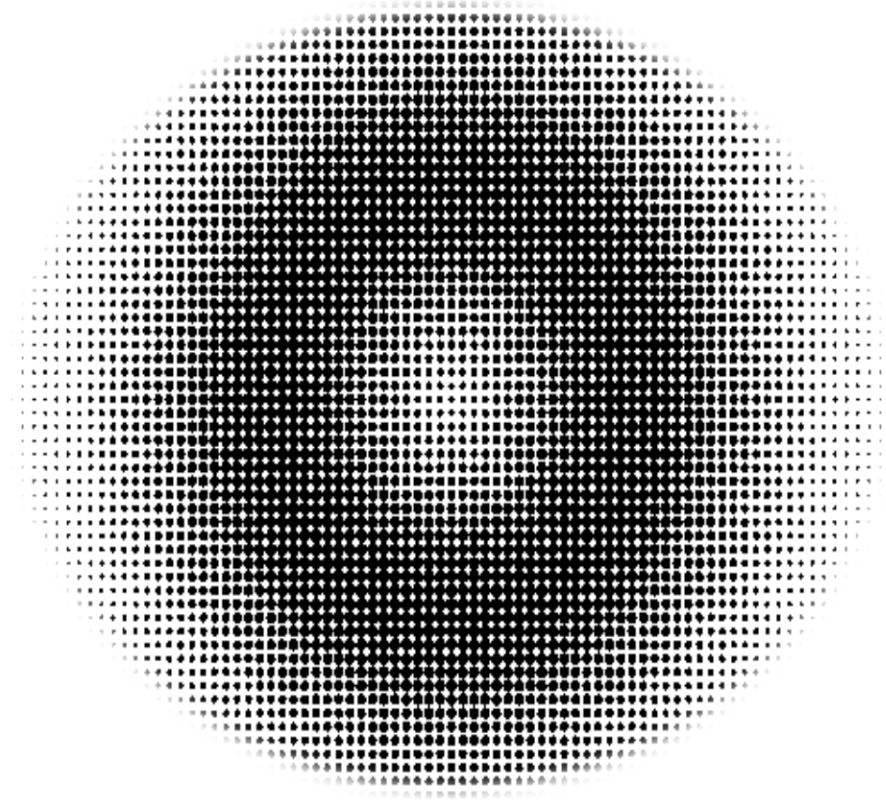




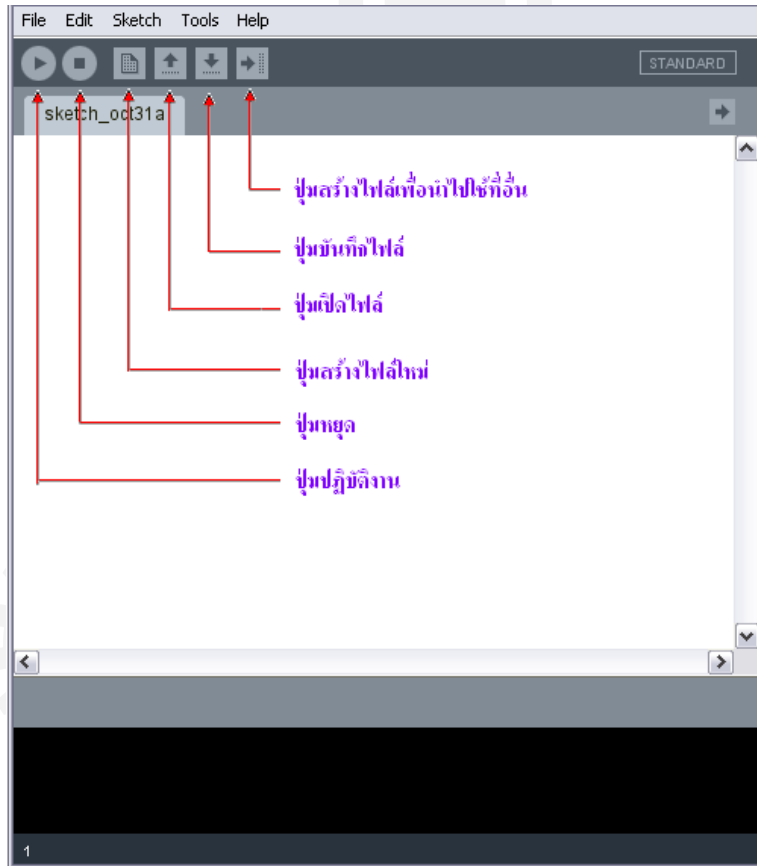
MARA : Manufacturing Automation and Robotics Academy

หลักสูตรการยกระดับฝีมือ
สาขา การควบคุมหุ่นยนต์ลำเลียง
(Control Robot transportation)

การเขียนโปรแกรม Interface ด้วย โปรแกรม Processing



แนะนำโปรแกรม/การติดตั้งโปรแกรม



ปุ่มบาร์ (ToolBar)

แท็บ (Tab)

ปุ่มสร้างไฟล์เพื่อแก้ไขที่อื่น

ปุ่มบันทึกไฟล์

ปุ่มเปิดไฟล์

ปุ่มสร้างไฟล์ใหม่

ปุ่มหยุด

ปุ่มปฏิบัติการ

พื้นที่เขียนข้อความ (Text editor)

พื้นที่ข้อความ (Message area)

คอนโซล (Console)

ลิงค์ดาวน์โหลดโปรแกรม Processing

<https://processing.org/download/>



3.5.4 (17 January 2020)

Windows 64-bit

Linux 64-bit

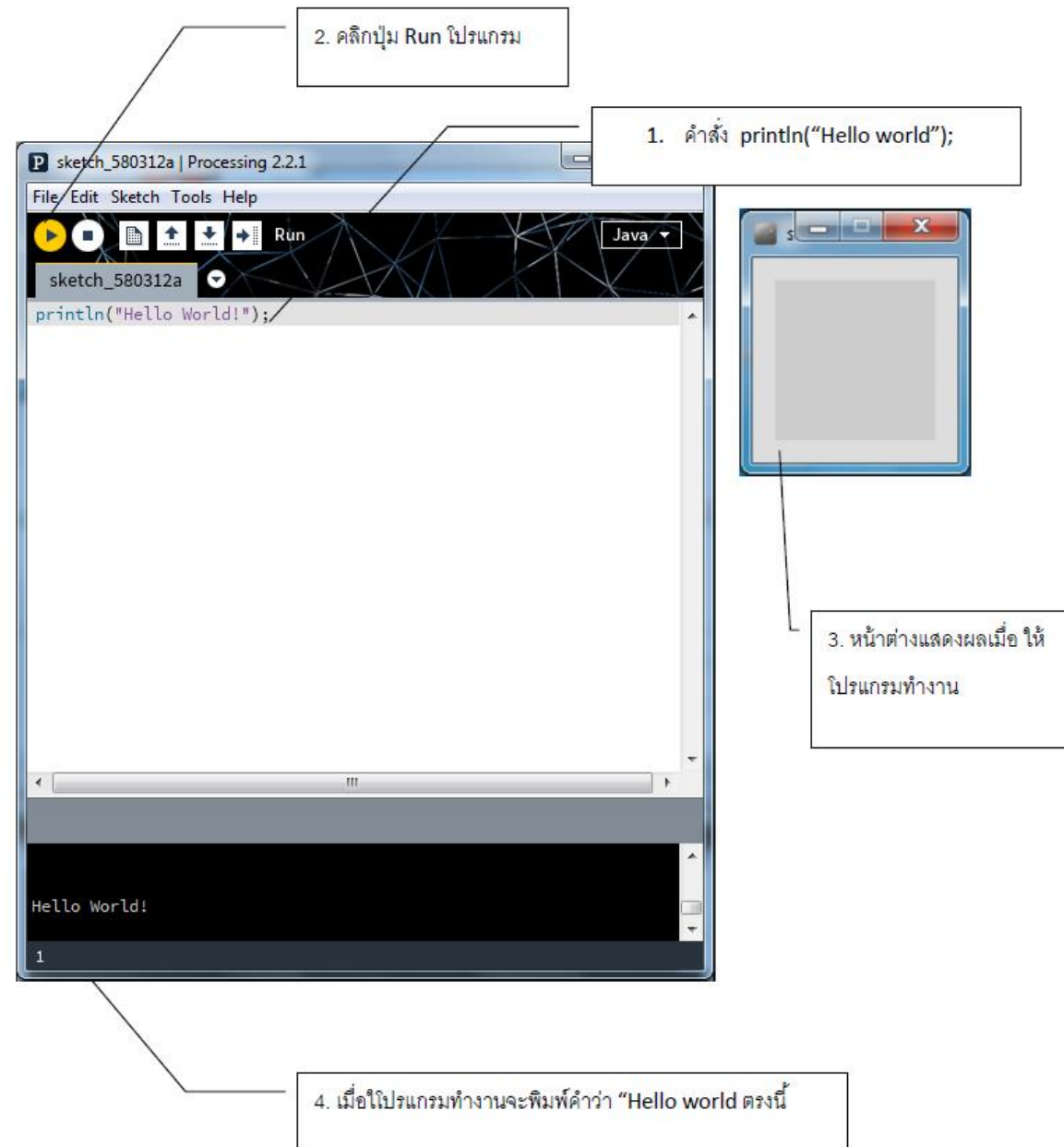
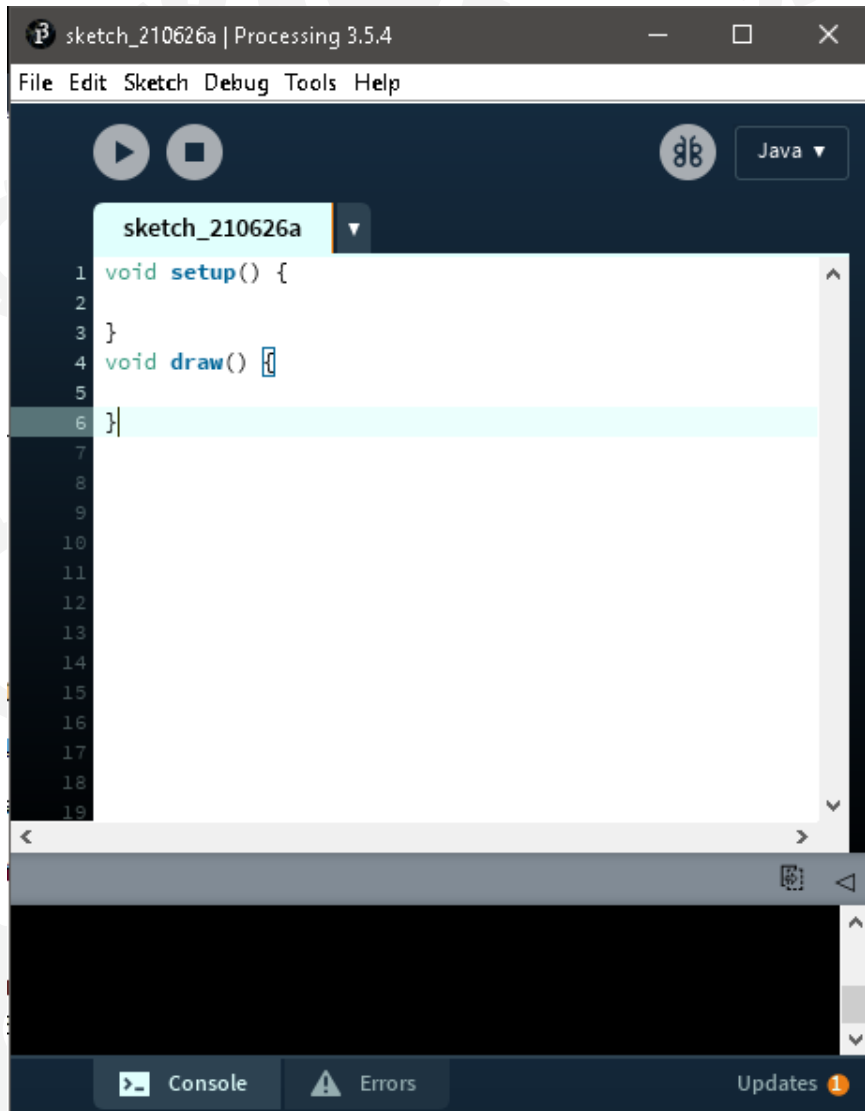
Mac OS X

Windows 32-bit

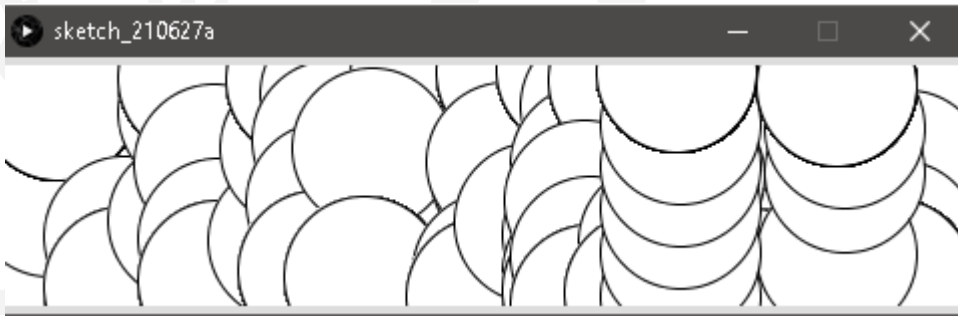
Processing เป็นซอฟต์แวร์ระบบเปิด เหมาะสำหรับผู้ที่ต้องการพัฒนาโปรแกรมเกี่ยวกับการสร้างภาพเคลื่อนไหวและการมีปฏิสัมพันธ์ สำหรับผู้ที่เคยใช้ชุดพัฒนาโปรแกรม **Arduino** เมื่อเห็นรูปร่างหน้าตาการอินเตอร์เฟซของ **Processing** แล้วจะรู้สึกคุ้นเคยมาก ด้วย เพราะเหมือนกันนั่นเอง อีกทั้ง **Processing** และ **Arduino** ใช้หลักการในการเขียนโปรแกรมเหมือนกัน โดยมีพื้นฐานมาจากภาษา **C/C++** รวมถึงการติดตั้งชุดพัฒนาก็เหมือนกันด้วย

สถาบันพัฒนาบุคลากรสาขาเทคโนโลยีการผลิตอัตโนมัติและหุ่นยนต์ กรมพัฒนาฝีมือแรงงาน

การใช้งานเครื่องมือเบื้องต้น



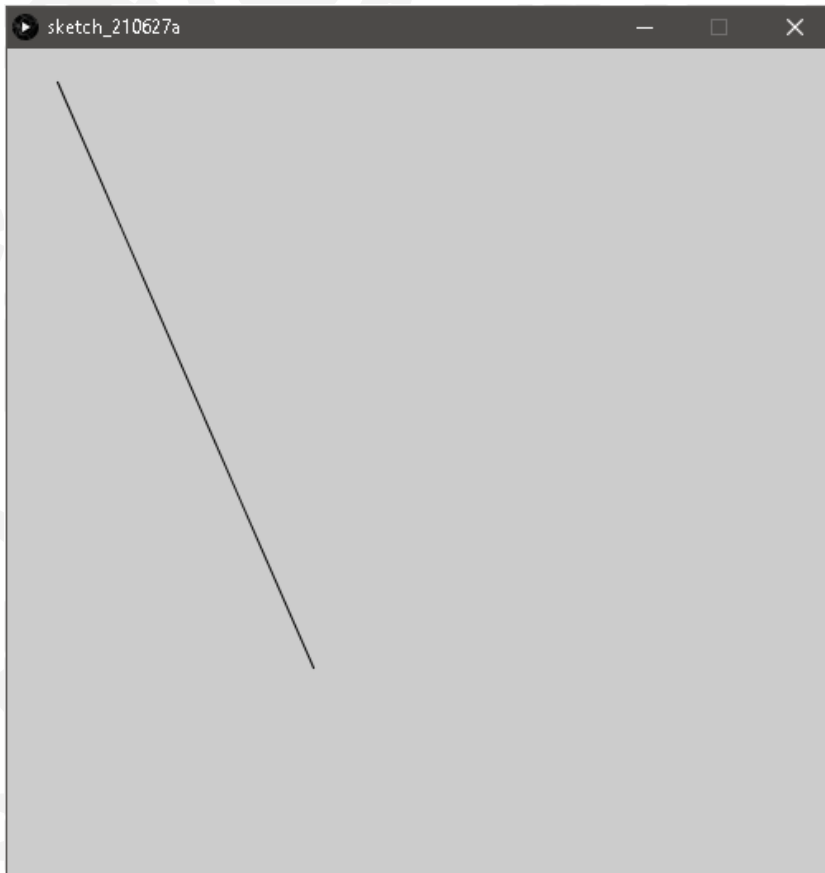
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 void setup() {
2   size(480, 120);
3   smooth();
4 }
5
6 void draw() {
7   if (mousePressed) {
8     fill(0);
9   } else {
10    fill(255);
11  }
12  ellipse(mouseX, mouseY, 80, 80);
13 }
```

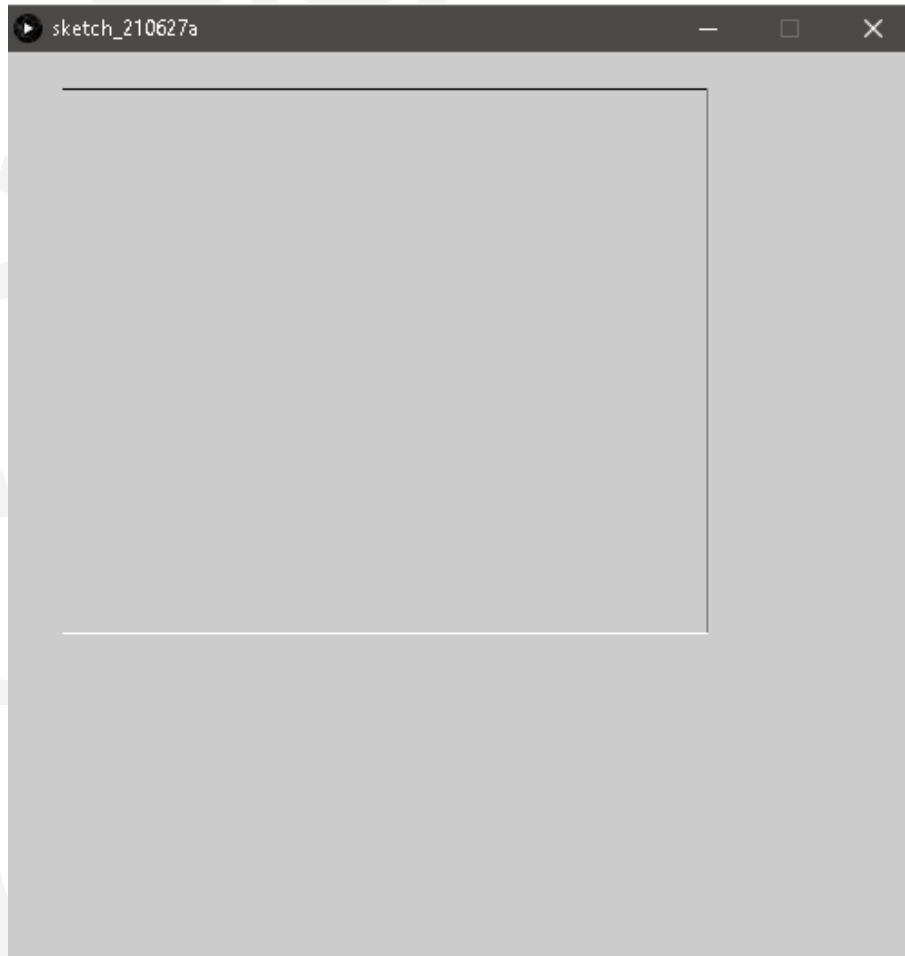
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 void setup() {
2   size(500,500);
3   line(30, 20, 185, 375);
4 }
5
6 void draw() {
7
8 }
```

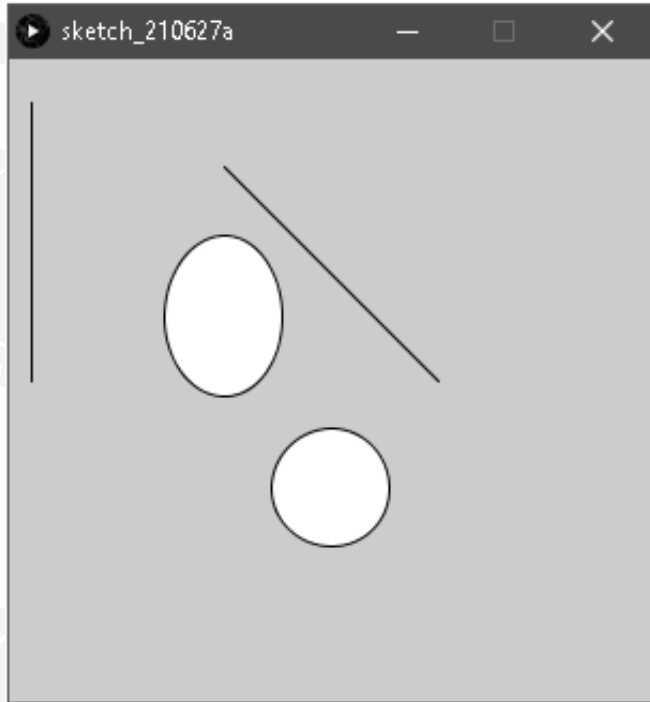
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 void setup() {
2   size(500,500);
3   line(30, 20, 385, 20);
4   stroke(126);
5   line(385, 20, 385, 320);
6   stroke(255);
7   line(385, 320, 30, 320);
8 }
9
10 void draw() {
11
12 }
```

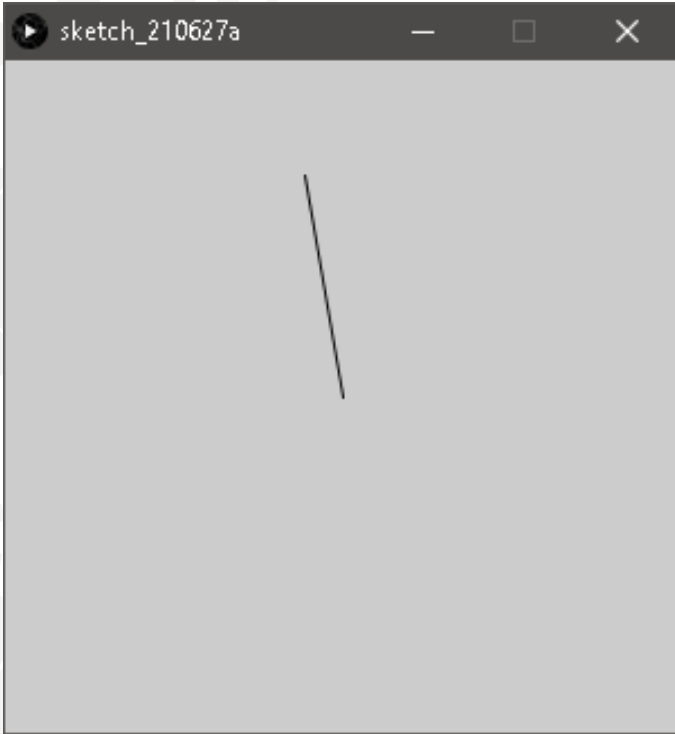
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 void setup()
2 {
3   size(300, 300);
4   background(204);
5   line(10,20,10,150);
6   line(100,50,200,150);
7   ellipse(100, 120, 55, 75);
8   ellipse(150, 200, 55, 55);
9 }
10 void draw()
11 {
12
13 }
```

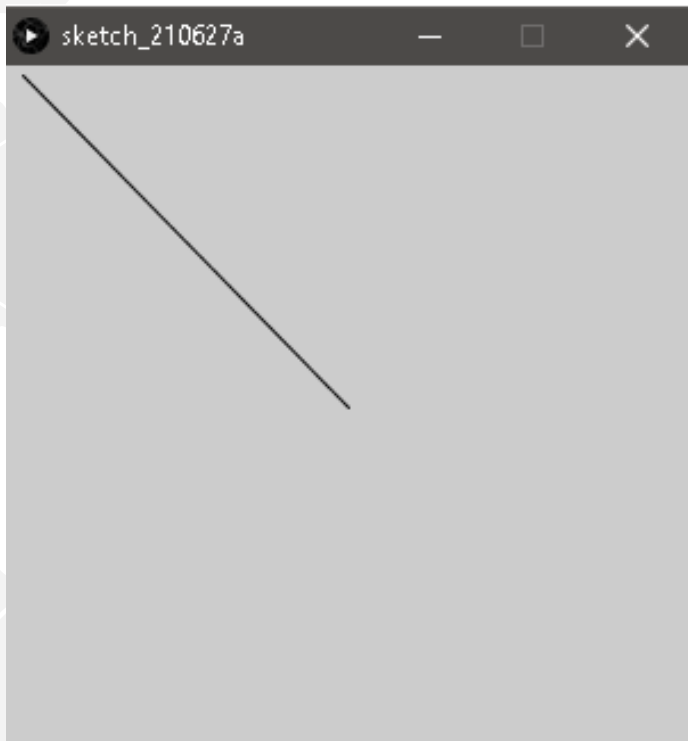
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 float r = 100;
2 float x,y;
3 float d = 0;
4 void setup()
5 {
6   size(300, 300);
7   loop();
8 }
9 void draw()
10 {
11   background(204);
12   x = width/2 + r*cos(d);
13   y = height/2 + r*sin(d);
14   line(width/2, height/2, x, y);
15   delay(10);
16   d = d+0.01;
17   if(d >= TWO_PI)
18   {
19     d = 0;
20   }
21 }
```

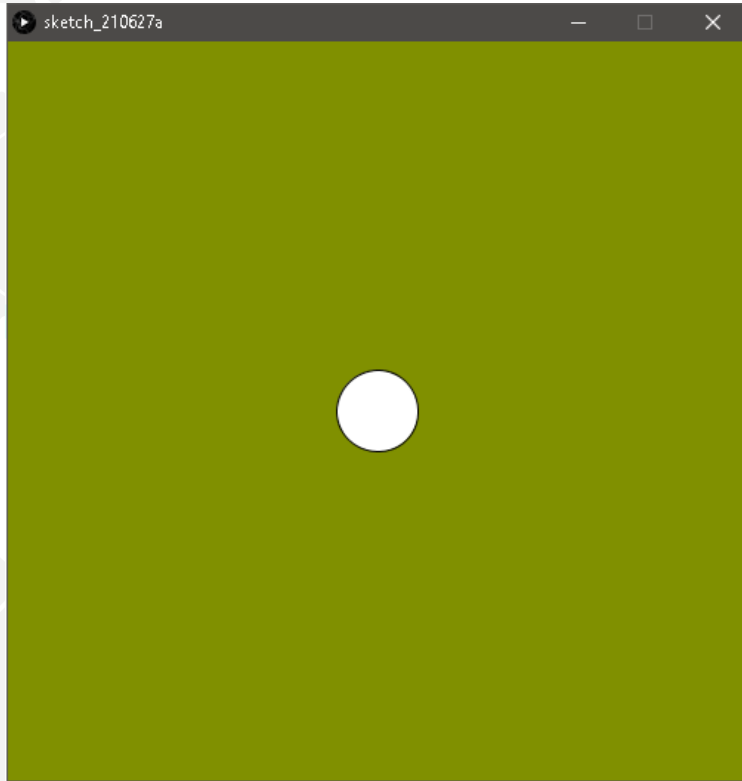
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 void setup()
2 {
3   size(300, 300);
4 }
5 void draw()
6 {
7   background(204);
8   line(150, 150, mouseX, mouseY);
9 }
10
```

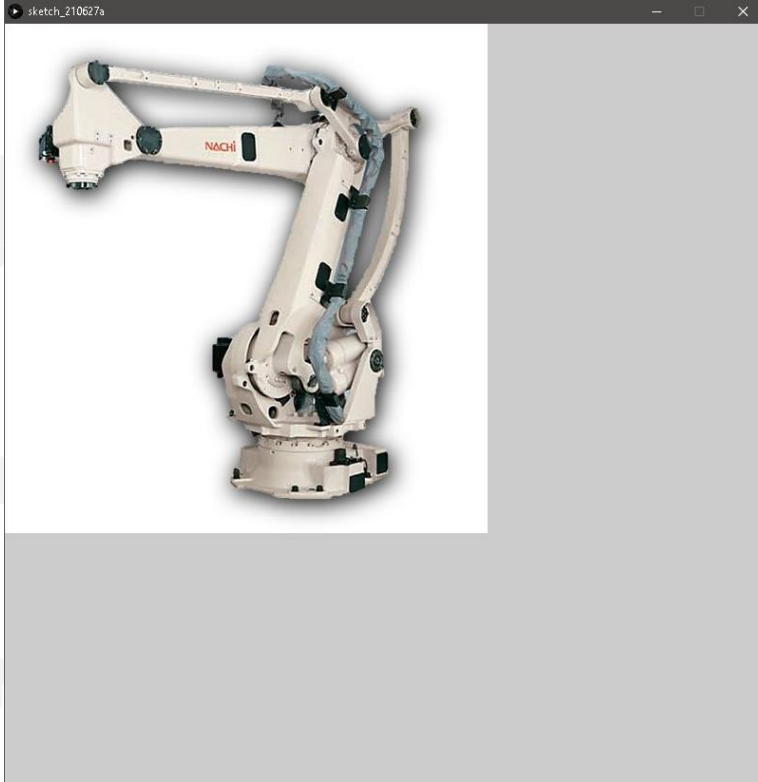
ตัวอย่างการเขียนรูป



Example Code:

```
sketch_210627a
1 int x,y;
2 void setup() {
3   size(500,500);
4   x = width/2;
5   y = height/2;
6 }
7
8 void draw() {
9   background(#809000);
10  ellipse(x, y, 55, 55);
11  if (mousePressed == true)
12  {
13    x = mouseX;
14    y = mouseY;
15  }
16 }
```

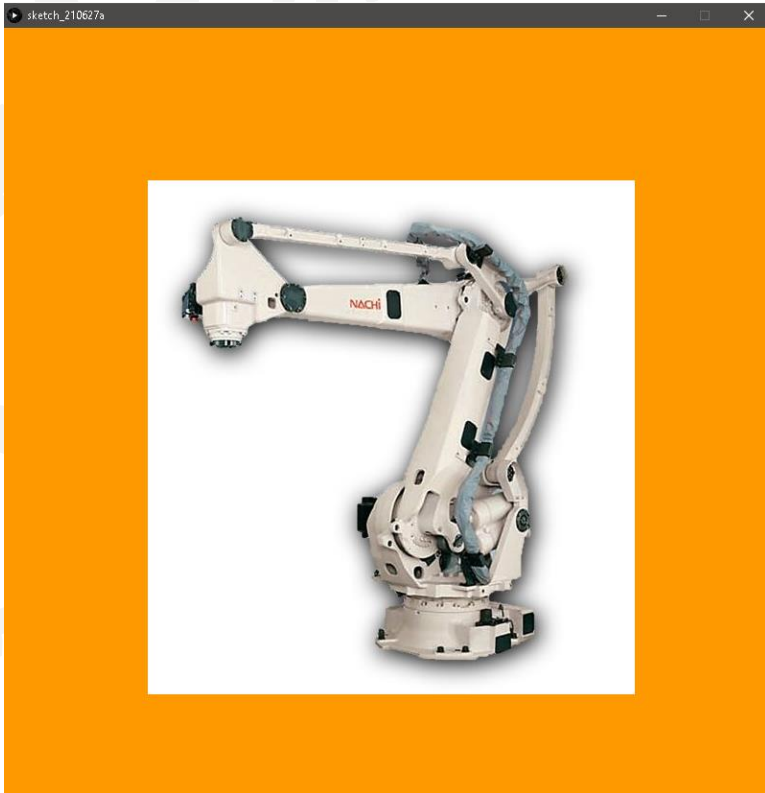
ตัวอย่างการเปิดไฟล์ภาพ



Example Code:

```
sketch_210627a
1 PImage img;
2
3 void setup() {
4   size(800,800);
5
6   img = loadImage("C:/Users/User/Pictures/LP130 (2).jpg");
7 }
8
9 void draw() {
10  image(img, 0, 0);
11 }
```

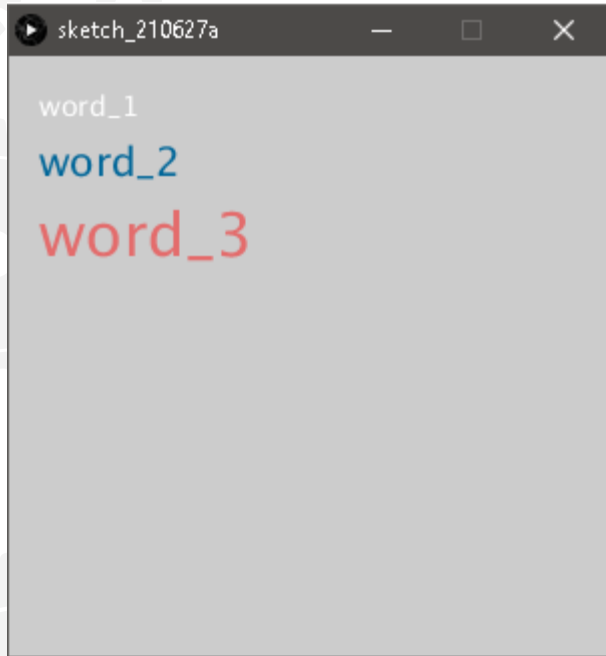
ตัวอย่างการเปิดไฟล์ภาพ



Example Code:

```
sketch_210627a
1 PImage img;
2 int x,y;
3 void setup() {
4   size(800,800);
5   // Images must be in the "data" directory to load correctly
6   img = loadImage("C:/Users/User/Pictures/LP130 (2).jpg");
7   x = width/2;
8   y = height/2;
9 }
10
11 void draw() {
12   background(#FF9900);
13   image(img, x, y);
14   if (mousePressed == true)
15   {
16     x = mouseX;
17     y = mouseY;
18   }
19 }
```

ตัวอย่างการพิมพ์ข้อความ



Example Code:

```
sketch_210627a
1 void setup()
2 {
3   size(300, 300);
4   background(204);
5   textSize(14);
6   text("word_1", 15, 30);
7
8   textSize(20);
9   fill(0, 102, 153);
10  text("word_2", 15, 60);
11
12  textSize(30);
13  fill(255, 0, 0, 120);
14  text("word_3", 15, 100);
15 }
16 void draw()
17 {
18
19 }
```

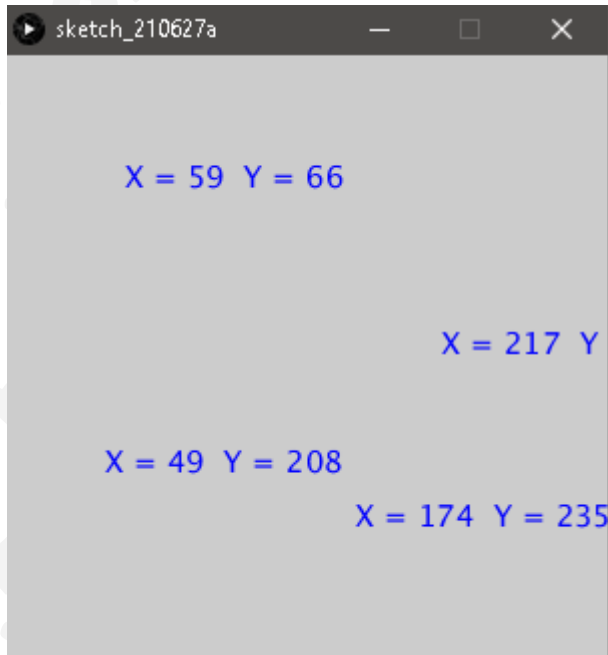
ตัวอย่างการพิมพ์ข้อความ



Example Code:

```
sketch_210627a
1 void setup()
2 {
3   size(300, 300);
4   background(204);
5 }
6 void draw()
7 {
8   textSize(30);
9   fill(255, 0, 0, 120);
10 }
11 void mouseClicked()
12 {
13   text("word", mouseX, mouseY);
14 }
```

ตัวอย่างการรับค่าจากเมาส์



Example Code:

```
sketch_210627a
1 String s;
2 void setup()
3 {
4   size(300, 300);
5   background(204);
6 }
7 void draw()
8 {
9   textSize(15);
10  fill(0, 0, 255);
11 }
12 void mouseClicked()
13 {
14  s = "X = " + str(mouseX) + " Y = " + str(mouseY);
15  text(s, mouseX, mouseY);
16 }
```

ตัวอย่างการสร้างปุ่มกดโดยใช้ ไลบรารี ControlP5



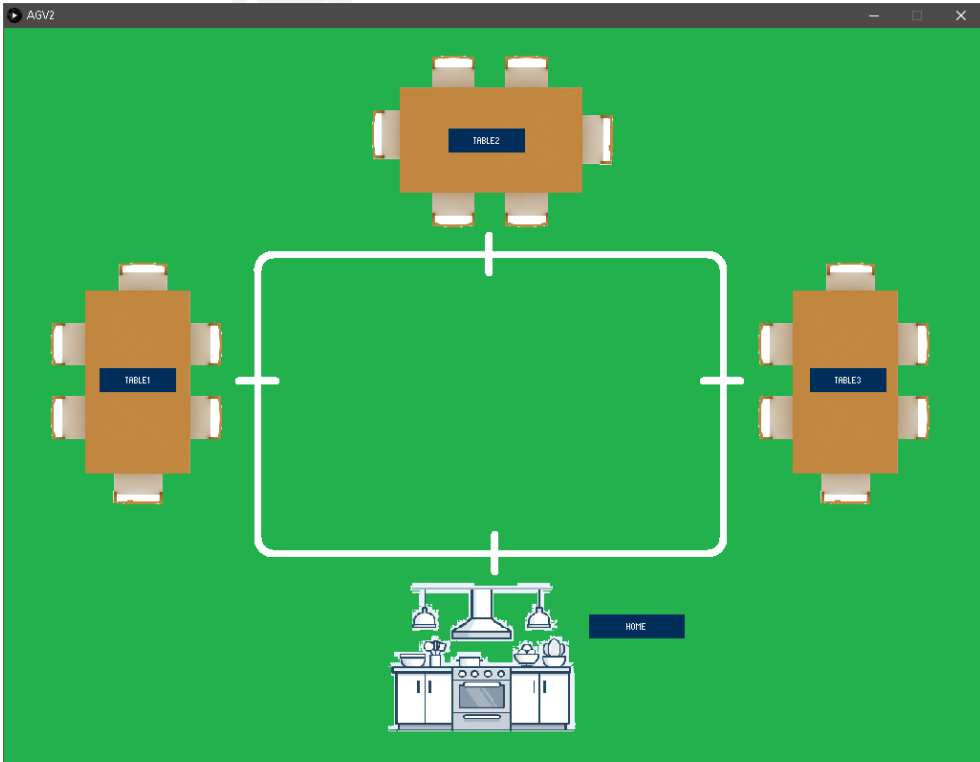
Example Code:

```
AGV1
1 import controlP5.*;
2 ControlP5 cp5;
3 import processing.net.*;
4
5 void setup() {
6   size(800 , 600);
7   cp5 = new ControlP5(this);
8   cp5.addButton("Button1")
9     .setPosition(0,0)
10    .setSize(100,25)
11   ;
12   cp5.addButton("Button2")
13     .setPosition(100,355)
14     .setSize(80,25)
15   ;
16   cp5.addButton("Button3")
17     .setPosition(465,105)
18     .setSize(80,25)
19   ;
20 }
21 void draw() {
22
23 }
24 public void Button1() {
25   println("Button1");
26 }
27 public void Button2() {
28   println("Button2");
29 }
30 public void Button3() {
31   println("Button3");
32 }
```

***การโหลดไลบรารีของโปรแกรม Processing

สถาบันพัฒนาบุคลากรสาขาเทคโนโลยีการผลิตอัตโนมัติและหุ่นยนต์ กรมพัฒนาฝีมือแรงงาน

ทดสอบการเขียนโปรแกรม 1



```
AGV2
1 PFont Mode;
2 PImage Map;
3 PImage ToTable1;
4 PImage ToTable2;
5 PImage ToTable3;
6 import controlP5.*;
7 ControlP5 cp5;
8 import processing.net.*;
9
10 void setup() {
11   size(1024 , 768);
12   Map = loadImage("MAP.png");
13   ToTable1 = loadImage("ToTable1.png");
14   ToTable2 = loadImage("ToTable2.png");
15   ToTable3 = loadImage("ToTable3.png");
16   cp5 = new ControlP5(this);
17
18   cp5.addButton("HOME")
19     .setPosition(612,612)
20     .setSize(100,25)
21   ;
22   cp5.addButton("Table1")
23     .setPosition(100,355)
24     .setSize(80,25)
25   ;
26   cp5.addButton("Table2")
27     .setPosition(465,105)
28     .setSize(80,25)
29   ;
30   cp5.addButton("Table3")
31     .setPosition(843,355)
32     .setSize(80,25)
33   ;
34   ;
35   image(Map, 0, 0);
36   textSize(50);
37   fill(0);
38   text("Command: HOME", 50, 50);
39 }
40 void draw() {
41 }
42 }
43 public void HOME() {
44   println("HOME");
45   image(Map, 0, 0);
46 }
47 public void Table1() {
48   println("Table1");
49   image(ToTable1, 0, 0);
50 }
51 public void Table2() {
52   println("Table2");
53   image(ToTable2, 0, 0);
54 }
55 public void Table3() {
56   println("Table3");
57   image(ToTable3, 0, 0);
58 }
```

ตัวอย่างการรับข้อมูลผ่านพอร์ตอนุกรม (Arduino)

```
Stable Library
=====
Native lib Version = RXTX-2.1-7
Java lib Version   = RXTX-2.1-7
[0] "COM1"
[1] "COM9"
0
1
2
3
7
```

คอนโซล แสดงหมายเลข
พอร์ตที่ใช้งานได้ และค่า
ตัวแปรที่รับได้

```
test1
1 int number = 0; //กำหนดค่าตัวแปร ให้มีค่าเริ่มต้นที่ 0
2 void setup()
3 {
4   Serial.begin(9600); //กำหนดความเร็วการสื่อสารไว้ที่ 9600 บิตต่อวินาที
5 }
6 void loop() //ทำงานต่อเนื่องไม่มีการหยุด
7 {
8   Serial.write(number); //ส่งค่าตัวแปร number ออกจากพอร์ตอนุกรม
9   number++; //เพิ่มค่าตัวแปร
10  if(number>=100) number=0; //ตรวจสอบว่าตัวแปรมีค่าเกิน 99
11  delay(300); // หน่วงเวลาการทำงานไว้ 300 ms
12 }
13 |
```

Example Code:

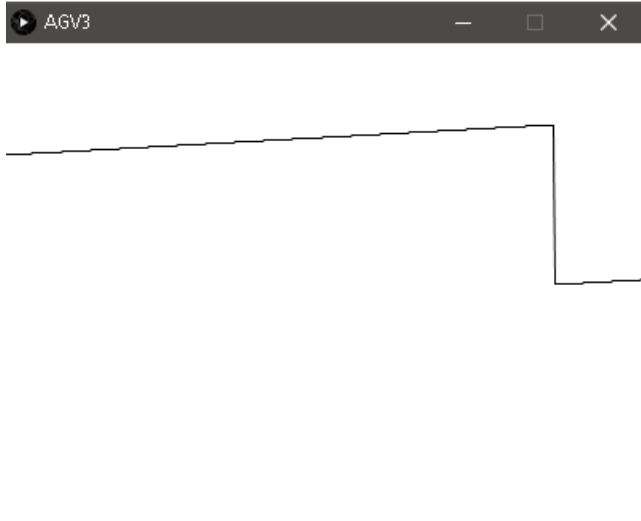
```
sketch_210627a
1 import processing.serial.*;
2
3 Serial myPort;
4
5 void setup()
6 {
7   size(300, 300);
8   println(Serial.list());
9   myPort = new Serial(this, Serial.list()[1], 9600);
10 }
11
12 void draw()
13 {
14   while (myPort.available() > 0)
15   {
16     int inByte = myPort.read();
17     println(inByte);
18   }
19 }
```

ตัวอย่างการรับข้อมูลผ่านพอร์ตอนุกรม (Arduino)

Example Code:

```
AGV3
1 import processing.serial.*;
2 Serial myPort;
3 float[] vals;
4 int zero_line;
5
6 void setup()
7 {
8   size(400,300);
9   smooth();
10  println(Serial.list());
11  myPort = new Serial(this, Serial.list()[0], 9600);
12  vals = new float[width];
13  zero_line = height/2;
14  for (int i = 0; i < vals.length; i++)
15  {
16    vals[i] = 0;    //initial value
17  }
18 }
```

```
19 void draw() {
20   background(255);
21   // Draw lines connecting all points
22   for (int i = 0; i < vals.length-1; i++) {
23     stroke(0);
24     strokeWeight(1);
25     line(i,zero_line -vals[i],i+1,zero_line -vals[i+1]);
26   }
27   // Slide everything down in the array
28   for (int i = 0; i < vals.length-1; i++) {
29     vals[i] = vals[i+1];
30   }
31   // Add a new value
32   while (myPort.available() > 0)
33   {
34     int inByte = myPort.read();
35     println(inByte);
36     vals[vals.length-1] = inByte;
37   }
38 }
```



ตัวอย่างการส่งข้อมูลผ่านพอร์ตอนุกรม (Arduino)

Example Code:

```
AGV4
1 import controlP5.*;
2 ControlP5 cp5;
3 import processing.net.*;
4 import processing.serial.*;
5 Serial myPort;
6
7 void setup() {
8   size(1024 , 768);
9   cp5 = new ControlP5(this);
10  println(Serial.list());
11  myPort = new Serial(this, Serial.list()[0], 9600);
12
13  cp5.addButton("Back")
14    .setPosition(465,612)
15    .setSize(150,50)
16    ;
17  cp5.addButton("Left")
18    .setPosition(100,355)
19    .setSize(150,50)
20    ;
21  cp5.addButton("Forward")
22    .setPosition(465,105)
23    .setSize(150,50)
24    ;
```

```
25 cp5.addButton("Right")
26   .setPosition(843,355)
27   .setSize(150,50)
28   ;
29  textSize(30);
30  fill(0);
31  text("Command: Stop", 50, 50);
32 }
33
34 void draw() {
35
36 }
37 public void Forward() {
38   println("Forward");
39   background(255);
40   myPort.write('W');
41   text("Command: Forward", 50, 50);
42 }
43 public void Back() {
44   println("Back");
45   background(255);
```

```
46 myPort.write('S');
47 text("Command: Back", 50, 50);
48 }
49 public void Left() {
50   println("Left");
51   background(255);
52   myPort.write('A');
53   text("Command: Left", 50, 50);
54 }
55 public void Right() {
56   println("Right");
57   background(255);
58   myPort.write('D');
59   text("Command: Right", 50, 50);
60 }
```

คำอธิบายรายวิชา

ศึกษาเกี่ยวกับ การใช้งานเบื้องต้นของหุ่นยนต์ MIR โครงสร้างภายในและการสร้างแผนที่

การควบคุมและใช้งาน MIR

1. ส่วนประกอบ MIR

- อุปกรณ์ระบบไฟฟ้า
- ระบบความปลอดภัย
- ตารางการรับน้ำหนัก
- Diagram การทำงานของ MIR

2. การสร้างแผนที่

- การเชื่อมต่อกับ MIR
- การบันทึกแผนที่โดยใช้เทคโนโลยี LiDAR
- การปรับแต่งแผนที่ที่ได้จากการบันทึก

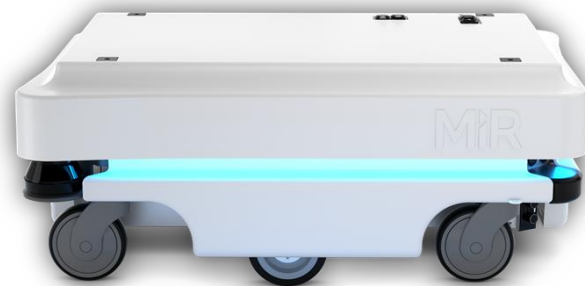
3. การเขียนและการตั้งค่าแผนที่

- การกำหนดพื้นที่ขอบเขตการทำงานของ MIR
- การเพิ่มตำแหน่งการทำงานและกำหนดจุด

4. การสร้างภารกิจ

- การเพิ่มจุดของภารกิจ
- การตั้งค่าของภารกิจ

MiR100



Max. speed

1.5 m/s



Run time

10 hours



Payload

100 kg

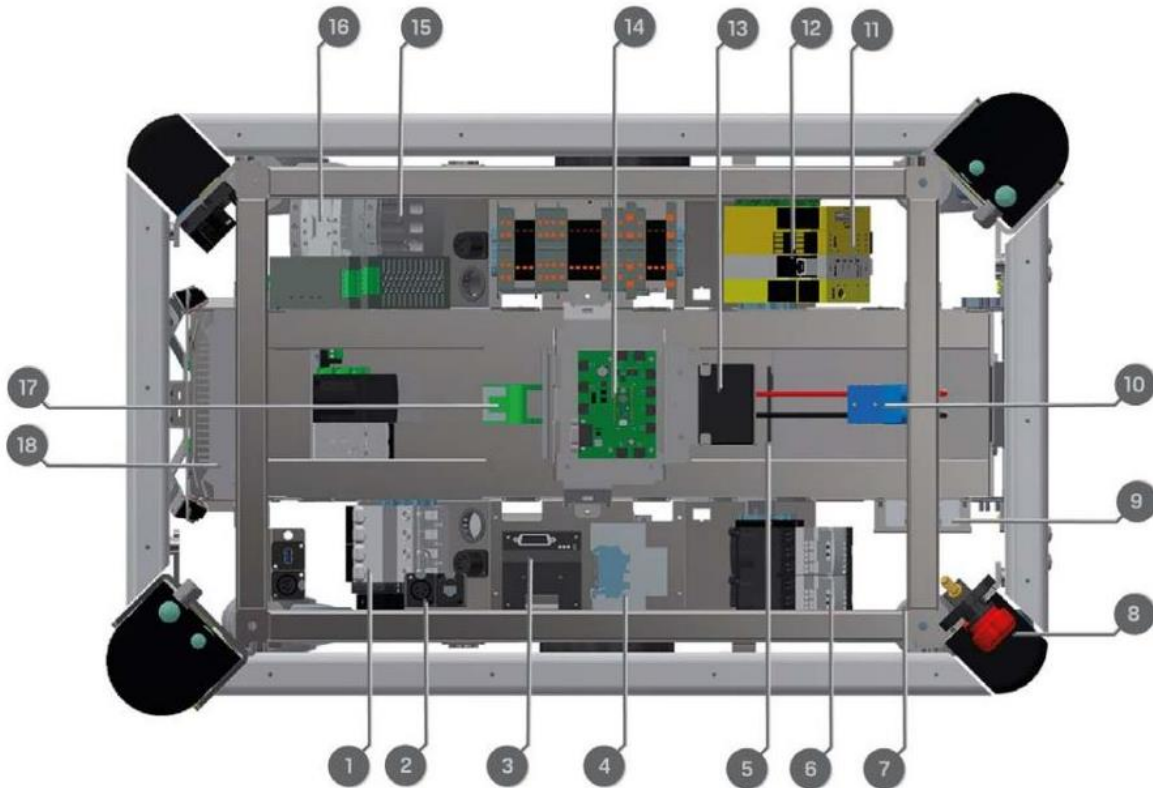


Dimensions

890 mm x 580 mm

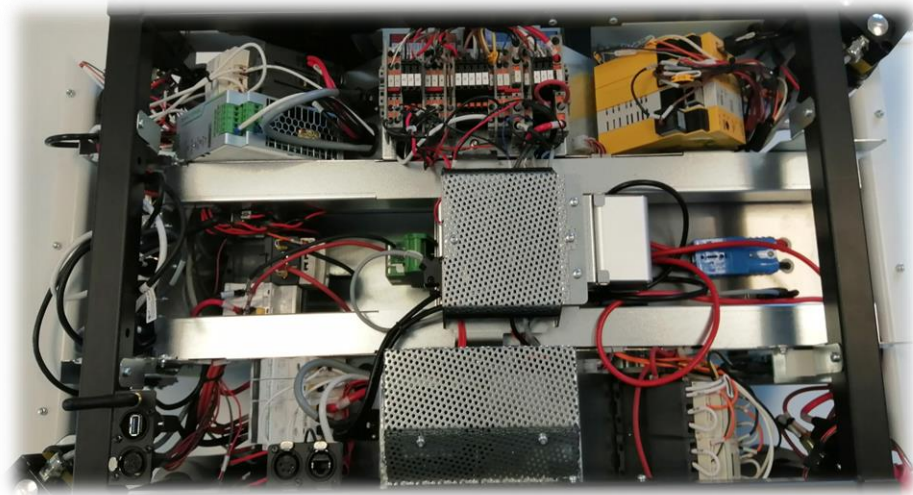
สถาบันพัฒนาบุคลากรสาขาเทคโนโลยีการผลิตอัตโนมัติและหุ่นยนต์ กรมพัฒนาฝีมือแรงงาน

ส่วนประกอบ MIR

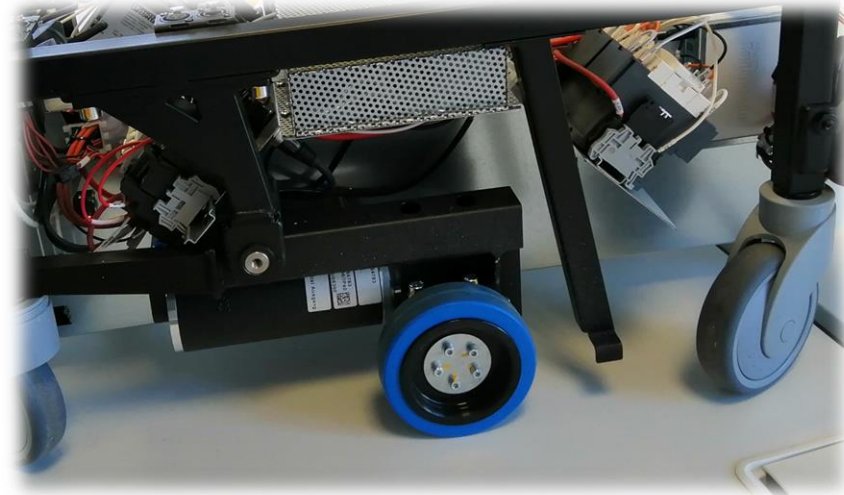


1	Breaker: automatic fuse between battery and components	2	Robot power off relay: releases the latching relay (pos. 17) when the robot is shutting down.
3	Motor controller: manages the two motor drives	4	Brake relay: short circuits motor windings for faster braking
5	Battery connector for extra battery	6	Safe torque off relay (controlled by Safety PLC)
7	CAN bus connection for	8	Battery disconnect switch
9	Router: local network, 2.4 and 5 GHz	10	Battery with connector: main power to the robot
11	Safety PLC	12	Optocoupler: emergency stop signal to motor controller
13	Loudspeaker	14	MiR board: interface board for gyroscope, accelerometer, ultrasound, light, on/off circuit, and CAN bus communication
15	24 V power supply: secures stable voltage for the robot computer and PLC	16	Latching relay: activates the 24 V power supply turning on the robot
17	Transient protection module: provides circuit protection for the power supplies by absorbing voltage spikes from battery or top mounted applications	18	Robot computer

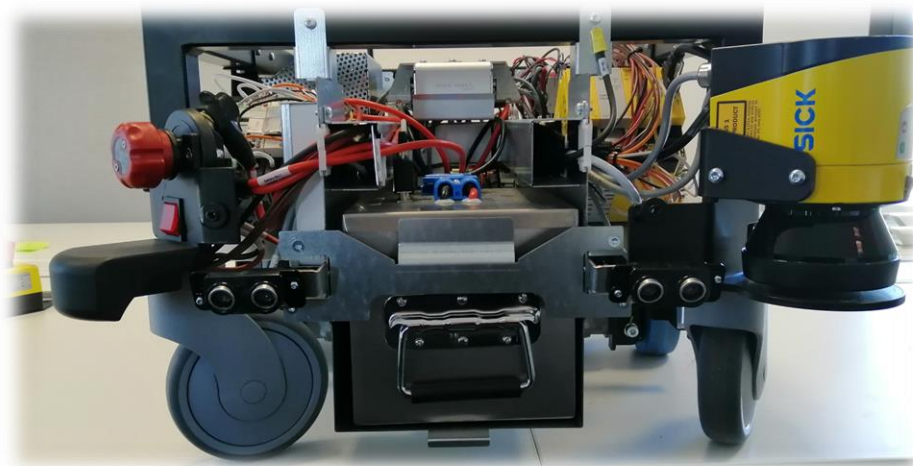
อุปกรณ์โดยรวม MIR



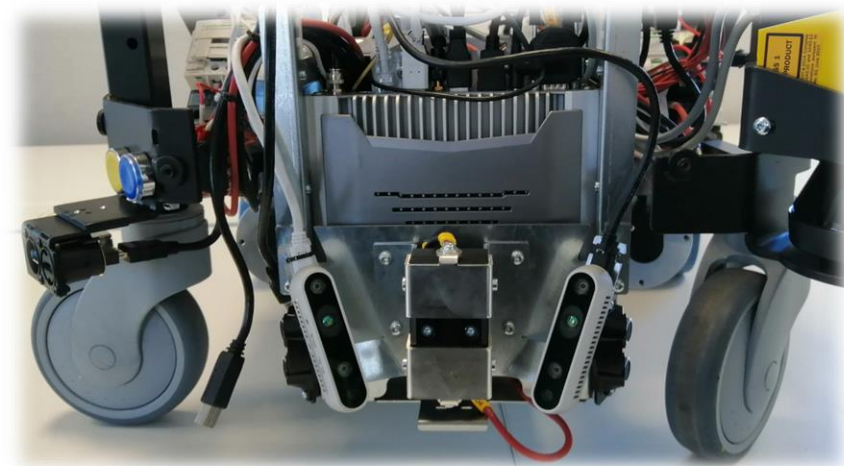
Top View



Side View



Back View



Front View



Roboteq SDC2160MR



NUC



router



Phoenix Contact Quint Power



Schneider Electric LC1D65A



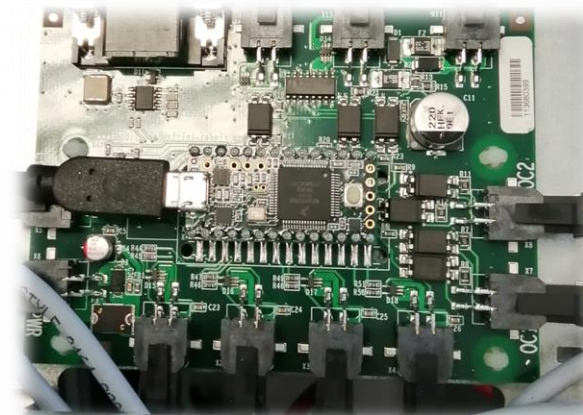
Schneider Electric LC1D18BL



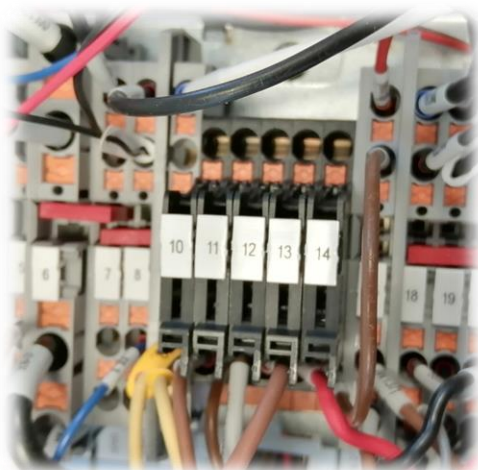
Schneider Electric IC60N



Schneider Electric RSB1A120BD



Teensy LED controller



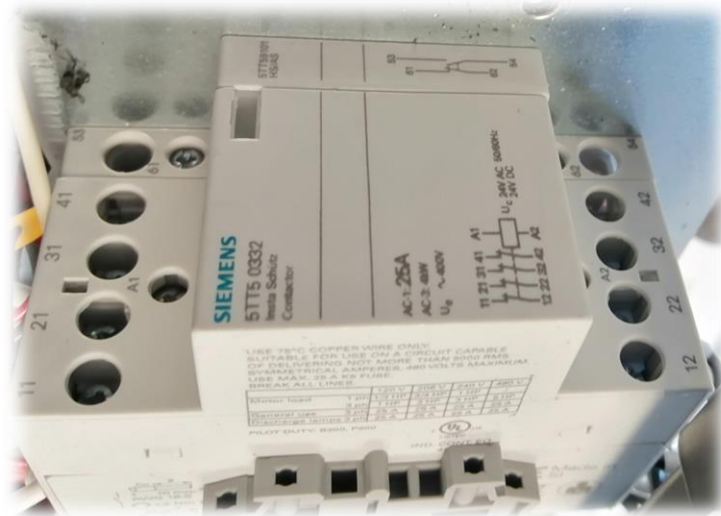
Fuse Board



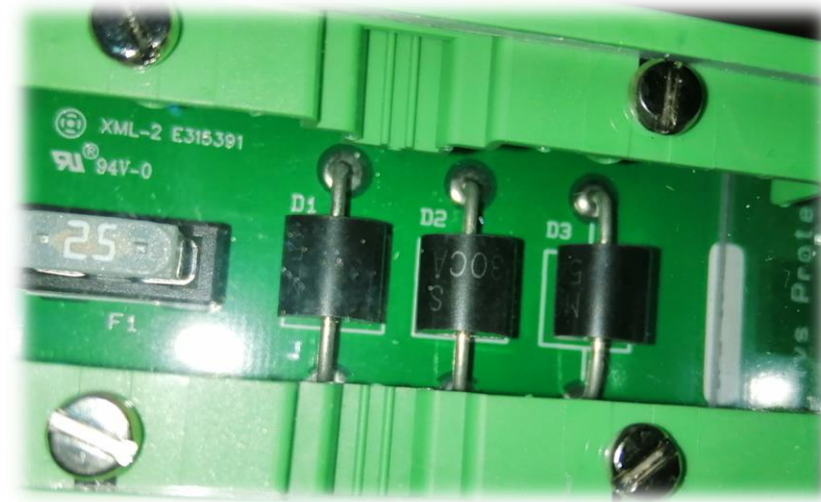
Schneider Electric LC1D093BL



SICK Safety PLC



Siemens 5tt5 0332 Contactor



Thyrector board

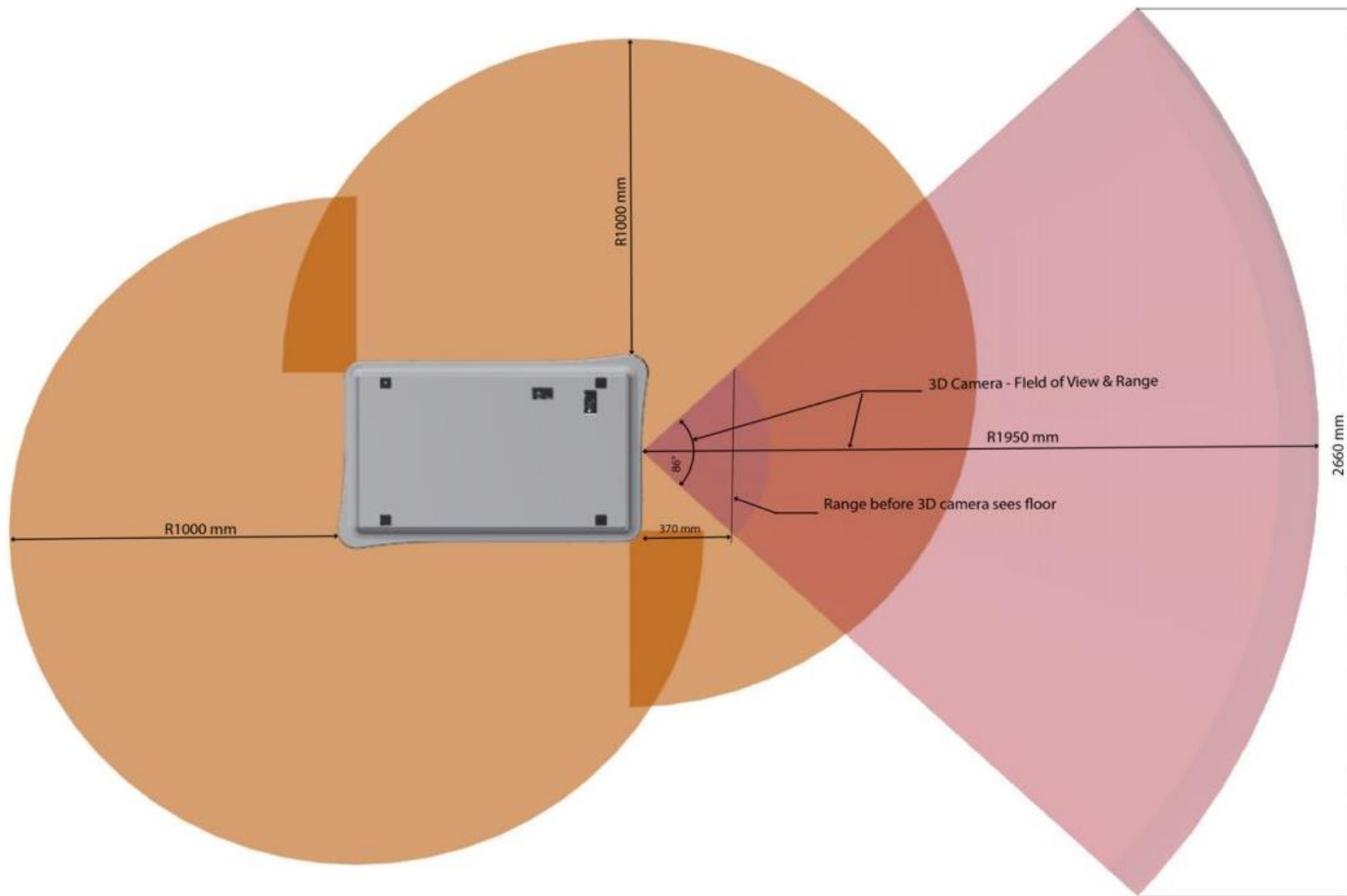


SICK S30B-2011BA



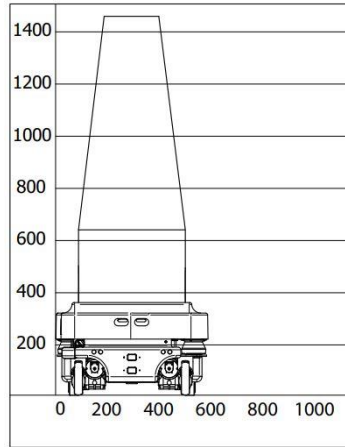
Gefeg-Neckar Motor

ระบบความปลอดภัย MIR Sensor

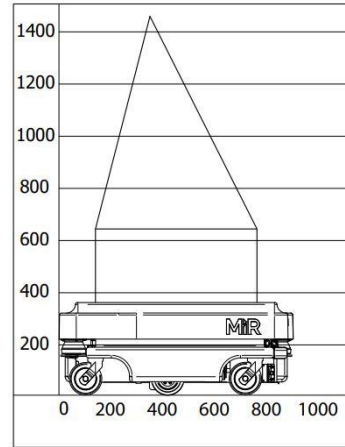


- กล้อง 3 มิติ Intel RealSense™ การตรวจจับวัตถุที่อยู่ด้านหน้าหุ่นยนต์:
ไปข้างหน้า 0-1950 มม.
สูงจากพื้น 50-995 มม.
- เลเซอร์สแกนเนอร์ SICK S300 (ด้านหน้าและด้านหลัง) 360°
- อัลตราโซนิก (4 ชุด) ตรวจสอบความวัตถุโปร่งใส

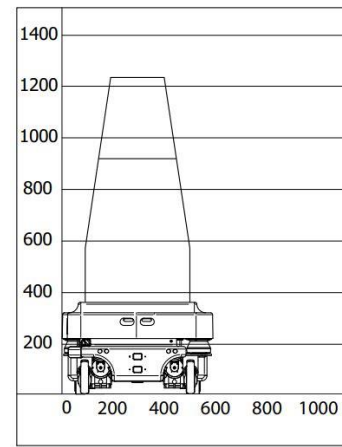
ตารางการรับน้ำหนัก



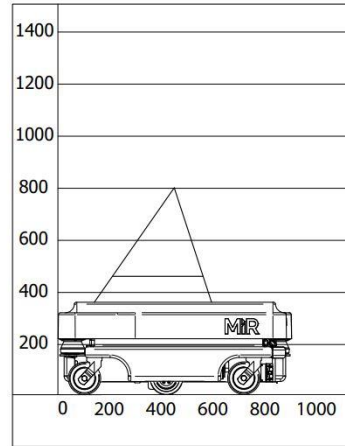
Units: mm
25kg payload



Units: mm
50kg payload



Units: mm
75kg payload



Units: mm
100kg payload

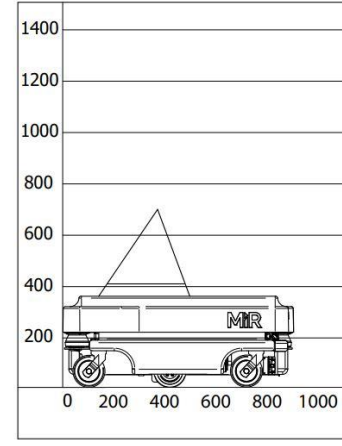
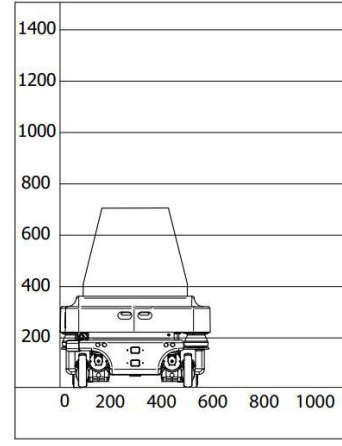
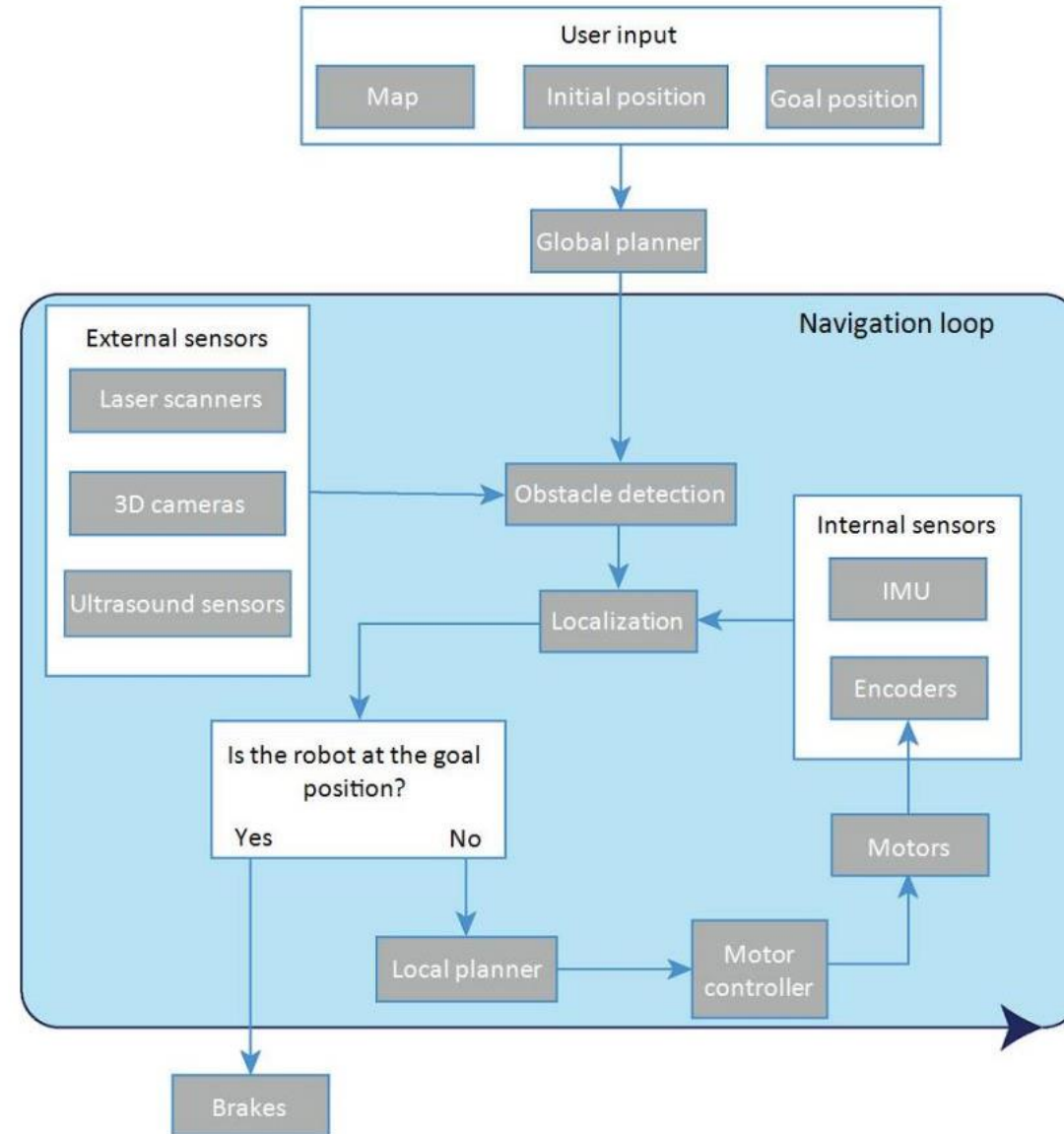


Diagram การทำงานของ MIR



ขั้นตอนการใช้งาน MIR

1. การเริ่มต้นการเชื่อมต่อ
2. การสร้างแผนที่
3. การตรวจสอบความเรียบร้อยของแผนที่
4. การกำหนดพื้นที่และตำแหน่งของการทำงาน
5. การสร้างคำสั่งการทำงาน

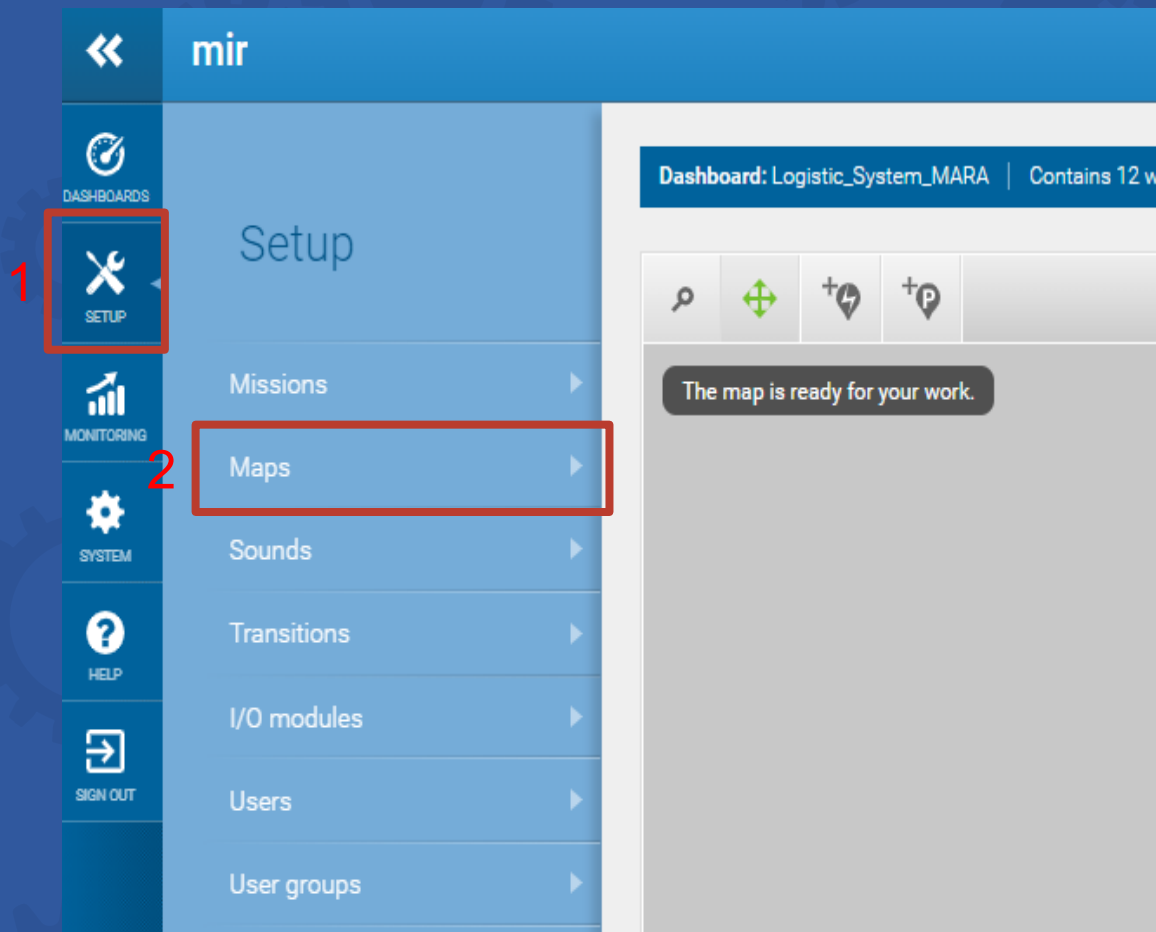


การเริ่มต้นการใช้งาน

The screenshot shows the 'mir' login page. At the top, there's a blue header with the 'mir' logo on the left and a navigation bar on the right that says 'Please choose a way to sign in:' with two tabs: 'Username and password' (which is selected) and 'PIN code'. Below the header, the main content area is white. On the left, there's a section titled 'Sign in by username and password' with instructions: 'Enter your username and password to sign in to the robot.', 'Your username and password should be given to you by either the robot administrator or found in the robot manual.', and 'If you don't have a username and password, please contact the robot administrator.' On the right, there's a red-bordered box containing the login form. It has two input fields: 'Username:' with a placeholder 'Enter your username...' and 'Password:' with a placeholder 'Enter your password...'. Below these fields is a green button with a key icon and the text 'Sign in'.

- เชื่อมต่อ WIFI กับ mir
 - WIFI name: MIR_201703002
 - Password: mirex4you
- เข้าเว็บ mir.com เพื่อล็อกอิน
 - Username : distributor
 - Password : distributor

การสร้างแผนที่



เริ่มต้นการสร้างแผนที่

1. คลิกที่เมนู **SETUP** ด้านซ้าย
2. คลิกที่หัวข้อ **Maps**

การสร้างแผนที่

geAtStation: chargingStationPosit **RUNNING** ✓ ALL OK ▲ 🇺🇸 ENGLISH ▲ 👤 DISTRIBUTOR ▲ 📊 100%

+ Create map Import site Clear filters

Page 1 of 2

	Created by	Functions
		EXPORT
	Distributor	↻ ✎ ✕
	Distributor	✓ ✎ ✕
	Distributor	✓ ✎ ✕

การสร้างแผนที่

Create map
Create a new map. ?

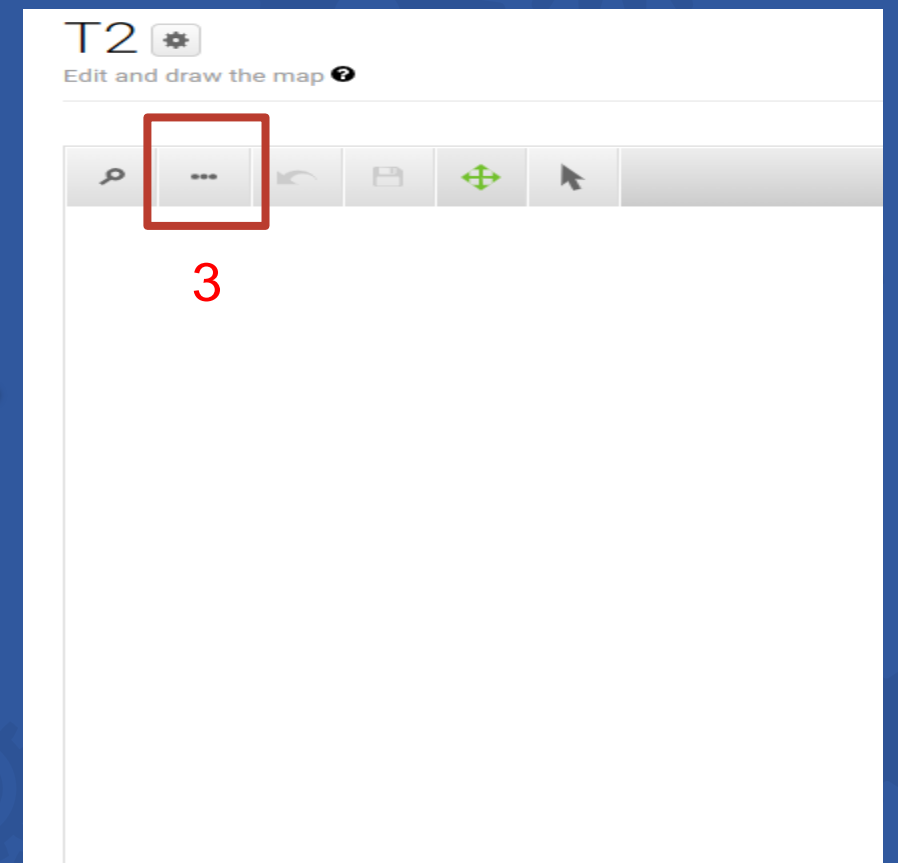
1 Name ?
Enter the map's name...

Site ?
Default site ▼ Create / Edit

2 ✓ Create map ↵ Cancel



1. กำหนดชื่อแผนที่
2. คลิกที่ปุ่มสร้างแผนที่ (Create map)
3. คลิกที่ปุ่ม ... เพื่อไปขั้นตอนถัดไป



Upload, download and record maps



Upload and overwrite

Replace existing map with uploaded map.



Record and overwrite

Replace existing map with new recording of map.



Upload and append

Upload a new map and append it to current map.



Record and append

Record a new map and append it to current map.

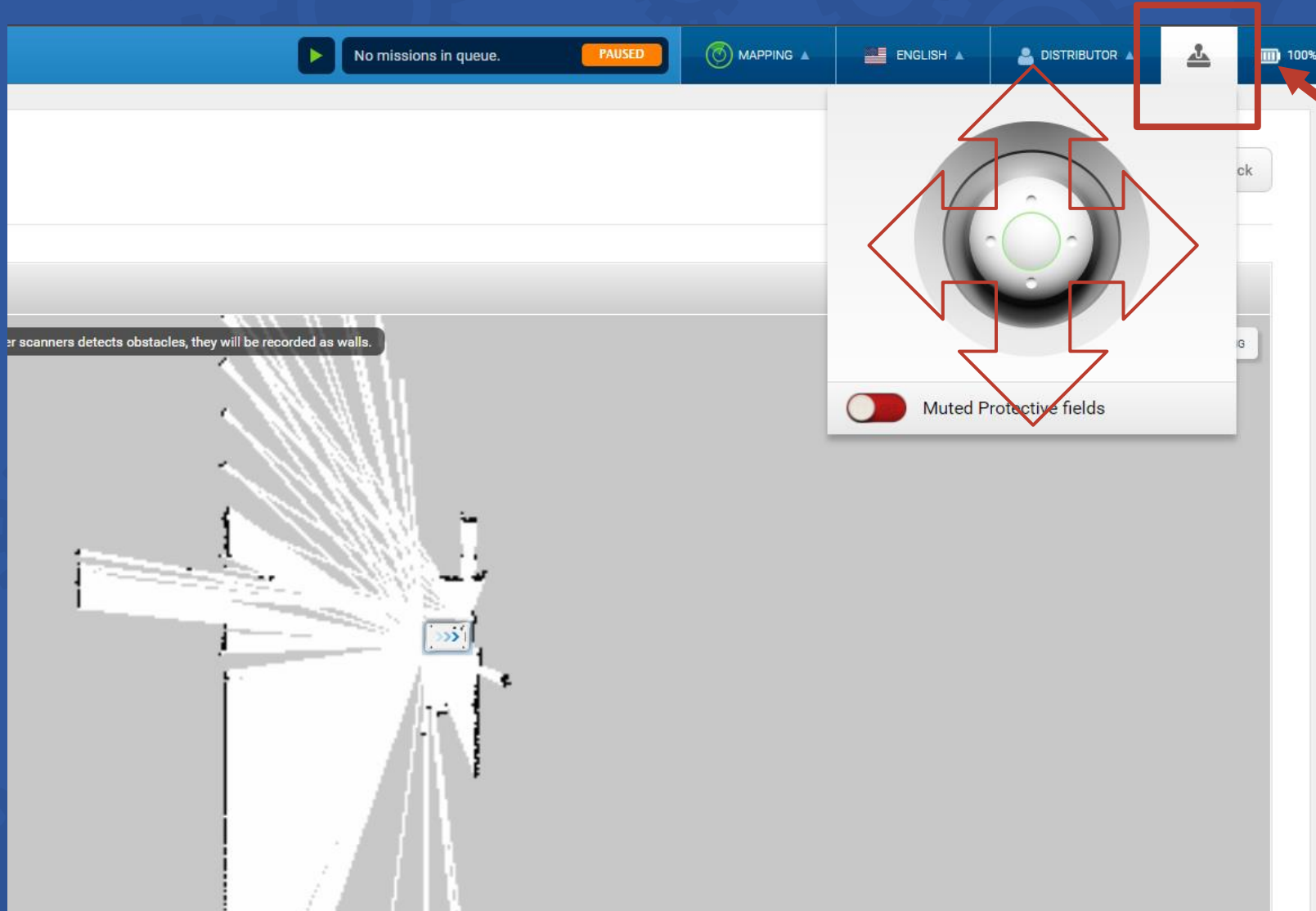


Download map

Download the current map.

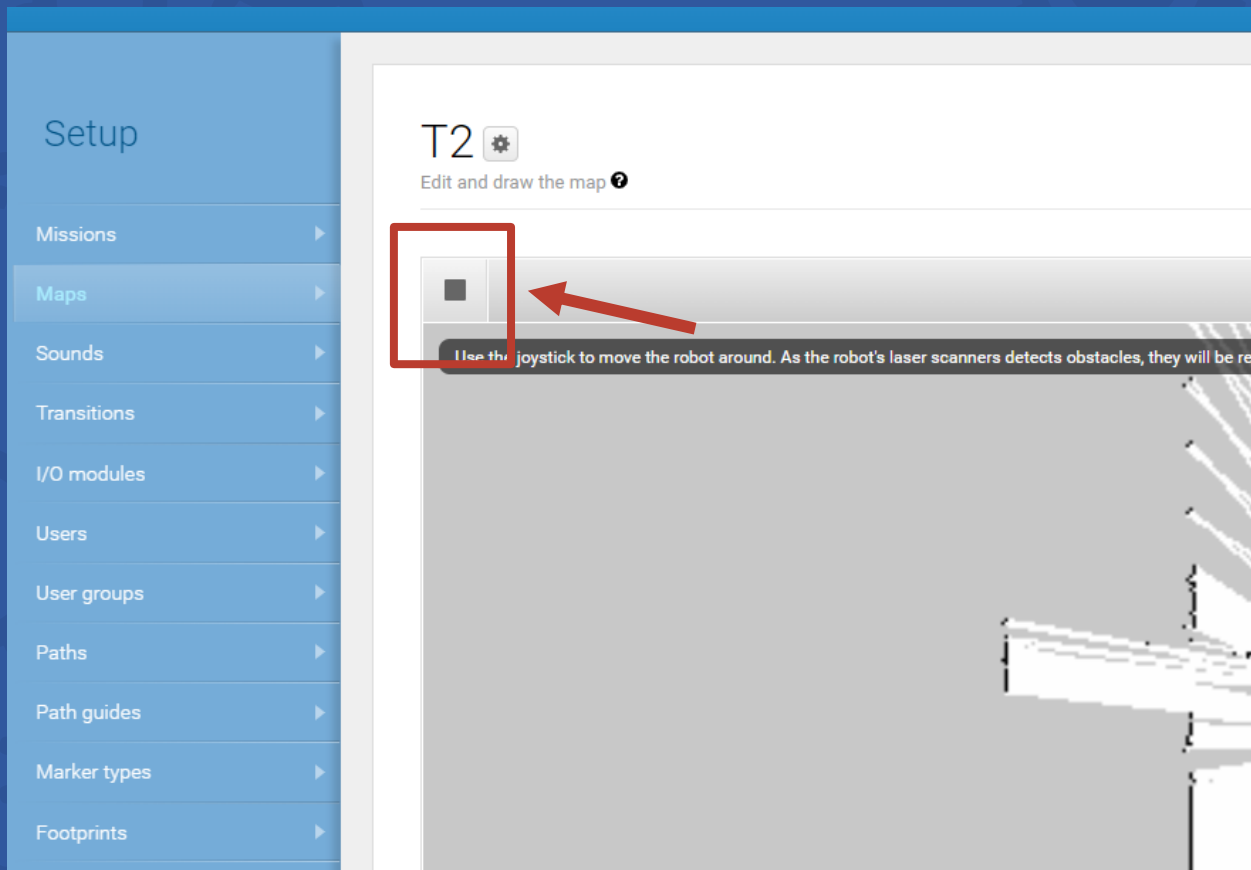
Cancel

การสร้างแผนที่



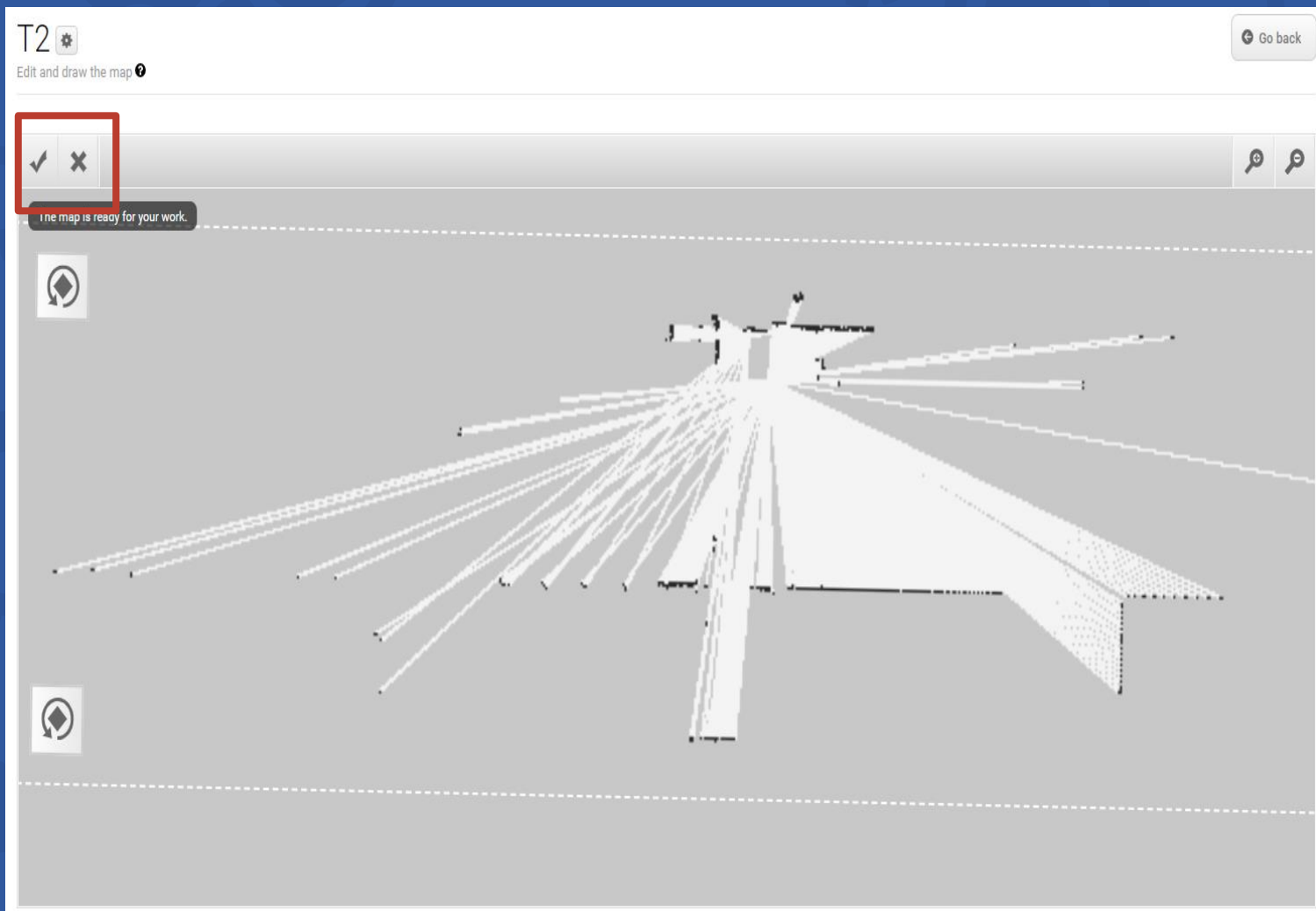
Remote ควบคุมการเคลื่อนที่
เพื่อเก็บข้อมูลและสร้างแผนที่
โดยจะต้องเคลื่อนที่ให้
ครอบคลุมพื้นที่ในการทำงาน
ทั้งหมด

การสร้างแผนที่



คลิกที่ปุ่ม Stop เพื่อจบขั้นตอน
การเก็บข้อมูลพื้นที่การทำงาน

การสร้างแผนที่



คลิกที่เครื่องหมาย / เพื่อ
ยืนยันการจบขั้นตอน

การสร้างแผนที่

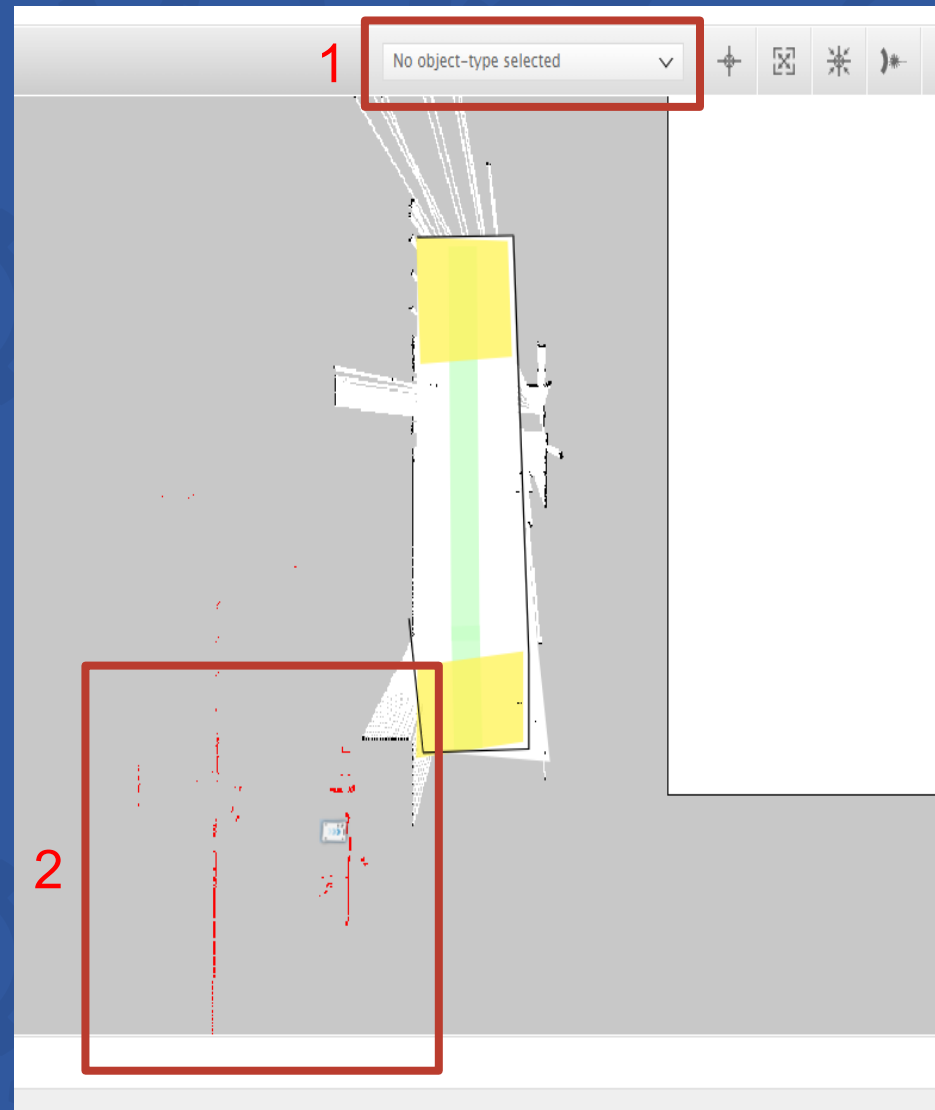
Maps
Create and edit maps. ⓘ

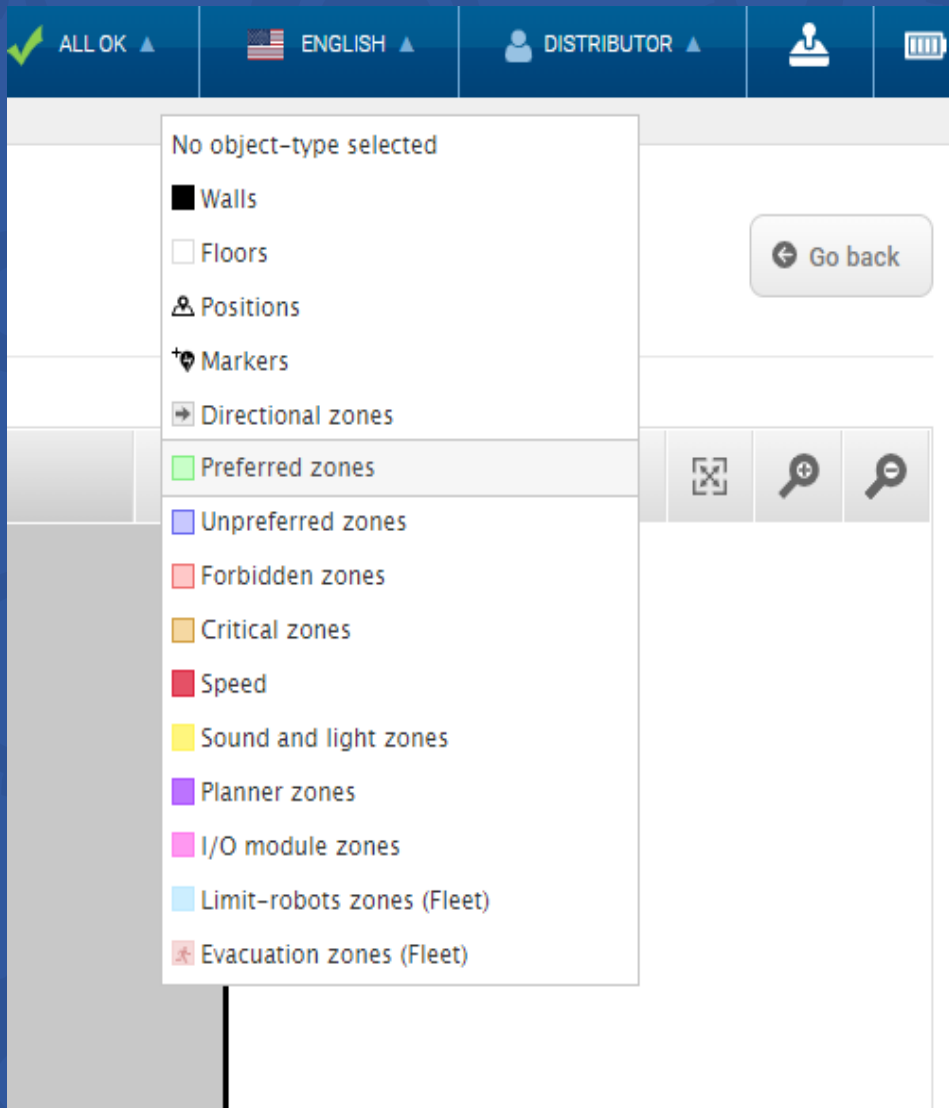
+ Create map Import site Clear filters

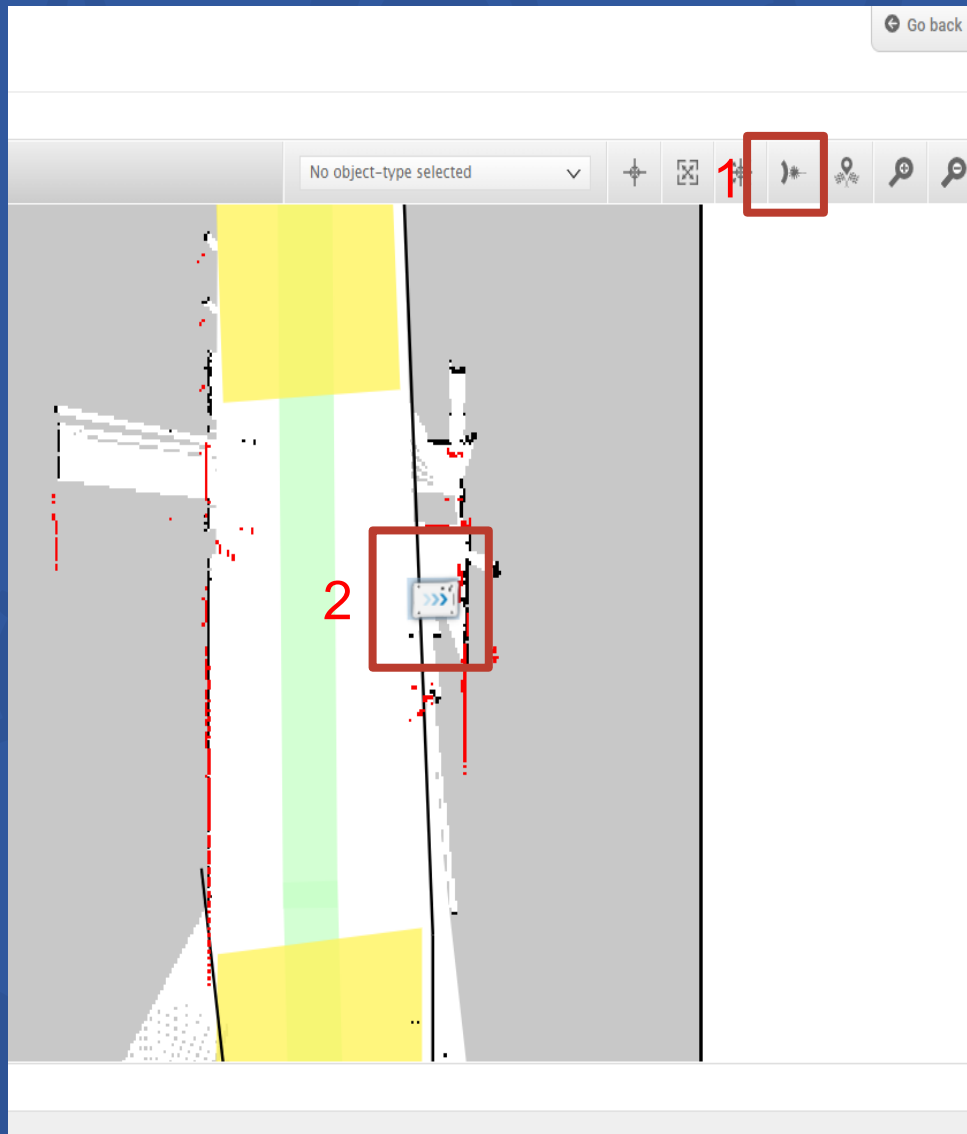
Filter: Write name to filter by... 17 item(s) found Page 1 of 2

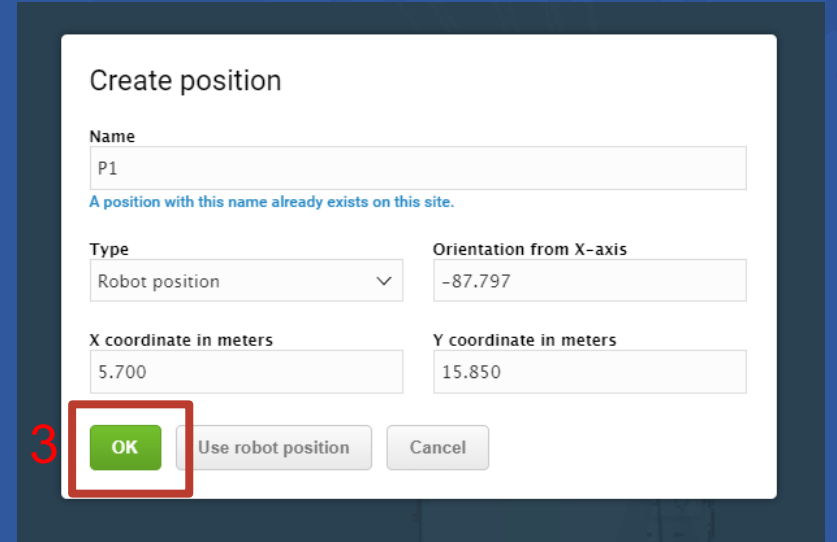
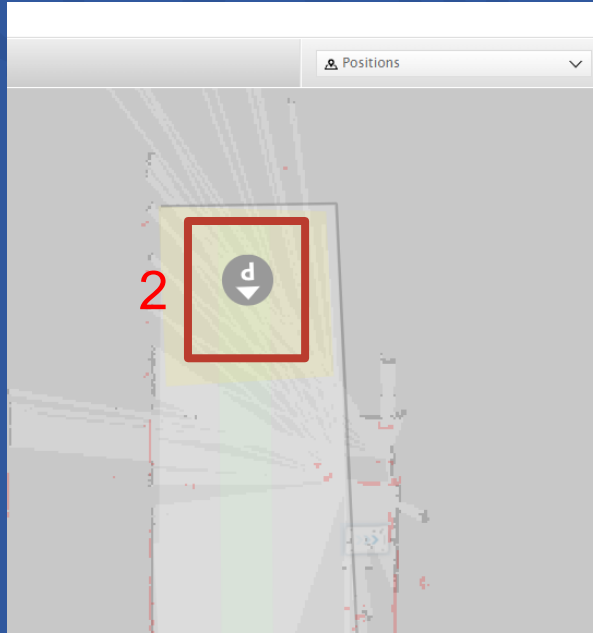
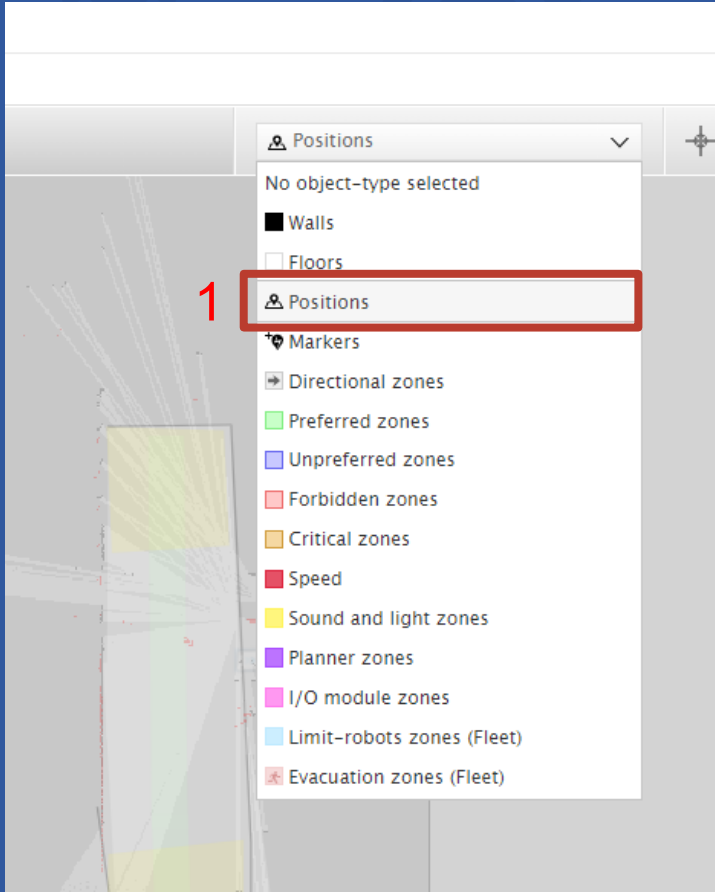
Name	Created by	Functions
Default site		EXPORT
T2 ACTIVE	Distributor	✓ ✎ ✕
Agv final	Distributor	✓ ✎ ✕
Agv tes 131121	Distributor	✓ ✎ ✕
G1.1	Distributor	✓ ✎ ✕
T1	Distributor	✓ ✎ ✕
Tess 2	Distributor	✓ ✎ ✕
g1	Distributor	✓ ✎ ✕
g2	Distributor	✓ ✎ ✕
g3	Distributor	✓ ✎ ✕
g4	Distributor	✓ ✎ ✕

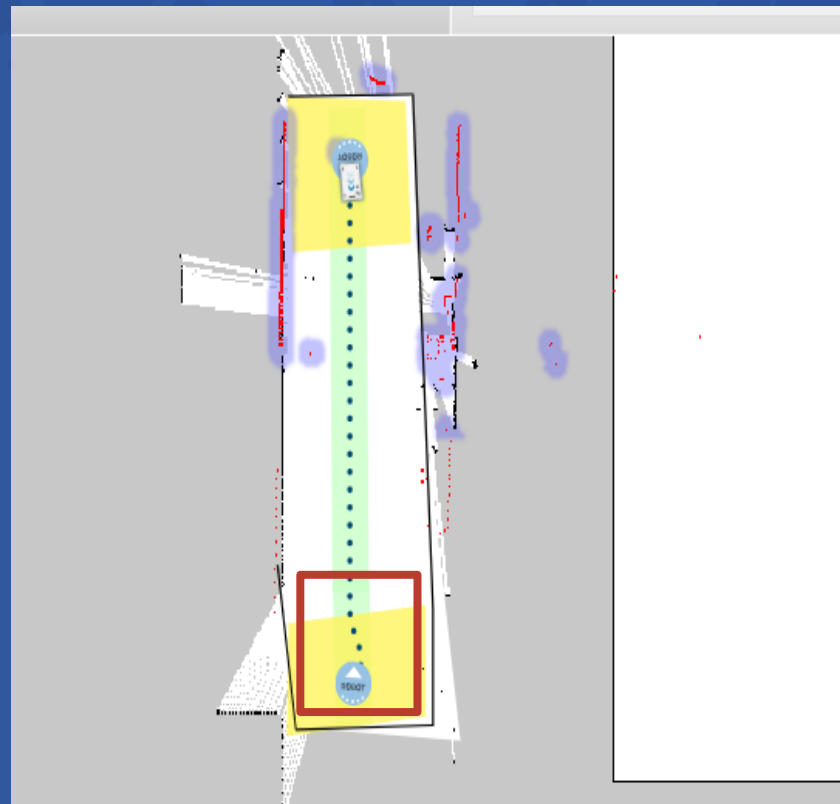
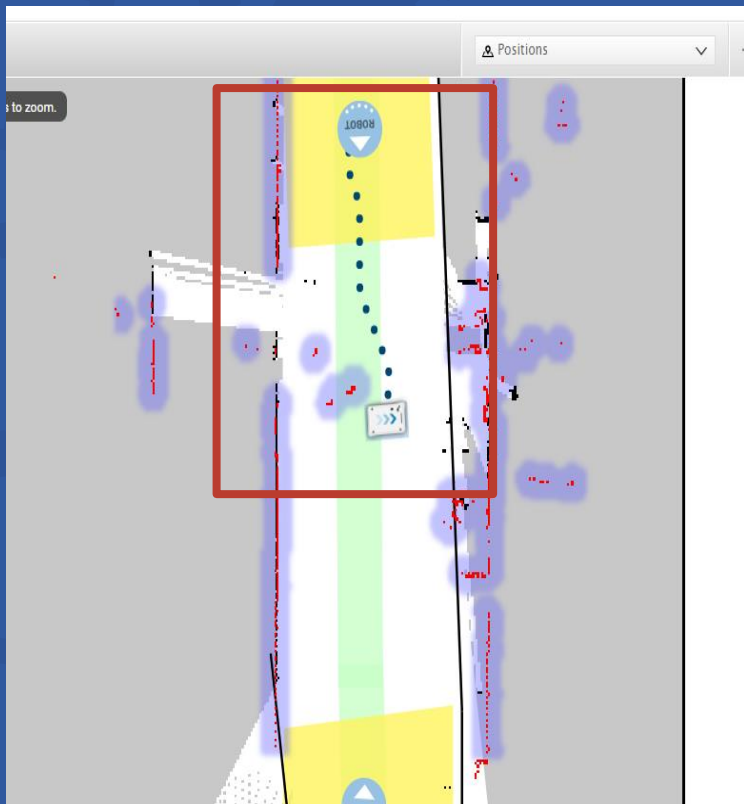
ตารางแสดงชื่อ Maps ต่าง ๆ
ที่ถูกบันทึกไว้ ด้านบนสุดจะ
แสดงชื่อ Maps ล่าสุดที่ทำการ
สร้างขึ้น











Setup

Missions

Maps

Sounds

Transitions

I/O modules

Users

User groups

Paths

Path guides

Marker types

Footprints

Missions

Create and edit missions.

2

+ Create mission

Show missions:

Default site

Create / Edit groups

	agv final Default site			
	Battery Default site			
	g1 Default site			
	G1.1 Default site			
	loop_jo Default site			
	loop_test Default site			
	loop123 Default site			
	m20 Default site			
	sun Default site			
	t10 Default site			

Mission queue

Moving to 'P2' (4.6 meters to goal)

Move : Position: P2

EXECUTING

Create mission

Create a new mission. ?

Name ?

T2

Mission group ?

Missions



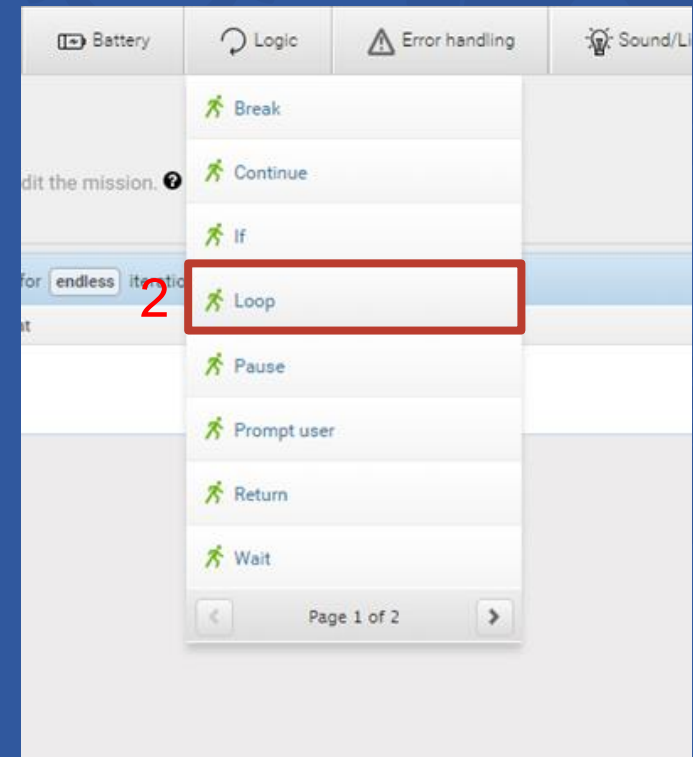
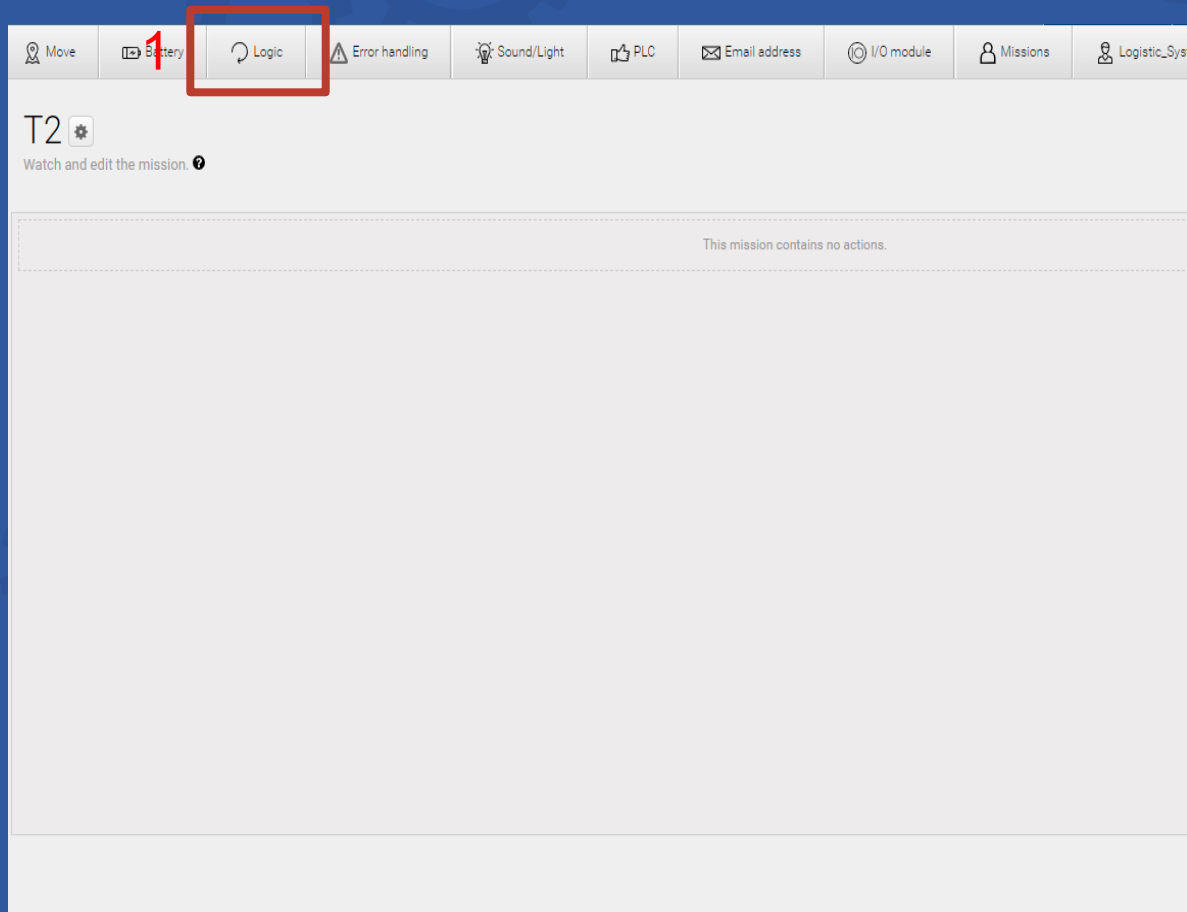
Create / Edit

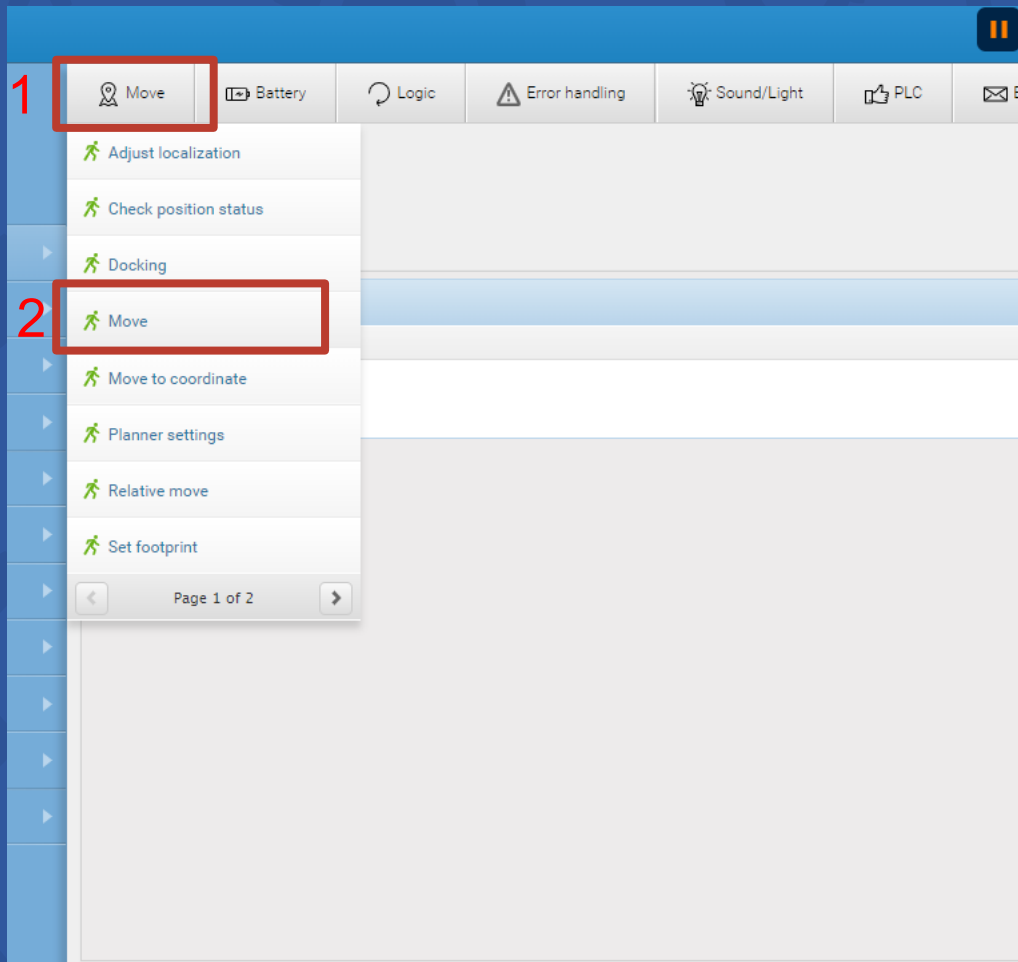
Site ?

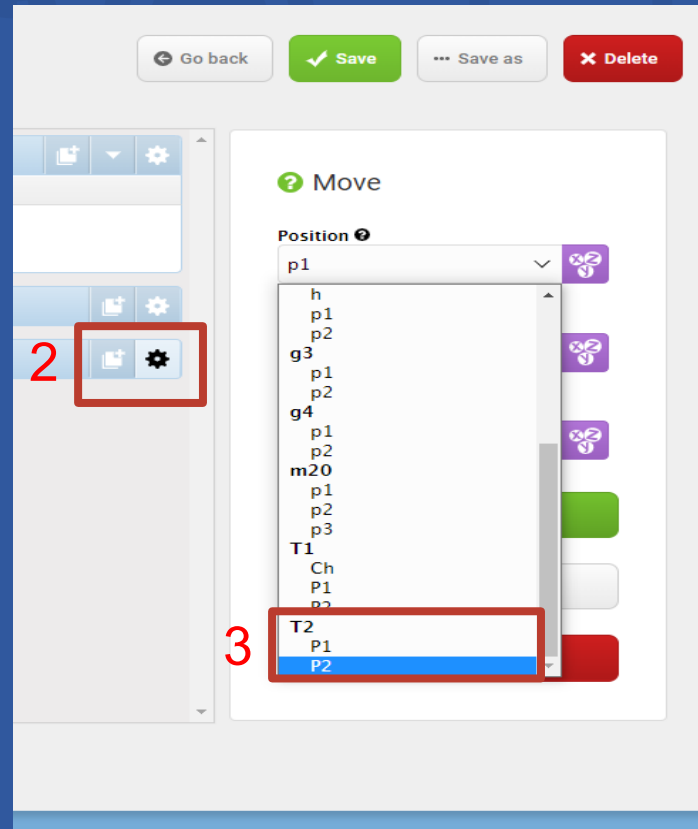
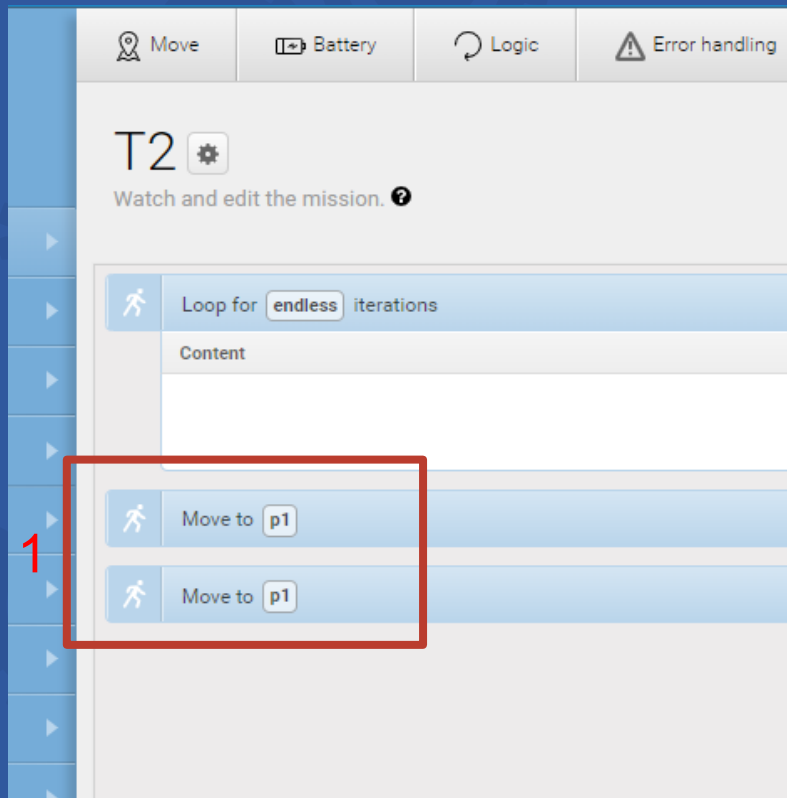
Default site

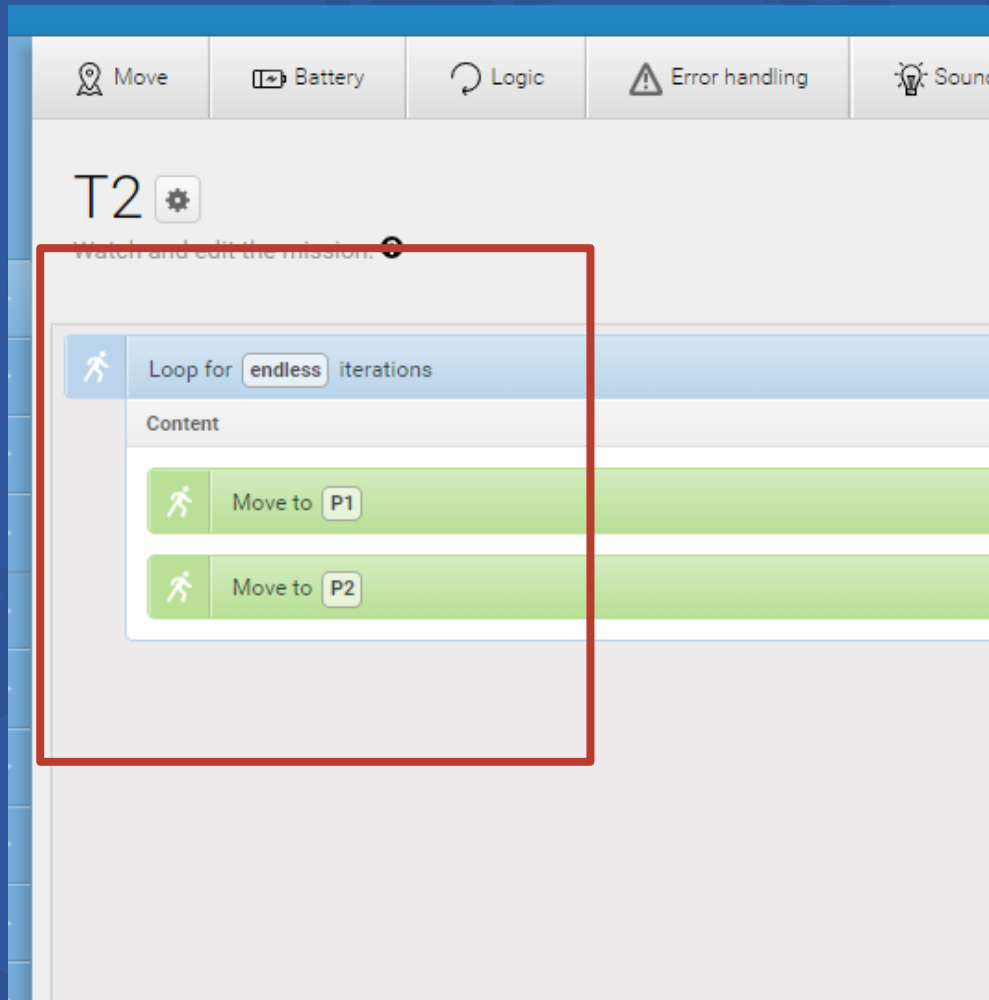
✓ Create mission

↶ Cancel









Show missions:
Default site ▼ Create / Edit groups

	agv final Default site			
	Battery Default site			
	g1 Default site			
	G1.1 Default site			
	loop_jo Default site			
	loop_test Default site			
	loop123 Default site			
	m20 Default site			
	sun Default site			
	t10 Default site			
	T2 Default site			

Mission queue

Moving to 'P1' (11.3 meters to goal)

2 ⊕ T2 EXECUTING

1

mir



T2

RUNNING



ALL OK

Dashboards

Dashboards

Default Dashboard

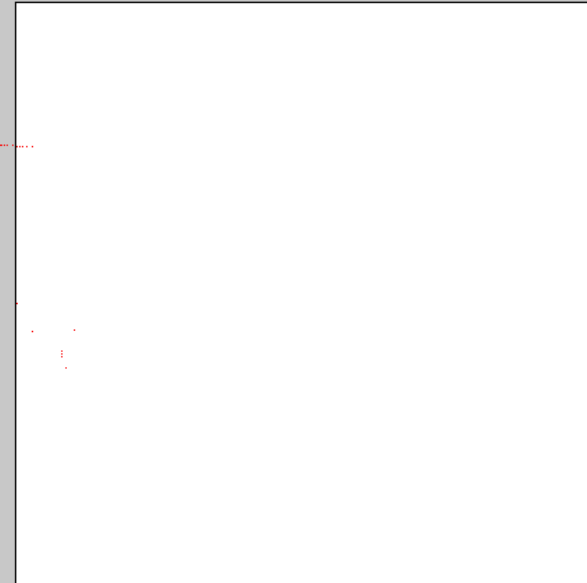
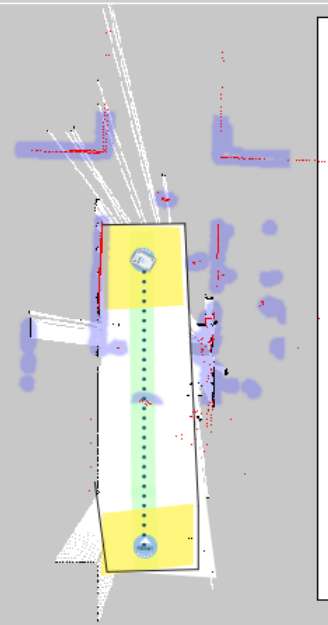
Logistic_System_MARA

AGV-Mara

Dashboard: Logistic_System_MARA | Contains 12 widget(s)



Drag the map to move your view or use the zoom-in and -out buttons to zoom.



✓ Succeeded Looping

15:35