

**รายงานวิชา Pre-Project รหัสวิชา 01216747**

นางสาวพัชรินทร์ แย้มเดช รหัสนักศึกษา 60010684

นางสาวภูริพิชญ์ ธรรมโม รหัสนักศึกษา 60010811

นางสาวสิริวิมล มีทอง รหัสนักศึกษา 60011075

เสนอ

ผศ.ดร.อุดม จันทร์จรัสสุข

ภาคเรียนที่ 2 ปีการศึกษา 2562

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมอุตสาหการ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**คำนำ**

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา Pre-Project รหัสวิชา 01216747 เนื้อหาเกี่ยวกับการนำ โปรแกรม Arduino มาใช้สร้างหุ่นยนต์เพื่อทำการแข่งขันบอลลูนด่าน จะมีทั้งเนื้อหาข้อมูลที่นำมาใช้สร้างหุ่นยุนต์ ขั้นตอน การวางแผนกลยุทธ์ทั้งเกมรุกและเกมรับ การออกแบบหุ่นยนต์ด้วยโปรแกรม solidworks การต่อวงจร แต่เนื่องด้วยปัญหา covid-19 ทำให้คณะผู้จัดทำไม่ได้แข่งขันกันจริงๆจึงทำให้กลุ่มของเราอาจะไม่ทราบถึงข้อบกพร่อง ปัญหาของการทำงานในครั้งนี้เท่าไหร่นัก

คณะผู้จัดทำหวังว่ารายงานฉบับนี้จะสามารถให้ความรู้ ให้ประโยชน์แก่ท่านผู้อ่านไม่มากก็น้อย หากมีความผิดพลาดประการใดคณะผู้จัดทำต้องขออภัยมา ณ ที่นี้

คณะผู้จัดทำ

**สารบัญ**

**เรื่อง หน้า**

บทนำ 1

ปัญหาหรือโจทย์ที่ต้องการแก้ไข 2

ขอบเขตของโครงงาน 3

ทฤษฏีที่เกี่ยวข้อง 3

วิธีการดำเนินงาน 13

แผนการดำเนินงาน 27

งบประมาณ 27

สิ่งที่คาดว่าจะได้รับ 28

สรุป 28

เอกสารที่เกี่ยวข้อง 29

**บทที่ 1**

**1.1 บทนำ**

ความหมายของ "หุ่นยนต์" โดยสถาบันหุ่นยนต์อเมริกา (The Robotics Institute of America) ได้ให้ความหมายไว้ ดังนี้ "หุ่นยนต์ถูกออกแบบมาเพื่อความสามารถหลากหลายโดยสามารถใช้ในการเคลื่อนย้ายวัสดุ อุปกรณ์ ผ่านโปรแกรมที่ถูกตั้งไว้เพื่อประสิทธิภาพการทำงานที่หลากหลาย"[1] โดยหุ่นยนต์ถูกแบ่งออกเป็น 2 ประเภทตามลักษณะการใช้งาน คือ 1) หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (fixed robot) เป็นหุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปไหนได้ด้วยตัวเอง มีลักษณะเป็นแขนกล (Robotic arms) ที่สามารถขยับและเคลื่อนไหวได้เฉพาะแต่ละข้อต่อมักใช้ในโรงงานอุตสาหกรรม เช่น โรงงานประกอบรถยนต์ 2) หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (mobile robot) เป็นหุ่นยนต์ที่สามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเองโดยการใช้ล้อหรือขาซึ่งปัจจุบันยังมีการทำวิจัยและพัฒนาเพื่อใช้งานในรูปแบบต่าง การใช้งาน**หุ่นยนต์**ในภาคอุตสาหกรรมส่วนใหญ่เป็นงานที่ไม่ซับซ้อนมากนัก โดยใช้**ระบบอัตโนมัติ** (Automation) ควบคุมให้**หุ่นยนต์**สามารถทำงานได้เองผ่านการเขียนโปรแกรมโดยหากเป็นงานที่มีความซับซ้อนมากขึ้นจะใช้ปัญญาประดิษฐ์(Artificial Intelligence) เข้าช่วยควบคุมเพื่อให้**หุ่นยนต์**มีความสามารถสูงขึ้นหรือฉลาดขึ้นนั่นเอง [2]

ประเทศไทยจึงมีการวางแผนสร้างฐานการผลิตหุ่นยนต์เพื่อตอบสนองความต้องการในประเทศและภูมิภาคอาเซียนโดยตรง ได้แก่หุ่นยนต์ที่ใช้ในอุตสาหกรรมการผลิตยานยนต์ โดยเฉพาะหุ่นยนต์ที่ใช้ในการเชื่อมโลหะ ซึ่งมีจำนวนมากเป็นอันดับหนึ่งของจำนวนหุ่นยนต์ที่นำเข้ามาในภูมิภาคอาเซียน หรือนับเป็นร้อยละ 38 ของจำนวนหุ่นยนต์ที่นำเข้าทั้งหมด โดยหุ่นยนต์เหล่านี้ มักจะมาในรูปแบบแขนหุ่นยนต์ที่มีแกนเคลื่อนที่แบบหมุน (Articulated Robot) รองลงมาคือ หุ่นยนต์ที่ใช้ในกระบวนการผลิตอัดฉีดพลาสติก ที่มีการนำเข้ามากเป็นอันดับสองในภูมิภาค หรือร้อยละ 19 ของจำนวนหุ่นยนต์ที่นำเข้าทั้งหมด โดยหุ่นยนต์เหล่านี้เป็นแขนหุ่นยนต์ที่มีทั้งรูปแบบแกนเคลื่อนแบบหมุน และรูปแบบแกนเคลื่อนที่แบบเชิงเส้น (Linear Gantry Robot) และหุ่นยนต์ที่มีความเชี่ยวชาญเฉพาะด้าน เช่น หุ่นยนต์ดำน้ำ และหุ่นยนต์ที่ใช้ในปฏิบัติการทางการแพทย์ โดยมุ่งเน้นรูปแบบที่ผลิตมาเพื่อสรีระของผู้ป่วยชาวเอเชีย ซึ่งการผลิตหุ่นยนต์ประเภทหลังนี้ จะเป็นการพัฒนาหลังจากที่ประเทศไทยมีประสบการณ์จากการผลิตหุ่นยนต์แบบแขนกลแล้ว[3]

จากข้อมูลดังกล่าวทางคณะผู้จัดทำได้ตระหนักถึงความสำคัญในการการสร้างหุ่นยนต์เพื่อการศึกษา ที่เป็นพื้นฐานและสามารถนำไปต่อยอดได้ในอนาคตและได้ทำการรวบรวมข้อมูลที่เกี่ยวข้องกับการสร้างหุ่นยนต์ จึงนำไปสู่การสร้างหุ่นยนต์จริงที่มีชื่อว่า “BUMBLEBEE” เพื่อใช้ในการแข่งขันการแข่งขันหุ่นยนต์โดยมีลักษณะคล้ายกับการเล่นบอลลูนด่านแบ่งเป็นทีมรุกและทีมรับสลับกันในการแข่งแต่ละรอบ แต่ละทีมจะประกอบด้วยหุ่นยนต์ 7 ตัว ฝ่ายทีมรุกจะต้องวิ่งไปหาทีมรับ จนผ่านเส้นแดงแล้วกลับมาอย่างปลอดภัย(ผ่านเส้นสีเหลือง) โดยที่ไม่ถูกทีมรับจับได้และจะเป็นฝ่ายชนะในการแข่งขัน หากหุ่นยนต์ตัวไหนถูกทีมรับจับได้จะหมดสิทธิ์จากการแข่งขันในรอบนั้น ส่วนทีมรับจะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น หากวิ่งออกนอกพื้นที่ก็จะหมดสิทธิ์จากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จหรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น

**บทที่ 2**

**2.1 ปัญหาหรือโจทย์ที่ต้องการแก้ไข**

การแข่งขันหุ่นยนต์มีลักษณะคล้ายกับการเล่นบอลลูนด่านหรือเล่นเตยด่าน โดยแบ่งเป็นทีมรุกและทีมรับสลับกันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ผ่ายทีมรุกจะต้องวิ่งไปหาฝั่งตรงข้าม จนผ่านเส้นแดง แล้วกลับมาอย่างปลอดภัย(ผ่านเส้นสีเหลือง)โดยที่ไม่ถูกทีมรับจับได้ก็จะเป็นฝ่ายชนะในการแข่งขันรอบนั้นหุ่นยนต์ที่ถูกจับได้จะถูกตัดออกจากการแข่งขันในรอบนั้นส่วนทีมรับจะสามารถวิ่งสกัดกั้นฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น ถ้าวิ่งออกนอกพื้นที่ก็จะถูกตัดออกจากการแข่งขันในรอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ทีมรับจะเป็นฝ่ายชนะการแข่งขันของแต่ละรอบจะยุติเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น

**2.1.2 แนวคิดในการแก้ปัญหาหรือในการสร้างหุ่นยนต์**

**กลยุทธ์เกมรุก**

สำหรับเกมส์รุก ทีมเรามีความเห็นว่าตัวรถควรมีเซ็นเซอร์ ทั้ง 4 ด้าน คือ ด้านซ้าย ด้านขวา ด้านหน้าและด้านหลัง หากกำลังเดินหน้า และเซ็นเซอร์ด้านหน้าสามารถจับตัวรถอีกคันได้ ให้ทำการเคลื่อนที่ไปทางซ้ายหรือขวา และหากเซ็นเซอร์ทางด้านซ้ายและขวา จับได้ว่ามีรถคันอื่น จะทำการถอยหลังแทน และจะมีการเพิ่มความเร็วของตัวรถ หากพบว่าเซ็นเซอร์ทั้ง 4 ด้าน ไม่พบสัญญาณของรถคันอื่น

รวมไปถึงเราจะมีระบบเซ็นเซอร์ตรวจจับเส้นทางว่าหากเซ็นเซอร์โดนเส้นสีแดงหมายความว่าได้ผ่านทีมรับมาได้แล้ว และเตรียมถอยหลังหรือกลับรถเพื่อเคลื่อนตัวกลับไปยังที่เดิม ถ้าเซ็นเซอร์โดนเส้นสีเหลืองหมายความว่า ทีมรุกสามารถกลับมายังฝั่งของตัวเองได้สำเร็จ ให้ทำการหยุดเครื่องได้

**กลยุทธ์เกมรับ**

สำหรับเกมส์รับ ทางทีมเราเห็นว่าอาจให้รถของสมาชิกในทีมอยู่ติดกัน โดยแถวแรก 3 คัน แถวที่ 2 มี 4 คัน แล้วเคลื่อนที่สลับฟันปลากัน เช่นแถวแรกเคลื่อนที่ไปทางซ้าย แถวที่สองจะเคลื่อนที่ไปทางขวา เพื่อลดช่องโหว่ที่จะให้อีกทีมผ่านเข้ามาได้ หรือหากมีการเคลื่อนที่ของฝ่ายตรงข้ามแบบ 1:1 แล้วนั้นจะมีเพื่อนคันข้างๆ เข้ามาช่วยป้องกันอีกทีม ทั้งนี้จะต้องมีระบบเซ็นเซอร์ที่ตรวจจับเส้นสีดำ เพื่อป้องกันไม่ให้ฝ่ายรับนั้นเคลื่อนรถออกจากโซนป้องกัน หากพบรถจะทำการเดินไปข้างซ้ายหรือขวา ขึ้นอยู่กับขอบของสนามว่าฝั่งไหน

**บทที่ 3**

**ขอบเขตของโครงงาน**

**3.1 ขนาด น้ำหนัก ความเร็ว**

ขนาดกว้าง x ยาว : 10 x 10 เซนติเมตร ความสูงไม่เกิน 13 เซนติเมตร ใช้ล้อ 4 ล้อในการขับเคลื่อน ความเร็ว 255 น้ำหนัก 300 กรัม

**3.2 เครื่องมือ อุปกรณ์ที่จำเป็นในการทำโครงงาน**

ล้อ 4 ล้อ Sensor DC Geared-Motors 2 ตัว ถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7 V ที่ชาร์จถ่าน สายไฟ โครงรถ

**บทที่ 4**

**ทฤษฎีที่เกี่ยวข้อง**

**4.1 ภาษา c**

* + 1. **ฟังก์ชัน**
* ฟังก์ชัน setup() เป็นฟังก์ชันการกำหนดค่าต่าง ๆ ในส่วนนี้มีการกำหนดค่าเพียงครั้งเดียวเท่านั้น เช่น กำหนดขาในการใช้งานให้เป็นขาอินพุตหรือขาเอาต์พุต การกำหนดค่าของการเรียกใช้ไลบรารี

                        void setup()

                        {

                        //เป็นส่วนของคำสั่ง สำหรับกำหนดการทำงานในโปรแกรม และทำเพียงครั้งเดียว

                        }

* ฟังก์ชัน loop() เป็นส่วนในการเขียนโปรแกรมและสั่งให้โปรแกรมทำงาน ซึ่งมีการทำงานเป็นแบบวนลูปไปเรื่อย ๆ ตามการเขียนโปรแกรมของผู้พัฒนาโปรแกรมเพื่อรับค่าจากอินพุต นำค่าที่ได้มาประมวลผล แล้วทำการส่งข้อมูลออกเอาต์พุตเพื่อควบคุมการทำงานตามโปรแกรม

                    void loop()

                    {

                    // เป็นโปรแกรมหลักของคำสั่ง ซึ่งในส่วนนี้โปรแกรมมีการทำงานตลอดเวลา

                    }

* + 1. **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 white หรือ Switch เพื่อให้โปรแกรมหยุดการทำงานจากการวนรอบโดยไม่มีเงื่อนไข
* คำสั่ง continue คือคำสั่งใช้สำหรับข้ามการทำงานของคำสั่งถัดไป คำสั่งนี้เขียนอยู่ใน คำสั่งการทำงานแบบวนรอบ ได้แก่ คำสั่ง do, for หรือ while
  + 1. **การดำเนินการเปรียบเทียบ**

เป็นเครื่องหมายที่ใช้ในการเปรียบเทียบทางคณิตศาสตร์ มีเครื่องหมายดังต่อไปนี้

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

**4.1.4 ชุดคำสั่ง**

เป็นชุดคำสั่งในการเขียนโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์ทำงานตามโปรแกรมที่ออกแบบไว้ [4]

1.คำสั่งดิจิตอล อินพุต/เอาต์พุต

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

         2.คำสั่งอนาล็อก อินพุต/เอาต์พุต

* คำสั่ง analogReference() เป็นการกำหนดค่าแรงดันอ้างอิงที่ใช้สําหรับอนาล็อกอินพุต
* คำสั่ง analogRead() เป็นการอ่านแรงดันไฟฟ้าแบบอนาล็อกและแปลงเป็นจํานวนเต็ม มีค่าระหว่าง 0 ถึง 1023
* คำสั่ง analogWrite() เป็นการใช้ PWM เขียนค่าออกทางพอร์ตที่กำหนด

        3.คำสั่งเวลา

* คำสั่ง millis() เป็นการหน่วงเวลามีหน่วยเป็นมิลลิวินาทีของ Arduino ทันทีที่มีไฟเลี้ยงเข้า Arduino
* คำสั่ง micros() เป็นการหน่วงเวลามีหน่วยเป็นไมโครวินาทีของ Arduino ทันทีที่มีไฟเลี้ยงเข้า Arduino
* คำสั่ง delay() เป็นการหน่วงเวลาตามค่าที่กำหนด มีหน่วยเป็นมิลลิวินาที
* คำสั่ง delayMicroseconds() เป็นการหน่วงเวลาตามค่าที่กำหนด

         4.คำสั่งคณิตศาสตร์

* คำสั่ง min() เป็นการหาค่าต่ำสุด
* คำสั่ง max() เป็นการหาค่ามากสุด

        5.คำสั่งบิตและไบต์

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

6. คำสั่งการติดต่อสื่อสาร

* คำสั่ง bit() เป็นการตั้งค่าบิตตามค่าที่กำหนด
* คำสั่ง Serial.begin() เป็นการกำหนดอัตราการส่งข้อมูล
* คำสั่ง Serial.end() เป็นการปิดใช้งานการสื่อสารแบบอนุกรม
* คำสั่ง Serial.available() เป็นการตรวจสอบการรับข้อมูลจากการสื่อสารแบบอนุกรม
* คำสั่ง Serial.read() เป็นการอ่านข้อมูลจากการสื่อสารแบบอนุกรมที่เข้ามา
* คำสั่ง Serial.peek() เป็นการส่งกลับไบต์ต่อไปของข้อมูลการสื่อสารแบบอนุกรม
* คำสั่ง Serial.print() เป็นการพิมพ์ข้อมูลไปยังพอร์ตอนุกรม
* คำสั่ง Serial.println() เป็นการพิมพ์ข้อมูลไปยังพอร์ตอนุกรม และขึ้นบรรทัดใหม่
* คำสั่ง Serial.write() เป็นการส่งข้อมูลไบต์ไปยังพอร์ตอนุกรม

**4.2 ไมโครคอนโทรลเลอร์**

ไมโครคอนโทรลเลอร์ (Microcontroller) คืออุปกรณ์ควบคุมขนาดเล็กซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียูหน่วยความจำและพอร์ต โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกันสามารถนำมาประยุกต์ใช้งานได้หลากหลายผ่านการออกแบบวงจรให้เหมาะกับงานต่างๆและสามารถโปรแกรมคำสั่งเพื่อควบคุมขา Input / Output เพื่อสั่งงานให้ไปควบคุมอุปกรณ์ต่างๆ เป็นระบบที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย ทั้งทางด้าน Digital และ Analog ยกตัวอย่างเช่น ระบบสัญญาณตอบระบบอัตโนมัติ, ระบบบัตรคิว, ระบบตอกบัตรพนักงานและอื่นๆ

**4.2.1 หน้าที่ส่วนต่างๆของไมโครคอนโทรลเลอร์**

1. หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือหน่วยความจำที่มีไว้สำหรับเก็บ โปรแกรมหลัก (Program Memory) เช่น Flash Memory ลักษณะการทำงานของหน่วยความจำนี้เป็น หน่วยความจำที่อ่าน-เขียนได้ด้วยไฟฟ้าเปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็น เหมือนกับกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยงในการทำงานข้อมูลจะหายไปคล้ายกับหน่วยความแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วๆไป แต่สำหรับ ไมโครคอนโทรลเลอร์สมัยใหม่หน่วยความจำข้อมูลมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มี ไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3.ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอกถือว่าเป็นส่วนที่สำคัญมากพอร์ตอินพุตรับสัญญาณเพื่อนำไปประมวลผลและส่งไปแสดงผลที่พอร์ตเอาต์พุต เช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณหรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณจำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดย แบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา เป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูงจังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงขึ้น การเขียนโปรแกรมไมโครคอนโทรลเลอร์ภาษาซีถือว่าเป็นภาษาระดับกลาง [5]

Arduino เป็นไมโครคอนโทรลเลอร์บอร์ดแบบสำเร็จรูปในยุคปัจจุบัน ซึ่งถูกสร้างมาจาก Controller ตระกูล ARM ของ ATMEL ข้อดีของไมโครคอนโทรลเลอร์บอร์ดคือเรื่องของ Open Source ที่สามารถนำไป พัฒนาต่อเป็นอุปกรณ์ต่างๆได้และความสามารถในการเพิ่ม Boot Loader เข้าไปที่ตัว ARM จึงท าให้การ Upload Code เข้าตัวบอร์ดสามารถทำได้ง่ายขึ้น และยังมีการพัฒนา Software ที่ใช้ในการควบคุมตัวบอร์ด ของ Arduino มีลักษณะเป็นภาษา C++ ที่โปรแกรมเมอร์มีความคุ้นเคยในการใช้งาน ตัวบอร์ดสามารถนำโมดูลมาต่อเพิ่มซึ่งทาง Arduino เรียกว่าเป็น shield เพื่อเพิ่มความสามารถเพิ่มขึ้น โดยกลุ่มของผู้จัดทำใช้บอร์ดรุ่น LGT8F328P

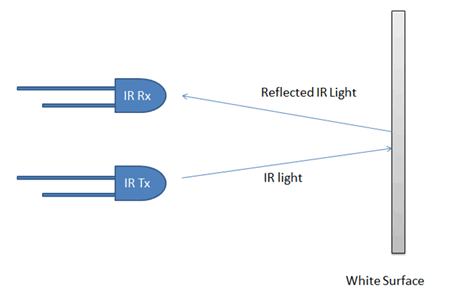
ตารางที่ 1 : รายละเอียดของ IR Infrared Obstacle Avoidance Sensor Module [6]

|  |  |
| --- | --- |
| ชิปไอซีไมโครคอนโทรลเลอร์ | LGT8F328P |
| ใช้แรงดันไฟฟ้า | 5V |
| รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ) | 7 – 12V |
| รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด) | 6 – 20V |
| พอร์ต Digital I/O | 14 พอร์ต (มี 6 พอร์ต PWM output) |
| พอร์ต Analog Input | 6 พอร์ต |
| กระแสไฟที่จ่ายได้ในแต่ละพอร์ต | 40mA |
| กระแสไปที่จ่ายได้ในพอร์ต 3.3V | 50mA |
| พื้นที่โปรแกรมภายใน | 16KB หรือ 32KB พื้นที่โปรแกรม, 500B ใช้โดย Booloader |
| พื้นที่แรม | 1 หรือ 2KB |
| พื้นที่หน่วยความจำถาวร (EEPROM) | 512B หรือ 1KB |
| ความถี่คริสตัล | 16MHz |
| ขนาด | 45x18 mm |

**4.3 โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุกีดขวาง**

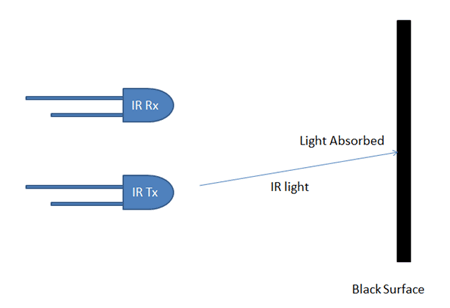
IR Infrared Obstacle Avoidance Sensor Module โดยโมดูลนี้มีตัวรับและตัวส่ง infrared ในตัว ตัวสัญญาณ(สีขาว) infrared จะส่งสัญญาณออกมา และเมื่อมีวัตถุมาบังคลื่นสัญญาณ infrared ที่ถูกส่งออกมาจะสะท้องกลับไปเข้าตัวรับสัญญาณ(สีดำ) สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้และสามารถปรับความไว ระยะการตรวจจับ ใกล้หรือไกลได้ ภายตัวเซ็นเซอร์แบบนี้จะมีตัวส่ง Emitter และ ตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้ไม่จำเป็นต้องเดินสายไฟทั้งสองฝั่งเหมือนแบบ Opposed Mode ทำให้การติดตั้งใช้งานได้ง่ายกว่า แต่อย่างไรก็ตามจำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เองโดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อน และ ทำให้ทำงานผิดพลาดได้

เซ็นเซอร์แบบนี้จะมีช่วงในการทำงาน หรือ ระยะในการตรวจจับจะได้ใกล้กว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง  Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับแผ่นสะท้อน Reflector อยู่ตลอดเวลาจะแสดงค่าเป็น **0**



รูปที่ 1 : หลักการทำงานของเซนเซอร์

หน้าที่หลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้จะแสดงค่าเป็น **1**



รูปที่ 2 : ลักษณะการทำงานแบบ Dark On หรือ Dark Operate

ซึ่งจะทำให้วงจรภายในรับรู้ได้ว่า มีวัตถุหรือชิ้นงานขวางอยู่ ทำให้สถานะของเอาท์พุตของตัวรับเปลี่ยนแปลงไป โดยเราเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate

ตารางที่ 2 : รายละเอียดของ IR Infrared Obstacle Avoidance Sensor Module [7]

|  |  |
| --- | --- |
| ไฟเลี้ยง VCC | 3.3-5Vdc |
| ดิจิตอลเอาต์พุต | 0 หรือ 1 |
| ระยะตรวจจับ สามารถปรับได้ตั้งแต่ | สามารถปรับได้ตั้งแต่ 2-30 cm |
| มุมในการตรวจจับ | 35 องศา |
| ขนาดบอร์ด | 3.1 x 1.5 cm |

**4.4 เซนเซอร์วัดระยะทาง**

เซนเซอร์วัดระยะทางด้วย Ultrasonic ใช้หลักการ ส่งคลื่นเสียงความถี่ต่ำ Ultrasonic ไปเมื่อคลื่นเสียงกระทบกับวัตถุจะมีการสะท้อนกลับมาเซนเซอร์จะจับเวลาที่ส่งคลื่นเสียงออกไปจนถึงคลื่นเสียงสะท้อนกลับมา เมื่อนำมาคำนวณกับเวลาที่เสียงเดินทางในอากาศแล้วจะได้ระยะทางออกมาโมดูล Ultrasonic ตรวจจับวัตถุ คำนวนระยะทางโดยใช้คลื่น มีลักษณะเป็นกรวยและไม่ใช่เส้นตรง จึงเหมาะสำหรับใช้ตรวจจับสิ่งกีดขวาง[8]



รูปที่ 3 : ตัวอย่างเซนเซอร์ US-205

|  |  |
| --- | --- |
| Supply voltage : | 2-10V |
| Signal input voltage: : | 1.8-7V |
| Max output current : | 3A (1.5A\*2) |
| Control signal: | PWM |

ตารางที่ 3 : คุณสมบัติของ US-025 [9]

**4.5 DC Motor Speed Control**

ประกอบด้วย 2 ส่วนใหญ่ ๆ คือ 1) H-bridge Driver และ 2) Pulse-width modulation (PWM) สิ่งที่เป็นพลังขับเคลื่อนหลักให้กับหุ่นยนต์นั้นคือ มอเตอร์ ซึ่งต้องการการควบคุม จาก Motor Driver ที่จะมาควบคุม ทั้ง ทิศทาง และ ความเร็ว ของมอเตอร์ซึ่งทางผู้จัดทำได้เลือกใช้ L298N Dual H-Bridge Motor Controllerคุณสมบัติของ L298N Dual H-Bridge Motor Controller ดังตารางที่ 4 โดยหลักการทำงาน

วงจร H-Bridge ของ L298N จะขับกระแสเข้ามอเตอร์ ตามขั้วที่กำหนดด้วยลอจิกเพื่อควบคุมทิศทาง ส่วนความเร็วของมอเตอร์นั้นจะถูกควบคุมด้วย สัญญาณ PWM (Pulse Width Modulation) เป็นวิธีการควบคุมการจ่ายกำลังโดยการปรับความกว้างของสัญญาณ Pulse ด้วยความถี่สูงเพื่อให้ได้กำลังเฉลี่ยเป็นไปตามส่วนที่ต้องการ ซึ่งต้องมีการปรับความถี่ให้เหมาะสมกับเป็นพารามิเตอร์ที่ใช้กำหนดสัดส่วนการทำงาน (ON) ของ Load (มอเตอร์)

รูปภาพประกอบด้วย ภาพหน้าจอ, จอภาพ, คอมพิวเตอร์, นั่ง

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 4 : ส่วนประกอบของ L298N Dual H-Bridge Motor Controller

|  |  |
| --- | --- |
| Operating Voltage | DC 3V-5.5V |
| Working current | 5.3 mA |
| Operating temperature | -40 °C-85°C |
| output method | GPIO |
| Induction angle | Less than 15 degrees |
| Detection distance | 2cm-600cm |
| Detection accuracy | 0.1cm+1% |

ตารางที่ 4 : คุณสมบัติ L298N Dual H-Bridge Motor Controller

**4.6 TCRT5000 Infrared Reflective sensor**

เป็นโมดูลตรวจจับวัตถุระยะใกล้ มีราคาถูก ขนาดเล็กสะดวกในการนำไปใช้ติดตั้งกับงานจำพวกหุ่นยนต์, Smart car, หุ่นยนต์หลบสิ่งกีดขวาง เป็นต้น ซึ่งสามารถแสดงได้ดังรูปที่ 8 และคุณสมบัติดังตารางที่ 5 โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณเป็นแสงอินฟราเรดออกไปตกกระทบกับวัตถุที่ตรวจพบในระยะ และทำการสะท้อนกลับมายังตัวหลอดโฟโต้ไดโอดที่ทำหน้าที่รับแสงอินฟราเรด โดยส่วนมาก ตัวโมดูลจะให้ค่า output ออกมาเป็น Digital signal แต่สำหรับบางโมดูอาจจะรองรับ output แบบ Analog signal ด้วย ส่วนตัว R ปรับค่านั้นใช้ในการปรับความไวต่อการตรวจจับแสงอินฟราเรด ซึ่งจะส่งผลต่อระยะในการตรวจพบวัตถุของตัวเซนเซอร์

รูปภาพประกอบด้วย วงจร

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 5 : แสดงส่วนประกอบของ TCRT5000 Infrared Reflective sensor

|  |  |
| --- | --- |
| ไฟเลี้ยง vcc | 3.3 – 5 V. |
| ดิจิติลเอาท์พุต | 0 หรือ 1 |

ตารางที่ 5 : คุณสมบัติเซ็นเซอร์วัดระยะทาง TCRT5000 Infrared Reflective sensor

**4.7 วงจร DC/DC Step-up (แรงดันปรับค่าได้)**

เป็นวงจรที่ทำหน้าที่เพิ่มแรงดันไฟฟ้า โดยที่ทางผู้จัดทำเลือกใช้เป็นรุ่น MT3608 สามารถแสดงส่วนประกอบได้ดังภาพที่ 9 และสามารถและคุณสมบัติดังตารางที่ 6 [10]

รูปภาพประกอบด้วย วงจร

คำอธิบายที่สร้างโดยอัตโนมัติ

รูปที่ 6 : ส่วนประกอบของ DC/DC Step-up รุ่น MT3608

|  |  |
| --- | --- |
| กระแสไฟขาออกสูงสุด(I max) : | 2 A |
| แรงดันไฟฟ้าเข้า (V in): | 2 V ~ 24 V |
| แรงดันขาออกสูงสุด(V 0ut max) | 28 V |
| ประสิทธิภาพ (%Eff) | > 93% |

ตารางที่ 6 : คุณสมบัติของ DC/DC Step-up รุ่น MT3608

โดยกฎของโอห์มกระแสไฟฟ้านั่นวงจรไฟฟ้านั้น จะแปรผัน ตรงกับ แรงดันของแหล่งจ่ายไฟฟ้าแต่จะแปรผกผันกับค่า ความต้านทานในวงจรไฟฟ้า” ดังสมการ [11]

รูปภาพประกอบด้วย นาฬิกา

คำอธิบายที่สร้างโดยอัตโนมัติข้อควรระวังเมื่อแรงดันเพิ่ม กระแสไฟไฟฟ้าจะต้องลดลงดังตัวอย่างที่แสดงในรูปที่ 7

รูปที่ 7 : ตัวอย่างความสัมพันธ์ระหว่างแรงดันเพิ่มกับกระแสไฟลดลง

**บทที่ 5**

**วิธีการดำเนินงาน**

**5.1อุปกรณ์ circuit**

1. Arduino board รุ่น LGT8F328P จำนวน 1 ชิ้น

2. 18650 Li-ion battery, Battery case, Li-ion Charging module จำนวน 1 ชิ้น

3. DC Geared-Motors จำนวน 2 ก้อน

4. H-bridge Driver จำนวน 1 ชิ้น

5. Breadboard จำนวน 1 ชิ้น

6. DC/DC Step-up Converter จำนวน 1 ชิ้น

7. TCRT5000 Infrared Reflective sensor จำนวน 1 ชิ้น

8. IR Infrared Obstacle Avoidance Sensor จำนวน 3 ชิ้น

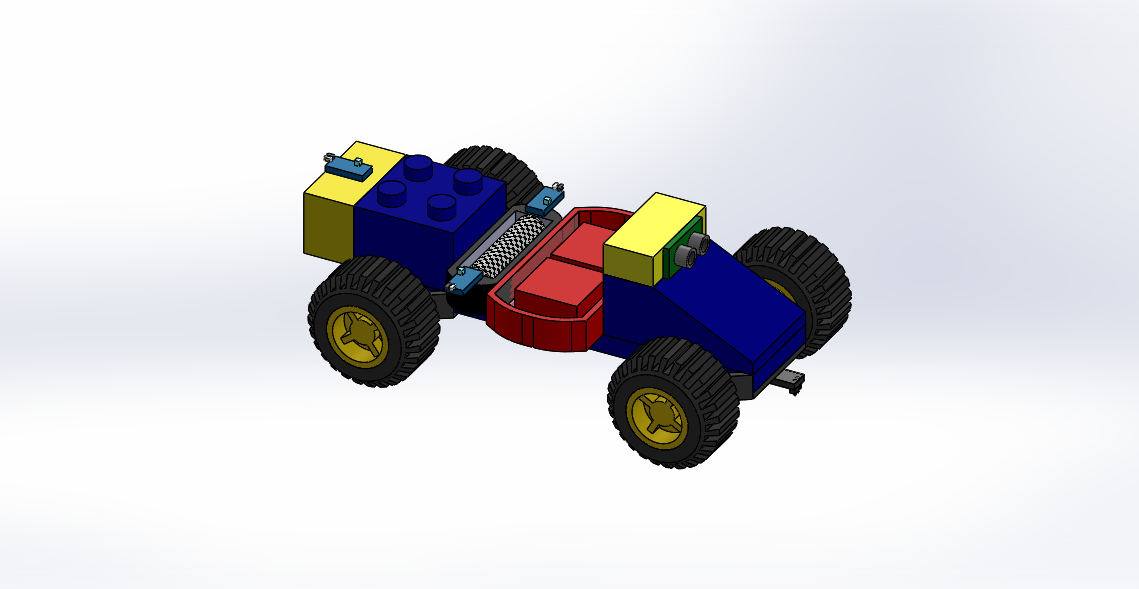
9. Ultrasonic Sensor จำนวน 1 ชิ้น

10. LED จำนวน 1 ชิ้น

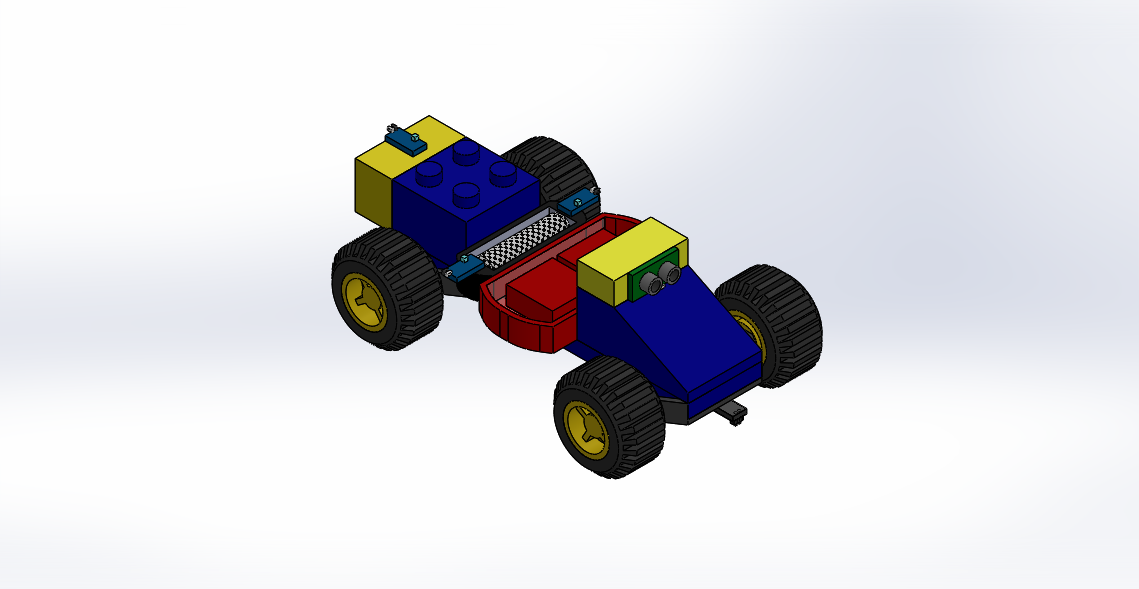
11. ตัวต้านทาน จำนวน 2 ชิ้น

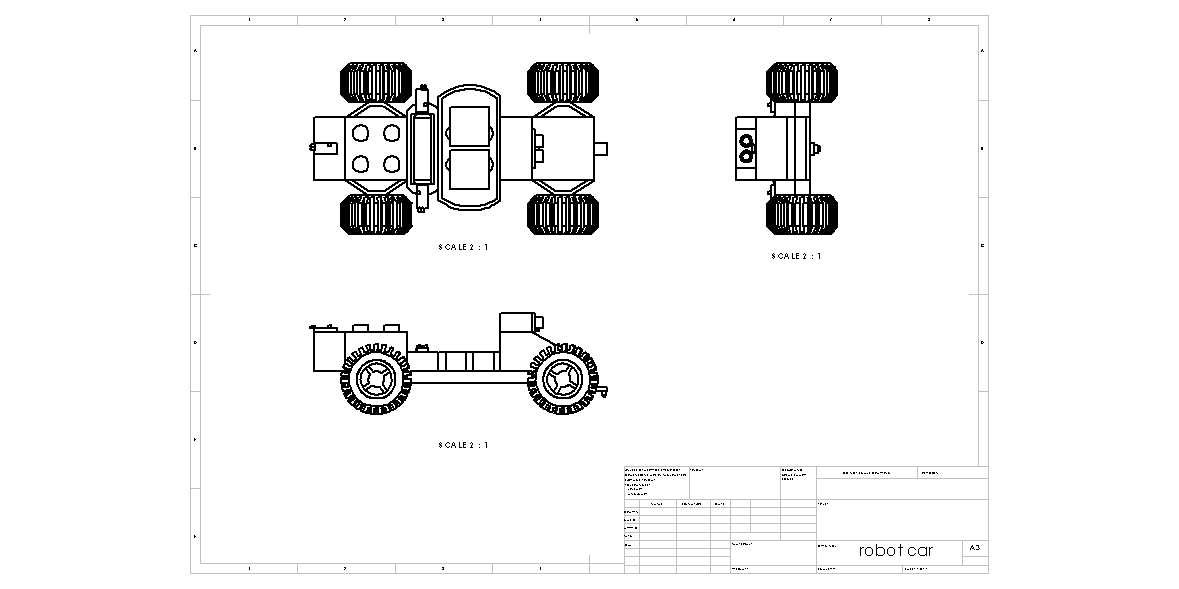
12. Switch จำนวน 1 ชิ้น

**5.2 แบบจำลองหุ่นยนต์**

****

รูปที่ 8 : ภาพ dimetric ของแบบจำลองหุ่นยนต์รถ



**อธิบายส่วนประกอบต่างๆตามหมายเลข**

รูปที่ 10 : drawing แบบจำลองรถยนต์

รูปที่ 9 : ภาพ isometric ของแบบจำลองหุ่นยนต์รถ

**หมายเลข 1 :** ที่ใส่บอร์ด 2 ชิ้นก็คือ Arduino board รุ่น LGT8F328P และ Breadboard

**หมายเลข 2 :** 18650 Li-ion batteryพร้อมรางถ่าน

**หมายเลข 3 :** IR Infrared Obstacle Avoidance Sensor

จะเห็นได้ว่ามี 3 ชิ้นโดยติดข้างรถบริเวณด้านขวา ด้านซ้าย และด้านหลัง โดย sensor ที่ติดฝั่งขวา เมื่อจับวัตถุฝั่งขวาในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 2 วิแล้วเลี้ยวรถไปฝั่งซ้ายหรือขวา ตามกลยุทธ์ sensor ที่ติดฝั่งซ้าย เมื่อจับวัตถุฝั่งซ้ายในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 2 วิแล้วเลี้ยวรถไปฝั่งขวาหรือซ้าย ตามกลยุทธ์ และ sensor ที่ติดหลังรถ หากจับเส้นสีแดง เมื่อพบแล้วไปชะลอ 3 วิ เพื่อเตรียมถอยหลังกลับหรือกลับรถ

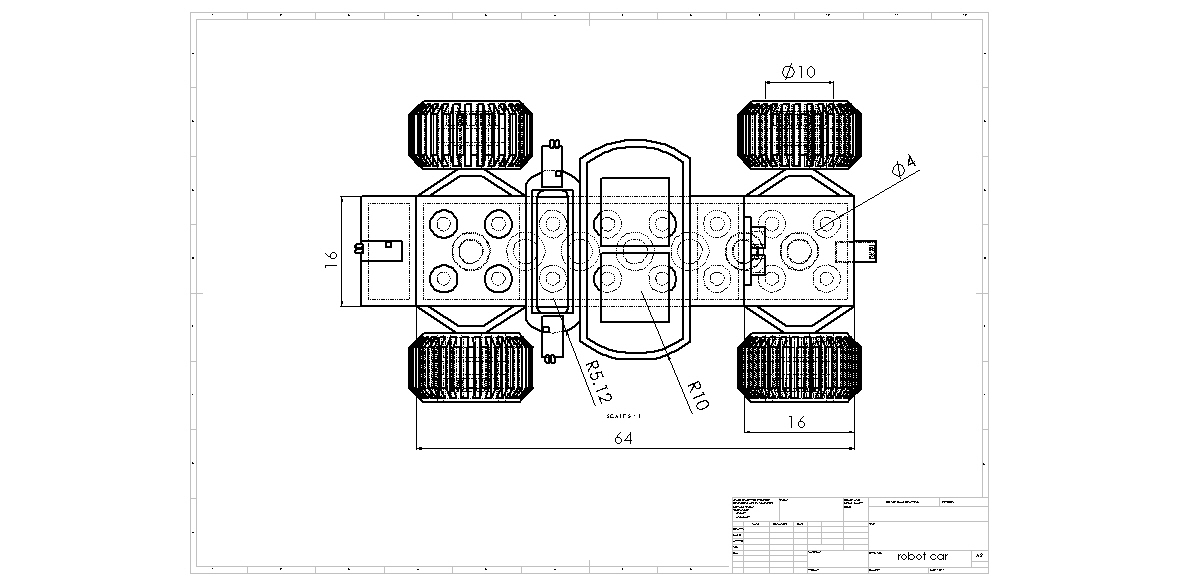
**หมายเลข 4 :** Ultrasonic Sensor

ใช้ Ultrasonic Sensor มาเป็น เซนเซอร์วัดระยะทาง

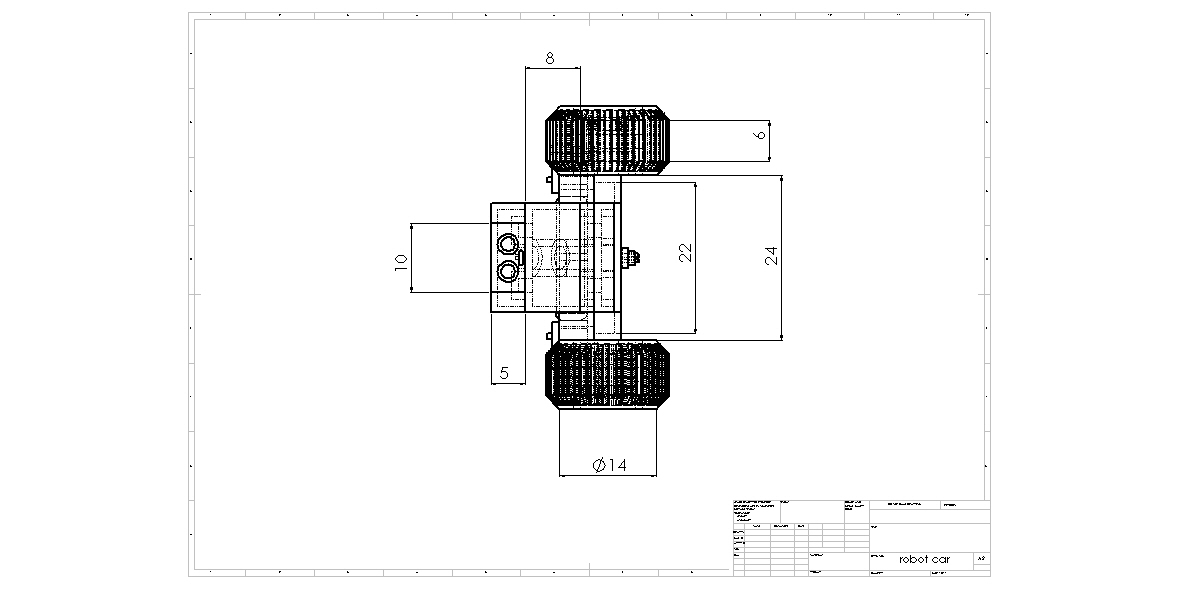
**หมายเลข 5 :** TCRT5000 Infrared Reflective sensor

จะติดบริเวณด้านหน้ารถใต้ท้องรถ sensor ตัวนี้จะทำหน้าที่ : จับวัตถุข้างหน้าในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 3 วิแล้วหยุดหรืออาจจะไปต่อตามคำสั่ง

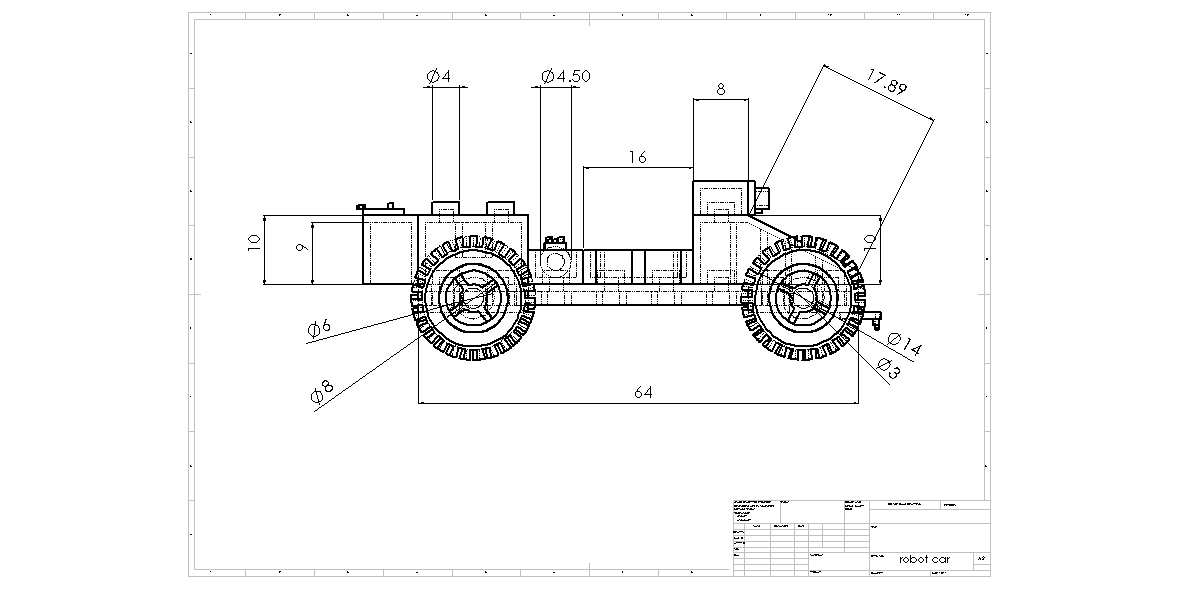
**รูป drawing อัตราส่วน 5:1 เพื่อระบุขนาดของแบบจำลองรถ**

****

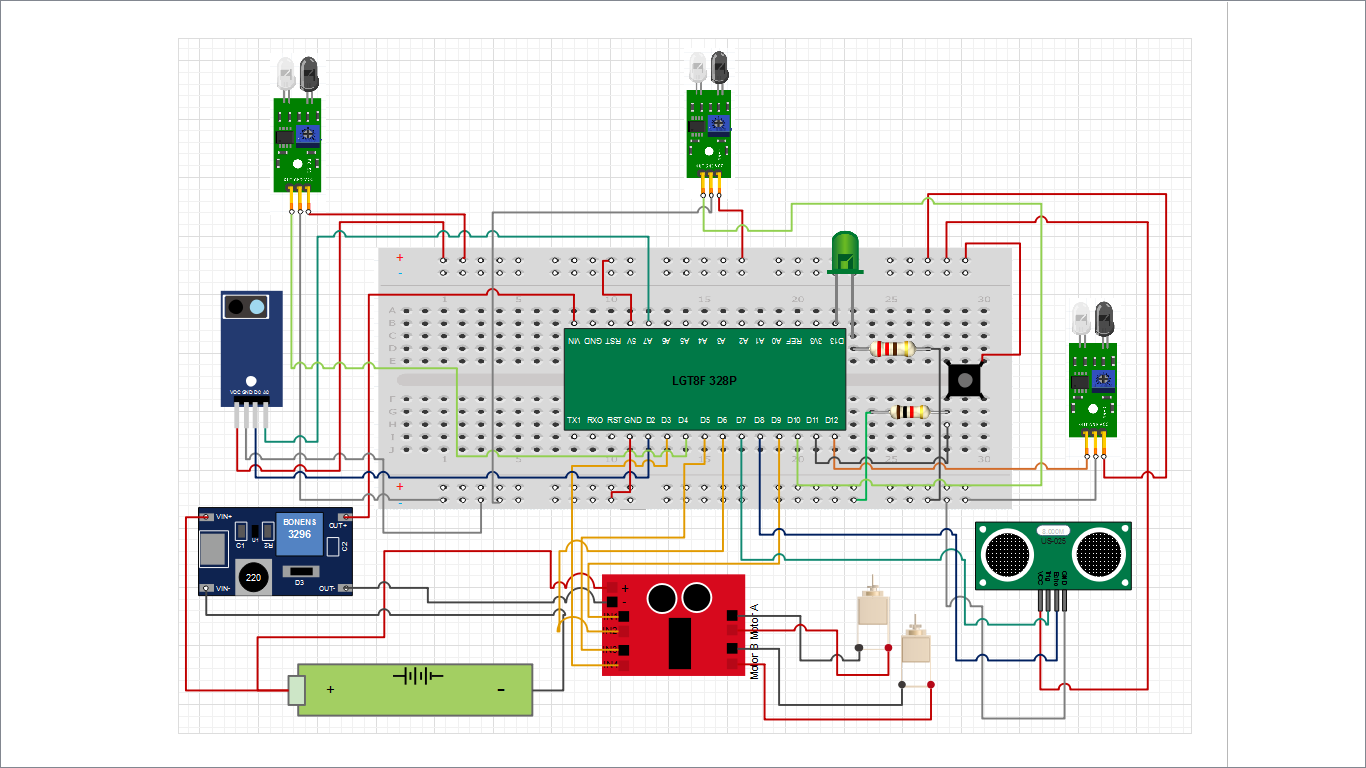
รูปที่ 11 : รูปด้าน TOP

****

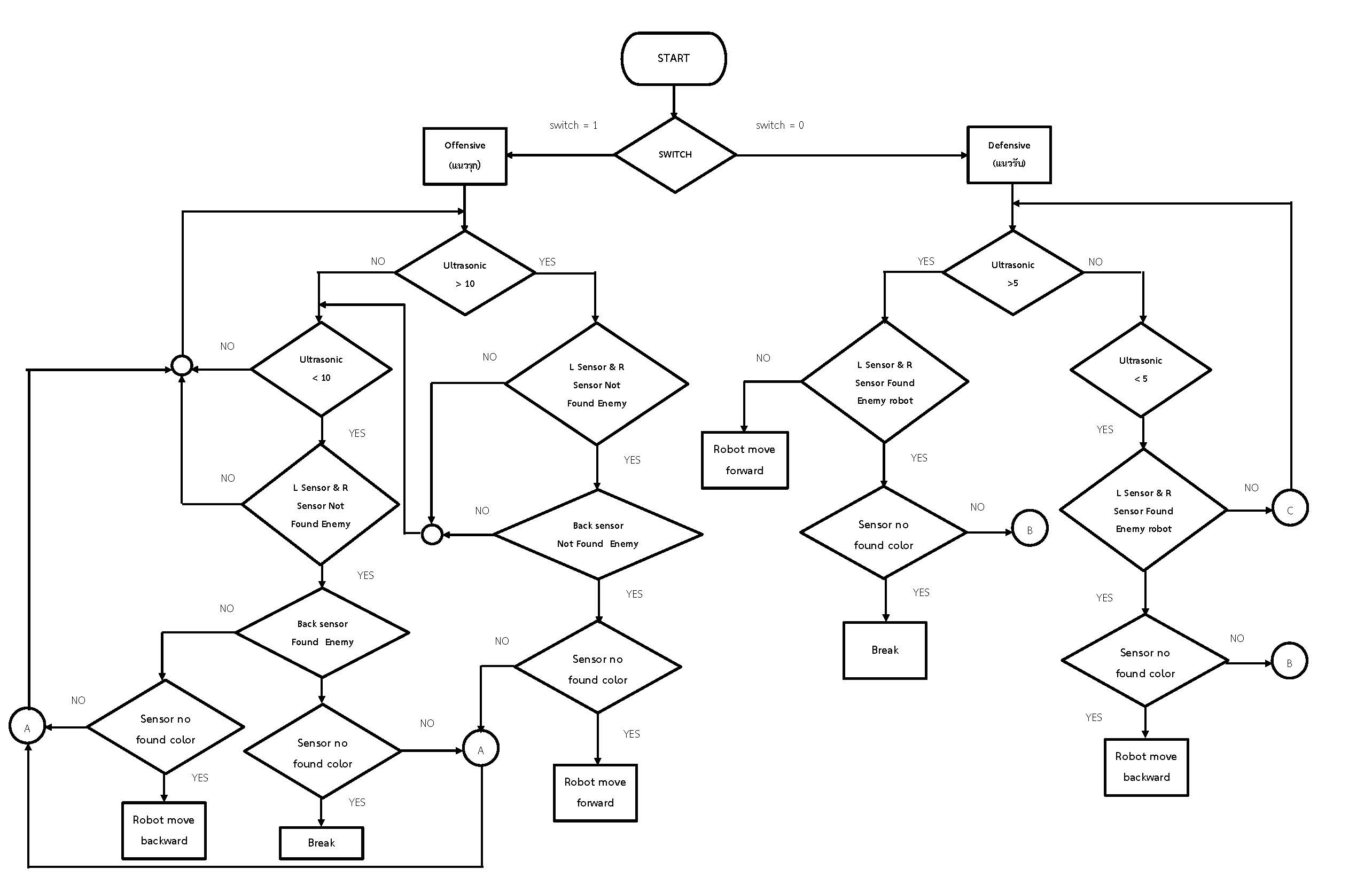
รูปที่ 12 : รูปด้าน FRONT

****

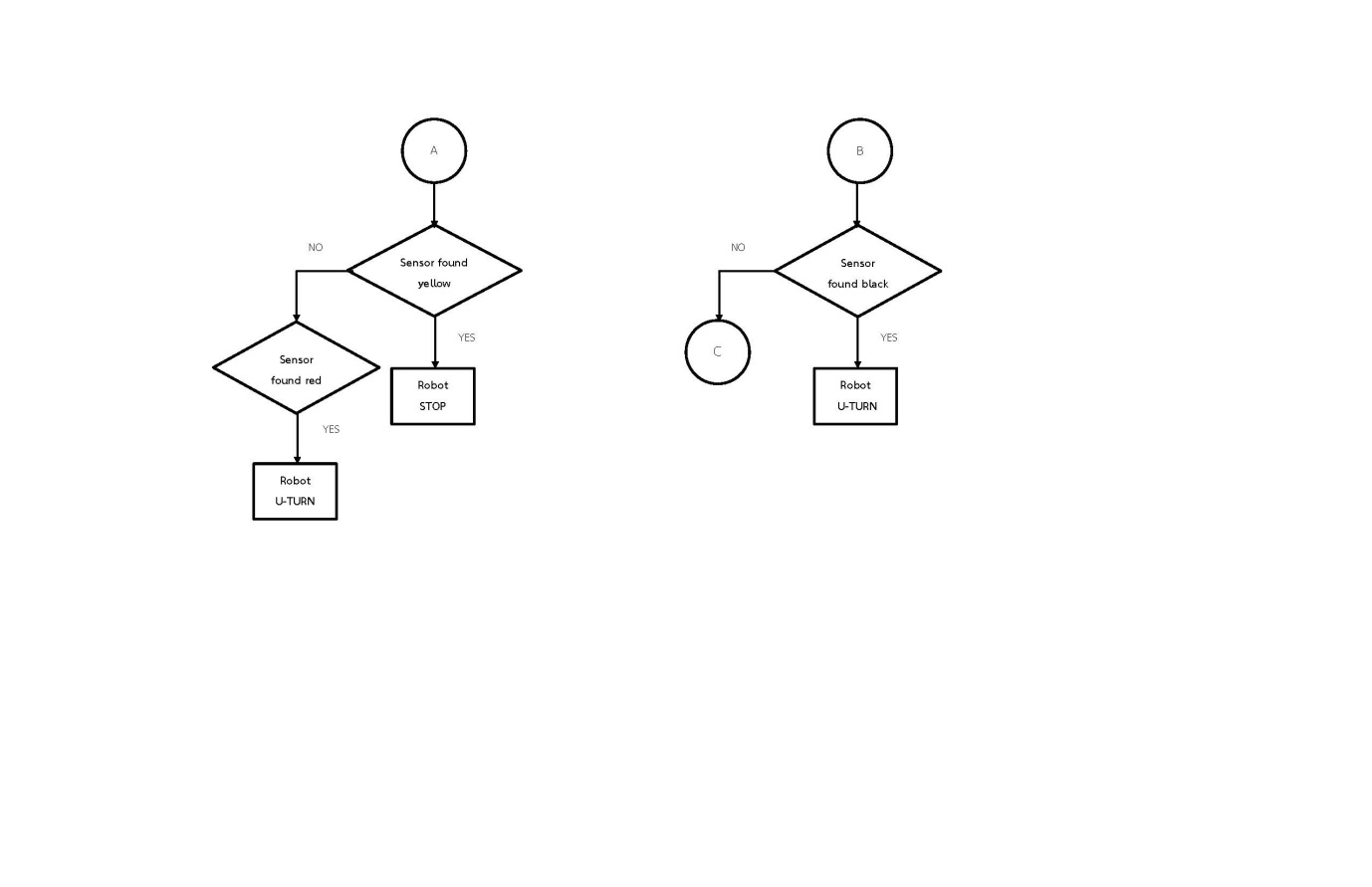
รูปที่ 13 : รูปด้าน SIDE

****

รูปที่ 14 : การต่อวงจรภายในหุ่นยนต์

****

รูปที่ 15 : รูป Flowchart

****

รูปที่ 16 : รูป Flowchart (ต่อ)

**5.3 คำสั่งที่ใช้งาน**

#define A1 D9 //moter A

#define A2 D6 //moter A

#define leftsensor D12 //sensorซ้าย

#define rightsensor D4 //sensorขวา

#define backsensor D10 //sensorหลัง

#define RED 1

#define YELLOW 2

#define BLACK 3

#define NOCOLOR 0

#define maxSpd 255 // motor max speed

#include <HCSR04.h>

HCSR04 hc(D7,D8);

int analogPin = A7;

int led = D13;

int switchPin = D11;

int val=0;

int old\_val=0;

int state=0;

int color=0;

int getColor() {

int NO\_color = analogRead(analogPin);

if ((NO\_color>2500)&&(NO\_color<2900))

return YELLOW ;

else if ((NO\_color>3000)&&(NO\_color<3500))

return RED ;

else if ((NO\_color>3600)&&(NO\_color<3900))

return BLACK;

else

return NOCOLOR;

}

void setup() {

pinMode(leftsensor, INPUT);

pinMode(rightsensor, INPUT);

pinMode(switchPin, INPUT);

pinMode(led, OUTPUT);

pinMode(A1, OUTPUT);

pinMode(A2, OUTPUT);

pinMode(B1, OUTPUT);

pinMode(B2, OUTPUT);

Serial.begin(115200);

}

void loop()

{

color = getColor();

val = digitalRead(switchPin);

if( (val==HIGH) && (old\_val==LOW))

{

state=!state;

}

old\_val=val;

if (state==1) //เมื่อกดสวิทซ์ 1 ครั้ง ใช้กลยุทธ์รุก

{

digitalWrite(led, HIGH);

if((hc.dist()>10)&&((digitalRead(leftsensor)==HIGH)||(digitalRead(rightsensor)==HIGH))&&(color == NOCOLOR)) // เดินหน้า

{

MotorAForward(maxSpd);

MotorBForward(maxSpd);

}

if((hc.dist()<10)&&((digitalRead(leftsensor)==HIGH)||(digitalRead(rightsensor)==HIGH))&&(digitalRead(backsensor) == LOW)&&(color == NOCOLOR)) // หยุด

{

MotorABreakTime(500);

MotorBBreakTime(500);

}

if((hc.dist()<10)&&((digitalRead(leftsensor)==LOW)||(digitalRead(rightsensor)==LOW))&&(digitalRead(backsensor) == HIGH)&&(color == NOCOLOR)) // เดินถอยหลัง

{

MotorARewardTime(3000);

MotorBRewardTime(3000);

}

if(color == YELLOW) // รถหยุด

{

MotorAStop();

MotorBStop();

}

if(color == RED) // กลับรถ

{

MotorAForwardTime(3000);

MotorBRewardTime(3000);

}

}

else

{

digitalWrite (led,LOW);

if((hc.dist()>5)&&((digitalRead(leftsensor)==HIGH)||(digitalRead(rightsensor)==HIGH))&&(color == NOCOLOR)) // ถอยหลัง

{

MotorAReward(maxSpd);

MotorBReward(maxSpd);

}

if((hc.dist()<5)&&((digitalRead(leftsensor)==HIGH)||(digitalRead(rightsensor)==HIGH))&&(color == NOCOLOR)) // เดินหน้า

{

MotorAForwardTime(3000);

MotorBForwardTime(3000);

}

if((digitalRead(leftsensor)==LOW)||(digitalRead(rightsensor)==LOW)&&(color == NOCOLOR)) // หยุด

{

MotorABreakTime(500);

MotorBBreakTime(500);

}

if(color == BLACK) // กลับรถ

{

MotorAForwardTime(5000);

MotorBRewardTime(5000);

}

}

delay(20);

}

void MotorAStop()

{

digitalWrite(A1, LOW);

digitalWrite(A2, LOW);

}

void MotorABreak()

{

digitalWrite(A1, HIGH);

digitalWrite(A2, HIGH);

}

void MotorBStop()

{

digitalWrite(B1, LOW);

digitalWrite(B2, LOW);

}

void MotorBBreak()

{

digitalWrite(B1, HIGH);

digitalWrite(B2, HIGH);

}

void MotorAForward(int speed)

{

digitalWrite(A2, LOW);

analogWrite(A1, speed);

}

void MotorBForward(int speed)

{

digitalWrite(B2, LOW);

analogWrite(B1, speed);

}

void MotorAReward(int speed)

{

digitalWrite(A1, LOW);

analogWrite(A2, speed);

}

void MotorBReward(int speed)

{

digitalWrite(B1, LOW);

analogWrite(B2, speed);

}

void MotorARewardTime(int time)

{

digitalWrite(A1, LOW);

analogWrite(A2, maxSpd);

delay (time);

}

void MotorBRewardTime(int time)

{

digitalWrite(B1, LOW);

analogWrite(B2, maxSpd);

delay (time);

}

void MotorAForwardTime(int time)

{

digitalWrite(A2, LOW);

analogWrite(A1, maxSpd);

delay (time);

}

void MotorBForwardTime(int time)

{

digitalWrite(B2, LOW);

analogWrite(B1, maxSpd);

delay (time);

}

void MotorABreakTime(int time)

{

digitalWrite(A1, HIGH);

digitalWrite(A2, HIGH);

delay (time);

}

void MotorBBreakTime(int time)

{

digitalWrite(B1, HIGH);

digitalWrite(B2, HIGH);

delay (time);

}

**บทที่ 6**

**แผนการดำเนินงาน**

**6.1 ทีมงาน ความรับผิดชอบของสมาชิกในทีมงาน**

1. นางสาวพัชรินทร์ แย้มเดช รหัสนักศึกษา 60010684 : ออกแบบตัวรถ,Flow Chart
2. นางสาวภูริพิชญ์ ธรรมโม รหัสนักศึกษา 60010811 : ต่อวงจร,เขียนโค้ด
3. นางสาวสิริวิมล มีทอง รหัสนักศึกษา 60011075 : รายงาน,เขียนโค้ด

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **รายการ** | **W 1-2** | **W 3-4** | **W 5-6** | **W 7-8** | **W 9-10** | **W 11-12** | **W 13-14** | **W 15** |
| เรียนรู้เนื้อหา |  |  |  |  |  |  |  |  |
| เรียนรู้การสร้างหุ่นยนต์  -3D Printing  -การขับเคลื่อน  -Arduino  -Program Technique  -Sensor interface |  |  |  |  |  |  |  |  |
| ทดสอบ |  |  |  |  |  |  |  |  |
| แข่งขัน |  |  |  |  |  |  |  |  |

**6.2 ตารางแผนการทำงาน**

(เนื่องจากสถานการณ์ Covid 19 ทำให้แผนการดำเนินงานที่ได้วางแผนไว้ไม่เป็นไปตามแผน)

**บทที่ 7**

**7.1 งบประมาณ**

ล้อ 2 คู่ ราคา คู่ละ 65 บาท ราคา 130 บาท

Sensor ราคา 85 บาท

DC Geared-Motors 2 ตัว ราคา 100 บาท

ถ่านชาร์จ Li-ion 18650 ขนาด 3400 mAh 3.7 V ราคา 55 บาท

ที่ชาร์จถ่าน ราคา 25 บาท

สายไฟ จำนวน 40 เส้น ราคา 40 บาท

โครงรถ ราคา 100 บาท

**รวมทั้งสิ้น เป็นเงิน 535 บาท**

**บทที่ 8**

**8.1 สิ่งที่คาดว่าจะได้รับ**

สามารถจำลองหุ่นยนตร์จากโปรแกรม Solid works ออกแบบหุ่นยนต์ สามารถใช้ โปรแกรม Fritzing ออกแบบวงจรให้กับ Arduino และมีการประกอบหุ่นยนต์ขึ้นมาร่วมกับการออกแบบโปรแกรมเพื่อใช้ในการเคลื่อนไหวของหุ่นยนต์จากโปรแกรม Arduino เพื่อใช้ในการแข่งขันบอลลูนด่านและในระหว่างการทำงานใช้โปรแกรม Gitub ในการแชร์ข้อมูลเพื่อแสดงความคืบหน้าในระหว่างการทำงานภายในกลุ่ม

**บทที่ 9**

**9.1 สรุป**

จากการออกแบบหุ่นยนต์โดยสร้างในรูปแบบของรถที่ใช้ในการแข่งขันบอลลูนด่าน โดยตัวรถสามารถเคลื่อนที่ได้ในทิศทางไปข้างหน้า ข้างหลัง เลี้ยวขวา เลี้ยวซ้ายที่ออกแบบเงื่อนไขผ่านโปรแกรม Arduino ที่มีการใช้เซนเซอร์ติดรถ และเมื่อจับวัตถุฝั่งขวาในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 2 วิแล้วเลี้ยวรถไปฝั่งซ้ายหรือขวาตามกลยุทธ์ และ sensor ที่ติดหลังรถหากจับเส้นสีแดงเมื่อพบแล้วจะชะลอ 3 วิ เพื่อเตรียมถอยหลังกลับหรือกลับรถ ใช้ร่วมกับใช้ Ultrasonic Sensor เป็นเซนเซอร์วัดระยะทางและTCRT5000 Infrared Reflective sensorจะติดบริเวณด้านหน้ารถใต้ท้องรถ sensor ตัวนี้จะทำหน้าที่จับวัตถุข้างหน้าในระยะ 10 ซม เมื่อพบวัตถุจะชะลอ 3 วิแล้วหยุดหรืออาจจะไปต่อตามคำสั่ง นอกจากนี้มีการออกแบบลักษณะของหุ่นยนต์ผ่านโปรแกรม Solid Work และใช้โปรแกรม Github ในการเก็บบันทึกและแชร์ข้อมูลระหว่างในทีม แต่ไม่สามารถนำไปปฏิบัติจริงได้เนื่องจากสถานการณ์โควิด19จึงส่งผลให้ไม่สามารถมาทำการแข่งขันได้

**หมายเหตุ อาจารย์สามารถเปิดดูไฟล์งานได้ดังนี้**

PROJECT\_CIRCUIT จะมีไฟล์แผงวงจรที่ต่อและรูปที่ต่อสำเร็จแล้ว

PROJECT\_CODE จะมีไฟล์ code เป็น .ino และ flowchart การทำงาน

PROJECT\_REPORT ไฟล์รายงาน มีไฟล์ pdf และ doc

PROJECT\_SOLIDWORKS จะมีงาน solidworks ที่เป็นไฟล์ STL และ SLDASM และ PNG

**เอกสารอ้างอิง**

[1] Robotics Online, Industirail Robotics Book ข้อมูลจาก[www.robotics.org/robotics/robotics-online-free-resources](http://www.robotics.org/robotics/robotics-online-free-resources) (วันที่สืบค้น 24 มีนาคม 2563)

[2] จับตาเอเชียตะวันออก, หุ่นยนต์สำหรับการผลิตอัจฉริยะ (Smart Manufacturing) ตามนโยบาย Industry 4.0 ข้อมูลจาก <http://www.eastasiawatch.in.th/th/articles/politics-and-economy/642/> (วันที่สืบค้น 24 มีนาคม 2563)

[3] สำนักงานคณะกรรมการนโยบายเขตพัฒนาพิเศษตะวันออก, อุตสาหกรรมหุ่นยนต์ ข้อมูลจาก[www.eeco.or.th/industry/%E0%B8%AD%E0%B8%B8%E0%B8%95%E0%B8%AA%E0%B8%B2%E0%B8%AB%E0%B8%81%E0%B8%A3%E0%B8%A3%E0%B8%A1%E0%B8%AB%E0%B8%B8%E0%B9%88%E0%B8%99%E0%B8%A2%E0%B8%99%E0%B8%95%E0%B9%8C](http://www.eeco.or.th/industry/%E0%B8%AD%E0%B8%B8%E0%B8%95%E0%B8%AA%E0%B8%B2%E0%B8%AB%E0%B8%81%E0%B8%A3%E0%B8%A3%E0%B8%A1%E0%B8%AB%E0%B8%B8%E0%B9%88%E0%B8%99%E0%B8%A2%E0%B8%99%E0%B8%95%E0%B9%8C) (วันที่สืบค้น 24 มีนาคม 2563)

[4] บทเรียนคอมพิวเตอร์ออนไลน์, ไมโครคอนโทรลเลอร์ ข้อมูลจาก <https://sites.google.com/site/mikhorkhxnthorllexr1/chud-kha-sang> (วันที่สืบค้น 24 มีนาคม 2563)

[5] เอกสารประกอบการสอนวิชาไมโครคอนโทรลเลอร์เบื้องต้น ข้อมูลจาก<http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_1.pdf> (วันที่สืบค้น 24 มีนาคม 2563)

[6] สื่อการสอนออนไลน์วิชาเทคนิคการอินเตอร์เฟส, บทที่ 1 บอร์ด Arduino คืออะไร ข้อมูลจาก <https://sites.google.com/site/karanwinatktech/unit1> (วันที่สืบค้น 24 มีนาคม 2563)

[7] Robot Siam เรียนรู้การสร้างหุ่นยนตร์, IR Infrared obstacle avoidance sensor ข้อมูลจาก <https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html> (วันที่สืบค้น 24 มีนาคม 2563)

[8] Arduino all, เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04 ข้อมูลจาก<https://www.arduinoall.com/article/233/33-arduino-%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99arduino%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B0%E0%B8%97%E0%B8%B2%E0%B8%87%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2%E0%B9%80%E0%B8%8B%E0%B9%87%E0%B8%99%E0%B9%80%E0%B8%8B%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B0%E0%B8%97%E0%B8%B2%E0%B8%87-ultrasonic-module-hc-sr04> (วันที่สืบค้น 24 มีนาคม 2563)

[9] Arduino all, โมดูลวัดระยะทาง Ultrasonic US-025/US-026 ultrasonic ranging module ข้อมูลจาก[www.arduinoall.com/product/3245/%E0%B9%82%E0%B8%A1%E0%B8%94%E0%B8%B9%E0%B8%A5%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B0%E0%B8%97%E0%B8%B2%E0%B8%87-ultrasonic-us-025-us-026-ultrasonic-ranging-module](http://www.arduinoall.com/product/3245/%E0%B9%82%E0%B8%A1%E0%B8%94%E0%B8%B9%E0%B8%A5%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B0%E0%B8%97%E0%B8%B2%E0%B8%87-ultrasonic-us-025-us-026-ultrasonic-ranging-module) (วันที่สืบค้น 24 มีนาคม 2563)

[10] วงจร DC/DC Step-up (แรงดันปรับค่าได้) ข้อมูลจาก

<https://www.thaiconverter.com/category/3/dc-step-up> (วันที่สืบค้น 25 มีนาคม 2563)

[11] ความต้านทานไฟฟ้า ข้อมูลจาก <http://www.g-tech.ac.th/vdo/moterdoc/%E0%B8%A7%E0%B8%87%E0%B8%88%E0%B8%A3%E0%B9%84%E0%B8%9F%E0%B8%9F%E0%B9%89%E0%B8%B2%201/%E0%B8%9A%E0%B8%97%E0%B8%97%E0%B8%B5%E0%B9%88%202%20%E0%B8%A7%E0%B8%87%E0%B8%88%E0%B8%A3%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%B2%E0%B8%99%E0%B8%97%E0%B8%B2%E0%B8%99.pdf> (วันที่สืบค้น 25 มีนาคม 2563)