

## 1. บทนำ

ความหมายของ "หุ่นยนต์" โดยสถาบันหุ่นยนต์อเมริกา (The Robotics Institute of America) ให้ความหมายไว้ ดังนี้ "หุ่นยนต์ถูกออกแบบมาเพื่อความสามารถหลากหลายโดยสามารถใช้ในการเคลื่อนย้ายวัสดุ อุปกรณ์ ผ่านโปรแกรมที่ถูกตั้งไว้เพื่อประสิทธิภาพการทำงานที่หลากหลาย" (A robot reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through various programmed motions for the performance of a variety of tasks.) [1] โดยหุ่นยนต์ถูกแบ่งออกเป็น 2 ประเภทตามลักษณะการใช้งาน คือ 1) หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (fixed robot) เป็นหุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปไหนได้ด้วยตัวเอง มีลักษณะเป็นแขนกล (Robotic arms) ที่สามารถหยิบและเคลื่อนไหวได้เฉพาะแต่ละข้อต่อมักใช้ในโรงงานอุตสาหกรรม เช่น โรงงานประกอบรถยนต์ 2) หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (mobile robot) เป็นหุ่นยนต์ที่สามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเองโดยการใช้ล้อหรือขาซึ่งปัจจุบันยังมีการทำวิจัยและพัฒนาเพื่อใช้งานในรูปแบบต่าง การใช้งานหุ่นยนต์ในภาคอุตสาหกรรมส่วนใหญ่เป็นงานที่ไม่ซับซ้อนมากนัก โดยใช้ระบบอัตโนมัติ (Automation) ควบคุมให้หุ่นยนต์สามารถทำงานได้เองผ่านการเขียนโปรแกรมโดยหากเป็นงานที่มีความซับซ้อนมากขึ้นจะใช้ปัญญาประดิษฐ์ (Artificial Intelligence) เข้าช่วยควบคุมเพื่อให้หุ่นยนต์มีความสามารถสูงขึ้นหรือฉลาดขึ้นนั่นเอง [2]

ประเทศไทยจึงมีการวางแผนสร้างฐานการผลิตหุ่นยนต์เพื่อตอบสนองความต้องการในประเทศและภูมิภาคอาเซียนโดยตรง ได้แก่ หุ่นยนต์ที่ใช้ในอุตสาหกรรมการผลิตยานยนต์ โดยเฉพาะหุ่นยนต์ที่ใช้ในการเชื่อมโลหะ ซึ่งมีจำนวนมากเป็นอันดับหนึ่งของจำนวนหุ่นยนต์ที่นำเข้ามาในภูมิภาคอาเซียน หรือนับเป็นร้อยละ 38 ของจำนวนหุ่นยนต์ที่นำเข้าทั้งหมด โดยหุ่นยนต์เหล่านี้ มักจะมาในรูปแบบแขนหุ่นยนต์ที่มีแกนเคลื่อนที่แบบหมุน (Articulated Robot) รองลงมาคือ หุ่นยนต์ที่ใช้ในกระบวนการผลิตอัดฉีดพลาสติก ที่มีการนำเข้ามากเป็นอันดับสองในภูมิภาค หรือร้อยละ 19 ของจำนวนหุ่นยนต์ที่นำเข้าทั้งหมด โดยหุ่นยนต์เหล่านี้เป็นแขนหุ่นยนต์ที่มีทั้งรูปแบบแกนเคลื่อนที่แบบหมุน และรูปแบบแกนเคลื่อนที่แบบเชิงเส้น (Linear Gantry Robot) และหุ่นยนต์ที่มีความเชี่ยวชาญเฉพาะด้าน เช่น หุ่นยนต์ดำน้ำ และหุ่นยนต์ที่ใช้ในปฏิบัติการทางการแพทย์ โดยมุ่งเน้นรูปแบบที่ผลิตมาเพื่อสรีระของผู้ป่วยชาวเอเชีย ซึ่งการผลิตหุ่นยนต์ประเภทหลังนี้จะเป็นการพัฒนาหลังจากที่ประเทศไทยมีประสบการณ์จากการผลิตหุ่นยนต์แบบแขนกลแล้ว [3]

จากข้อมูลดังกล่าวทางคณะผู้จัดทำได้ตระหนักถึงความสำคัญในการการสร้างหุ่นยนต์เพื่อการศึกษา ที่เป็นพื้นฐานและสามารถนำไปต่อยอดได้ในอนาคตและได้ทำการรวบรวมข้อมูลที่เกี่ยวข้องกับการสร้างหุ่นยนต์ จึงนำไปสู่การสร้างหุ่นยนต์จริงที่มีชื่อว่า “Bumblebee” เพื่อใช้ในการแข่งขันการแข่งขันหุ่นยนต์โดยมีลักษณะคล้ายกับการเล่นบอลลูนด้านแบ่งเป็นทีมรุกและทีมรับ สลับกันในการแข่งแต่ละรอบ แต่ละทีมจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาทีมรับ จนผ่านเส้นแดงแล้วกลับมาอย่างปลอดภัย(ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้และจะเป็นฝ่ายชนะในการแข่งขัน หากหุ่นยนต์ตัวไหนถูกทีมรับจับได้จะหมดสิทธิ์จากการแข่งขันในรอบนั้น ส่วนทีมรับจะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น หากวิ่ง

ออกนอกพื้นที่ก็จะหมดสิทธิ์จากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น

## 2) ทฤษฎีที่เกี่ยวข้อง

### 2.1 ภาษา c

#### 2.1.1 ฟังก์ชัน

1.ฟังก์ชัน setup() เป็นฟังก์ชันการกำหนดค่าต่าง ๆ ในส่วนนี้มีการกำหนดค่าเพียงครั้งเดียวเท่านั้น เช่น กำหนดค่าในการใช้งานให้เป็นขาอินพุตหรือขาเอาต์พุต การกำหนดค่าของการเรียกใช้ไลบรารี

```
void setup()
{
    //เป็นส่วนของคำสั่ง สำหรับกำหนดการทำงานในโปรแกรม และทำเพียงครั้งเดียว
}
```

2.ฟังก์ชัน loop() เป็นส่วนในการเขียนโปรแกรมและสั่งให้โปรแกรมทำงาน ซึ่งมีการทำงานเป็นแบบวนลูปไปเรื่อย ๆ ตามการเขียนโปรแกรมของผู้พัฒนาโปรแกรมเพื่อรับค่าจากอินพุต นำค่าที่ได้มาประมวลผล แล้วทำการส่งข้อมูลออกเอาต์พุตเพื่อควบคุมการทำงานตามโปรแกรม

```
void loop()
{
    // เป็นโปรแกรมหลักของคำสั่ง ซึ่งในส่วนนี้โปรแกรมมีการทำงานตลอดเวลา
}
```

#### 2.1.2 คำสั่งควบคุม

คำสั่ง if คือการตรวจสอบเงื่อนไขการทำงานของโปรแกรมถ้าเงื่อนไขเป็นจริง ให้ทำงานตามคำสั่งที่กำหนดนั้น

คำสั่ง if...else คือคำสั่งกำหนดเงื่อนไขการทำงานของโปรแกรม โดยมี 2 เงื่อนไข ถ้าเงื่อนไขเป็นจริงทำงานตามคำสั่งที่กำหนดแบบหนึ่ง ถ้าเงื่อนไขเป็นเท็จทำงานตามคำสั่งที่กำหนดอีกแบบหนึ่ง

คำสั่ง for คือคำสั่งให้โปรแกรมทำงานซ้ำตามจำนวนรอบที่ต้องการมีรูปแบบคำสั่ง

คำสั่ง Switch case คือคำสั่งเพื่อกำหนดการทำงานของโปรแกรมหลาย ๆ เงื่อนไข ถ้าตัวแปรที่กำหนดตรงกับเงื่อนไขนั้น ๆ ทำให้โปรแกรมทำงานตามที่กำหนดไว้แต่ละเงื่อนไข

คำสั่ง while คือคำสั่งทำซ้ำแบบวนรอบ ถ้าเงื่อนไขเป็นจริงโปรแกรมทำงานตามคำสั่งที่เขียนไว้ในวงเล็บปีกกา แต่ถ้าเงื่อนไขเป็นเท็จโปรแกรมจบการทำงานในคำสั่ง while

คำสั่ง do..while คือคำสั่งทำซ้ำแบบวนรอบ โดยมีการทำงานตรงกันข้ามกับคำสั่ง

while คือทำงานตามคำสั่งที่เขียนไว้ในวงเล็บปีกกา แล้วจึงมาตรวจสอบเงื่อนไข แต่ถ้าเงื่อนไขเป็นเท็จ โปรแกรมจบการทำงานในคำสั่ง do

คำสั่ง break คือคำสั่งใช้ร่วมกับคำสั่งการทำงานแบบวนรอบ ได้แก่ คำสั่ง do, for while หรือ Switch เพื่อให้โปรแกรมหยุดการทำงานจากการวนรอบโดยไม่มีเงื่อนไข

คำสั่ง continue คือคำสั่งใช้สำหรับข้ามการทำงานของคำสั่งถัดไป คำสั่งนี้เขียนอยู่ใน คำสั่งการทำงานแบบวนรอบ ได้แก่ คำสั่ง do, for หรือ while

### 2.1.3 การดำเนินการเปรียบเทียบ

เป็นเครื่องหมายที่ใช้ในการเปรียบเทียบทางคณิตศาสตร์ มีเครื่องหมายดังต่อไปนี้

- เครื่องหมาย == เป็นการเปรียบเทียบเท่ากับ
- เครื่องหมาย != เป็นการเปรียบเทียบไม่เท่ากับ
- เครื่องหมาย < เป็นการเปรียบเทียบน้อยกว่า
- เครื่องหมาย > เป็นการเปรียบเทียบมากกว่า
- เครื่องหมาย <= เป็นการเปรียบเทียบน้อยกว่าหรือเท่ากับ
- เครื่องหมาย >= เป็นการเปรียบเทียบมากกว่าหรือเท่ากับ

2.1.4 ชุดคำสั่งเป็นชุดคำสั่งในการเขียนโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามโปรแกรมที่ออกแบบไว้ โดยมีคำสั่งต่าง ๆ ดังนี้

#### 1. คำสั่งดิจิทัล อินพุต/เอาต์พุต

- คำสั่ง pinMode() เป็นการกำหนดพอร์ตเป็นอินพุตหรือเอาต์พุต
- คำสั่ง digitalWrite() เป็นการเขียนข้อมูลออกพอร์ตที่กำหนด
- คำสั่ง digitalRead() เป็นการอ่านข้อมูลเข้าพอร์ตที่กำหนด

#### 2. คำสั่งอนาล็อก อินพุต/เอาต์พุต

- คำสั่ง analogReference() เป็นการกำหนดค่าแรงดันอ้างอิงที่ใช้สำหรับอนาล็อกอินพุต
- คำสั่ง analogRead() เป็นการอ่านแรงดันไฟฟ้าแบบอนาล็อกและแปลงเป็นจำนวนเต็ม มีค่า

ระหว่าง 0 ถึง 1023

- คำสั่ง analogWrite() เป็นการให้ PWM เขียนค่าออกทางพอร์ตที่กำหนด

#### 3. คำสั่งเวลา

- คำสั่ง millis() เป็นการห้วงเวลามีหน่วยเป็นมิลลิวินาทีของ Arduino ทันทีที่มีไฟเลี้ยงเข้า

Arduino

- คำสั่ง micros() เป็นการห้วงเวลามีหน่วยเป็นไมโครวินาทีของ Arduino ทันทีที่มีไฟเลี้ยงเข้า

Arduino

- คำสั่ง delay() เป็นการห้วงเวลาตามค่าที่กำหนด มีหน่วยเป็นมิลลิวินาที
- คำสั่ง delayMicroseconds() เป็นการห้วงเวลาตามค่าที่กำหนด มีหน่วยเป็นไมโครวินาที

#### 4. คำสั่งคณิตศาสตร์

- คำสั่ง min() เป็นการหาค่าต่ำสุด
- คำสั่ง max() เป็นการหาค่ามากที่สุด

#### 5. คำสั่งสุ่มตัวเลข

- คำสั่ง randomSeed() เป็นการกำหนดค่าเริ่มต้นของฟังก์ชัน random
- คำสั่ง random() เป็นการสุ่มค่าตัวเลขระหว่างตัวเลขที่กำหนด

#### 6. คำสั่งบิตและไบต์

- คำสั่ง LowByte() เป็นตัวแปรของไบต์ต่ำสุด
- คำสั่ง highByte() เป็นตัวแปรของไบต์สูงสุด
- คำสั่ง bitRead() เป็นการอ่านบิตของตัวแปร
- คำสั่ง bitWrite() เป็นการเขียนบิตของตัวแปร
- คำสั่ง bitSet() เป็นการตั้งบิตของตัวแปรเท่ากับ 1
- คำสั่ง bitClear() เป็นการตั้งบิตของตัวแปรเท่ากับ 0
- คำสั่ง bit() เป็นการตั้งค่าบิตตามค่าที่กำหนด

#### 7. คำสั่งการติดต่อสื่อสาร

- คำสั่ง Serial.begin() เป็นการกำหนดอัตราการส่งข้อมูล
- คำสั่ง Serial.end() เป็นการปิดใช้งานการสื่อสารแบบอนุกรม
- คำสั่ง Serial.available() เป็นการตรวจสอบการรับข้อมูลจากการสื่อสารแบบอนุกรม
- คำสั่ง Serial.read() เป็นการอ่านข้อมูลจากการสื่อสารแบบอนุกรมที่เข้ามา
- คำสั่ง Serial.peek() เป็นการส่งกลับไบต์ต่อไปของข้อมูลการสื่อสารแบบอนุกรม
- คำสั่ง Serial.flush() เป็นการลบข้อมูลทั้งหมดในบัฟเฟอร์
- คำสั่ง Serial.print() เป็นการพิมพ์ข้อมูลไปยังพอร์ตอนุกรม
- คำสั่ง Serial.println() เป็นการพิมพ์ข้อมูลไปยังพอร์ตอนุกรม และขึ้นบรรทัดใหม่
- คำสั่ง Serial.write() เป็นการส่งข้อมูลไบต์ไปยังพอร์ตอนุกรม