



หุ่นยนต์เคลื่อนที่และ ROS2

หน่วยที่ 1-2 | WAVE ROVER และ UGV01 | วิชาหุ่นยนต์เคลื่อนที่ ระดับชั้นมัธยมศึกษาปีที่ 6

ครูเชาวลักษ์ณ ขาวษ์ • โรงเรียนธัญรัตน์

หน่วยการเรียนรู้ที่ 1

พื้นฐานหุ่นยนต์เคลื่อนที่และ WAVE ROVER



ทำความรู้จักหุ่นยนต์เคลื่อนที่



ศึกษาโครงสร้าง WAVE ROVER



ควบคุมหุ่นยนต์ด้วยคำสั่ง JSON
เบื้องต้น

หลังเรียนจบหน่วยที่ 1 ผู้เรียนสามารถ

- ✓ อธิบายประเภทของหุ่นยนต์เคลื่อนที่ได้
- ✓ เชื่อมต่อและควบคุม WAVE ROVER ผ่าน Web Interface ได้
- ✓ เขียนคำสั่ง JSON ควบคุมความเร็วล้อได้
- ✓ อ่านค่าจากเซนเซอร์ IMU ได้



หุ่นยนต์เคลื่อนที่คืออะไร

ความหมายและประเภทของ Mobile Robot



หุ่นยนต์แบบล้อ (Wheeled Robot)



หุ่นยนต์แบบสายพาน (Tracked Robot)



หุ่นยนต์แบบขา (Legged Robot)



การใช้งานจริงในอุตสาหกรรมและการเกษตร



เปรียบเทียบ 3 แพลตฟอร์มที่จะเรียน

แพลตฟอร์ม	โครงสร้าง	มอเตอร์
WAVE ROVER	4WD ตัวถังโลหะเต็มรูปแบบ	N20 ไม่มี Encoder
UGV01	แบบสายพาน ช่วงล่างอิสระ	มี Encoder
UGV02	4WD 6 ล้อ รับน้ำหนักสูง	JGB37-520 มี Encoder



หัวใจของหุ่นยนต์ ESP32 Driver Board

โครงสร้างแผงควบคุมหลัก



ชิป ESP32 พร้อม WiFi และ Bluetooth ในตัว



รองรับการเชื่อมต่อ Raspberry Pi และ Jetson Nano



จอแสดงผล OLED ในตัว



เซนเซอร์ IMU 9 แกน สำหรับวัดทิศทางและความเร่ง



ระบบพลังงาน UPS Power Module

การจ่ายไฟให้หุ่นยนต์



แบตเตอรี่ 18650 จำนวน 3 ก้อนต่ออนุกรม ความจุ 7800mAh



ชิป INA219 ตรวจวัดแรงดันและกระแสไฟแบบเรียลไทม์



ระบบป้องกันไฟเกิน กระแสเกิน และลัดวงจร



ชาร์จพร้อมใช้งานได้ในเวลาเดียวกัน



สเปคมอเตอร์ N20 ของ WAVE ROVER

รายละเอียดมอเตอร์ที่ใช้ขับเคลื่อน



แรงดันพิกัด 12V กระแสพิกัด 0.055A



แรงบิดพิกัด 0.09 กิโลกรัม-เซนติเมตร



ความเร็วรอบตัวเปล่า 66 รอบต่อนาที



ตัวอย่างนุ่มลดแรงกระแทก ความเร็วสูงสุด 1.25 เมตรต่อวินาที



การเชื่อมต่อ WiFi โหมด Access Point

ขั้นตอนการเชื่อมต่อครั้งแรก



เปิดสวิตช์และรอ ESP32 สร้างฮอตสปอตชื่อ UGV



เชื่อมต่อด้วยรหัสผ่าน 12345678



เปิดเบราว์เซอร์ Chrome พิมพ์ IP Address ของหุ่นยนต์



อ่านข้อมูลจากจอ OLED บรรทัดที่ 1-4



การเชื่อมต่อ WiFi โหมด Station

การเชื่อมต่อเข้าเครือข่ายที่รู้จัก



ใช้คำสั่ง CMD_WIFI_APSTA ระบุ SSID และรหัสผ่าน



หุ่นยนต์จะได้รับ IP จากเราเตอร์อัตโนมัติ



บันทึกการตั้งค่าไว้ใช้ครั้งต่อไป



ควบคุมจากอุปกรณ์อื่นในเครือข่ายเดียวกันได้



หน้าตา Web Control Interface

องค์ประกอบของหน้าเว็บควบคุม



ปุ่มทิศทางสำหรับบังคับการเคลื่อนที่



ปุ่มปรับความเร็ว SLOW MIDDLE FAST



หน้าต่าง Feedback Information สำหรับส่งคำสั่ง JSON



ระบบ Heartbeat Detection หยุดอัตโนมัติเมื่อขาดการเชื่อมต่อ



รู้จัก JSON การสื่อสารกับหุ่นยนต์

แนวคิดพื้นฐานของ JSON



รูปแบบข้อมูลที่มนุษย์และเครื่องอ่านง่าย



โครงสร้าง Key-Value คู่กัน



เหตุผลที่หุ่นยนต์ใช้ JSON ในการรับคำสั่ง



ช่องทางส่งคำสั่ง Serial USB HTTP และ ESP-NOW



คำสั่งควบคุมความเร็วล้อ CMD_SPEED_CTRL

```
{"T":1, "L":0.5, "R":0.5}
```



L คือความเร็วล้อซ้าย R คือความเร็วล้อขวา



ค่าความเร็วอยู่ในช่วง -0.5 ถึง +0.5



ค่าบวกคือเดินหน้า ค่าลบคือถอยหลัง



เป็นคำสั่งที่แนะนำให้ใช้ควบคุมการเคลื่อนที่



คำสั่งควบคุม PWM มอเตอร์โดยตรง

```
{"T":11, "L":164, "R":164}
```



CMD_PWM_INPUT สำหรับการดีบักเท่านั้น



ค่า PWM อยู่ในช่วง -255 ถึง +255



ข้อจำกัดของมอเตอร์ไฟตรงที่ความเร็วต่ำ



ไม่แนะนำให้ใช้งานจริง ให้ใช้ CMD_SPEED_CTRL แทน



ควบคุมหุ่นยนต์ผ่าน Python และ HTTP

การเขียนสคริปต์ควบคุมระยะไกล



ใช้ไลบรารี requests ส่งคำสั่งผ่าน HTTP



โครงสร้าง URL แบบ ip-address/js?json=คำสั่ง



ตัวอย่างโค้ด http_simple_ctrl.py



ทดลองรันสคริปต์ควบคุมหุ่นยนต์จริง



ปฏิบัติการที่ 1 ประกอบและทดสอบ WAVE ROVER

1

ตรวจสอบการติดตั้งแบตเตอรี่และหัวไฟ

2

เชื่อมต่อ WiFi และเปิดหน้าเว็บควบคุม

3

ทดสอบเดินหน้า ถอยหลัง เลี้ยวซ้ายขวา

4

บันทึกผลการทดสอบลงใบงาน



ปฏิบัติการที่ 2 อ่านค่าเซนเซอร์ IMU

1

ส่งคำสั่ง {"T":126} เพื่อขอข้อมูล IMU

2

อ่านค่ามุมหัน สนามแม่เหล็กโลก ความเร่ง และอุณหภูมิ

3

วิเคราะห์ความสัมพันธ์ของค่าเมื่อหุ่นยนต์เคลื่อนที่

4

สรุปผลเป็นตารางเปรียบเทียบ



สรุปหน่วยที่ 1



โครงสร้างและสเปคของ WAVE ROVER



การเชื่อมต่อ WiFi ทั้งสองโหมด



คำสั่ง JSON พื้นฐานที่ใช้ควบคุม



ต่อไป: หน่วยที่2 ระบบควบคุมแบบClosed-loop กับUGV01

หน่วยการเรียนรู้ที่ 2

ระบบควบคุมแบบ Closed-loop กับ UGV01



ทำความรู้จักมอเตอร์แบบมี Encoder



เรียนรู้หลักการควบคุม PID



ทดลองควบคุม UGV01 แบบแม่นยำ

หลังเรียนจบหน่วยที่ 2 ผู้เรียนสามารถ



อธิบายความแตกต่างระหว่าง Open-loop และ Closed-loop ได้



ปรับค่า PID เบื้องต้นได้



ใช้คำสั่ง CMD_ROS_CTRL ควบคุมความเร็วเชิงเส้นและมุมได้



วิเคราะห์ความแม่นยำการเคลื่อนที่ได้



Encoder คืออะไร ทำงานอย่างไร

หลักการพื้นฐานของเซนเซอร์วัดรอบ



อุปกรณ์วัดตำแหน่งและความเร็วการหมุนของมอเตอร์



ส่งสัญญาณพัลส์กลับมายัง ESP32



ทำให้ทราบความเร็วจริงของล้อแต่ละข้าง



เป็นพื้นฐานของการควบคุมแบบวงปิด



Open-loop เทียบกับ Closed-loop

ความแตกต่างของสองระบบควบคุม



Open-loop ไม่มีการตรวจสอบผลลัพธ์ ใช้ใน WAVE ROVER



Closed-loop มีการป้อนกลับข้อมูลตลอดเวลา ใช้ใน UGV01



ข้อดีของ Closed-loop คือความแม่นยำและความเสถียร



ผลกระทบต่อการเคลื่อนที่เป็นเส้นตรง



โครงสร้างช่วงล่างอิสระของ UGV01

จุดเด่นด้านกลไก



ระบบ Independent Suspension ทั้งสี่ด้าน



ลดแรงกระแทกจากภูมิประเทศขรุขระ



เมื่อปีนทางลาดชัน ช่วงล่างหดตัวเพิ่มแรงยึดเกาะ



วางอลูมิเนียมมาตรฐาน 1020 สำหรับต่อขยายอุปกรณ์



หลักการควบคุม PID เบื้องต้น

แนวคิดของตัวควบคุม PID



P คือสัดส่วนความคลาดเคลื่อนปัจจุบัน



I คือผลรวมความคลาดเคลื่อนสะสม



D คืออัตราการเปลี่ยนแปลงความคลาดเคลื่อน



การปรับสามค่านี้ทำให้มอเตอร์เสถียรและแม่นยำ



คำสั่งตั้งค่า Motor PID

```
{"T":2, "P":200, "I":2500, "D":0, "L":255}
```



PID ควบคุมพฤติกรรมตอบสนองของมอเตอร์



L คือค่าสำรองสำหรับ Windup Limit



ใช้ได้เฉพาะรุ่นที่มี Encoder เช่น UGV01



ทดลองปรับค่าและสังเกตความแตกต่าง



ปฏิบัติการที่ 3 ปรับค่า PID บน UGV01

1

ทดสอบค่า PID เริ่มต้นแล้วบันทึกพฤติกรรมหุ่นยนต์

2

ปรับค่า P เพิ่มขึ้นและลดลง สังเกตผล

3

ปรับค่า I และ D ทีละตัว สังเกตผล

4

สรุปค่าที่เหมาะสมที่สุดสำหรับพื้นผิวที่ทดสอบ



คำสั่งควบคุมแบบ ROS CMD_ROS_CTRL

```
{"T":13, "X":0.1, "Z":0.3}
```



X คือความเร็วเชิงเส้น หน่วยเมตรต่อวินาที



Z คือความเร็วเชิงมุม หน่วยเรเดียนต่อวินาที



ใช้ได้เฉพาะรุ่นที่มี Encoder



เตรียมพร้อมสำหรับการเชื่อมต่อกับ ROS2



Serial Port Continuous Feedback

การรับข้อมูลป้อนกลับแบบต่อเนื่อง



เปิดใช้งานด้วยคำสั่ง {"T":131, "cmd":1}



หุ่นยนต์ส่งข้อมูลสถานะโดยไม่ต้องร้องขอ



เหมาะสำหรับระบบ ROS ที่ต้องการข้อมูลเรียลไทม์



เปรียบเทียบกับโหมดถาม-ตอบแบบปกติ



ปฏิบัติการที่ 4 ทดสอบความแม่นยำการเคลื่อนที่

1

กำหนดเส้นทางเดินหน้าระยะทางที่แน่นอน

2

ทดสอบด้วย WAVE ROVER แบบ Open-loop

3

ทดสอบด้วย UGV01 แบบ Closed-loop

4

วัดระยะเบี่ยงเบนและสรุปผลเปรียบเทียบ



สรุปหน่วยที่ 2



ความแตกต่างของระบบควบคุมสองแบบ



หลักการและการปรับค่า PID



คำสั่ง ROS Control ที่เตรียมพร้อมสำหรับ ROS2



ต่อไป: หน่วยที่3 UGV02 และการสื่อสารไร้สายขั้นสูง