

โครงการเรื่อง  
Smart Parking

โดย

นาย ทรงพล จงเกริกเกียรติ รหัสนิสิต 6031016921

นาย ภูมิ เลิศฤทธิมหาชัย รหัสนิสิต 6030462921

นาย ภาณุพงศ์ บุญหลง รหัสนิสิต 6030447521

นาย ศิวัช พงษ์พานิช รหัสนิสิต 6030559121

เสนอต่อ

อ.ดร. พิษณุ สิทธิอมร

วิชา ห้องปฏิบัติการระบบฝังตัว

รหัสวิชา 2110366 ภาคเรียนที่ 1/2562

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

จุฬาลงกรณ์มหาวิทยาลัย

## Smart Parking

จัดทำโดย

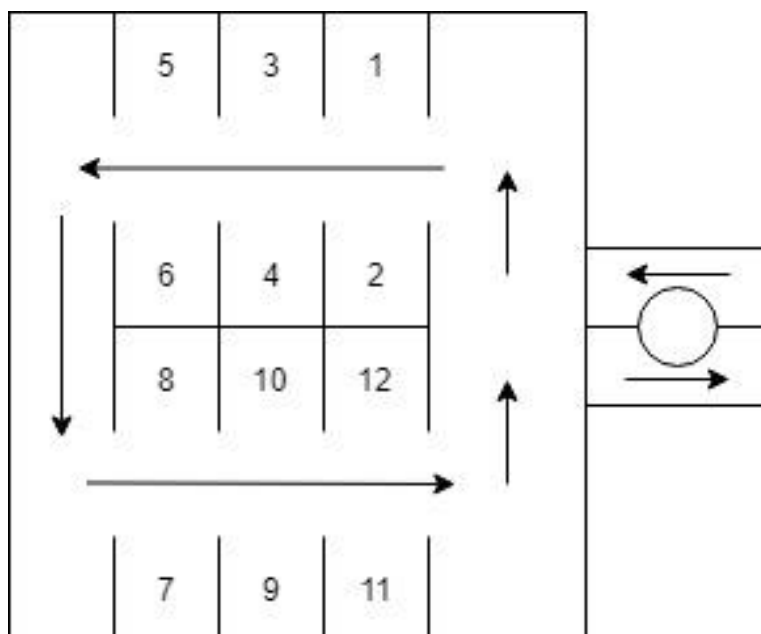
- |                            |            |
|----------------------------|------------|
| 1. นายทรงพล จงเกริกเกียรติ | 6031016921 |
| 2. นายภูมิ เลิศฤทธิมหาชัย  | 6030462921 |
| 3. นายภาณุพงศ์ บุญหลง      | 6030447521 |
| 4. นายศิวัช พงษ์พานิช      | 6030559121 |

Github Repository : [https://github.com/Siwat-Keng/Embedded\\_Project.git](https://github.com/Siwat-Keng/Embedded_Project.git)

### System Architecture (รับผิดชอบโดย นาย ภูมิ เลิศฤทธิมหาชัย)

ทำการออกแบบ Algorithm และรูปแบบในการทำงานของ Smart Parking โดยทำให้ผู้ขับขี่สามารถเข้าถึงที่จอดรถที่ใกล้ที่สุดได้ โดยกำหนดพฤติกรรมของผู้ใช้งานคือ

1. เมื่อผู้ขับขี่เข้ามาจะต้องทำการแลกบัตรกับพนักงานที่ป้อมยาม
2. ผู้ขับขี่จะได้รับบัตรหมายเลขของที่จอดรถที่ผู้ขับต้องการเข้าใช้งาน
3. ผู้ขับขี่เข้าช่องจอดรถตามที่ระบุไว้
4. เมื่อผู้ขับขี่ใช้งานเสร็จ ทำการวนรถออกจากช่องจอดรถและนำบัตรไปคืนที่ป้อมยาม
5. เจ้าหน้าที่บริเวณป้อมยามเก็บบัตรและคิดค่าบริการที่จอดรถตามเกณฑ์



แผนภาพตัวอย่างลานจอดรถ

ในช่องจอดรถแต่ละช่องจะมีสถานะของตนเองซึ่งแสดงด้วยสีผ่านโปรแกรม ดังนี้

1. สีเขียว - พร้อมใช้งาน
2. สีเหลือง - มีรถกำลังต้องการใช้/กำลังออก
3. สีแดง - มีรถจอดอยู่
4. สีดำ - ไม่อนุญาตให้ใช้ได้

#### การทำงานและหลักการทำงานของโปรแกรม

1. เมื่อเริ่มต้นโปรแกรมมา ช่องจอดรถทุกช่องจะอยู่ในสถานะพร้อมใช้งาน(สีเขียว)
2. เมื่อมีผู้ขับที่ต้องการเข้ามาใช้บริการ โปรแกรมจะทำการแสดงหมายเลขช่องให้ทราบว่าให้ไปจอดช่องใด จากนั้นโปรแกรมก็จะเปลี่ยนช่องจอดรถช่องนั้นให้เป็นสถานะมีรถกำลังต้องการใช้(สีเหลือง)
3. เมื่อผู้ขับเข้าจอดในช่อง สถานะของช่องจอดรถในโปรแกรมนั้นจะเปลี่ยนเป็นมีรถจอดอยู่(สีแดง) และทำการจับเวลาสำหรับคิดค่าบริการค่าจอด
4. เมื่อผู้ขับที่ต้องการออกจากที่จอดรถ เมื่อผู้ขับไม่อยู่ในช่องนั้นแล้ว สถานะจะเป็นกำลังออก(สีเหลือง) และเวลาสำหรับคิดค่าบริการค่าจอดจะหยุดลง
5. เมื่อผู้ขับเข้ามาถึงป้อมยามและต้องการออก เจ้าหน้าที่ป้อมยามจะทำการยกเลิกการใช้งานช่องจอดรถนั้น และแสดงค่าบริการที่จอดรถให้ผู้ขับชำระเงิน และช่องนั้นจะกลับสู่สถานะพร้อมใช้งาน(สีเขียว)

#### Algorithm ในการคำนวณหาช่องจอดรถที่ใกล้ที่สุด

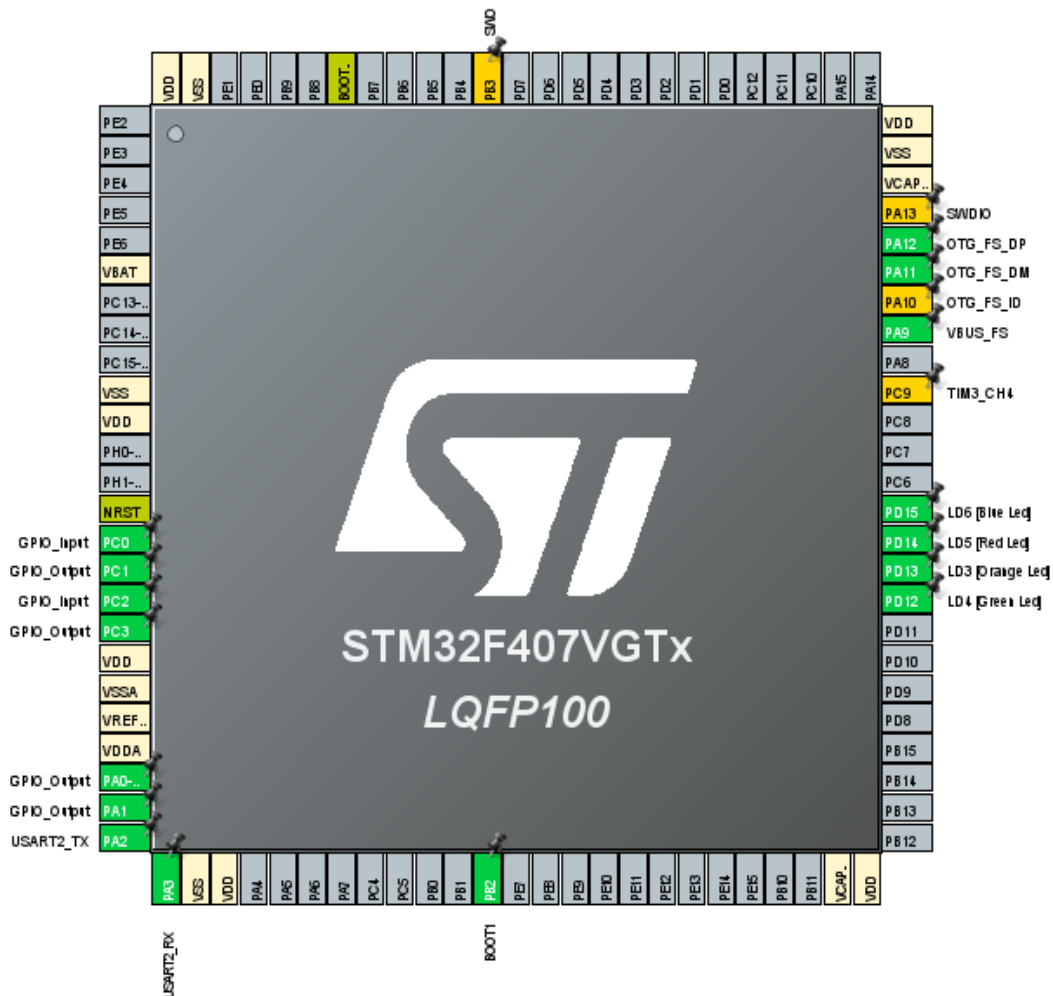
เนื่องจากทางผู้พัฒนาได้ทำที่จอดรถที่มีเพียงชั้นเดียวและมีจำนวนช่องหมายเลข 1-12 (ในอนาคตอาจมีเพิ่มขึ้นได้) โดยมีการกำหนดหมายเลขให้หมายเลขน้อยแทนช่องที่พบก่อนเมื่อขับรถตามเส้นทางที่บังคับในที่จอดรถ จึงได้ใช้ Greedy Algorithm โดยเลือกที่จอดรถที่ว่างอยู่ที่อยู่ใกล้ป้อมยามที่สุด (ซึ่งกำหนดไว้ด้วยเลขที่จอด ยิ่งเลขน้อยยิ่งใกล้ป้อมยาม)

#### กรณีผิดปกติต่างๆและแนวทางการรับมือ

เนื่องจากมีความเป็นไปได้ที่ผู้ขับที่สามารถเกิดความเข้าใจผิดหรือไม่เคารพกติกาการใช้ที่จอดรถทีมงานจึงได้ยกตัวอย่างและแนวทางการแก้ไขดังนี้

**กรณีทั่วไป** ผู้ขับรถไม่เข้าจอดในที่จอดรถที่ถูกจอง(สีเหลือง)

- ระบบจะส่งเสียงเตือนให้ยารู้



## STM32F4 DISCOVERY

ใช้ GPIO\_Output เพื่อ trigger Ultrasonic sensor ทั้งหมด 2 pins ได้แก่ PC1 และ PC3 ใช้ GPIO\_Input เพื่อรับ echo จาก Ultrasonic sensor อีกจำนวน 2 pins ได้แก่ PC0 และ PC2 ใช้ PA0 และ PA1 เป็น GPIO\_Output เพื่อทำการแสดงสถานะของที่จอดรถนั้น ๆ มี Timer ที่ PC9 ซึ่งเป็น Interrupt timer เพื่อ Update สถานะของที่จอดรถ และใช้ UART ในการติดต่อกับ ESP8266 โดยมี TXD เป็น PA2 และมี RXD เป็น PA3 ซึ่งรับข้อมูลแบบ Interrupt

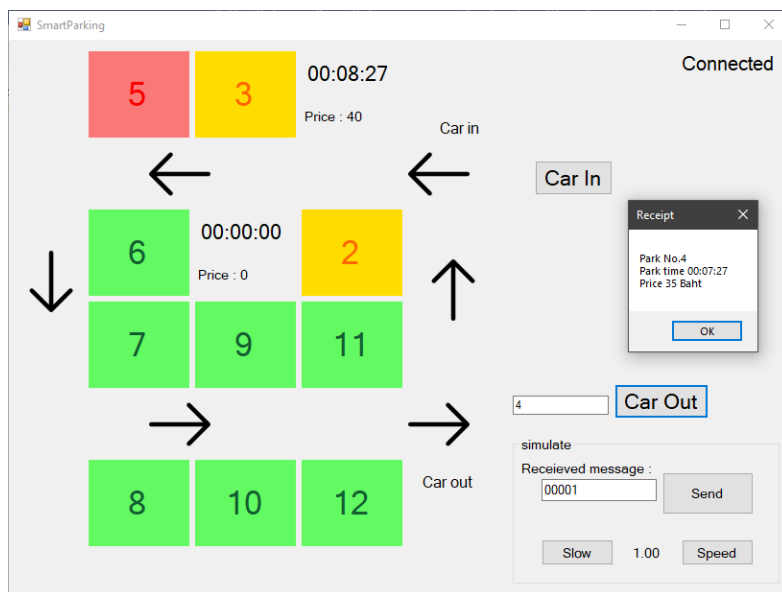
## ESP8266

มีการใช้ Library SoftwareSerial.h เพื่อทำการส่งข้อมูลระหว่าง STM32F4 DISCOVERY และ ESP8266 โดยมี D6 เป็น RX และ D7 เป็น TX หลังจากนั้นทำการรับ/ส่งข้อมูลให้กับ Netpie โดยใช้ Library MicroGear.h

## UI Designer and Developer (รับผิดชอบโดย นายภาณุพงศ์ บุญหลง)

ในส่วนของ UI ได้มีการออกแบบให้มีความสัมพันธ์กับสถาปัตยกรรมของระบบที่ได้วางโครงไว้สำหรับเจ้าหน้าที่ควบคุม ให้สามารถใช้งานได้อย่างสะดวก โดยสิ่งที่เราให้ความสำคัญมากที่สุดในการออกแบบคือความชัดเจนและการตอบสนองของระบบเกิดขึ้นทันทีหลังจากเกิดแอคชั่นจากเจ้าหน้าที่ควบคุมและผู้ขับรถ ซึ่งสาเหตุที่เราเลือก 2 ข้อนี้ยึดเป็นหลักในการออกแบบ เพราะทำให้ระบบของเรามีความถูกต้องและแม่นยำมากที่สุดเท่าที่สามารถจะเป็นไปได้ ระบบของเราเป็นระบบจัดการที่จอดรถ ให้กับผู้ใช้รถ เพื่อให้หาที่จอดรถที่ว่างอยู่ที่ใกล้ที่สุด จึงมีความจำเป็นที่ระบบจะต้องได้ตอบกับ เจ้าหน้าที่ควบคุมได้ตามที่ต้องการ

### พิจารณา UI



นี่เป็นตัวอย่างต้นแบบของ Interface ที่เจ้าหน้าที่ควบคุมจะได้ใช้

- เมื่อมีผู้ขับเข้ามาใช้ที่จอดรถ เจ้าหน้าที่ควบคุมจะกดปุ่ม Car In ระบบจะแสดงหมายเลขที่จอดรถที่ใกล้ที่สุดและสีของที่จอดรถนั้นจะเปลี่ยนจากสีเขียวเป็นสีแดง
- หากผู้ขับเข้าไปจอดในที่จอดรถที่เป็นสีเขียว(ที่จอดรถที่ไม่ได้ถูกจองโดยระบบ) ระบบจะส่งเสียงเตือนออกมา
- เมื่อผู้ขับเข้าไปจอดในที่จอดรถที่ถูกจอง(ที่จอดรถที่เป็นสีแดงซึ่งถูกจองโดยระบบ) สีของที่จอดรถนั้นจะเปลี่ยนจากสีแดงเป็นสีแดงแล้วจะเริ่มจับเวลา
- เมื่อผู้ขับขับรถออกมาจากที่จอดนั้น สีของที่จอดรถจะเปลี่ยนจากสีแดงเป็นสีเขียว แล้วตัวจับเวลาจะหยุด

- เมื่อผู้ขับที่ต้องการออกจากลานจอดรถ เจ้าหน้าที่ควบคุมจะพิมพ์เลขที่จอดรถลงใน Textbox แล้วจึงกดปุ่ม Car Out ระบบจะแสดงหมายเลขที่จอดรถ เวลาที่จอดรถ และค่าจอดรถ ที่จอดรถนั้นจะเปลี่ยนจากสีเหลืองเป็นสีเขียวและตัวจับเวลาจะถูกรีเซ็ต
- เมื่อเจ้าหน้าที่ควบคุมคลิกบนที่จอดรถสักที่หนึ่งจะแสดงเวลาและค่าจอดรถของที่จอดรถนั้น เมื่อกดบนที่ว่างบน Interface หมายเลขที่จอดรถจะกลับมามีค่าเดิม

เนื่องจากเซนเซอร์มีจำนวนจำกัดจึงสร้างส่วนของ simulate ใช้ในการจำลองสถานการณ์จอดรถ

- ข้อความที่ใส่ลงใน Textbox นั้นจำลองข้อความที่โปรแกรมรับจาก NETPIE ทำงานโดย โปรแกรมจะส่งข้อความไปที่ NETPIE และ NETPIE จะส่งข้อความกลับมาที่โปรแกรมอีกครั้ง
- ปุ่ม Speed, Slow ใช้ในการลด,เพิ่ม Interval ของ Timer ตามลำดับ

- ตัวอย่างสถานการณ์

ที่จอดรถหมายเลข 1 : ว่าง(สีเขียว)

ที่จอดรถหมายเลข 2 : ถูกจอง(สีเหลือง)

ที่จอดรถหมายเลข 3 : ไม่ว่าง(สีแดง)

ที่จอดรถหมายเลข 4 : ถูกจอง(สีเหลือง)

ที่จอดรถหมายเลข 5 : ไม่ว่าง(สีแดง)

ที่จอดรถหมายเลข 6-12 : ว่าง(สีเขียว)

ตัวอักษรที่ใช้ในข้อความคือ 1 : มีรถไปจอด

0 : ไม่มีรถไปจอดหรือมีรถขับออกจากที่จอดนั้น

ตัวอย่างข้อความ (ไม่จำเป็นต้องส่งข้อความของทั้ง 12 ช่อง)

- 100110

- (1) ระบบจะส่งเสียงเตือนเพราะที่จอดรถหมายเลข 1 ไม่ได้ถูกจอง โดยระบบ
- (0) ที่จอดรถหมายเลข 2 ยังคงเป็นสีเหลืองเพราะไม่มีรถไปจอด
- (0) ที่จอดรถหมายเลข 3 เปลี่ยนจากสีแดงเป็นสีเหลืองเพราะรถขับออกจากที่จอดรถ แล้วตัวจับเวลาจะหยุด
- (1) ที่จอดรถหมายเลข 4 เปลี่ยนจากสีเหลืองเป็นสีแดงเพราะมีรถขับไปจอด และที่จอดรถนั้นจะเริ่มจับเวลา
- (1) ที่จอดรถหมายเลข 5 ไม่มีการเปลี่ยนแปลง
- (0) ที่จอดรถหมายเลข 6 ไม่มีการเปลี่ยนแปลง

## Team Management (รับผิดชอบโดย นายทรงพล จงกริกเกียรติ)

เนื่องจาก Project นี้ต้องใช้เวลาในการจัดการและบริหารการทำงานของลูกทีม  
ข้าพเจ้าจึงแบ่งงานออกเป็นช่วงต่างๆ ดังนี้

1. เริ่มต้นด้วยการคิดว่าการทำอะไรที่สามารถตอบโจทย์ของการแก้ไขปัญหาต่างๆ  
ซึ่งทีมงานเล็งเห็นว่าการจอตลอดเป็นปัญหาหนึ่งที่เรากำลังเจอจากแล้วจะต้องเสียเวลาในการค้นหาที่จอต  
เป็นอย่างมาก ไม่ว่าจะเป็นการหาที่จอตที่ห่าง หรือที่สาธารณะต่างๆก็ตามแต่  
หากเราสามารถที่จะทำการระบุได้ว่ารถแต่ละคันควรเข้าจอตบริเวณไหนก็จะทำให้สามารถประหยัดเว  
ลาได้เป็นอย่างมากทีเดียว (11/11/2019)
2. เมื่อทุกคนเห็นแนวทางของ Project แล้วจะต้องไปในทิศทางใด  
จึงได้มีการแบ่งหน้าที่ของแต่ละคนออกตามความถนัด  
แล้วแยกย้ายกันไปศึกษาข้อมูลที่เกี่ยวข้องกับ Project ในครั้งนี้  
โดยที่ใครมีหน้าที่ในส่วนไหนก็ให้ไปศึกษาในส่วนนั้นเป็นหลัก  
แต่ก็ให้ทุกคนทำการศึกษาในส่วนของคนอื่นมาด้วย แต่ไม่ต้องลงลึกมากนัก  
การทำแบบนี้จะทำให้เวลาที่มาทำงานร่วมกันจริงๆจะสามารถคุยและปรึกษาหารือได้อย่างเข้าใจ  
แล้วงานจะได้ดำเนินไปอย่างสะดวกและรวดเร็ว(11/11/2019)
3. เมื่อผ่านไป 1 สัปดาห์ แต่ละคนก็นำสิ่งที่ตนเองไปค้นคว้ามาคุยกัน  
เพื่อจะได้ทราบถึงปัญหาและแนวทางต่างๆในการพัฒนา  
พร้อมออกแบบลักษณะและอุปกรณ์ที่จะต้องนำมาใช้ใน Project ครั้งนี้ เช่น Infrared sensor,  
สายไฟ, Breadboard ฯลฯ  
และส่งรายชื่ออุปกรณ์ต่างๆให้อาจารย์ตรวจสอบอีกทีว่าอุปกรณ์ที่ต้องการจะใช้มีความถูกต้องและเห  
มาะสมกับโปรเจกต์ที่ต้องการจะทำหรือไม่ (18/11/2019)
4. นัดคุยกันครั้งสุดท้ายก่อนเข้าสัปดาห์สอบ โดยทำความเข้าใจภาพรวมของ Project  
อีกครั้งหนึ่งและรับอุปกรณ์ต่างๆ จากอาจารย์และกลับมารวมตัวกันอีกครั้งหลังสอบเสร็จ  
(25/11/2019)

5. เริ่มลงมือทำงานครั้งแรกหลังจากสอบเสร็จ โดยแบ่งหน้าที่การทำงานออกเป็น 2 ส่วนหลักๆคือส่วนของ hardware และส่วนของ software ในส่วนของ hardware เมื่อได้ลองลงมือทำไปสักระยะหนึ่งแล้วพบว่าตัวเซนเซอร์อินฟาเรดใช้งานค่อนข้างยาก ทางกลุ่มจึงได้มีการประชุมหารือและสรุปได้ว่าจะเปลี่ยนมาใช้เซนเซอร์อัลตราโซนิกแทน ในส่วนของด้านซอฟต์แวร์นั้นก็ต้องแบ่งเป็น 2 ส่วนอีกคือส่วนของ user interface และส่วนของการส่งรับข้อมูล โดยในส่วนของการรับส่งข้อมูลนี้ก็จะทำควบคู่ไปกับส่วนของ hardware เพราะเป็นพาร์ทที่มีความเกี่ยวข้องกัน
6. ในช่วงสัปดาห์สุดท้ายก่อนที่จะถึงเดดไลน์จะเป็นช่วงที่ทุกคนจะนำงานในส่วนของตัวเองมาปรึกษากับกลุ่มอีกทีว่ามีข้อแก้ไขหรือไม่ เมื่อตรวจสอบความเรียบร้อยในงานของแต่ละคนและทำให้แน่ใจว่าทุกคนเข้าใจตรงกันแล้ว ก็ได้เริ่มกระบวนการ Integrated หรือการรวมงานทั้งหมดเข้าเป็นงาน Project ขึ้นเดียวให้ได้ ในขั้นตอนนี้จะมีการปรับแก้ในส่วนๆต่างๆไม่มากก็น้อยเพื่อให้รูปงานออกมาสำเร็จและทำงานได้ตาม requirements ที่วางไว้
7. หลังจากทำงาน Project เสร็จแล้วก็ถึงขั้นตอนทำเอกสารซึ่งได้ขึ้นโครงไว้ก่อนหน้านี้แล้ว ซึ่งขั้นตอนนี้ถือเป็นขั้นตอนสุดท้ายของงานนี้