

# PARKSELF

## (ระบบจอดรถอัตโนมัติ)

### ที่มาความสำคัญ

การหาที่จอดรถในอาคารหรือลานจอดรถเป็นสิ่งที่ทำได้ยาก เนื่องจากเราไม่ทราบว่าที่จอดรถช่องไหนว่างบ้าง เราจึงต้องวนไปเรื่อยๆ เพื่อหาที่จอดรถ ซึ่งทำให้เปลืองน้ำมัน และยังเสียเวลาอีกด้วย เราจึงคิดค้นระบบอัตโนมัติในการจอดรถ เพื่อลดอัตราการสูญเสียพลังงานซึ่งรวมไปถึงค่าน้ำมัน และเวลาของผู้ขับขี่ในการหาที่จอดรถ อีกทั้งยังทำให้จอดรถได้เป็นระเบียบและประหยัดพื้นที่มากขึ้นอีกด้วย

จากการที่ได้ศึกษาอาคารจอดรถอัตโนมัติหลายๆที่ ทำให้เราได้ทราบว่าถ้าต้องการระบบจอดรถอัตโนมัติจะต้องมีการติดตั้งที่เยอะมาก แทบจะต้องสร้างอาคารใหม่ เราจึงคิดโปรเจกต์ parkself ขึ้นมา เพื่อให้ผู้ใช้สามารถมีระบบจอดรถอัตโนมัติในที่จอดรถเดิมของตนเองได้ ไม่ต้องสร้างอาคารใหม่ เพียงแค่ซื้อหุ่นยนต์และ pallet ของเราไปติดตั้ง โดยหุ่นยนต์ของเรานั้นสามารถทำงานได้ในทุกแบบของอาคาร เพราะใช้อัลกอริทึม SLAM ในการ Mapping และ Localization

### SLAM (Simultaneous localization and mapping)

คือ อัลกอริทึมที่หุ่นยนต์ใช้ในการสร้าง Map ของสภาพแวดล้อมรอบๆตัว และ Localization หรือบอกตำแหน่งของตัวเองใน map ได้ในเวลาเดียวกัน

### ระบบจอดรถอัตโนมัติที่มีอยู่แล้ว

- ตัวอย่างในประเทศไทย

- ☐ อาคารจอดรถอัตโนมัติ ท่าเทียบเรือแหลมบาลีฮาย พัทยาใต้





รูปภายในอาคารจอดรถอัตโนมัติ ทำเทียบเรือแหลมบาลีฮาย

Ref : <https://news.thaiza.com/local/349748/>

อาคารจอดรถนี้เป็นอาคารจอดรถอัตโนมัติที่ใหญ่ที่สุดในประเทศไทย สามารถรองรับรถยนต์ได้ถึง 417 คัน เปิดให้บริการ 24 ชม. จัดทำเพื่อให้บริการประชาชนและนักท่องเที่ยวซึ่งส่วนใหญ่ข้ามฝั่งไปยัง เกาะล้าน แก้ปัญหาที่จอดรถไม่เพียงพอเนื่องจากที่ดินที่พัทยานีมีมูลค่าสูง

ตัวอาคารมีความสูง 7 ชั้น ทำงานด้วยระบบอัตโนมัติ เมื่อเข้ามาใช้งานเพียงกดปุ่มรับบัตรและนำรถเข้าช่องตัวอาคารซึ่ง มีทั้งหมด 6 ประตู ระบบจะทำการเก็บรถเข้าจอดอัตโนมัติแบบไร้คนขับ ทำงานด้วยตัวหุ่นยนต์ 3 ตัว คอยรับรถจากผู้มาใช้บริการ พร้อมกับระบบป้องกันนิรภัยอย่างดี โดยใช้เวลาในการเข้าจอด 2 นาที/คัน

ข้อยกเว้น สำหรับรถยนต์ที่ติดแก๊ส รถยนต์ที่โหลดเตี้ย และรถที่มีขนาดใหญ่(เช่น รถตู้) ไม่สามารถเข้าใช้บริการได้

Ref : <https://www.youtube.com/watch?v=D4KaPAYgPfE>

อาคารจอดรถระบบอัตโนมัติทำเทียบเรือแหลมบาลีฮาย เป็นระบบอัตโนมัติแบบสามารถเคลื่อนที่ได้ 2 แกน และมีตัว Pallet ที่สามารถวิ่งเข้าที่จอดรถได้



Car size	Length	Max 5,250 mm
	Width	Max 2,150 mm
	Height	Max 1,550 / 1,950 mm
	Weight	Max 2,200 kg
Electricity Power	AC 380, 50/60Hz, 3 Phase	
Availability	SEDAN, SUV, RV	

รูปแบบอาคารจอดรถ TD System ของ บริษัท พัทธนันท์ ปาร์คกิ้ง จำกัด ซึ่งเป็นบริษัทที่ทำอาคารจอดรถระบบอัตโนมัติทำ  
เทียบเรือแหลมบาลีฮาย

Ref : [http://pattananparking.com/robotic\\_system.php](http://pattananparking.com/robotic_system.php)

-ตัวอย่างในต่างประเทศ

☐ CarTower Discovery (TurmFahrt) ที่ประเทศเยอรมัน



รูปภายนอกอาคารจอร์ด CarTower Discovery (TurmFahrt)

Ref : [http://www.imgrum.org/media/1585597641452978396\\_32672803](http://www.imgrum.org/media/1585597641452978396_32672803)



รูปภายในอาคารCarTower Discovery (TurmFahrt)

อาคารจอร์ด TurmFahrt สูง 48 เมตร ที่ชั้น 20 มีจุดชมวิวเพื่อให้นักท่องเที่ยวชมวิวดูได้ เนื่องจากอาคารนี้เป็นที่เก็บรถก่อนนำรถไปส่งให้ผู้ลูกค้าของยี่ห้อ Volkswagen มีทั้งหมด 2 อาคาร 1 อาคาร เก็บรถได้



400 คัน และเป็นระบบจอดรถอัตโนมัติที่เร็วที่สุดในโลกในหมวดลิฟท์พิเศษด้วยความเร็วในการบันทึก 2 เมตร/วินาที รถยนต์ที่นำมาจอดที่นี่จะใช้เวลาในการจอด 1 นาที 44 วินาที จากทางเข้าของอาคารไปถึงที่จอดรถสูงสุด จึงเป็นระบบจอดรถอัตโนมัติที่เร็วที่สุดในโลก

TurmFahrt เป็นระบบอัตโนมัติแบบTOR Park system สามารถเคลื่อนที่ได้ 1 แกนในการเคลื่อนที่แกน Z สามารถหมุนได้รอบแกน Z และมีตัว Pallet ที่สามารถวิ่งเข้าที่จอดรถได้

Ref : <https://www.autostadt.de/en/explore-the-autostadt/car-towers>

<https://www.youtube.com/watch?v=2M213o0XRho>



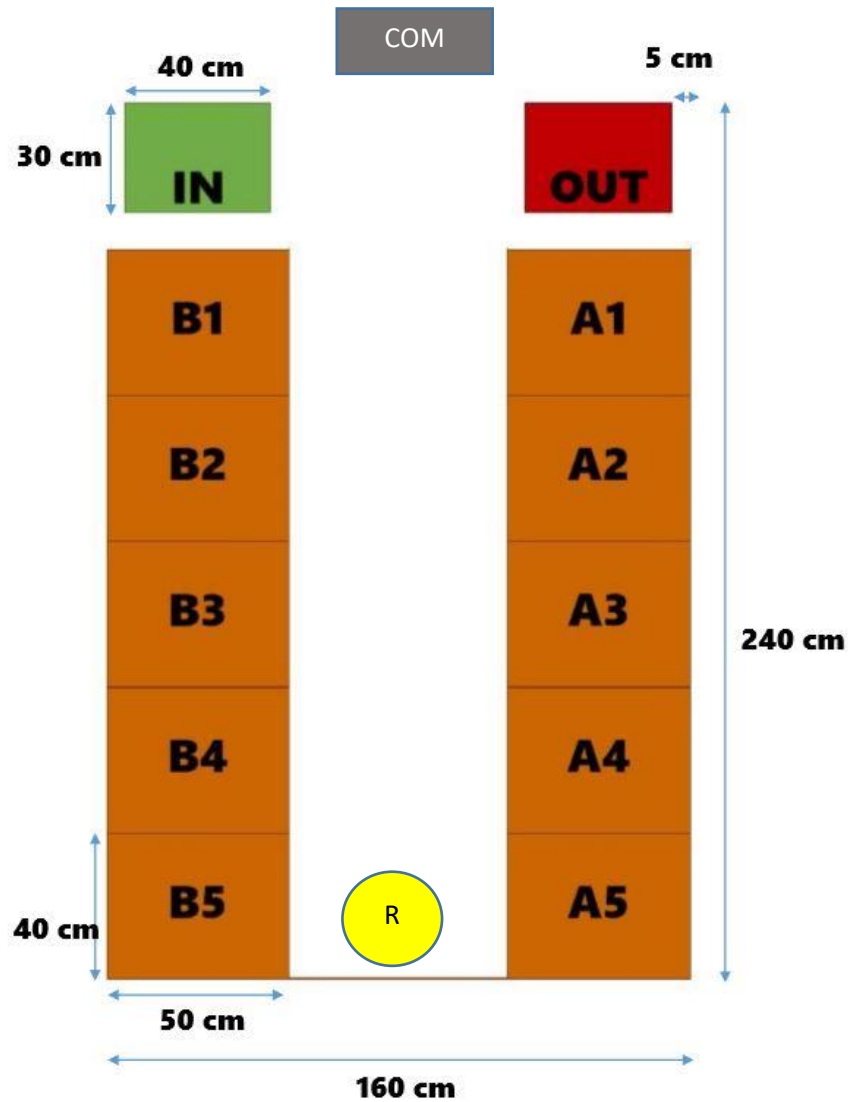
รูปการชมวิผ่านอาคารจอดรถ TurmFahrt

Ref : [http://www.tagungsplaner.de/entertainment\\_packages/1281](http://www.tagungsplaner.de/entertainment_packages/1281)

## ภาพรวมของระบบ



ในการออกแบบหุ่นยนต์จรวดอัตโนมัติจะใช้ในการทำแบบจำลองขึ้นมา ซึ่งจะใช้แบบรถบังคับจำลองขนาด 1:10 ของขนาดจริง เป็นลานจอดรถสำหรับรถ 10 คัน และจะจำลองระบบที่ใช้ในมหาวิทยาลัยพระจอมเกล้าธนบุรี(ผู้ใช้ทุกคนมีรหัสนักศึกษา)ก่อนเท่านั้น ซึ่งระบบจะมีการทำงานดังนี้



### ขั้นตอนการใช้งาน




รูป map ของที่จอดรถจำลอง


### การนำรถเข้าจอด

1. ผู้ใช้นำรถมาจอดบน pallet ซึ่งเตรียมไว้ที่โซน  IN
2. ผู้ใช้ลงจากรถ ไปยังคอมพิวเตอร์  COM จากนั้นจะเจอกับ GUI ซึ่งจะให้กรอกรหัสนักศึกษาลงไป จากนั้นระบบจะบอกกลับมารถของผู้ใช้จะไปถูกจอดอยู่ในช่องใดของลานจอดรถ

ในการเลือกที่จอดครั้นเราได้เลือกจากช่องที่ใกล้  IN  OUT มากที่สุดก่อนเพื่อหุ่นยนต์จะได้เคลื่อนที่น้อยที่สุดทำให้ประหยัดพลังงาน

3. จากนั้นหุ่นยนต์จะได้รับคำสั่งให้มารับรถของผู้ใช้จากจุด  IN ไปยังช่องที่กำหนด เมื่อหุ่นยนต์ทำตามคำสั่งแล้วยังไม่ได้รับคำสั่งต่อไป หุ่นยนต์จะอยู่ที่เดิมเพื่อรอคำสั่ง

### การนำรถออก

1. ผู้ใช้จะต้องไปยังคอมพิวเตอร์  COM และพิมพ์รหัสนักศึกษาอีกครั้ง
2. หุ่นยนต์จะนำรถของผู้ใช้มาคืนให้ที่จุด  OUT
3. ผู้ใช้ขึ้นรถและขับลง pallet กลับบ้านได้ 😊

## วัสดุและอุปกรณ์ที่ใช้ทั้งหมดในการทำแบบจำลองระบบจรวดอัตโนมัติ

ล้อ Omni	3	ล้อ
มอเตอร์ EX106+	4	ตัว
Nucleo	1	ตัว
Odriod XU4	1	ตัว
Raspberry Pi WiFi dongle	1	ตัว
MPU9250 (IMU)	1	ตัว
MAX485	1	ตัว
Stepdown(12V to 5V , 12V to 3.3V)	2	ตัว
Battery 12 V	1	อัน
Lidar	1	อัน
3D Print		
พลาสติก		

### ราคา

ล้อ Omni	$600 * 3 = 1800$ บาท
Motor gear servo Encoder 12V 150 RPM (แบบ EX106+)	$650 * 4 = 2600$ บาท
Nucleo	400 บาท
Odroid XU4	4,500 บาท
Raspberry Pi WiFi dongle	350 บาท
MPU9250 (IMU)	480 บาท
MAX485	30 บาท
Stepdown (12V to 5V, 12V to 3.3V)	$38 * 2 = 76$ บาท
Battery 12 V	1390 บาท
Lidar	6,000 บาท



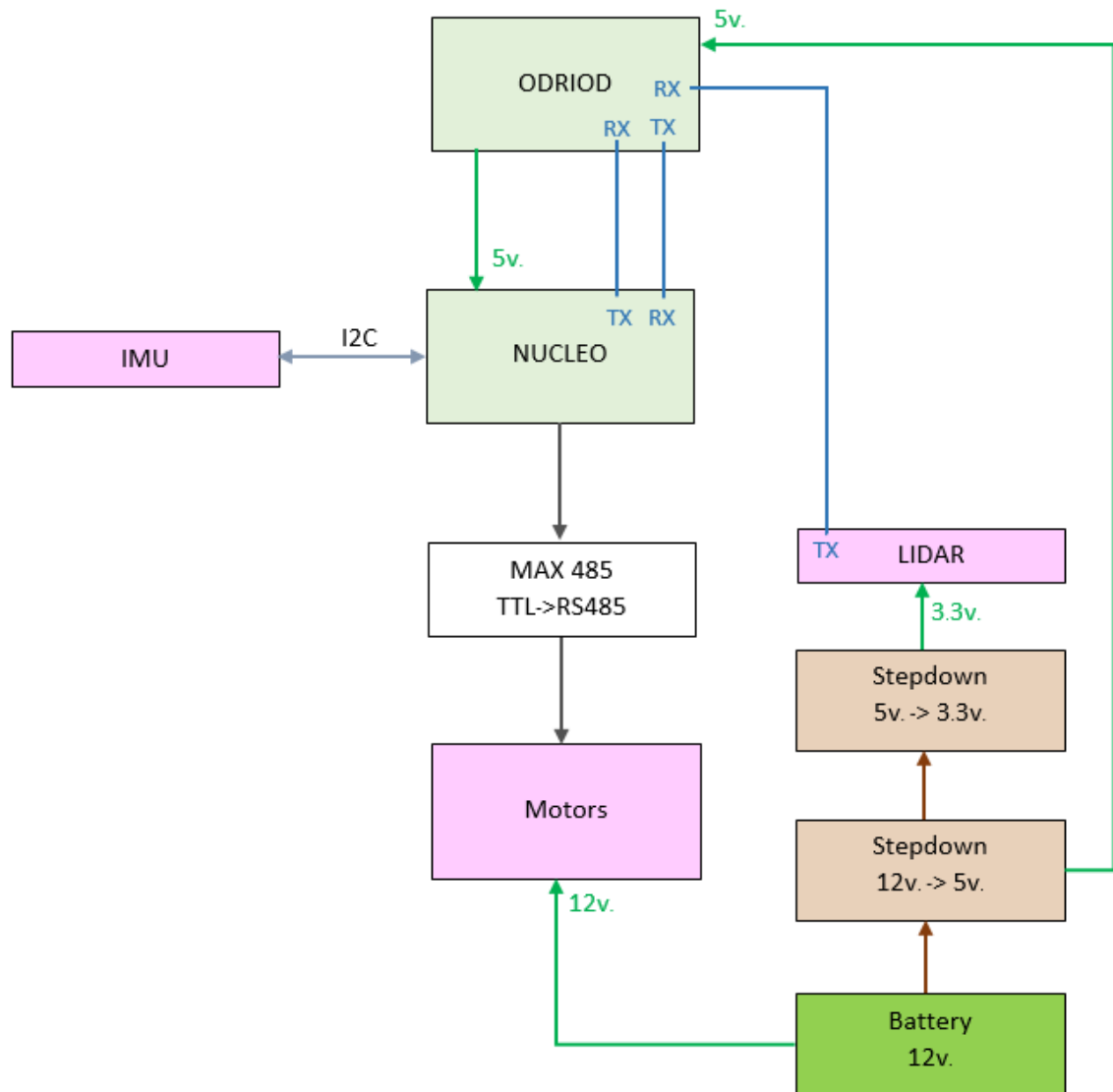
## การออกแบบหุ่นยนต์จ่อครถอัตโนมัติ

### - การออกแบบทางกล

- การเลือกล้อ Omni เนื่องจากเราต้องการทำระบบจ่อครถอัตโนมัติที่มีลานจ่อครถแบบช่องๆ คล้ายในห้างซึ่งถ้าออกแบบให้ใช้ ล้อ Omni จะทำให้หุ่นยนต์สามารถเคลื่อนที่หน้า หลัง ซ้าย ขวาได้โดยไม่ต้องเข้าโค้ง ทำให้ประหยัดพื้นที่ของลานจ่อครถส่งผลให้มีที่จ่อครถมากขึ้น และ Coupling ออกแบบเป็นแบบใช้ friction
- การออกแบบให้หุ่นยนต์มี 2 ชั้น เนื่องจากชั้นแรกใช้ในการติดตั้ง Lidar เพื่อการทำ SLAM และชั้นที่สองใช้ในการติดตั้งระบบการยก Pallet
- ระบบการยก Pallet เลือกใช้รูปแบบ mechanism 4-bar linkage เป็นแบบ Parallelogram linkage เพื่อให้แผ่นหน้าสัมผัสกับ Pallet ขนานกับพื้นตลอดเวลาซึ่งการออกแบบเน้นให้มีขนาดเล็ก และชิ้นส่วนที่ใช้ในการประกอบน้อย
- ในการประกอบ part ที่ชิ้นใหญ่จะใช้วัสดุเป็นพลาสติก เช่น ฐานของหุ่นยนต์ ในส่วนอื่นๆ ที่มีความละเอียดของชิ้นงาน หรือมีขนาดเล็กจะใช้ 3D print ในการขึ้นรูปทั้งหมด
- ความเร็วสูงสุดของหุ่นยนต์  
เนื่องจากมอเตอร์สามารถทำความเร็วได้มากที่สุด 57 rpm  
ความเร็วหุ่นยนต์ =  $\sin\left(57 \times \frac{2\pi r}{60}\right) = 0.155 \text{ m/s}$  ;  $r = 0.03 \text{ m}$

- การออกแบบทางไฟฟ้า

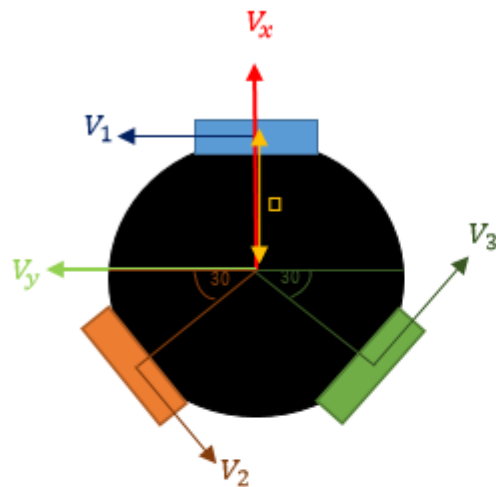
○ Diagram การติดต่อและส่งไฟฟ้าระหว่างอุปกรณ์



- Odriod เป็นศูนย์รวมการติดต่อระหว่าง Lidar, Nucleo และคอมพิวเตอร์ ซึ่งเราจะ run ROS ทั้งบนคอมพิวเตอร์ เพื่อแสดงผลผ่าน RVIZ และ อ่านค่า Lidar ผ่าน ROS บน Odriod อีกทั้ง Odriod ยังมีหน้าที่รับคำสั่งจากคอมพิวเตอร์ และส่งคำสั่งนั้นต่อไปให้ Nucleo เพื่อทำการสั่งมอเตอร์ให้เคลื่อนที่ตามที่ต้องการ

- การออกแบบทาง Software

○ การ **mapping** ความเร็วรถไปเป็นความเร็วแต่ละล้อ



$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & R \\ -\cos 30 & -\cos 60 & R \\ \cos 30 & -\cos 60 & R \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix}$$

- ระบบจอร์นของเราใช้การทำ SLAM ในการทำ Localization และ mapping โดยในรอบแรกของการทำงานเราจะให้หุ่นยนต์เดินไปใน map เพื่อทำการสร้าง map ขึ้นมาก่อน จากนั้นจึงสามารถใช้งานได้ การทำ SLAM นั้นทำให้ระบบจอร์นของเรายืดหยุ่นมากขึ้น สามารถใช้ได้กับที่จอร์นหลายรูปแบบ

☐ การทำ SLAM

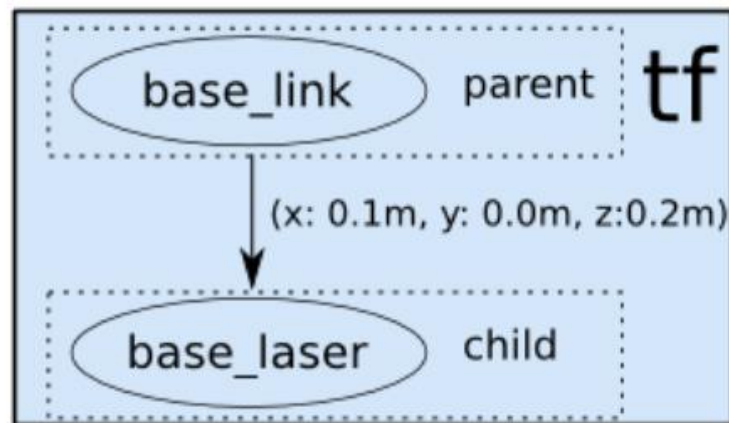
- ☐ ใช้ package ของ hector slam

Ref : [http://wiki.ros.org/hector\\_slam/Tutorials/SettingUpForYourRobot](http://wiki.ros.org/hector_slam/Tutorials/SettingUpForYourRobot)

[http://wiki.ros.org/hector\\_mapping](http://wiki.ros.org/hector_mapping)

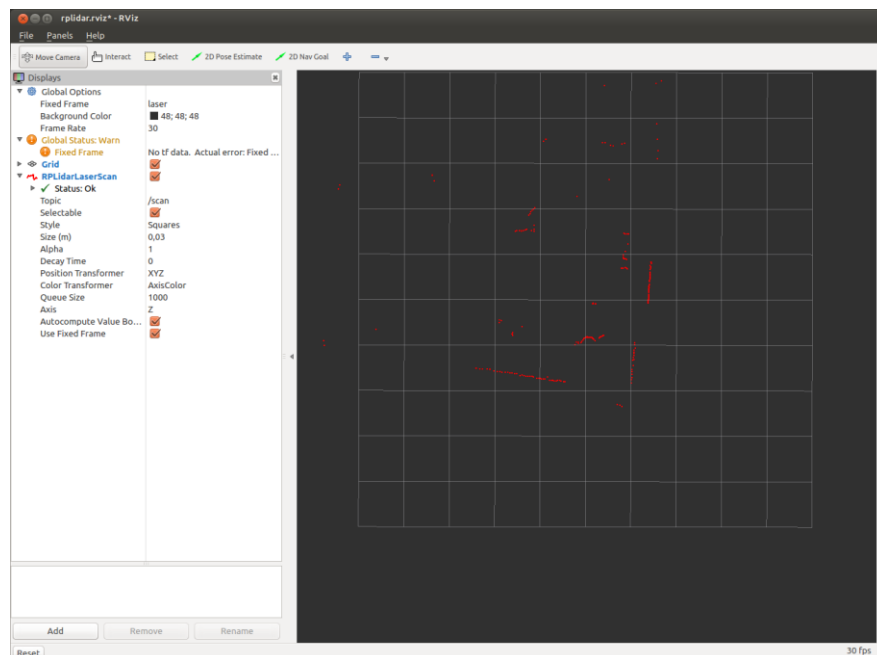


จากรูปตัวอย่าง จะเห็นว่า Laser ไม่ได้ถูกติดตั้งที่ตำแหน่งเดียวกับ Base Frame ทำให้ค่าที่ได้จากการอ่านเซ็นเซอร์ของ Lidar จะไม่ตรงกับหุ่น ดังนั้นสิ่งที่เราต้องทำคือ ทำการ Transform Frame จาก Base Frame ไป Laser Frame โดยการใช้ Package ที่ชื่อว่า tf



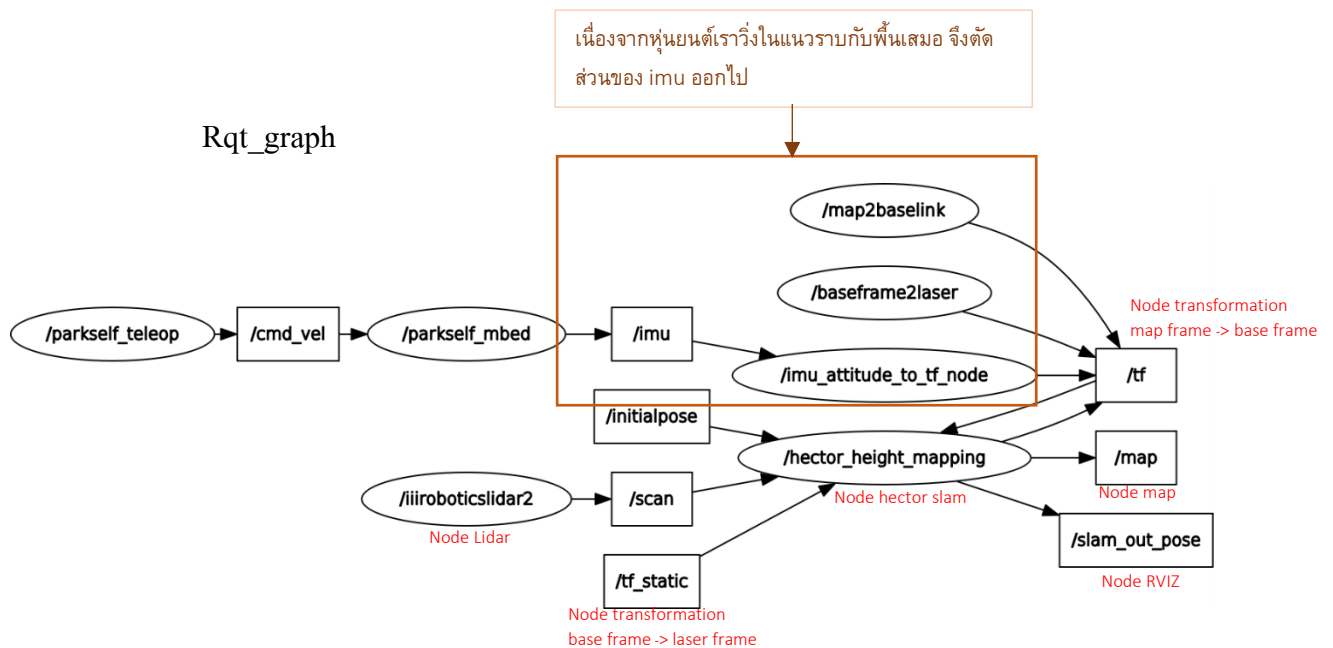
ซึ่งในส่วนหุ่นยนต์ Parkself ของเรานั้น ตำแหน่งเซ็นเซอร์ Lidar จะติดตั้งอยู่ที่ตำแหน่งเดียวกับ Base Frame จึงไม่จำเป็นต้องทำการ Transform Frame

หลังจากที่ทำการ Setup Frame เรียบร้อยแล้วก็จะทำการทดสอบว่า Lidar สามารถอ่านค่าได้หรือไม่ โดยการนำค่าที่ส่งออกมาจาก Topic /scan ไปแสดงผลบน RVIZ



จากการนำค่ามาแสดงผลจะเห็นว่า Lidar สามารถตรวจเจอสิ่งกีดขวางได้

เมื่อตรวจสอบเรียบร้อยแล้วก็จะทำการนำ Package ของ Hector\_Slam มาใช้



Topic	Message type	Publisher	Subscriber
/cmd_vel	geometry_msgs/Twist	/parkself_teleop	/parkself_mbed
/imu	sensor_msgs/Imu	/parkself_mbed	/imu_attitude_to_tf_node
/initialpose	geometry_msgs/PoseWithConvianceStamped	-	/hector_height_mapping
/scan	sensor_msgs/LaserScan	/iiroboticslidar2	/hector_height_mapping
/tf_static	tf2_msgs/TFMessage	-	/hector_height_mapping
/tf	tf2_msgs/TFMessage	/map2baselink /baseframe2laser /imu_attitude_to_tf_node /hector_height_mapping	/hector_height_mapping
/map	nav_msgs/OccupancyGrid	/hector_height_mapping	-
/slam_out_pose	geometry_msgs/PoseStamped	/hector_height_mapping	-

Node Lidar จะส่งข้อมูลความห่างระหว่างตัวเองกับ obstacle มาในรูป 2D ผ่าน topic ชื่อ “/scan”

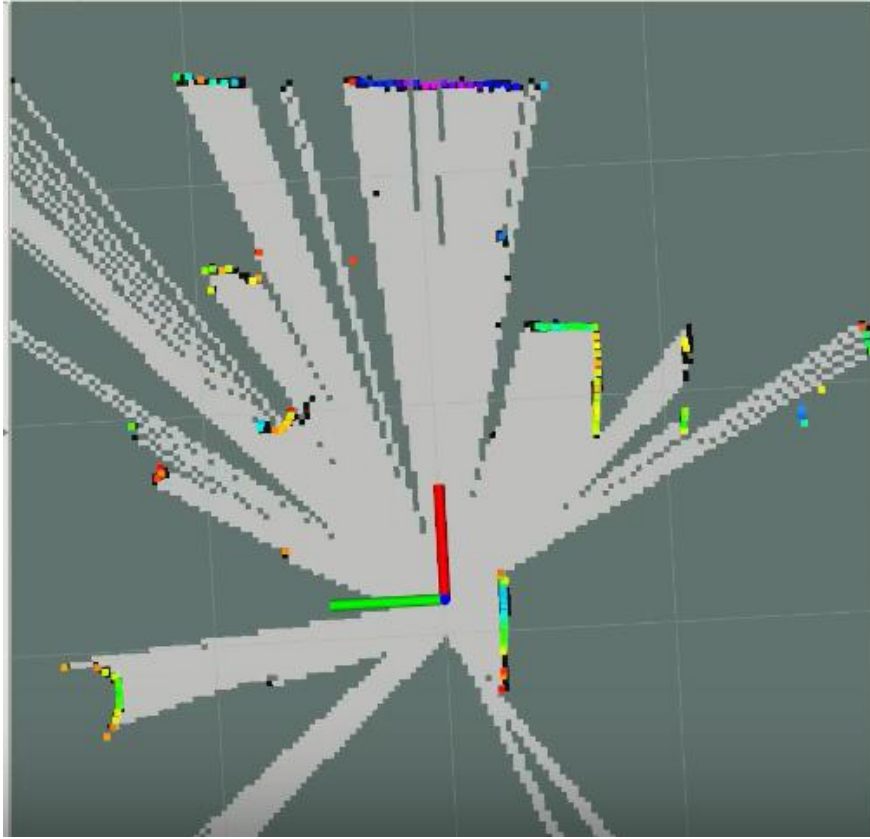
ให้กับ Node hector slam เพื่อทำการหาตำแหน่งของตัวเอง และ สร้าง Map ออกมา โดยจะสังเกตเห็นได้ว่า

จาก Node hector slam นั้นจะมี Output Topic ออกมา 2 อันคือ 1. /map 2. /slam\_out\_pose ซึ่งเราจะนำ 2

Topic นี้ส่งไปให้ RVIZ แสดงผลตำแหน่งของหุ่นกับแผนที่ และ นอกจากนี้ยังนำค่าจาก Topic

/slam\_out\_pose ไปทำการ Control หุ่นต่อไป

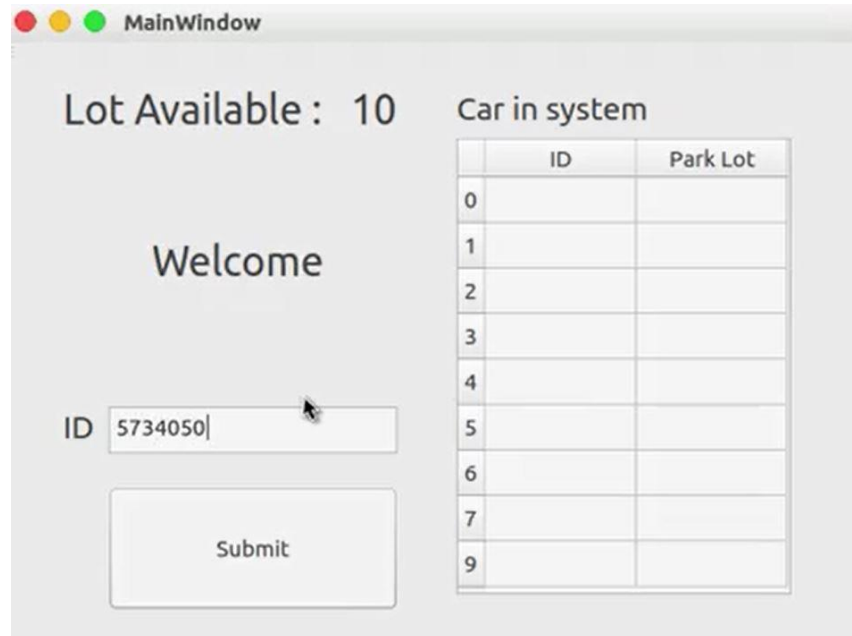




รูปตัวอย่างที่ได้จากการแสดงผลบน RVIZ

## □ การทำ GUI

ใช้โปรแกรม qt ในการทำ GUI ซึ่งใช้ภาษา C++ ในการเขียน



ช่อง ID ไว้สำหรับ กรอก รหัสนักศึกษาเพื่อนำรถเข้าจอดหรือนำรถออกจากลานจอดรถ หลังจากกรอก ID เสร็จให้กดปุ่ม Submit เพื่อยืนยัน หลังจากนั้นจะแสดงค่ารถที่จอดอยู่ทั้งหมดในลานจอดรถตอนนี้ในตาราง Car in system และจะแสดงค่า ช่องจอดรถที่ว่างอยู่ว่ามีกี่ช่องตรง Lot Available

### ปัญหาที่พบ

1. บอร์ด stepdown มีการใช้ switching regulator เพื่อลดแรงดันไฟฟ้าให้เหลือ 5 V แต่ปัญหาที่ตามมาคือ switching regulator มี ripple ที่เกิดจากการแปลงไฟ ทำให้ไฟที่เข้า odriod ไม่เสถียร
  - แก้ปัญหา ใส่ capacitor เข้าไปเพื่อ ลด ripple และกรองสัญญาณรบกวนความถี่สูงออกไป เพราะว่า C ทำหน้าที่เป็น low pass filter ในตัว
2. หุ่นยนต์เข้าในช่องที่จอดรถแล้วค่าที่อ่านได้จาก lidar เพี้ยน เพราะว่า ตั้งค่า length การอ่านค่า lidar ผิด ตั้งค่าไกลเกินไปทำให้อ่านผนังที่อยู่ใกล้ๆ ไม่เจอ
  - แก้ปัญหา แก้ค่า length การอ่านให้ถูกต้อง ลดระยะ length การอ่านลง

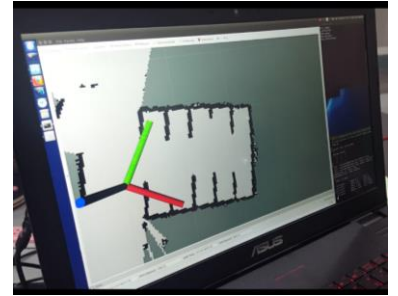
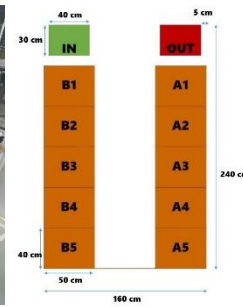
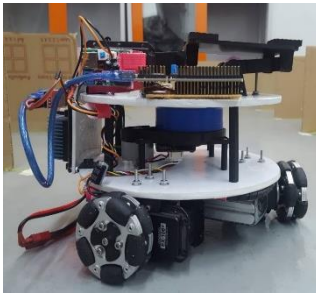
## การทดสอบ

1. ทดสอบการเคลื่อนที่ของหุ่นยนต์จากจุดหนึ่งไปอีกจุดหนึ่ง : <https://youtu.be/zArprLdjBQ>
2. ทดสอบการเคลื่อนที่และยกและวางPallet : <https://youtu.be/Mwb62RoV81k>
3. ทดสอบการนำรถไปจอดแบบ differential drive robot : <https://youtu.be/a6179N44QdM>
4. ทดสอบการ Mapping & Localization (SLAM) ในลานจอดรถจำลอง : <https://youtu.be/6HYM-FK-uZY>
5. ทดสอบการ Mapping & Localization (SLAM) ในสถานที่ที่มีสิ่งกีดขวางอื่นๆ : <https://youtu.be/2rrewLvYRqg>
6. ทดสอบการใช้ GUI ในการนำรถเข้า-ออก ที่จอดรถจำลอง โดยจะเลือกจอดช่องที่ใกล้ทางเข้า-ออกที่สุด ก่อน  
เสมอ : <https://youtu.be/VzuiPRfFy-Y>
7. ทดสอบการนำรถไปจอดโดยใช้ SLAM : <https://youtu.be/HoaKIPLXdo>
8. ทดสอบการนำรถจากช่อง B1 ไปยัง A3 และนำรถจากช่อง A4 ไปยังทางออก โดยใช้ SLAM แบบ robot ใน  
RVIZ ด้วย เปิด 2 จอคู่กันค่ะ มันจะรันไปพร้อมกัน  
robot ในสนามจริง : <https://youtu.be/RYkC8fQSltg>  
RVIZ : [https://www.youtube.com/watch?v=106BSAoI\\_wA&feature=youtu.be](https://www.youtube.com/watch?v=106BSAoI_wA&feature=youtu.be)

## วิดีโอการทำงานของระบบ

- วิ่งแบบ differential drive robot : <https://youtu.be/cwKKh0i59XQ>
- วิ่งสี่ล้อขับเคลื่อนแบบ omni : <https://youtu.be/Y11sob1zI40>

## สรุปผลงาน



## Parkself

### ระบบจอดรถอัตโนมัติ (หุ่นยนต์ล้อ Omni 3 ล้อ)

#### ที่มาและความสำคัญ

เนื่องจากการหาที่จอดรถในอาคารหรือลานจอดรถเป็นสิ่งที่ทำได้ยาก เนื่องจากเราไม่ทราบว่ามีที่จอดรถช่องไหนว่างบ้าง เราจึงต้องวนไปเรื่อยๆ เพื่อหาที่จอดรถ ซึ่งทำให้เปลืองน้ำมัน และยังเสียเวลาอีกด้วย เราจึงคิดค้นระบบอัตโนมัติในการจอดรถ เพื่อลดอัตราการสูญเสียพลังงานซึ่งรวมถึงค่าน้ำมัน และเวลาของผู้ขับขี่ในการหาที่จอดรถ อีกทั้งยังทำให้จอดรถได้เป็นระเบียบและประหยัดพื้นที่มากขึ้นอีกด้วย

#### จุดเด่นของ Parkself

ในปัจจุบันการติดตั้งระบบที่จอดรถอัตโนมัตินั้นเราจำเป็นต้องเปลี่ยนโครงสร้างของที่จอดรถเกือบทั้งหมดซึ่งทำให้เกิดความยุ่งยาก แต่เนื่องจาก parkself มีการทำ SLAM จึงใช้ได้กับที่จอดรถทุกที่ และเพียงแค่ซื้อหุ่นยนต์และ pallet ของเราไป ก็จะสามารถติดตั้งระบบจอดรถอัตโนมัติได้ในพื้นที่จอดรถเดิม

#### Function ของ Parkself

- หุ่นยนต์สามารถยกและวาง pallet ได้ตามเป้าหมาย
- หุ่นยนต์สามารถทำการ mapping และ localization ได้ด้วยการใช้ SLAM

#### วิธีการใช้

##### การนำรถเข้าจอด

1. ผู้ใช้นำรถมาจอดบน pallet ซึ่งเตรียมไว้ในโซน **IN**
2. ผู้ใช้ลงจากรถ ไปยังคอมพิวเตอร์ **COM** จากนั้นจะเจอกับ GUI ซึ่งจะให้การอธิบายขั้นตอนการลงจอด จากนั้นระบบจะบอกกลับมาว่ารถของผู้ใช้จะไปถูกจอดอยู่ในช่องใดของลานจอดรถ
3. จากนั้นหุ่นยนต์จะได้รับคำสั่งให้มารับรถของผู้ใช้จากจุด **IN** ไปยังช่องที่กำหนด เมื่อหุ่นยนต์ทำตามคำสั่งแล้วยังไม่ได้รับคำสั่งต่อไป หุ่นยนต์จะอยู่ที่เดิมเพื่อรอคำสั่ง

##### การนำรถออก

1. ผู้ใช้จะต้องไปยังคอมพิวเตอร์ **COM** และพิมพ์รหัสนักศึกษาอีกครั้ง
2. หุ่นยนต์จะนำรถของผู้ใช้มาคืนให้ที่จุด **OUT**
3. ผู้ใช้ขึ้นรถและขับลง pallet กลับบ้านได้

วิดีโอการทำงานของระบบ : <https://youtu.be/cwKKh0i59XQ>