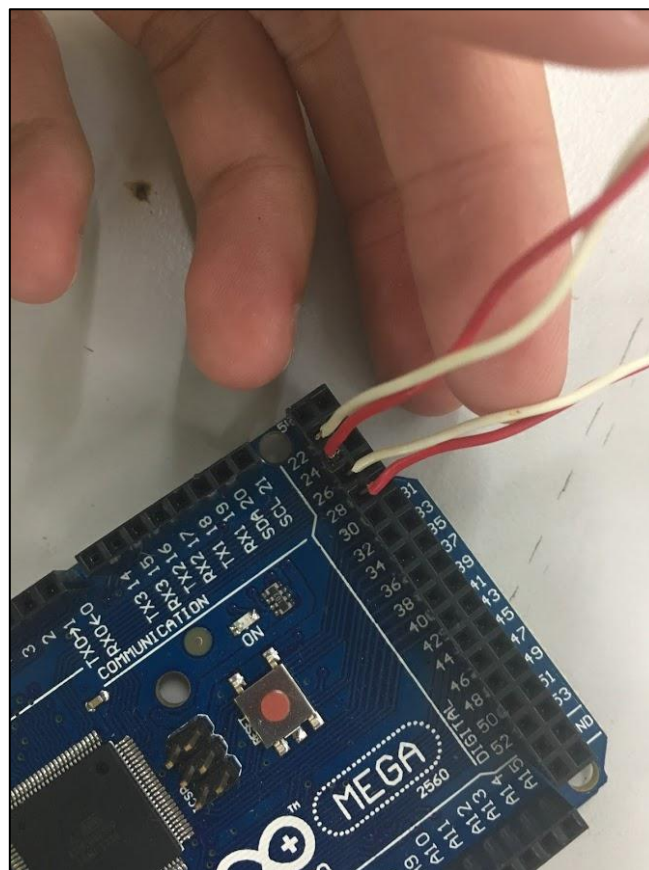
ภาพที่ 7 แบบจำลองสถานจอดรถยนต์



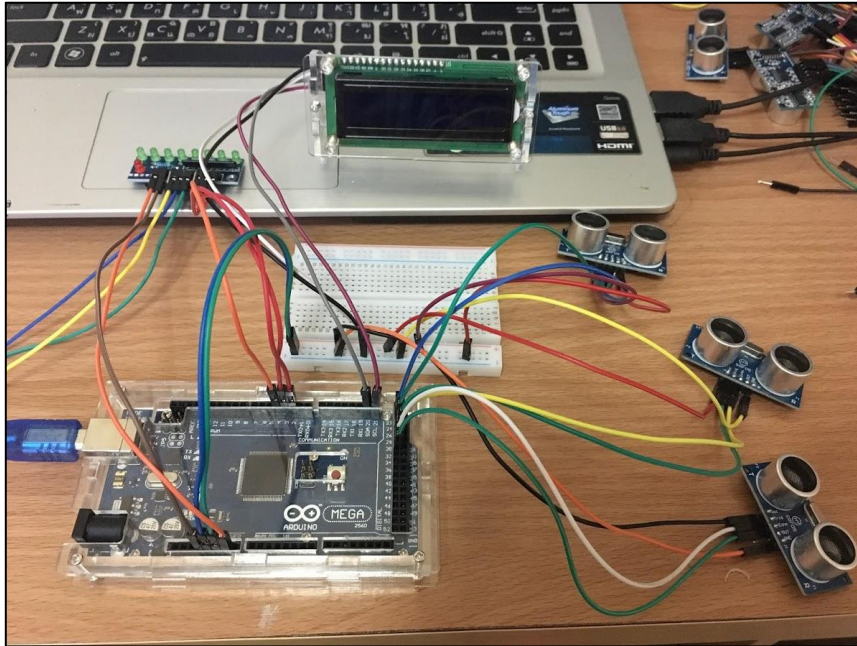
ภาพที่ 8 ทดสอบการต่อเซ็นเซอร์อัลตราโซนิก



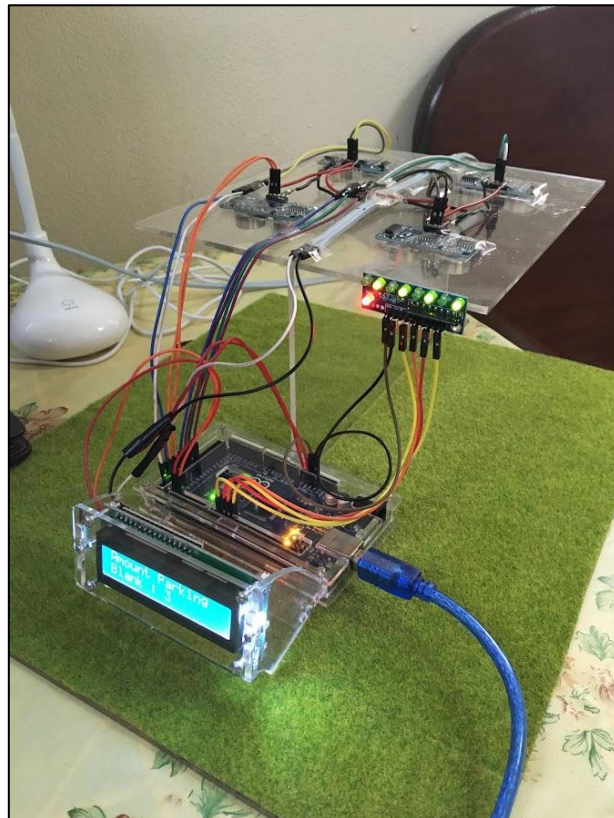
ภาพที่ 9 การออกแบบ แบบจำลองลานจอดรถ



ภาพที่ 10 ทดสอบการต่อเซ็นเซอร์อัลตราโซนิก



ภาพที่ 11 ทดสอบการต่อเซ็นเซอร์อัลตราโซนิก ไฟแสดงผล และจอ LCD



ภาพที่ 11 ประกอบโครงสร้างแบบจำลองลานจอดรถยนต์

