A logo of a robot club

Description automatically generated

**YBR-Tawan**

WRO Future Engineer

Documentation

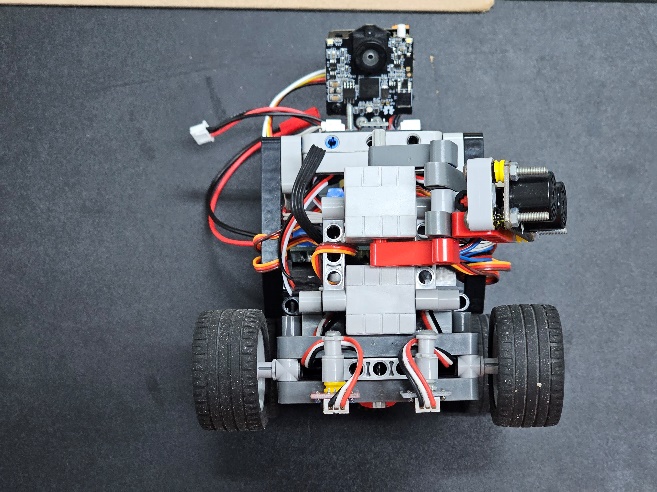
(Thai-version)

เอกสารนี้แบ่งออกเป็น

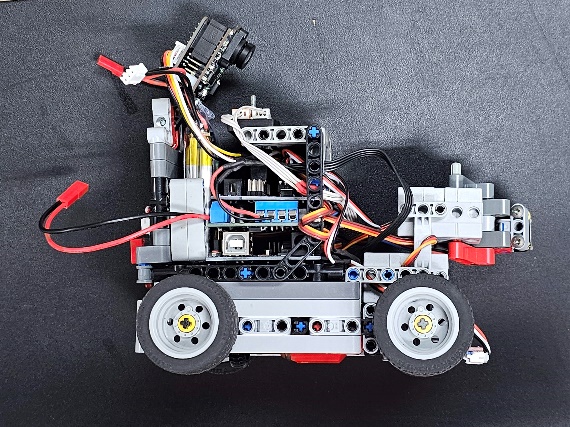
1. รูปภาพทีมงานและรถ
2. ข้อมูลทางวิศวกรรม
   1. การจัดการการเคลื่อนไหว
   2. การจัดการพลังงานและการตรวจเช็ค
   3. การจัดการอุปสรรค
   4. ปัจจัยทางวิศวกรรม
3. Youtube Link
4. Github link

**รูปภาพหุ่น**

หน้า

****

ขวา

****

บน

A toy car with wheels and wires

Description automatically generated

ซ้าย

A toy vehicle with wheels and wires

Description automatically generated

หลัง

A toy robot on a table

Description automatically generated

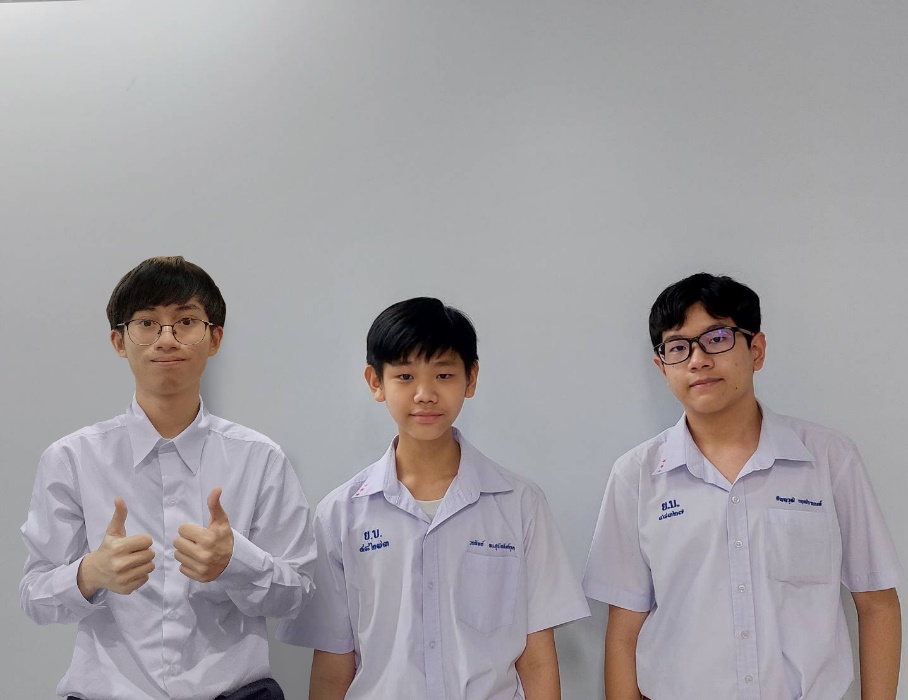
ล่าง

**A close up of a toy

Description automatically generated**

**รูปภาพทีมงาน**

****

****

**2. ข้อมูลทางวิศวกรรม**

**2.1 การจัดการการเคลื่อนไหว**

A grey and black lego block with black cord

Description automatically generatedเราใช้ Power Functions L-Motor ในการขับเคลื่อนของหุ่น

เป็นมอเตอร์ที่ง่ายและ เราเลือกมอเตอร์นี้เนื่องจากความง่ายต่อการเชื่อมต่อมอเตอร์เข้ากับหุ่นยนต์ของเรา และมอเตอร์นี้ก็มีราคาที่ถูกอีกด้วย

**ข้อมูลทางเทคนิค**

A graph with a red line and blue dots

Description automatically generated

น้ำหนัก:

A close up of a device

Description automatically generated

ความเร็วสูงสุด และ ปริมาณกระแสไฟฟ้า

A close up of a device

Description automatically generated

ความเร็ว เทียบกับ volt

A grey box with a wire

Description automatically generatedเราใช้ Geekservo 2kg 360 Degrees ในการเลี้ยวของหุ่น และใช้หมุน Ultrasonic Sensor

เป็น servo ที่ใช้ได้กับ LEGO ใช้ง่ายและสะดวกต่อการสร้างหุ่น

**ข้อมูลทางเทคนิค**

- ความเร็วในการหมุนแกน 60 องศา/0.12 วินาที

- แรงบิด 2 กิโลกรัม.เซนติเมตร ที่ไฟเลี้ยง 4.8V

- กระแสไฟฟ้าขณะไม่มีโหลด 20 ถึง 50mA

- กระแสไฟฟ้าขณะบังคับแกนหมุน (Stall) 600 ถึง 700mA

- กระแสไฟฟ้าสงบขณะหยุดทำงาน 7mA

- ความกว้างของสัญญาณพัลส์ที่ต้องการ 0.6 ถึง 2.4 มิลลิวินาที

- หมุนได้ 0 ถึง 360 องศา

Code การหมุน servo

void steering\_servo(int degree) {

  servo1.write((90 + max(min(degree, 45), -45)) / 2);

}

**2.2 การจัดการพลังงานและการตรวจเช็ค**

A green circuit board with black and white lights

Description automatically generated

เราใช้ ZX-03 Reflector ในการตรวจค่าสีบนสนาม 2 ตัว (แดง และ น้ำเงิน)

Light Reflector นี้เป็น sensor ที่ใช้วัดค่าแสงสะท้อน และในสนามนี้ เราใช้ sensor นี้เพื่อที่จะ check เส้นตอนที่หุ่นเลี้ยว

A close-up of a blue circuit board

Description automatically generatedเราใช้ sensor GY-25 เพื่อกำหนดทิศทางของหุ่นยนต์ของเราบนสนาม

เนื่องจากหุ่นจำเป็นต้องเดินเป็นเส้นตรงเป็นส่วนใหญ่ เราจึงเลือกใช้ sensor gyro เพื่อให้หุ่นรู้ทิศทางตัวมันเอง

Dataset: <http://mkpochtoi.ru/GY25_MANUAL_EN.pdf>

Library: <https://github.com/ElectronicCats/mpu6050>

A close-up of a camera

Description automatically generatedเราใช้ Pixy 2.1 เพื่อตรวจจับสิ่งกีดขวางที่อยู่บนสนาม

Module กล้องตัวนี้มาพร้อม library และ function ที่ผู้พัฒนาของ pixy camera ได้ทำไว้แล้ว จึงง่ายต่อการใช้มาก

4’Documentaion: <https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:start>

Software and Library: <https://pixycam.com/downloads-pixy2/>

A small black and gold device

Description automatically generatedเราใช้ Ultrasonic Sensor ( SEN0307 ) เพื่อวัดระยะห่างระหว่างหุ่นยนต์กับผนัง

Sensor ตัวนี้ใช้เอาต์พุตแรงดันไฟฟ้าแบบอะนาล็อก และให้การวัดระยะทางที่แม่นยำภายใน 2-500 ซม. ด้วยความละเอียด 1 ซม. และความแม่นยำ ±1% ซึ่มเหมาะกันการแข่งขันนี้มาก

A close-up of a blue circuit board

Description automatically generated

เราใช้ Arduino UNO สมองสำหรับหุ่นยนต์ของเรา

เราใช้ แบตเตอรี่ลิโพ Helicox 2200mah (7.4V) เพื่อให้พลังงานหุ่นของเรา

**การเดินสาย**

// Motor

ENB -> Arduino UNO Pin 11

INB -> Arduino UNO Pin 13

// Servos

STEER\_SRV -> Arduino UNO Pin 9

ULTRA\_SRV -> Arduino UNO Pin 8

// Ultrasonic Sensor

ULTRA\_DATA\_PIN -> Arduino UNO Pin 2

// Light Sensors

RED\_DATA\_SEN -> Arduino UNO Pin 0

BLUE\_DATA\_SEN -> Arduino UNO Pin 1

// Button

BUTTON\_DATA -> Arduino UNO Pin 3

// Pixy Camera

PIXY\_SDA -> Arduino UNO Pin 4

PIXY\_SCL -> Arduino UNO Pin 5

// Gryo

TX -> Arduino UNO RX

RX -> Arduino UNO TX

// Battery

BATTERY -> PWR\_IN of Arduino UNO

**2.3 การจัดการอุปสรรค**

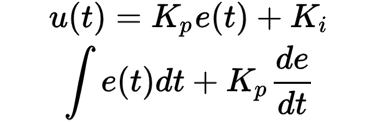
2.3.1 รอบคัดเลือก

ในรอบคัดเลือกหุ่นยนต์จะใช้ Ultrasonic sensor เพื่อคำนวณระยะทางระหว่างกำแพงของสนามกับหุ่น  
A drawing of a square with a blue and green object

Description automatically generated A drawing of a triangle with a light on the top

Description automatically generated with medium confidence

หุ่นจะนำระยะทางของหุ่นจากกำแพงและ องศาของ sensor gyro มาคำนวณเป็น steering degree (องศาการเลี้ยว) เพื่อให้หุ่นคงระยะห่างระหว่างกำแพงได้ด้วยสูตร **Proportional Integral Derivative (PID)**



สูตรการเดิน

    getIMU();

    line\_detection();

    ultra\_servo(pvYaw, TURN);

    int wall\_distance = getDistance();

    motor\_and\_steer(-1 \* compassPID.Run(pvYaw + ((wall\_distance - 25) \* 1) \* ((float(TURN == 'R') - 0.5) \* 2)));

2.3.2 รอบชิง

ในรอบคัดเลือกหุ่นยนต์จะใช้ Pixy camera Ultrasonic sensor และ Gyro เพื่อนำมาคำนวณ steering degree (องศาการเลี้ยว)

A white square with a blue rectangle and orange squares

Description automatically generated

หุ่นยังคงใช้ PID เหมือนกับรอบคัดเลือก แต่ตัวแปลที่เพิ่มขึ้นมาคือ avoidance degree (องศาในการเลี้ยวหลบ) โดย code จะเป็นดังต่อไปนี้

สูตรคำนวนองศาในการเลี้ยวหลบ

float calculate\_avoidance() {

  int blocks = pixy.ccc.getBlocks();

  found\_block = false;

  if (blocks) {

    int signature = -1;       // Signature of the object you want to detect

    int targetHeight = 10;    // Height of the object in centimeters

    float focalLength = 2.3;  // Focal length of the camera in centimeters

    float cameraFOV = 80.0;   // Field of view of the camera in degrees

    int largestBlockIndex = -1;

    int largestBlockArea = 0;

    for (int i = 0; i < blocks; i++) {

      if (pixy.ccc.blocks[i].m\_height > 1.33 \* float(pixy.ccc.blocks[i].m\_width)) {

        int objectArea = pixy.ccc.blocks[i].m\_width;

        // \* pixy.ccc.blocks[i].m\_height;

        found\_block = true;

        if (objectArea > largestBlockArea) {

          largestBlockIndex = i;

          largestBlockArea = objectArea;

          signature = pixy.ccc.blocks[i].m\_signature;

        }

      }

    }

    if (signature != -1) {

      int objectHeight = pixy.ccc.blocks[largestBlockIndex].m\_height;

      float distance = (targetHeight \* focalLength \* 100) / objectHeight;

      float blockCenterX = pixy.ccc.blocks[largestBlockIndex].m\_x;

      float blockCenterY = pixy.ccc.blocks[largestBlockIndex].m\_y;

      float deltaX = blockCenterX - pixy.frameWidth / 2;

      float deltaY = blockCenterY - pixy.frameHeight / 2;

      float detected\_degree = deltaX \* 40 / pixy.frameWidth;

      float blockPositionX = distance \* sin(degreesToRadians(detected\_degree));

      float blockPositionY = distance \* cos(degreesToRadians(detected\_degree)) - 17;

      if (signature == 1) {

        avoidance\_degree = max(radiansToDegree(atan2(blockPositionX + 9, blockPositionY)), 5);

        Blocks\_TURN = 'R';

      } else {

        avoidance\_degree = min(radiansToDegree(atan2(blockPositionX - 9, blockPositionY)), -5);

        Blocks\_TURN = 'L';

      }

    }

  }

  return avoidance\_degree;

}

**2.4 ปัจจัยทางวิศวกรรม**

หุ่นที่เราสร้างขึ้นมาเป็นหุ่นที่ออกแบบมาเพื่อใช้สำหรับงานนี้โดยเฉพาะ โดยใช้ชิ้นส่วน LEGO และน็อตต่างๆ ในการยึดชิ้นส่วนของหุ่นเข้าด้วยกัน

**A toy car with wheels and wires

Description automatically generated**

ปัญหาของหุ่นที่เราพบ

1. แรงบิดของมอเตอร์

มอเตอร์ตัวนี้มีแรงบิดอยู่ที่ 18 N.cm ซึ่งเราพบว่าหาก รันมอเตอร์โดยใช้ แบตเตอรี่ลิโพ 7.4V ที่ power **ประมาณ 20-30%** จะสามารถทำให้หุ่นยนต์เริ่มเคลื่อนที่ได้

1. แบตเตอรี่ลิโพ

เนื่องจาก Arduino UNO ไม่มี indicator ที่ใช้เตือนว่าแบตเตอรี่หมด เราจึงต้องวัด แบตเตอรี่อยู่ตลอด แบตเตอรี่ **ไม่ควรมีประจุต่ำกว่า 7.7V** เพราะค่าของ Program อาจจะผิดพลาดได้

1. Gyro Drift

หมายถึงปรากฏการณ์ที่เอาต์พุตของไจโรสโคปค่อยๆ เบี่ยงเบนไปจากค่าที่คาดไว้เมื่อเวลาผ่านไป เพราะฉะนั้นตอนเปิดหุ่นของเรา **จำเป็นต้องวางหุ่นไว้ที่พื้นสนามก่อน และจึงจะค่อยๆ เสียบแบตเตอรี่ได้**

1. Loose Power Connection

บางครั้งหุ่นจะ reset ตัวเอง เหตุการณ์นี้เกิดจากสายที่เสียบเข้าที่ Arduino UNO อาจจะหลวม หรือไม่ได้บัดกรีสายก่อน ทำให้ไม่สามารถส่ง ไฟฟ้า ไปที่ Pixy camera, Ultrasonic sensor, Light sensor 2 ตัว, Gyro sensor, servo และ motor ได้อย่างทั่วถึง

**3. YouTube Video**

Qualification Round (ทำภารกิจสำเร็จประมาณ 37 วินาที)

<https://youtu.be/zxlOy2REIk4>

Final Round (ทำภารกิจสำเร็จประมาณ 45 วินาที)

<https://youtu.be/RNagUxmCIuk>

**4. GitHub link**

<https://github.com/natapol2547/YBR-Tawan>

YB Robot Club **[YBR-Tawan]**