การพัฒนาระบบลานจอดรถสั่งการด้วยเสียง

Development of Smart Car Parking System using Voice Command

บุญเกิด ธีรศรัณยานนท์

วินัย บุญทา

อินทัช เวนานนท์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะครุศาสตร์อุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

พ.ศ.2561

การพัฒนาระบบลานจอดรถสั่งการด้วยเสียง

บุญเกิด ธีรศรัณยานนท์

วินัย บุญทา

อินทัช เวนานนท์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะครุศาสตร์อุตสาหกรรม

มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

พ.ศ.2561

Development of Smart Car Parking System using Voice Command

Boonkerd Teerasarunyanon
Winai Boonta
Intouch Wenanont

This Project Report Submitted in Partial Fulfillment of
The Requirement for the Degree of Science in Technical Education
Program in Electrical Engineering
Department of Computer Engineering
Faculty of Industrial Education
Rajamangala University of Technology Phra Nakhon
2018

ใบรับรองโครงงาน

ชื่อโครงการ	การพัฒนาระบบลานจอดรถสั่งการด้วยเสียง		
ชื่อนักศึกษา	นายบุญเกิด	ธีรศรัณยานนท์	รหัส 035850504008-1
	นายวินัย	บุญทา	รหัส 035850504030-5
	นายอินทัช	เวนานนท์	รหัส 035850504004-0
แขนงวิชา	วิศวกรรมคอมพิวเตอร์		
อาจารย์ที่ปรึกษา	อาจารย์ศรัณย์	ฉัตรธัญญกิจ	

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะครุศาสตร์อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร อนุมัติให้โครงงานนี้ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรคณะครุศาสตร์อุตสาหกรรมบัณฑิต

.....

(ดร.มนตรี บุญเรืองเศษ)

หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงการ

..... ประธานกรรมการ

(อาจารย์ศรัณย์ ฉัตรธัญญกิจ)

..... กรรมการ

(ดร.ขนิษฐา ดีสุบิน)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์.ดร.ว่าที่ร้อยตรี พรชัย เตชะธนเศรษฐ์)

ชื่อโครงการ	การพัฒนาระบบลานจอดรถสั่งการด้วยเสียง	
ผู้รับผิดชอบโครงการ	นายบุญเกิด	ธีรศรัณยานนท์
	นายวินัย	บุญทา
	นายอินทัช	เวนานนท์
แขนงวิชา	วิศวกรรมคอมพิวเตอร์	
อาจารย์ที่ปรึกษา	อาจารย์ศรัณย์	ฉัตรธัญญกิจ

บทคัดย่อ

รายงานฉบับนี้ได้นำเสนอแบบจำลองระบบลานจอดรถอัจฉริยะที่มีความสามารถในการค้นหาตำแหน่งที่ว่างในอาคารจอดรถผ่านโทรศัพท์มือถือโดยใช้เสียง โดยมีวัตถุประสงค์เพื่อความสะดวกและรวดเร็วในการค้นหาที่จอดรถและเก็บข้อมูลจำนวนผู้ใช้บริการลานจอดรถในแต่ละวัน ระบบที่พัฒนาขึ้นใช้ไมโครคอนโทรเลอร์ NodeMCU ESP8266 เพื่อตรวจจับสถานะช่องว่างและนับจำนวนผู้ใช้บริการของลานจอดรถโดยใช้เซนเซอร์ Infrared และเซนเซอร์ Ultrasonic และส่งสถานะและจำนวนผู้ใช้งานไปยังฐานข้อมูล Firebase ส่วนระบบการสั่งงานด้วยเสียงพัฒนาโดยใช้ Dialogflow สร้าง Chatbot และเชื่อมต่อกับ Google Assistant ซึ่งผู้ใช้งานสามารถโต้ตอบโดยใช้เสียงในการค้นหาที่จอดรถได้โดยตรง จากการทดลองพบว่าระบบลานจอดรถที่ได้พัฒนาขึ้นสามารถแสดงผลได้อย่างถูกต้อง โดยทดลองเพิ่มรถเข้าไปในลานจอดรถ ระบบสามารถแสดงสถานะที่เปลี่ยนแปลงได้ถูกต้อง และเมื่อสอบถามหาที่จอดรถ ระบบสามารถค้นหาที่จอดรถที่ใกล้ที่สุดได้ ในอนาคตมีการวางแผนเพื่อพัฒนาอุปกรณ์เพิ่มเติมและนำไปใช้ในสถานที่จริง

(จำนวนทั้งสิ้น 53 หน้า)

Project Name	Development of Smart Car Parking System using Voice Command	
Responsible For Project	Mr. Boonkerd	Teerasarunyanon
	Mr. Winai	Boonta

	Mr. Intouch	Wenanont
Major	Computer Engineering	
Advisors Program	Mr. Sarun	Chattunyakit

Abstract

This project proposes a simulation model of intelligent parking system that has the ability to search for vacancies in the building or parking through mobile phones using voice command. The purpose of this project is to be convenient and quick in searching for parking spaces and to count the number of car entering park area each day. The developed system uses the NodeMCU ESP8266 microcontroller to detect the vacancy status and count the number of users of the parking lot using infrared and ultrasonic sensors. The collected data will be sent to Firebase database. For voice command system, it is developed by using Dialogflow to create ChatBot and connect with the Google Assistant. Users can interact using voice to search the vacancy in parking area, directly. From the experiment, it is found that the developed parking lot system can display the vacancies correctly. When increasing the number of cars into the parking lot, the system can display the status that has changed correctly. Moreover, when inquiring for parking, the system can find the nearest vacancy for user. In the future, there are plans to develop additional equipment and implement the proposed system in real locations.

(**Total of 53 pages**)

กิตติกรรมประกาศ

ผู้จัดทำโครงงานขอขอบพระคุณอาจารย์ที่ปรึกษา อาจารย์ศรัณย์ ฉัตรธัญกิจ เป็นอย่างสูงที่กรุณาให้คำปรึกษาแนะนำแนวทางต่างๆ เพื่อให้ผู้จัดทำโครงการสามารถดำเนินการจัดทำโครงการครั้งนี้ได้อย่างถูกต้องและ ให้แง่คิดที่มีประโยชน์อย่างยิ่งต่อผู้จัดทำโครงการ

โครงงานฉบับนี้นี้ได้รับทุนอุดหนุนงบประมาณจากโครงการส่งเสริมสิ่งประดิษฐ์และนวัตกรรมเพื่อคนรุ่นใหม่ประจำปีงบประมาณ พ.ศ.2562 มหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร

สุดท้ายนี้คุณค่าหรือประโยชน์ใดๆ อันพึงมีของรายงานฉบับนี้ ผู้จัดทำโครงการขอขอบพระคุณ บิดา มารดา ครู อาจารย์ ตลอดจนผู้มีพระคุณทุกท่านซึ่งเป็นผู้ประสิทธิประสาทวิชาความรู้และให้กำลังใจมาโดยตลอด

บุญเกิด ธีรศรัณยานนท์
วินัย บุญทา
อินทัช เวนานนท์

สารบัญ

	หน้า
บทคัดย่อ	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญ(ต่อ)	จ
สารบัญ(ต่อ)	ฉ
สารบัญตาราง	ช
สารบัญภาพประกอบ	ซ
สารบัญภาพประกอบ(ต่อ)	ฌ
บทที่ 1 บทนำ	1
1.1 ความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 วิธีการดำเนินงาน	2
1.5 แผนการดำเนินงาน	2
1.6 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 เอกสารและทฤษฎีที่เกี่ยวข้อง	4
2.1 ซอฟต์แวร์ที่เกี่ยวข้อง	4
2.1.1 Firebase	4
2.2.2 Dialogflow	5
2.2.3 IOT	5
2.2.4 Chatbot	6
2.2.5 Google Assistant	6
2.2 ฮาร์ดแวร์ที่เกี่ยวข้อง	7
2.2.1 NodeMCU ESP8266	7
2.2.2 Node.js	7
2.2.3 Arduino	8

สารบัญ(ต่อ)

หน้า

บทที่ 2 เอกสารและทฤษฎีที่เกี่ยวข้อง(ต่อ)

2.2.4 infrared Sensor 8

2.2.5 Ultrasonic Sensor 9

2.3 งานวิจัยที่เกี่ยวข้อง 9

2.3.1 การออกแบบและพัฒนาแบบจำลองลานจอดรถอัจฉริยะ 9

2.3.2 การออกแบบและสร้างแบบจำลองลานจอดรถอัจฉริยะ 10

2.3.3 ชุดจำลองลานจอดรถแบบอัจฉริยะ 10

บทที่ 3 ขั้นตอนและวิธีดำเนินการ 11

3.1 แผนการดำเนินโครงการ 11

3.2 ขั้นตอนการทำโครงการ 13

3.2.1 ขั้นตอนการทำโครงการ 13

3.2.2 ออกแบบและพัฒนาระบบ 13

3.2.3 ทดสอบการใช้งาน 13

3.2.4 จัดทำเอกสารโครงการ 13

3.3 โครงสร้างของระบบการทำงาน 13

3.3.1 ภาพรวมของระบบ 13

3.3.2 การออกแบบระบบ Chatbot 14

3.3.3 การออกแบบระบบโมเดลลานจอดรถ 16

3.3.4 การออกแบบระบบนับจำนวนรถที่ใช้งานในแต่ละวัน 23

บทที่ 4 ผลการดำเนินงาน 24

4.1 วิธีการทดสอบ 24

4.1.1 เครื่องมือใช้ในการทดสอบ 24

4.1.2 (ก) การทดสอบการตรวจจับของเซนเซอร์สำหรับวงจรตรวจสอบสถานะช่องจอดรถ 24

4.1.2 (ข) การทดสอบการระบบ Chatbotตรวจสอบสถานะช่องจอดรถ 24

4.2 ผลการทดสอบ 29

4.3 อุปกรณ์ที่ประกอบสำเร็จ 29

สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปปัญหาและข้อเสนอแนะ	30
5.1 สรุปผลการทดลอง	30
5.2 ปัญหาและอุปสรรคในการดำเนินงาน	30
5.3 ข้อเสนอแนะ	30
บรรณานุกรม	31
ผนวก ก โค้ดที่ใช้ในการพัฒนาระบบลานจอดรถสั่งการด้วยเสียง	32
ผนวก ข คู่มือในการใช้งานระบบลานจอดรถสั่งการด้วยเสียง	49
ประวัติผู้จัดทำ	51

สารบัญตาราง

หน้า

ตารางที่ ก.1 โค้ดสำหรับส่งสถานะเซ็นเซอร์ช่อง A 1-4 ขึ้น Firebase ใน โปรแกรม Arduino 33

ตารางที่ ก.2 โค้ดสำหรับส่งสถานะเซ็นเซอร์ช่อง B 5-8 ขึ้น Firebase ใน โปรแกรม Arduino 34

ตารางที่ ก.3 โค้ดสำหรับส่งค่านับจำนวนผู้ที่มาใช้จานจอดรถในแต่ละวันของ ใน โปรแกรม Arduino 37

ตารางที่ ก.4 โค้ดสำหรับ Chatbot ของ Dialogflow Fulfillment 41

สารบัญภาพประกอบ

หน้า

ภาพที่ 2.1.3 IOT 5

ภาพที่ 2.1.4 Chatbot 6

ภาพที่ 2.1.6 NodeMCUESP8266 7

ภาพที่ 2.2.2 Node.js 7

ภาพที่ 2.2.3 Arduino 8

ภาพที่ 2.2.4 Infrared Sensor 8

ภาพที่ 2.2.5 Ultrasonic Sensor 9

ภาพที่ 3.1 ผังงานแสดงแผนการดำเนินโครงการ 12

ภาพที่ 3.2 แผนภาพแสดงการเชื่อมต่อของระบบโดยรวม 14

ภาพที่ 3.3 ผังงานแสดงแผนการออกแบบระบบ Chatbot 15

ภาพที่ 3.4 แผนผังแสดงการออกแบบระบบโมเดลลานจอดรถ 16

ภาพที่ 3.5 แผนภาพการเชื่อมต่อระหว่าง Infrared Sensor กับบอร์ดไมโครคอนโทรลเลอร์ตัวที่ 1-4 17

ภาพที่ 3.6 แผนภาพการเชื่อมต่อระหว่าง Infrared Sensor กับบอร์ดไมโครคอนโทรลเลอร์ตัวที่ 5-8 18

ภาพที่ 3.7 แผนภาพการเชื่อมต่อระหว่าง Infrared Sensor และ Ultrasonic Sensor กับบอร์ดไมโครคอนโทรลเลอร์ 19

ภาพที่ 3.8 แผนภาพการออกแบบโมเดลลานจอดรถ 20

ภาพที่ 3.9 แผนภาพการติดตั้งเซ็นเซอร์ตรวจจับในโมเดลลานจอดรถ 21

ภาพที่ 3.10 แผนภาพการติดตั้งเซ็นเซอร์ตรวจจับสำหรับนับจำนวนรถ 22

ภาพที่ 3.11 แผนผังแสดงการออกแบบระบบนับจำนวนรถที่ใช้งานในแต่ละวัน 23

ภาพที่ 4.1 การทดสอบของระบบลานจอดรถกรณีมีรถจอดโซน A 25

ภาพที่ 4.2 การทดสอบของระบบลานจอดรถกรณีมีรถจอดโซน B 25

ภาพที่ 4.3 การทดสอบของระบบลานจอดรถกรณีไม่มีรถจอดทุกช่อง 26

สารบัญภาพประกอบ(ต่อ)

ภาพที่ 4.4 การทดสอบของระบบลานจอดรถกรณีมีรถจอดทุกช่อง 26
ภาพที่ 4.5 ก. การทดสอบของระบบลานจอดรถกรณีมีรถผ่านระบบเซนเซอร์ตรวจจับ 27
ภาพที่ 4.6 ตัวอย่างการทำงานของระบบ Chatbot 28
ภาพที่ 4.7 อุปกรณ์ที่ประกอบสำเร็จ 29
ภาพ ข.1 คู่มือการใช้งาน Chatbot ลานจอดรถอัจฉริยะ 50

บทที่ 1
บทนำ

1.1 ความสำคัญของปัญหา

รถยนต์นับว่าเป็นปัจจัยที่จำเป็นต่อมนุษย์ในการเดินทาง จากสถิติกรมกการขนส่งทางบก รถจดทะเบียนสะสมทั่วประเทศมีปริมาณมากขึ้นทุกปี และในปี 2561 มีการจดทะเบียนสะสม มีจำนวนทั้งสิ้น 38,428,304 คัน ประกอบด้วย รถจักรยานยนต์จำนวน 20,548,178 คัน รถยนต์นั่งส่วนบุคคลไม่เกิน 7 คน จำนวน 8,796,543 คัน รถยนต์บรรทุกส่วนบุคคลจำนวน 6,451,730 คัน รถยนต์นั่งส่วนบุคคลเกิน 7 คน จำนวน 424,922 คัน ส่วนรถโดยสารประจำทางรวมจำนวนทั้งสิ้น 78,390 คัน ในขณะที่รถโดยสารไม่ประจำทางสะสมจำนวน 68,015 คัน และรถบรรทุกจำนวน 1,090,765 คัน (ข้อมูล ณ วันที่ 31 มกราคม 2561) ซึ่งแน่นอนว่าจำนวนรถที่เพิ่มขึ้นทุกๆ ปีนั้น มีผลกระทบต่อสถานที่ในการจอดรถ เช่น ห้างสรรพสินค้า สถานที่ท่องเที่ยว และสถานที่ติดต่อทางราชการ

ดังนั้นโครงงานนี้จึงได้นำเสนอระบบการตรวจสอบที่จอดรถผ่านสมาทโฟนโดยใช้การสั่งงานผ่านเสียงที่ออกแบบมาเพื่อตรวจสอบ จำนวนที่ว่างก่อนมาถึงจุดหมาย สามารถรู้ตำแหน่งที่ว่างใกล้ที่สุด ของผู้ใช้บริการ

ระบบการการตรวจสอบที่จอดรถผ่านสมาทโฟน จะเป็นสิ่งอำนวยความสะดวกโดยใช้คำสั่งเสียงในการค้นหาที่จอดรถ คุณสามารถค้นหาที่จอดรถก่อนถึงจุดหมายได้ในขณะขับรถโดยให้ไม่เสียสมาธิ สามารถระบุตำแหน่งพื้นที่ว่างที่ใกล้ที่สุดเมื่อคุณจะเข้าไปจอดในสถานที่นั้นๆ และยังสามารถและยังสามารถนับรถที่เข้ามาใช้บริการในแต่ละวันได้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ใช้ NodeMCU ESP8266 เพื่อเชื่อมต่อ เซ็นเซอร์เข้ากับ Firebase

1.2.2 เพื่อความสะดวกและรวดเร็วในการค้นหาที่จอดรถ

1.2.3 เพื่อพัฒนาระบบลานจอดรถโดยอาศัยการสั่งงานด้วยเสียง

1.2.4 เพื่อเก็บข้อมูลจำนวนผู้ใช้บริการลานจอดรถในแต่ละวัน

1.3 ขอบเขตของโครงงาน

1.3.1 ศึกษาการทำงานของเซ็นเซอร์ในแบบจำลองลานจอดรถ โดยใช้เซ็นเซอร์ Infrared Barrier Sensor ระยะทางของเซ็นเซอร์อยู่ที่ 2- 30 เซนติเมตร

1.3.2 ใช้ NodeMCU ESP8266 เพื่อเชื่อมต่อ เซ็นเซอร์เข้ากับ Firebase

1.3.3 ใช้ Firebase รับคำสั่งเสียงที่อยู่ใน Dialogflow

1.3.4 สร้างระบบลานจอดรถต้นแบบและทดสอบเพื่อให้ได้ตามวัตถุประสงค์

1.3.5 ตรวจสอบจำนวนรถยนต์ที่เข้ามาใช้ลานจอดรถโดยอุปกรณ์ตรวจจับอัตราโซนิค

1.3.6 ระบุตำแหน่งจอดรถที่ว่างและรายงานผลตำแหน่งที่ใกล้ที่สุดผ่านเสียงในสมาทโฟน

1.4 วิธีการดำเนินงาน

1.4.1 ศึกษาการทำงานของเซ็นเซอร์และอุปกรณ์ต่างๆที่เกี่ยวข้อง

1.4.2 ศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์

1.4.3 ศึกษาระบบการทำงาน อินเตอร์เน็ตของสรรพสิ่ง (IOT)

1.4.4 ศึกษาการทำงานของ Chatbot

1.4.5 ออกกแบบระบบ Development of Smart Car Parking

1.4.6 ทดสอบระบบโดยใช้แบบจำลอง

1.4.7 วิเคราะห์และปรับปรุงแก้ไขระบบ

1.4.8 จัดทำคู่มือปริญญานิพนธ์

1.5 แผนการดำเนินงาน

รายการ	พ.ศ. 2561				พ.ศ. 2561		
	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาหาข้อมูล	■	■					
2. ออกกแบบระบบ			■				
3. สร้างชิ้นงาน				■			
4. ทดสอบระบบโดยใช้แบบจำลอง					■		
5. วิเคราะห์และปรับปปรุงแก้ไขระบบ					■		
6. สรุปผลการทำงานและจัดรูปเล่มปริญญานิพนธ์						■	■

1.6 **ประโยชน์ที่คาดว่าจะได้รับ**

1.6.1 ผู้ใช้งานได้รับความสะดวกและรวดเร็ว

1.6.2 เพื่อใช้งานระบบลานจอดรถโดยอาศัยการสั่งงานด้วยเสียง

1.6.3 ช่วยในการเก็บข้อมูลจำนวนผู้ใช้บริการลานจอดรถในแต่ละวัน

1.6.4 ลดการสิ้นเปลืองพลังงานโดยไม่จำเป็นของผู้ใช้บริการในการค้นหาที่จอดรถ

บทที่ 2
เอกสารและทฤษฎีที่เกี่ยวข้อง

ในการศึกษาทฤษฎีที่เกี่ยวข้อง ผู้จัดทำโครงงานได้ศึกษาเอกสารในหัวข้อต่อไปนี้

2.1 ซอฟต์แวร์ที่เกี่ยวข้อง

- 2.1.1 Firebase
- 2.1.2 Dialogflow
- 2.1.3 IOT
- 2.1.4 Chatbot
- 2.1.5 Google Assistant

2.2 ฮาร์ดแวร์ที่เกี่ยวข้อง

- 2.2.1 ESP8266
- 2.2.2 Node.js
- 2.2.3 Arduino
- 2.2.4 infrared Sensor
- 2.2.5 Ultrasonic Sensor

2.3 งานวิจัยที่เกี่ยวข้อง

- 2.3.1 การออกแบบและพัฒนาแบบจำลองลานจอดรถอัจฉริยะ
- 2.3.2 การออกแบบและสร้างแบบจำลองลานจอดรถอัจฉริยะ
- 2.3.3 ชุดจำลองลานจอดรถแบบอัจฉริยะ

2.1 ซอฟต์แวร์ที่เกี่ยวข้อง

2.1.1 ความหมายของ Firebase

Firebase [6] คือ ผู้ช่วยส่วนตัวที่สามารถสั่งงานด้วยเสียง แยกแยะเสียงพูดได้ จากเดิมที่ให้บริการอยู่บนมือถือ Google Pixel และลำโพง Google Home (กุเกิ้ล โฮม) กำลังจะขยายไปให้บริการบน Android ออกเป็นแอพพลิเคชั่นบน iOS และอุปกรณ์อื่นๆ ซึ่ง Firebase มีความสามารถในการตอบรับ โต้ตอบกลับคล้ายกับความสามารถของ Siri และมีความสามารถในการเข้าถึงการทำงานต่างๆ

2.1.2 ความหมายของ Dialogflow

Dialogflow [5] หรือ Api.ai เป็น product ที่ถูกพัฒนาขึ้นโดย Speaktoit แต่ถูก Google ซื้อ และนำไปพัฒนาต่อยอดในปี 2016 และพึ่งเปลี่ยนชื่อมาเป็น Dialogflow เป็นส่วนที่ทำหน้าที่ทำความความใจสิ่งที่ผู้ใช้งานพูด หรือสื่อสารออกมา ซึ่งถ้าดูจากภาพจะพบว่า Agent ก็จะคือส่วนสีส้มๆ ที่มีหน้าที่ในการรับ Query หรือ Input ที่ผู้ใช้ส่งเข้ามา (ผ่านอุปกรณ์ใดๆ) จากนั้น Agent ก็จะทำ NLU เพื่อดูว่าผู้ใช้ต้องการอะไร หมายถึง Intent ไหน จากนั้นก็ Response Actionable data กลับไปหาผู้ใช้งาน

2.1.3 ความหมายของ IOT

IOT [10] คือ เครือข่ายวัตถุอุปกรณ์พาหนะสิ่งปลูกสร้างและสิ่งของอื่นๆที่มีวงจรอิเล็กทรอนิกส์ ซอฟต์แวร์เซ็นเซอร์และการเชื่อมต่อกับเครือข่ายฝังตัวอยู่และทำให้วัตถุเหล่านั้นสามารถเก็บบันทึกและแลกเปลี่ยนข้อมูลได้ อินเทอร์เน็ตของสรรพสิ่งทำให้วัตถุสามารถรับรู้สภาพแวดล้อมและถูกควบคุมได้จากระยะไกลผ่านโครงสร้างพื้นฐานเครือข่ายที่มีอยู่แล้ว ทำให้เราสามารถผสานโลกกายภาพกับระบบคอมพิวเตอร์ได้แนบแน่นมากขึ้น ผลที่ตามมาคือประสิทธิภาพที่ ความแม่นยำ และประโยชน์ทางเศรษฐกิจที่เพิ่มมากขึ้น

ภาพที่ 2.1.3 IOT

2.1.4 ความหมายของ Chatbot

Chatbot [11] เป็นโปรแกรมที่พัฒนาขึ้นเพื่อจำลองบทสนทนาของมนุษย์ ให้สามารถพูดคุย สื่อสารกับมนุษย์ผ่านทางเสียงหรือข้อความแบบ real-time โดย Chatbot นั้น ได้มีการพัฒนาขึ้นมา 2 แบบ คือ

Rule-Based Bot (กำหนดด้วยกฎต่างๆ) และ AI Bot (ปัญญาประดิษฐ์) โดย Rule-Based Bot นั้น จะทำงานตามกฎและคีย์เวิร์ดที่ถูกกำหนดไว้ ถ้าหากคำถามไม่ตรงกับที่กำหนดไว้ Chatbot อาจให้คำตอบที่ไม่ดีหรือทำงานผิดพลาดได้

ภาพที่ 2.1.4 Chatbot

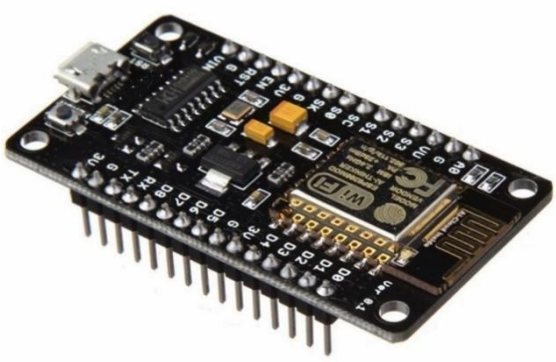
2.1.5 ความหมายของ Google Assistant

Google Assistant [4] คือระบบสั่งงานด้วยเสียงจากคำพูดของเรา ปกติแล้วระบบนี้จะอยู่ในมือถือ Google Pixel ที่ทาง Google พัฒนาขึ้นมาเอง แต่ตอนนี้ทาง Google คงเห็นว่าพร้อมแล้วที่จะพา Google Assistant ออกมาโลดแล่นยังมือถือรุ่นอื่นๆ ได้แล้วจึงตัดสินใจปล่อยออกมาให้ดาวน์โหลดสำหรับมือถือแอนดรอยด์ทั่วไปสามารถใช้งานได้ ร่วมถึง iOS ก็เช่นกัน

2.2 ฮาร์ดแวร์ที่เกี่ยวข้อง

2.2.1 ความหมายของ NodeMCU ESP8266

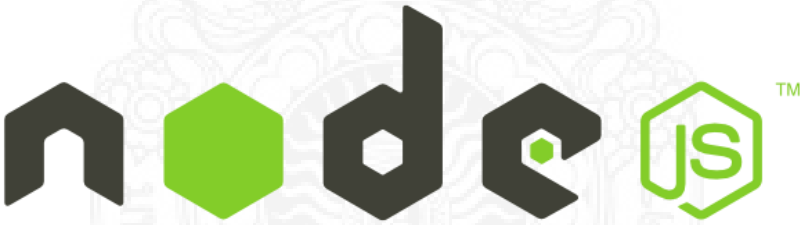
NodeMCU ESP8266 [7] คือ โมดูล wifi ภายในมีเฟิร์มแวร์ทำงานในลักษณะ Serial-to-WiFi ที่ช่วยให้อุปกรณ์อื่นๆ เช่น MCU สามารถต่อเข้ากับ internet ได้โดยใช้ port serial (ขา Tx, ขา Rx) และใช้คำสั่งในการควบคุมการทำงาน



ภาพที่ 2.2.1 NodeMCU ESP8266

2.2.2 ความหมายของ Node.js

Node.js [8] คือ เทคโนโลยีฝั่ง Server Side ที่ถูกพัฒนาด้วยภาษา JavaScriptเดิมทีภาษา JavaScript ทำงานฝั่ง Client เป็นหลัก NodeJS เป็น Client หรือ Server ก็ขึ้นอยู่กับจุดประสงค์ของผู้ใช้งานนั้น แต่จุดตั้งต้นเริ่มมาจาก Server Side เป็นหลัก ปัจจุบัน NodeJS ถูกนำมาทำเป็น Web Server , Mobile Hybrid , IOT , Webkit , TVOS ,OS อื่นๆอีกมาก เรียกได้ว่าเข้าถึงได้หลากหลายเทคโนโลยี



ภาพที่ 2.2.2 Node.js

2.2.3 ความหมายของ Arduino

Arduino [9] คือ เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบOpen Source คือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และSoftwareตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ดังนั้นจึงเหมาะสำหรับผู้เริ่มต้นศึกษา ทั้งนี้ผู้ใช้งานยังสามารถดัดแปลง เพิ่มเติม พัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้อีกด้วย

ภาพที่ 2.2.3 Arduino

2.2.4 ความหมายของ Infrared Sensor

Infrared Sensor [12] คือ โมดูลอ่านค่าสะท้อนกลับของแสง ใช้ไฟ 3.3-5V เหมาะสำหรับใช้กับ Arduino ให้เอาต์พุตออกมา 2 แบบคือแบบดิจิตอลสามารถปรับค่าที่ต้องการได้เมื่อค่าที่อ่านได้ถึงระดับที่ต้องการก็จะส่งค่า 1 ออกมา ถ้ายังไม่ถึงระดับก็จะส่งค่า 0 ออกมา และอีกแบบคือเอาต์พุตแบบอนาล็อก

ภาพที่ 2.2.4 Infrared Sensor

2.2.5 ความหมายของ Ultrasonic Sensor

Ultrasonic Sensor [13] คือ อุปกรณ์อิเล็กทรอนิกส์ สำหรับวัดระยะห่างด้วยคลื่นอัลตราโซนิค (ใช้คลื่นเสียงความถี่ ประมาณ 40kHz) มีสองส่วนหลักคือ ตัวส่งคลื่นที่ทำหน้าที่สร้างคลื่นเสียงออกไปในการวัดระยะแต่ละครั้ง แล้วเมื่อไปกระทบวัตถุหรือสิ่งกีดขวาง คลื่นเสียงถูกสะท้อนกลับมายังตัวรับแล้วประมวลผลด้วยวงจรอิเล็กทรอนิกส์ภายในโมดูลถ้าจับเวลาในการเดินทางของคลื่นเสียงในทิศทางไปและกลับ และถ้าทราบความเร็วเสียงในอากาศ ก็จะสามารถคำนวณระยะห่างจากวัตถุกีดขวางได้

ภาพที่ 2.2.5 Ultrasonic Sensor

2.3 งานวิจัยที่เกี่ยวข้อง

2.3.1 การออกแบบและพัฒนาแบบจำลองลานจอดรถอัจฉริยะ

ดิเรกกาฬ ภักดี, นิทัศน์ สุภรัตน์ และ ปรีชา เอี่ยมราคิน. (2014). การออกแบบและพัฒนาแบบจำลองลานจอดรถอัจฉริยะได้ออกแบบและพัฒนาแบบจำลองการจอดรถอัจฉริยะ เป็นการพัฒนาจากโครงงานเดิมที่ถูกสร้างขึ้นเมื่อปี พ.ศ. 2546 ให้มีระบบการจัดการที่มีประสิทธิภาพมากขึ้น เจ้าหน้าที่ทำการเลือกตำแหน่งที่จอดรถที่ทางเข้าอาคารจอดรถ หลังจากนั้นผู้ใช้จะได้รับบัตรจอดรถเป็นรหัสบาร์โค้ดซึ่งมีข้อมูลทะเบียนรถของผู้ใช้ สำหรับใช้ในการยืนยันการเข้าจอดลานจอดรถเพื่อเป็นการรักษาความปลอดภัยให้กับเจ้าของรถ เมื่อผู้ใช้เข้าจอด ณ ตำแหน่งที่ระบุแล้วที่หน้าจอโปรแกรมจะแสดงสถานะที่ตำแหน่งที่จอดรถนั้นเพื่อเป็นการแจ้งให้เจ้าหน้าที่ทราบว่าตำแหน่งนี้มีผู้จอดแล้วแต่ถ้ามีผู้จอดรถผิดตำแหน่งที่เลือกไว้จะมีการแสดงสถานะที่หน้าจอคอมพิวเตอร์และมีการแจ้งเตือนผ่านทางเจ้าหน้าที่ประจำอาคาร หลังจากนั้นเมื่อผู้ใช้ต้องการจะออกจากลานจอดรถ จะใช้บัตรจอดรถเข้าเครื่องอ่านบาร์โค้ดที่จุดประตูทางออกเพื่อตรวจสอบรหัสบาร์โค้ด ในการแสดงข้อมูลของรถที่จะออกจากอาคาร

2.3.2 การออกแบบและสร้างแบบจำลองลานจอดรถอัจฉริยะ

เทพฤทธิ์ บุญยืน, นวพล เปี่ยมปิยชาติ และ อัศนัย ทุนดี. (2014). การออกแบบและสร้างแบบจำลองลานจอดรถอัจฉริยะ.โครงงานวิศวกรรมศาสตร์นี้นำเสนอการออกแบบและสร้างลานจอดรถอัจฉริยะเพื่อแก้ไขปัญหาเบื้องต้นของการหาตำแหน่งจอดรถในลานจอดรถขนาดใหญ่โดยการจอดรถจะประกอบด้วยชุดเชื่อมต่อกับผู้ใช้ในลักษณะกราฟิกตรงทางเข้าของลานจอดรถซึ่งผู้ใช้สามารถเรียกตำแหน่งจอดรถที่ว่างผ่านหน้าจอคอมพิวเตอร์ได้ดังนั้นผู้ใช้สามารถเลือกตำแหน่งจอดรถเพื่อให้เกิดความสะดวกรวดเร็วเมื่อตำแหน่งจอดรถถูกเลือกจะมีระบบไฟนำทางซึ่งติดตั้งบนถนนโดยใช้ LEDs นำผู้ใช้ไปยังตำแหน่งปลายทางเมื่อรถจอดเรียบร้อย เซ็นเซอร์ตรงตำแหน่งจอดรถซึ่งใช้โฟโต้ทรานซิสเตอร์ร่วมกับอินฟราเรดทำการส่งสัญญาณไปแสดงผลที่หน้าจอคอมพิวเตอร์โดยแบบจำลอง โดยแบบจำลองที่สร้างจะมีขนาดจำนวน 3 ชั้น และ ตำแหน่งจอด 24 ตำแหน่ง

2.3.3 ชุดจำลองลานจอดรถแบบอัจฉริยะ

กิตติวร รัตนโชติ, พงษ์ ศักดิ์ จันทร์ไทย และ พิพัฒน์ อินทร์มณี. (2014). ชุดจำลองลานจอดรถแบบอัจฉริยะ. โครงงานชุดสาธิตการจอดรถอัจฉริยะสร้างขึ้นเพื่อเป็นตัวอย่างในการจัดระบบการจอดรถในสถานที่ประกอบกาต่างๆอาทิ เช่น อาคาร บริษัท ห้างสรรพสินค้า เพื่อให้ผู้ใช้ยานพาหนะมีความสะดวกและความปลอดภัยมากขึ้นโดยชุดสาธิตการจอดรถอัจฉริยะนี้ใช้ไมโครคอนโทรลเลอร์เบอร์ 89451 RD2 ควบคุมการทำงานและใช้เซ็นเซอร์อินฟราเรดตรวจจับสภาวะการเข้าของออกของรถซึ่งส่งไปยังไมโครคอนโทรลเลอร์เพื่อทำการประมวลผลทำให้ทราบว่ามีที่จอดรถอยู่ที่ในชั้นนั้นและมีการจัดระบบป้องกันการขโมยรถโดยใช้บัตรแถบแม่เหล็กโดยผู้ที่นำรถเข้ามาจอดนั้นจะได้รับบัตรแถบแม่เหล็กโดยบัตรนี้จะมีรหัสข้อมูลตรงกับช่องจอดนั้น ๆ ถ้านำรถออกไปผิดคันจะมีสัญญาณเตือนภัยโดยทันที และ กล้องจะทำงานหมุนไปยังตำแหน่งที่เกิดขึ้นว่ามีสิ่งผิดปรกติอย่างไรแล้วทำการแก้ปัญหาที่เกิดขึ้น

บทที่ 3
ขั้นตอนและวิธีดำเนินการ

การออกแบบการพัฒนาระบบลานจอดรถสั่งการด้วยเสียง (Development of Smart Car Parking System using Voice Command) มีวัตถุประสงค์เพื่อออกแบบและการพัฒนาระบบลานจอดรถสั่งการด้วยเสียง ขึ้นมาจริง โดยขั้นตอนการออกแบบโครงงานมีรายละเอียด ดังนี้

3.1 แผนการดำเนินโครงงาน

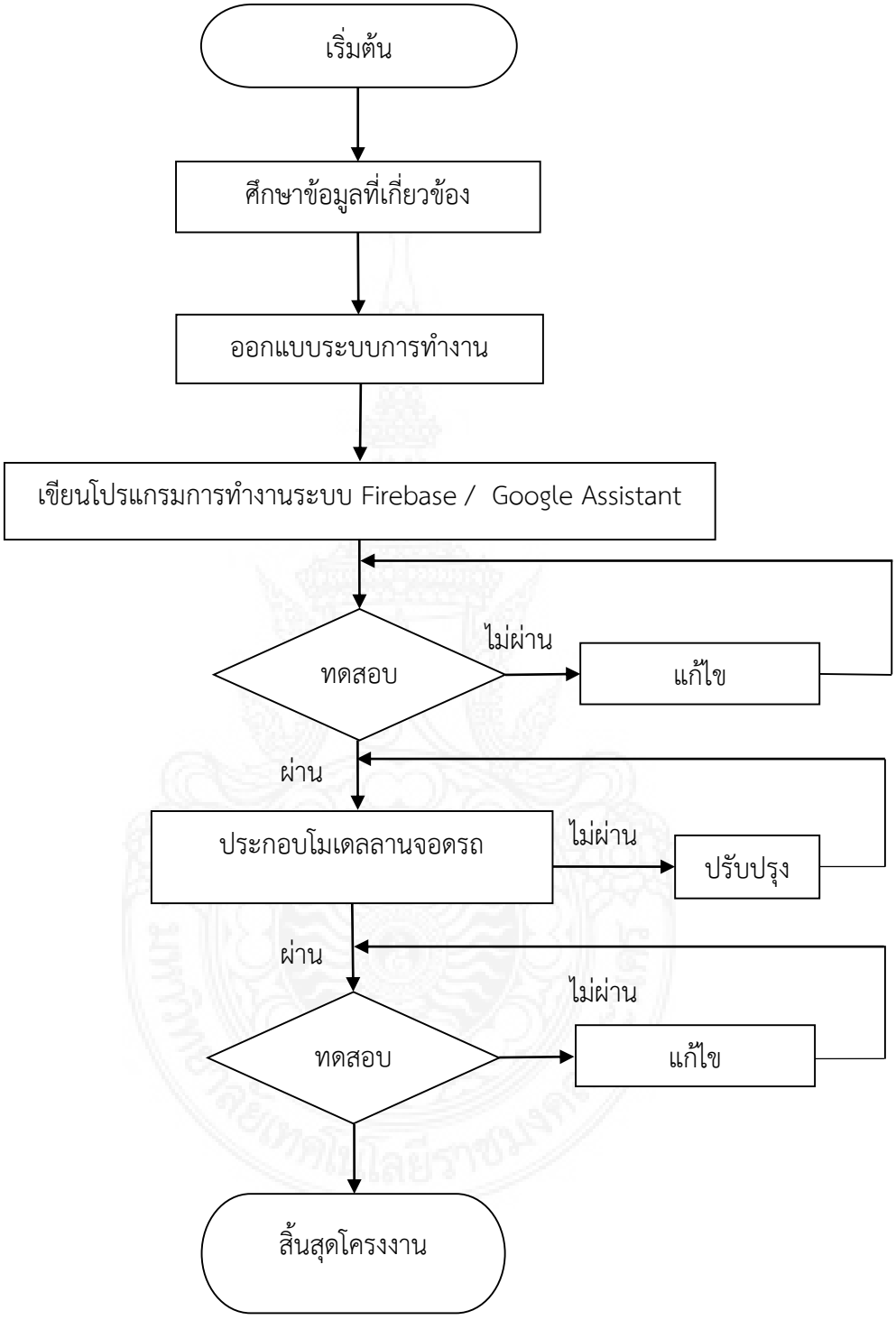
3.2 ขั้นตอนการทำโครงงาน

3.3 โครงสร้างของระบบการทำงาน

3.1 แผนการดำเนินโครงงาน

จากการศึกษาทฤษฎีที่เกี่ยวข้องสามารถเขียนแผนผังแสดงขั้นตอนและวิธีการดำเนินงานต่างๆ เป็นแผนภาพการทำงานได้ดังภาพที่ 3.1

เริ่มจากศึกษาข้อมูลของการทำงานของเซ็นเซอร์ในแบบจำลองลานจอดรถ ระบบฐานข้อมูล Firebase และอุปกรณ์ที่จะใช้ในโครงงานนี้แล้วนำข้อมูลที่ศึกษาทั้งหมดมาสรุปและเริ่มทำการออกแบบระบบการทำงานในการวางแผนระดับการทำงานของวงจรทั้งหมดเพื่อให้การทำงานมีความชัดเจนเป็นขั้นตอนหลังจากที่ออกแบบแล้ว เริ่มการเขียนโปรแกรมการทำงาน ให้สามารถติดต่อเชื่อมต่อระบบฐานข้อมูลถึงกันได้เมื่อเสร็จขั้นตอนการเขียนโปรแกรม ทำการทดสอบระบบการทำงานของ Chatbot หากเกิดข้อผิดพลาดก็ทำการแก้ไขและทดสอบใหม่ หากทดสอบผ่านก็ทำการออกกแบบระบบลานจอดรถจำลอง Development of Smart Car Parking System using Voice Command เมื่อเสร็จเริ่มทดสอบระบบหากเกิดข้อผิดพลาดก็ทำการแก้ไขและทดสอบใหม่ หากทดสอบผ่านเป็นการเสร็จสิ้นแผนการดำเนินโครงงาน



ภาพที่ 3.1 ผังงานแสดงแผนการดำเนินโครงงาน

3.2 ขั้นตอนการทำโครงงาน

3.2.1 ขั้นตอนการทำโครงงาน

1. ศึกษาข้อมูลที่เกี่ยวข้องกับการทำระบบลานจอดรถอัจฉริยะ
   - นับจำนวนรถเข้า-ออก
   - หาที่จอดรถที่มีช่องว่าง
2. ศึกษาหลักการทำงานของอุปกรณ์
   - บอร์ด NodeMCU ESP8266
   - Infrared Sensor
   - Ultrasonic Sensor
   - Arduino
3. ศึกษา Chatbot ของแอพพลิเคชั่น Firebase, Dialogflow
   - การทำ Chatbot โต้ตอบ
4. ศึกษาโปรแกรม Firebase
   - การรับค่า-ส่งค้า
   - การจัดเก็บค่า

3.2.2 ออกแบบและพัฒนาระบบ

- ออกแบบโมเดลลานจอดรถ
- พัฒนาระบบ Chatbot

3.2.3 ทดสอบการใช้งาน

- จำลองโมเดลลานจอดรถ

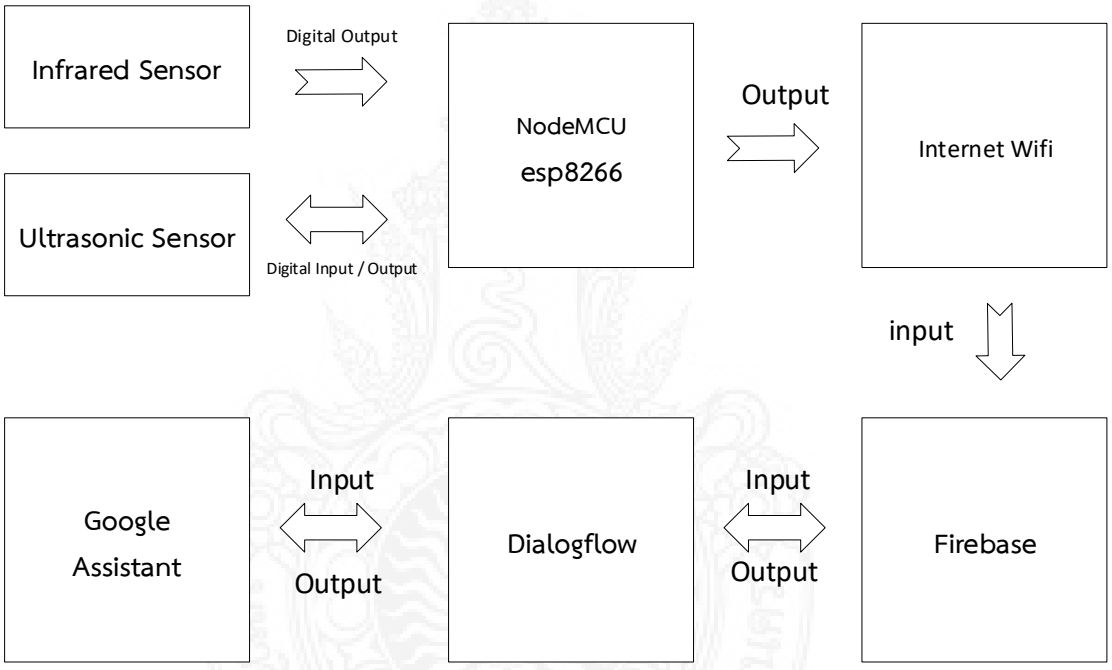
3.2.4 จัดทำเอกสารโครงงาน

- ทำวิจัย 5 บท

3.3 โครงสร้างของระบบการทำงาน

3.3.1 ภาพรวมของระบบ

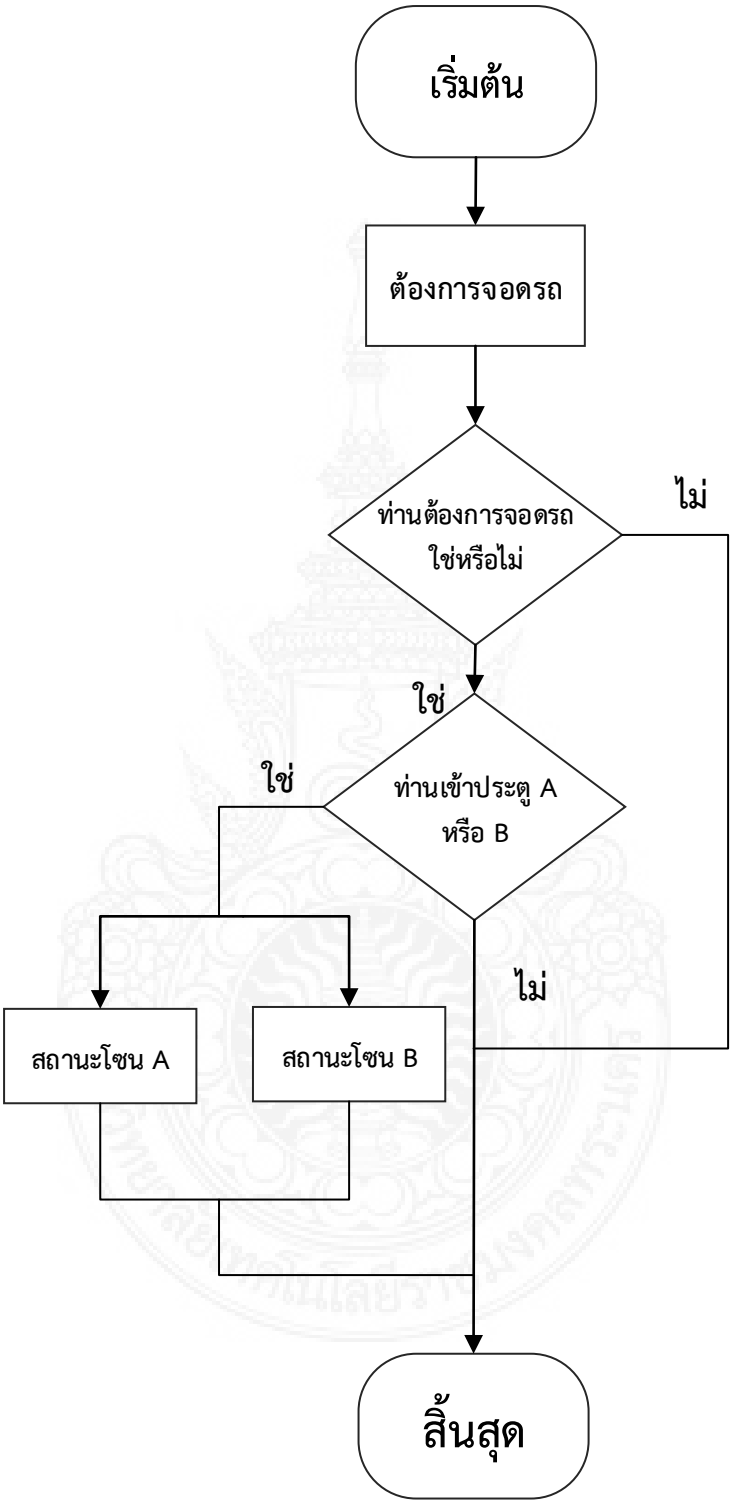
เมื่อผู้ใช้ลานจอดรถเข้ามาจอดในช่องจอดรถเซนเซอร์ทำการตรวจจับรถที่เข้ามาจอดในช่องจอดรถเพื่อส่งค่าที่ได้ไปยังบอร์ดไมโครคอนโทรเลอร์ NodeMCU ESP8266 จากนั้นบอร์ดไมโครคอนโทรเลอร์ NodeMCU ESP8266 ทำการส่งข้อมูลผ่านเครือข่ายไร้สาย WIFI ไปยัง Firebase เพื่อเก็บลงยังฐานข้อมูล และ นำไปใช้ในส่วนของ Dialogflow ในการดึงค่าสถานะ นำไปแจ้งผ่าน Google Assistant ดังภาพที่ 3.2



ภาพที่ 3.2 แผนภาพแสดงการเชื่อมต่อของระบบโดยรวม

3.3.2 การออกแบบระบบ Chatbot

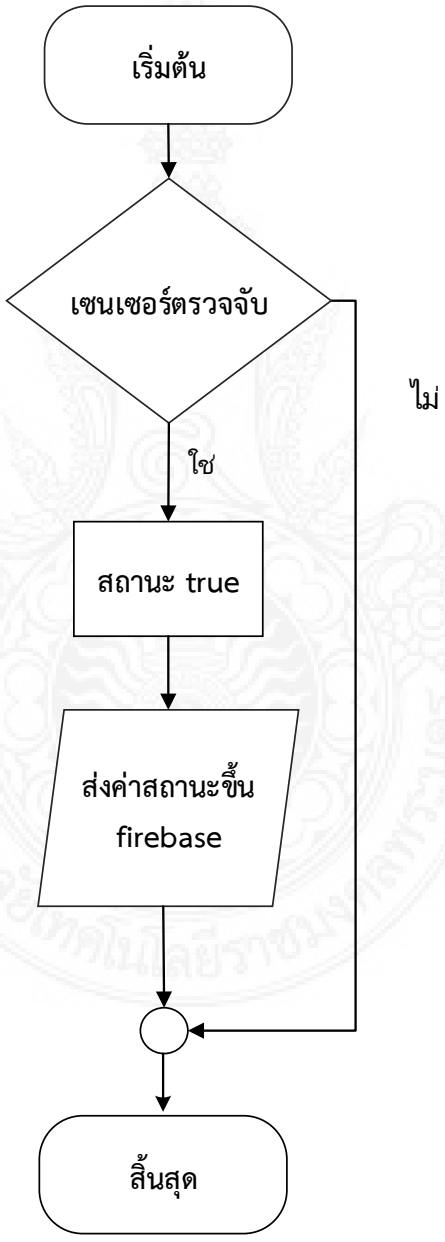
หลักการทำงานของระบบ Chatbot เริ่มจากใช้แอพพลิเคชั่น Google Assistant ถาม Chatbot ให้หาที่จอดรถแล้ว Chatbot จะถามกลับว่าต้องการจอดรถใช่หรือไม่ ถ้าตอบใช่ Chatbot จะถามต่อว่าจะจอดรถในโซน A หรือ B แล้วให้เลือกที่จอดรถ แล้ว Chatbot จะบอกจำนวนที่จอดรถในโซนนั้น หากตอบว่าไม่ระบบจะสิ้นสุดทันที ดังภาพที่ 3.2



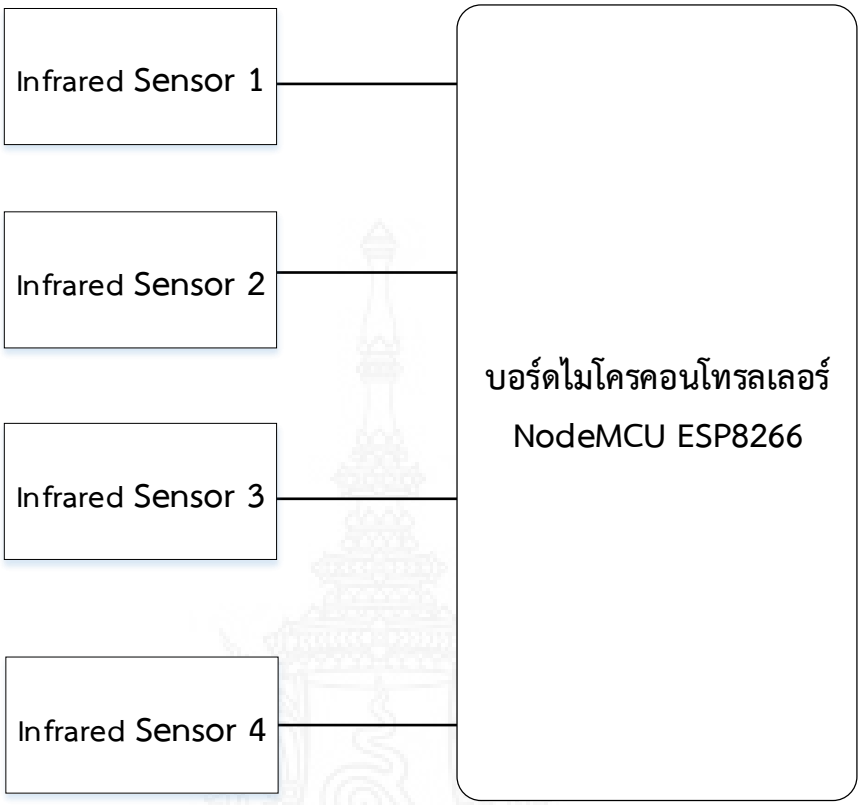
ภาพที่ 3.3 ผังงานแสดงแผนการออกแบบระบบ Chatbot

3.3.3 การออกแบบระบบโมเดลลานจอดรถ

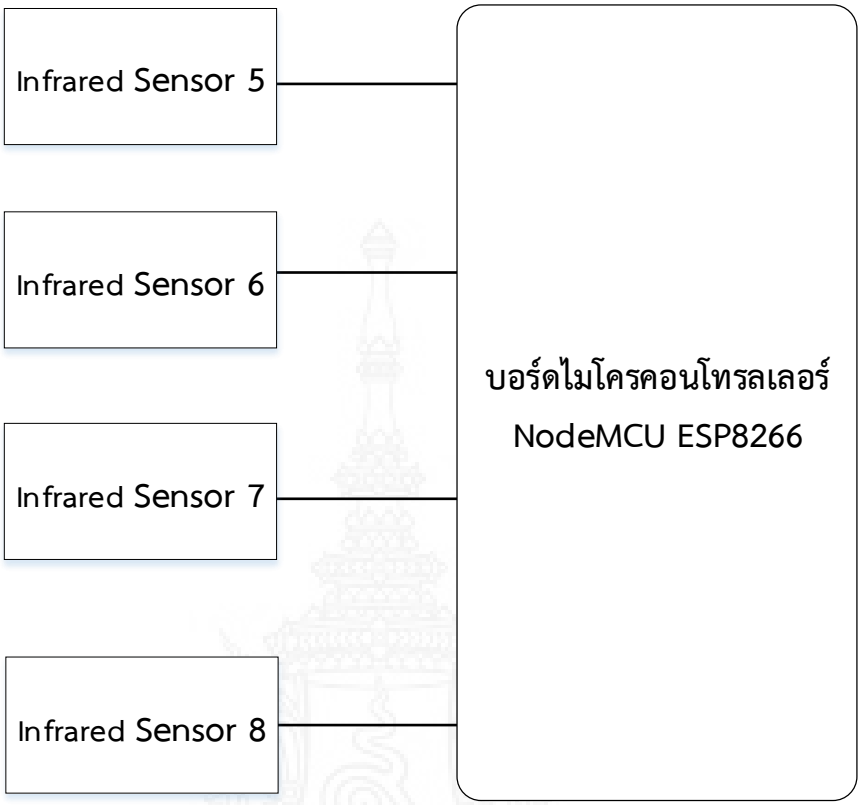
หลักการทำงานของโมเดลลานจอดรถจำลองผลการทำงานของระบบพบว่าเมื่อมีรถเข้าจอด ระบบเซนเซอร์ตรวจจับส่งค่าไปประมวลผลที่ไมโครคอนโทรเลอร์ NodeMCU ESP8266 แล้วในส่วนของ Firebase จะแสดงผลมีช่องว่าง เมื่อไมโครคอนโทรเลอร์ NodeMCU ESP8266 ทำการส่งค่าไปยังฐานข้อมูลเป็น true เมื่อมีรถเข้ามาจอดหน้าจอแสดงผลเปลี่ยนเป็นสถานะ ไม่ว่าง แล้ว เมื่อนำรถออกค่าจะเป็น false เป็นสถานะว่าง ดังภาพที่ 3.3



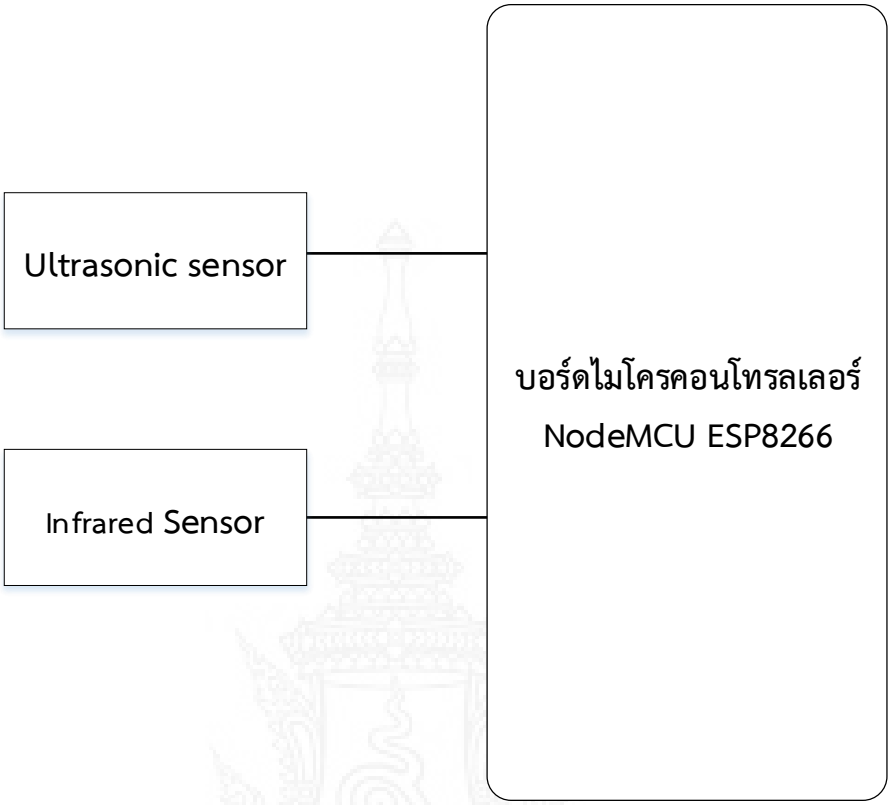
ภาพที่ 3.4 แผนผังแสดงการตรวจสอบรถในโมเดลลานจอดรถ



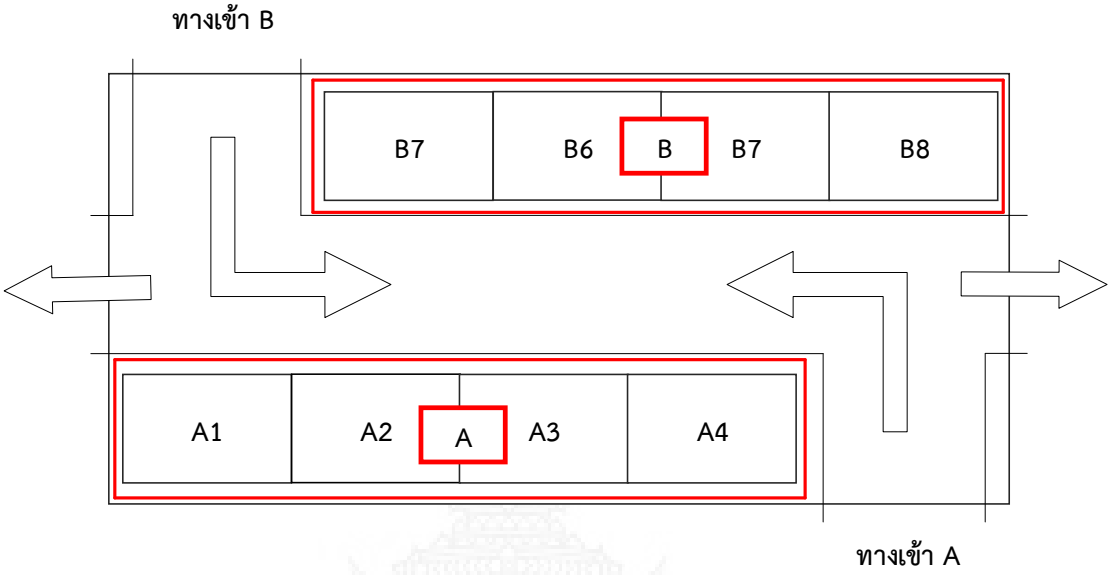
ภาพที่ 3.5 แผนภาพการเชื่อมต่อระหว่าง Infrared Sensor กับบอร์ดไมโครคอนโทรลเลอร์ตัวที่ 1-4



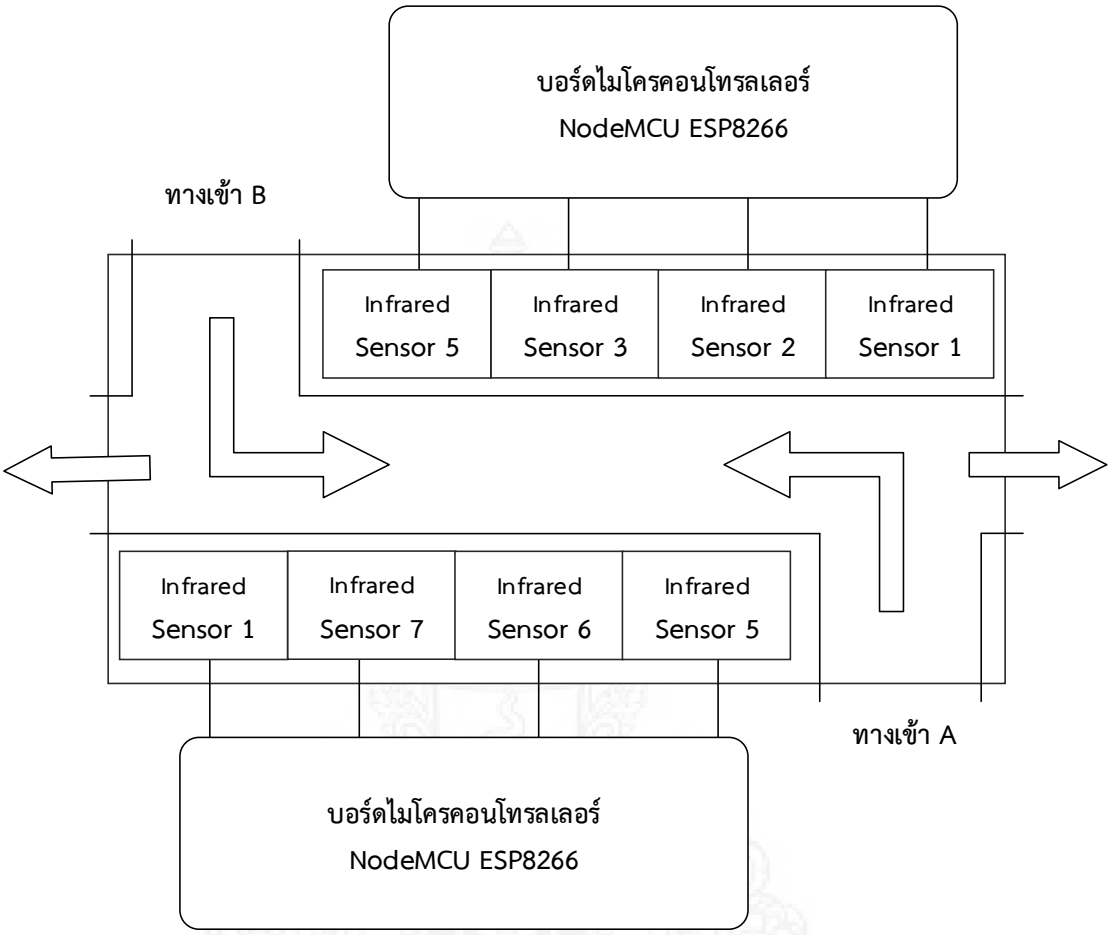
ภาพที่ 3.6 แผนภาพการเชื่อมต่อระหว่าง Infrared Sensor กับบอร์ดไมโครคอนโทรลเลอร์ตัวที่ 5-8



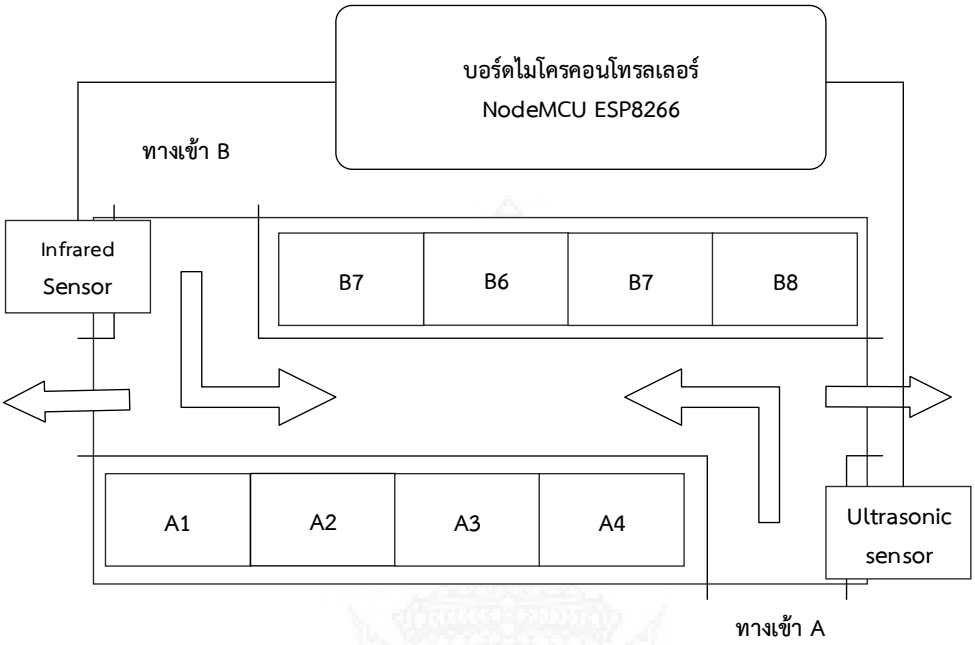
ภาพที่ 3.7 แผนภาพการเชื่อมต่อระหว่าง Infrared Sensor และ Ultrasonic Sensor กับบอร์ดไมโครคอนโทรลเลอร์



ภาพที่ 3.8 แผนภาพการออกแบบโมเดลลานจอดรถ



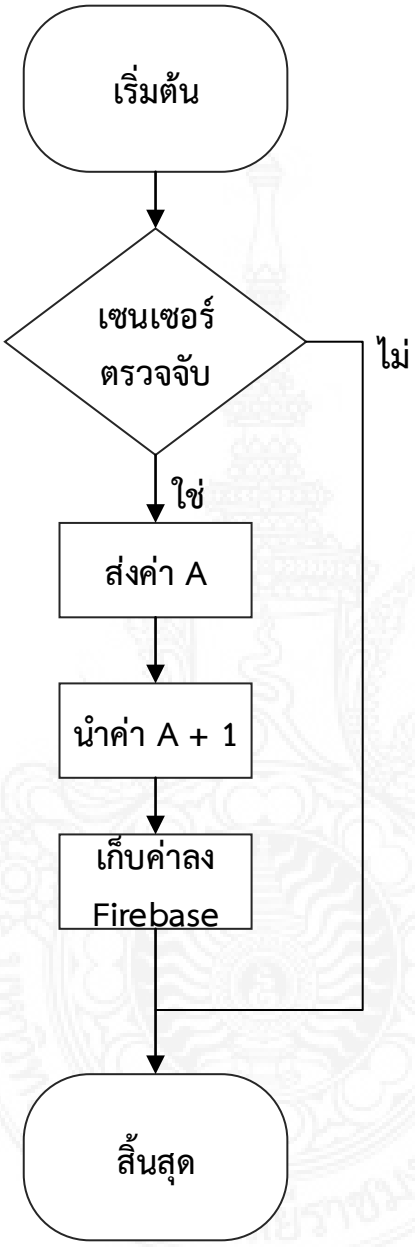
ภาพที่ 3.9 แผนภาพการติดตั้งเซ็นเซอร์ตรวจจับในโมเดลลานจอดรถ



ภาพที่ 3.10 แผนภาพการติดตั้งเซ็นเซอร์ตรวจจับสำหรับนับจำนวนรถ

3.3.4 การออกแบบระบบนับจำนวนรถที่ใช้งานในแต่ละวัน

หลักการทำงานของระบบนับจำนวนรถที่ใช้งานในแต่ละวันเมื่อมีรถผ่านระบบเซนเซอร์ตรวจจับจะส่งค่าไปประมวลผลเป็นค่า A และนำค่า A + 1ทำการส่งค่าไปยัง Firebase เพื่อเก็บลงยังฐานข้อมูล ดังภาพที่ 3.10



ภาพที่ 3.11 แผนผังแสดงการออกแบบระบบนับจำนวนรถที่ใช้งานในแต่ละวัน

บทที่ 4
ผลการดำเนินงาน

ในการศึกษาผลของการดำเนินงาน ผู้จัดทำโครงงานได้ศึกษาในหัวข้อต่อไปนี้

4.1 วิธีการทดสอบ

4.2 ผลการทดสอบ

4.3 อุปกรณ์ที่ประบกอบสำเร็จ

4.1 วิธีการทดสอบ

4.1.1 เครื่องมือที่ใช้ในการทดสอบ

ผู้จัดทำได้ทำการสร้างต้นแบบโมเดลลานจอดรถ โดยการออกแบบจะมีอยู่ 2 ส่วน คือ ส่วนของตัวระบบเซนเซอร์ตรวจจับและส่วนของระบบแอพพลิเคชั่น Google Assistant โดยมีระบบแชทบอทของ Dialogflow ในการคุยโต้ตอบกับผู้ใช้งาน

4.1.2 ขั้นตอนการทดลอง

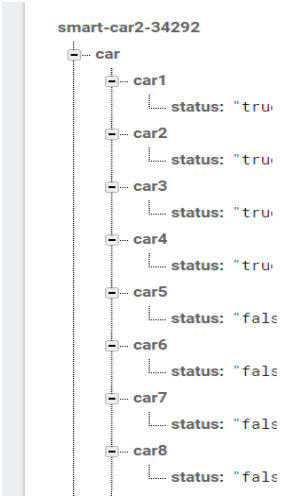
(ก) ส่วนของโมเดลลานจอดรถ มีระบบเซนเซอร์ตรวจจับ คอยเช็คสถานะไม่ว่างแล้วส่งค่าไปยัง Firebase ในการเก็บลงฐานข้อมูล

(ข) ทำการทดสอบ ระบบลานจอดรถสั่งการด้วยเสียง โดยการใช้แอพ Google Assistant ในการสอบถามสถานะช่องว่างในแบบจำลองลานจอดรถ

4.1.2 (ก) การทดสอบการตรวจจับของเซนเซอร์สำหรับวงจรตรวจสอบสถานะช่องจอดรถ

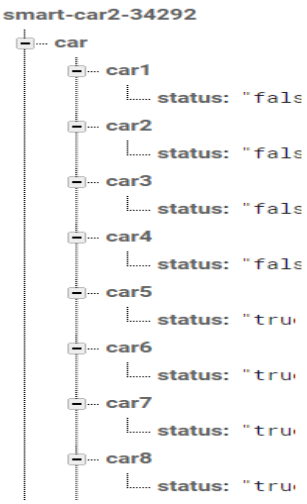
ระบบตรวจสอบสถานะช่องจอดรถ โดยใช้เซนเซอร์อินฟราเรดในการตรวจสอบสถานะช่องจอดรถโดยมีหลักการดังนี้

- กรณีที่มีรถยนต์เข้ามาจอดในช่องจอดรถ หรือ อยู่ในระยะการตรวจจับเซนเซอร์ที่กำหนดไว้จะส่งผลทำให้เซนเซอร์ในวงจรตรวจสอบสถานะช่องจอดรถช่องนั้นสั่งการไปที่บอร์ดเพื่อส่งค่า เป็น true ขึ้นฐานข้อมูล Firebase
- กรณีที่ยังไม่มีรถยนต์เข้ามาจอดในช่องจอดรถ หรือ ไม่อยู่ในระยะการตรวจจับของเซนเซอร์ที่กำหนดไว้จะไม่มีการส่งสถานะใด ๆ ทั้งสิ้น



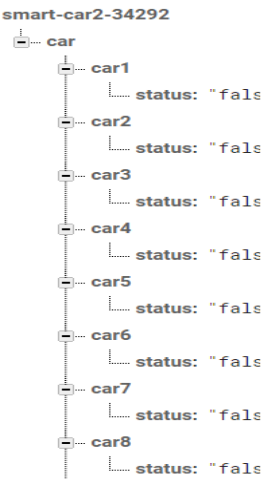
(ก) (ข)

ภาพที่ 4.1 การทดสอบของระบบลานจอดรถกรณีมีรถจอดโซน A ช่อง 1 - 4 (ก) ภาพลานจอดรถขณะมีรถจอดโซน A และ (ข) ระบบฐานข้อมูลแสดงสถานะเป็น true กรณีมีรถเข้ามาจอดโซน A



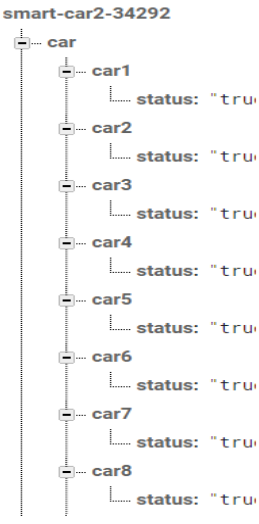
(ก) (ข)

ภาพที่ 4.2 การทดสอบของระบบลานจอดรถกรณีมีรถจอดโซน B ช่อง 5 - 8 (ก) ภาพลานจอดรถขณะมีรถจอดโซน B และ (ข) ระบบฐานข้อมูลแสดงสถานะเป็น true กรณีมีรถเข้ามาจอด โซน B



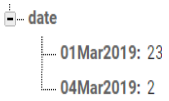
(ก) (ข)

ภาพที่ 4.3 การทดสอบของระบบลานจอดรถกรณีไม่มีรถจอดทุกช่อง (ก) ภาพลานจอดรถกรณีไม่มีรถจอดทุกช่อง และ (ข) ระบบฐานข้อมูลแสดงสถานะเป็น false กรณีที่ไม่มีรถเข้ามาจอดในช่องจอดรถ



(ก) (ข)

ภาพที่ 4.4 การทดสอบของระบบลานจอดรถกรณีมีรถจอดทุกช่อง (ก) ภาพลานจอดรถกรณีมีรถจอดทุกช่องและ (ข) ระบบฐานข้อมูลแสดงสถานะเป็น true กรณีที่มีรถจอดทุกช่อง



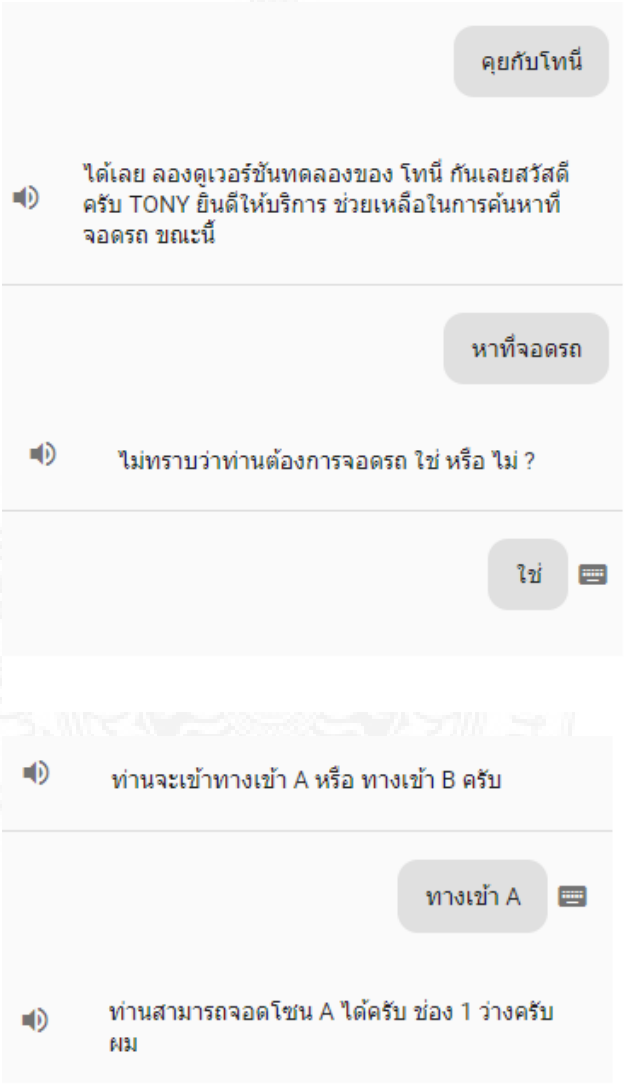
(ก) (ข)

ภาพที่ 4.5 การทดสอบของระบบลานจอดรถกรณีมีรถผ่านระบบเซนเซอร์ตรวจจับ (ก) ภาพลานจอดรถกรณีมีรถผ่านระบบเซนเซอร์ตรวจจับ และ (ข) ระบบฐานข้อมูลแสดงสถานะจำนวนรถ กรณีที่ยังมีรถยนต์ผ่านระบบเซนเซอร์ตรวจจับ

4.1.2 (ข) การทดสอบการระบบ Chatbotตรวจสอบสถานะช่องจอดรถ

ระบบ Chatbotตรวจสอบสถานะช่องจอดรถโดยมีหลักการดังนี้

- กรณีที่ต้องการจอดรถโดยการป้อนคำสั่ง “ คุยกับโทนี่” ระบบจะดึงข้อมูลเวอร์ชั่นทดลองของลานจอดรถขึ้นมาใช้งาน และ ถามหาที่จอดรถ ระบบจะถามกลับว่าท่านต้องการจอดรถใช่หรือไม่ถ้าตอบว่าใช้ ระบบจะสอบถามว่าท่านเข้าจอดรถทางประตู A หรือ ประตู B หากตอบ A ระบบจะเลือกช่องว่างที่ใกล้ที่สุดให้ท่าน ดังภาพที่ 4.1 ก.



ภาพที่ 4.6 ตัวอย่างการทำงานของระบบ Chatbot

4.2 ผลการทดสอบ

ในการทดสอบอุปกรณ์โมเดลลานจอดรถ ผลที่ได้จากการทดสอบมีดังนี้

4.2.1 จากการทดลองกับโมเดลลานจอดรถจำลองผลการทำงานของระบบพบว่าเมื่อมีรถเข้าจอดระบบเซนเซอร์ตรวจจับส่งค่าไปประมวลผลที่ไมโครคอนโทรเลอร์ NodeMCU ESP8266 แล้วในส่วนของ Firebase จะแสดงผลมีช่องว่าง เมื่อไมโครคอนโทรเลอร์ NodeMCU ESP8266) ทำการส่งค่าไปยังฐานข้อมูลเป็น true เมื่อมีรถเข้ามาจอดหน้าจอแสดงผลเปลี่ยนเป็นสถานะ ไม่ว่าง แล้ว เมื่อนำรถออกค่าจะเป็น false เป็นสถานะว่าง

4.2.2 จากการทดลองกับโมเดลลานจอดรถจำลองผลการทำงานของระบบพบว่าเมื่อมีรถผ่านประตูระบบเซนเซอร์ตรวจจับส่งค่าไปประมวลผลที่ไมโครคอนโทรเลอร์ NodeMCU ESP8266 แล้วในส่วนของ Firebase จะแสดงผลจำนวนรถที่ใช้บริการในแต่ละวันได้

4.2.3 จากการทดลองกับ Google Assistant กรณีสอบถามสถานะว่างโมเดลลานจอดรถจำลองสามารถ ระบุตำแหน่งที่ใกล้เคียงที่สุดได้

4.3 อุปกรณ์ที่ประกอบสำเร็จ

ภาพที่ 4.7 อุปกรณ์ที่ประกอบสำเร็จ

บทที่ 5
สรุปปัญหาและข้อเสนอแนะ

ในการศึกษาสรุปปัญหาและข้อเสนอแนะ ผู้จัดทำโครงงานได้ศึกษาในหัวข้อต่อไปนี้

5.1 สรุปผลการทดลอง

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

5.3 ข้อเสนอแนะ

5.1 สรุปผลการทดลอง

5.1.1 จากการทดลองกับโมเดลลานจอดรถจำลองผลการทำงานของระบบพบว่าเมื่อมีรถเข้าจอด ระบบเซนเซอร์ตรวจจับส่งค่าไปประมวลผลที่ไมโครคอนโทรเลอร์ NodeMCU ESP8266 แล้วในส่วนของ Firebase จะแสดงผลมีช่องว่าง เมื่อไมโครคอนโทรเลอร์ NodeMCU ESP8266 ทำการส่งค่าไปยังฐานข้อมูลเป็น true เมื่อมีรถเข้ามาจอดหน้าจอแสดงผลเปลี่ยนเป็นสถานะ ไม่ว่าง แล้ว เมื่อนำรถออกค่าจะเป็น false เป็นสถานะว่าง

5.1.2 จากการทดลองกับโมเดลลานจอดรถจำลองผลการทำงานของระบบพบว่าเมื่อมีรถผ่านประตูระบบเซนเซอร์ตรวจจับส่งค่าไปประมวลผลที่ไมโครคอนโทรเลอร์ NodeMCU ESP8266 แล้วในส่วนของ Firebase จะแสดงผลจำนวนรถที่ใช้บริการในแต่ละวันได้

5.1.3 จากการทดลองกับ Google Assistant กรณีสอบถามสถานะว่างโมเดลลานจอดรถจำลองสามารถ ระบุตำแหน่งที่ใกล้เคียงที่สุดได้

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

5.2.1 เซนเซอร์วัดระยะอาจจะทำงานผิดพลาดจากการวัดระยะถ้าหากช่องจอดทางเข้าออกมีขนาดกว้างกว่า 2 ถึง 30 เซนติเมตร

5.2.2 ในกรณีที่มีวัตถุที่มีขนาดเล็กกว่า รถยนต์มาบังอุปกรณ์ปลายทางระบบจะไม่ทราบว่า วัตถุนั้นไม่ใช่รถยนต์เช่น หากมีคนบังอุปกรณ์ปลายทางเป็นระยะเวลานานระบบจะนับคนที่บังนั้นเป็นรถยนต์ที่แล่นผ่าน 1 คัน

5.3 ข้อเสนอแนะ

5.3.1 ศึกษาการออกแบบระบบฐานข้อมูล Firebase และประโยชน์ของ Firebase ให้มากยิ่งขึ้น

5.3.2 ศึกษาการออกแบบระบบ Chatbot และประโยชน์ของ Chatbot ให้มากยิ่งขึ้น

5.3.3 ศึกษาการนำข้อมูลจากอุปกรณ์รับรู้ตรวจวัดระยะ Infrared Sensor มาใช้ประโยชน์โดยวิธีอื่นๆ

บรรณานุกรม

[1] ดิเรกกาฬ ภักดี, นิทัศน์ สุภรัตน์ และ ปรีชา เอี่ยมราคิน. (2014). การออกแบบและพัฒนาแบบจาลองลานจอดรถอัจฉริยะได้ออกแบบและพัฒนาแบบจาลองการจอดรถ

[2] เทพฤทธิ์ บุญยืน, นวพล เปี่ยมปิยชาติ และ อัศนัย ทุนดี. 2014. การออกแบบและสร้างแบบจาลองลานจอดรถอัจฉริยะ

[3] กิตติวร รัตนโชติ, พงษ์ ศักดิ์ จันทร์ไทย และ พิพัฒน์ อินทร์มณี. 2014. ชุดจาลองลานจอดรถแบบอัจฉริยะ

[4] Google Assistant เลขาส่วนตัวอัจฉริยะ สั่งงานมือถือด้วยเสียงผ่าน AI สมองกล. 2018, October 04. Firebase เลขาส่วนตัวอัจฉริยะ สั่งงานมือถือด้วยเสียงผ่าน AI สมองกล (1nd ed.) [Online]. Available: https://www.whatphone.net/application/google-assistant

[5] Petch Kruapanich. 2015, Apri 19. Dialogflow (1 nd ed). [Online]. Available: https://medium.com/linedevth/2cae8214c647

[6] Jedsada Saengow.2015, May 16. Firebase (1 nd ed). [Online]. Available: https://medium.com/jed-ng/d48bfac67b14

[7] THITI YAMSUNG. 2015, October 22. ESP8266 (1 nd ed). [Online]. Available: https://www.thitiblog.com/?s=esp8266

[8] Chai Phonbopit. 2015, July 31. Node.js (1 nd ed). [Online]. Available: https://www.thaiprogrammer.org/2016/02/nodejs

[9] ThaiArduino.Club. 2017, July 24. Arduino (1 nd ed). [Online]. Available: https://thaiarduino.club/what-is-arduino/

[10] DevBun. 2017, March 24. IOT (1 nd ed). [Online]. Available: https://blog.sogoodweb.com/Article/Detail/59554

[11] Anak Mirasing. 2017, Feb 28. Chatbot(1 nd ed). [Online]. Available: https://medium.com/@igroomgrim/3e6165dd34b8

[12] Infrared Sensor (มปป). Infrared Sensor (1 nd ed). [Online]. Available: https://www.gravitechthai.com/product_detail.php?d=1384

[13] Ultrasonic Sensor (มปป). Ultrasonic Sensor (1 nd ed). [Online]. Available: http://cpre.kmutnb.ac.th/esl/learning/index.php?article=hc-sr04-ultrasonic

ผนวก ก

โค้ดที่ใช้ในการพัฒนาระบบลานจอดรถสั่งการด้วยเสียง

โค้ดที่ใช้ในการพัฒนาระบบลานจอดรถสั่งการด้วยเสียง

ตารางที่ ก.1 โค้ดสำหรับส่งสถานะเซ็นเซอร์ช่อง A 1-4 ขึ้น Firebase ในโปรแกรม Arduino

```
1  #include <ArduinoJson.h>
2  #include <Firebase.h>
3  #include <FirebaseArduino.h>
4  #include <ESP8266WiFi.h>
5  #define FIREBASE_HOST "smart-car2-34292.firebaseio.com"
6  #define FIREBASE_AUTH "sLTc2jCLLpjQy66sy91yYpR0EgMzJVdVvCuiEPGi"
7  #define WIFI_SSID "Edex"
8  #define WIFI_PASSWORD "qqqqqqqq"
9  void setup() {
10   Serial.begin(9600);
11   // connect to wifi.
12   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
13   Serial.print("connecting");
14   while (WiFi.status() != WL_CONNECTED) {
15     Serial.print(".");
16     delay(500);
17   }
18   Serial.println();
19   Serial.print("connected: ");
20   Serial.println(WiFi.localIP());
21   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
22   pinMode(D1, INPUT);
23   pinMode(D2, INPUT);
24   pinMode(D3, INPUT);
25   pinMode(D5, INPUT);
```

ตารางที่ ก.1 (ต่อ) โค้ดสำหรับส่งสถานะเซ็นเซอร์ช่อง A 1-4 ขึ้น Firebase ในโปรแกรม Arduino

```
26  }void loop(){
27    int sensor_Value1 = digitalRead(D1);
28    if (sensor_Value1 == HIGH)
29  {
30      Firebase.setString("car/car4/status", "false");
31    } else {
32      Serial.println("Sensor 1 Low");
33      Firebase.setString("car/car4/status", "true");
34    }
35    int sensor_Value2 = digitalRead(D2);
36    if (sensor_Value2 == HIGH)
37    {
38      Firebase.setString("car/car3/status", "false");
39    } else {
40      Firebase.setString("car/car3/status", "true");
41    }
42    int sensor_Value3 = digitalRead(D3);
43    if (sensor_Value3 == HIGH)
44    {
45      Firebase.setString("car/car2/status", "false");
46    } else {
47      Firebase.setString("car/car2/status", "true");
48    }
49    int sensor_Value5 = digitalRead(D5);
50    if (sensor_Value5 == HIGH)
51    {
52      Firebase.setString("car/car1/status", "false");
53    } else {
54      Firebase.setString("car/car1/status", "true"); }  }
```

ตารางที่ ก.2 โค้ดสำหรับส่งสถานะเซ็นเซอร์ช่อง B 5-8 ขึ้น Firebase ในโปรแกรม Arduino

```
1  #include <ArduinoJson.h>
2  #include <Firebase.h>
3  #include <FirebaseArduino.h>
4  #include <ESP8266WiFi.h>
5  // Set these to run example.
6  #define FIREBASE_HOST "smart-car2-34292.firebaseio.com"
7  #define FIREBASE_AUTH "sLTc2jCLLpjQy66sy91yYpR0EgMzJVdVvCuiEPGi"
8  #define WIFI_SSID "Edex"
9  #define WIFI_PASSWORD "qqqqqqqq"
10 void setup() {
11   Serial.begin(9600);
12   // connect to wifi.
13   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
14   Serial.print("connecting");
15   while (WiFi.status() != WL_CONNECTED) {
16     Serial.print(".");
17     delay(500);
18   }
19   Serial.println();
20   Serial.print("connected: ");
21   Serial.println(WiFi.localIP());
22   Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
23   pinMode(D1, INPUT);
24   pinMode(D2, INPUT);
25   pinMode(D3, INPUT);
26   pinMode(D5, INPUT);
27 }
28 void loop(){
```

ตารางที่ ก.2 (ต่อ) โค้ดสำหรับส่งสถานะเซ็นเซอร์ช่อง B 5-8 ขึ้น Firebase ในโปรแกรม Arduino

```
30  int sensor_Value1 = digitalRead(D1);
31    if (sensor_Value1 == HIGH)
32    {
33      Firebase.setString("car/car5/status", "false");
34    } else {
35      Firebase.setString("car/car5/status", "true");}
36    int sensor_Value2 = digitalRead(D2);
37    if (sensor_Value2 == HIGH)
38    {
39      Firebase.setString("car/car6/status", "false");
40    } else  {
41      Firebase.setString("car/car6/status", "true"); }
42    int sensor_Value3 = digitalRead(D3);
43    if (sensor_Value3 == HIGH) {
44      Firebase.setString("car/car7/status", "false");
45    } else {
46      Firebase.setString("car/car7/status", "true"); }
47    int sensor_Value5 = digitalRead(D5);
48    if (sensor_Value5 == HIGH)  {
49      Firebase.setString("car/car8/status", "false");
50    } else {
51      Firebase.setString("car/car8/status", "true");
52    }
53  }
54
```

ตารางที่ ก.3 โค้ดสำหรับส่งค่านับจำนวนผู้ที่มาใช้จานจอดรถในแต่ละวันของ ในโปรแกรม Arduino

```
1  #include <ArduinoJson.h>
2  #include <Firebase.h>
3  #include <FirebaseArduino.h>
4  #include <ESP8266WiFi.h>
5  #include <time.h>
6  #include <stdio.h>
7  #define TRIGGER_PIN  D1
8  #define ECHO_PIN     D2
9  #define FIREBASE_HOST "smart-car2-34292.firebaseio.com"
10 #define FIREBASE_AUTH "sLTc2jCLLpjQy66sy91yYpR0EgMzJVdVvCuiEPGi"
11 #define WIFI_SSID "Edex"
12 #define WIFI_PASSWORD "qqqqqqqq"
13 int sensor_value1;
14 int sensor_value2;
15 int sensor_value3;
16 int time_value1;
17 int timezone = 7 * 3600;
18 int dst = 0;
19 int count=0;
20 void setup() {
21   Serial.begin(9600);
22   // connect to wifi.
23   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
24   Serial.print("connecting");
25   while (WiFi.status() != WL_CONNECTED) {
26     Serial.print(".");
27     delay(500);
28   }
29   Serial.println();
30   Serial.print("connected: ")
```

ตารางที่ ก.3 (ต่อ) โค้ดสำหรับส่งค่านับจำนวนผู้ที่มาใช้จานจอดรถในแต่ละวันของ ในโปรแกรม Arduino

```
31  Serial.println(WiFi.localIP());
32    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
33    pinMode(TRIGGER_PIN, OUTPUT);
34    pinMode(ECHO_PIN, INPUT);
35    pinMode(D3, INPUT);
36    pinMode(D4, INPUT);
37    configTime(timezone, dst, "pool.ntp.org", "time.nist.gov", "time.uni.net.th");
38  //ดึงเวลาจาก Server
39      Serial.println("\nWaiting for time");
40      while (!time(nullptr)) {
41        Serial.print(".");
42        delay(1000);
43      }}
44  String getTime() {
45    WiFiClient client;
46    while (!!!client.connect("google.com", 80)) {
47      Serial.println("connection failed, retrying..."); }
48    client.print("HEAD / HTTP/1.1\r\n\r\n");
49    while(!!!client.available()) {
50       yield();
51    }
52    while(client.available()){
53      if (client.read() == '\n') {
54        if (client.read() == 'D') {
55          if (client.read() == 'a') {
56            if (client.read() == 't') {
57              if (client.read() == 'e') {
58                if (client.read() == ':') {
59                  client.read();
```

ตารางที่ ก.3 (ต่อ) โค้ดสำหรับส่งค่านับจำนวนผู้ที่มาใช้จานจอดรถในแต่ละวันของ ในโปรแกรม Arduino

60	String theDate = client.readStringUntil('\r');
61	client.stop();
62	return theDate;
63	}
64	}
65	}
66	}
67	}
68	}
69	}
70	}
71	
72	String getValue(String data, char separator, int index)
73	{
74	int found = 0;
75	int strIndex[] = {0, -1};
76	int maxIndex = data.length()-1;
77	
78	for(int i=0; i<=maxIndex && found<=index; i++){
79	if(data.charAt(i)==separator || i==maxIndex){
80	found++;
81	strIndex[0] = strIndex[1]+1;
82	strIndex[1] = (i == maxIndex) ? i+1 : i;
83	}
84	}
85	return found>index ? data.substring(strIndex[0], strIndex[1]) : "";
86	}
87	void loop()
88	{ sensor_value1 = digitalRead(D3);

ตารางที่ ก.3 (ต่อ) โค้ดสำหรับส่งค่านับจำนวนผู้ที่มาใช้จานจอดรถในแต่ละวันของ ในโปรแกรม Arduino

```
91    sensor_value2 = digitalRead(D4);
92  //
93    long duration, distance;
94    digitalWrite(TRIGGER_PIN, LOW);  // Added this line
95    delayMicroseconds(2); // Added this line
96    digitalWrite(TRIGGER_PIN, HIGH);
97    delayMicroseconds(10); // Added this line
98    digitalWrite(TRIGGER_PIN, LOW);
99    duration = pulseIn(ECHO_PIN, HIGH);
100   distance = (duration/2) / 29.1;
101   Serial.print(distance);
102   Serial.println(" cm");
103 //
104   String now_time = getTime();
105   String node_time = getValue(now_time,' ',1) + getValue(now_time,' ',2) +
106 getValue(now_time,' ',3);
107   time_value1 = Firebase.getInt("car/date/"+node_time);
108   if (sensor_value1 == LOW || distance <=5)
109   {
110      Serial.println(node_time);
111     time_value1 = time_value1 + 1;
112     Firebase.set("car/date/"+node_time, time_value1);
113   }
114  delay(2000);
115 }
```

ตารางที่ ก.4 โค้ดสำหรับ Chatbot ของ Dialogflow Fulfillment

```
1  // See https://github.com/dialogflow/dialogflow-Fulfillment-nodejs
2  // for Dialogflow Fulfillment library docs, samples, and to report issues
3  'use strict';
4
5  const functions = require('firebase-functions');
6  const {WebhookClient} = require('dialogflow-Fulfillment');
7  const {Card, Suggestion} = require('dialogflow-Fulfillment');
8
9
10
11
12 process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging
13 statements
14
15 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request,
16 response) => {
17   const agent = new WebhookClient({ request, response });
18   console.log('Dialogflow Request headers: ' +
19 JSON.stringify(request.headers));
20   console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
21
22   function welcome(agent) {
23     agent.add(`Welcome to my agent!`);
24   }
25
26   function fallback(agent) {
27     agent.add(`I didn't understand`);
28     agent.add(`I'm sorry, can you try again?`);
29 }
30
```

ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
31  ////////////////// หาระยะที่ใกล้ที่สุดโซน A ////////////////////////////////
32
33    function areaA(agent) {
34     // Get a database reference to our posts
35     var db = admin.database();
36     var ref = db.ref("car");
37
38     // Attach an asynchronous callback to read the data at our posts
39  reference
40     ref.on("value", function(snapshot) {
41       console.log(snapshot.val());
42       var status = [snapshot.val().car1.status.toString(),
43                     snapshot.val().car5.status.toString(),
44                     snapshot.val().car2.status.toString(),
45                     snapshot.val().car6.status.toString(),
46                     snapshot.val().car3.status.toString(),
47                     snapshot.val().car7.status.toString(),
48                     snapshot.val().car4.status.toString(),
49                     snapshot.val().car8.status.toString(),
50                                 ];
51       var i = 0;
52       var count_vb = 0;
53       var strTrue = "false";
54       for(i=0; i<4; i++)
55       {
56         if(status[i].trim() === 'false')
57         {
58           count_vb++;
59         }
         }
```

ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
61  if(status[0].trim() === 'false')
62      {
63        agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 1 ว่างครับผม`);
64      }
65      else if(status[1].trim() === 'false')
66      {
67        agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 5 ว่างครับผม`);
68      }
69      else if(status[2].trim() === 'false')
70      {
71        agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 2 ว่างครับผม`);
72      }
73      else if(status[3].trim() === 'false')
74      {
75        agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 6 ว่างครับผม`);
76      }
77      else if(status[4].trim() === 'false')
78      {
79        agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 3 ว่างครับผม`);
80      }
81      else if(status[5].trim() === 'false')
82      {
83        agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 7 ว่างครับผม`);
84      }
85      else if(status[6].trim() === 'false')
86      {
87        agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 4 ว่างครับผม`);
88      }
89      else if(status[7].trim() === 'false')
```

ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
90    {
91          agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 8 ว่างครับผม`);
92        }
93        else
94        {
95          agent.add(`ขณะนี้ลานจอดรถ A ไม่มีที่ว่างสำหรับจอดรถครับ `);
96        }
97
98      }, function (errorObject) {
99        console.log("The read failed: " + errorObject.code);
100     });
101 }
102   ////////////////////////////////////////////////////////////////////////////
103   ////////////////// หาระยะที่ใกล้ที่สุดโซน B ////////////////////////////////
104    function areaB(agent) {
105     // Get a database reference to our posts
106     var db = admin.database();
107     var ref = db.ref("car");
108
109     // Attach an asynchronous callback to read the data at our posts
110 reference
111     ref.on("value", function(snapshot) {
112       console.log(snapshot.val());
113       var status = [snapshot.val().car8.status.toString(),
114                     snapshot.val().car4.status.toString(),
115                     snapshot.val().car7.status.toString(),
116                     snapshot.val().car3.status.toString(),
117                     snapshot.val().car6.status.toString(),
118                     snapshot.val().car2.status.toString(),
```

ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
119  snapshot.val().car5.status.toString(),
120                       snapshot.val().car1.status.toString(),
121                                 ];
122       var i = 0;
123       var count_vb = 0;
124       var strTrue = "false";
125       for(i=0; i<4; i++)
126       {
127         if(status[i].trim() === 'false')
128         {
129           count_vb++;
130         }
131       }
132
133       if(status[0].trim() === 'false')
134       {
135         agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 8 ว่างครับผม`);
136       }
137       else if(status[1].trim() === 'false')
138       {
139         agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 4 ว่างครับผม`);
140       }
141       else if(status[2].trim() === 'false')
142       {
143         agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 7 ว่างครับผม`);
144       }
145       else if(status[3].trim() === 'false')
146       {
147         agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 3 ว่างครับผม`);
```

ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
149 }
150       else if(status[4].trim() === 'false')
151       {
152         agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 6 ว่างครับผม`);
153       }
154       else if(status[5].trim() === 'false')
155       {
156         agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 2 ว่างครับผม`);
157       }
158       else if(status[6].trim() === 'false')
159       {
160         agent.add(`ท่านสามารถจอดโซน B ได้ครับ ช่อง 5 ว่างครับผม`);
161       }
162       else if(status[7].trim() === 'false')
163       {
164         agent.add(`ท่านสามารถจอดโซน A ได้ครับ ช่อง 1 ว่างครับผม`);
165       }
166       else
167       {
168         agent.add(`ขณะนี้ลานจอดรถ B ไม่มีที่ว่างสำหรับจอดรถครับ `);
169       }
170
171     }, function (errorObject) {
172       console.log("The read failed: " + errorObject.code);
173     });
174 }
175
176 ////////////////////////////////////////////////////////////////////////////////
    ////////////
```

ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
177 function areaall(agent) {
178     // Get a database reference to our posts
179     var db = admin.database();
180     var ref = db.ref("car");
181 
182    // Attach an asynchronous callback to read the data at our posts
183 reference
184     ref.on("value", function(snapshot) {
185       console.log(snapshot.val());
186       var status = [snapshot.val().car1.status.toString(),
187                     snapshot.val().car2.status.toString(),
188                     snapshot.val().car3.status.toString(),
189                     snapshot.val().car4.status.toString(),
190                     snapshot.val().car5.status.toString(),
191                     snapshot.val().car6.status.toString(),
192                     snapshot.val().car7.status.toString(),
193                     snapshot.val().car8.status.toString(),
194                                 ];
195       var i = 0;
196       var count_all = 0;
197       var strTrue = "false";
198       for(i=0; i<8; i++)
199       {
200         if(status[i].trim() === 'false')
201         {
202           count_all++;
203         }
204       }
205       if(count_all > 0)
206       {
```

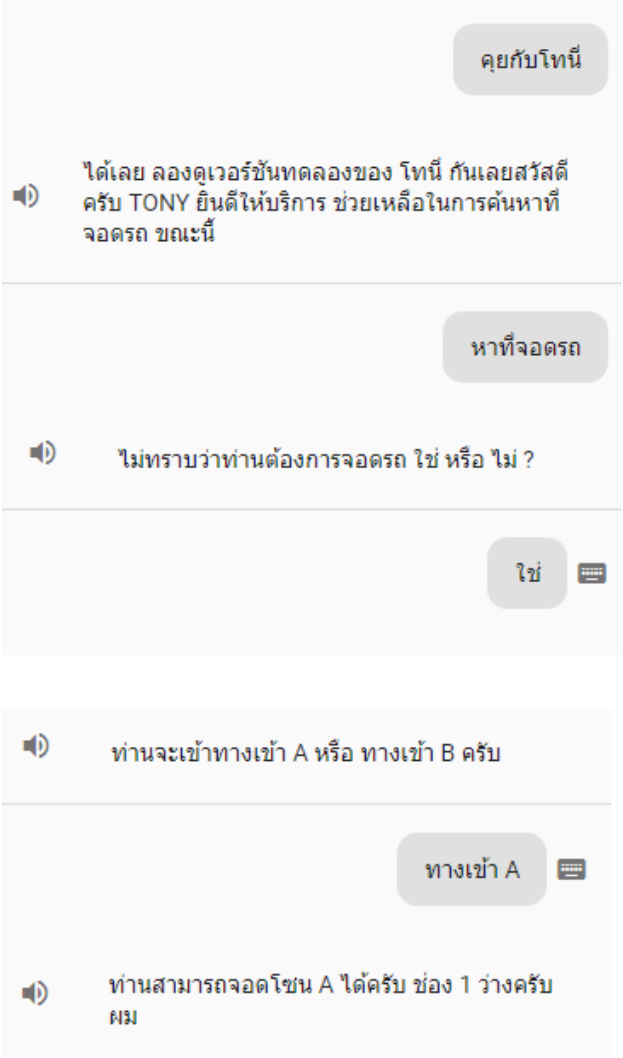
ตารางที่ ก.4 (ต่อ) โค้ดสำหรับ Chatbotของ Dialogflow Fulfillment

```
207 agent.add(`ขณะนี้ลานจอดรถมีช่องว่างทั้งหมด ` + count_all.toString() + ` ที่ครับผม
208 ท่านต้องการจอดรถ ใช่ หรือ ไม่ `);
209
210       }
211       else
212       {
213         agent.add(`ขณะนี้ลานจอดรถไม่มีที่ว่างสำหรับจอดรถครับ`);
214       }
215
216     }, function (errorObject) {
217       console.log("The read failed: " + errorObject.code);
218     });
219 }
220
221   let intentMap = new Map();
222   intentMap.set('Default Welcome Intent', welcome);
223   intentMap.set('Default Fallback Intent', fallback);
224   intentMap.set('yes - A', areaA);
225   intentMap.set('yes - B', areaB);
226   intentMap.set('area', areaall);
227   intentMap.set('areaA', areaA);
228   intentMap.set('areaB', areaB);
229     agent.handleRequest(intentMap);
230 });
231 //ตัวเชื่อมต่อ Add the Firebase Admin SDK to Your Server
232 const admin = require("firebase-admin");
233 admin.initializeApp({
234   credential: admin.credential.applicationDefault(),
235   databaseURL: 'https://smart-car2-34292.firebaseio.com'});
236
```

ผนวก ข

คู่มือในการใช้งานระบบลานจอดรถสั่งการด้วยเสียง

คู่มือการใช้งาน
Chatbotลานจอดรถอัจฉริยะ



ขั้นตอนที่ 1 ท่านเปิดใช้งานโดยพูดว่า “ คุยกับโทนี่ ”

ขั้นตอนที่ 2 ท่านต้องการหาที่จอดรถโดยพูดว่า “ หาที่จอดรถ ”

ขั้นตอนที่ 3 ระบบจะสอบถามว่าท่านต้องการจอดรถใช่ หรือ ไม่

ขั้นตอนที่ 4 ระบบจะสอบถามว่าท่านเข้าจอดรถทางประตู A หรือประตู B หากตอบ A ระบบจะเลือกช่องว่างที่ใกล้ที่สุดให้ท่าน

ภาพที่ ข.1 คู่มือการใช้งาน Chatbot ลานจอดรถอัจฉริยะ

ประวัติผู้จัดทำ

ชื่อ-นามสกุล	นายบุญเกิด ธีรศรัณยานนท์
วัน/เดือน/ปีเกิด	12 กันยายน พ.ศ. 2539
สถานที่เกิด	ชลบุรี
ที่อยู่	1255/4 หมู่ 4 ซอย นครทอง อำเภอ เมือง จังหวัด สมุทรปราการ 10270
ประวัติการศึกษา	ระดับมัธยมศึกษาตอนปลาย
	โรงเรียนมัธยมด่านสำโรง จังหวัดสมุทรปราการ ปีการศึกษา 2557

ประวัติผู้จัดทำ

ชื่อ-นามสกุล	นายวินัย บุญทา
วัน/เดือน/ปีเกิด	10 พฤษภาคม พ.ศ. 2537
สถานที่เกิด	กาญจนบุรี
ที่อยู่	6/157 ซอย เลี่ยงเมืองนนทบุรี ตำบล สวนใหญ่ อำเภอ เมืองนนทบุรี จังหวัด นนทบุรี
ประวัติการศึกษา	ระดับมัธยมศึกษาตอนปลาย
	โรงเรียนรัตนาธิเบศร์ ปีการศึกษา 2557

ประวัติผู้จัดทำ

ชื่อ-นามสกุล	นายอินทัช เวนานนท์
วัน/เดือน/ปีเกิด	2 ตุลาคม พ.ศ. 2539
สถานที่เกิด	ลพบุรี
ที่อยู่	92/92 หมู่ 4 ตำบล เขาสามยอด ถนน พหลโยธิน อำเภอ เมือง จังหวัด ลพบุรี 15000
ประวัติการศึกษา	ระดับมัธยมศึกษาตอนปลาย โรงเรียนพิบูลวิทยาลัย ลพบุรี ปีการศึกษา 2557