



รายงานวิชา Pre-Project รหัสวิชา 01216747

จัดทำโดย

นายปुरुณพนธ์ คำศรี	รหัสนักศึกษา 60010627
นางสาวยลรดี ศิริรักษ์	รหัสนักศึกษา 60010840
นายวิศิษฐ์ศักดิ์ บุญทอง	รหัสนักศึกษา 60010942

เสนอ

ผศ.ดร.อุดม จันทร์จรัสสุข

ภาคเรียนที่ 2 ปีการศึกษา 2562

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Pre-Project รหัสวิชา 01216747 ซึ่งประกอบไปด้วยส่วนของแนวคิดในการทำโปรเจค กลยุทธ์ในเชิงรุกและรับ วงจรไฟฟ้า การออกแบบตัวรถ และ Programming Code ในการควบคุมการทำงาน โดยทางคณะผู้จัดทำขอขอบคุณ ผศ.ดร.อุดม จันทร์จรัสสุข ในการให้คำปรึกษาและช่วยเหลือในการทำโปรเจคและรายงานฉบับนี้ให้สำเร็จไปได้ด้วยดี

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่ารายงานฉบับนี้จะมีประโยชน์แก่ผู้ที่สนใจไม่มากนักน้อย และหากรายงานฉบับนี้มีข้อผิดพลาดประการใด ทางคณะผู้จัดทำขออภัยไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

## สารบัญ

เรื่อง	หน้า
บทนำ	1
ปัญหาหรือโจทย์	1
แนวคิดในการแก้ปัญหา	2
ขอบเขตของโครงการ	2
แนวคิดและเบื้องหลังที่จำเป็นในการทำโครงการ	3
วิธีการดำเนินงาน	9
Flow Chart	13
Programming Codes	13
เอกสารอ้างอิง	21

## บทนำ

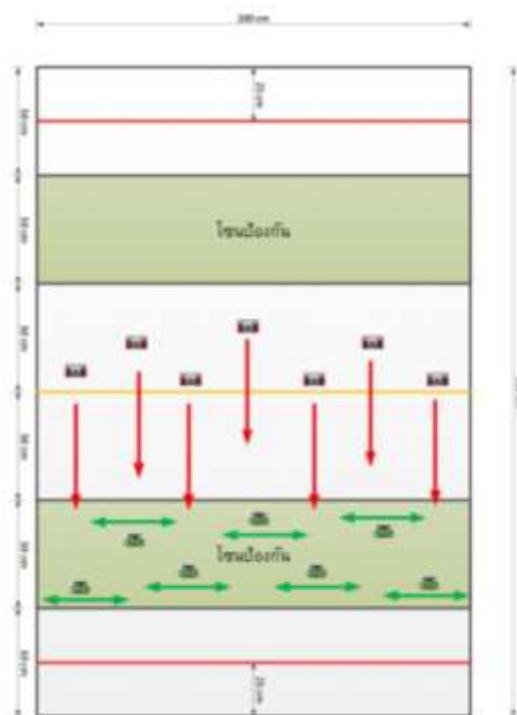
หุ่นยนต์ หรือ ไรบอต (robot) คือเครื่องจักรกลชนิดหนึ่ง มีลักษณะโครงสร้างและรูปร่างแตกต่างกัน หุ่นยนต์ในแต่ละประเภทจะมีหน้าที่การทำงานในด้านต่าง ๆ ตามการควบคุมโดยตรงของมนุษย์ การควบคุมระบบต่าง ๆ ในการสั่งงานระหว่างหุ่นยนต์และมนุษย์ สามารถทำได้โดยทางอ้อมและอัตโนมัติ โดยทั่วไป หุ่นยนต์ถูกสร้างขึ้นเพื่อสำหรับงานที่มีความยากลำบาก เช่น งานสำรวจในพื้นที่บริเวณแคบหรืองานสำรวจดวงจันทร์ ดาวเคราะห์ที่ไม่มีสิ่งมีชีวิต ในปัจจุบันเทคโนโลยีของหุ่นยนต์เจริญก้าวหน้าอย่างรวดเร็ว และเริ่มเข้ามา มีบทบาทกับชีวิตของมนุษย์ในด้านต่าง ๆ เช่น ด้านอุตสาหกรรมการผลิตแตกต่างจากเมื่อก่อนที่หุ่นยนต์มักถูกนำไปใช้ ในงานอุตสาหกรรมเป็นส่วนใหญ่ ปัจจุบันมีการนำหุ่นยนต์มาใช้งานมากขึ้น เช่น หุ่นยนต์ที่ใช้ในทางการแพทย์ หุ่นยนต์สำหรับงานสำรวจ หุ่นยนต์ที่ใช้งานในอวกาศ หรือแม้แต่หุ่นยนต์ที่ถูกสร้างขึ้นเพื่อเป็น เครื่องเล่นของมนุษย์ จนกระทั่งในปัจจุบันนี้ได้มีการพัฒนาให้หุ่นยนต์นั้นมีลักษณะที่คล้ายมนุษย์ เพื่อให้อาศัย อยู่ร่วมกันกับมนุษย์ ให้ได้ในชีวิตประจำวัน [1]

สำหรับสถาบันหุ่นยนต์แห่งสหรัฐอเมริกา (The Robotics Institute of America : RIA) จะจำแนกเป็น 2 ประเภทใหญ่ๆ ตามลักษณะการเคลื่อนที่ ดังนี้

1. หุ่นยนต์ชนิดติดตั้งอยู่กับที่ (Fixed Robot) หุ่นยนต์ประเภทนี้จะมีลักษณะเป็นแขนกล สามารถ ขยับและเคลื่อนที่ได้เฉพาะข้อต่อ นิยมใช้ในงานอุตสาหกรรม เช่น อุตสาหกรรมยานยนต์, การประกอบรถยนต์ เป็นต้น
2. หุ่นยนต์ชนิดเคลื่อนที่ได้ (Mobile Robot) หุ่นยนต์ประเภทนี้สามารถเคลื่อนที่ได้ด้วยตัวเอง โดย การใช้ล้อ ขา หรือการขับเคลื่อนแบบอื่นๆ เช่น โซเจอร์เนอร์ (Sojourner) หุ่นยนต์อุปกรณ์สำรวจดาวอังคาร ซึ่งมีลักษณะคล้ายรถ มีล้อ 6 ล้อ หนัก 11.5 กิโลกรัม โดยมากับยานอวกาศ Mar Pathfinder ซึ่งลงจอดบน ดาวอังคารได้สำเร็จในปี 1997 เป็นต้น [2]

## ปัญหาหรือโจทย์

การแข่งขันหุ่นยนต์มีลักษณะคล้ายกับการเล่นบอลลูกด้านหรือการเล่นเตย โดยแบ่งเป็น 2 ทีม คือ ทีมรุกและทีมรับสลับกันในการแข่งแต่ละรอบ โดยทีมหนึ่งจะประกอบด้วยหุ่นยนต์ 7 ตัว ทีมรุกจะต้องวิ่งไป ทางฝั่งตรงข้ามจนผ่านเส้นสีแดง แล้วกลับมาที่ฝั่งตัวเอง (ผ่านเส้นสีเหลือง) โดยที่ไม่ให้ทีมรับจับได้ ก็จะเป็น ฝ่ายชนะในการแข่งขันรอบนั้น หุ่นยนต์ที่ถูกจับได้จะถูกตัดออกจากการแข่งขันในรอบนั้น ส่วนทีมรับจะสามารถวิ่งสกัดกันฝ่ายตรงข้ามในพื้นที่ป้องกันเท่านั้น ถ้าวิ่งออกนอกพื้นที่ก็จะถูกตัดออกจากการแข่งขันใน รอบนั้นเช่นกัน ถ้าไม่มีหุ่นยนต์ตัวไหนสามารถผ่านด่านได้ ทีมรับจะเป็นฝ่ายชนะ การแข่งขันของแต่ละรอบ จะสิ้นสุดลงเมื่อทีมรุกสามารถผ่านด่านได้สำเร็จ หรือเมื่อทีมใดทีมหนึ่งไม่เหลือผู้เล่น



ภาพสนามแข่งขัน

## แนวคิดในการแก้ปัญหา

### กลยุทธ์เกมรุก

การเดินรถตัวรถจะมีเซ็นเซอร์ 5 ตัว คือ เซ็นเซอร์ใต้ท้องรถเพื่อจับว่าถึงเส้นที่กำหนดหรือยัง เซ็นเซอร์ด้านหน้า ด้านข้างซ้ายขวา และด้านหลังรถเพื่อตรวจจับรถคันอื่น เมื่อเซ็นเซอร์ด้านหน้าหรือด้านหลังตรวจจับรถคันอื่นได้รถจะหยุด เซ็นเซอร์ที่ใต้ท้องรถจะหาหน้าที่ตรวจจับเส้นสีแดงว่ารถได้ผ่านโซนป้องกันมาแล้ว หลังจากนั้นจะทำการถอยหลังและเมื่อเซ็นเซอร์จับเส้นเหลืองรถจะหยุด เป็นการสิ้นสุดการแข่งขัน

### กลยุทธ์เกมรับ

จะทำการตั้งรับฝ่ายตรงข้ามโดยการวิ่งเดินหน้าและถอยหลังในแนวนอน(แนวตรงข้ามกับเกมรับ) ใช้เซ็นเซอร์ใต้ท้องรถตรวจจับเส้นขอบสนาม เมื่อพบเส้นจะเปลี่ยนทิศการวิ่งเพื่อไม่ให้รถออกนอกสนาม

## ขอบเขตของโครงการ

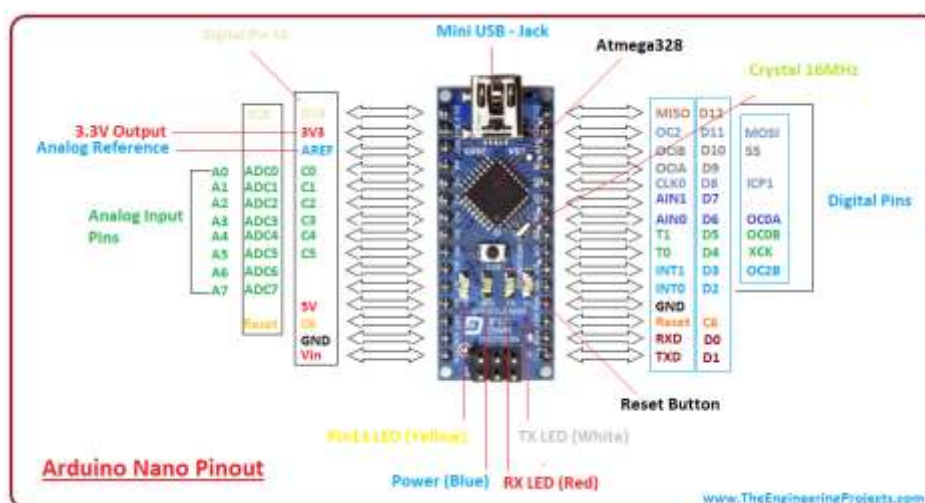
ขนาดกว้าง x ยาว : 10 x 10 เซนติเมตร

ความสูงไม่เกิน 13 เซนติเมตร

## แนวคิดและเบื้องหลังที่จำเป็นในการทำโครงการ

### Arduino board (LGT8F328P) [3]

Feature	LGT8F328P
DAC output	Yes (D4)
ADC	12bit (9 channel)
ADC Sampling rate Max.	500 KSPS
Analog Comparator	2
Unique ID	Yes
Internal reference resoltuion	±0.5%
PWM dead zone control	Yes
High current push - pull PWM	Yes
Computing Accelerator (DSC)	Yes
Stacking expansion system	Yes
Speed	32 Mhz
OUTPUT	27 pin
INPUT	30 pin



ภาพ Arduino Nano Pinout [4]

### 18650 Li-ion battery (3.7V 3400mAh)

แบตเตอรี่ 18650 ถูกเรียกตามเส้นผ่านศูนย์กลางและความยาวของตัว (18mm x 65mm) ด้านในตัวแบตเตอรี่เป็น cell ชนิด Lithium-Ion แบตเตอรี่ชนิดนี้ถูกใช้มาค่อนข้างนานแล้ว เพียงแต่อยู่ในรูปแบบของ Battery Pack จึงไม่ค่อยคุ้นเคยกันมากนัก แต่ในปัจจุบันเนื่องจากแบตเตอรี่ 18650 สามารถจ่ายกระแสได้สูงกว่าแบตเตอรี่ขนาด AA และความจุที่สูงกว่ามาก จึงเป็นที่นิยมแพร่หลายในการใช้กับอุปกรณ์อิเล็กทรอนิกส์ต่างๆ แต่ด้วยคุณสมบัติเฉพาะของแบตเตอรี่ชนิด Lithium นั้นไม่เหมือนแบตเตอรี่ธรรมดา เช่น Ni-mh หรือแบตเตอรี่แห้ง ตัวแบตเตอรี่เองต้องจึงมีการออกแบบระบบป้องกันให้มีความปลอดภัยมากขึ้น ซึ่ง

แต่ละ cell ปกติจะมีชุดป้องกันด้วยกัน 3 ชั้น โดยระบบป้องกันแบบ PCT และ CID ตามปกติจะมีอยู่ใน cell อยู่แล้ว ส่วน Protected PCB นั้นขึ้นอยู่กับผู้ผลิต



ภาพโครงสร้างภายในของ 18650 Li-ion battery

โครงสร้างภายในของแบตเตอรี่ 18650

1. PCT จะเป็นตัวป้องกันความร้อนของ Cell เกินพิกัด และตัดการทำงานของแบตเตอรี่ลง โดยจะสามารถกลับมาใช้งานได้เมื่ออุณหภูมิเข้าสู่สภาวะปกติ
2. CID นั้นจะเป็นวาล์วป้องกันความดันภายใน Cell เกินพิกัดจนอาจนำไปสู่การระเบิดได้ โดยวาล์วตัวนี้จะทำหน้าที่ตัดการทำงานของ Cell ถาวร ไม่สามารถคืนสภาวะกลับมาใช้ใหม่ได้อีก หากสังเกตที่ขั้วของแบตเตอรี่จะพบรูเล็กๆ ที่ถูกออกแบบไว้สำหรับระบายแก๊สหากมีแรงดันผิดปกติภายใน cell นั้นเอง
3. Protected PCB หรือเรียกกันว่า 18650 แบบมีวงจร ซึ่งเป็นวงจรอิเล็กทรอนิกส์ขนาดเล็กที่ฝังไว้ภายในตัวของตัวแบตเตอรี่ ทำหน้าที่คอยป้องกันการใช้กระแสเกิน (Over Current) ป้องกันแรงดันชาร์ตเกิน (Over Charge Voltage) และป้องกันการใช้ไฟในระดับโวลต์ที่ต่ำกว่ากำหนด (Over Discharge) โดยภายในจะมี IC ที่คอย ตรวจสอบอยู่ตลอดเวลาโดย IC ถูกออกแบบให้มีการกินกระแสน้อยมากๆ ในระดับไมโครแอมป์ ซึ่งแทบจะไม่ส่งผลกับปริมาณแบตเตอรี่เลย

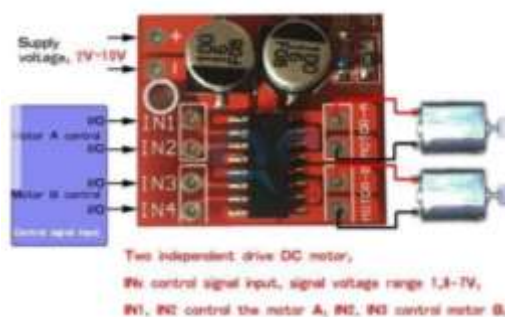
ระดับแรงดันใช้งานปกติของแบตเตอรี่ชนิด lithium จะอยู่ที่ 3.7V โดยแรงดันที่ชาร์จเต็มจะอยู่ที่ 4.2V เนื่องจากคุณสมบัติเฉพาะของ Cell ชนิด lithium นั้น หากมีถึงกระแสไฟจากแบตเตอรี่จนแรงดันต่ำกว่า 2.5V นั้นจะทำให้ cell เสียหายถาวร ไม่สามารถนำกลับมาใช้ใหม่ได้ (เครื่องชาร์จจะไม่ยอมชาร์ตหากแรงดันใน Cell ต่ำกว่าที่กำหนด) ฉะนั้น วงจรป้องกัน Protected PCB จึงเป็นสิ่งสำคัญอย่างมากในการใช้งานร่วมกับอุปกรณ์ทั่วไป โดยผู้ใช้ไม่ต้องกังวลว่าจะใช้งานจากแบตเตอรี่หมดจน cell พัง อีกทั้งหากมีการถึงกระแสเกินกำหนดหรือมีการใช้แรงดันชาร์ตเกินพิกัด วงจรจะทำการตัดการทำงานอัตโนมัติ ส่วน 18650 แบบไม่มีวงจร นิยมใช้กับอุปกรณ์อิเล็กทรอนิกส์ที่ถูกออกแบบมาเฉพาะซึ่งมีวงจรควบคุมอยู่ภายนอกแล้ว เช่น วงจร BMS หรือ PCM ซึ่งมีหลักการทำงานคล้ายกัน จึงไม่จำเป็นต้องมีวงจรป้องกันภายในตัว cell ทำให้ประหยัดต้นทุนในการผลิต มักพบเห็นใน Battery Pack [5]

## DC Geared-Motors

มอเตอร์ไฟฟ้ากระแสตรง (Direct current motor) หรือเรียกว่า ดี.ซี มอเตอร์ (D.C. MOTOR) เป็นเครื่องกลไฟฟ้ากระแสตรงที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล มีทั้งชนิดกระตุ้นฟิวด์จากภายนอก (Separated excited motor) และชนิดกระตุ้นฟิวด์ด้วยตัวเอง (Self excited motor) มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรม เพราะมีคุณสมบัติที่เด่นในด้านการปรับความเร็วรอบตั้งแต่ความเร็วรอบต่ำสุดไปจนถึงความเร็วรอบสูงสุด

หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง (Direct current motor) เมื่อแรงดันไฟฟ้ากระแสตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะแปร่งผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้น จนเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกันตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกันและทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ อาร์มาเจอร์ที่หมุนนี้เรียกว่า โรเตอร์ (Rotor) [6]

## H-bridge Driver



ภาพ H-bridge Driver

### คุณสมบัติ H-bridge Driver

Supply voltage : 2-10V

Signal input voltage : 1.8-7V

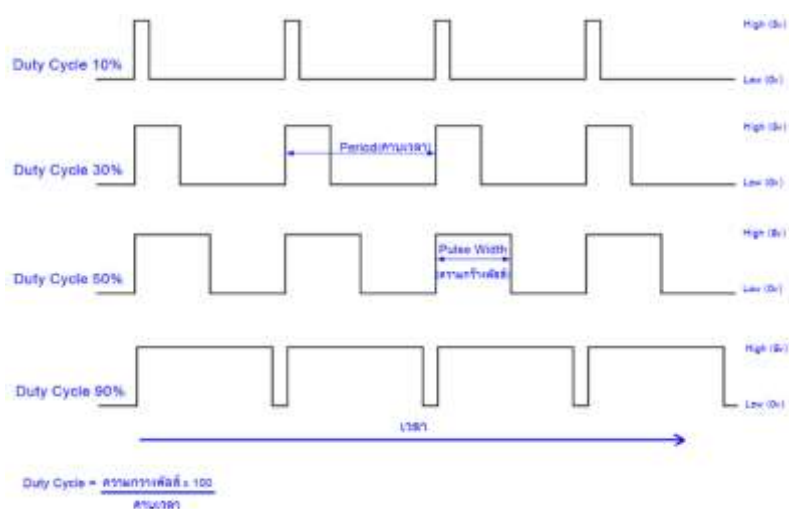
Max output current : 1.5A\*2

Control signal : PWM

PWM หมายถึง Pulse Width Modulation เป็นเทคนิคที่ Arduino ใช้ในการควบคุมวงจร และ เขียนค่าแบบ Analog ด้วยพอร์ต Digital คือ โดยปกติแล้ว พอร์ต Digital จะสามารถมีได้แค่ 2 สถานะ คือ HIGH (5V) กับ LOW (0V) เท่านั้น จึงทำให้สร้างค่าสัญญาณลอจิกได้เพียง เปิดหรือปิด (1 หรือ 0 , มีไฟหรือไม่มีไฟ) แค่นั้น

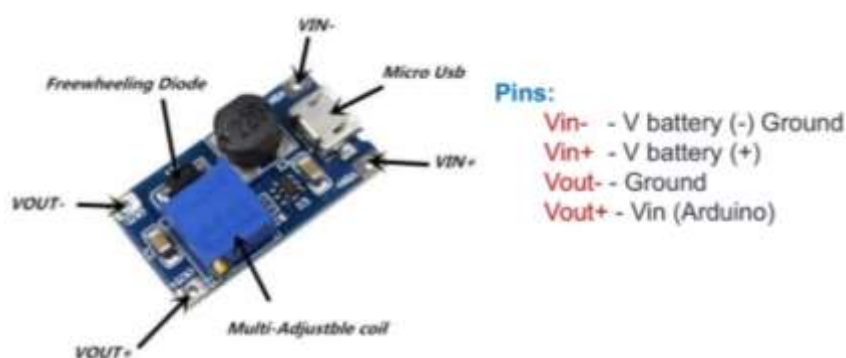


ซึ่งการใช้เทคนิค PWM นั้น จะเป็นการทำให้พอร์ตดิจิทัล สามารถเขียนค่าได้มากกว่า HIGH หรือ LOW โดยทำให้สามารถเขียนค่าเป็นแบบอนาล็อกได้ (อาจเป็น 0-255 หรือ 0-1023) โดยวิธีการนั้น จะใช้การปรับสถานะของสัญญาณลอจิก HIGH / LOW สลับกันไปมาด้วยคาบเวลาหนึ่งๆ โดยค่าที่ได้นั้นจะ ขึ้นอยู่กับ สัดส่วนเวลาของสัญญาณในช่วงเวลาที่มีสถานะเป็น HIGH กับช่วงเวลาที่ เป็น LOW โดย ช่วงเวลา ทั้งหมดที่สัญญาณมีสถานะเป็น HIGH นั้นเราจะเรียกว่าเป็น "ความกว้าง Pulse (Pulse Width)" โดยสัญญาณ พัลส์ เมื่อเทียบ % ของช่วงเวลาที่ เป็น HIGH (หรือก็คือ % ของ Pulse Width) กับ % ของคาบเวลา (Period) ของพัลส์ลูกนั้นๆ เรียกว่า Duty Cycle [7]



### DC/DC Step-up Converter

เป็นวงจรที่ทำหน้าที่เพิ่มแรงดันไฟฟ้า เนื่องจากถ่าน Li-ion ให้แรงดันที่ 3.7-4.2V ซึ่งไม่เพียงพอต่อการต้องการของบอร์ด Arduino



ภาพส่วนประกอบ DC/DC Step-up

ข้อควรระวัง เมื่อแรงดันเพิ่ม กระแสที่จ่ายได้จะลดลง



ถ้าประสิทธิภาพอยู่ที่ 80% กระแส output จะเหลือ 0.4A

### TCRT5000 Infrared Reflective sensor



ภาพ TCRT5000 Infrared Reflective sensor

เป็นโมดูลตรวจจับวัตถุระยะใกล้ มีราคาถูก ขนาดเล็ก สะดวกในการนำไปใช้ติดตั้งกับงานจำพวก หุ่นยนต์, Smart car, หุ่นยนต์หลบสิ่งกีดขวาง เป็นต้น โดยการทำงานของตัวโมดูลนี้ เริ่มต้นโดยให้ หลอด Infrared LED ทำการส่งสัญญาณ เป็นแสงอินฟราเรดออกไปตกกระทบกับวัตถุที่ตรวจพบในระยะ และทำการสะท้อนกลับมายังตัวหลอดโฟโตไดโอดที่ทำหน้าที่รับแสงอินฟราเรด

โดยส่วนมาก ตัวโมดูลจะให้ค่า output ออกมาเป็น Digital signal แต่สำหรับบางโมดูลอาจจะรองรับ output แบบ Analog signal ด้วย ส่วนตัว R ปรับค่านั้นใช้ในการปรับความไวต่อการตรวจจับแสงอินฟราเรด ซึ่งจะส่งผลต่อระยะในการตรวจพบวัตถุของตัวเซนเซอร์ [8]

### IR Infrared Obstacle Avoidance Sensor

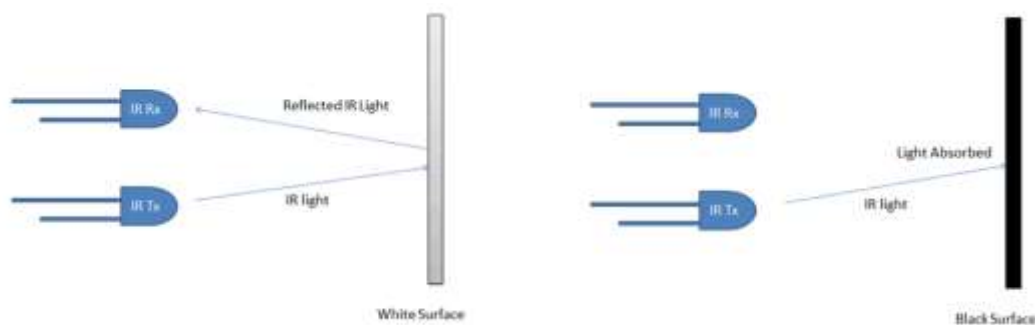


ภาพ IR Infrared Obstacle Avoidance Sensor

โมดูลเซ็นเซอร์แสงสำหรับตรวจจับวัตถุสีขาว IR Infrared Obstacle Avoidance Sensor Module โดยโมดูลนี้ จะมีตัวรับและตัวส่ง infrared ในตัว ตัวสัญญาณ(สีขาว) infrared จะส่งสัญญาณออกมา และเมื่อมีวัตถุมาบัง คลื่นสัญญาณ infrared ที่ถูกส่งออกมาจะสะท้อนกลับเข้าไปในตัวรับสัญญาณ (สีดำ) สามารถนำมาใช้ตรวจจับวัตถุที่อยู่ตรงหน้าได้ และสามารถปรับความไว ระยะการตรวจจับ ใกล้หรือไกลได้

ภายในตัวเซ็นเซอร์แบบนี้จะมีตัวส่ง Emitter และ ตัวรับ Receiver ติดตั้งภายในตัวเดียวกัน ทำให้ไม่จำเป็นต้องเดินสายไฟทั้งสองฝั่ง เหมือนแบบ Opposed Mode ทำให้การติดตั้งใช้งานได้ง่ายกว่า แต่อย่างไรก็ตามจำเป็นต้องติดตั้งตัวแผ่นสะท้อนหรือ Reflector ไว้ตรงข้ามกับตัวเซ็นเซอร์เอง โดยโฟโต้เซ็นเซอร์แบบที่ใช้แผ่นสะท้อนแบบนี้จะเหมาะสำหรับชิ้นงานที่มีลักษณะทึบแสงไม่เป็นมันวาว เนื่องจากอาจทำให้ตัวเซ็นเซอร์เข้าใจผิดว่าเป็นตัวแผ่นสะท้อน และ ทำให้ทำงานผิดพลาดได้

เซ็นเซอร์แบบนี้จะมีช่วงในการทำงาน หรือ ระยะในการตรวจจับจะได้ใกล้กว่าแบบ Opposed mode ซึ่งในสภาวะการทำงานปกติตัวรับ Receiver จะสามารถรับสัญญาณแสงจากตัวส่ง Emitter ได้ตลอดเวลา เนื่องจากลำแสงจะสะท้อนกับแผ่นสะท้อน Reflector อยู่ตลอดเวลา จะแสดงค่า เป็น 0 หน้าหลักของเซ็นเซอร์ชนิดนี้ จะคอยตรวจจับวัตถุที่เคลื่อนที่ตัดผ่านหน้าเซ็นเซอร์ เมื่อวัตถุ หรือ ชิ้นงานผ่านเข้ามาที่หน้าเซ็นเซอร์ แล้วจะการขวางลำแสงที่ส่งจากตัวส่ง Emitter ที่ส่งไปยังแผ่นสะท้อน จึงทำให้ตัวรับ Receiver ไม่สามารถรับลำแสงที่จะสะท้อนกลับมาได้ จะแสดงค่า เป็น 1 ซึ่งจะทำให้วงจรภายในรับรู้ได้ว่า มีวัตถุหรือชิ้นงานขวางอยู่ ทำให้สถานะของเอาต์พุตของตัวรับเปลี่ยนแปลงไป โดยเราเรียกลักษณะการทำงานแบบนี้ว่า Dark On หรือ Dark Operate [9]



### Ultrasonic Sensor



ภาพ Ultrasonic Sensor

Ultrasonic sensor คือ เป็นอุปกรณ์ทำงานด้วยคลื่นเสียงความถี่สูง ที่มนุษย์ไม่สามารถได้ยิน มีทิศทางแน่นอนและไม่มีการเลี้ยวเบน โดยอัลตราโซนิก แบ่งเป็นสองส่วน คือ Transmitter เป็นแหล่งให้กำเนิดเสียง Ultrasonic และ Receiver เป็นตัวรับคลื่นเสียงที่สะท้อนกลับมา ซึ่งเราสามารถใช้ sensor ตัวนี้ในการตรวจจับวัตถุและวัดระยะที่ห่างจากวัตถุได้

การวัดระยะทางจากการวัดเวลาที่เสียงใช้ในการเดินทางไปกระทบวัตถุและสะท้อนกลับมา ซึ่งเป็นหลักการเดียวกับสัตว์ตามธรรมชาติ เช่น ค้างคาว ที่แม้ตาจะมองไม่เห็นแต่ค้างคาวใช้การฟังเสียงสะท้อนในการเดินทาง และมนุษย์เองก็ได้ความรู้เหล่านี้มาประยุกต์ใช้ให้เกิดประโยชน์

นำไปใช้ในการหลบหลีกสิ่งกีดขวาง เช่น การให้หุ่นยนต์เดินโดยที่ไม่ชนสิ่งกีดขวาง การนำมาใช้กับรถยนต์เพื่อเป็นตัวกำหนดสัญญาณเตือนว่ามีวัตถุอยู่ในระยะใกล้ (เซนเซอร์ถอยหลังสำหรับรถยนต์)

ข้อดีของอัลตราโซนิก คือ สามารถตรวจจับวัตถุโดยใช้คลื่นเสียงที่มีความถี่สูงซึ่งเป็นคลื่นความถี่ที่มนุษย์ไม่ได้ยิน นอกจากนี้ Ultrasonic Sensor ยังมีระยะการตรวจจับสูงสามารถตรวจจับวัตถุโดยไม่สนใจสีหรือพื้นผิวของวัตถุ ใช้ได้กับทั้งของแข็งและของเหลว ทั้งเปียกและแห้ง สามารถตรวจจับวัตถุที่มีไอฝุ่นได้

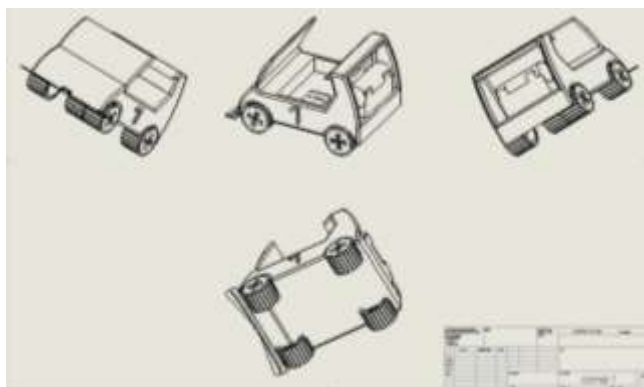
ข้อจำกัดของอัลตราโซนิก คือ อัลตราโซนิกไม่สามารถทำงานในที่แคบ ในสถานที่ที่เป็นสุญญากาศได้ เพราะคลื่นเสียงต้องเดินทางผ่านตัวกลาง ultrasonic ไม่เหมาะกับวัตถุผิวนุ่มและผิวโค้ง นอกจากนี้ควรหลีกเลี่ยงการติดตั้ง ultrasonic ในบริเวณใกล้ๆ กัน [10]

## วิธีการดำเนินงาน

### อุปกรณ์ 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 1 ชิ้น

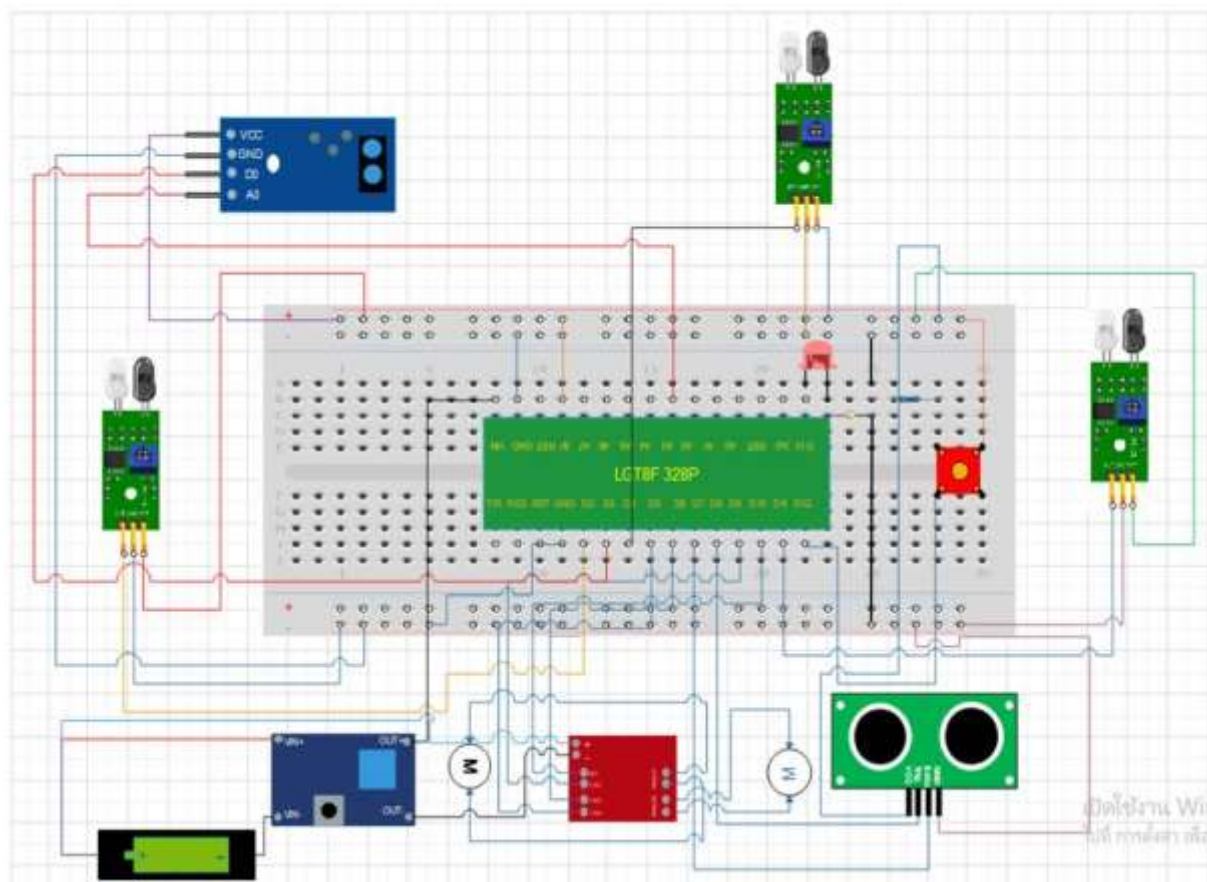
การออกแบบหุ่นยนต์

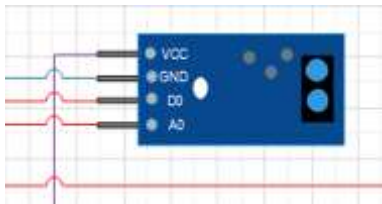
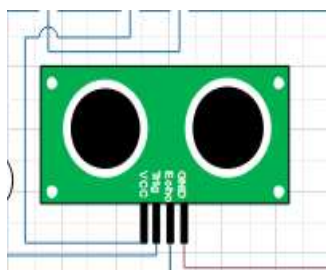
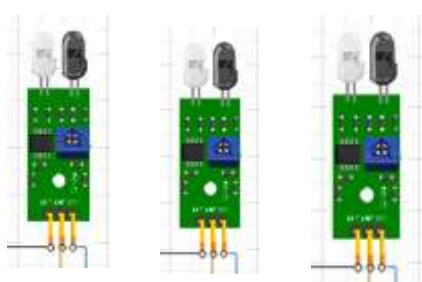

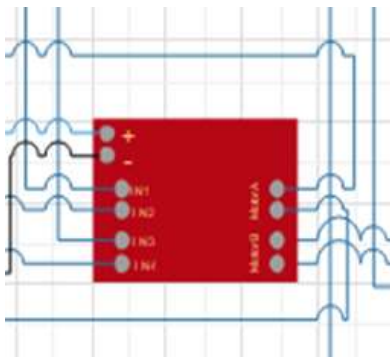


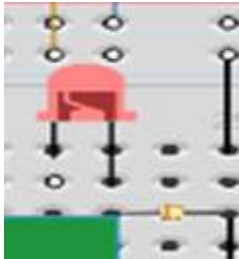
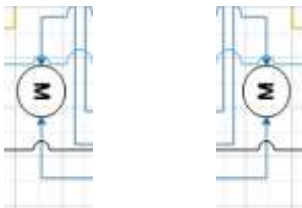
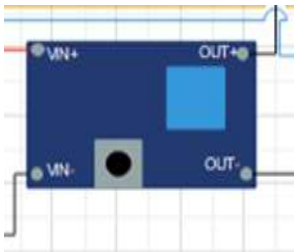
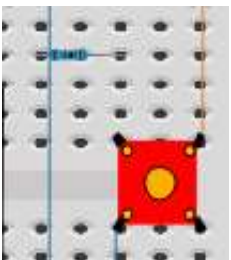
โปรแกรมที่ใช้ออกแบบ : Solid work

ขนาด : 6\*5\*4 เซนติเมตร

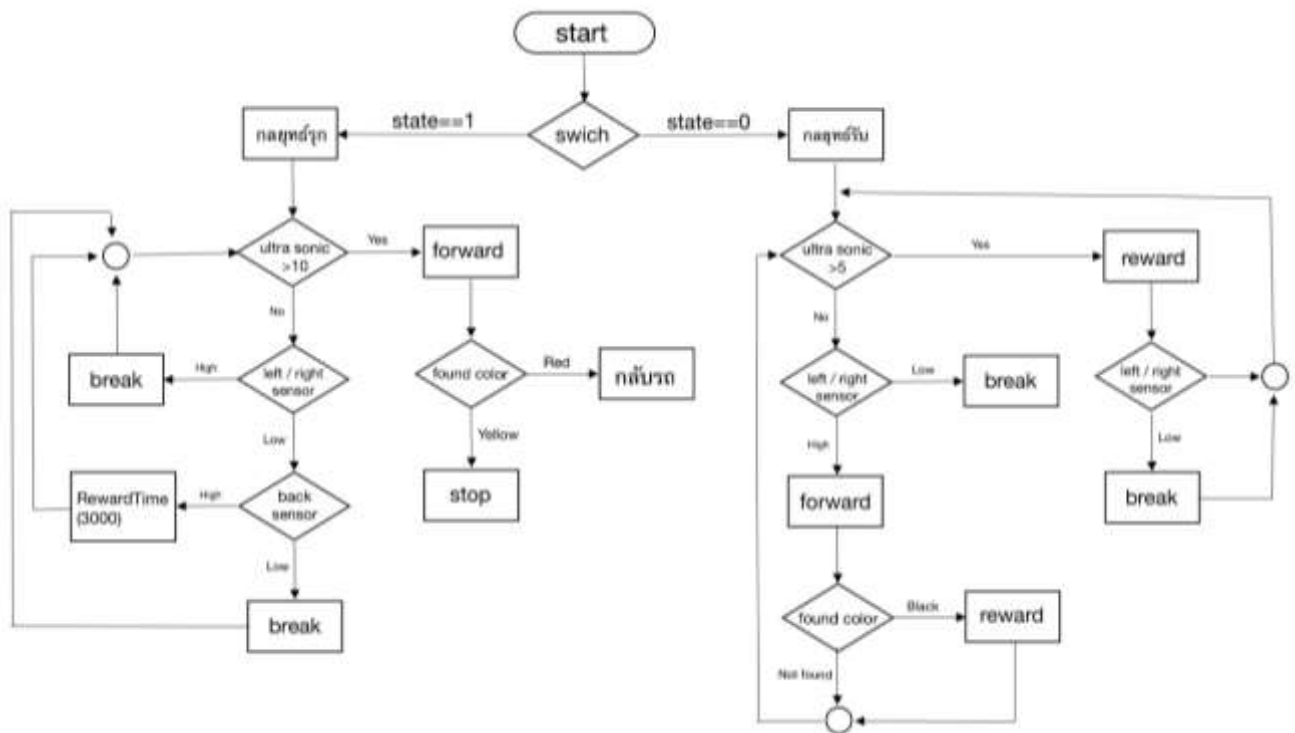
การต่อวงจร



อุปกรณ์	รูปภาพ	การต่อ
TCRT5000 Infrared Reflective sensor		VCC: 5V GND: Ground D0: Digital output (0/1): D3 A0: Analog output: A3
Ultrasonic Sensor		VCC: 5V Trig: Digital output (0/1): D8 Echo: Digital Input (0/1): D7 GND: Ground
IR Infrared Obstacle Avoidance Sensor		OUT: Digital Input (0/1): D2,D4,D11 GND: Ground VCC: 5V
18650 Li-ion battery, Battery case, Li-ion Charging		ขั้วบวก: VIN+ ของ DC/DC Step-up Converter และบวกของ H-bridge Driver ขั้วลบ: VIN- ของ DC/DC Step-up Converter
H-bridge Driver		บวก: ขั้วบวกของ 18650 Li-ion battery ลบ: Ground IN1: D10 IN2: D9 IN3: D6 IN4: D5 4ช่องด้านขวา: Motor 1 and 2

LED และตัว ต้านทาน		ขาบวกของ LED: ขาบวกตัวต้านทาน ขาลบของ LED: D13 ขาบวกตัวต้านทาน: ขาบวกของ LED ขาลบตัวต้านทาน: Ground
DC Motors		ต่อเข้ากับ H-bridge Driver (แผงสีแดง)
DC/DC Step-up Converter		Vin-: V battery (-) Ground Vin+: V battery (+) Out-: H-bridge Driver(-) Out+: VIN (Arduino)
Switchและตัว ต้านทาน	- 	ขาต้านซ้ายของswitch: D12และตัว ต้านทาน ขาต้านขวาของSwitch: 5V ขาบวกของตัวต้านทาน: ขาต้านซ้าย ของswitch ขาลบของตัวต้านทาน: Ground

## Flow Chart



## Programming Codes

```

#define IN1 D10 //motor A
#define IN2 D9 //motor A
#define IN3 D6 //motor B
#define IN4 D5 //motor B

#define left D2 //leftsensor
#define right D11 //rightsensor
#define back D4 //backsensor

#define RED 1
#define YELLOW 2
#define BLACK 3
#define NOCOLOR 0
#define maxSpd 255 // motor max speed

```



```

#include <HCSR04.h>

HCSR04 hc(D8,D7); //initialisation class HCSR04 (trig,echo);

int analogPin = A6; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่6

int led1 = D13;
int buttonPin = D12;
int val=0;
int old_val=0;
int state=0;

int color=0;
int getColor() {
    int NO_color = analogRead(analogPin);    //อ่านค่าสัญญาณ analog ขา6 ที่ต่อกับ
TCRT5000
    if ((NO_color>1900)&&(NO_color<2200))      //สีเหลือง
        return YELLOW;
    else if ((NO_color>2200)&&(NO_color<2500))  //สีแดง
        return RED;
    else if ((NO_color>3600)&&(NO_color<3900))  //สีดำ
        return BLACK;
    else                                     //ไม่พบสี
        return NOCOLOR;
}

void setup() {
    pinMode(left, INPUT);
    pinMode(right, INPUT);
    pinMode(back, INPUT);

```

```

pinMode(buttonPin, INPUT);
pinMode(led1, OUTPUT);
pinMode(IN1, OUTPUT);
pinMode(IN2, OUTPUT);
pinMode(IN3, OUTPUT);
pinMode(IN4, OUTPUT);
Serial.begin(115200);
}

void loop()
{
  color = getColor();
  val = digitalRead(buttonPin);
  if( (val==HIGH) && (old_val==LOW))
  {
    state=!state;
  }
  old_val=val;
  if (state==1) //เมื่อกดสวิตช์ 1 ครั้ง ใช้กลยุทธรีก
  {
    digitalWrite(led1, HIGH);

    if((hc.dist()>10)&&((digitalRead(left)==HIGH)||((digitalRead(right)==HIGH))&&(color ==
NOCOLOR)) // เดินหน้าเมื่อ sensor ด้านหน้า ด้านข้าง และด้านล่างไม่ทำงาน
    {
      MotorAForward(maxSpd);
      MotorBForward(maxSpd);
    }

    if((hc.dist()<10)&&(digitalRead(back)==LOW)&&((digitalRead(left)==HIGH)||((digitalRead(right)==HI

```

GH))&&(color == NOCOLOR)) // break เมื่อ sensor ด้านหน้า ด้านหลังทำงาน แต่ด้านข้างและด้านล่าง  
ไม่ทำงาน

```
{
  MotorABreakTime(1000);
  MotorBBreakTime(1000);
}
```

if((hc.dist()<10)&&(digitalRead(back)==HIGH)&&((digitalRead(left)==LOW)||((digitalRead(right)==LOW))&&(color == NOCOLOR)) // เดินถอยหลัง 3วิ เมื่อ sensor ด้านหน้าและข้างซ้ายหรือขวาทำงาน  
แต่ด้านหลังไม่ทำงาน

```
{
  MotorARewardTime(3000);
  MotorBRewardTime(3000);
}
```

if(color == RED) // เดินกลับรถ เมื่อ sensor ด้านล่างตรวจจับเส้นสีแดงได้

```
{
  MotorAForwardTime(5000);
  MotorBRewardTime(5000);
}
```

if(color == YELLOW) // รถหยุด เมื่อ sensor ด้านล่างตรวจจับเส้นสีเหลืองได้

```
{
  MotorAStop();
  MotorBStop();
}
}
else
{
  digitalWrite (led1,LOW);
```

```

    if((hc.dist()>5)&&((digitalRead(left)==HIGH)||digitalRead(right)==HIGH)&&(color ==
NOCOLOR)) // ถอยหลังเมื่อ sensor ด้านหน้า ด้านข้าง และด้านล่าง ไม่ทำงาน

    {
        MotorAReward(maxSpd);
        MotorBReward(maxSpd);
    }

    if((hc.dist()<5)&&((digitalRead(left)==HIGH)||digitalRead(right)==HIGH)&&(color ==
NOCOLOR)) // เดินหน้าเมื่อ sensor ด้านหน้าทำงาน แต่ด้านข้างและล่างไม่ทำงาน

    {
        MotorAForwardTime(3000);
        MotorBForwardTime(3000);
    }

    if((digitalRead(left)==LOW)||digitalRead(right)==LOW)&&(color == NOCOLOR)) // เบรค
เมื่อ sensor ด้านข้างซ้ายหรือขวาทำงาน แต่ล่างไม่ทำงาน

    {
        MotorABreakTime(1000);
        MotorBBreakTime(1000);
    }

    if(color == BLACK) // กลับรถ เมื่อ sensor ด้านล่างตรวจจับเส้นสีดำได้

    {
        MotorAForwardTime(5000);
        MotorBRewardTime(5000);
    }

    }

    delay(20);

```

```
}
```

```
void MotorAStop()
```

```
{  
    digitalWrite(IN1, LOW); // motor stop  
    digitalWrite(IN2, LOW);  
}
```

```
void MotorABreak()
```

```
{  
    digitalWrite(IN1, HIGH); // motor break  
    digitalWrite(IN2, HIGH);  
}
```

```
void MotorBStop()
```

```
{  
    digitalWrite(IN3, LOW); // motor stop  
    digitalWrite(IN4, LOW);  
}
```

```
void MotorBBreak()
```

```
{  
    digitalWrite(IN3, HIGH); // motor break  
    digitalWrite(IN4, HIGH);  
}
```

```
void MotorAForward(int speed)
```

```
{  
    digitalWrite(IN2, LOW);  
    analogWrite(IN1, speed);  
}
```

```
}
```

```
void MotorBForward(int speed)
```

```
{
```

```
    digitalWrite(IN4, LOW);
```

```
    analogWrite(IN3, speed);
```

```
}
```

```
void MotorAReward(int speed)
```

```
{
```

```
    digitalWrite(IN1, LOW);
```

```
    analogWrite(IN2, speed);
```

```
}
```

```
void MotorBReward(int speed)
```

```
{
```

```
    digitalWrite(IN3, LOW);
```

```
    analogWrite(IN4, speed);
```

```
}
```

```
void MotorARewardTime(int time)
```

```
{
```

```
    digitalWrite(IN1, LOW);
```

```
    analogWrite(IN2, maxSpd);
```

```
    delay (time);
```

```
}
```

```
void MotorBRewardTime(int time)
```

```
{
```

```
    digitalWrite(IN3, LOW);
```

```
    analogWrite(IN4, maxSpd);  
    delay (time);  
}
```

```
void MotorAForwardTime(int time)  
{  
    digitalWrite(IN2, LOW);  
    analogWrite(IN1, maxSpd);  
    delay (time);  
}
```

```
void MotorBForwardTime(int time)  
{  
    digitalWrite(IN4, LOW);  
    analogWrite(IN3, maxSpd);  
    delay (time);  
}
```

```
void MotorABreakTime(int time)  
{  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, HIGH);  
    delay (time);  
}
```

```
void MotorBBreakTime(int time)  
{  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, HIGH);  
    delay (time);  
}
```

## เอกสารอ้างอิง

- [1] <http://krunisit.rwb.ac.th/robot.html>
- [2] <https://blog.ourgreenfish.com>
- [3] <https://www.mcucity.com/product/2618/lgt8f328p-adc-12bit-8ch-dac-8bit-1ch-30-i-o-compatible-arduino-nano-3-0>
- [4] <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>
- [5] <https://www.thaiconverter.com/article/>
- [6] <https://industrypro.co.th/a-c-motor-vs-d-c-motor/>
- [7] <http://naringroup.blogspot.com/2016/03/arduino-pwm.html>
- [8] <https://www.arduitronics.com/product/1295/ir-reflective-obstacle-avoidance-line-tracking-sensor-tcrt5000>
- [9] <https://robotsiam.blogspot.com/2016/10/ir-infrared-obstacle-avoidance-sensor.html>
- [10] <http://pilaiwan15.blogspot.com/2015/09/ultrasonic-sensor.html>