

# INDEX



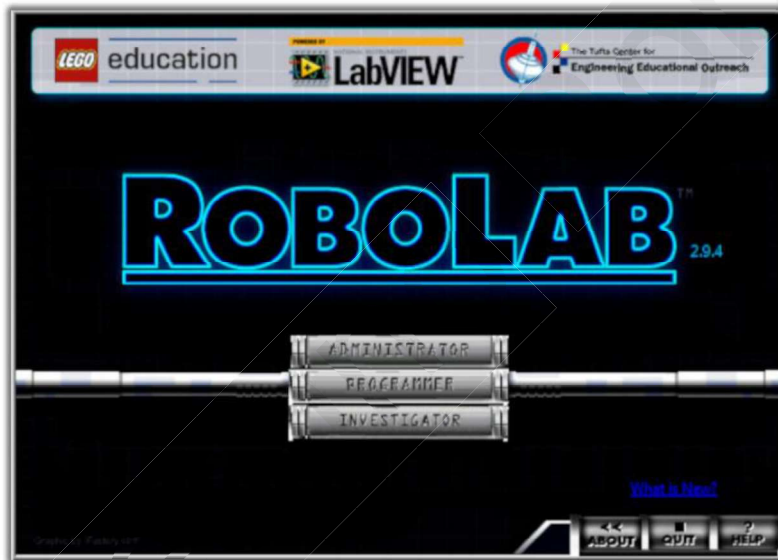
|                                  | Page  |
|----------------------------------|-------|
| THE ROBOLAB                      |       |
| - เริ่มต้นกับ ROBOLAB            | 1     |
| ADMINISTRATOR                    | 2     |
| PROGRAMMER                       |       |
| - Pilot                          | 3     |
| - Inventor                       | 4-5   |
| TOOL PALETTE                     | 6     |
| Icon - Arrange                   | 7     |
| Functions                        | 8     |
| - Wait For                       | 9-10  |
| - Structures                     | 11    |
| - Fork                           | 12    |
| - Jumps                          | 13-14 |
| - Loop                           | 15-16 |
| - Subroutine                     | 17-18 |
| -Modifiers                       | 19    |
| Workshop 1                       |       |
| - Line tracking                  | 20-23 |
| Workshop 2                       |       |
| - Line tracking ระบุเส้นตัด      | 24    |
| Workshop 3                       |       |
| - Remote Robot โดยเซนเซอร์สัมผัส | 25-26 |

# THE ROBOLAB

โปรแกรม RoboLab นี้ เป็นโปรแกรมพื้นฐานที่ใช้เขียนคำสั่งควบคุมการทำงานของ Intelligent NXT Brick เป็นซอฟต์แวร์ที่ได้รับการพัฒนาขึ้นจาก Lab View Software โดยบริษัท National Instruments ลักษณะโปรแกรมเป็นการใช้ไอคอนรูปภาพต่างๆ มาเขียนเรียงกันเป็นชุดคำสั่ง โดยใช้หลักการที่เรียกว่า "Drag & Drop" เปรียบเหมือนการต่อวงจรที่ต้องใช้สายไฟ (WIRE) เชื่อมระหว่างจุดต่างๆ การทำงานของโปรแกรมจะเดินตามสายไฟที่เราสร้างขึ้นอย่างเป็นระบบเริ่มตั้งแต่ไฟเขียว (Begin) จนไปถึงไฟแดง (End) ทำให้ผู้เรียนสะดวกกับการเรียนรู้และเข้าใจได้อย่างรวดเร็ว

นอกจากโปรแกรม Robolab ที่กล่าวมาแล้ว NXT ยังสามารถรองรับการเรียนโปรแกรมประเภทอื่นๆ ได้อีก อาทิเช่น LEGO  
® MINDSTORMS ® Education NXT Software , ภาษา C , Visual Basic, JAVA และอื่นๆ สามารถหาข้อมูลเพิ่มเติมได้ที่เว็บไซต์  
[www.gammaco.com/wro](http://www.gammaco.com/wro) , [www.lego.com/education](http://www.lego.com/education) , [www.csee.tufts.edu](http://www.csee.tufts.edu) , <http://www.legoengineering.com/content>,  
[www.lugnet.com](http://www.lugnet.com)

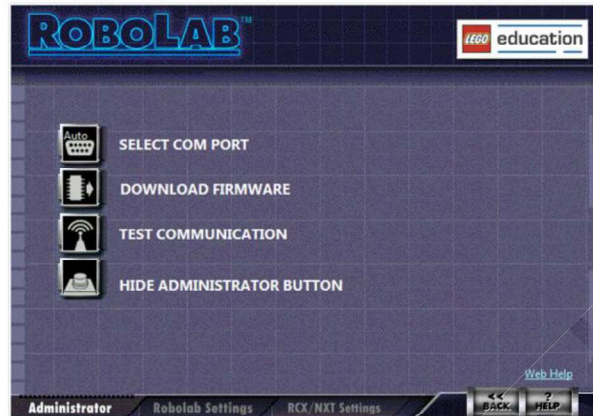
## เริ่มต้นกับ ROBOLAB



ทำการคลิกที่ ไอคอน ROBOLAB 2.5.9 หรือภายในเมนู START จากนั้นจะพบกับหน้าจอแนะนำโปรแกรม ROBOLAB ที่ต้องการใช้งานเราจะเข้าที่ เมนู Run ROBOLAB จากนั้นจะพบกับอีก 3 เมนูหลักที่ใช้ในการเขียนโปรแกรม คือ

1. **ADMINISTRATOR** ใช้สำหรับจัดการ Hardware ต่างๆ ของหุ่นยนต์ เช่นการกำหนดพอร์ตใช้งานที่ติดต่อกับ USB Transmitter การ Firmware การทดสอบการติดต่อสื่อสารระหว่าง NXT กับ Computer
2. **PROGRAMMER** เป็นเมนูที่ใช้ในการเขียนโปรแกรมควบคุมการทำงานของ NXT
3. **INVESTIGATOR** ใช้สำหรับการเขียนคำสั่งการทำงาน การทดลองทางด้านวิทยาศาสตร์ การเก็บข้อมูลต่างๆ ในลักษณะของ Data logging

คำสั่งใน ADMINISTRATOR ประกอบด้วย



1. SELECT COM PORT เมื่อเราติดตั้ง USB Transmitter เข้ากับ Computer จะต้องเลือกพอร์ตที่ใช้งานให้กับคอมพิวเตอร์ด้วย โดยปกติแล้วจะต่อเข้ากับ USB พอร์ต (Default Auto detect)
2. DOWNLOAD FIRMWARE เมื่อเราประกอบชุด NXT และใส่แบตเตอรี่เรียบร้อยแล้ว ในครั้งแรกที่ใช้งานจะต้องทำการ Download Firmware ก่อนที่จะใช้งานเสมอ การ Firmware คล้ายกับการลงปฏิบัติการให้กับ NXT นั่นเอง ให้ NXT รู้จักส่วนประกอบของตัวเอง การ Download FIRMWARE ใช้เวลาประมาณ 3-5 นาที การ Download Firmware นี้จะทำในครั้งแรกเพียงครั้งเดียว การดาวน์โหลดโปรแกรมในครั้งต่อไปไม่จำเป็นต้อง Download Firmware ใหม่ ยกเว้นในกรณีที่มีการเปลี่ยนแบตเตอรี่ใหม่เท่านั้นระหว่างการใช้งานปกติ ขณะที่เราดาวน์โหลดโปรแกรม NXT อยู่ นั้น มีความเตือนให้ Download Firmware ขึ้นมาโดยไม่ทราบสาเหตุ (โดยที่ไม่ได้ถอดแบตเตอรี่ออกหรือเปลี่ยนใหม่) อาจเกิดจากการที่ขั้วแบตเตอรี่หลวมก็ได้
3. TEST NXT COMMUNICATION เมื่อเราติดตั้ง สาย USB Transmitter เรียบร้อยแล้วสามารถทดสอบการติดต่อสื่อสารระหว่าง Computer และ NXT ได้โดยใช้เมนูนี้ หาก NXT ไม่พร้อมที่จะใช้งานก็จะมีข้อความเตือนขึ้นมาที่หน้าจอตัวอย่าง อาจเกิดจากสาเหตุการไม่เปิดตัว NXT หรือสาเหตุอื่น เช่น พอร์ตที่เสียบหลวม



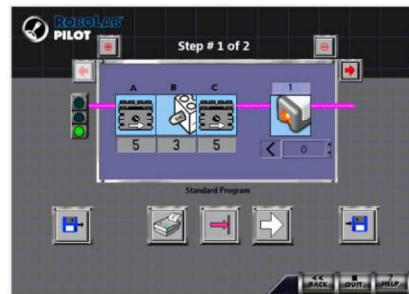
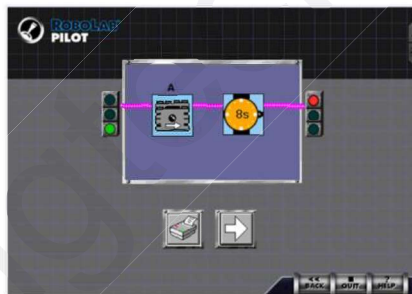
## PROGRAMMER

เป็นเมนูที่ใช้ในการเขียนโปรแกรมสั่งงาน NXT



### 1. Pilot

เป็นโปรแกรมแนะนำการเขียนโปรแกรมเบื้องต้นสำหรับผู้เริ่มต้นใช้งาน เป็นการแนะนำให้รู้จักโครงสร้างโปรแกรม จะมีการกำหนดลำดับขั้นตอนของโปรแกรมมาให้แล้ว เมื่อต้องการก็ให้เลือกคุณสมบัติที่เราต้องการในช่องลำดับการทำงานนั้นๆ ระดับการทำงาน Pilot แบ่งออกเป็น 4 ระดับ



## PROGRAMMER

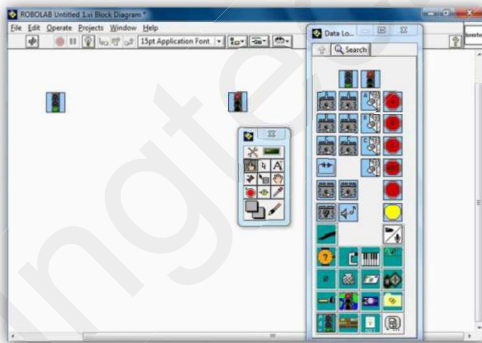
## เป็นเมนูที่ใช้ในการเขียนโปรแกรมสั่งงาน NXT

## 2. Inventor

โปรแกรมนี้ช่วยให้นักเรียนสร้างหุ่นยนต์ด้วยตนเองเป็นการโปรแกรมโดยอาศัยหลักการ "Drag & Drop" คือการเลือกไอคอนที่ต้องการใช้งาน แล้วมาวางเรียงต่อกันเป็นลำดับของโปรแกรมเนื่องจาก เราสามารถเลือกใช้งานไอคอนและสามารถลำดับการทำงานต่างๆ ได้อิสระผู้เรียนจึงสามารถประยุกต์โปรแกรมให้มีความหลากหลายมากขึ้น Inventor สามารถแบ่งระดับการใช้งาน ออกเป็น 4 ระดับถ้าหาก เลือกใช้งานในระดับต่ำ (Inventor 1) ก็จะมีไอคอนการใช้งานให้เลือกน้อยกว่า แต่ถ้าหากเลือกเป็น Inventor 4 จะมีไอคอนให้ครบทุกตัว



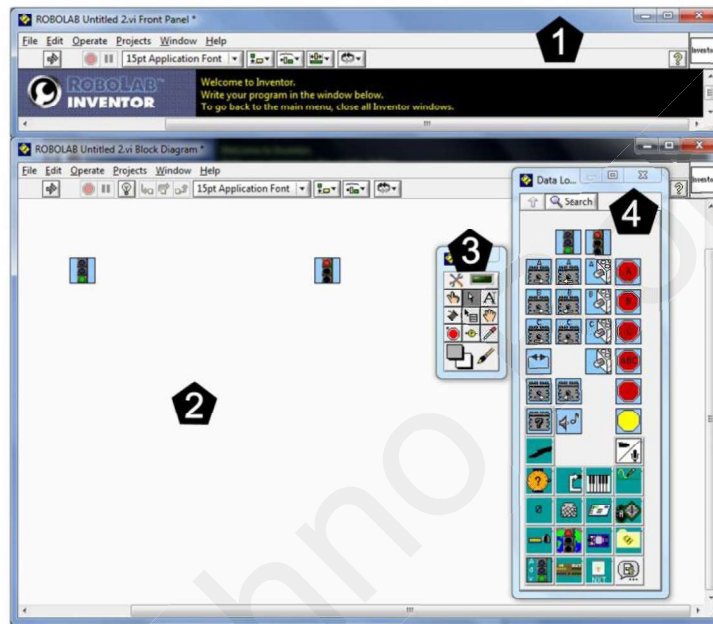
Inventor 1 เป็นระดับแรก  
ในการเขียนโปรแกรม  
สามารถสร้างโปรแกรม  
ได้ด้วยฟังก์ชันพื้นฐาน



Inventor 4 เป็นระดับสุดท้าย  
รวบรวมความสามารถทั้งหมด  
ที่หุ่นยนต์สามารถทำได้  
โดยจะเห็นได้ว่ามีฟังก์ชัน  
มากกว่าระดับ 1 2 และ 3

## โปรแกรม Inventor

โปรแกรม Inventor เป็นโปรแกรมต่อเนื่องจากโปรแกรม Pilot การสร้างโปรแกรมทำได้โดยการเลือกและวางไอคอนคำสั่งจาก Functions palette ไปยังหน้าต่าง Diagram เมื่อเชื่อมต่อคำสั่งเหล่านี้เข้าด้วยกันจะเกิดการสร้างโปรแกรมขึ้นอย่างไม่จำกัด โปรแกรม Inventor มี 4 ระดับ โดยที่ระดับ 1 เป็นระดับที่ง่ายที่สุดและระดับ 4 สามารถทำการพลิกแพลงและสร้างสรรค์ได้เต็มที่ โปรแกรม Inventor ใช้ภาษาเขียนโปรแกรม Lab VIEW



หน้าต่าง Panel

1 หน้าต่าง Panel นี้จะไม่ได้ใช้ในโปรแกรม Inventor แต่ต้องเปิดไว้เพื่อให้โปรแกรมทำงาน การปิดหน้าต่าง Panel จะเป็นการปิดหน้าต่าง Diagram และ Functions palette และเป็นการกลับเข้าสู่เมนูหลัก

2

Tools Palette  
พื้นที่สำหรับสร้าง

3

Functions Palette  
ประกอบด้วยไอคอนคำสั่ง  
ที่ใช้ในการสร้างโปรแกรม

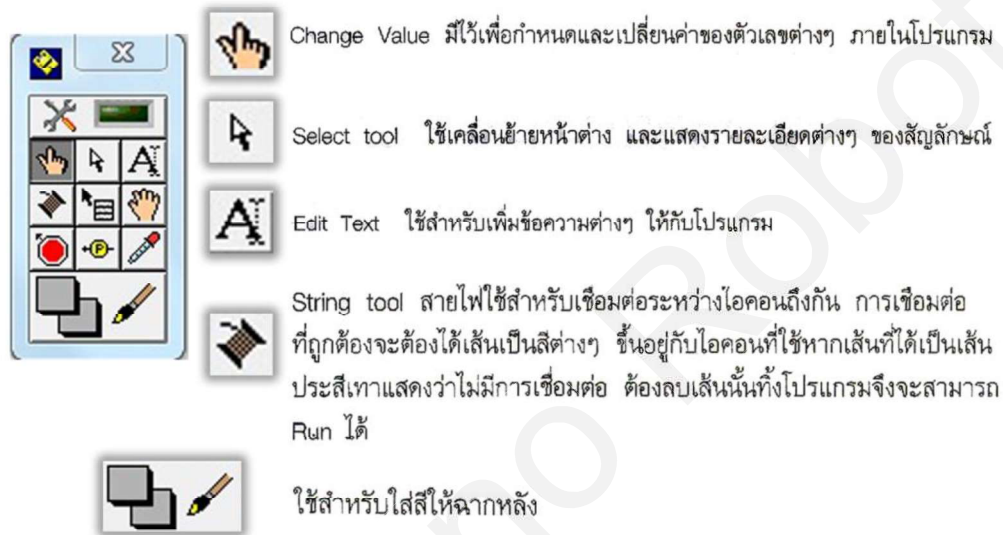
4

หน้าต่าง Diagram  
พื้นที่สำหรับสร้างโปรแกรม

## หน้าต่างเครื่องมือ (Tool Palette)

เป็นหน้าต่างที่มีเครื่องมือต่างๆซึ่งใช้ในการเขียนโปรแกรม

หน้าต่างเครื่องมือประกอบด้วยสัญลักษณ์สำคัญดังนี้



## เทคนิค

เราสามารถเปลี่ยนเครื่องมือต่างๆ ได้เร็วขึ้น ด้วยการปุ่ม Tab และ Spacebar เช่น Tab เป็นการเปลี่ยนเครื่องมือระหว่าง Change Value , Select tool , Edit Text และ String tool ส่วน Spacebar เป็นการสลับระหว่าง Select tool และ String tool (ถ้าใช้ได้คล่องจะมีประโยชน์)

**รู้สัณณีก่อนเริ่มโปรแกรม :** การเชื่อมของไอคอนต่างๆ เข้าด้วยกันที่ปลายสาย (Terminal) ของไอคอนนั้นๆ ต้องเป็นสีเดียวกัน จึงจะเชื่อมกันได้โดยการแบ่งสีเป็นดังนี้

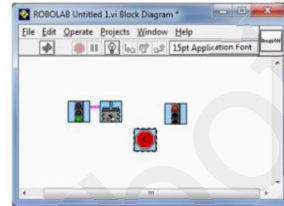
1. สีชมพู เป็นสายแสดงถึงเส้นหลักของโปรแกรมเป็นการดำเนินลำดับของโปรแกรม
2. สีน้ำเงิน เป็นค่าตัวเลขจำนวนเต็ม
3. สีส้ม เป็นตัวเลขทศนิยม
4. สีเขียว เป็นตำแหน่ง Address เช่นการกำหนด Port

## การเรียกใช้ไอคอนและการจัดเรียง

เมื่อต้องการใช้ไอคอน จาก Function Palette ให้ทำการเปลี่ยนเครื่องมือในฝั่งของ Tools Palette (เครื่องมือ) ให้เป็น Select Tool (ลูกศร) จากนั้นคลิกเลือกไอคอนที่ต้องการ และคลิกอีกครั้งเพื่อเป็นการวางไอคอนบนพื้นที่ สำหรับการเขียนโปรแกรม แต่หากต้องการเชื่อมไอคอน เข้าด้วยกันต้องใช้ String tool (หลอดค้าย) สังเกตว่าเมื่อเชื่อมต่อกันแล้วเส้นจะเป็นสีชมพูเสมอ หากเป็นสีดำนั้นแสดงว่าการเชื่อมต่อระหว่างไอคอน มีปัญหาต้องทำการแก้ไข (จะพูดถึงในหัวข้อถัดไป)

### การเปลี่ยนแปลง และลบไอคอน

ใช้เครื่องมือ Select tool (ลูกศร) เลือกไอคอนที่ต้องการลบ โดยคลิกที่ไอคอนค่าตั้งนั้น 1 ครั้งจะ active (เส้นประสีเทา) และกดปุ่ม delete เพื่อเอาไอคอนนั้นออก การลบสายเชื่อมต่อทำได้โดยการดับเบิ้ลคลิกที่สายและกดปุ่ม delete



### การเชื่อมต่ออัตโนมัติ

เมื่อเลือกและวางไอคอนบนหน้าต่างโปรแกรมแล้ว เส้นไอคอนให้ใกล้กัน สายเชื่อมต่อจะเชื่อมโดยอัตโนมัติ การเชื่อมต่ออัตโนมัตินี้ทำให้สร้างโปรแกรมได้เร็วขึ้นเนื่องจากไม่จำเป็นต้องใช้เครื่องมือเชื่อมต่อเพื่อเชื่อมไอคอนหรือตัวเปลี่ยนแปลง (modifiers) เข้าด้วยกัน หากต้องการใช้เครื่องมือเชื่อมต่อ (String tool) เพื่อเชื่อมไอคอน ก็สามารถทำได้เช่นกัน



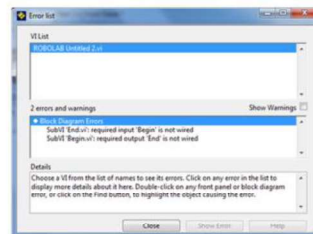
### ปัญหาจากการเชื่อมต่อไอคอน

ขณะกำลังเชื่อมต่อ หากส่วนใดส่วนหนึ่งของสายไม่เชื่อมติดกันจะเกิดเป็นเส้นประสีดำ โดยปกติการเชื่อมต่อเส้นควรเชื่อมติดกันเป็นสีชมพู หากเชื่อมต่อสายไปยังตำแหน่งที่ไม่ถูกต้องจะเกิดสายเป็นเส้นประสีดำ ลูกศรการทำงานจะเกิดเป็นภาพแตกด้วย (Run) หากไม่นำสายเชื่อมต่อที่ไม่ถูกต้องออก จะไม่สามารถทำการดาวน์โหลดข้อมูลได้ ถ้าแก้ไขการเชื่อมต่อโปรแกรมได้สมบูรณ์ไอคอน Run หรือ Download จะเป็นลูกศรสีขาว

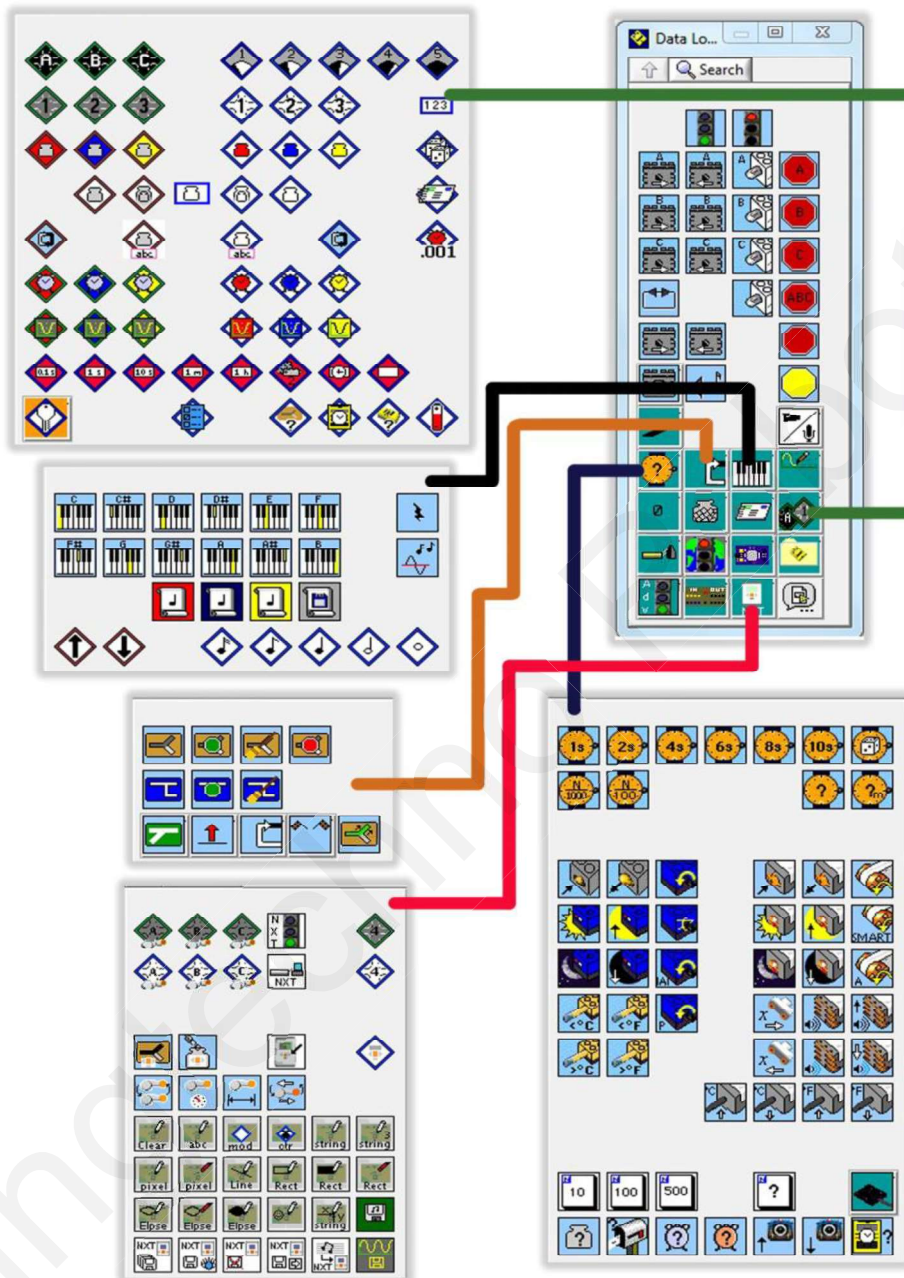


รู้ไว้ไว้ว่า : การลบเส้นที่เราไม่สามารถหาสาเหตุได้ หรือจำนวนเส้นที่ error มีจำนวนมาก ให้ใช้คำสั่ง Remove Broken Wires ใน Edit หรือใช้ CTRL+B (เรียกว่าเยอะ)

\* อีกกรณีหากการดาวน์โหลดปรากฏว่ามี error ลูกศรดาวน์โหลดเป็นรูปแตก โดยปกติไม่สามารถดาวน์โหลดได้ แต่หากเรายังคลิกดาวน์โหลด โปรแกรมจะทำการหาส่วนที่ error ของโปรแกรมให้



# Functions



## Wait for (ทำงานจนกระทั่ง)

เป็นไอคอนที่ใช้ในการกำหนดระยะเวลา  
ในการทำงานของหุ่นยนต์แบ่งเป็นสองชนิด

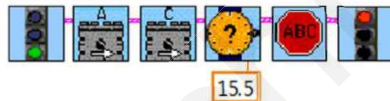
1. โดยเวลา
2. โดยเซนเซอร์



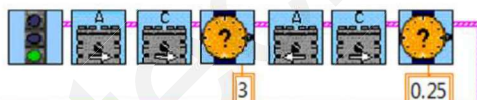
### 1. ตัวอย่างคำสั่งโดยใช้เวลา



มอเตอร์ A และ C เดินหน้า 4 วินาที



มอเตอร์ A และ C เดินหน้า 15.5 วินาที



เดินหน้า 3 วินาที เลี้ยวซ้าย 0.25 วินาที

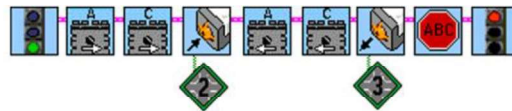


เลี้ยวขวา 0.25 วินาที ถอยหลัง 25 วินาที

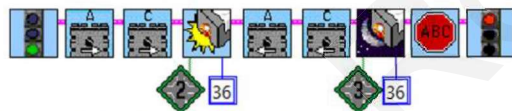
(ในกรณีมอเตอร์ ซ้ายคือA ขวาคือC)

## 2. ตัวอย่างคำสั่งโดยใช้เซนเซอร์

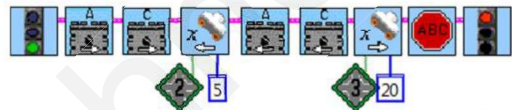
(กำหนดให้ มอเตอร์ ช่าย A ขาว C)



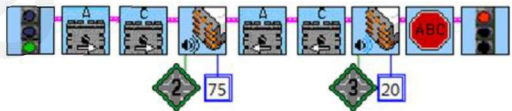
หุ่นยนต์เดินทางจนกระทั่งทัชเซนเซอร์ที่พอร์ต 2 จะถูกกดจากนั้นหุ่นยนต์จะถอยหลังจนกว่าทัชเซนเซอร์ที่พอร์ต 3 จะปล่อยจากนั้นมอเตอร์จึงหยุดทำงาน



หุ่นยนต์เดินทางจนกระทั่งไลต์เซนเซอร์ที่พอร์ต 2 จะเจอค่ามากกว่า 36 จากนั้นหุ่นยนต์จะถอยหลังจนกว่าไลต์เซนเซอร์ที่พอร์ต 3 จะค่าน้อยกว่า 36 จากนั้นมอเตอร์จึงหยุดทำงาน



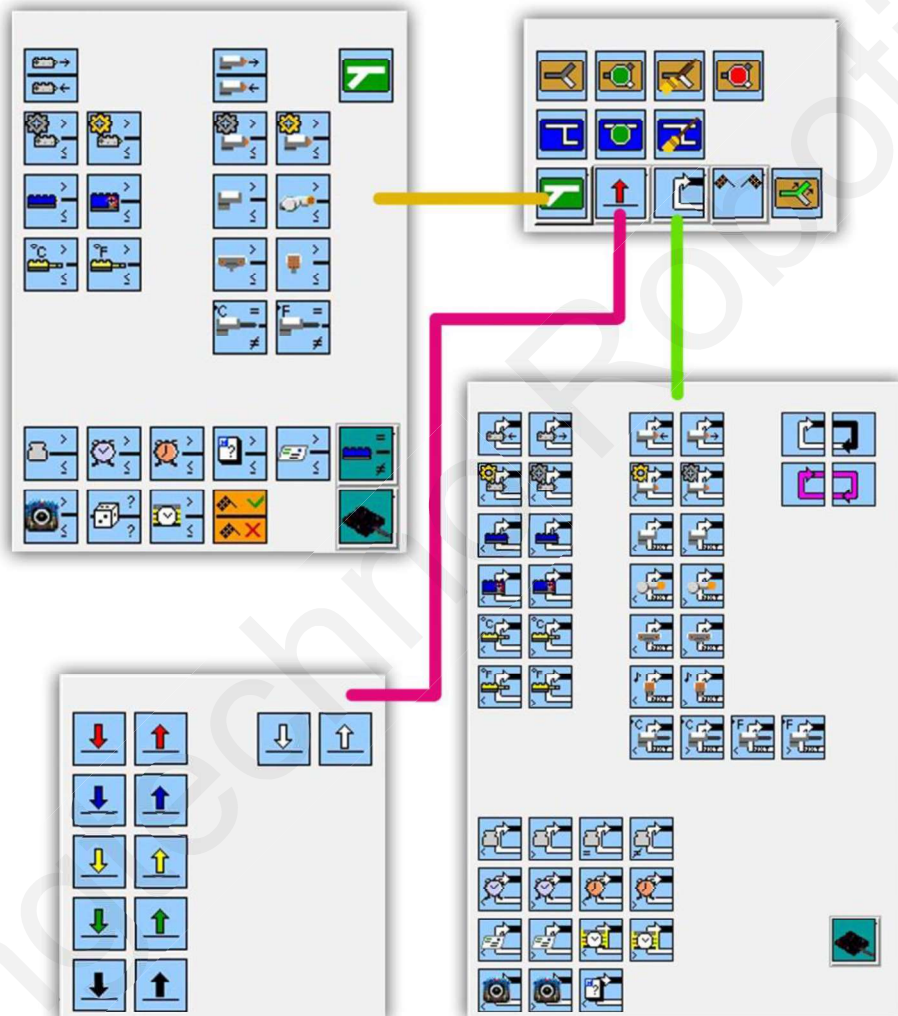
หุ่นยนต์เดินทางจนอัลตราโซนิกเซนเซอร์เจอค่าน้อยกว่า 5 ซม. จากนั้นหุ่นยนต์จะถอยหลังจนกว่าอัลตราโซนิกเซนเซอร์จะเจอค่ามากกว่า 20 ซม. จากนั้นมอเตอร์จึงหยุดทำงาน



หุ่นยนต์เดินทางจนกว่าชวาร์เซนเซอร์เจอค่ามากกว่า 75% จากนั้นหุ่นยนต์จะถอยหลังจนกว่าชวาร์เซนเซอร์จะเจอค่าน้อยกว่า 20% จากนั้นมอเตอร์จึงหยุดทำงาน

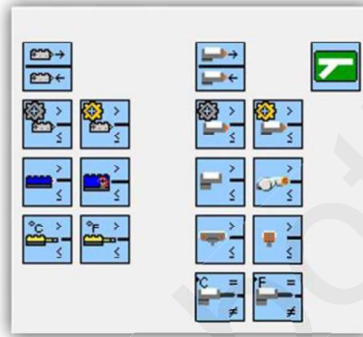
Structure (โครงสร้างโปรแกรม)

ประกอบไปด้วยเมนูย่อยสำคัญดังนี้

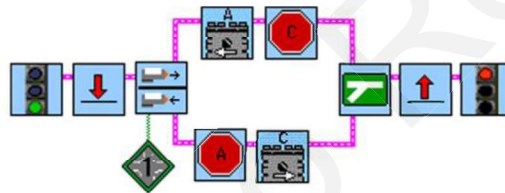


## Fork (การเปรียบเทียบโดยทางแยก)

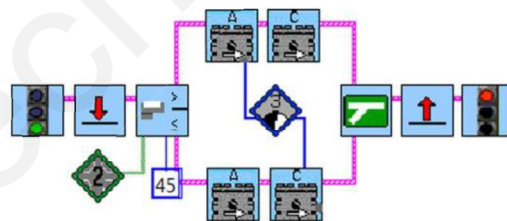
ใช้ในการสร้างเงื่อนไขของการ  
ทำงานโดยเซนเซอร์ต่างๆ  
เช่น กด - ไม่กด  
ค่าแสงมากกว่า - น้อยกว่า  
ระยะทางมากกว่า - น้อยกว่า



ตัวอย่างการเปรียบเทียบค่าและการทำงาน



เมื่อเซนเซอร์ที่พอร์ต 1 ถูกกดหุ่นยนต์จะหมุนซ้าย  
แต่หากถูกปล่อยหุ่นยนต์จะหมุนขวา



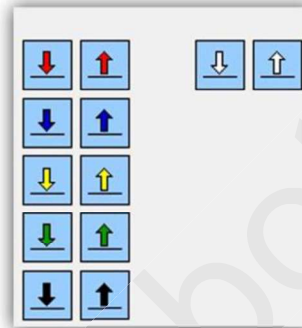
เมื่อเซนเซอร์แสงที่พอร์ต 2 เจอค่ามากกว่า 45(เจอขาว) หุ่นยนต์จะวิ่งเบนซ้าย  
และเมื่อเจอค่าน้อยกว่า 45(เจอดำ) หุ่นยนต์จะวิ่งเบนขวา  
: หุ่นยนต์เดินทางตามเส้นโดยใช้เซนเซอร์เดียว

## Jumps ประมวลผลแบบกระโดด

ประกอบไปด้วย 2 เครื่องหมาย

1. Jump 

2. Land 



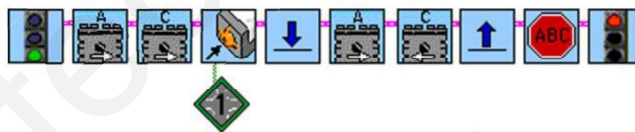
โดยมีหลักการใช้งานคือ ตัว Land จะไม่วิ่งไปหาตัว Jump แต่ตัว Jump จะกระโดดไปหาตัว Land สี่เดียวกันเสมอ

ตัวอย่างการใช้งานและการทำงาน

1. การใช้งาน Jump-Land เพียงหนึ่งคู่

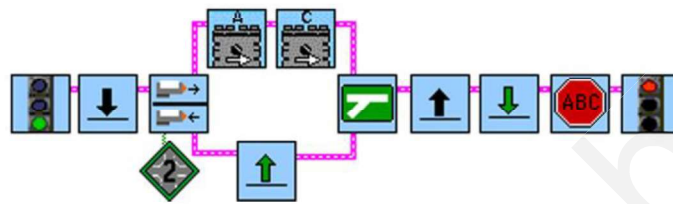


หุ่นยนต์จะเดินหน้าตลอดเวลาไม่มีวันหยุด

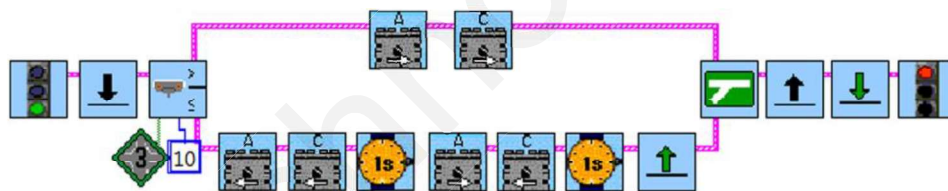


หุ่นยนต์เดินหน้าจนกว่าที่เซนเซอร์ที่พอร์ต 1 จะถูกกด จากนั้นหุ่นยนต์จะเลี้ยวขวาตลอดเวลา

## 2. การใช้งานJump-Land มากกว่า 1 คู่



หุ่นยนต์จะเดินหน้าไปเรื่อยๆ แต่ถ้าหากทัชเซนเซอร์ที่พอร์ต 2 ถูกกดแล้ว หุ่นยนต์จะหยุดทันที



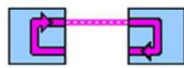
หุ่นยนต์จะเดินหน้าไปเรื่อยๆ จนกว่าอัลตราโซนิกเซนเซอร์ที่พอร์ต 3 เจอค่าน้อยกว่า 10 ซม. จากนั้นหุ่นยนต์จะถอยหลัง 1 วินาที และ เลี้ยวขวากลับตัว 1 วินาที

## Loop ทำงานตามคำสั่งแบบวน

เป็นสัญลักษณ์ที่ใช้กรอบบนชุดคำสั่งที่ต้องการให้ทำซ้ำ

โดยมีสองประเภท

1. วนแบบไม่มีเงื่อนไข (ไปเรื่อย)



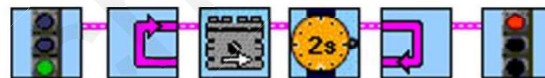
2. วนแบบมีเงื่อนไข (วนจนกว่า)



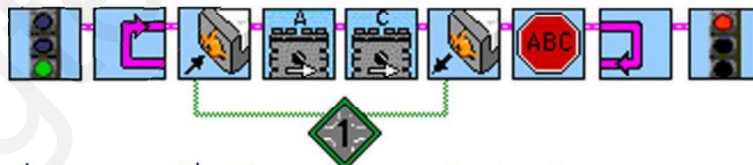
1. ตัวอย่างการใช้ Loop แบบไม่มีเงื่อนไข (คล้ายการใช้ Jmp - Land)



หลอดไฟที่พอร์ต B จะเปิดอยู่ตลอดเวลา



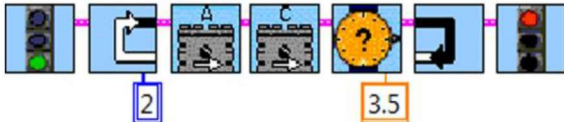
มอเตอร์ที่ต่อกับพอร์ต ABC จะเดินหน้าตลอดเวลา (2วินาทีไปเรื่อย ๆ)



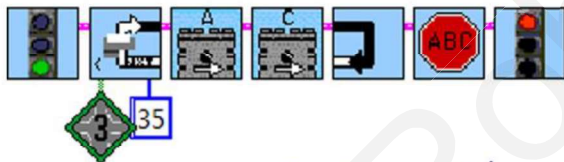
เมื่อทัชเซนเซอร์ที่พอร์ต 1 ถูกกดหุ่นยนต์จะเดินหน้าจนกว่าจะถูกปล่อย

## Loop ทำงานตามคำสั่งแบบวน

## 2. ตัวอย่างการใช้ Loop แบบมีเงื่อนไข



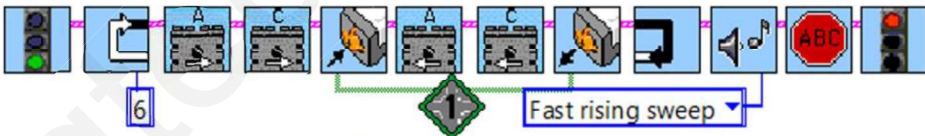
หุ่นยนต์เดินหน้า 7 วินาที



หุ่นยนต์เดินหน้าจกว่าไลท์เวนเซอร์ที่พอร์ต 3  
จะเจค่าแสงน้อยกว่า 35 (เจดดำ)



หุ่นยนต์จะเดินหน้าจนกว่าอัลตราโซนิกที่พอร์ต 1 จะเจอค่าน้อยกว่า 10 เซนติเมตร



หุ่นยนต์จะเดินหน้าไปเรื่อยๆจนกว่าทัชเซนเซอร์ที่พอร์ต 1 จะถูกกดแล้วหุ่นยนต์จะถอยหลังจนกว่าจะถูกปล่อยแล้วเดินหน้าต่อจะใช้เงื่อนไขนี้ได้ต่อกันครั้งแล้วNXTจะส่งเสียงร้องและหยุดโปรแกรม

## Subroutine

เป็นสัญลักษณ์ที่ใช้แทนชุดคำสั่ง ประกอบด้วย 2 ส่วนที่สำคัญ

1. Creat Subroutine

(กำหนดชุดคำสั่ง)



2.Run Subroutine

(เรียกใช้ชุดคำสั่ง)



วิธีใช้ Creat Subroutine



Begin of Subroutine

จุดสำคัญที่ต้องเชื่อมมีดังต่อไปนี้

Number of subroutine

1.Begin of Subroutine

เชื่อมกับชุดคำสั่งที่ต้องการ

2.Number of Subroutine

เชื่อมกับเลขระบุชุดคำสั่ง

วิธีใช้ Run Subroutine

มีจุดเชื่อมเพียงจุดเดียวคือ

Number of Subroutine

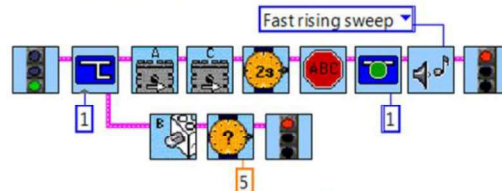
Number of Subroutine

ใช้เชื่อมกับเลขระบุชุดคำสั่งที่ต้องการใช้

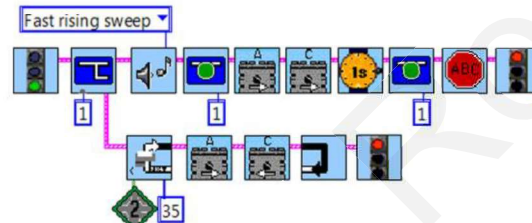


## Subroutine

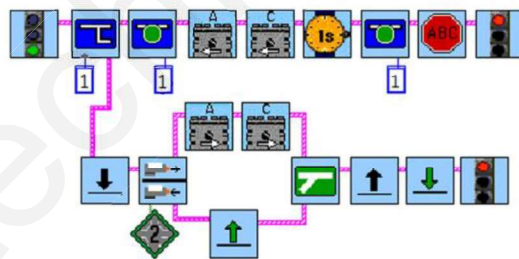
## ตัวอย่างการใช้ Subroutine



หุ่นยนต์จะเดินหน้าไป 2 วินาที จากนั้นจะหยุด และ เปิดหลอดไฟ  
ที่พอร์ต B 5 วินาที จากนั้นส่งเสียงและจบโปรแกรม



หุ่นยนต์จะส่งเสียงแล้วเลี้ยวขวาจนกว่าไลต์เซนเซอร์  
ที่พอร์ต 3 จะเจอค่าน้อยกว่า 35 (เจอดำ) แล้วเดินตรงต่อไป 1 วินาที  
จากนั้นหุ่นยนต์จะเลี้ยวขวาจนกว่าจะเจอค่าอีกครั้งแล้วหยุด



หุ่นยนต์จะเดินตรงไปเรื่อยๆจนกว่าเซนเซอร์ที่พอร์ต 2 สองจะถูกกด  
จากนั้นหุ่นยนต์จะถอยหลัง 1 วินาที และเดินตรงต่อไปเรื่อยๆจนกว่าจะถูกกด  
แต่ครั้งที่ 2 นี้ถูกกดแล้วจะหยุดเดินและจบโปรแกรม

## Modifiers

เป็นสัญลักษณ์ที่ใช้กำกับคำสั่งต่างๆ ซึ่งในชั้นพื้นฐานนี้มีสัญลักษณ์ที่จำเป็นอยู่ 3 ชนิด

### 1. พอร์ตไฟเข้า (INPUT)



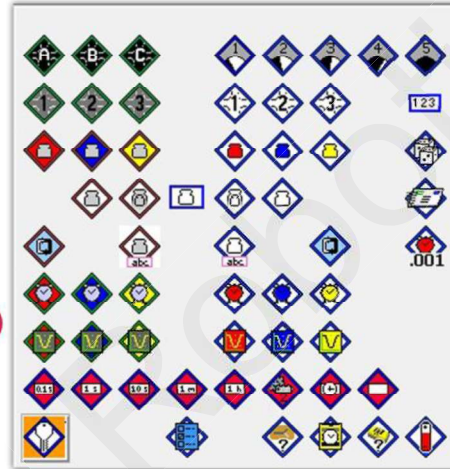
### 2. พอร์ตไฟออก (OUTPUT)



### 3. กำลังไฟฟ้า (POWER MOTER)



### 4. จำนวน (NUMMERIC) 123



ตัวอย่างการใช้ (Modifiers) ในแบบต่าง

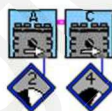


รอกจนกว่าทัชเซนเซอร์ที่พอร์ต 2 จะถูกกดและ พอร์ต 3 จะถูกปล่อย



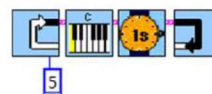
มอเตอร์ A และ C เดินหน้าช่วยมอเตอร์ B หยุด

### 3. POWER MOTER



มอเตอร์ A เดินช้ากว่า C (เดินเอียงซ้าย)

### 4. NUMMERIC

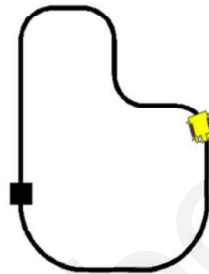


ส่งเสียงคีย์ C 5 ครั้ง

### Line-Follow หรือ Line-Tracking

การเดินทางตามเส้นของหุ่นยนต์มีขั้นตอนดังนี้

1. ประกอบหุ่นยนต์และติดตั้งเซนเซอร์
2. วัดค่าแสง และ คำนวณค่ากลาง
3. วางแผนผังกรณีและเขียนโปรแกรม
4. ทดสอบโปรแกรมและแก้ไขปรับปรุง



#### เริ่มการทำงาน

มีตัวอย่างหุ่นยนต์ในแผ่น VCD ประกอบการสอน

#### 1. ประกอบหุ่นยนต์และติดตั้งเซนเซอร์

การประกอบหุ่นยนต์นั้นเป็นส่วนสำคัญอย่างยิ่งในการทำให้หุ่นยนต์มีประสิทธิภาพสูงสุดในการทำงานรวมทั้งการติดตั้งเซนเซอร์นั้นควรสูงจากพื้นไม่เกิน 1 ซม. จะทำให้ค่าแสงนั้นนิ่งและไม่คลาดเคลื่อนเมื่อแสงเปลี่ยนกะทันหัน



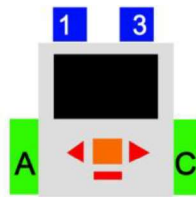
โดยในการเขียนโปรแกรมใน Workshop นี้จะใช้

ไลต์เซนเซอร์ 2 ตัว และ มอเตอร์ 2 ตัว

## Line-Follow หรือ Line-Tracking

### 2. วัดค่าแสง และ คำนวณค่ากลาง

เริ่มแรกให้ติดตั้งไลต์เซนเซอร์ทางซ้าย - ขวาโดยต่อกับพอร์ต 1 และ 3 ตามลำดับ



ทำการวัดค่าแสงโดยการเข้าโหมด VIEW เลือก Reflecte และเลือกพอร์ตตามต้องการจากนั้นบันทึกค่าลงในตารางดังตัวอย่าง



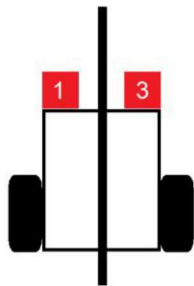
| พอร์ต | ดำ | ขาว | ค่ากลาง |
|-------|----|-----|---------|
| 1     | 20 | 68  | 44      |
| 3     | 20 | 68  | 44      |

ค่ากลางนี้หาโดยมีสูตรดังนี้  $\frac{\text{ดำ} + \text{ขาว}}{2} = \text{ค่ากลาง}$

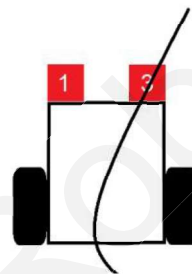
## Line-Follow หรือ Line-Tracking

### 3.วางแผนผังกรณีและเขียนโปรแกรม

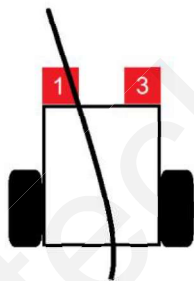
กรณีที่จะเกิดกับเซนเซอร์ทั้ง 2 ตัวในการเดินตามเส้นนั้นมี 4 แบบ



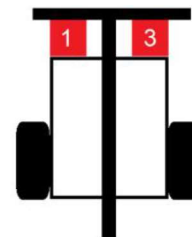
1 และ 3 เจอขาวทั้งคู่  
เดินตรงต่อเพื่อให้ค้อมเส้น



1 เจอขาว 3 เจอดำ  
เลี้ยวขวาเพื่อให้ค้อมเส้น



1 เจอดำ 3 เจอขาว  
เลี้ยวซ้ายเพื่อให้ค้อมเส้น



1 และ 3 เจอดำทั้งคู่  
หยุดหรือเดินตรงต่อตามแต่กรณี

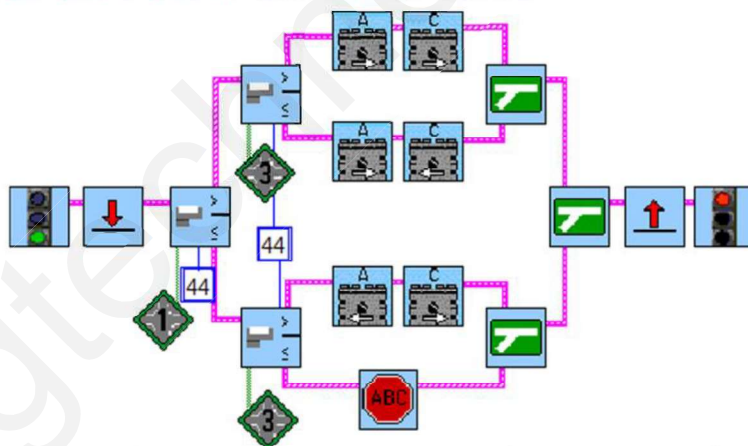
## Line-Follow หรือ Line-Tracking

### 3.วางแผนผังกรณีและเขียนโปรแกรม (ต่อ)

จากการจำลองกรณีทั้ง 4 แล้ว กระบวนการในการทำงานของหุ่นยนต์จะเป็นดังนี้



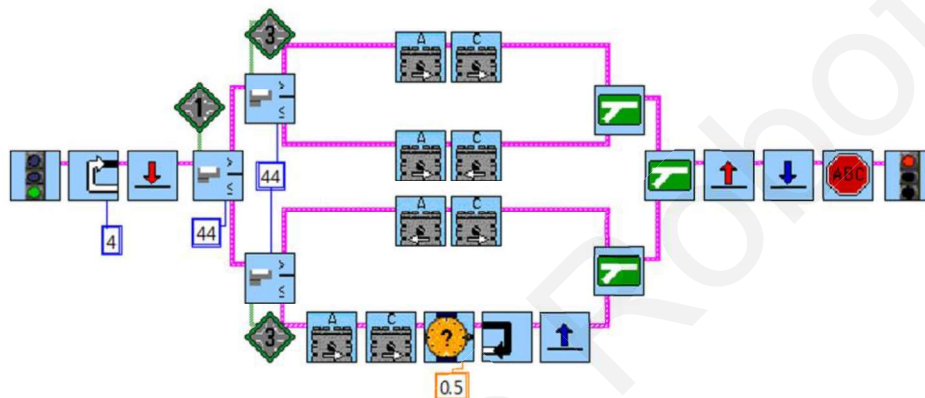
สามารถนำมาเขียนบน ROBOLAB ได้ดังนี้



อย่าลืม : ใส่ค่าแสง และ กำหนดพอร์ตของเซนเซอร์แสง

## Line Follow แบบระบุจำนวนเส้นตัด

การเดินตามเส้นแบบระบุเส้นตัดจะต้องใช้หลักการ 2 ชนิดร่วมกัน คือ Jump - Land และ Loop ดังตัวอย่างต่อไปนี้



เทคนิค : การใช้ Loop นับเส้นดำ (ดำ- ดำ)

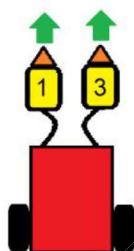
1. เมื่อเซนเซอร์จับได้ ดำ- ดำ แล้วหุ่นยนต์จะต้องเดินหน้าไปให้พ้นเส้นดำเพื่อให้ Loop นับเส้นดำได้ครั้งเดียว
2. เมื่อนับเส้นแล้วโปรแกรมจะเปรียบเทียบค่าที่ระบุจำนวน Loop หากยังไม่ถึงจำนวนโปรแกรมจะกลับไปเริ่มใหม่
3. เมื่อนับเส้น ดำ-ดำ ได้ครบตามจำนวนที่ระบุแล้ว (ในตัวอย่างคือ 4 เส้น) Loop จะคลายและทำการ Jump ออกนอกเงื่อนไขเดิมจากนั้นมอเตอร์จะหยุด และ จบโปรแกรม

## หุ่นยนต์บังคับมือโดย ทักษะเซนเซอร์ 2 ตัว

ในการทำหุ่นยนต์บังคับมือนี้จะใช้ทักษะเซนเซอร์ 2 ตัว ในที่นี้จะใช้พอร์ต 1 และ พอร์ต 3 ในการบังคับซ้ายขวา

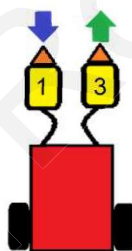
### เริ่มต้นกับการเขียนโปรแกรมหุ่นยนต์บังคับมือ

1. ติดตั้งและกำหนดพอร์ตดังนี้ พอร์ต 1 คือด้านซ้าย พอร์ต 3 คือด้านขวา
2. วางผังการทำงานของโปรแกรมในกรณีที่จะเกิดขึ้นกับเซนเซอร์ทั้งหมด



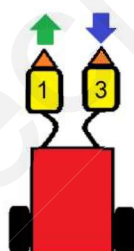
เซนเซอร์ 1 และ 3 ถูกปล่อย

หุ่นยนต์หยุดอยู่กับที่



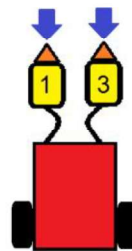
เซนเซอร์ 1 ถูกกด 3 ถูกปล่อย

หุ่นยนต์เลี้ยวซ้าย



เซนเซอร์ 1 ถูกปล่อย 3 ถูกกด

หุ่นยนต์เลี้ยวขวา



เซนเซอร์ 1 และ 3 ถูกกด

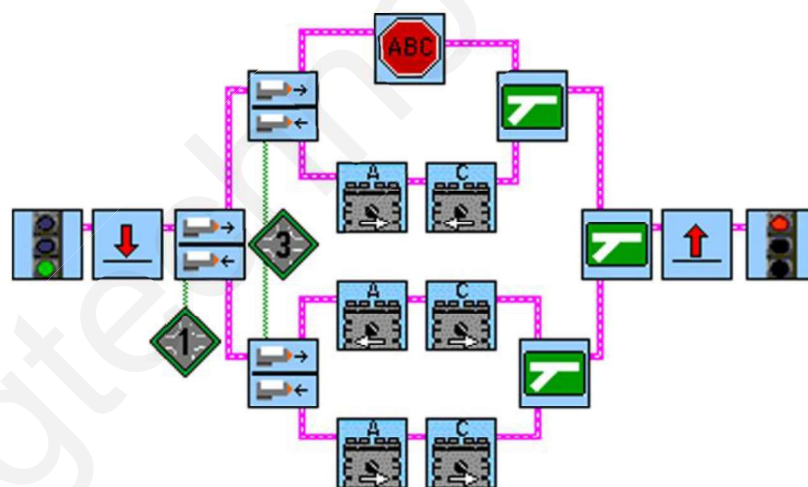
หุ่นยนต์เดินหน้า

## หุ่นยนต์บังคับมือโดย ทักษะเซนเซอร์ 2 ตัว

### 3.วางผังโปรแกรมและเขียนโปรแกรมลง ROBOLAB



สามารถนำมาเขียนบน ROBOLAB ได้ดังนี้



ลักษณะจะคล้ายกันกับโปรแกรมเดินตามเส้น