

Automotive Minions

รถจัดเก็บสินค้าอัตโนมัติ

Member

นายจุฬภัทร จิรชัย	57340500013
นายเจตน์นัท หอมจันทนากุล	57340500015
นายวุฒิภัทร โชคอนันตทรัพย์	57340500067
นายศิริวัชร ศกศวัตเมฆิทร์	57340500071

Insitute of Field Robotics, KMUTT



สารบัญ

เรื่อง	หน้า
สารบัญ	2
รายการรูปภาพ	3
บทที่ 1 บทนำ	4
1.1 ที่มาและความสำคัญ	4
1.2 วัตถุประสงค์	4
1.3 ขอบเขตของโครงการ	4
บทที่ 2 ออกแบบระบบ	5
2.1 คำอธิบายโครงการ	5
2.2 องค์ประกอบโดยรวมของระบบ	5
2.2.1 Cloud	6
2.2.2 High Level	6
2.2.3 Low Level	6
บทที่ 3 พัฒนาระบบ	7
3.1 Hardware	7
3.2 Server and Database	8
3.3 GUI for Monitor	9
3.4 GUI for Control	10
3.5 Create map with SLAM	11
3.6 Create prototype robot	12

รายการรูปภาพ

รูป	หน้า
รูปที่ 2.1 องค์ประกอบโดยรวมของระบบ	5
รูปที่ 3.1 ส่วนประกอบของหุ่นยนต์	7
รูปที่ 3.2 ลักษณะของหุ่นยนต์สองล้อ	7
รูปที่ 3.3 ตารางแสดงชื่อและชนิดข้อมูล	8
รูปที่ 3.4 GUI สำหรับแสดงผลสถานะการทำงาน	9
รูปที่ 3.5 GUI สำหรับทดสอบการทำงาน	10
รูปที่ 3.6 ทดสอบการสร้างแผนที่ด้วย SLAM ผ่าน RViz ขั้นที่ 1	11
รูปที่ 3.7 ทดสอบการสร้างแผนที่ด้วย SLAM ผ่าน RViz ขั้นที่ 2	11
รูปที่ 3.8 ขั้นตอนการประกอบหุ่นยนต์ต้นแบบ ขั้นที่ 1	12
รูปที่ 3.9 ขั้นตอนการประกอบหุ่นยนต์ต้นแบบ ขั้นที่ 2	12
รูปที่ 3.10 ขั้นตอนการประกอบหุ่นยนต์ต้นแบบ ขั้นที่ 3	12

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันการเคลื่อนย้ายสินค้าต่างๆภายในโรงงานยังใช้แรงงานมนุษย์อยู่ โดยวิธีการทั่วไปที่โรงงานนิยมใช้คือ การใช้รถเข็นลำเลียงเพื่อเคลื่อนย้ายสินค้า ซึ่งผู้ที่ใช้รถเข็นจะต้องมีประสบการณ์ในการใช้ และต้องมีการระวังอยู่ตลอดเวลาการใช้งานรถเข็น แต่เนื่องจากการทำงานของมนุษย์มีความเสี่ยงที่จะทำให้เกิดความผิดพลาดได้สูงจากปัจจัยหลายอย่าง เช่น การพักผ่อนไม่เพียงพอ สภาพร่างกายไม่พร้อม สภาพจิตใจที่ไม่พร้อม อารมณ์ต่างๆ ซึ่งความผิดพลาดจะนำมาซึ่งความเสียหายต่อโรงงานได้ ดังนั้นการพัฒนาการขับเคลื่อนอัตโนมัติ AGV (Automated Guided Vehicle) ซึ่งเป็นเครื่องจักรชนิดหนึ่งที่ใช้คอมพิวเตอร์ในการควบคุม มีความสามารถในการเดินทางตามเส้นทางที่กำหนดไว้ การนำรถขับเคลื่อนอัตโนมัติมาเพื่อทดแทนการทำงานของมนุษย์ในส่วนนี้จึงเป็นสิ่งสำคัญ โดยนอกจากจะตัดปัญหาในส่วนการทำงานที่ผิดพลาดแล้ว ยังช่วยให้โรงงานมีประสิทธิภาพในการบริหารจัดการคลังสินค้ามากขึ้น ด้วยแนวคิดดังกล่าวจึงทำให้เกิดหัวข้อโครงงานนี้ขึ้นมา

1.2 วัตถุประสงค์

วัตถุประสงค์ในการทำโครงงานนี้ขึ้นมาก็เพื่อที่จะได้พัฒนาระบบการเคลื่อนย้ายลำเลียงสินค้าอัตโนมัติภายในคลังสินค้า พัฒนาระบบการขับเคลื่อนของหุ่นยนต์สองล้อสำหรับการขนย้ายสินค้า และได้ประยุกต์ใช้ IoT (Internet of Things) กับระบบจัดเก็บคลังสินค้าภายในโรงงานอุตสาหกรรม

1.3 ขอบเขตของโครงงาน

- 1.3.1 ออกแบบและพัฒนาระบบการเคลื่อนย้ายสินค้าภายในคลังสินค้าอัตโนมัติ
- 1.3.2 จำลองรถขับเคลื่อนอัตโนมัติโดยใช้หุ่นยนต์สองล้อ (Differential Drive Robot)
- 1.3.3 ออกแบบและพัฒนาให้หุ่นยนต์เคลื่อนที่ไปยังตำแหน่งที่กำหนดได้
- 1.3.4 ออกแบบและพัฒนาระบบ Cyber-Physical System (CPS) ให้กับการทำงานของ AGV

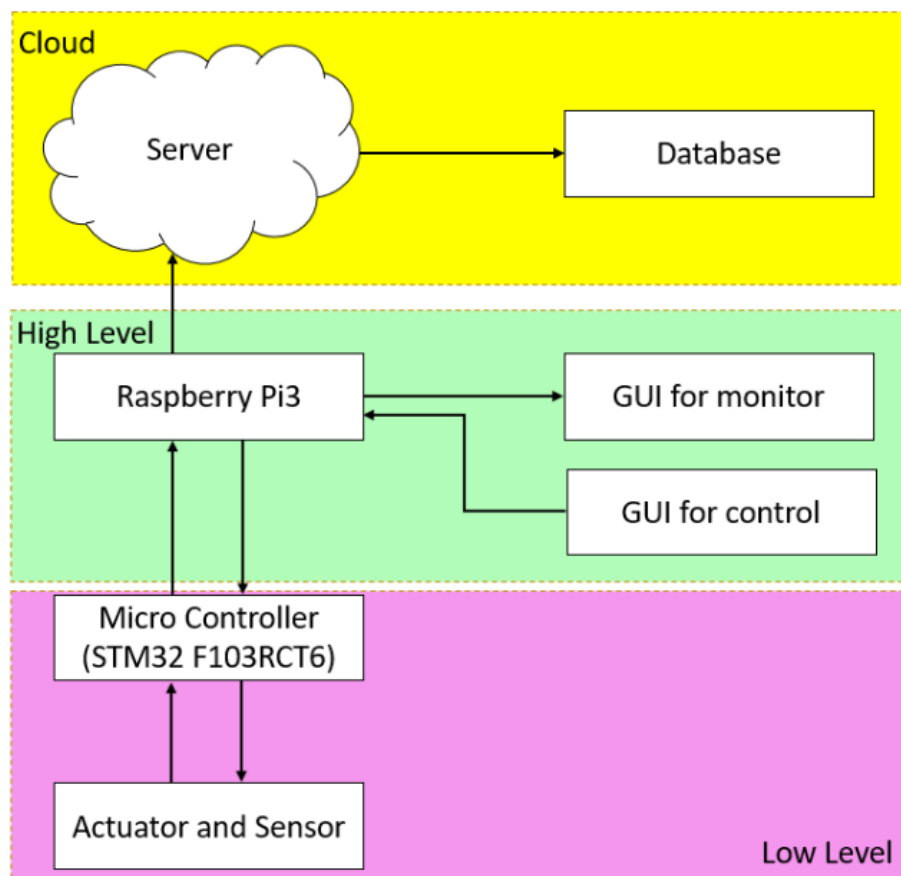
บทที่ 2

ออกแบบระบบ

2.1 คำอธิบายโครงการ

เป็นโครงการที่พัฒนาระบบการรับ-ส่งสินค้าเพื่อจัดเก็บใน Ware-House อัตโนมัติ โดยใช้ AGV ซึ่งจะอยู่ในรูปแบบของหุ่นยนต์ขับเคลื่อนสองล้อ โดยหุ่นยนต์สามารถยกสินค้า แล้วเดินไปวางตามจุดจัดเก็บที่กำหนดโดยใช้ Lidar ในการสร้างแผนที่เพื่อการนำทาง และจะมีการส่งข้อมูลไปบน Cloud เพื่อแสดงผลให้ผู้ใช้งานสามารถรับรู้ถึงสถานะของหุ่นยนต์ จากนั้นจะทำการส่งข้อมูลกลับไปยังหุ่นยนต์เพื่อให้หุ่นยนต์จัดเก็บสินค้าตามจุดจัดเก็บที่กำหนดได้ และเมื่อส่งสินค้าเสร็จแล้วก็จะส่งสถานะการทำงานกลับมายัง Cloud เพื่อจัดเก็บข้อมูล ซึ่งหุ่นยนต์จะส่งข้อมูลที่เกี่ยวข้องกับตัวหุ่นยนต์เองขึ้นไปบนระบบเพื่อนำไปประมวลผลและพัฒนาการทำงาน โดยข้อมูลของตัวหุ่นยนต์จะประกอบไปด้วย น้ำหนักที่หุ่นยนต์แบกรับ แรงดันไฟฟ้าที่ใช้ กระแสไฟฟ้าที่ใช้ ระยะเวลาที่ใช้ในการทำงาน จุดเริ่มต้น จุดสิ้นสุด วันและเวลาที่ปฏิบัติงาน

2.2 องค์ประกอบโดยรวมของระบบ



รูปที่ 2.1: องค์ประกอบโดยรวมของระบบ

ระบบสามารถแบ่งออกได้เป็น 3 องค์ประกอบใหญ่ด้วยกันคือ Cloud, High level และ Low level ดังรูปที่ 2.1 ซึ่งในแต่ละส่วนจะมีหน้าที่แตกต่างกันไปดังนี้

2.2.1 Cloud

เป็นส่วนประกอบที่นำมาใช้ในการจัดเก็บข้อมูลเพื่อให้สามารถนำผลลัพธ์ไปวิเคราะห์ประสิทธิภาพการทำงาน หรือพฤติกรรมของระบบได้ มีองค์ประกอบทั้งหมด 2 ส่วนด้วยกันคือ Server และ Database โดย Server เปรียบเสมือนตัวกลางที่ใช้ในการรับส่งข้อมูลระหว่างอุปกรณ์และ Database ซึ่ง Database เป็นฐานข้อมูลที่ถูกใช้ในการเก็บข้อมูลเกี่ยวกับสถานะและประวัติการทำงานของหุ่นยนต์

2.2.2 High Level

เป็นส่วนที่ใช้ในการประมวลผลหลักเพื่อสั่งการและแสดงผลการทำงานต่างๆ โดยจะมี Raspberry Pi3 เป็นคอนโทรลเลอร์ที่ใช้ในการจัดการข้อมูลและประมวลผลรวมไปถึงการแสดงผลในรูปแบบต่างๆ ซึ่งจะมี GUI for monitor ในการแสดงผลจากเซนเซอร์และอุปกรณ์ต่าง ๆ และมี GUI for control ที่เป็นอุปกรณ์ที่ให้ผู้ใช้งานได้ทำงานสั่งการระบบเพื่อให้หุ่นยนต์มีการทำงานตามคำสั่ง

2.2.3 Low Level

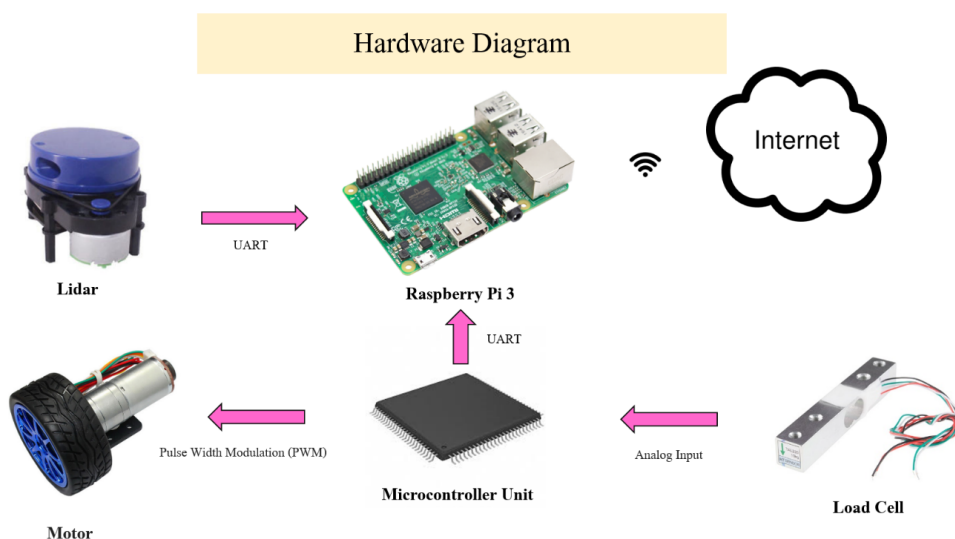
เป็นส่วนที่รับคำสั่งต่างๆ จาก High Level แล้วนำไปสั่งการกลไกและอุปกรณ์ต่างๆ ให้มีการทำงานเป็นไปตามที่กำหนดไว้ และทำการอ่านค่าจากเซนเซอร์และอุปกรณ์ต่างๆ เพื่อส่งข้อมูลกลับไปยัง High Level ได้ทำการประมวลผล แสดงผล และทำหน้าที่ในการตัดสินใจ เพื่อสั่งงานให้หุ่นยนต์มีการทำงานอย่างมีระบบ

บทที่ 3

พัฒนาระบบ

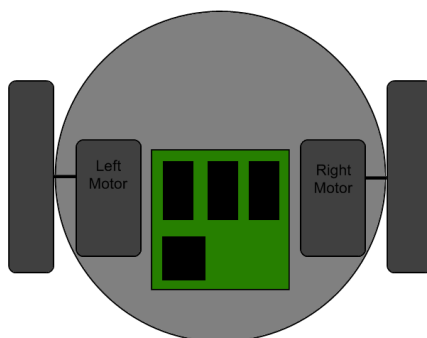
3.1 Hardware

ฮาร์ดแวร์พื้นฐานของหุ่นยนต์ขับเคลื่อน 2 ล้อตัวนี้จะประกอบไปด้วยฮาร์ดแวร์ดังต่อไปนี้มอเตอร์ ไมโครคอนโทรลเลอร์ เซ็นเซอร์วัดแรง เซ็นเซอร์วัดแสงรอบทิศทางในแนวสองมิติ และคอมพิวเตอร์ขนาดเล็ก โดยอุปกรณ์ทั้งหมดนี้จะมีการเชื่อมต่อกัน ดังรูปที่ 3.1 ซึ่งรูปแบบในการเชื่อมต่อก็จะภายในฮาร์ดแวร์กันเองก็จะมีที่ใช้ทั้ง



รูปที่ 3.1: ส่วนประกอบของหุ่นยนต์

หมดหลักๆสามชนิดคือ 1.PWM 2.UART 3.ADC ส่วนการเชื่อมต่อจากฮาร์ดแวร์ขึ้นไปบนซอฟต์แวร์นั้นจะผ่านทางสัญญาณไร้สาย โครงสร้างของหุ่นยนต์เคลื่อนที่ 2 ล้อ ล้อและมอเตอร์จะถูกติดตั้งในลักษณะที่เรียกว่า Differential drive เพราะว่าการติดตั้งแบบนี้จะมีข้อดี คือ การเลี้ยว หรือการหมุน สามารถทำได้โดยตำแหน่งของหุ่นยนต์ยังอยู่ที่เดิม ซึ่งเหมาะแก่การนำมาเป็นหุ่นยนต์ในการขนส่งสินค้าที่มีพื้นที่จำกัดได้



รูปที่ 3.2: ลักษณะของหุ่นยนต์สองล้อ

3.2 Server and Database

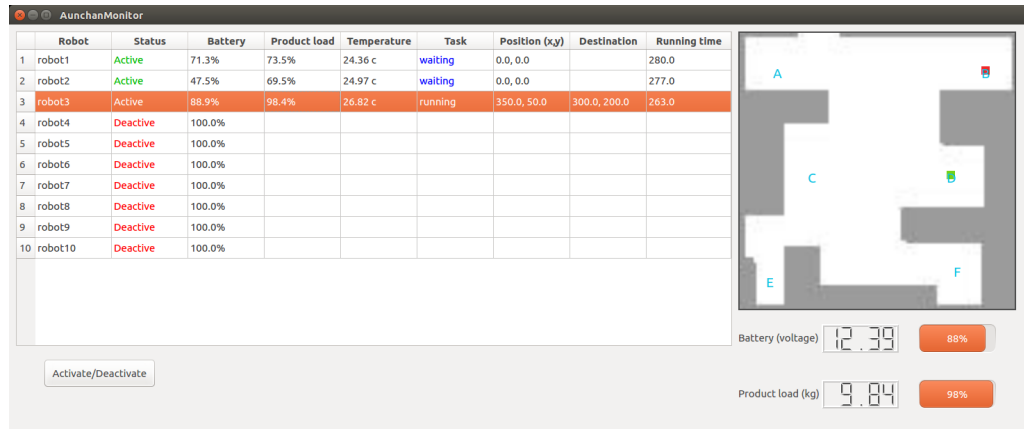
เป็นส่วนที่ใช้ในการเก็บข้อมูลต่างๆของหุ่นยนต์ทั้งหมด โดยหุ่นยนต์ทุกตัวจะเชื่อมต่อผ่าน Server เดียวกัน ซึ่ง Server จะถูกสร้างขึ้นบน Microsoft Azure และหุ่นยนต์ทุกตัวมีการใช้ Database ร่วมกัน แต่มีการใช้ตารางในการเก็บข้อมูลที่แตกต่างกัน โดยในแต่ละตารางจะมีทั้งหมด 10 หัวข้อ ประกอบไปด้วย วันที่ทำงาน เวลาที่ทำงาน แรงดันไฟฟ้า กระแสไฟฟ้า น้ำหนักที่แบกรับ ระยะทางที่ใช้ เวลาที่ใช้ จุดเริ่มต้น ปลายทาง และ อุณหภูมิ ณ ช่วงเวลานั้นๆ ซึ่งจะมีชนิดของตัวแปรดังนี้

ชื่อ	ชื่อ Column	ชนิดของตัวเก็บข้อมูล
วันที่ทำงาน	Date	date
เวลาที่ทำงาน	Time	time
แรงดันไฟฟ้า	Voltage Battery	float
กระแสไฟฟ้า	Current Battery	float
น้ำหนักที่แบกรับ	Load	float
ระยะทางที่ใช้	Distance	float
เวลาที่ใช้	Duration Time	time
จุดเริ่มต้น	Start Point	varchar(255)
ปลายทาง	Finish Point	varchar(255)
อุณหภูมิ	Temperature	float

รูปที่ 3.3: ตารางแสดงชื่อและชนิดข้อมูล

มีการพัฒนาระบบผ่านการเขียนโปรแกรมภาษา Python บน Virtual Machine และ Query Editor ภายใน Microsoft Azure เพื่อรับค่าที่หุ่นยนต์แต่ละตัวทำการส่งข้อมูลมาที่ Server ส่วนกลางโดยใช้รูปแบบ MQTT จากนั้นจึงทำการอ่านค่าจากการส่งข้อมูลรูปแบบ MQTT แล้วนำไปส่งข้อมูลขึ้นไปเก็บไว้ที่ Database

3.3 GUI for Monitor

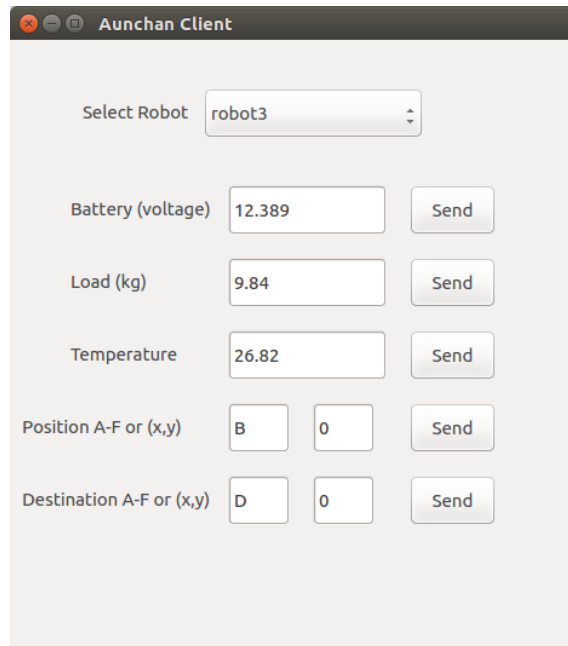


รูปที่ 3.4: GUI สำหรับแสดงผลสถานะการทำงาน

จากรูปที่ 3.4 เป็นส่วนที่มีไว้เพื่อตรวจสอบสถานะต่างๆของหุ่นยนต์ ซึ่งสามารถแบ่งเป็นดังนี้

- ชื่อ/หมายเลขของหุ่นยนต์
- สถานะปัจจุบัน แบ่งเป็น
 - Active คือ เปิดการใช้งานอยู่
 - Deactive คือ ไม่เปิดใช้งาน
 - Low Battery คือ พลังงานใกล้หมด โดยคิดจากแบตเตอรี่ต่ำกว่า 15% เมื่อหุ่นยนต์เข้าสู่สถานะนี้หากหุ่นยนต์มีน้ำหนักสินค้า 0% ทำการวิ่งไปชาร์จแบตเตอรี่อัตโนมัติ แต่ถ้าหากมีน้ำหนักสินค้ามากกว่านั้นหุ่นยนต์จะทำงานต่อไป
 - Charging คือ กำลังชาร์จพลังงาน หากหุ่นยนต์เข้าสู่สถานะนี้จะ กลายเป็นสถานะเช่นเดียวกับ Deactive และ ระบบจะไปทำการ Activate หุ่นยนต์ตัวอื่น 1 ตัวอย่างอัตโนมัติ เพื่อมาทำงานแทนตนเอง
 - Overload คือ สินค้าที่ขนมีน้ำหนักเกิน 100%
 - Overheat คือ มีอุณหภูมิสูงเกินที่กำหนด
- แบตเตอรี่ สามารถดูได้ทั้งแบบ voltage และแบบ percent
- น้ำหนักของสินค้าที่อ่านได้ สามารถดูได้ทั้งแบบ kg และแบบ percent
- อุณหภูมิที่อ่านได้จากเซนเซอร์ มีหน่วยเป็นองศาเซลเซียส
- ภาระหน้าที่ แบ่งเป็น
 - Waiting คือ กำลังรอเรียกใช้งานให้ไปรับ/ส่งสินค้า
 - Running คือ ได้รับการมอบหมายงานแล้ว กำลังจะเคลื่อนที่ไปจุดที่กำหนด
- ทดสอบการทำงานของหุ่นยนต์อีวมานานอยด์
- ตำแหน่งของหุ่นยนต์ ณ ปัจจุบัน
- ตำแหน่งของจุดหมายปลายทาง
- เวลาการเปิดใช้งานของหุ่นยนต์ นับตั้งแต่หุ่นยนต์เปลี่ยนเป็นสถานะ Active
- แผนที่, ตำแหน่งของหุ่นยนต์ในแผนที่ (จุดสีแดง) และจุดหมายปลายทาง (จุดสีเขียว)

3.4 GUI for Control

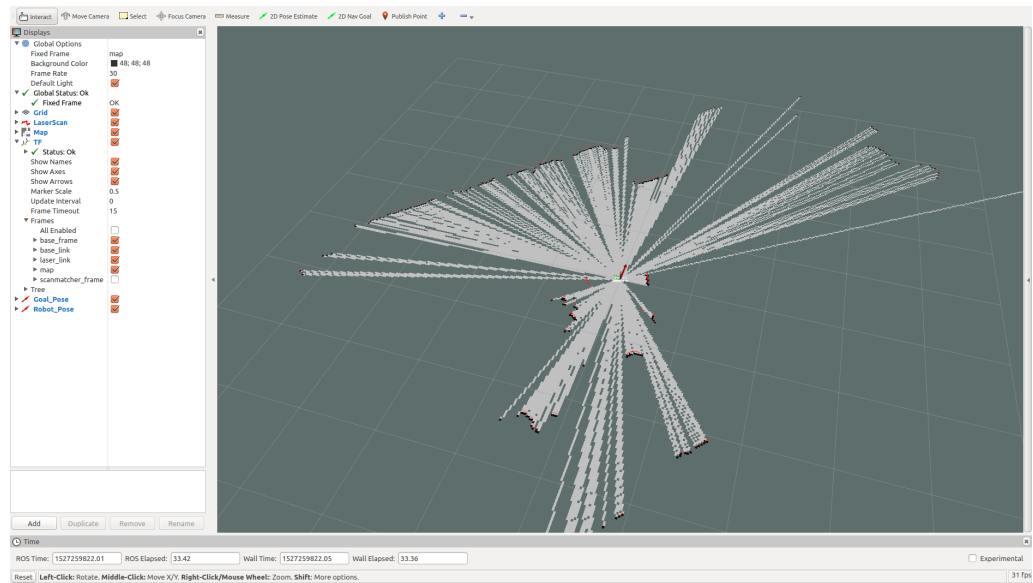


รูปที่ 3.5: GUI สำหรับทดสอบการทำงาน

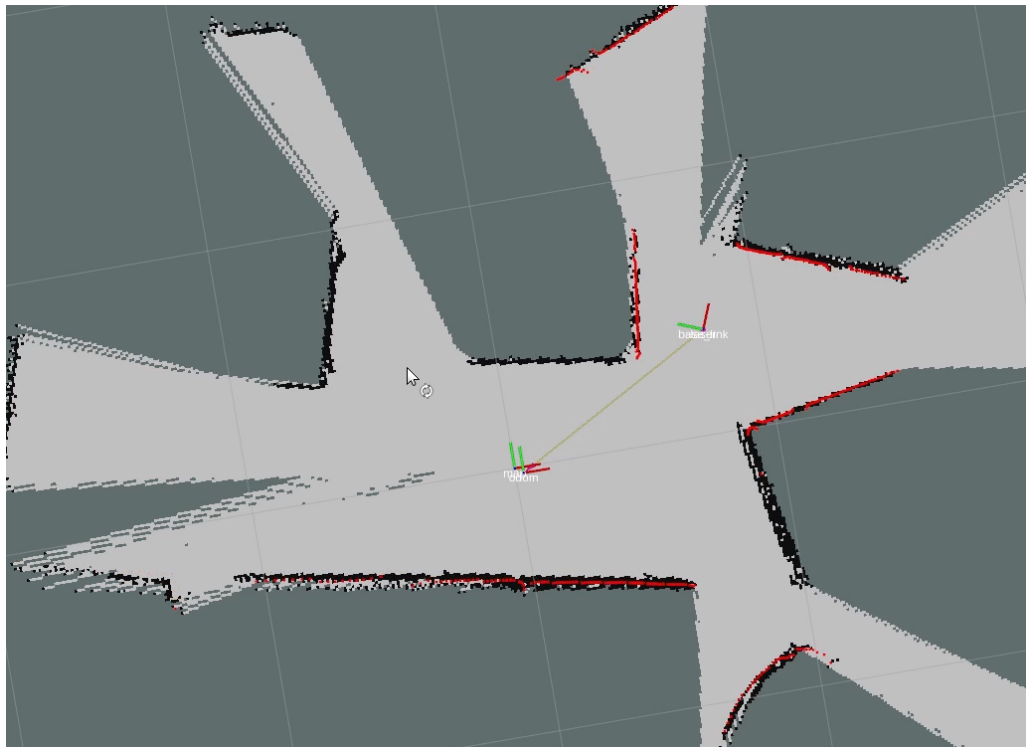
จากรูปที่ 3.5 มีไว้เพื่อทดลองการปรับเปลี่ยนค่าสถานะต่างๆของหุ่นยนต์ ประกอบไปด้วย

- 1 เลือกหุ่นยนต์ตัวที่ต้องการปรับเปลี่ยนค่า
- 2 เปลี่ยนค่าแบตเตอรี่โดยที่ 12.5v คือ 100% และ 11.5v คือ 0%
- 3 เปลี่ยนค่าน้ำหนักสินค้าโดยที่ 10 kg คือ 100% และ 0 kg คือ 0%
- 4 เปลี่ยนค่าอุณหภูมิ
- 5 เปลี่ยนตำแหน่งปัจจุบันของหุ่นยนต์ โดยที่สามารถใส่ A-F เพื่อไปจุดที่ mark ไว้ในแผนที่ได้ หรือใส่พิกัด x,y เพื่อให้ไปอยู่ตำแหน่งนั้นๆในแผนที่
- 6 เปลี่ยนตำแหน่งจุดหมายปลายทางของหุ่นยนต์ โดยที่สามารถใส่ A-F เพื่อไปจุดที่ mark ไว้ในแผนที่ได้ หรือใส่พิกัด x,y เพื่อให้ไปอยู่ตำแหน่งนั้นๆในแผนที่

3.5 Create map with SLAM

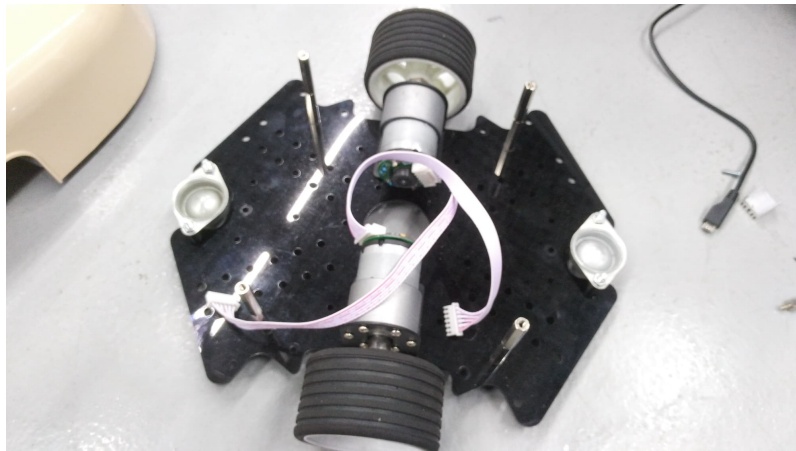


รูปที่ 3.6: ทดสอบการสร้างแผนที่ด้วย SLAM ผ่าน RViz ขั้นที่ 1

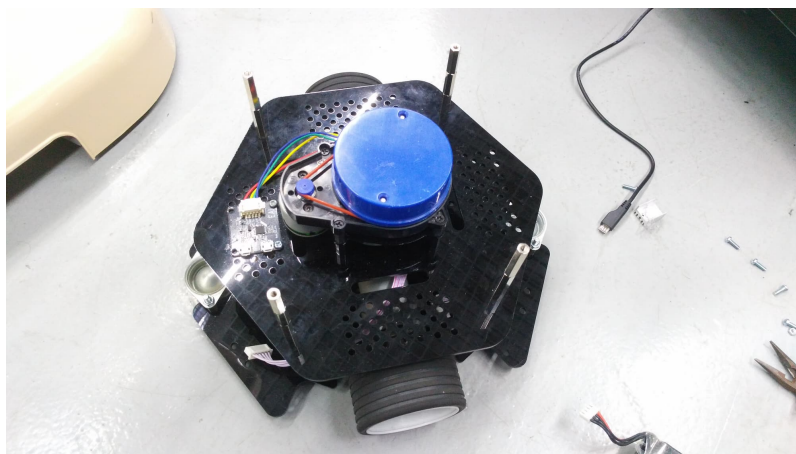


รูปที่ 3.7: ทดสอบการสร้างแผนที่ด้วย SLAM ผ่าน RViz ขั้นที่ 2

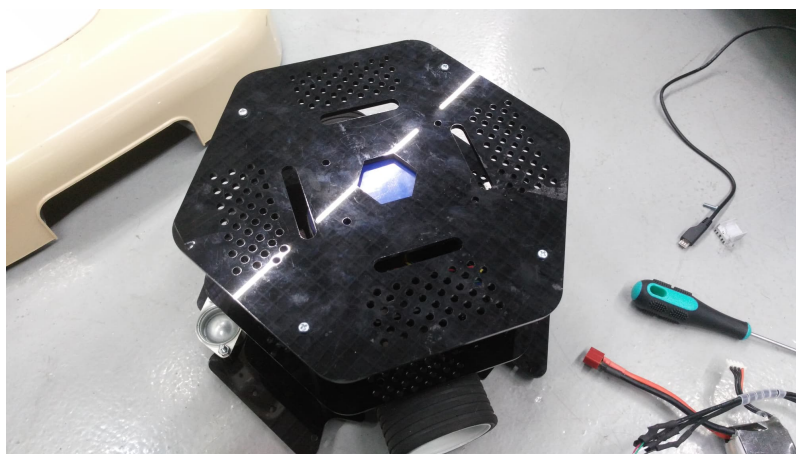
3.6 Create prototype robot



รูปที่ 3.8: ขั้นตอนการประกอบหุ่นยนต์ต้นแบบ ขั้นที่ 1



รูปที่ 3.9: ขั้นตอนการประกอบหุ่นยนต์ต้นแบบ ขั้นที่ 2



รูปที่ 3.10: ขั้นตอนการประกอบหุ่นยนต์ต้นแบบ ขั้นที่ 3