



KMITL
สถาบันเทคโนโลยี
พระจอมเกล้า
เจ้าคุณทหารลาดกระบัง

IBOT

NEWNORMAL CAMP 13

- 22-26 DEC 2020 -

“พบกับจุดเริ่มต้นของการผจญภัย”

ロボ
ット

R
O
B
O
T



IBOT CAMP

แนะนำตัวกันหน่อย

ชื่อ – นามสกุล:

ชื่อเล่น :

เบอร์โทร :

Facebook :

Line ID : Instagram :

จังหวัด :

Download ไฟล์สื่อการสอน.





ประวัติสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ในปี พ.ศ. 2528 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี และสถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ได้รับการจัดตั้งเป็นนิติบุคคลซึ่งมีฐานะเป็นกรมในทบวงมหาวิทยาลัย ปัจจุบันสังกัด สำนักงานคณะกรรมการการอุดมศึกษา กระทรวงศึกษาธิการ โดยมีวัตถุประสงค์เพื่อให้ การศึกษาวิจัย ส่งเสริมและให้บริการวิชาการทางด้านวิทยาศาสตร์เทคโนโลยีและครุศาสตร์ อุตสาหกรรม รวมทั้ง ทำนุบำรุงศิลปะและวัฒนธรรมของชาติ

ชื่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประกอบด้วย พระนาม “พระจอมเกล้า” ซึ่งพระบาทสมเด็จพระเจ้าอยู่หัวภูมิพลอดุลยเดช ได้มี พระบรมราชานุญาตให้ อัญเชิญพระบรมราชลัญจกร “พระมหาพิชัยมงกุฏ” ของ พระบาทสมเด็จพระจอมเกล้าเจ้าอยู่หัว รัชกาลที่ ๔ เป็นตราสัญลักษณ์ประจำสถาบันฯ นับเป็นนามมงคลยิ่ง ส่วนของคำว่า “เจ้าคุณทหาร” นั้น มีไว้เพื่อเป็นอนุสรณ์แด่ ท่านเจ้าพระยาสุรวงษ์ไวยวัฒน์ (วร บุนนาค) ตามที่ ท่านเลี่ยม พรตพิทยพยัต ทายาทของท่าน ได้แจ้งความประสงค์ไว้ในการบริจาคที่ดิน ซึ่งเป็นที่ตั้ง ของสถาบันฯ ในปัจจุบัน...

ภาควิชาโทรคมนาคมและโครงข่าย

พวกเรามักเรียกกันง่ายๆว่า “ภาคโทร” ภาคนี้มีชื่อเสียงมานาน มากๆ และเป็นภาคแรกของลาดกระบังเลยนะ โดยภาคนี้ก็จะเรียนเกี่ยวกับ การสื่อสารทั้งหลาย ไม่ว่าจะเป็นโทรศัพท์ ดาวเทียม ใยแก้วนำแสง และอื่นๆ อีกมากมาย แต่ไม่ใช่ว่าจะเรียนเฉพาะการสื่อสารนะ ยังมีเกี่ยวกับ อิเล็กทรอนิกส์ แล้วก็คอมพิวเตอร์ด้วย งานที่เกี่ยวกับภาควิชานี้ก็จะเป็น งานที่เกี่ยวกับระบบการสื่อสารทั้งหมด ระบบคอมพิวเตอร์ งานสื่อสารใน สายการบิน วิทยุการบิน และบริษัท หรือองค์กรที่เกี่ยวกับการสื่อสารต่างๆ เป็นต้น

ภาควิชาวิศวกรรมไฟฟ้า

คำว่า “Power” เป็นชื่อที่ทุกคนเรียกภาคนี้อย่างติดปาก ซึ่ง น้องๆคงจะสงสัยกันสินะว่าทำไมต้องเรียกว่า “Power???” ที่เป็นแบบนี้ ก็เพราะว่าภาควิชานี้ได้จะศึกษาเกี่ยวกับไฟฟ้ากำลัง หรือไฟฟ้าแรงดันสูง(High-Voltage)นั่นเอง การศึกษาจะมีตั้งแต่การเริ่มต้นผลิตไฟฟ้า,เครื่องจักรกลทางไฟฟ้า, ระบบส่งจ่ายทางไฟฟ้า, อิเล็กทรอนิกส์กำลัง, และเทคโนโลยีพลังงาน รวมไปถึงการออกแบบทางไฟฟ้าซึ่งสายงานของภาคนี้นับได้ว่ากว้างมากเลยทีเดียวถึงกับกล่าวได้ว่า“ที่ใดมีไฟฟ้าที่นั่นมี วิศวกรไฟฟ้า”

ภาควิชาวิศวกรรมพลังงานไฟฟ้า

ภาคนี้นุ่งเน้นการพัฒนาทางพลังงาน โดยเฉพาะพลังงานไฟฟ้า และเน้นการพัฒนาแบบยั่งยืน โดยอาศัยนวัตกรรมไฟฟ้าทางด้าน สมาร์ทกริด และ ไมโครกริด เพื่อรองรับการใช้พลังงานที่ยั่งยืนบนพื้นฐาน ตามทฤษฎีเศรษฐกิจพอเพียง และตอบสนองต่อความต้องการทางสังคม และอุตสาหกรรมให้รองรับเทคโนโลยีสมัยใหม่ และศึกษาการจัดการ บริหารพลังงานให้เกิด

ประสิทธิภาพอย่างสูงสุด, การใช้พลังงานหมุนเวียน และพลังงานทดแทน โดยการพิจารณาทางเลือกของการใช้พลังงาน

ภาควิชาอิเล็กทรอนิกส์

ถ้าพูดถึงอิเล็กทรอนิกส์ น้องๆหลายคน คงคิดถึง TV วิทยุ ของเล่น ต่างๆนานา จริงๆแล้วมันก็ใช่นะแต่ว่าเราทำอะไรได้มากกว่านั้น อยากรู้มั๊ย ละว่าเราเรียนอะไรกันบ้างนะเออ! โดยรวมแล้วนะเราจะเรียนรู้คุณสมบัติของอุปกรณ์ รวมทั้ง การนำเอาอุปกรณ์เหล่านั้น มาใช้งานอย่างเช่น ทรานซิสเตอร์ ไดโอด ตัวเก็บประจุ ตัวเหนี่ยวนำ เป็นต้น สงสัยกันใช่มั๊ยละ ว่าเอาไปทำอะไรได้บ้าง อ่านนี้แล้วจะหายสงสัย ก็เช่น เป็นวงจรขยายสัญญาณ วงจรขับกระแส วงจรรับส่งคลื่นวิทยุ วงจรแปลงสัญญาณและอื่นๆ อีกมากมายที่เราใช้ในชีวิตประจำวันเกือบลืมนะ! เราเรียนเกี่ยวกับ หลักการวัด หลักการทำงานของระบบ Sensor รวมทั้ง วงจรที่ใช้ในงานวัดเพื่อให้เกิด ประสิทธิภาพดีที่สุด เกี่ยวกับ Microprocessor เราก็เรียนด้วยนะจ๊ะ ^_^ อิเล็กทรอนิกส์ทำให้เรานำอุปกรณ์ที่มีมาสร้างเป็นวงจรที่เราต้องการใช้งาน และเลือกใช้ให้เหมาะสมกับงานด้วย

ภาควิชาวิศวกรรมเครื่องกล

เรียนทุกอย่างครบน้องที่เกี่ยวกับ เรื่องร้อนๆเย็นๆ อย่างพวก เทอร์โมไดนามิกส์ การแลกเปลี่ยนความร้อน พวก heater หรือ heat exchanger อะไรพวกนั้นแหละ 555 หรือพวกระบบทำความเย็นต่างๆ ทั้ง ในโรงงานและตามบ้านเรือน เครื่องยนต์ รถยนต์ หรือถ้าจะให้ยากไป อีกชั้น ก็จะเป็นพวกศึกษาของไหลต่างๆ เช่น fluid mechanics หรือพวก ไฮดรอลิก นิวแมติก ต่างๆ รวมไปถึงระบบ ควบคุมในโรงงานต่างๆด้วย ถ้า ยังไม่ทำทนายพอก็จะมีพวก power plant อีกอย่าง อันนี้เกี่ยวกับพวก โรงงานผลิตไฟฟ้าแหละ สนุกมาก แต่น้องก็ต้องมีพื้นฐานไ้อพวก ร้อนๆ เย็นๆนี้มาก่อนแหละ หรือจะไปสายทางด้าน design ก็จะมีพวกออกแบบ เครื่องจักรกล อันนี้สนุกมากเพราะมันกว้าง ...

คิดยังไงก็ได้ สนุก ไม่ค่อย จำกัดความคิดดี แต่ถ้าไม่ชอบก็จะมีประเภท การควบคุมการทำงานของโรงงานให้เราเรียน อันนี้ถ้าเก่งๆ ก็เทพได้เลย... เพราะ ถ้าเรามีความรู้แล้ว เรายังสามารถควบคุมมันได้อย่างมีประสิทธิภาพด้วยเนี่ยได้เปรียบ

ภาควิชาวิศวกรรมขนส่งทางราง

ได้ศึกษาตั้งแต่โครงสร้างพื้นฐาน การขนส่งทางราง องค์ประกอบ และการออกแบบระบบรางรถไฟ การตัดทางรถไฟ การซ่อมบำรุงรถไฟ หน้าที่และการทำงานของระบบต่างๆของรถไฟการวิเคราะห์แรงต้าน และ การใช้พลังงานของรถไฟ การสร้างและซ่อมบำรุงรางรถไฟการควบคุมการจราจรของรถไฟและระบบการส่งสัญญาณ หรือที่เรียกว่า ระบบอาณัติ สัญญาณ (Rail Signaling) การวางแผนงานเกี่ยวกับระบบขนส่งทางราง

ซึ่งใช้องค์ความรู้ด้านการบริหารจัดการและเศรษฐศาสตร์ด้วย การ วิเคราะห์ความปลอดภัยของการเดินรถไฟและรางรถไฟ การคาดคะเนการ ลีกรถของระบบรถไฟและรางรถไฟ เพื่อให้สามารถซ่อมบำรุงระบบได้ ก่อนที่จะเกิดอุบัติเหตุขึ้นการวิเคราะห์ปัญหาทางวิศวกรรมขนส่งทางราง รวมไปถึงระบบรถไฟความเร็วสูงที่กำลังมีการเร่งพัฒนากันอย่างมากด้วย

ภาควิชาวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์หรือที่เรียกกันว่า “ภาคคอม” เป็น ภาควิชาที่เรียนเกี่ยวกับคอมพิวเตอร์และไฟฟ้าบ้างเล็กน้อย ภาคนี้อาจจะเรียน พื้นฐานเกี่ยวกับคอมพิวเตอร์ทั้งหมด เช่น การเขียนโปรแกรม ดิจิตอล เทคโนโลยีการสื่อสาร ก่อนที่จะลงลึกไปตาม Field ต่าง ๆ ทั้ง Database (ฐานข้อมูล), Hardware, Programming (การเขียนโปรแกรม), Image Processing (การประมวลผลภาพ), Network (ระบบเครือข่าย), Security (ระบบรักษาความปลอดภัย), Artificial Intelligent (ปัญญาประดิษฐ์) และ อื่นๆ อีกมากมาย

ภาควิชาวิศวกรรมเกษตรอัจฉริยะ

ถ้าพูดถึงภาคนี้นักหลายคนอาจคิดว่าคงจะเรียนเกี่ยวกับการทำนา ปลูกข้าวปลูกผักปลูกหญ้า กล้วย แต่จริงๆ แล้วมันมีอะไรมากกว่านั้น ก็คือ ภาคนี้นี้จะเรียนคล้ายๆ กับภาคเครื่องกลอะนะ แต่ว่าจะเน้นไปทาง เครื่องจักรกลที่ใช้ในการทำเกษตรกรรม เช่น เครื่องไถ เครื่องหว่าน หรือว่า เครื่องเก็บเกี่ยวผลผลิต แล้วก็ยังมีการออกแบบการใช้งานแล้ว ก็มีการ

ทดสอบเครื่องมือด้วย นอกจากเรียนเครื่องแล้วยังมีเรื่องของจัดการดิน และน้ำ ว่าง่ายๆ ก็คือการวางระบบชลประทานนั่นแหละ แล้วก็เรื่อง พลังงานและสิ่งแวดล้อมว่าเอาง่ายๆ ก็คือการใช้ ประโยชน์จากการใช้ พลังงานทดแทนควบคู่ไปกับการลดผลกระทบต่อสิ่งแวดล้อมจากการใช้ พลังงานนั้นๆ อีกด้วย

ภาควิชาวิศวกรรมโยธา

ภาควิชาวิศวกรรมโยธา เป็นสายงานที่เกี่ยวกับการออกแบบ ควบคุมงานและเป็นที่ปรึกษา การก่อสร้างโดยใช้ความรู้เกี่ยวกับเทคนิค ก่อสร้าง การวางแผนก่อสร้าง การประมาณราคา การวิเคราะห์โครงสร้าง จะทำงานร่วมกับสถาปนิกเพื่อให้การก่อสร้างเป็นไปอย่างสะดวก (ดี เร็ว ประหยัด และไม่โกงกิน)

ภาควิชาวิศวกรรมเคมี

โดยภาพรวมแล้วจะเรียนคล้ายภาคเครื่องกลแต่จะมีปฏิกิริยาเคมี เข้ามาเกี่ยวข้อง และจะ เน้นไปทางอุปกรณ์ที่ใช้ในทางอุตสาหกรรม วิชาที่ เรียนส่วนใหญ่จะเกี่ยวกับสมดุลในด้านต่างๆ (มวล, พลังงาน, โมเมนตัม) ซึ่งจะไม่เหมือนกับตอนที่น้องเรียนในชั้นมัธยม แต่ถ้าน้องมีพื้นฐานทาง คณิตศาสตร์, ฟิสิกส์, ปริมาณสารสัมพันธ์ และ Rate laws ของเคมี จะช่วยน้องได้มากทีเดียว งาน

สำหรับสาขานี้ก็มีหลากหลาย ประมาณว่า เกือบทุกอย่างในอุตสาหกรรมเราทำได้หมด เช่น น้ำมัน, พลาสติก, กระดาษ, เครื่องอุปโภค-บริโภค...

ภาควิชาวิศวกรรมปิโตรเคมี

อุตสาหกรรมที่นำเอาสิ่งที่ได้จากการกลั่นแยกสาร มาทำปฏิกิริยา แล้วขึ้น รูปเป็นพอลิเมอร์ หรือเม็ดพลาสติก แล้วจึงส่งขายเป็นเม็ด เพื่อให้ บริษัทอื่นฉีดขึ้นรูปต่อไป บริษัททางด้านปิโตรเคมีเช่น Aromatics , SCG , ไทยพลาสติก , ปตท. กรู๊ป

ภาควิชาวิศวกรรมอาหาร

สำหรับภาควิชาวิศวกรรมอาหาร หรือที่เรียกกันว่า “ภาค Food” หลายคนคงคิดว่าเรียนภาคนี้ไป เรียนไปทำอาหารใช้หรือเปล่า แท้จริงแล้วภาคนี้เรียนเกี่ยวกับอุตสาหกรรมการผลิตอาหาร ว่าอาหารที่อร่อยๆที่เรากิน เราสามารถผลิตเป็นจำนวนมากๆ โดยคงความอร่อย ความสดได้หรือเปล่า โดยทั่วไปเนื้อหาวิชาที่เราเรียนกันจะคล้ายๆ กับวิศวกรรมเครื่องกล แต่จะมีวิทยาศาสตร์อาหารเพิ่มเข้ามา โดยเราจะคำนึงถึงปัจจัยที่มีผลต่อการผลิตอาหารอีกด้วย เพราะฉะนั้น เมื่อจบมาสายงานก็จะเป็น วิศวกรประจำโรงงานอุตสาหกรรม แปรรูปอาหาร ควบคุมการผลิต นักวิจัยออกแบบเครื่องจักรที่เกี่ยวกับอาหารก็ได้

ภาควิชาวิศวกรรมอุตสาหการ

วิศวกรรมอุตสาหการ ภาคเราเรียกกันสั้นๆ ว่าภาคอุตฯ หรือ IE ก่อนอื่นก็ต้องขอบคุณพวกพี่ๆ ที่ได้สะสมชื่อเสียงอันมีค่าไว้ให้กับน้องรุ่นหลัง ทำให้น้องๆ สามารถเข้ามาทำงานในบริษัทชั้นนำได้โดยง่ายและทำให้ ภาคอุตฯของเราเป็นภาควิชาชั้นนำของวิศวะในประเทศไทย โดยภาคนี้จะ

เรียนเกี่ยวกับการวางแผน ออกแบบ พัฒนา ปรับปรุงระบบและการบริหาร คนกับวัสดุเครื่องมือให้ทำงานประสานกันได้อย่างมีประสิทธิภาพ

ภาควิชาวิศวกรรมแมคคาทรอนิกส์และอัตโนมัติ

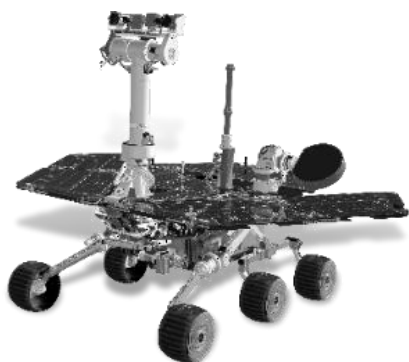
สาขาวิชานี้ประกอบด้วยสามสาขาให้เลือกเรียนด้วยกันได้แก่ วิศวกรรมแมคคาทรอนิกส์ วิศวกรรมระบบอัตโนมัติ และวิศวกรรมระบบควบคุม ซึ่งการศึกษาในปีแรกๆนั้นจะเป็นการเรียนรวมกันก่อน แล้วจะมีการแยกไปตามสาขาที่ถนัดในปีการศึกษาที่สูงขึ้น ส่วนรายละเอียดแต่ละสาขานั้นสามารถสอบถามพี่ๆได้เลย

ภาควิชาวิศวกรรมหุ่นยนต์และปัญญาประดิษฐ์

เป็นสาขาของวิศวกรรมที่ออกแบบมาเพื่อตอบโจทย์วิวัฒนาการยุคดิจิทัล โดยเฉพาะในอนาคตอันใกล้ที่หลายๆอย่างจะถูกแทนที่ด้วยหุ่นยนต์และปัญญาประดิษฐ์ หลักสูตรนี้จึงเป็นหลักสูตรที่สร้างวิศวกรเพื่อพัฒนาซอฟต์แวร์สำหรับ AI และ Robot โดยเฉพาะ

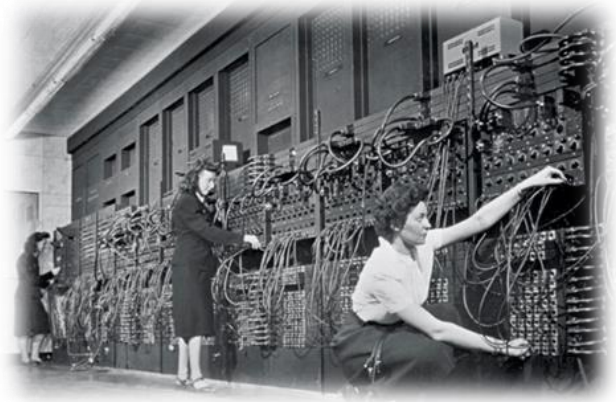
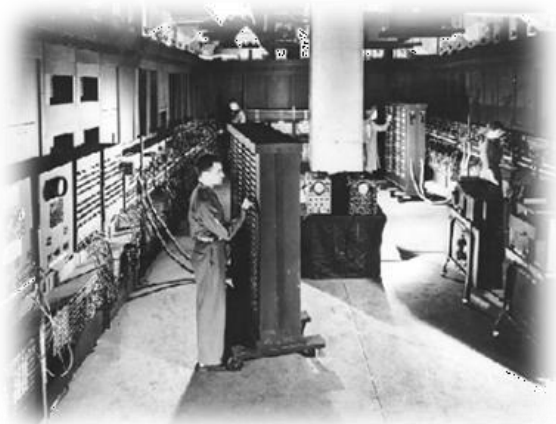
หุ่นยนต์คืออะไร?

หุ่นยนต์ (robot) คือเครื่องจักรกลชนิดหนึ่ง หุ่นยนต์ในแต่ละประเภท จะมีหน้าที่การทำงานในด้านต่างๆ ตามการควบคุมโดยตรงของมนุษย์ ซึ่งสามารถควบคุมได้ทั้งทางอ้อมและอัตโนมัติ โดยหุ่นยนต์นั้นถูกสร้าง เพื่อให้ทำงานที่ยากลำบาก เช่น งานสำรวจบริเวณแคบๆ, งานสำรวจอวกาศและด้านอุตสาหกรรมเป็นต้น เทคโนโลยีของหุ่นยนต์ เจริญก้าวหน้าอย่างรวดเร็ว ทำให้มีบทบาทกับชีวิตของมนุษย์มากขึ้น เช่น หุ่นยนต์ที่ใช้งานทางการแพทย์ งานสำรวจงานในอวกาศหรือแม้แต่หุ่นยนต์ที่ถูกสร้างขึ้นเพื่อเป็นเครื่องเล่นของมนุษย์และหุ่นยนต์มีลักษณะคล้ายมนุษย์เพื่อให้อยู่ร่วมกันกับมนุษย์ในชีวิตประจำวันได้



คอมพิวเตอร์เครื่องแรกของโลก!!!

คอมพิวเตอร์เครื่องแรกของโลกมีชื่อว่าเครื่อง ENIAC โดยย่อมาจาก Electronics Numerical Integrator and Computer โดยถูกพัฒนาขึ้นที่ ประเทศสหรัฐอเมริกา เมื่อปี 1946 โดย ดร.จอห์น ดับเบิลยู. มอชลีย์ (John W. Mauchly) และ เจ เพรสเพอร์ เอกเคิร์ต (J. Presper Eckert) ภายใต้เงินทุนจากกองทัพสหรัฐอเมริกา ซึ่งต้องการสร้างระบบคำนวณวิถี ของกระสุนปืนใหญ่ เครื่อง ENIAC เป็นเครื่องคำนวณทางอิเล็กทรอนิกส์ เครื่องแรกที่ได้รับการยอมรับว่ามันคือเครื่องคอมพิวเตอร์เครื่องแรกของ โลก ซึ่งสร้างจากหลอดสุญญากาศจำนวน 18,000 หลอด มีขนาดใหญ่มาก ต้องใช้พื้นที่วางระบบถึง 15,000 ตารางฟุต หนักรวม 30 ตัน ใช้พลังงาน มากถึง 140 กิโลวัตต์ (ในขณะที่คอมพิวเตอร์ตั้งโต๊ะในปัจจุบันกินไฟเพียง แค่ราวๆ 500 วัตต์) โดยสร้างขึ้นมาบนพื้นฐานของเครื่อง ABC (Atanasoff- Berry Computer) ซึ่งถูกคิดและผลิตโดย Dr. John Vincent Atanasoff และ Clifford Berry

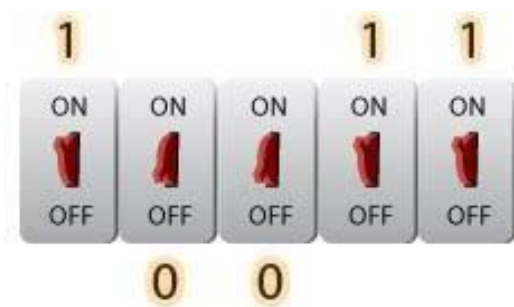


คอมพิวเตอร์กับเลขฐานสอง

ตัวเลขที่คนเราใช้ในชีวิตประจำวันคือเลขฐาน 10 ซึ่งประกอบด้วย

ตัวเลขจำนวน 10 ตัว คือ เลข 0 ถึงเลข 9 เหตุผลที่คนเราใช้เลขฐาน 10 อาจเป็นเพราะว่ามนุษย์เรามีนิ้วมืออยู่ 10 นิ้ว จึงนำมาใช้เป็นเครื่องมือช่วย ในนับเลขหรือการคำนวณ แต่สำหรับการประมวลผลในคอมพิวเตอร์จะใช้ ระบบเลขฐานสอง ที่ประกอบด้วยตัวเลข 2 ตัว คือ เลข 0 และเลข 1 เพราะว่าภายในเครื่องคอมพิวเตอร์ประกอบด้วยวงจรรีเลย์ทรานซิสต์ ซึ่งมี หลักการทำงานเป็นแบบดิจิทัลที่มีใช้ระดับแรงดันไฟฟ้า 2 ระดับ คือ สวิตช์เปิด (on) กับสวิตช์ปิด (off) โดยกำหนดให้สถานะของการ "เปิด" แทนด้วยเลข "0" และ "ปิด" แทนด้วยเลข "1" ซึ่งเลขฐานสองจำนวนหนึ่ง หลัก เราเรียกว่า "บิต" (Bit – Binary Digit)

นอกจากนี้คอมพิวเตอร์ยังมีการใช้งานตัวเลขฐานอื่น ๆ อีก คือ เลขฐานแปด ที่ประกอบด้วยตัวเลข 8 ตัว คือ 0 ถึง 7 และเลขฐานสิบหก ที่ประกอบด้วยตัวเลข 0 ถึง 9 และตัวอักษรอีก 6 ตัวคือ A, B, C, D, E และ F ซึ่งมีค่าเท่ากับเลข 10 ถึง 15



$$5 = \begin{array}{ccccccc} \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bigcirc & \bullet & \bigcirc & \bullet \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$88 = \begin{array}{ccccccc} \bigcirc & \bullet & \bigcirc & \bullet & \bullet & \bigcirc & \bigcirc & \bigcirc \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}$$

การแปลงเลขฐานสองกับฐานสิบ

การแปลงเลขฐานสิบเป็น เลขฐานสอง โดยปกติจะใช้วิธีการ หารสั้น ซึ่งผลลัพธ์จากการแปลงจะได้จากการมองเศษที่ได้จากการ หารด้วย 2 ในแต่ละขั้นจากด้านล่าง ด้านขึ้นบน เช่น ตัวอย่างการแปลง เลข 13 ฐาน 10 เป็นฐาน 2 ดังนี้

13				นำ 13 ตั้ง
2)13			
2)6	เศษ	1	
2)3	เศษ	0	
2)1	เศษ	1	
	0	เศษ	1	
				คำตอบ 01101

สำหรับการแปลงเลขฐานสองเป็นเลขฐานสิบนั้น ทำได้โดยการนำ เลขโดดแต่ละตัวในเลขฐานสอง คูณด้วยค่าประจำหลักของเลขโดดแต่ละหลัก แล้วนำมาบวกรวมกัน ซึ่งค่าประจำหลักของเลขฐานสองเริ่มจาก หลักทางขวาไปซ้ายจะเท่ากับ 20, 21, 22, 23, 24, ... หรือ 1, 2, 4, 8, 16, ...

ตัวอย่างเช่นการแปลงเลขฐานสอง 10011011 เป็นเลขฐาน 10 ดังนี้

128	64	32	16	8	4	2	1
1	0	0	1	1	0	1	1
<hr/>							
128 + 0 + 0 + 16 + 8 + 0 + 2 + 1							
= 155							

การเขียนโปรแกรมคืออะไร

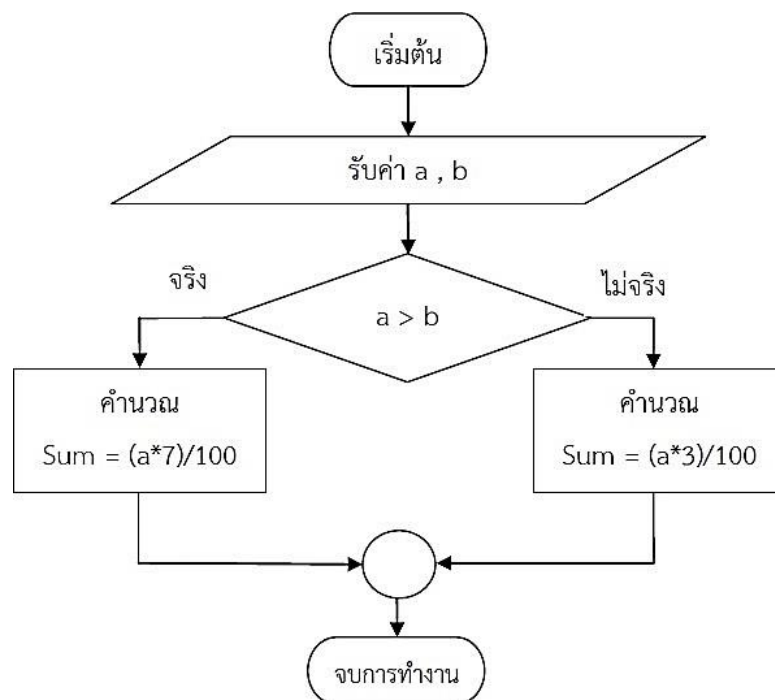
การเขียนโปรแกรมบนคอมพิวเตอร์ เป็นเหมือนการเขียนและ รวบรวมคำสั่งเล็กๆ ในรูปแบบของ “ภาษาคอมพิวเตอร์” จนกลายเป็น ชุดคำสั่งใหญ่ๆ ที่สามารถสั่งให้ คอมพิวเตอร์นั้นทำงานอย่างใดอย่างหนึ่ง ตามที่เราต้องการได้ เพราะคอมพิวเตอร์นั้นไม่สามารถเข้าใจภาษาของมนุษย์ได้โดยตรง ก็เลยต้องมี “ภาษาคอมพิวเตอร์” มาใช้เป็นสื่อกลาง ระหว่างมนุษย์กับคอมพิวเตอร์ ซึ่งก็มักจะอยู่ในรูปแบบของคำสั่งต่าง ๆ ที่มีลักษณะตามข้อกำหนดต่าง ๆ ของภาษาคอมพิวเตอร์นั้น ๆ เช่น ภาษา C, Java, Python

การเขียนโปรแกรมเพื่อควบคุมหุ่นยนต์












ภายในหุ่นยนต์ทั่ว ๆ ไปนั้นจะมีสิ่งหนึ่งที่เปรียบเหมือนเป็นสมอง คอยทำหน้าที่ประมวลผล และออกคำสั่งเพื่อควบคุมการทำงานของ อุปกรณ์ต่างๆ ของตัวหุ่นยนต์ โดยสิ่งนั้นก็คือ “ไมโครคอนโทรลเลอร์” ซึ่งเป็นอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก และมีการทำงานในลักษณะ เดียวกันกับคอมพิวเตอร์ ดังนั้นเราจึงสามารถเขียนโปรแกรมลงบน ไมโครคอนโทรลเลอร์เพื่อกำหนดการทำงานของหุ่นยนต์ได้ โดยส่วนใหญ่ ในปัจจุบันจะ ใช้ภาษา C และ Assembly ในการเขียนคำสั่งควบคุม

การเขียนโฟลว์ชาร์ต

ในการเขียนโปรแกรมที่ดีนั้น มักจะเริ่มต้นด้วยการออกแบบโปรแกรม ว่าการทำงานของโปรแกรมตั้งแต่เริ่มต้นไปจนถึงสิ้นสุดควรจะเป็นอย่างไร วิธีการสำคัญอย่างหนึ่งในการออกแบบโปรแกรมคือการเขียนโฟลว์ชาร์ต (Flow chart) ที่แสดงให้เห็นถึงการทำงานของโปรแกรมในขั้นตอนต่าง ๆ อย่างเป็นกระบวนการ และเมื่อเราออกแบบโปรแกรมด้วยโฟลว์ชาร์ตให้ สามารถตอบโจทย์หรือแก้ปัญหาตามที่เราต้องการได้แล้ว เราก็จะเริ่มต้น การเขียนโปรแกรมด้วยภาษาคอมพิวเตอร์และคำสั่งต่าง ๆ จนกลายเป็นโปรแกรมที่สามารถทำงานได้โดยสมบูรณ์ ตัวอย่างการเขียน เช่น



สัญลักษณ์ต่าง ๆ ที่ใช้ในการเขียนโฟลว์ชาร์ต

ภาพสัญลักษณ์	ความหมาย
 Start/End Symbol	เริ่มต้น/สิ้นสุด, การเริ่มต้นหรือการลงท้าย
 Connection Symbol	จุดเชื่อมต่อ ในหน้าเดียวกัน
 Connection Symbol	จุดเชื่อมต่อคนละหน้า
 Monitor	จอภาพแสดงผล
 Processing	การประมวลผลทั่วไป ยกเว้นการอ่านข้อมูลและ การแสดงผลลัพธ์
 Input/Output Data	รับหรือแสดงข้อมูล โดยไม่ระบุชนิดอุปกรณ์
 Decision Symbol	การตัดสินใจ การเปรียบเทียบ (จะมีทิศทางออก 2 ทิศทาง คือกรณีที่มีผลตรวจสอบเงื่อนไขเป็นเท็จและเป็นจริง)
 Manual input	การรับข้อมูล เข้าทางแป้นพิมพ์
 Document Output	เอกสารแสดงผล, การแสดงผลทางเครื่องพิมพ์
 Preparation	ใช้กำหนดค่าต่างๆล่วงหน้า ซึ่งเป็นการทำงาน ภายในช่วงหนึ่งถึงซ้ำๆกัน
 Flow line	เส้นแสดงลำดับกิจกรรม

การเขียนโปรแกรมด้วยภาษา C

- ตัวแปรและการดำเนินการ

ตัวแปรคือสิ่งที่เรากำหนดขึ้นและตั้งชื่อให้เพื่อนำมาเก็บข้อมูล ซึ่งตัวแปร ในภาษาซีนั้นจริงๆแล้วจะเก็บเพียงแค่เลข 1 กับ 0 เท่านั้น เนื่องจากคอมพิวเตอร์เก็บ ข้อมูลเพียงแค่ 0 กับ 1 เรียกว่า bit แต่สามารถเก็บหลายๆ bit ได้ เช่น เลข 8 bit จะเก็บข้อมูลได้ $2^8 = 256$ ตัว คือ ตั้งแต่ 000000002 ไล่ไปถึง 111111112 ซึ่ง สามารถนำไปประยุกต์เก็บเป็นเลขจำนวนลบ (1's หรือ 2's complement) หรือ เก็บอักขระ (ASCII Code) ได้ ซึ่งตัวแปรในภาษาซีนั้นมีหลายชนิดแต่ชนิดก็สามารถเก็บข้อมูลได้แตกต่างกัน

ชนิดของตัวแปร	ขนาด (bits)	ขอบเขต	ข้อมูลที่เก็บ
char	8	-128 ถึง 127	ข้อมูลชนิดอักขระ ใช้เนื้อที่ 1 byte
unsigned char	8	0 ถึง 255	ข้อมูลชนิดอักขระ ไม่คิดเครื่องหมาย
int	16/32	-32768 ถึง 32767 -2^{31} ถึง $2^{31} - 1$	ข้อมูลชนิดจำนวนเต็ม ใช้เนื้อที่ 2 byte
unsigned int	16/32	0 ถึง 65,535 / 0 ถึง 2^{32}	ข้อมูลชนิดจำนวนเต็ม ไม่คิดเครื่องหมาย
short	8	-128 ถึง 127	ข้อมูลชนิดจำนวนเต็ม แบบสั้น ใช้เนื้อที่ 1 byte

unsigned short	8	0 ถึง 255	ข้อมูลชนิดจำนวนเต็ม แบบสั้น ไม่คิดเครื่องหมาย
long	32	-2,147,483,648 ถึง 2,147,483,648	ข้อมูลชนิดจำนวนเต็ม แบบ ยาว ใช้เนื้อที่ 4 byte
unsigned long	32	0 ถึง 4,294,967,296	ข้อมูลชนิดจำนวนเต็ม แบบยาว ไม่คิด เครื่องหมาย
float	32	3.4×10^{-38} ถึง 3.4×10^{38}	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 4 byte
double	64	3.4×10^{-308} ถึง 3.4×10^{308}	ข้อมูลชนิดเลขทศนิยม ใช้เนื้อที่ 8 byte

*** 8 Bit = 1 byte

*** ขนาดของตัวแปร int จะขึ้นกับ Compiler และระบบปฏิบัติการ

*** unsigned คือตัวแปรแบบไม่คิดเครื่องหมาย (บวก/ติดลบ)

รูปแบบการประกาศตัวแปร

ตัวแปรที่นิยมใช้ในค่านี้อจะเป็นชนิด `int` ที่เรียกว่า “ตัวแปรจำนวนเต็ม” โดยสามารถประกาศตัวแปรได้ทั้งแบบระบุค่าเริ่มต้น และไม่ระบุค่าเริ่มต้น

การประกาศตัวแปรแบบไม่ระบุค่าเริ่มต้น

ชนิดตัวแปร ชื่อตัวแปร;

โดยสามารถกำหนดค่าในภายหลังได้โดยใช้คำสั่งรูปแบบ

ชนิดตัวแปร ชื่อตัวแปร;

และการประกาศแบบระบุค่าเริ่มต้นเลยตั้งแต่แรก ทำได้โดยใช้คำสั่ง

ชนิดตัวแปร ชื่อตัวแปร = ค่าเริ่มต้น;

ตัวอย่างการประกาศตัวแปร

```
int x;           // เราจะได้ตัวแปรจำนวนเต็ม ซึ่งมีชื่อว่า x
int y = 1;       // เราจะได้ตัวแปรจำนวนเต็ม ซึ่งมีชื่อว่า y
                  // และทำการกำหนดค่าให้เท่ากับ 1 ทันที

void main()
{
    x = 3;        // เป็นการกำหนดค่าให้กับตัวแปร x
}                // ที่มีการประกาศไว้แล้ว ให้มีค่าเท่ากับ 3
```

ตัวดำเนินการ

คือการกระทำต่างๆทั้ง การกำหนดค่า , การกระทำทางคณิตศาสตร์ การเปรียบเทียบ, การกระทำทางตรรกศาสตร์ และการกระทำในระดับบิต

- ตัวดำเนินการทางคณิตศาสตร์(มักจะใช้ค่านวนค่าต่างๆในโปรแกรม)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
+	นำค่าทางซ้ายบวกด้วยค่าทางขวา	$A + B$
-	นำค่าทางซ้ายลบด้วยค่าทางขวา	$A - B$
*	นำค่าทางซ้ายคูณด้วยค่าทางขวา	$A * B$
/	นำค่าทางซ้ายหารด้วยค่าทางขวา	A / B
%	หาเศษจากการหารค่าทางซ้ายกับค่าทางขวา	$A \% B$

- ตัวดำเนินการเปรียบเทียบ (มักจะใช้สำหรับการตรวจสอบเงื่อนไข)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
==	เป็นจริงเมื่อค่าทางซ้ายเท่ากับค่าทางขวา	$A == B$
!=	เป็นจริงเมื่อค่าทางซ้ายไม่เท่ากับค่าทางขวา	$A != B$
>	เป็นจริงเมื่อค่าซ้ายมากกว่าค่าขวา	$A > B$
>=	เป็นจริงเมื่อค่าทางซ้ายมากกว่าหรือเท่ากับค่าทางขวา	$A >= B$
<	เป็นจริงเมื่อค่าซ้ายน้อยกว่าค่าขวา	$A < B$
<=	เป็นจริงเมื่อค่าซ้ายน้อยกว่าหรือเท่ากับค่าขวา	$A <= B$

ตัวดำเนินการทางตรรกศาสตร์(มักใช้ในการตรวจสอบหลายเงื่อนไขพร้อมกัน)

เครื่องหมาย	ความหมาย	ตัวอย่าง	คำอธิบาย
&&	And	A && B	เป็นจริงเมื่อ A และ B เป็นจริง
	Or	A B	เป็นเท็จเมื่อ A หรือ B เป็นเท็จ
!	Not	!A	เป็นจริงเมื่อ A เป็นเท็จ

ตัวดำเนินการทางยูนารี (เป็นตัวดำเนินการอย่างย่อ)

ตัวดำเนินการ	ความหมาย	ตัวอย่าง	การทำงาน
++	เพิ่มค่าของตัวแปรขึ้นหนึ่งหน่วย	A = B++	A = B B = B + 1
		A = ++B	B = B + 1 A = B
--	ลดค่าของตัวแปรลงหนึ่งหน่วย	A = B--	A = B B = B - 1
		A = --B	B = B - 1 A = B
+=	เพิ่มค่าของตัวแปรจากค่าเดิมตามที่ ต้องการ	A += B	A = A + B
-=	ลดค่าของตัวแปรจากค่าเดิมตามที่ ต้องการ	A -= B	A = A - B

การเขียนโปรแกรมแบบกำหนดเงื่อนไข

การกำหนดเงื่อนไข คือการกำหนดว่า หากเหตุการณ์ขณะนั้นเป็น แบบที่หนึ่งก็ให้ทำตามคำสั่งของเงื่อนไขที่หนึ่ง และหากเหตุการณ์ขณะนั้น เป็นแบบที่สองก็ให้ไปทำตามคำสั่งของเงื่อนไขที่สอง ยกตัวอย่างเช่น มีตัวแปรเก็บตัวเลขที่มีค่าต่างกัน เราอยากทราบว่าตัวแปรตัวใดมีค่า มากกว่ากัน เราก็ใช้การเขียนโปรแกรมแบบกำหนดเงื่อนไขนี้เข้ามาช่วยได้

- คำสั่ง if

เป็นคำสั่งที่ใช้ตัดสินใจว่าจะทำชุดคำสั่งที่อยู่ภายใต้คำสั่ง if หรือไม่ ถ้านิพจน์ (เงื่อนไข) ที่อยู่หลังคำสั่ง if เป็นจริงก็จะทำงานตามคำสั่งที่อยู่ ภายใต้ชุดคำสั่ง if แต่ถ้านิพจน์นั้นเป็นเท็จ ก็จะข้ามชุดคำสั่งได้ if ไปเลย

- คำสั่ง else if

คำสั่ง else if เป็นเงื่อนไขที่เรากำหนดเพิ่มเติมจากเงื่อนไขใน if แรก ซึ่งโปรแกรมจะตรวจสอบเงื่อนไขใน else if แต่ละตัวเป็นลำดับ หากนิพจน์ ที่อยู่หลัง else if นั้นๆเป็นจริง ก็จะทำงานคำสั่งภายใต้ else if ชุดนั้น ชุดเดียวแล้วข้ามเงื่อนไขอื่นที่เหลือทั้งหมดทันที

- คำสั่ง else

หากเงื่อนไขหลัง if หรือ else if ทั้งหมดนั้นเป็นเท็จ โปรแกรมจะข้ามมา ทำชุดคำสั่งที่อยู่ ใน else ทันที

<pre> if (เงื่อนไข) { คำสั่ง... ; } else if (เงื่อนไข) { คำสั่ง... ; } else { คำสั่ง... ; } </pre>	<pre> int x = 10, y = 11, z = 12; void main() { if (x > y && x > z) { printf("Max = x"); } else if (y > x && y > z) { printf("Max = y"); } else { printf("Max = z"); } } </pre>
--	--

การเขียนโปรแกรมแบบวนซ้ำ

คือการทำงานเดิมซ้ำๆ ไปเรื่อยๆ จนกว่าภายใต้เงื่อนไขที่กำหนดไว้

- คำสั่ง for

```

for (สถานะเริ่มต้น ; เงื่อนไข ; การกระทำเมื่อจบหนึ่งรอบ)
{
    คำสั่ง... ;
}

```

รูปแบบการเขียนลูป for

คำสั่ง for เป็นคำสั่งควบคุมการทำซ้ำที่มีการตรวจสอบเงื่อนไขก่อน การทำงานทุกครั้ง การทำงานคือตอนแรกจะเริ่มกำหนดค่าตัวแปรก่อนแล้วมาตรวจสอบเงื่อนไข หลังจากนั้นจึง มาทำคำสั่งภายในลูป และในตอนจบของ ทุกๆ ลูป จะทำการเปลี่ยนแปลงค่าก่อน แล้วจึงไปเช็คเงื่อนไข แล้ววนซ้ำๆ แบบนี้ ไปจนกว่าเงื่อนไขจะเป็นเท็จ

```

void main()
{
    for (int c = 0; c < 10; c++)
    {
        printf("A");
    }
}

ผลลัพธ์
>>> AAAAAAAAAA

```


- คำสั่ง while

เป็นคำสั่งที่ตรวจสอบเงื่อนไขก่อนจึงจะเข้าไปทำงานภายในลูปทุกครั้ง และ จะหยุดทำซ้ำเมื่อเงื่อนไขนั้นเป็นเท็จ

<pre>while (เงื่อนไข) { คำสั่ง... ; }</pre>	<pre>int c; void main() { c = 0; while (c < 10) { printf("B"); c++; } }</pre>
---	---

รูปแบบและตัวอย่างการเขียนลูป while

- คำสั่ง do...while

เป็นคำสั่งที่ทำงานเหมือนกับ while แต่เพียงแค่ว่าจะทำงานในลูปไปก่อน หนึ่งรอบ แล้วจึงไปเช็คเงื่อนไข และถ้าเงื่อนไขเป็นจริงจะทำต่อจนเป็นเท็จ

<pre>do { คำสั่ง... ; }while (เงื่อนไข) ;</pre>	<pre>int c; void main() { c = 0; do { printf("C"); c++; } while (c < 10); }</pre>
---	---

รูปแบบและตัวอย่างการเขียนลูป do while

การสร้างฟังก์ชันย่อย

บางครั้งในการเขียนโปรแกรม ก็มักจะมีชุดคำสั่งที่ถูกเรียกใช้งาน บ่อยๆ หลายๆ ครั้งๆ และเป็นคำสั่งที่เหมือนกันทุกประการ เราสามารถ นำชุดคำสั่งนั้นไปสร้างเป็น ฟังก์ชันย่อย ได้ดังนี้

```
int x = 1;
int y = 7;

void sayHello()
{
    printf("Hello ");
    printf("World!");
    x++;
    y++;
}

void main()
{
    sayHello();
}
```

นำคำสั่ง รวมกันเป็นชุด
และประกาศเป็นฟังก์ชันไว้ใช้งาน

เรียกใช้งานฟังก์ชันที่ประกาศไว้

โดยฟังก์ชันที่เราสร้างขึ้นมานี้ในตอนนี้จะยังไม่มีค่ารับค่าและคืนค่า ใดๆ การประกาศฟังก์ชันในรูปแบบนี้จะทำให้เราสามารถใช้คำสั่งชุดเดิม หลายๆ ครั้งได้สะดวกยิ่งขึ้น เหมือนกับการรวมคำสั่งๆ หลายๆ คำสั่งให้ กลายเป็นคำสั่งชุดเดียว ดังเช่นตัวอย่าง ได้รวมคำสั่งแสดงผลและเพิ่มค่าตัวแปรไว้ด้วยกัน เวลาจะใช้งานก็จะเรียกโดยใช้คำสั่ง sayHello(); ได้ทันที

การประกาศฟังก์ชันรูปทั่วไป

ในรูปแบบทั่วไปแล้วนั้น ฟังก์ชันสามารถรับค่าเข้ามา และทำการ คำนวณค่าที่ได้จากการประมวลผลของฟังก์ชันได้ โดยการประกาศฟังก์ชันแบบ ทั่วไปนั้นจะมีรูปแบบเป็น

```
ชนิดค่า ชื่อฟังก์ชัน (ชนิดค่า พารามิเตอร์ที่1,ชนิดค่า พารามิเตอร์ที่2,...)
{
    คำสั่งต่าง ๆ...;
    return ค่าที่ต้องการจะคืนกลับ;
}
```

ตัวอย่างเช่น ฟังก์ชันสำหรับบวกเลข จะมีการประกาศและเรียกใช้งานดังนี้

```
int ans;

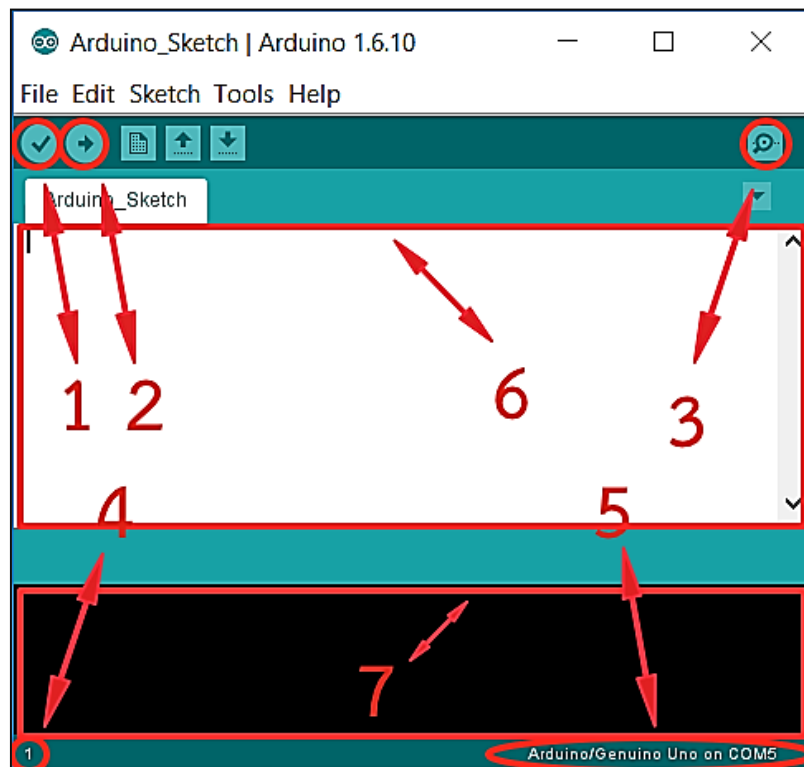
int addNumber(int x, int y)
{
    return x+y;
}

void main()
{
    ans = addNumber(3,6); // สุดท้ายจะได้ ans เท่ากับ 3+6 = 9
}
```

การเรียกใช้ฟังก์ชันแบบ `ans = addnumber(3,6);` สิ่งที่เกิดขึ้นคือจะมี การส่งจำนวน 3 และ 6 เข้าสู่ฟังก์ชันในฐานะของตัวแปร x และ y ตามที่ ประกาศไว้ และฟังก์ชันก็จะคืนค่าผลบวกของสองจำนวนนี้กลับไปเก็บไว้ใน ตัวแปร ans โดยเมื่อรันเสร็จแล้วตัวแปร ans จะมีค่าเท่ากับ 9 ในที่สุด

การใช้งานซอฟต์แวร์ Arduino เพื่อเขียนโปรแกรมหุ่นยนต์ mBot

สำหรับการเขียนโปรแกรมควบคุมหุ่นยนต์ mBot ที่เราใช้ในครั้งนี้ จะต้องใช้ซอฟต์แวร์ที่มีชื่อว่า “Arduino” เพื่อทำการ Compile และ Upload โค้ดโปรแกรมของเราเข้าสู่หน่วยความจำภายในของหุ่นยนต์ mBot โดยตัวโปรแกรมจะเป็นไอคอนสีเขียวมินต์ (∞) อยู่บนหน้าจอ Desktop ซึ่งเมื่อเราเข้าโปรแกรมแล้วก็จะปรากฏหน้าต่างขึ้นมาดังนี้

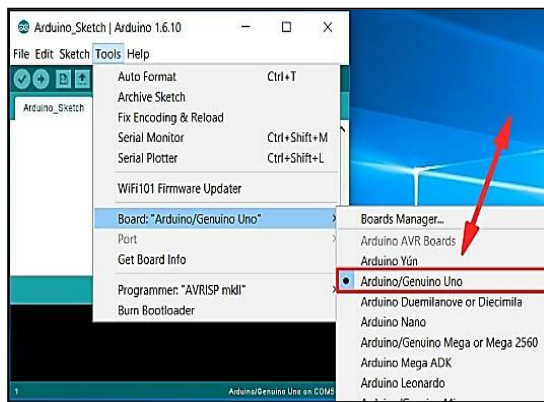


ส่วนประกอบในหน้าต่างโปรแกรม Arduino

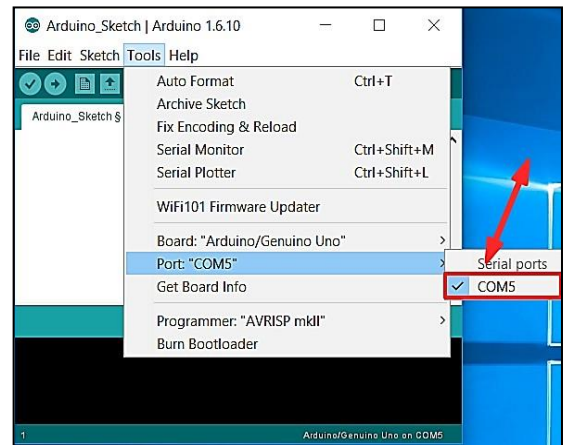
1. **Verify** : ปุ่มสำหรับการ Compile (ตรวจสอบ) โค้ดโปรแกรม
2. **Upload** : เมื่อกดปุ่มนี้ ตัวซอฟต์แวร์ Arduino จะทำการ Compile และทำการ Upload โค้ดโปรแกรมที่เราเขียนเข้าสู่หน่วยความจำภายใน ไมโครคอนโทรลเลอร์ของหุ่น mBot ทั้งนี้หากไม่พบความผิดพลาดใดๆ
3. **Serial Monitor** : เป็นปุ่มสำหรับเปิดหน้าต่างที่ใช้ในการสื่อสาร กับหุ่น mBot ในรูปแบบของ Serial (การสื่อสารผ่านพอร์ตอนุกรม) โดย เราสามารถดูการแสดงผลของข้อความต่างๆ ตามโปรแกรมที่เขียนไว้ได้จาก หน้าต่างนี้
4. หมายเลขบรรทัดที่เรากำลังพิมพ์อยู่ในขณะนี้ (ตำแหน่งของเคอร์เซอร์)
5. ส่วนแสดงข้อมูล, รุ่นของไมโครคอนโทรลเลอร์ภายในตัวหุ่นยนต์ mBot และหมายเลขของพอร์ต COM ที่กำลังเชื่อมต่อกับหุ่น mBot ในขณะนี้
6. หน้าต่างสำหรับการเขียนโค้ดโปรแกรม
7. **Status Bar** : เป็นแถบแสดงสถานะและข้อมูล
(เช่น บอกว่า Compile ผ่านไหม หากไม่ผ่านคาดว่าเกิดจากอะไร เป็นต้น)

การเชื่อมต่อกับหุ่น Mbot

หุ่นยนต์ mBot จะเชื่อมต่อกับคอมพิวเตอร์โดยใช้สาย USB ซึ่งหลังจากทำการเสียบสายและเปิดสวิตช์บนตัวหุ่นยนต์ mBot แล้ว หากจะทำการอัปโหลดโปรแกรมที่เขียนบนซอฟต์แวร์ Arduino จะต้องทำ การตั้งค่าการเชื่อมต่อต่าง ๆ ดังต่อไปนี้



1.เปิด



โปรแกรมแล้วกดที่เมนู Tools -> Board แล้วเลือก ตรงส่วนของ Board เป็น

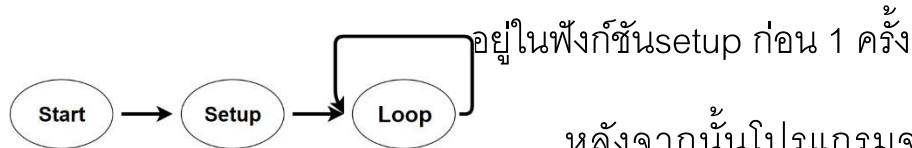
“Arduino/Genuino Uno”

2.จากนั้นกดที่เมนู Tools -> Port แล้ว เลือกพอร์ต COM x จากในตัวเลือกที่ขึ้นมา เช่น ในที่นี้คือ COM5

(หมายเลขของพอร์ต COM อาจไม่เหมือนกับในตัวอย่าง)

การเขียนโปรแกรมภาษา C บน Arduino

ปกติในการเขียนโปรแกรมในภาษาต่างๆ ฟังก์ชันแรกที่โปรแกรม จะทำงานคือ ฟังก์ชันที่มีชื่อว่า main แต่ในโปรแกรม Arduino นี้จะมีฟังก์ชันหลักอยู่ 2 ฟังก์ชันคือ ฟังก์ชัน setup และ loop โดยจะมีหลักการอยู่ว่า เมื่อเริ่มทำงานโปรแกรมจะทำคำสั่งที่



หลังจากนั้นโปรแกรมจะ ทำคำสั่งภายในใน ฟังก์ชัน loop ซึ่งจะมีการทำงานเป็นลักษณะที่มีการวนซ้ำๆ คือเมื่อทำงานจนไปถึงคำสั่ง สุดท้ายของฟังก์ชัน loop ก็จะกลับมาทำงานที่คำสั่งแรกใหม่อีกครั้ง

ดังนั้นเมื่อเริ่มต้นเขียนโปรแกรม Arduino สิ่งแรกที่เราควรทำ คือการประกาศ ฟังก์ชัน setup() และ loop() ขึ้น โดยเมื่อ ประกาศเสร็จแล้วให้ลองกดปุ่ม Verify ดู ถ้าหากพิมพ์ถูกต้องจะ ไม่ขึ้น Error ใดๆ และเมื่อกด Upload หากไม่มีปัญหาอะไร ก็จะ ขึ้นสถานะว่า Done uploading. ซึ่งหมายความว่าโปรแกรมที่เรา เขียนนั้น ถูกอัปโหลดเข้าหุ่น mBot เรียบร้อยแล้ว

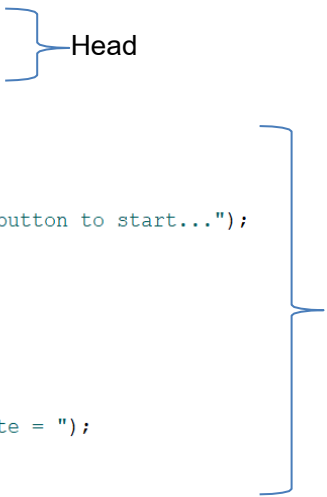


โครงสร้างของโปรแกรมบน Arduino

```
#include <iBotC.h>
#define PI 3.14159
int btn;

void setup()
{
    iBotC();
    println("Press a button to start...");
    waitButton();
}

void loop()
{
    btn = button();
    print("Button state = ");
    println(btn);
}
```



โครงสร้างนั้นมี 2 ส่วนคือ

1. Program head ส่วนหัวของโปรแกรม ใช้เขียนคำสั่งพิเศษบางอย่าง ก่อนเข้าสู่การทำงาน หรือกำหนดค่าเริ่มต้น การทำงานเริ่มต้นต่างๆ เช่น #include <...> จะนำไฟล์ไลบรารีที่มีชื่ออยู่ในเครื่องหมาย < > มารวมเข้า กับโปรแกรม เช่น #include <IBotC.h> เพื่อให้สามารถใช้ฟังก์ชันและ คำสั่งในไลบรารีนั้นๆได้ (กรณีนี้คือไลบรารีชุดคำสั่งสำหรับหุ่นยนต์ mBot) #define เป็นการนิยามให้ค่าๆหนึ่งในโปรแกรมมีความหมายที่เราต้องการ เช่น #define PI 3.14159 จะทำให้ทุกที่ในโปรแกรมของเราที่มีคำว่า PI มีความหมายว่า 3.14159 ซึ่งช่วยให้เราไม่ต้องพิมพ์ค่าเดิมซ้ำๆ หรือต้องตามไปแก้ตัวเลขในโปรแกรมทุกๆที่

ตัวอย่างของส่วนหัวของโปรแกรม

```
#include <iBotC.h>
#define PI 3.14159
int btn;
```

2. Program body จะเป็นส่วนที่เราใช้ในการกำหนดการทำงานต่างๆ ของโปรแกรม ซึ่งแนวทางในการเขียนก็จะขึ้นอยู่กับเทคนิคของแต่ละคน ใน Arduino นั้นจะเริ่มทำงานจากฟังก์ชัน void setup() ครั้งเดียวก่อน แล้วจะเข้าไปทำงานที่ void loop() ซึ่งเป็นฟังก์ชันที่ทำงานวนซ้ำไปเรื่อยๆ

```
void setup()
{
    iBotC();
    println("Press a button to start...");
    waitButton();
}

void loop()
{
    btn = button();
    print("Button state = ");
    println(btn);
}
```

การแสดงผลเบื้องต้น

ใน Arduino มีคำสั่งที่ใช้แสดงผลทางหน้าจอโดยสามารถแสดงได้ ทั้งข้อความ และ ค่าของตัวแปร โดยมีรูปแบบการใช้คำสั่งดังนี้

```
Serial.print("ข้อความ");
```

```
Serial.print(ตัวแปร);
```

```
Serial.println("ข้อความ");
```

```
Serial.println(ตัวแปร);
```

คำสั่ง Serial.println(); นั้นจะแสดง ข้อความหรือค่าตัวแปร เสร็จแล้วจะขึ้น บรรทัดใหม่ ส่วน Serial.print(); นั้น จะไม่มีการขึ้นบรรทัดใหม่ และก่อนที่จะใช้คำสั่งที่เกี่ยวข้องกับ Serial นั้น เราจะต้องสั่งเปิดช่องการสื่อสารแบบ Serial และกำหนด baudrate ก่อน ด้วยคำสั่ง Serial.begin(9600);

```
#include <iBotC.h>

int x = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Serial.print("I BOT CAMP : ");
  Serial.println(x);
  delay(1000);
  x++;
}
```

คำสั่ง delay(); คือคำสั่งที่จะทำการหน่วงเวลาการกระทำของคำสั่งก่อน หน้าไว้ชั่วคราว ซึ่งจะค้างเป็นเวลาที่เราระบุไว้ในวงเล็บด้านหลังแล้วจึง ทำงานต่อ โดยเวลาที่เราระบุไว้ จะมีหน่วยเป็นมิลลิวินาที (ms)

ชุดคำสั่งต่าง ๆ ของหุ่น mBot การเรียกใช้งานไลบรารีชุดคำสั่ง iBotC

Library คือ ชุดคำสั่งที่ถูกสร้างขึ้นและถูกรวบรวมไว้เพื่อให้ เราสามารถนำมาเขียนโปรแกรมได้ง่ายขึ้น โดย #include <iBotC.h> เป็นคำสั่งสำหรับการเรียกใช้ไลบรารีสำหรับหุ่น mBot และคำสั่ง iBotC(); เป็นคำสั่งที่ใช้เตรียมความพร้อมของหุ่น mBot เพื่อที่จะใช้งานกับไลบรารี

โดยคำสั่งนี้จะต้องถูกใช้งานในฟังก์ชัน `setup()` เป็นคำสั่งแรกเสมอ โดย โครงร่างของการเขียนโปรแกรมสำหรับหุ่น mBot บนซอฟต์แวร์ Arduino น้องๆสามารถเริ่มต้นได้โดยการเขียนโปรแกรมตามนี้เลย

```
#include <iBotC.h>

void setup()
{
    iBotC();
}

void loop()
{
```

การขับเคลื่อนมอเตอร์ของหุ่น mBot

เป็นคำสั่งที่ทำให้มอเตอร์ที่ขับเคลื่อนล้อต่าง ๆ

`motor(sp1, spd2);`

เป็นคำสั่งที่ทำให้มอเตอร์ที่ขับเคลื่อนล้อ หมุนตามความเร็วที่กำหนด โดย `spd1` เป็นความเร็วของล้อข้างซ้าย และ `spd2` เป็นความเร็วของล้อขวา หากใส่ค่าเป็นบวกจะทำให้ล้อหมุนไปด้านหน้า และหากเป็นลบจะทำให้ล้อ หมุนไปด้านหลัง การใส่ค่าเป็น 0 จะทำให้ล้อหยุดหมุน (ค่าสูงสุดคือ 255) โดยคำสั่งนี้จะไม่มีการกำหนดเวลาในการทำงาน เมื่อสั่งแล้วล้อจะหมุนไปเรื่อยๆ จนกว่าจะมีการสั่งให้หยุด หรือสั่งให้เปลี่ยนแปลงความเร็ว

`motor(sp1, spd2,time);`

เป็นคำสั่งที่ใช้ในการขับเคลื่อนล้อเช่นเดียวกันกับคำสั่งที่แล้ว แต่จะเพิ่ม parameter มาอีก 1 ตัวคือ time สำหรับใช้กำหนดว่าต้องการให้ล้อหมุน เป็นเวลานานเท่าใด โดยเวลาที่กำหนดจะมีหน่วยเป็น milliseconds (1000 ms = 1 วินาที) และเมื่อสิ้นสุดช่วงเวลาที่กำหนดแล้วล้อจะ หยุดหมุนโดยอัตโนมัติ

ตัวอย่าง

```
motor(-100,-100);           //เดินถอยหลังด้วยความเร็ว 100
delay(1000);                 // หน่วงเวลา 1 วินาที
motor(0,0);                  //สั่งให้ล้อทั้งสองข้างหยุดหมุน
motor(-100,100,1000);       // หมุนทวนเข็มนาฬิกาเป็นเวลา 1 วินาที
```

การใช้งานหลอดไฟ LED บนตัวหุ่น mBot

สำหรับหลอดไฟ LED ที่อยู่บนตัวหุ่น mBot น้อย ๆ สามารถใช้ คำสั่งเพื่อเปิดหลอดไฟให้ติดเป็นสีตามที่กำหนดได้ โดยใช้คำสั่งดังนี้

โดยสีที่น้อย ๆ สามารถเลือกใช้ได้ จะมีดังนี้

สี	ขาว	แดง	เหลือง	เขียว	น้ำเงิน	ฟ้า	ม่วง
ชื่อ	"White"	"Red"	"Yellow"	"Green"	"Blue"	"Cyan"	"Purple"

และถ้าน้องๆต้องการปิดหลอดไฟ LED ก็สามารทำได้โดยใช้คำสั่งนี้

```
led("Off");
```

ตัวอย่าง

```
led("White"); // เปิดหลอดไฟเป็นสีเขียว
delay(1000); // หน่วงเวลา 1 วินาที
led("Red"); // เปิดหลอดไฟเป็นสีแดง
delay(1000); // หน่วงเวลา 1 วินาที
led("Green"); // เปิดหลอดไฟเป็นสีเขียว
delay(1000); // หน่วงเวลา 1 วินาที
led("Off"); // ปิดหลอดไฟ LED
```

การใช้งานลำโพง Buzzer บนหุ่น mBot

บนหุ่น mBot มีลำโพง Buzzer ซึ่งสามารถส่งเสียงออกมาได้ ซึ่งอาจจะมีประโยชน์ในการใช้ตรวจสอบการทำงานของโปรแกรมได้ด้วย ในการใช้งาน สามารถทำได้โดยใช้คำสั่ง

```
beep([จำนวนครั้ง]);
```

ซึ่งจะทำให้ลำโพง Buzzer ส่งเสียงดังปิ๊บๆ ออกมาตามจำนวนครั้ง ที่เรากำหนดได้

```
int x= button();
```

การใช้งานปุ่มกด Tact switch บนหุ่น mBot

บนหุ่น mBot มีปุ่มกดเอนกประสงค์ที่สามารถนำไปใช้เพิ่ม ความสามารถในการเขียนโปรแกรมให้ตอบสนองต่อการกดปุ่มของผู้ใช้ได้ โดยการอ่านค่าสถานะของปุ่มจะใช้คำสั่งดังนี้

โดยหากปุ่มถูกกด ตัวแปร x จะมีค่าเป็น 1 และหากไม่กดจะมีค่าเป็น 0 นอกจากนี้ยังสามารถใช้คำสั่งเพื่อให้หุ่นยนต์รอการกดปุ่มจากเราก่อนที่จะเริ่มการทำงานของโปรแกรมได้ โดยใช้คำสั่ง

```
waitButton();
```

การใช้งานเซนเซอร์ตรวจจับเส้น

ที่บริเวณด้านหน้าของหุ่น mBot จะมีเซนเซอร์ตรวจจับเส้นที่สามารถนำมาใช้ในการเขียนโปรแกรมเพื่อให้หุ่นยนต์เดินตามเส้นได้ โดยที่ บนตัวเซนเซอร์จะมีจุดอ่านค่าสีอยู่ทั้งหมด 4 จุด ได้แก่ จุดที่ 0,1,2 และ 3 ตามลำดับ ในการใช้งานจะต้องประกาศตัวแปรขึ้นมาเพื่อเก็บค่าที่อ่านได้ เช่น สำหรับการอ่านค่าจากเซนเซอร์ ณ จุดที่ 0 (ซ้ายสุด) จะใช้คำสั่งดังนี้

```
int s0 = sensor0();
```

ซึ่งค่าที่ได้จะมีค่าเป็น 1 หากที่จุดนั้นเป็นสีของเส้นสนาม และจะได้ค่าเป็น 0 หากอ่านได้สีพื้นหลังของสนาม

***หมายเหตุ ก่อนการใช้งานน้องๆจะต้องนำหุ่นไปทำการจดจำค่าสีของเส้นและพื้นสนาม โดยการนำหุ่นไปวางให้เซนเซอร์ตรวจจับเส้น อยู่บนพื้นที่สีเส้นสนาม แล้วกดปุ่มบนเซนเซอร์ 1 ครั้ง (รอจนไฟกระพริบเสร็จ) ต่อจากนั้นให้นำหุ่นไปวางให้เซนเซอร์ตรวจจับเส้นอยู่บนพื้นหลังสนามแล้วกดปุ่ม 2 ครั้ง และรอให้ไฟกระพริบจนเสร็จอีกเช่นกัน จากนั้น ให้ตรวจสอบโดยลองนำเซนเซอร์ไปวางบนเส้น โดยจุดที่ตรวจจับสีเส้นสนามได้จะมีไฟติดขึ้นมา

ตัวอย่างการใช้งาน

```
waitButton(); // รอรับคำสั่งด้วยการกดปุ่ม
while(true)
{
    int s0 = sensor0(); // เก็บค่า sensor0 ไว้ในตัวแปร s0
    int s1 = sensor1(); // เก็บค่า sensor1 ไว้ในตัวแปร s1
    int s2 = sensor2(); // เก็บค่า sensor2 ไว้ในตัวแปร s2
    int s3 = sensor3(); // เก็บค่า sensor3 ไว้ในตัวแปร s3

    if (s1 == 1 && s2 == 1) // ถ้าเซนเซอร์ทั้งสองอยู่บนเส้นสนาม
    {
        // หุ่นจะเดินหน้าไปเล็กน้อย
        motor(100, 100, 100);
    }
    // ถ้าหุ่นหลุดเส้นไปทางขวา
    if (s1 == 1 && s2 == 0)
    {
        // หุ่นจะเลี้ยวซ้ายกลับมาหาเส้น
        motor(100, 200, 100);
    }
    // ถ้าหุ่นหลุดเส้นไปทางซ้าย
    if (s1 == 0 && s2 == 1)
    {
        // หุ่นจะเลี้ยวขวากลับมาหาเส้น
        motor(200, 100, 100);
    }
}
```

ส่วนของโปรแกรมตัวอย่างต่อไปนี้จะเป็นการเขียนให้หุ่นยนต์ เดินตามเส้นไปเรื่อยๆ จนถึงทางแยกแล้วจบการทำงาน โดยโปรแกรม จะเริ่มการทำงานเมื่อกดปุ่ม

การใช้งานเซนเซอร์วัดระยะทางอัลตราโซนิก

ที่บริเวณด้านหน้าของหุ่นจะมีเซนเซอร์อีกตัวหนึ่งที่หันไปด้านหน้า ซึ่งมีหน้าที่ในการวัดระยะห่างจากวัตถุหรือสิ่งกีดขวางที่อยู่บริเวณหน้าหุ่น โดยในการใช้งานจะต้องประกาศตัวแปรขึ้นมาเพื่อเก็บค่าระยะห่าง ดังนี้

```
int distance = ultraSonic();
```

โดยค่าที่ได้มาจะเป็นระยะห่างจากสิ่งกีดขวางที่พบหน้าหุ่น mBot โดยจะมีหน่วยเป็นเซนติเมตร และมีตัวอย่างการประยุกต์ใช้ดังนี้

ตัวอย่างการใช้งาน

```
waitButton(); // รอให้กดปุ่มเพื่อเริ่มการทำงาน

while(true) // กำหนดให้ทำงานแบบวนซ้ำๆ
{
    int distance = ultraSonic();

    if(distance < 10) // หากพบสิ่งกีดขวางในระยะน้อยกว่า 10 ซม.
    {
        led("Red"); // สั่งให้หลอดไฟ LED ติดเป็นสีแดง
    }
}
```


บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

บันทึกช่วยจำ (MEMO)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

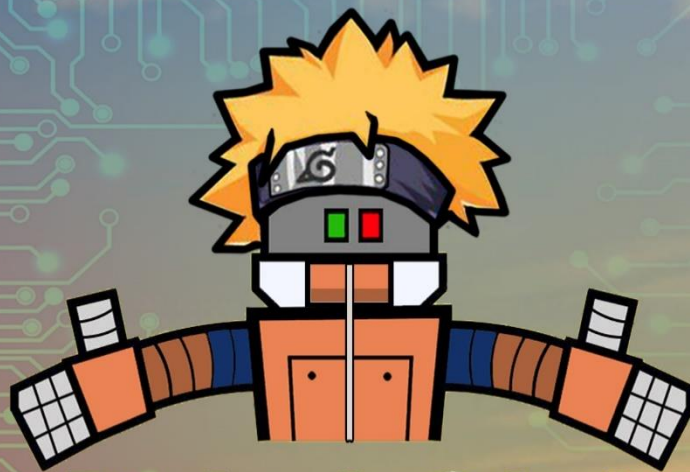
.....

.....

.....

.....

.....



Robot Club

Engineering Ladkrabang

Imagineering



@NALC