โรงจอดรถจักรยานยนต์อัจฉริยะ
SMART MOTORBIKE PARKING

เทพสิทธิ์ ชัยเบญจกัลยาณี
ปฐวี เปียคล้าย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

โรงจอดรถจักรยานยนต์อัจฉริยะ
SMART MOTORBIKE PARKING

เทพสิทธิ์ ชัยเบญจกัลยาณี
ปฐวี เปียคล้าย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART MOTORBIKE PARKING

TEPPASIT CHAIBENJAKALLAYANEE
PATAVEE PEAKLAI

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2561

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โรงจอดรถจักรยานยนต์อัจฉริยะ

SMART MOTORBIKE PARKING

ผู้จัดทำ นาย เทพสิทธิ์ ชัยเบญจกัลยาณี 58010480

นาย ปฐวี เปียคล้าย 58010715

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.รัชนี กุลยานนท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โรงจอดรถจักรยานยนต์อัจฉริยะ

โดย
นายเทพสิทธิ์ ชัยเบญจกัลยาณี 58010480
นายปฐวี เปียคล้าย 58010715

อาจารย์ที่ปรึกษา
ผู้ช่วยศาสตราจารย์ ดร.รัชนี กุลยานนท์

ปีการศึกษา 2561

บทคัดย่อ

ปัญหาจำนวนรถที่มีปริมาณมากเป็นปัญหาสำคัญที่ทั่วไปในบริเวณที่มีผู้คนหนาแน่น ซึ่งปัญหานี้เป็นปัญหาที่ส่งผลกระทบกับทุกคน หนึ่งในปัจจัยหลักที่เป็นสาเหตุให้เกิดปัญหาการจราจรก็คือ พื้นที่มีไม่เพียงพอต่อการรองรับยานพาหนะ การจอดรถที่ไม่เป็นระเบียบ การจอดรถในพื้นที่ห้ามจอด ผู้จัดทำจึงอยากนำเทคโนโลยีเข้ามาช่วยในการจัดระเบียบการจอดรถยนต์ และรถจักรยานยนต์ เพื่อเพิ่มคุณภาพในการใช้พื้นที่ในบริเวณที่มีการใช้ยานพาหนะที่หนาแน่น โดยการจำลองอาคารจอดรถอัตโนมัติ ซึ่งใช้ระบบอัตโนมัติในการจอดรถ รับรถ และใช้ระบบอัตโนมัติในการเก็บรถ ชิ้นงานนี้มีวัตถุประสงค์เพื่อศึกษา และพัฒนาระบบอัตโนมัติที่ใช้ในโรงจอดรถ โดยใช้ Arduino ในการควบคุมลิฟท์ยก ใช้โปรแกรม Sketup ออกแบบตัวอาคารเพื่อตอบสนองต่อความสะดวกในการใช้งานของผู้ใช้ ดังนั้นจึงเหมาะในการเรียนรู้ และพัฒนาโดยการวิเคราะห์ และคำนวณด้วยทฤษฎีวิศวกรรม เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดในทุกๆ ด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMART MOTORBIKE PARKING

By
Mr. Teppasit Chaibenjakallayanee 58010480
Mr. Patavee Peaklai 58010715

Advisor
Asst.Porf Dr. Rutchanee Gullayanon

Academic Year 2018

ABSTRACT

Large amount of vehicle is a significant problem in high dense population area which effect everyone daily life. Typical reasons of traffic jam are such as lack of parking area, disorderly parking and prohibited area parking. WE want to use technology to help organize parking for vehicles to increase efficiency of area usage in vehicle crowded area by simulate automatic parking building which parking, pick up and storing vehicle will be automated. This project is intended to study and develop automatic systems in the parking building by using Arduino to control lifting and Sketch up to design the building in response to user convenience so it is suitable for learning and developed by analysis and calculated with engineering theory for the best result in every each way.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์จากอาจารย์ประจำภาควิชาวิศวกรรมการวัดและควบคุม ที่ได้ถ่ายทอดความรู้ทั้งในภาคทฤษฎี และภาคปฏิบัติให้ผู้จัดทำได้นำมาปรับใช้ในการทำงานเพื่อให้การเขียนรายงานค้นคว้าจนออกมาเป็นฉบับที่สมบูรณ์ที่สุด ทางผู้จัดทำขอขอบพระคุณอาจารย์ที่ให้คำแนะนำ และให้คำปรึกษา รวมถึงถ่ายทอดความรู้ประสบการณ์ให้แก่ผู้จัดทำตลอดระยะเวลาที่ได้จัดทำโครงงานนี้ และสามารถนำไปใช้ในการศึกษาต่อในอนาคต โดยเฉพาะอย่างยิ่ง ผู้ช่วยศาสตราจารย์ ดร.รัชนี กุลยานนท์ ผู้เป็นอาจารย์ที่ปรึกษาของโครงงานนี้ โดยท่านอาจารย์ได้ให้คำแนะนำต่างๆ ทั้งในด้านการทำงาน และการแก้ไขปัญหาต่างๆ ที่เกิดขึ้นระหว่างการทำโครงงานนี้ ซึ่งผู้จัดทำสามารถนำมาประยุกต์ใช้ในการทำโครงงานนี้ได้ และขอขอบพระคุณเพื่อนๆ และพี่ๆ ในต่างภาควิชาที่คอยให้คำแนะนำในด้านต่างๆ ที่สามารถนำมาประยุกต์ใช้ในตัวชิ้นงาน และช่วยแก้ปัญหาต่างๆ

ขอขอบพระคุณผู้สร้างเว็บไซต์ และเอกสารอ้างอิงต่างๆ ที่ผู้จัดทำนำมาใช้เป็นข้อมูลในการทำโครงงาน และเป็นแหล่งอ้างอิง

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณครอบครัวที่คอยให้การสนับสนุน และคอยเป็นกำลังใจที่ดีแก่ผู้จัดทำมาโดยตลอด

คณะผู้จัดทำ
นายเทพสิทธิ์ ชัยเบญจกัลยาณี
นายปฐวี เปียคล้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ศึกษาปัญหาที่เกี่ยวข้องกับโครงงาน	1
1.3 สำรวจความคิดเห็นจากนักศึกษาและศิษย์เก่า คณะวิศวกรรมศาสตร์ สจล.	2
1.4 วัตถุประสงค์ของการทำโครงงาน	3
1.5 ขอบเขตของการทำโครงงาน	3
1.6 วิธีดำเนินการทำโครงงาน การจัดทำโครงงาน	3
1.7 ประโยชน์ที่คาดว่าจะได้รับ	4
1.8 รายละเอียดของปริญญานิพนธ์	4
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	5
2.1 ซอฟต์แวร์ที่เกี่ยวข้อง	5
2.1.1 SketUp 2018	5
2.1.2 Visual Studio	6
2.1.3 Arduino IDE	6
2.2 อุปกรณ์ที่เกี่ยวข้อง	6
2.2.1 Stepping Motor	6
2.2.2 Stepping Motor Drive	7
2.2.3 ลิมิตสวิตช์ (Limit Switch)	8
2.2.4 สายพาน (Belt)	8
2.2.5 Switching Power Supply	9
บทที่ 3 วิธีการดำเนินงาน	10
3.1 ออกแบบระบบพื้นฐานที่เกี่ยวข้องกับโครงงาน	10
3.2 ออกแบบโครงสร้างจำลองของอาคารจอดรถ	11
3.3 การเขียนโปรแกรม Parking Control System	13
3.4 การเขียนโปรแกรม Arduino	14
3.5 ทำการเชื่อมอุปกรณ์ที่เกี่ยวข้องเข้าด้วยกัน	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	**16**
4.1 แผนผังการทำงานของอาคารจอดรถอัจฉริยะ	16
4.2 อาคารจำลอง	16
4.3 ตัวลิฟท์	17
4.4 Parking Control System วินโดแอพพลิเคชัน	18
4.5 ผลการนำรถจำลองเข้าจอด	19
บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ	**20**
5.1 สรุปผลการทดลอง	20
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	20
5.3 แนวทางการแก้ไข	20
เอกสารอ้างอิง	**21**
ภาคผนวก	**22**
ภาคผนวก ก Arduino Code	23
ภาคผนวก ข Visual Studio 2017 Code	34
ภาคผนวก ค Arduino Uno R3	49
ภาคผนวก ง Stepping motor NEMA 17	57
ภาคผนวก ง โปสเตอร์	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
1.1 สภาพปัญหาการจอดรถบริเวณอาคารเรียนรวม 12 ชั้น คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	1
1.2 สภาพปัญหาการจอดรถบริเวณหน้าภาควิชาวิศวกรรมเครื่องกล สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	2
1.3 ผลจากการทำแบบสอบถามความเห็นในการจอดรถของนักศึกษาคณะวิศวกรรมศาสตร์	2
2.1 โปรแกรม SketUp 2018	5
2.2 หน้าต่างขณะใช้งานโปรแกรม SketUp 2018	5
2.3 โปรแกรม Visual Studio 2017	6
2.4 โปรแกรม Arduino IDE	6
2.5 Stepping Motor รุ่น NEMA17	6
2.6 ภาพโครงสร้าง Stepping Motor	7
2.7 Stepping Motor Drive รุ่น TB6600	7
2.8 แผงผังการทำงานของ Stepping Motor	8
2.9 ลิมิสวิตช์ (Limit Switch)	8
2.10 สายพาน	8
2.11 Switching Power Supply	9
2.12 แผนผังการทำงานของ Switching Power Supply	9
3.1 Block Diagram แสดงอุปกรณ์และหลักการทำงานของระบบ	10
3.2 Sequence Diagram แสดงลำดับการทำงานของระบบ	11
3.3 แบบของโครงสร้างจำลองมุม Perspective View	11
3.4 แบบของโครงสร้างจำลองมุม Front View	12
3.5 รูปแสดงโครงสร้างและระยะห่างของช่องจอดรถ	12
3.6 รูปแสดงระหว่างการเขียนโปรแกรม Parking Control System (1)	13
3.7 รูปแสดงระหว่างการเขียนโปรแกรม Parking Control System (2)	13
3.8 รูปแสดงระหว่างการเขียนโปรแกรมควบคุม Arduino (1)	14
3.9 รูปแสดงระหว่างการเขียนโปรแกรมควบคุม Arduino (2)	14
3.10 แผนผังแสดงการเชื่อมต่ออุปกรณ์ไฟฟ้า และโปรแกรมที่เขียนขึ้นมา	15
4.1 แผนผังการทำงานของอาคารจอดรถอัจฉริยะ	16
4.2 อาคารจอดรถจำลอง	16
4.3 รูปจากด้านบนของลิฟท์ยกรถ	17
4.4 รูปจากด้านฐานของลิฟท์ยกรถ	17
4.5 หน้าตาของโปรแกรม Parking Control System พร้อมรายละเอียด	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 ตารางผลการทดลองนำเข้ารถเข้าจอดครั้งที่ 1	19
4.2 ตารางผลการทดลองนำเข้ารถเข้าจอดครั้งที่ 2	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

เนื่องจากปัจจุบันพื้นที่ในกรุงเทพมหานคร ปริมณฑล หรือตามเมืองใหญ่มีขนาดจำกัดที่ใช้ในการจอดรถจักรยานยนต์ จำนวนรถจักรยายนต์มีจำนวนเพิ่มขึ้นอย่างต่อเนื่องในทุกๆ ปี และปัญหาอาชญากรรมมีมากขึ้นในหลายรูปแบบ

โครงงานนี้จึงมีความคิดที่จะนำนวัตกรรม และเทคโนโลยี เข้ามาแก้ไขปัญหา เพื่อตอบสนองต่อการใช้งานที่เพิ่มขึ้น โดยใช้พื้นที่ที่มีอย่างจำกัดให้เกิดประโยชน์สูงสุด พร้อมทั้งอำนวยความสะดวกแก่ผู้ที่ใช้งานโดยให้สอดคล้องกับไทยแลนด์ 4.0

1.2 ศึกษาปัญหาที่เกี่ยวข้องกับโครงงาน

ขั้นตอนนี้เป็นการศึกษาปัญหาที่อยู่รอบตัว และเกี่ยวข้องกับคนส่วนใหญ่ในสังคม จากการลงพื้นที่การศึกษาปัญหาในคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พบว่าปัญหาที่เกิดขึ้นนั้นคือ พื้นที่จอดรถจักรยานยนต์ไม่เพียงพอต่อความต้องการ การจอดที่ไม่เป็นระเบียบ และเสี่ยงให้เกิดอาชญากรรมต่างๆ ไม่ว่าจะเป็นการโจรกรรมหรือปัญหาการชนแล้วหนี เป็นต้น อีกทั้งยังเป็นการใช้พื้นที่อย่างไม่เกิดประสิทธิภาพอีกด้วยดังรูปที่ 1.1 และรูปที่ 1.2

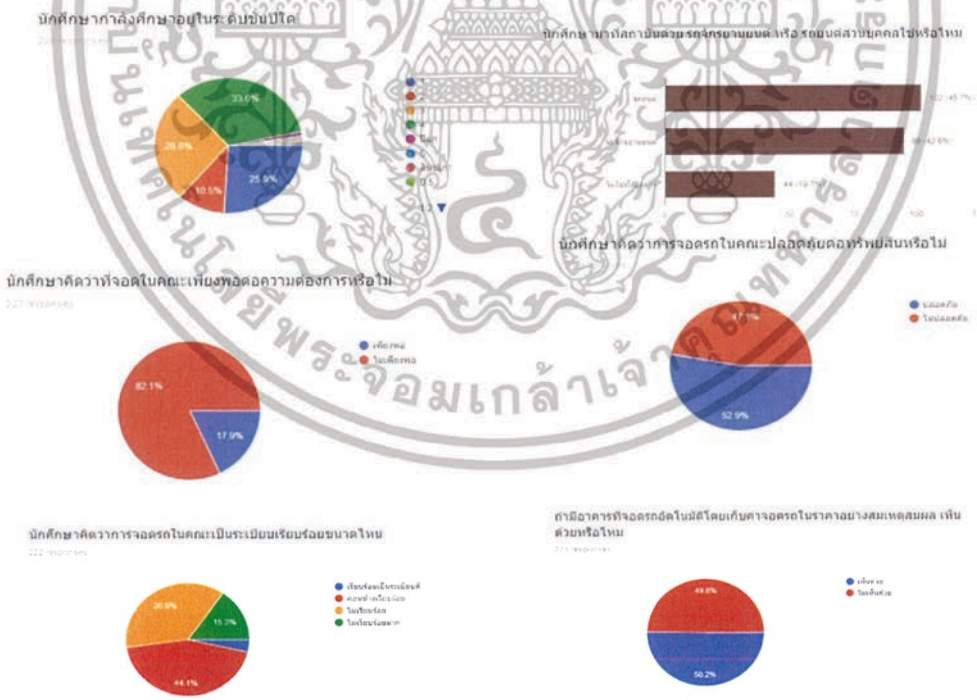
รูปที่ 1.1 สภาพปัญหาการจอดรถบริเวณอาคารเรียนรวม 12 ชั้น คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1.2 สภาพปัญหาการจอดรถบริเวณหน้าภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1.3 สำรวจความคิดเห็นจากนักศึกษาและศิษย์เก่า คณะวิศวกรรมศาสตร์ สจล.

ขั้นตอนนี้เป็นการสำรวจความเห็นจากนักศึกษาในสถาบันฯ โดยใช้ Google Forms เป็นเครื่องมือในการทำแบบสอบถามต่อการจอดรถจักรยานยนต์ ปัญหาที่เกิดขึ้น และเสนอทางเลือกด้วยการสร้างโรงจอดรถจักรยานยนต์เพื่อนำมาทำเป็นโครงงาน Smart Motorbike Parking เพื่อเป็นต้นแบบในอนาคตดังรูปที่ 1.3



รูปที่ 1.3 ผลจากการทำแบบสอบถามความเห็นในการจอดรถของนักศึกษาคณะวิศวกรรมศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วัตถุประสงค์ของการทำโครงงาน

1. เพื่อศึกษาปัญหาของการจอดรถจักรยานยนต์ และปัญหาในการใช้บริการของลานจอดรถ

2. เพื่อศึกษาหลักการออกแบบโครงสร้างจำลองของโรงจอดรถจักยานยนต์ โดยใช้ทรัพยากรอย่างเกิดคุณค่ามากที่สุด โดยใช้โปรแกรม SketUp 2018

3. เพื่อศึกษาหลักการออกแบบ และสร้างวินโดแอพพลิเคชัน Smart Motorbike Parking โดยการใช้โปรแกรม Visual Studio 2017

4. เพื่อศึกษาหลักการออกแบบ และสร้างระบบควบคุมอัตโนมัติของอาคารจอดรถจักรยานยนต์

5. เพื่อศึกษาการเลือกใช้อุปกรณ์ทางไฟฟ้าให้เหมาะสมตามโครงงานที่จัดทำ

1.5 ขอบเขตของการทำโครงงาน

1. ออกแบบและสร้างโครงสร้างจำลองของอาคารจอดรถจักรยานยนต์อัจฉริยะ
2. ออกแบบและสร้างวินโดแอพพลิเคชัน Smart Motorbike Parking
3. ออกแบบและเขียนโปรแกรมระบบควบคุมของลิฟท์อาคารจอดรถ
4. ออกแบบระบบไฟฟ้า และตู้ควบคุม
5. จัดซื้ออุปกรณ์ทางไฟฟ้า และประกอบตู้ควบคุมตามที่ออกแบบไว้
6. ทำการเชื่อมต่ออุปกรณ์ต่างๆ และวินโดแอพพลิเคชันให้ทำงานด้วยกันตามที่ออกแบบ

1.6 วิธีดำเนินการทำโครงงาน การจัดทำโครงงาน

เริ่มต้นจากการศึกษาปัญหาการจอดรถจักรยานยนต์ในคณะวิศวกรรมศาสตร์ และสำรวจความเห็นจากนักศึกษาในสถาบันฯ จากนั้นนำปัญหาที่ศึกษามาออกแบบเป็นโครงงาน Smart Motorbike Parking เพื่อเป็นต้นแบบในการจัดการกับปัญหา โดยแนวคิดที่จะใช้เทคโนโลยีระบบอัตโนมัติ เพื่อให้สะดวกสบายต่อผู้ใช้งาน

นำแนวคิดที่ได้มาออกแบบตัววินโดแอพพลิเคชันด้วยโปรแกรม Visual Studio 2017 ออกแบบโครงสร้างจำลองของอาคารจอดรถด้วยโปรแกรม SketUp 2018 ออกแบบโปรแกรมระบบควบคุม Arduino ระหว่างการออกแบบส่วนต่างๆ จะต้องทำการค้นหาข้อมูลของอุปกรณ์ต่างๆ เพื่อให้เหมาะสมกับการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. เป็นต้นแบบในการแก้ปัญหาการจอดรถจักรยานยนต์ในพื้นที่ที่มีพื้นที่น้อย
2. มีความรู้เกี่ยวกับการออกแบบ และเขียนโปรแกรมระบบควบคุม Arduino และนำไปต่อยอดในอนาคต
3. มีความรู้เกี่ยวกับการออกแบบ และพัฒนาแอพพลิเคชัน
4. มีความรู้เกี่ยวกับการเลือกซื้ออุปกรณ์ไฟฟ้า

1.8 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย 5 บท และ 5 ภาคผนวก ซึ่งมีรายละเอียดดังต่อไปนี้

บทที่ 1 บทนำ เป็นการกล่าวถึงความเป็นมาและที่มาของปริญญานิพนธ์ วัตถุประสงค์ของการทำโครงงาน ขอบเขตของการทำโครงงาน ขั้นตอนการดำเนินงาน ประโยชน์ที่คาดว่าจะได้รับ และรายละเอียดของปริญญานิพนธ์

บทที่ 2 อุปกรณ์ และซอฟต์แวร์ที่เกี่ยวข้องในการออกแบบ เป็นการเพิ่มเติมความรู้ความเข้าใจในซอฟต์แวร์อุปกรณ์ต่างๆ ก่อนจะเริ่มการทำโครงงาน

บทที่ 3 ขั้นตอนการดำเนินงาน เป็นการอธิบายขั้นตอนการทำชิ้นงานโดยละเอียดทั้งในส่วนของฮาร์ดแวร์ และซอฟต์แวร์

บทที่ 4 ผลการดำเนินงาน เป็นการแสดงผลการใช้งานของอาคารจำลองจอดรถ และระบบที่ออกแบบ

บทที่ 5 ผลสรุป และข้อเสนอแนะ เป็นบทสรุปภาพรวมของชิ้นงานรวมถึงสิ่งที่จะพัฒนาต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและความรู้ที่เกี่ยวข้อง

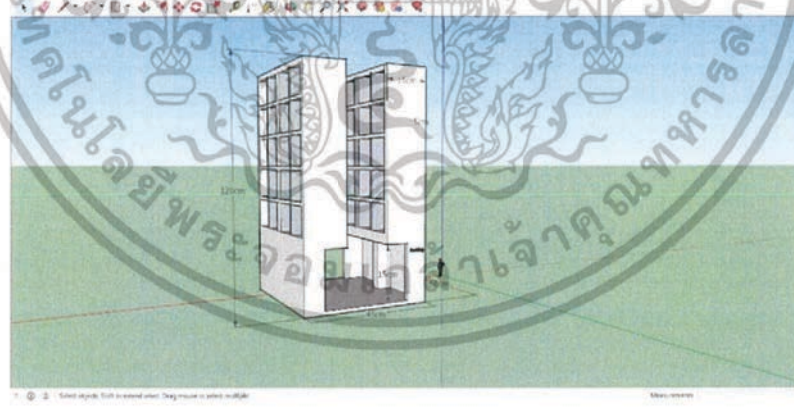
สำหรับบทนี้จะกล่าวถึงทฤษฎี และความรู้ต่างๆ ที่เกี่ยวข้องกับการพัฒนาวินโดแอพพลิเคชัน และอุปกรณ์ที่เกี่ยวข้องได้ทำการค้นคว้า และศึกษาข้อมูลจากแหล่งต่างๆ โดยแบ่งรายละเอียดเป็นหัวข้อดังต่อไปนี้

2.1 ซอฟต์แวร์ที่เกี่ยวข้อง

2.1.1 SketUp 2018

เป็นโปรแกรมที่ใช้ในการออกแบบโครงสร้างจำลองของอาคารจอดรถ Smart Motorbike Parking โดยตัวโปรแกรมสามารถมองรอบวัตถุได้รอบ 360 องศา ทำให้ผู้ใช้สามารถมองเห็นได้ แก้ไขหากมีข้อผิดพลาดในการออกแบบ นอกจากนั้นยังมีเครื่องมือในการอำนวยความสะดวกต่อผู้ใช้ดังรูปที่ 2.1 และรูปที่ 2.2

รูปที่ 2.1 โปรแกรม SketUp 2018



รูปที่ 2.2 หน้าต่างขณะใช้งานโปรแกรม SketUp 2018

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 Visual Studio

เป็นโปรแกรมตัวหนึ่งที่เป็นเครื่องมือที่ช่วยพัฒนาซอฟต์แวร์ และระบบต่างๆ ซึ่งใช้พัฒนาวินโดแอพพลิเคชันทำงานติดต่อสื่อสารพูดคุยสั่งงานกับบอร์ด Arduino ดังรูปที่ 2.3

รูปที่ 2.3 โปรแกรม Visual Studio 2017

2.1.3 Arduino IDE

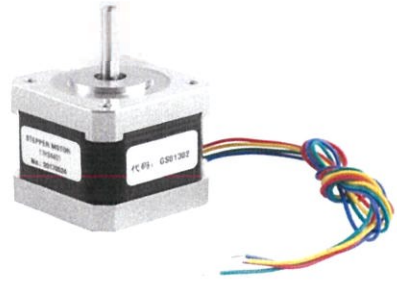
เป็นโปรแกรมที่ใช้สำหรับเขียนโปรแกรมสำสั่งการทำงานต่างๆ ให้กับบอร์ดไมโครคอนโทรลเลอร์ Arduino เพื่อรับค่าจากเซนเซอร์ และควบคุมอุปกรณ์ต่างๆ ในระบบดังรูปที่ 2.4

รูปที่ 2.4 โปรแกรม Arduino IDE

2.2 อุปกรณ์ที่เกี่ยวข้อง

2.2.1 Stepping Motor

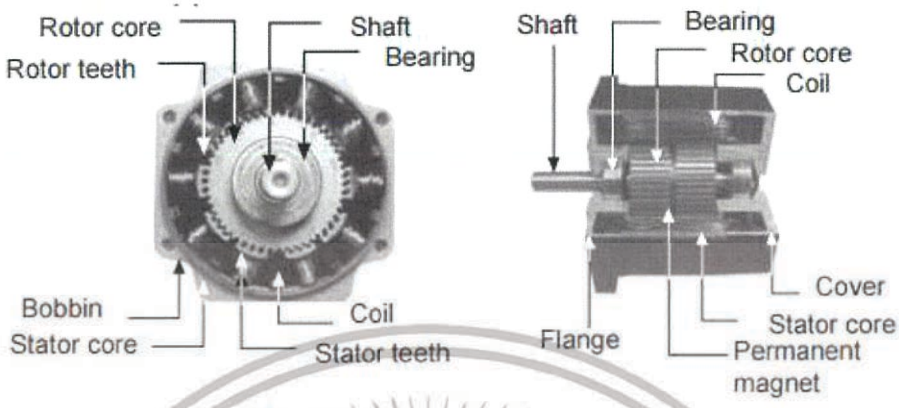
Stepping Motor หรือ Stepper Motor เป็นมอเตอร์ไฟฟ้าที่ขับเคลื่อนด้วยพัลส์ โดยโครงสร้างภายในนั้นจะประกอบไปด้วยขั้วแม่เหล็กบนสเตเตอร์ (Stator) ทำมาจากแผ่นเหล็กวงแหวน



รูปที่ 2.5 Stepping Motor รุ่น NEMA17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีซี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยแต่ละซี่ที่ยื่นออกมานั้นจะมีขดลวด (คอยล์) พันอยู่ เมื่อมีกระแสผ่านคอยล์จะเกิดสนามแม่เหล็กไฟฟ้าขึ้นดังรูปที่ 2.5 และรูปที่ 2.6



รูปที่ 2.6 ภาพโครงสร้าง Stepping Motor

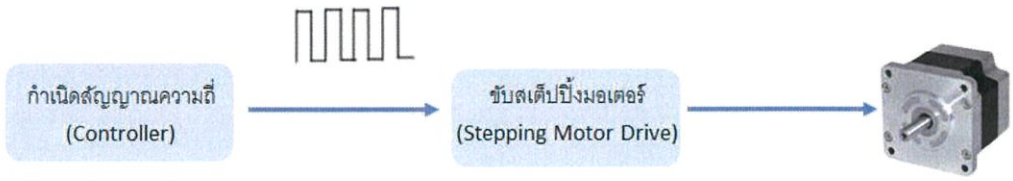
ในการทำงานของ Stepping Motor หรือ Stepper Motor นั้นจะไม่สามารถขับเคลื่อนหรือทำงานเองได้ จำเป็นต้องมีวงจรอิเล็กทรอนิกส์ที่ใช้ในการสร้างสัญญาณหรือจ่ายพัลส์ไปให้วงจรขับ Stepping Motor (Stepping Motor Drive) การสร้างสัญญาณนั้นจำเป็นต้องสร้าง และเรียงลำดับของสัญญาณด้วย และอีกสิ่งที่สำคัญคือ การดูตำแหน่งของสายที่ทำการต่อเข้ากับตัว Stepping Motor

2.2.2 Stepping Motor Driver

Stepping Motor Drive เป็นอุปกรณ์ขับ Stepping Motor ส่งสัญญาณไฟฟ้าไปยัง Stepping Motor ให้สามารถเคลื่อนที่ไปยังตำแหน่งต่างๆ ได้ ซึ่ง Stepping Motor Drive จะทำงานร่วมกับตัว Controller ด้วยเช่นกันดังรูปที่ 2.7 และรูปที่ 2.8

รูปที่ 2.7 Stepping Motor Drive รุ่น TB6600

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แผงผังการทำงานของ Stepping Motor

2.2.3 ลิมิตสวิตช์ Limit Switch

ลิมิตสวิตช์ (Limit switch) เป็นสวิตช์ที่จำกัดระยะทาง การทำงานอาศัยแรงกดภายนอกมากระทำ เช่น วางของทับที่ปุ่มกดหรือลูกเบี้ยวมาชนที่ปุ่มกด และเป็นผลทำให้หน้าสัมผัสที่ต่ออยู่กับก้านชนเปิด-ปิดตามจังหวะของการชนดังรูปที่ 2.9

รูปที่ 2.9 ลิมิตสวิตช์ (Limit Switch)

2.2.4 สายพาน (Belt)

สายพานเป็นอุปกรณ์ที่คล้องโยงเครื่องจักรต่างๆ เพื่อพาให้หมุนไปด้วยกัน เป็นส่วนรองรับวัสดุสิ่งของต่างๆ ทำให้วัสดุขนถ่ายที่อยู่บนสายพานนั้นเคลื่อนที่ตามสายพานไปด้วยดังรูปที่ 2.10

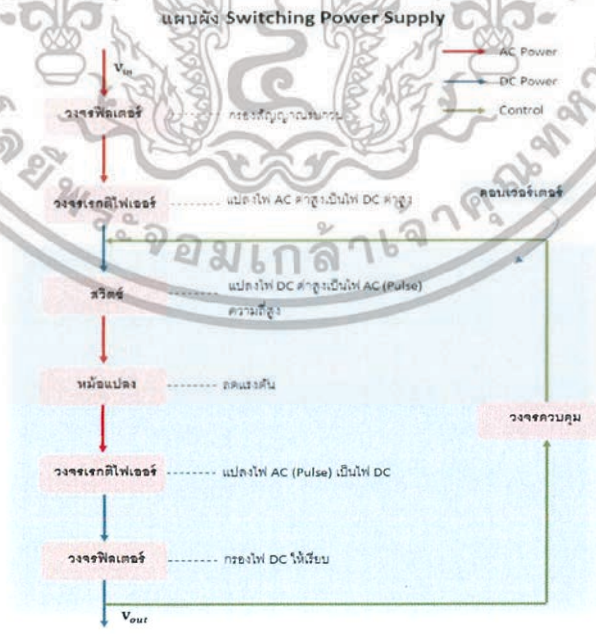
รูปที่ 2.10 สายพาน (Belt)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 Switching Power Supply

Switching Power Supply นั้นถูกสร้างขึ้นมาเพื่อใช้ในงานอิเล็กทรอนิกส์ เป็นแหล่งจ่ายไฟให้กับอุปกรณ์ต่างๆ และสามารถเปลี่ยนแรงดันไฟจากไฟสลับโวลต์สูงให้เป็นแรงดันไฟตรงโวลต์ต่ำได้ Switching Power Supply จะประกอบด้วย 3 ส่วนใหญ่ๆ คือ 1. วงจรฟิลเตอร์และเรกติไฟเออร์ ทำหน้าที่แปลงแรงดันไฟสลับเป็นไฟตรง 2. คอนเวอร์เตอร์ ทำหน้าที่แปลงไฟตรงเป็นไฟสลับความถี่สูง และแปลงกลับเป็นไฟตรงโวลต์ต่ำ 3. วงจรควบคุม ทำหน้าที่ควบคุมการทำงานของคอนเวอร์เตอร์ เพื่อให้ได้แรงดันเอาต์พุตตามต้องการดังรูปที่ 2.11 และรูปที่ 2.12

รูปที่ 2.11 Switching Power Supply



รูปที่ 2.12 แผนผังการทำงานของ Switching Power Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

ในส่วนของบทนี้จะพูดถึงวิธีการดำเนินในการเริ่มต้นจากการศึกษาปัญหาจนถึงขั้นตอนที่ทำอยู่ในปัจจุบันอยู่โดยแบ่งขั้นตอนออกเป็นดังนี้

3.1 ออกแบบระบบพื้นฐานที่เกี่ยวข้องกับโครงงาน

3.2 ออกแบบโครงสร้างจำลองของอาคารจอดรถ

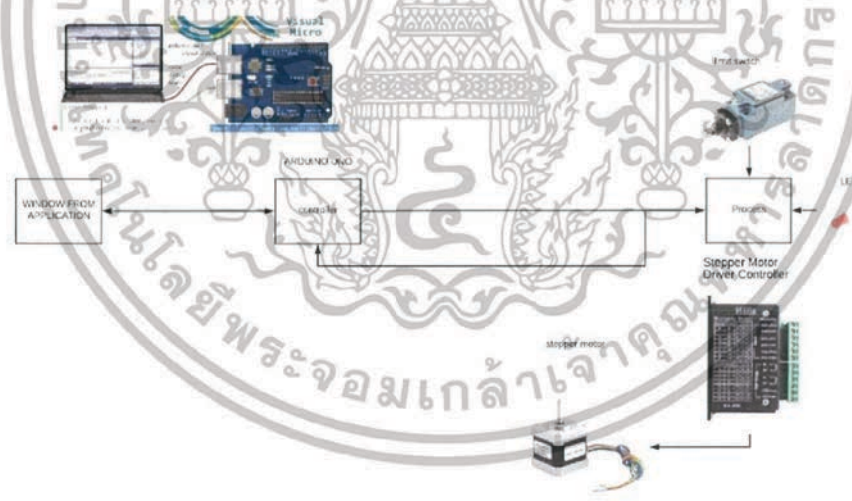
3.3 การเขียนโปรแกรม Parking Control System

3.4 การเขียนโปรแกรม Arduino

3.5 ทำการเชื่อมอุปกรณ์ที่เกี่ยวข้องเข้าด้วยกัน

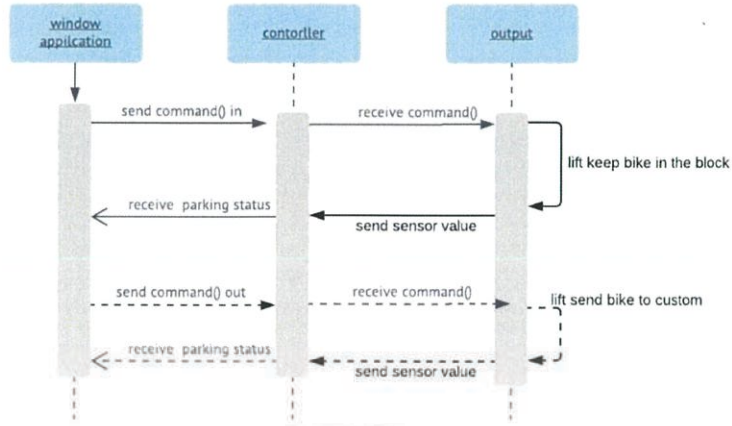
3.1 ออกแบบระบบพื้นฐานที่เกี่ยวข้องกับโครงงาน

ในส่วนนี้จะพูดถึงการออกแบบระบบการทำงานของทั้งระบบเป็นแบบ Block Diagram เพื่อแสดงถึงลำดับการทำงานของระบบ โดยระบบจะแบ่งเป็น 2 ส่วนหลักๆ คือ ส่วนของวินโดแอพพลิเคชันเพื่อป้อนคำสั่งให้ตัวควบคุม และส่วนของตัวควบคุมเพื่อควบคุมอุปกรณ์ไฟฟ้าต่างๆ ตามคำสั่งที่ป้อนเข้าไปดังรูปที่ 3.1 และรูปที่ 3.2



รูปที่ 3.1 Block Diagram แสดงอุปกรณ์และหลักงานทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

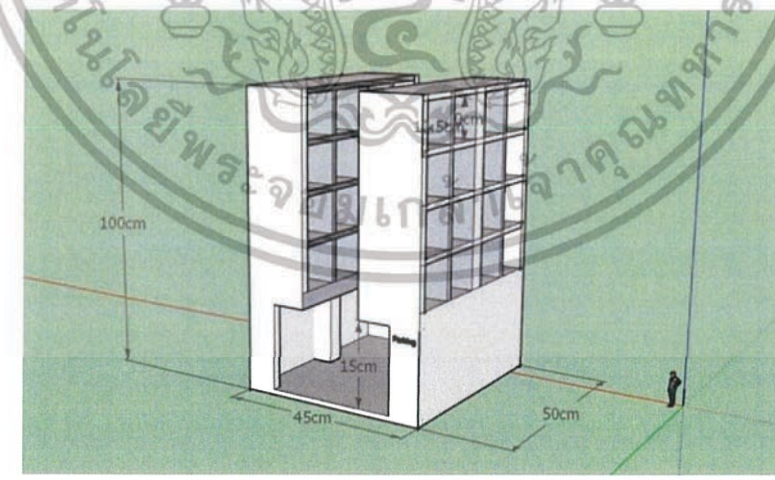


รูปที่ 3.2 Sequence Diagram แสดงลำดับการทำงานของระบบ

3.2 ออกแบบโครงสร้างจำลองของอาคารจอดรถ

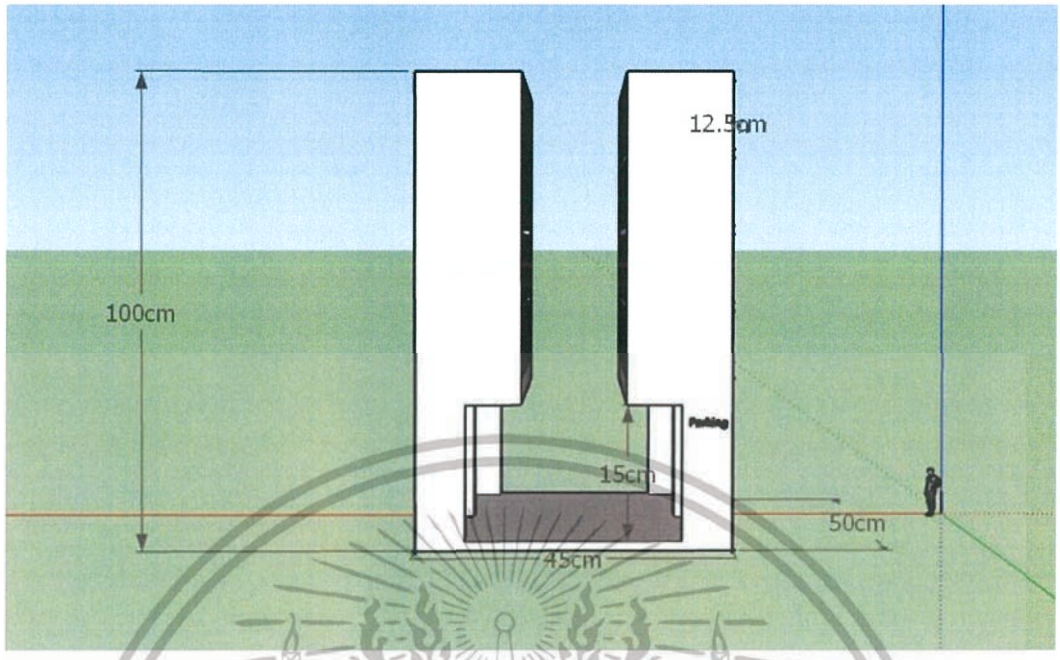
ในส่วนนี้จะพูดถึงการออกแบบโครงสร้างจำลองของอาคารจอดรถ โดยหลังจากออกแบบระบบทั้งหมดของโครงงานแล้ว จำเป็นต้องออกแบบโครงสร้างของอาคารจอดรถเพื่อที่จะทราบขนาดและจำนวนในการจอดรถ โดยการออกแบบได้ใช้โปรแกรม SketUp 2018 ในการออกแบบ

โดยตัวอาคารได้ออกแบบตัวอาคารเพื่อให้ได้ประสิทธิภาพสูงสุด และใช้เนื้อที่ในการสร้างตัวอาคารน้อยที่สุด อาคารถูกออกแบบให้สามารถก่อสร้างได้ง่าย และใช้ค่าใช้จ่ายในการก่อสร้างที่ไม่แพงจนเกินไป โดยขนาดของอาคารมีพื้นที่ชั้นล่างกว้าง 45 เซนติเมตร และสูง 150เซนติเมตร เป็นพื้นที่สำหรับติดตั้งตัวลิฟท์ และเป็นพื้นที่สำหรับนำรถเข้าตัวลิฟท์ พื้นที่ด้านบนเป็นช่องขนาดกว้าง 12.5 เซนติเมตร สูง 20 เซนติเมตร ในแต่ละชั้นประกอบด้วยช่องบรรจุรถสี่ช่องซ้อนกันสี่ชั้นเป็นพื้นที่สำหรับการนำรถเข้าไปจอดดังรูปที่ 3.3 ถึงรูปที่ 3.5

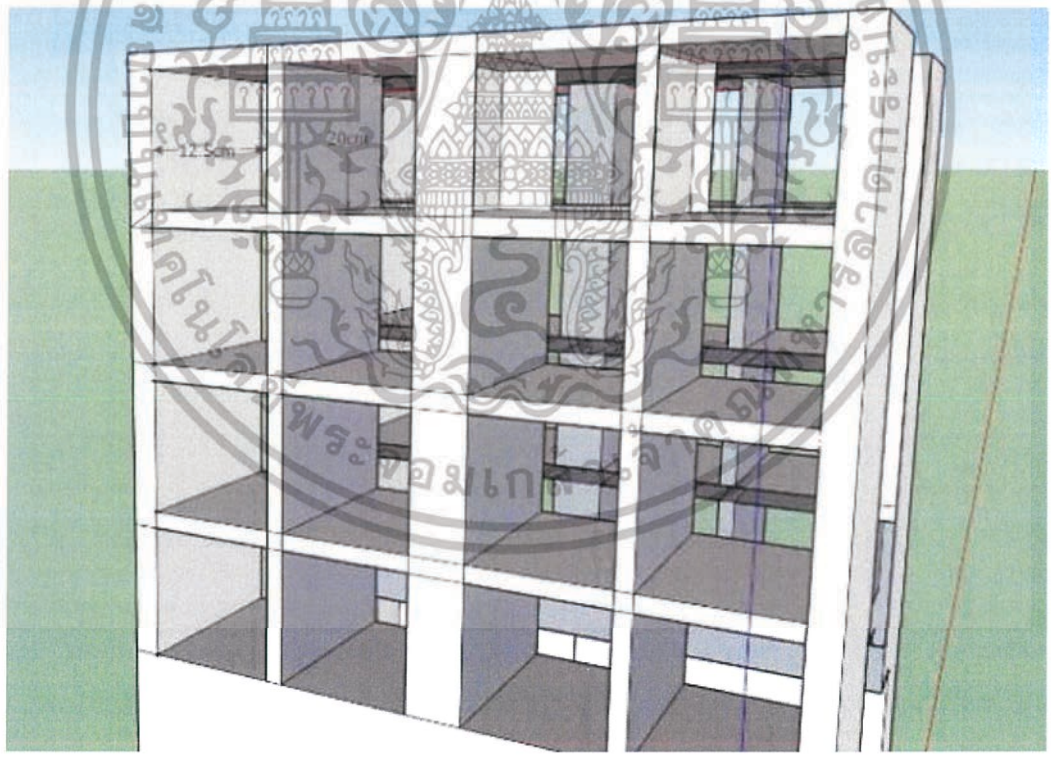


รูปที่ 3.3 แบบของโครงสร้างจำลองมุม Perspective View

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แบบของโครงสร้างจำลองมุม Front View

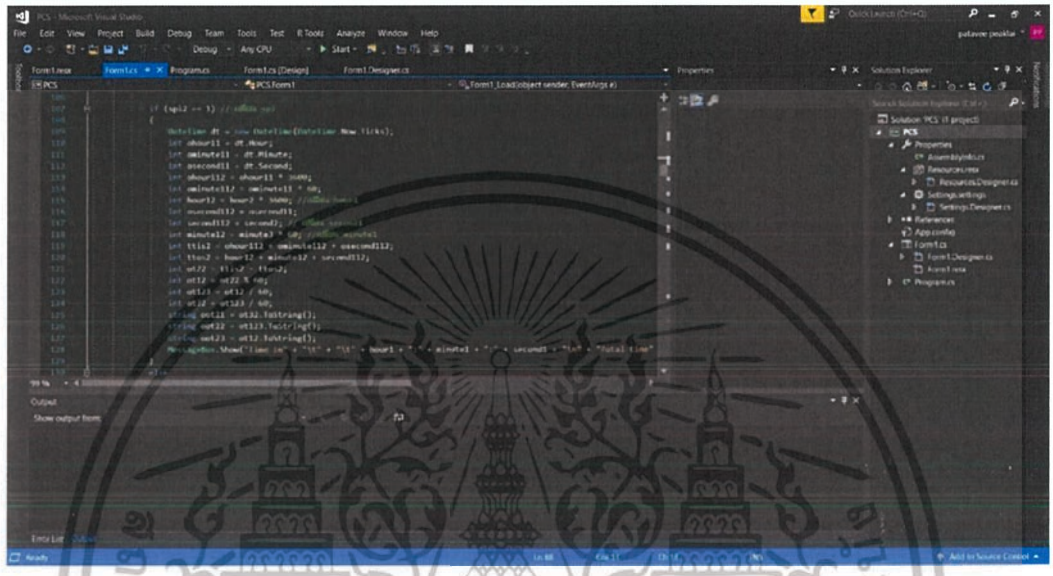


รูปที่ 3.5 รูปแสดงโครงสร้างและระยะห่างของช่องจอดรถ

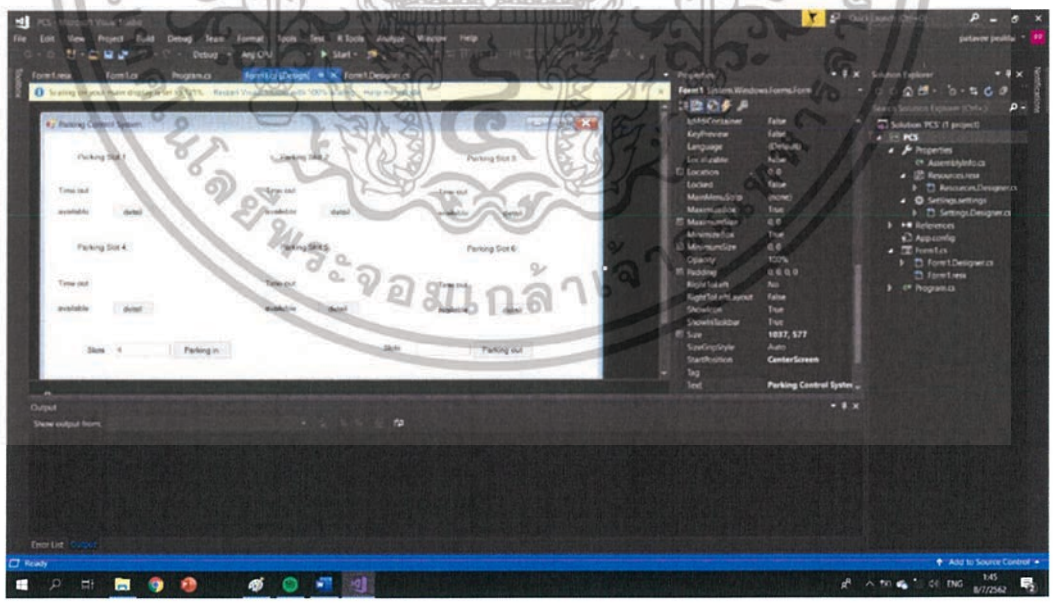
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเขียนโปรแกรม Parking Control System

ในส่วนของการเขียนโปรแกรม Parking Control System นั้นเขียนเพื่อสร้างส่วนต่อประสานระหว่างผู้ใช้งาน และตัวควบคุม และทำหน้าที่ส่งคำสั่งการจอดรถเข้า และนำรถออก บอกรายละเอียดเกี่ยวกับการจอดรถ ได้แก่ สถานะของช่อง เวลาที่จอดรถ เวลาของคันก่อนหน้าที่นำออกจากช่องในส่วนของโค้ดโปรแกรมสามารถดูในภาคผนวกได้ดังรูปที่ 3.6 และรูปที่ 3.7



รูปที่ 3.6 รูปแสดงระหว่างการเขียนโปรแกรม Parking Control System (1)

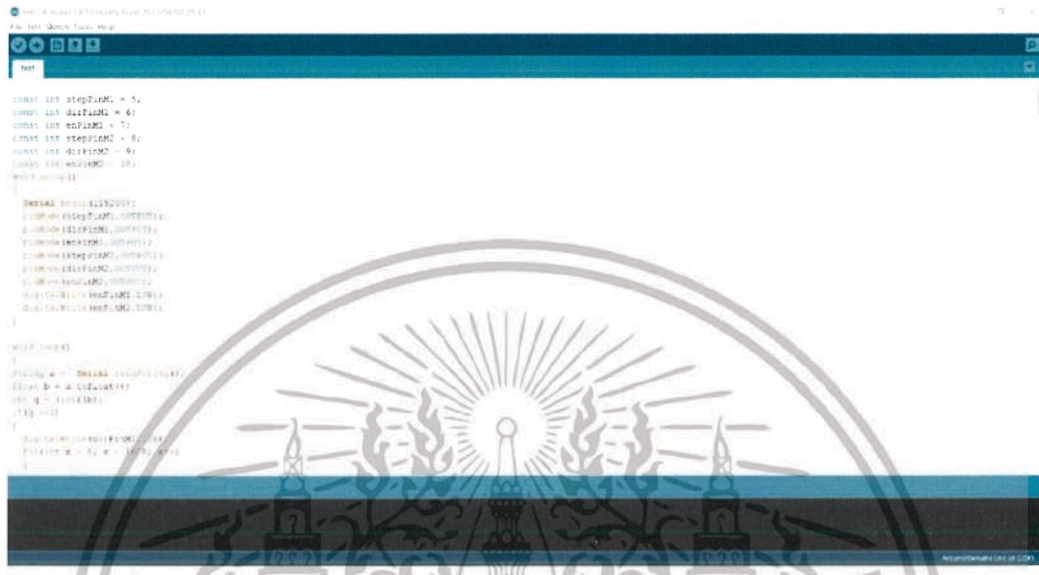


รูปที่ 3.7 รูปแสดงระหว่างการเขียนโปรแกรม Parking Control System (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเขียนโปรแกรม Arduino

ในส่วนของการเขียนโปรแกรม Arduino นั้นเขียนเพื่อรับคำสั่งจากตัววินโดแอพพลิเคชันและป้อนคำสั่งควบคุมให้กับอุปกรณ์ไฟฟ้าที่เกี่ยวข้อง ในส่วนของตัวโค้ดโปรแกรมสามารถดูในภาคผนวกได้ดังรูปที่ 3.8 และรูปที่ 3.9



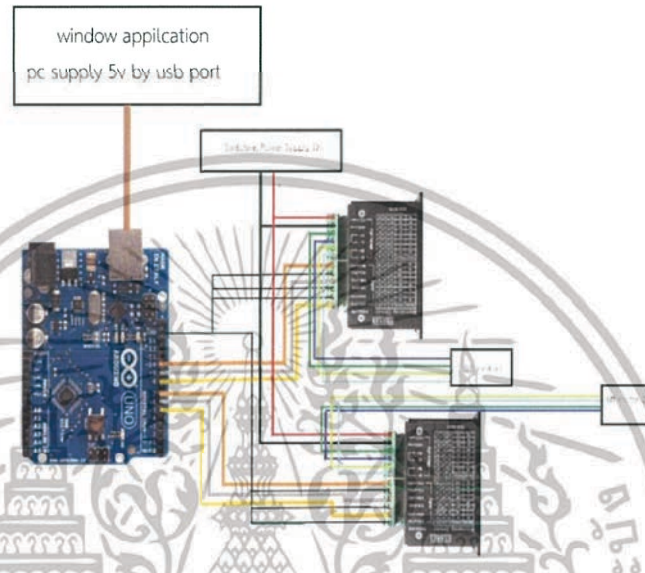
รูปที่ 3.8 รูปแสดงระหว่างการเขียนโปรแกรมควบคุม Arduino (1)

รูปที่ 3.9 รูปแสดงระหว่างการเขียนโปรแกรมควบคุม Arduino (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ทำการเชื่อมอุปกรณ์ที่เกี่ยวข้องเข้าด้วยกัน

ในส่วนนี้เป็นการนำทุกระบบที่ออกแบบมาทำงานร่วมกัน โดยนำส่วนของโปรแกรม Parkin Control System โปรแกรมควบคุม Arduino และอุปกรณ์ไฟฟ้า มาทำการเชื่อมต่อโดยแสดงเป็นแผนผังดังรูปที่ 3.10



รูปที่ 3.10 แผนผังแสดงการเชื่อมต่ออุปกรณ์ไฟฟ้า และโปรแกรมที่เขียนขึ้นมา

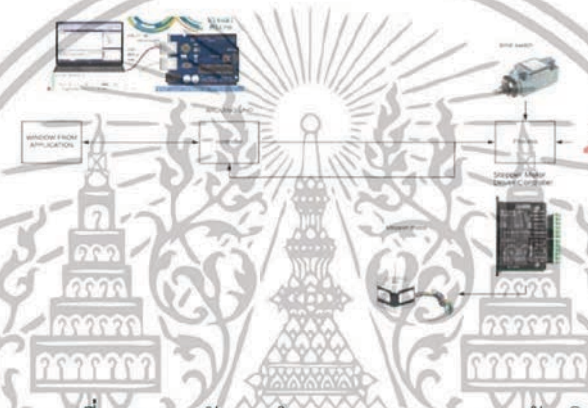
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

ในส่วนของบทนี้จะเป็นการพูดถึงผลการทดลอง และภาพรวมต่างๆ ของระบบในโครงงานนี้โดยแบ่งออกได้ดังนี้

4.1 แผนผังการทำงานของอาคารจอดรถอัจฉริยะ

โดยแผนผังนี้ได้จากการออกแบบระบบพื้นฐานที่ทำในบทที่ 3.3 นำอุปกรณ์ และซอฟต์แวร์ ที่ทำขึ้นมาทำงานรวมกันดังรูปที่ 4.1



รูปที่ 4.1 แผนผังการทำงานของอาคารจอดรถอัจฉริยะ

4.2 อาคารจำลอง

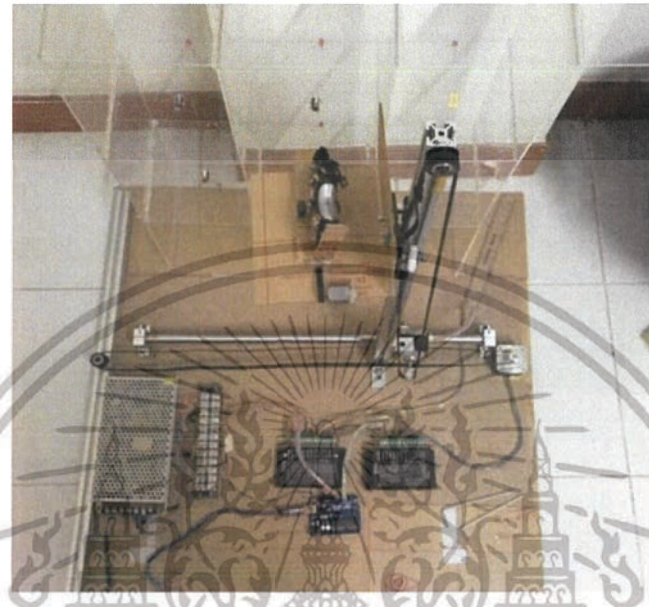
อาคารจำลองที่ถูกประกอบขึ้นจากการออกแบบด้วยโปรแกรม SketUp 2018 ชิ้นส่วนของตัวอาคารวัสดุทำจากอะคริลิกใสประกอบกัน ตัวอาคารมีจำนวน 3 ชั้น ชั้นละ 3 ช่องสามารถจอดรถจำลองได้ 9 คัน ซึ่งแต่ละช่องจะใช้ Limit Switch ในตัวยืนยันการเข้าจอดรถและเชื่อมดังรูปที่ 4.2

รูปที่ 4.2 อาคารจอดรถจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

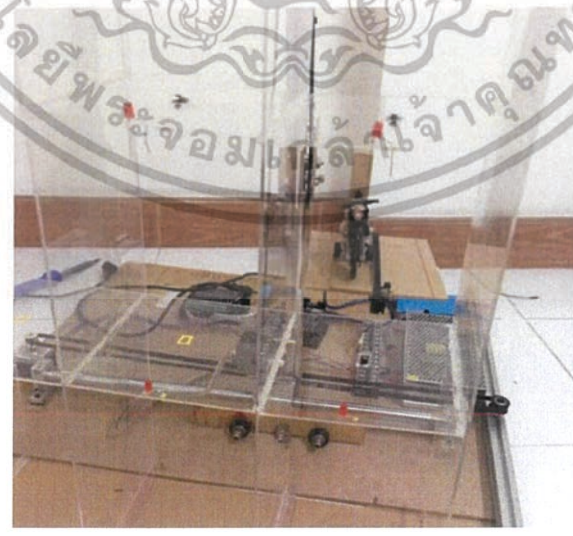
4.3 ตัวลิฟท์

ชิ้นส่วนของตัวลิฟท์จำลองประกอบด้วยแกนลิฟท์ประกอบขึ้นจากอลูมิเนียมโปรไฟล์นำมาประกอบกันในแกนนอน และแกนตั้งเพื่อเป็นแกนสำหรับรองรับตัวลิฟท์ โดยแกนของลิฟท์จะทำให้เคลื่อนที่ไปตามช่องต่างๆ ด้วยมอเตอร์ขับสายพานในทั้งสามแกนดังรูปที่ 4.3



รูปที่ 4.3 รูปจากด้านบนของลิฟท์ยกรถ

ที่ฐานของเสาแกนจะมีล้อที่ประกอบขึ้นจากโดยนำอะคริลิกที่เหลือใช้ เพื่อเพิ่มความแข็งแรงและช่วยรับน้ำหนักของแกนตั้ง และฐานทั้งสองข้างของแกนนอนถูกยึดไว้กับฐานไม้อัดด้วยน็อตยึด



รูปที่ 4.4 รูปจากด้านฐานของลิฟท์ยกรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 Parking Control System วินโดแอพพลิเคชัน

เป็นโปรแกรมที่พัฒนาเพื่อการสั่งการนำรถเข้า-ออกตามช่องสั่งการผ่านหน้าอินเตอร์เฟส ซึ่งสามารถกำหนดช่องที่ต้องการนำรถเข้าจอด และแสดงสถานะของแต่ละช่อง ระยะเวลาในการเข้าจอด และเวลาเริ่มต้นที่รถเข้า เพื่อให้นำข้อมูลส่วนนี้ไปพัฒนาในการคำนวณค่าใช้จ่ายต่อไป และโปรแกรมยังมีระบบความปลอดภัย โดยที่ไม่สามารถสั่งการนำรถเข้าจอดในช่องที่มีสถานะไม่ว่างอยู่ เพื่อเป็นการป้องกันการเกิดอุบัติเหตุจากความผิดพลาดของผู้เข้าจอดเองดังรูปที่ 4.5

รูปที่ 4.5 หน้าตาของโปรแกรม Parking Control System พร้อมรายละเอียด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 ผลการนำรถจำลองเข้าจอด

จากการนำระบบทุกอย่างมาทำงานรวมกันได้ทำการทดสอบนำรถเข้าจอด 9 ช่องทดลองช่องละ 5 ครั้ง ได้ผลการทดลองตามตารางที่ 4.1

ตารางที่ 4.1 ตารางผลการทดลองนำเข้ารถเข้าจอดครั้งที่ 1

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5
ช่องเก็บรถที่ 1	✓	✗	✓	✓	✓
ช่องเก็บรถที่ 2	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 3	✓	✓	✓	✓	✗
ช่องเก็บรถที่ 4	✗	✓	✓	✓	✓
ช่องเก็บรถที่ 5	✓	✗	✓	✗	✓
ช่องเก็บรถที่ 6	✓	✓	✗	✓	✓
ช่องเก็บรถที่ 7	✓	✗	✓	✓	✓
ช่องเก็บรถที่ 8	✓	✓	✓	✗	✓
ช่องเก็บรถที่ 9	✓	✓	✓	✓	✓

จากผลการทดลองครั้งที่ 1 พบว่ารถไม่สามารถเข้าจอดได้ตรงตามช่องจอดได้ทุกครั้ง เนื่องจากสายพานที่ใช้นั้นเกิดการหย่อนยานจากการเคลื่อนที่ของลิฟท์ และหน้าสัมผัสของสายพานและตัวพูเลย์ที่เชื่อมกับมอเตอร์ไม่สัมผัสกันอย่างต่อเนื่อง ทำให้รถไม่สามารถนำรถเข้าจอดได้อย่างถูกต้อง โดยทำการแก้ไขด้วยการลดระยะของสายพาน และเพิ่มความตึงให้กับสายพาน และให้หน้าสัมผัสของพูเลย์ และตัวสายพานสัมผัสกันอย่างเหมาะสม จึงทำการทดองอีกครั้ง โดยการทดลองนำเข้าช่องจอด 9 ช่อง ช่องละ 5 ครั้งเหมือนการทดลองครั้งที่ 1 และผลของการทดลองได้ตามตารางที่ 4.2

ตารางที่ 4.2 ตารางผลการทดลองนำเข้ารถเข้าจอดครั้งที่ 2

	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5
ช่องเก็บรถที่ 1	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 2	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 3	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 4	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 5	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 6	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 7	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 8	✓	✓	✓	✓	✓
ช่องเก็บรถที่ 9	✓	✓	✓	✓	✓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

จากผลการดำเนินงานของอาคารจอดรถจักรยานยนต์อัจฉริยะ สามารถประกอบได้ใกล้เคียงกับแบบที่ออกแบบไว้ในโปรแกรม SketUp อย่างไรก็ตามตัวอาคารพบปัญหาในการสร้าง เนื่องจากโครงสร้างของอาคารจอดรถทำจากอะคริลิกซึ่งพบปัญหาในด้านความแข็งแรง ในการสร้างตัวอาคารจึงมีการตัดส่วนชั้นล่าง ซึ่งมีในโปรแกรม SketUp ออกเพื่อความแข็งแรงของตัวอาคาร

ในส่วนของ Software ได้ใช้โปรแกรม Visual Studio 2017 และใช้ Arduino เป็นตัวควบคุม Stepping Motor โดยตัวลิฟท์จะเคลื่อนที่ไประยะเท่าไรนั้นขึ้นอยู่กับค่าพัลส์ที่กำหนดในตัวโปรแกรม จากการทดลองจ่ายค่าพัลส์นั้นจะทราบระยะที่ตัวลิฟท์เคลื่อนที่ไปพอดีกับแต่ละช่อง จะได้มาซึ่งค่าพัลส์ของมอเตอร์ทั้งสามตัว และได้ระยะที่พอดีกับช่องทั้ง 9 ช่อง หลังจากนั้นก็เขียนหน้าอินเตอร์เฟสเพื่อให้ง่ายต่อการสั่งการ

5.1 สรุปผลการทดลอง

จากการทดลองนำรถเข้าออกนั้นตัวโค้ด และหน้าอินเตอร์เฟสสามารถใช้งานได้ดีแต่ปัญหาที่พบคือ ตัวลิฟท์มีข้อผิดพลาดในการเคลื่อนที่ซึ่งเกิดจากปัญหาส่วนชิ้นงานที่ไม่สามารถดึงสายพานให้ตึงเพียงพอได้ ซึ่งหลังจากการแก้ไขตัวชิ้นงานแล้วนั้นก็สามารถเคลื่อนที่ได้ตามที่กำหนดไว้ และมีการแก้ไข้หน้าอินเตอร์เฟสให้ใช้งานง่ายขึ้น สำหรับผู้ใช้งานส่วนเพิ่มเติมเกี่ยวกับข้อมูล และระยะเวลาในการจอดของรถแต่ละคัน

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

1. วัสดุที่ใช้ทำอาคารจอดรถจำลองขาดความแข็งแรง
2. ไม่มีอุปกรณ์ช่างที่ใช้ในขั้นตอนการปฏิบัติงาน
3. ขณะเครื่องทำงานเกิดปัญหากับสายพาน
4. ความหนักของเสาทำให้ตัวลิฟท์โอนเอน
5. อุปกรณ์ที่วางแผนไว้มีราคาที่แพงเกินไป

5.3 แนวทางการแก้ไข

1. ใช้วัสดุต่างๆ ที่หาได้มาเพิ่มความแข็งแรงให้กับตัวอาคาร เช่น กาวยาง
2. ปรึกษาอาจารย์ที่ปรึกษาปริญญานิพนธ์ เพื่อหาวิธียืมอุปกรณ์ที่จำเป็น
3. ยึดสายพานให้ตึงตลอดเวลา
4. หาวัสดุต่างๆ มายึดตัวเสาให้เคลื่อนที่น้อยที่สุด
5. ใช้อุปกรณ์ในห้องโปรเจคที่เหลือใช้นำมาประยุกต์ใช้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] **"หลักการทำงานของ Stepping motor"** เข้าถึงได้จาก: https://www.factomart.com/th/factomartblog /principle-of-stepping-motor/

[2] **"หลักการทำงานของ Switching power supply"** เข้าถึงได้จาก: https://mall.factomart.com/principle-of-switching-power-supply/

[3] **"เริ่มเขียนโปรแกรมแรกกับ Arduino UNO R3"** เข้าถึงได้จาก:https://thaiarduino.club/first-program-with-arduino-uno-r3/

[4] **"TB6600 5A Stepper motor driver and Arduino"** เข้าถึงได้จาก: https://www.youtube.com/watch?v=SeqlgZLqI-k

[5] **"สอนเขียนโปรแกรมภาษา C# เบื้องต้น"** เข้าถึงได้จาก: https://www.youtube.com/playlist?list=PLEE74DyIkwEm84UiA8fRGvJSecZaHZ9KY

[6] **"การคำนวณค่า stepper ของสายพานสำหรับการเคลื่อนที่ของ Stepping motor driver"** เข้าถึงได้จาก: https://www.youtube.com/watch?v=JC2-f-XsoTg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ด Arduino รับคำสั่งจาก โปรแกรม Parking Control System เพื่อควบคุม Lift motor

```
const int stepPinM1 = 5;
const int dirPinM1 = 6;
const int enPinM1 = 7;
const int stepPinM2 = 8;
const int dirPinM2 = 9;
const int enPinM2 = 10;
void setup()
{
Serial.begin(115200);
pinMode(stepPinM1,OUTPUT);
pinMode(dirPinM1,OUTPUT);
pinMode(enPinM1,OUTPUT);
pinMode(stepPinM2,OUTPUT);
pinMode(dirPinM2,OUTPUT);
pinMode(enPinM2,OUTPUT);
digitalWrite(enPinM1,LOW);
digitalWrite(enPinM2,LOW);
}
void loop()
{
String a =  Serial.readString();
float b = a.toFloat();
int q = (int)(b);
if(q ==1)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 2)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 3)
{
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 1700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 1700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 4)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);}
else if(q == 5)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 6)
{
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
}
if(q ==11)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 1600; x++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 12)
{
digitalWrite(dirPinM1,LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 13)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 1000; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 14)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 1600; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 15)
{
digitalWrite(dirPinM1,LOW);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM1,HIGH);
for(int x = 0; x < 800; x++)
{
digitalWrite(stepPinM1,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM1,LOW);
delayMicroseconds(500);
}
delay(1000);
}
else if(q == 16)
{
digitalWrite(dirPinM2,LOW);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
delayMicroseconds(500);
}
delay(1000);
digitalWrite(dirPinM2,HIGH);
for(int x = 0; x < 700; x++)
{
digitalWrite(stepPinM2,HIGH);
delayMicroseconds(500);
digitalWrite(stepPinM2,LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delayMicroseconds(500);
}
delay(1000);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Visual studio 2017 code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;
namespace PCS
{
    public partial class Form1 : Form
    {
        private SerialPort myport;
        string s, j;
        int spi1 = 0, spi2 = 0, spi3 = 0, spi4 = 0, spi5 = 0, spi6 = 0;
        int hour1;
        int minute1;
        int second1;
        int hour2;
        int minute2;
        int second2;
        int hour3;
        int minute3;
        int second3;
        int hour4;
        int minute4;
        int second4;
        int hour5;
        int minute5;
        int second5;
        int hour6;
        int minute6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int second6;
public Form1()
{
    InitializeComponent();
    intt();

}
private void intt()
{
    myport = new SerialPort();
    myport.BaudRate = 115200;
    myport.PortName = "COM4";


    myport.Open();

}
private void button1_Click(object sender, EventArgs e)
{

    if (spi1 == 1) // เปลี่ยน spi
    {
        DateTime dt = new DateTime(DateTime.Now.Ticks);
        int ohour11 = dt.Hour;
        int ominute11 = dt.Minute;
        int osecond11 = dt.Second;
        int ohour112 = ohour11 * 3600;
        int ominute112 = ominute11 * 60;
        int hour12 = hour1 * 3600; //เปลี่ยน hour1
        int osecond112 = osecond11;
        int second112 = second1; // เปลี่ยน second1
        int minute12 = minute1 * 60; //เปลี่ยน minute1
        int ttis2 = ohour112 + ominute112 + osecond112;
        int ttos2 = hour12 + minute12 + second112;
        int ot22 = ttis2 - ttos2;
        int ot12 = ot22 % 60;
        int ot123 = ot12 / 60;
        int ot32 = ot123 / 60;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            string oot21 = ot32.ToString();
            string oot22 = ot123.ToString();
            string oot23 = ot12.ToString();
            MessageBox.Show("Time in" + "\t" + "\t" + hour1 + ":" + minute1 + ":" +
second1 + "\n" + "Total time" + "\t" + oot21 + ":" + oot22 + ":" + oot23, "Detail");
        }
        else
        {
            MessageBox.Show("00:00:00", "Detail");
        }
    }
        private void Form1_Load(object sender, EventArgs e)
    {
        timer1.Start();
        label29.Text = DateTime.Now.ToLongTimeString();
    }

    private void label14_Click(object sender, EventArgs e)
    {

    }


    private void label16_Click(object sender, EventArgs e)
    {

    }

    private void button2_Click(object sender, EventArgs e)
    {

        if (spi2 == 1) // เปลี่ยน spi
        {
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour11 = dt.Hour;
            int ominute11 = dt.Minute;
            int osecond11 = dt.Second;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int ohour112 = ohour11 * 3600;
            int ominute112 = ominute11 * 60;
            int hour12 = hour2 * 3600; //เปลี่ยน hour1
            int osecond112 = osecond11;
            int second112 = second2; // เปลี่ยน second1
            int minute12 = minute3 * 60; //เปลี่ยน minute1
            int ttis2 = ohour112 + ominute112 + osecond112;
            int ttos2 = hour12 + minute12 + second112;
            int ot22 = ttis2 - ttos2;
            int ot12 = ot22 % 60;
            int ot123 = ot12 / 60;
            int ot32 = ot123 / 60;
            string oot21 = ot32.ToString();
            string oot22 = ot123.ToString();
            string oot23 = ot12.ToString();
            MessageBox.Show("Time in" + "\t" + "\t" + hour1 + ":" + minute1 + ":" +
second1 + "\n" + "Total time" + "\t" + oot21 + ":" + oot22 + ":" + oot23, "Detail");
        }
        else
        {
            MessageBox.Show("00:00:00", "Detail");
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {

        if (spi3 == 1) // เปลี่ยน spi
        {
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour11 = dt.Hour;
            int ominute11 = dt.Minute;
            int osecond11 = dt.Second;
            int ohour112 = ohour11 * 3600;
            int ominute112 = ominute11 * 60;
            int hour12 = hour3 * 3600; //เปลี่ยน hour1
            int osecond112 = osecond11;
            int second112 = second3; // เปลี่ยน second1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int minute12 = minute3 * 60; //เปลี่ยน minute1
            int ttis2 = ohour112 + ominute112 + osecond112;
            int ttos2 = hour12 + minute12 + second112;
            int ot22 = ttis2 - ttos2;
            int ot12 = ot22 % 60;
            int ot123 = ot12 / 60;
            int ot32 = ot123 / 60;
            string oot21 = ot32.ToString();
            string oot22 = ot123.ToString();
            string oot23 = ot12.ToString();
            MessageBox.Show("Time in" + "\t" + "\t" + hour1 + ":" + minute1 + ":" +
second1 + "\n" + "Total time" + "\t" + oot21 + ":" + oot22 + ":" + oot23, "Detail");
        }
        else
        {
            MessageBox.Show("00:00:00", "Detail");
        }
    }

    private void button4_Click(object sender, EventArgs e)
    {

        if (spi4 == 1) // เปลี่ยน spi
        {
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour11 = dt.Hour;
            int ominute11 = dt.Minute;
            int osecond11 = dt.Second;
            int ohour112 = ohour11 * 3600;
            int ominute112 = ominute11 * 60;
            int hour12 = hour4 * 3600; //เปลี่ยน hour1
            int osecond112 = osecond11;
            int second112 = second4; // เปลี่ยน second1
            int minute12 = minute4 * 60; //เปลี่ยน minute1
            int ttis2 = ohour112 + ominute112 + osecond112;
            int ttos2 = hour12 + minute12 + second112;
            int ot22 = ttis2 - ttos2;
            int ot12 = ot22 % 60;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int ot123 = ot12 / 60;
            int ot32 = ot123 / 60;
            string oot21 = ot32.ToString();
            string oot22 = ot123.ToString();
            string oot23 = ot12.ToString();
            MessageBox.Show("Time in" + "\t" + "\t" + hour1 + ":" + minute1 + ":" +
second1 + "\n" + "Total time" + "\t" + oot21 + ":" + oot22 + ":" + oot23, "Detail");
        }
        else
        {
            MessageBox.Show("00:00:00", "Detail");
        }
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        label29.Text = DateTime.Now.ToLongTimeString();
    }

    private void button5_Click(object sender, EventArgs e)
    {

        if (spi5 == 1) // เปลี่ยน spi
        {
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour11 = dt.Hour;
            int ominute11 = dt.Minute;
            int osecond11 = dt.Second;
            int ohour112 = ohour11 * 3600;
            int ominute112 = ominute11 * 60;
            int hour12 = hour5 * 3600; //เปลี่ยน hour1
            int osecond112 = osecond11;
            int second112 = second5; // เปลี่ยน second1
            int minute12 = minute5 * 60; //เปลี่ยน minute1
            int ttis2 = ohour112 + ominute112 + osecond112;
            int ttos2 = hour12 + minute12 + second112;
            int ot22 = ttis2 - ttos2;
            int ot12 = ot22 % 60;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int ot123 = ot12 / 60;
            int ot32 = ot123 / 60;
            string oot21 = ot32.ToString();
            string oot22 = ot123.ToString();
            string oot23 = ot12.ToString();
            MessageBox.Show("Time in" + "\t" + "\t" + hour1 + ":" + minute1 + ":" +
second1 + "\n" + "Total time" + "\t" + oot21 + ":" + oot22 + ":" + oot23, "Detail");
        }
        else
        {
            MessageBox.Show("00:00:00", "Detail");
        }
    }

    private void label11_Click(object sender, EventArgs e)
    {

    }


    private void button6_Click(object sender, EventArgs e)
    {
        if (spi6 == 1) // เปลี่ยน spi
        {
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour11 = dt.Hour;
            int ominute11 = dt.Minute;
            int osecond11 = dt.Second;
            int ohour112 = ohour11 * 3600;
            int ominute112 = ominute11 * 60;
            int hour12 = hour6 * 3600; //เปลี่ยน hour1
            int osecond112 = osecond11;
            int second112 = second6; // เปลี่ยน second1
            int minute12 = minute6 * 60; //เปลี่ยน minute1
            int ttis2 = ohour112 + ominute112 + osecond112;
            int ttos2 = hour12 + minute12 + second112;
            int ot22 = ttis2 - ttos2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            int ot12 = ot22 % 60;
            int ot123 = ot12 / 60;
            int ot32 = ot123 / 60;
            string oot21 = ot32.ToString();
            string oot22 = ot123.ToString();
            string oot23 = ot12.ToString();
            MessageBox.Show("Time in" + "\t" + "\t" + hour6 + ":" + minute6 + ":" +
second6 + "\n" + "Total time" + "\t" + oot21 + ":" + oot22 + ":" + oot23, "Detail");
        }
        else
        {
            MessageBox.Show("00:00:00", "Detail");
        }
    }

    private void label19_Click(object sender, EventArgs e)
    {

    }

    private void button7_Click(object sender, EventArgs e)
    {
        string f;
        f = textBox1.Text;
        int a = Int32.Parse(f);
        if (spi1 == 0 && a == 1)
        {
            myport.WriteLine(f);
            int x = Int32.Parse(f);
            if (x == 1)
            {
                label8.Text = "unavailable";
                DateTime dt = new DateTime(DateTime.Now.Ticks);
                hour1 = dt.Hour;
                minute1 = dt.Minute;
                second1 = dt.Second;
                textBox1.Text = "";
                MessageBox.Show("Parking in complete");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    spi1 = 1;
}
if (spi2 == 0 && a == 2)
{
    myport.WriteLine(f);
    int x = Int32.Parse(f);
    if (x == 2)
    {
        label9.Text = "unavailable";
        DateTime dt = new DateTime(DateTime.Now.Ticks);
        hour2 = dt.Hour;
        minute2 = dt.Minute;
        second2 = dt.Second;
        textBox1.Text = "";
        MessageBox.Show("Parking in complete");
    }
    spi2 = 1;
}
if (spi3 == 0 && a == 3)
{
    myport.WriteLine(f);
    int x = Int32.Parse(f);
    if (x == 3)
    {
        label10.Text = "unavailable";
        DateTime dt = new DateTime(DateTime.Now.Ticks);
        hour3 = dt.Hour;
        minute3 = dt.Minute;
        second3 = dt.Second;
        textBox1.Text = "";
        MessageBox.Show("Parking in complete");
    }
    spi3 = 1;
}
if (spi4 == 0 && a == 4)
{
    myport.WriteLine(f);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int x = Int32.Parse(f);
        if (x == 4)
        {
            label11.Text = "unavailable";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            hour4 = dt.Hour;
            minute4 = dt.Minute;
            second4 = dt.Second;
            textBox1.Text = "";
            MessageBox.Show("Parking in complete");
        }
        spi4 = 1;
    }
    if (spi5 == 0 && a == 5)
    {
        myport.WriteLine(f);
        int x = Int32.Parse(f);
        if (x == 5)
        {
            label12.Text = "unavailable";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            hour5 = dt.Hour;
            minute5 = dt.Minute;
            second5 = dt.Second;
            textBox1.Text = "";
            MessageBox.Show("Parking in complete");
        }
        spi5 = 1;
    }
    if (spi6 == 0 && a == 6)
    {
        myport.WriteLine(f);
        int x = Int32.Parse(f);
        if (x == 6)
        {
            label13.Text = "unavailable";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            hour6 = dt.Hour;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            minute6 = dt.Minute;
            second6 = dt.Second;
            textBox1.Text = "";
            MessageBox.Show("Parking in complete");
        }
        spi6 = 1;
    }
}
private void button8_Click(object sender, EventArgs e)
{
    string f;
    f = textBox2.Text;
    int a = Int32.Parse(f);
    a += 10;
    if (spi1 == 1 && a == 11)
    {
        string z = a.ToString();
        myport.WriteLine(z);
        int x = Int32.Parse(f);
        if (x == 1)
        {
            label8.Text = "available";
            MessageBox.Show("Parking out complete");
            textBox2.Text = "";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour1 = dt.Hour;
            int ominute1 = dt.Minute;
            int osecond1 = dt.Second;
            label17.Text = ohour1 + ":" + ominute1 + ":" + osecond1;
        }
        spi1 = 0;
    }
    if (spi2 == 1 && a == 12)
    {
        string z = a.ToString();
        myport.WriteLine(z);
        int x = Int32.Parse(f);
        if (x == 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        label9.Text = "available";
        MessageBox.Show("Parking out complete");
        textBox2.Text = "";
        DateTime dt = new DateTime(DateTime.Now.Ticks);
        int ohour2 = dt.Hour;
        int ominute2 = dt.Minute;
        int osecond2 = dt.Second;
        label24.Text = ohour2 + ":" + ominute2 + ":" + osecond2;
    }
    spi2 = 0;
}
if (spi3 == 1 && a == 13)
{
    myport.WriteLine(f);
    int x = Int32.Parse(f);
    if (x == 3)
    {
        label10.Text = "available";
        MessageBox.Show("Parking out complete");
        textBox2.Text = "";
        DateTime dt = new DateTime(DateTime.Now.Ticks);
        int ohour3 = dt.Hour;
        int ominute3 = dt.Minute;
        int osecond3 = dt.Second;
        label25.Text = ohour3 + ":" + ominute3 + ":" + osecond3;

    }
    spi3 = 0;
}
if (spi4 == 1 && a == 14)
{
    myport.WriteLine(f);
    int x = Int32.Parse(f);
    if (x == 4)
    {
        label11.Text = "available";
        MessageBox.Show("Parking out complete");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            textBox2.Text = "";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour4 = dt.Hour;
            int ominute4 = dt.Minute;
            int osecond4 = dt.Second;
            label26.Text = ohour4 + ":" + ominute4 + ":" + osecond4;

        }
        spi4 = 0;
    }
    if (spi5 == 1 && a == 15)
    {
        myport.WriteLine(f);
        int x = Int32.Parse(f);
        if (x == 5)
        {
            label12.Text = "available";
            MessageBox.Show("Parking out complete");
            textBox2.Text = "";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
            int ohour5 = dt.Hour;
            int ominute5 = dt.Minute;
            int osecond5 = dt.Second;
            label27.Text = ohour5 + ":" + ominute5 + ":" + osecond5;

        }
        spi5 = 0;
    }
    if (spi6 == 1 && a == 16)
    {
        myport.WriteLine(f);
        int x = Int32.Parse(f);
        if (x == 6)
        {
            label13.Text = "available";
            MessageBox.Show("Parking out complete");
            textBox2.Text = "";
            DateTime dt = new DateTime(DateTime.Now.Ticks);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

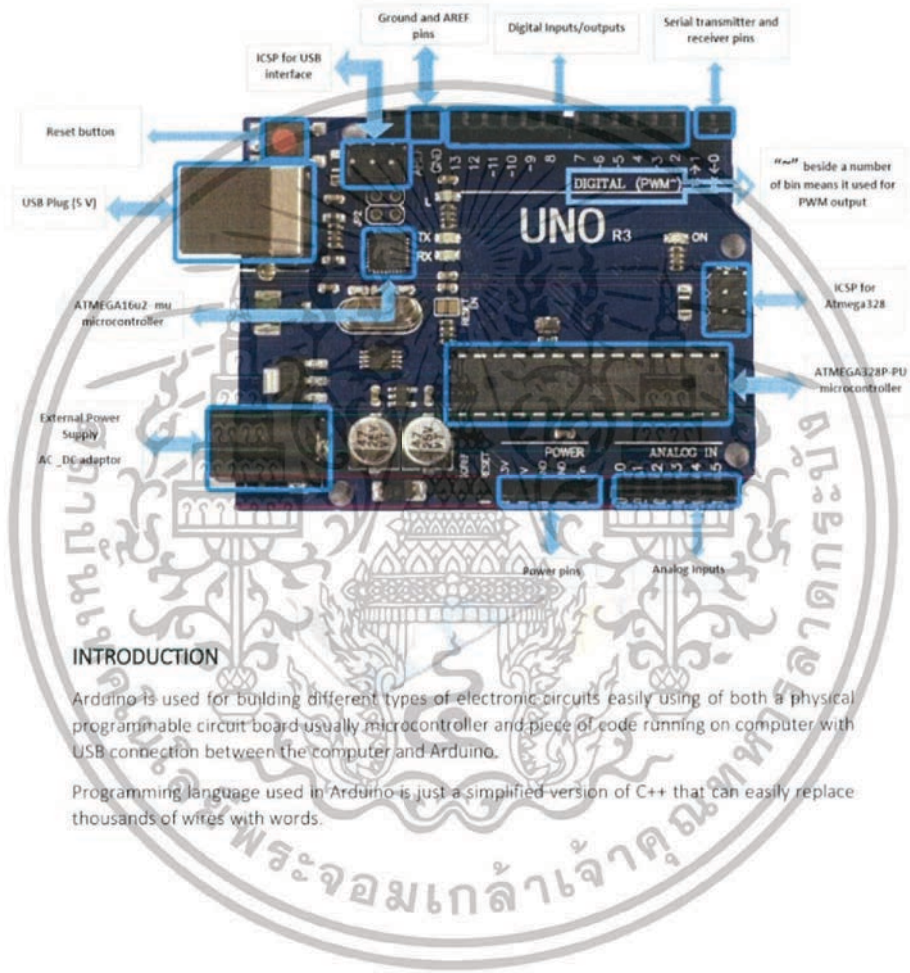
```
                int ohour6 = dt.Hour;
                int ominute6 = dt.Minute;
                int osecond6 = dt.Second;
                label28.Text = ohour6 + ":" + ominute6 + ":" + osecond6;

            }
            spi6 = 0;
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Arduino Uno R3

Arduino Uno R3



รูปที่ ค.1 คุณสมบัติของ Arduino Uno R3 (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARDUINO UNO-R3 PHYSICAL COMPONENTS

ATMEGA328P-PU microcontroller

The most important element in Arduino Uno R3 is ATMEGA328P-PU is an 8-bit Microcontroller with flash memory reach to 32k bytes. It's features as follow:

- **High Performance, Low Power AVR**
- **Advanced RISC Architecture**
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- **High Endurance Non-volatile Memory Segments**
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1K Bytes EEPROM
  - 512/1K/1K/2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- **Peripheral Features**
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
  - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
  - Temperature Measurement
  - Programmable Serial USART

รูปที่ ค.2 คุณสมบัติของ Arduino Uno R3 (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I2 C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

• Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

- 23 Programmable I/O Lines
- 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF

• Operating Voltage:

- 1.8 - 5.5V

• Temperature Range:

- -40°C to 85°C

• Speed Grade:

- 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V

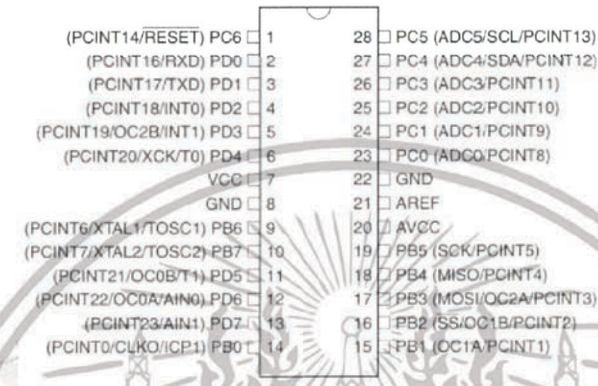
• Power Consumption at 1 MHz, 1.8V, 25°C

- Active Mode: 0.2 mA
- Power-down Mode: 0.1 µA
- Power-save Mode: 0.75 µA (Including 32 kHz RTC)

รูปที่ ค.3 คุณสมบัติของ Arduino Uno R3 (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Pin configuration



ATMEGA16u2- mu microcontroller

Is a 8-bit microcontroller used as USB driver in Arduino Uno R3 it's features as follow:

- High Performance, Low Power AVR

• Advanced RISC Architecture

- 125 Powerful Instructions – Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz

• Non-volatile Program and Data Memories

- 8K/16K/32K Bytes of In-System Self-Programmable Flash
- 512/512/1024 EEPROM
- 512/512/1024 Internal SRAM
- Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
- Data retention: 20 years at 85°C/ 100 years at 25°C

รูปที่ ค.4 คุณสมบัติของ Arduino Uno R3 (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by on-chip Boot Program hardware-activated after reset
- Programming Lock for Software Security

• USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion

- Complies fully with Universal Serial Bus Specification REV 2.0
- 48 MHz PLL for Full-speed Bus Operation: data transfer rates at 12 Mbit/s
- Fully independent 176 bytes USB DPRAM for endpoint memory allocation
- Endpoint 0 for Control Transfers: from 8 up to 64-bytes
- 4 Programmable Endpoints:
  - IN or Out Directions
  - Bulk, Interrupt and Isochronous Transfers
  - Programmable maximum packet size from 8 to 64 bytes
  - Programmable single or double buffer
- Suspend/Resume Interrupts
- Microcontroller reset on USB Bus Reset without detach
- USB Bus Disconnection on Microcontroller Request

• Peripheral Features

- One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
- One 16 bit Timer/Counter with Separate Prescaler, Compare and Capture Mode(three 8-bit PWM channels)
- USART with SPI master only mode and hardware flow control (RTS/CTS)
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

• On Chip Debug Interface (debug WIRE)

• Special Microcontroller Features

- Power-On Reset and Programmable Brown-out Detection
- Internal Calibrated Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

- 22 Programmable I/O Lines
- QFN32 (5x5mm) / TQFP32 packages

รูปที่ ค.5 คุณสมบัติของ Arduino Uno R3 (5)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Operating Voltages**
  - 2.7 - 5.5V
- **Operating temperature**
  - Industrial (-40°C to +85°C)
- **Maximum Frequency**
  - 8 MHz at 2.7V - Industrial range
  - 16 MHz at 4.5V - Industrial range
- **Pin configuration**

รูปที่ ค.6 คุณสมบัติของ Arduino Uno R3 (6)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OTHER ARDUINO UNO R3 PARTS

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e.1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

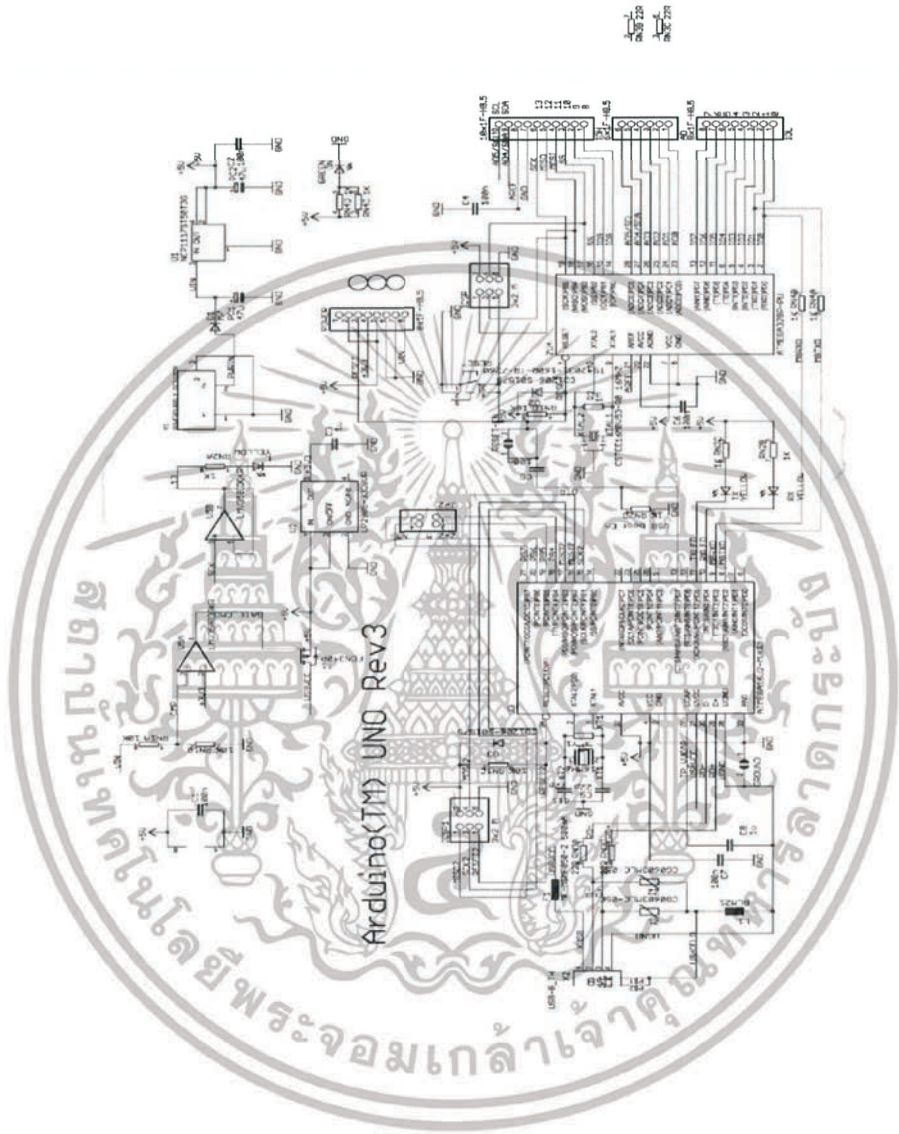
There are a couple of other pins on the board:

- AREF: Reference voltage for the analog inputs. Used with analogReference().
- Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

รูปที่ ค.7 คุณสมบัติของ Arduino Uno R3 (7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ARDUINO UNO R3 SCHEMATIC DIAGRAM



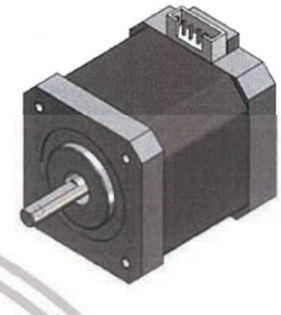
รูปที่ ค.8 คุณสมบัติของ Arduino Uno R3 (8)

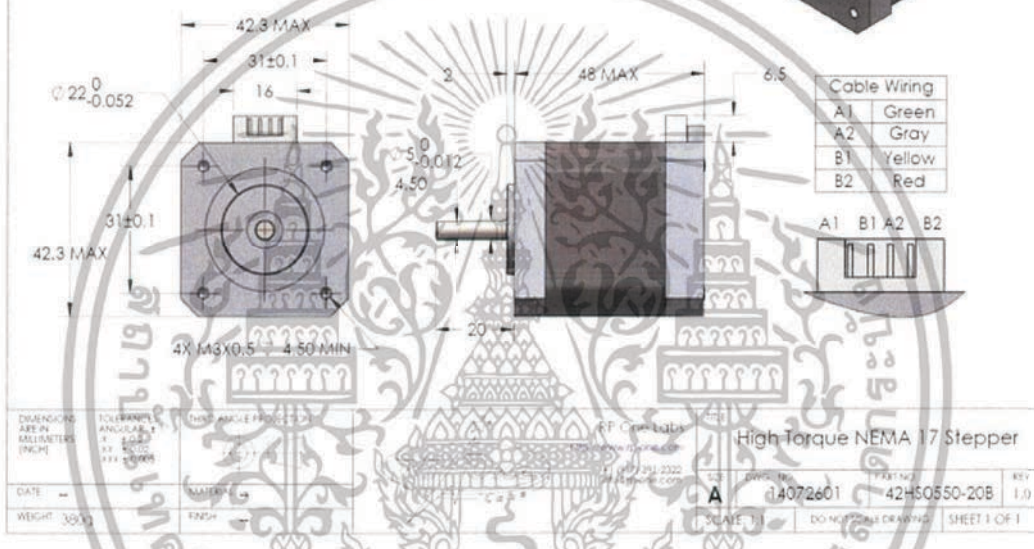
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

Stepping motor NEMA 17

General Specs		Electrical Specs	
Step Angle	1.8°	Rated Voltage	5.0V
Number of Phase	2	Rated Current	1.5A
Insulation Resistance	100MΩ min (500V DC)	Resistance per Phase	3.3 Ω±10%
Insulation Class	Class B	Inductance per Phase	5.5mH±20%
Rotor Inertia	72g· cm	Holding Torque	550mN*m
Mass	0.38kg	Detent Torque	20mN*m



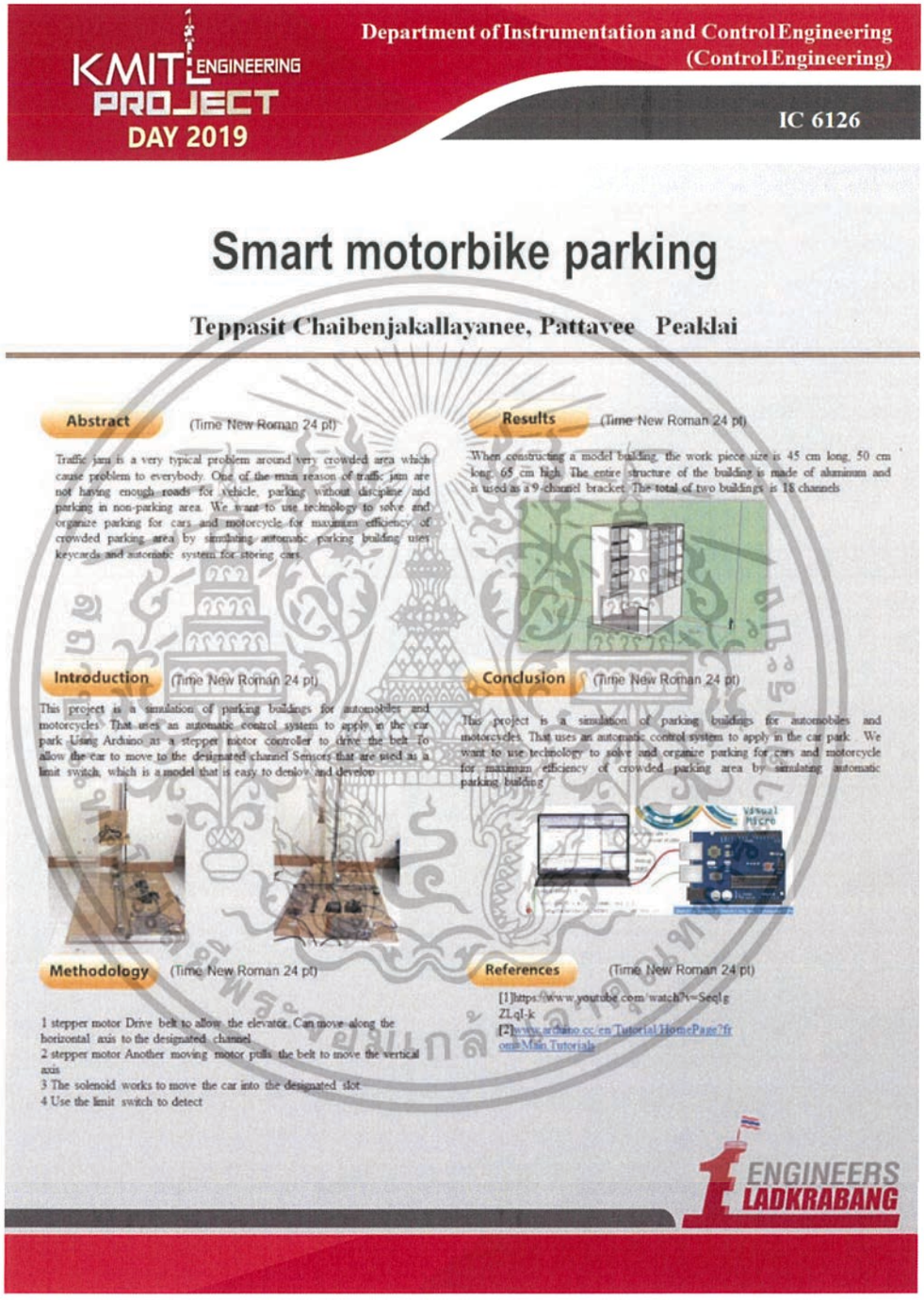


รูปที่ ง.9 คุณสมบัติของ Stepping motor NEMA 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

Poster



รูปที่ จ.1 โปสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้