บทที่ 1 – อะไรคือ**หุ่นยนต์?**

Project Lab Robotics Education

หุ่นยนต์คือ

วิศวกรรม เครื่องกล

Mechanical Engineering

วิศวกรรม ไฟฟ้า

Electrical Engineering

วิศวกรรม โปรแกรม คอมพิวเตอร์

Software Engineering

หุ่นยนต์คือ

สิ่งแวดล้อม

รับรู้และวัดการเปลี่ยนแปลงที่เกิดขึ้นในสิ่งแวดล้อม (โลกภายนอกหุ่นยนต์)

สร้างการเปลี่ยนแปลงในสิ่งแวดล้อม (โลกภายนอกหุ่นยนต์)

ROBOT

การรับรู้ Sensing → กระประมวลผล Computation → การกระทำ Actuation

ตัวอย่างหุ่นยนต์

เครื่องจักรในโรงงาน

หุ่นยนต์ดูดฝุ่น

หุ่นยนต์ผ่าตัด

หุ่นยนต์สองขา

แขนหุ่นยนต์ในโรงงาน

หุ่นยนต์ทรงตัว

หุ่นยนต์รถขับเอง

หุ่นยนต์เครื่องบิน

หน้าที่ของหุ่นยนต์ และงานในอนาคต

- เครื่องจักรทำงานได้ 24 ชั่วโมง
- เครื่องจักรนั้นสม่ำเสมอ แน่นอน ไม่บ่น ไม่อู้งาน ไม่ขี้เกียจ
- งานในอนาคตคืออะไร คืองานสร้างสรรค์ และใช้งาน เทคโนโลยี

พนักงานโรงงาน

พนักงานซุปเปอร์มาร์เกต

พนักงานสนามบิน

พนักงาน call center

พนักงานส่งของ

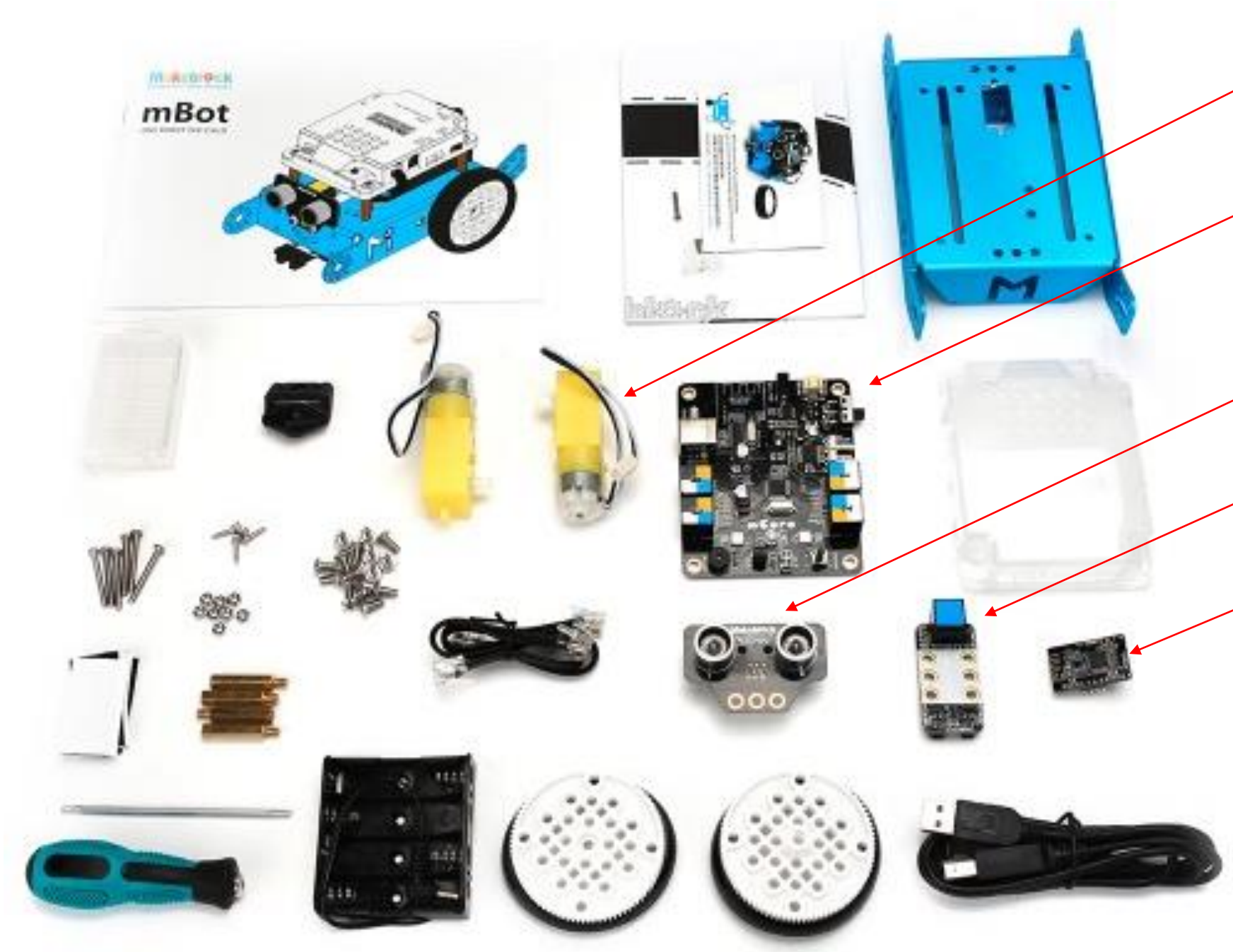
พนักงานขับรถ

วีดีโอหุ่นยนต์ – Atlas from Boston Dynamics

วีดีโอหุ่นยนต์ – Handle from Boston Dynamics

mBot

Components



บทที่ 1 – อะไรคือการเขียน **โปรแกรม?**

Project Lab Robotics Education

อะไรคือ Computer Program?

อะไรคือ Computer Program?

ผมทำงานได้เร็วมากๆ
แต่ช่วยผมที
ผมคิดเองไม่เป็น
บอกผมด้วยว่า จะให้ทำอะไรบ้าง

สั่งงาน คอมพิวเตอร์

สอน คอมพิวเตอร์

ปัญหาคือคอมพิวเตอร์ฉลาดไม่พอที่จะเข้าใจภาษามนุษย์

การสั่งงานคอมพิวเตอร์

Sandbox.py - C:/python/handsOnPython/myFiles/Sandbox.py

File Edit Format Run Options Windows Help

```
def happyBirthday(name):
    song = '''
    Happy Birthday To You!
    Happy Birthday To You!
    Happy Birthday To You!
    Happy Birthday Dear %s
    Happy Birthday To You!
    '''
    print song % name;

def division(dividend, divisor):
    print dividend, "divided by", divisor, "equals", dividend/divisor
    print "With a remainder of", dividend % divisor

def main():
    '''
    name = raw_input("Please enter your name: ");
    happyBirthday(name);
    '''
    division(1,1)
    division(10,5)
    while (1):
        usr_choice = raw_input("Would you like to perform a simple division? (Y/N)")
        if (usr_choice == "Y" or usr_choice == "y"):
            dividend = input("Please input the dividend:")
            divisor = input("Please input the divisor:")
            division(dividend, divisor)
            endProcess = raw_input("Would you like to repeat? (Y/N)")
            if (endProcess == "N" or endProcess == "n"):
                break;
        elif (usr_choice == "N" or usr_choice == "n"):
            print "Thank you!"
            break;
        else:
            print "Please input either Y or N."

main()
```

Ln: 24 Col: 52

- โปรแกรมคือ วิธีคุยกับ Computer
- Computer จะทำงานตามคำสั่งในโปรแกรม

- ใช้ภาษาที่ **ชัดเจน**
- ทำงาน **ทีละคำสั่ง**
- ทำงาน **ทีละบรรทัด**

ภาษา Scratch

- เช่นเดียวกับภาษามนุษย์ คอมพิวเตอร์ก็มีหลายภาษา
- แต่ละภาษามีแนวคิดเหมือนๆกัน
- เราจะใช้ภาษา **Scratch**
- **Scratch** เป็นภาษาที่ **ไม่ต้องพิมพ์**
- **Scratch** เป็นภาษาที่ **ใช้การลากแล้ววางบล็อกคำสั่ง (Drag & Drop)**

mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Disconnected - Not saved

File Edit Connect Boards Extensions Language Help

Boards
- Arduino
- Arduino Uno
- Arduino Leonardo
- Arduino Nano (mega328)
- Arduino Mega 1280
- Arduino Mega 2560
- Makeblock
- Starter/Ultimate (Orion)
- Me Uno Shield
- mBot (mCore)
- mBot Ranger (Auriga)
- Ultimate 2.0 (Mega Pi)
- Others
- PicoBoard

Scripts

Motion
Looks
Sound
Pen
Data&Blocks

mBot

mBot Program
run forward at
set motor M1
set servo Port1
set led on board all red 0 green
set led Port1 all red 0 green 0
set led strip Port1 Slot2 all red
play tone on note C4 beat Half
show face Port1 number: 0
show face Port1 x: 0 y: 0 characters
show time Port1 hour: 10 : min:
show drawing Port1 x: 0 y: 0 draw
set 7-segments display Port1 number
set light sensor Port3 led as On
set camera shutter Port1 as Press
set mini fan Port1 blow clockwise
light sensor light sensor on board

x: -23
y: 3

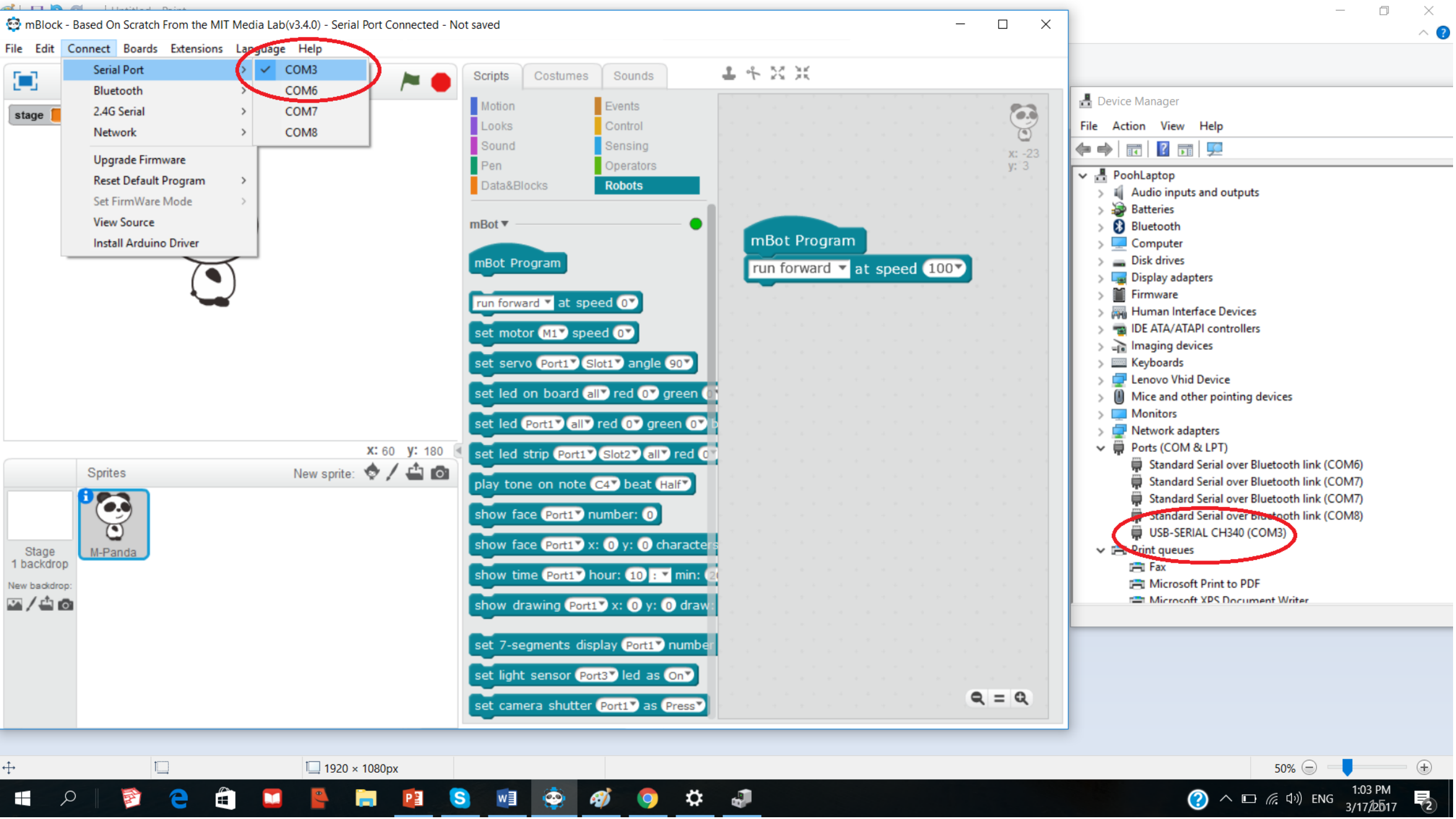
mBot Program
run forward at speed 100

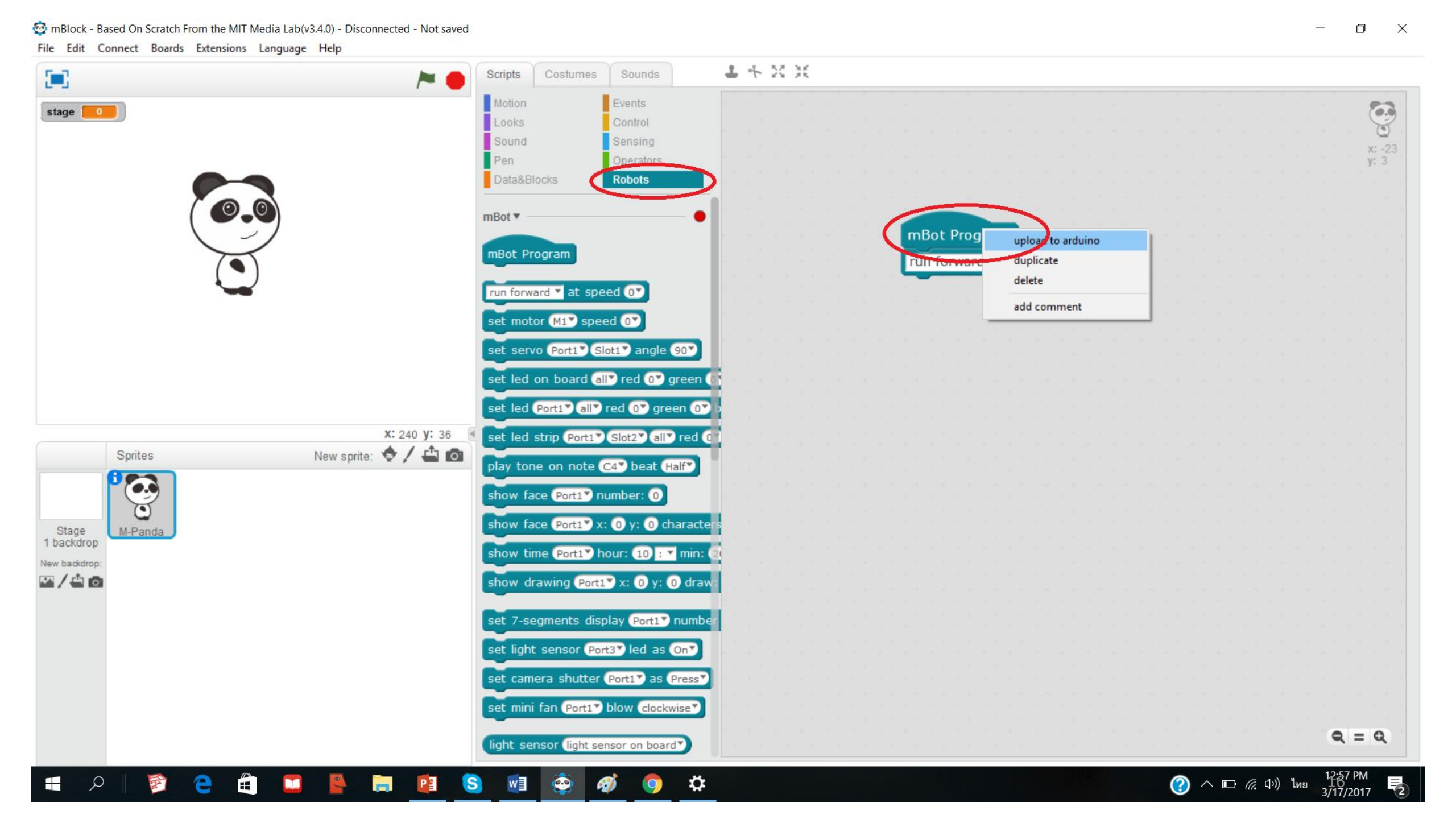
Back | Upload to Arduino | Edit with Arduino IDE

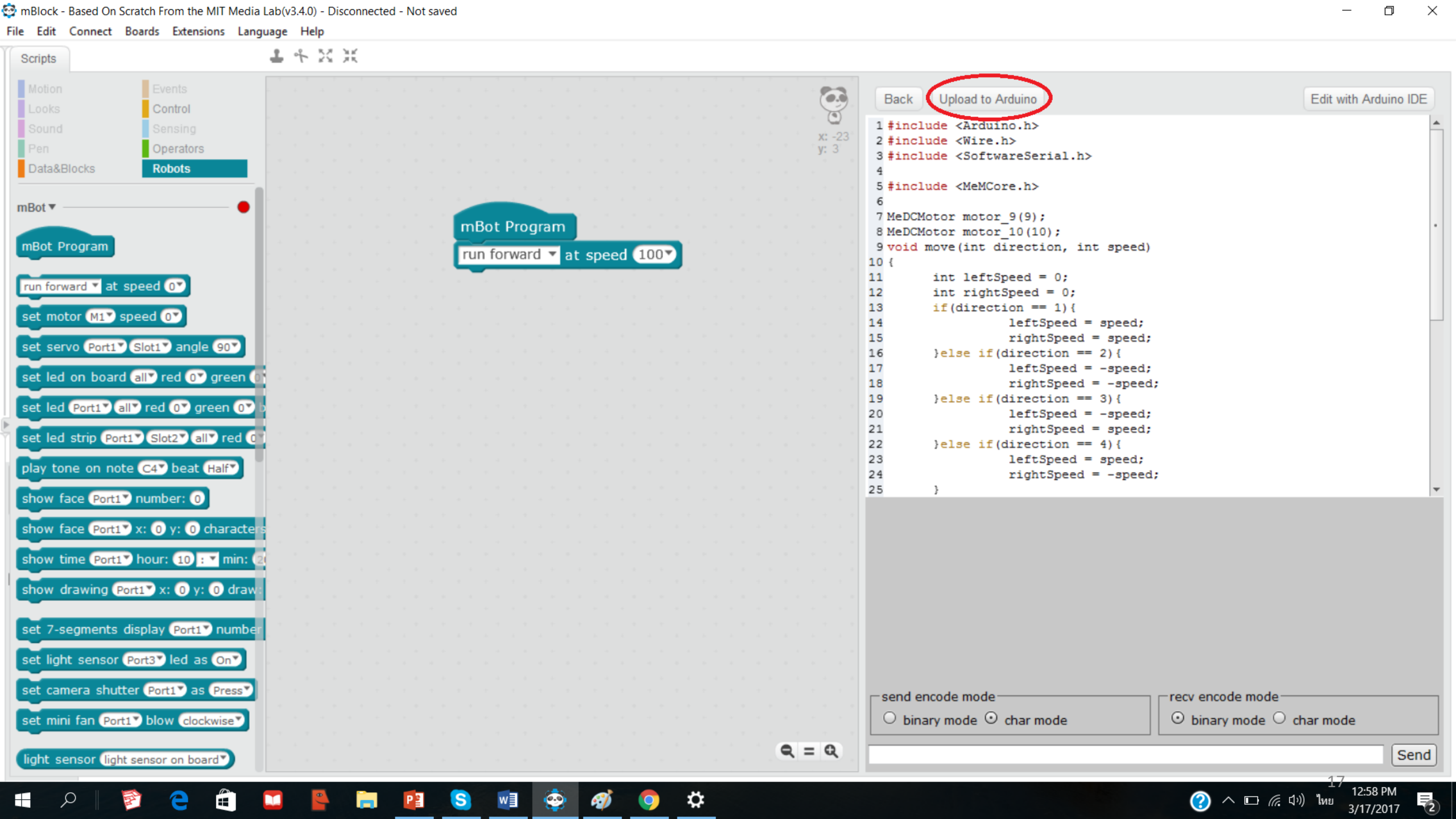
```
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 #include <MeMCore.h>
6
7 MeDCMotor motor_9(9);
8 MeDCMotor motor_10(10);
9 void move(int direction, int speed)
10 {
11       int leftSpeed = 0;
12       int rightSpeed = 0;
13       if(direction == 1){
14                 leftSpeed = speed;
15                 rightSpeed = speed;
16       }else if(direction == 2){
17                 leftSpeed = -speed;
18                 rightSpeed = -speed;
19       }else if(direction == 3){
20                 leftSpeed = -speed;
21                 rightSpeed = speed;
22       }else if(direction == 4){
23                 leftSpeed = speed;
24                 rightSpeed = -speed;
25       }
```

send encode mode: binary mode / char mode
recv encode mode: binary mode / char mode
Send

12:59 PM
3/17/2017


mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Serial Port Connected - Not saved
File Edit Connect Boards Extensions Language Help
Serial Port
Bluetooth
2.4G Serial
Network
Upgrade Firmware
Reset Default Program
Set FirmWare Mode
View Source
Install Arduino Driver
COM3
COM6
COM7
COM8
stage
Scripts Costumes Sounds
Motion Looks Sound Pen Data&Blocks
Events Control Sensing Operators Robots
mBot
mBot Program
run forward at speed 0
set motor M1 speed 0
set servo Port1 Slot1 angle 90
set led on board all red 0 green 0
set led Port1 all red 0 green 0
set led strip Port1 Slot2 all red 0
play tone on note C4 beat Half
show face Port1 number: 0
show face Port1 x: 0 y: 0 characters
show time Port1 hour: 10 : min:
show drawing Port1 x: 0 y: 0 draw:
set 7-segments display Port1 number
set light sensor Port3 led as On
set camera shutter Port1 as Press
mBot Program
run forward at speed 100
x: -23
y: 3
x: 60 y: 180
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
M-Panda
Device Manager
File Action View Help
PoohLaptop
Audio inputs and outputs
Batteries
Bluetooth
Computer
Disk drives
Display adapters
Firmware
Human Interface Devices
IDE ATA/ATAPI controllers
Imaging devices
Keyboards
Lenovo Vhid Device
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Standard Serial over Bluetooth link (COM6)
Standard Serial over Bluetooth link (COM7)
Standard Serial over Bluetooth link (COM7)
Standard Serial over Bluetooth link (COM8)
USB-SERIAL CH340 (COM3)
Print queues
Fax
Microsoft Print to PDF
Microsoft XPS Document Writer
1920 × 1080px
50%
ENG
1:03 PM
3/17/2017


mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Disconnected - Not saved
File Edit Connect Boards Extensions Language Help
stage 0
x: 240 y: 36
Sprites
New sprite:
Stage
1 backdrop
New backdrop:
M-Panda
Scripts
Costumes
Sounds
Motion
Looks
Sound
Pen
Data&Blocks
Events
Control
Sensing
Operators
Robots
mBot
mBot Program
run forward at speed 0
set motor M1 speed 0
set servo Port1 Slot1 angle 90
set led on board all red 0 green
set led Port1 all red 0 green 0
set led strip Port1 Slot2 all red
play tone on note C4 beat Half
show face Port1 number: 0
show face Port1 x: 0 y: 0 characters
show time Port1 hour: 10 : min:
show drawing Port1 x: 0 y: 0 draw
set 7-segments display Port1 number
set light sensor Port3 led as On
set camera shutter Port1 as Press
set mini fan Port1 blow clockwise
light sensor light sensor on board
mBot Prog
upload to arduino
duplicate
delete
add comment
x: -23
y: 3
12:57 PM
3/17/2017


mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Disconnected - Not saved
File Edit Connect Boards Extensions Language Help
Scripts
Motion Looks Sound Pen Data&Blocks
Events Control Sensing Operators Robots
mBot
mBot Program
run forward at speed 0
set motor M1 speed 0
set servo Port1 Slot1 angle 90
set led on board all red 0 green 0
set led Port1 all red 0 green 0
set led strip Port1 Slot2 all red 0
play tone on note C4 beat Half
show face Port1 number: 0
show face Port1 x: 0 y: 0 characters
show time Port1 hour: 10 : min:
show drawing Port1 x: 0 y: 0 draw:
set 7-segments display Port1 number
set light sensor Port3 led as On
set camera shutter Port1 as Press
set mini fan Port1 blow clockwise
light sensor light sensor on board
mBot Program
run forward at speed 100
x: -23
y: 3
Back
Upload to Arduino
Edit with Arduino IDE
1 #include <Arduino.h>
2 #include <Wire.h>
3 #include <SoftwareSerial.h>
4
5 #include <MeMCore.h>
6
7 MeDCMotor motor_9(9);
8 MeDCMotor motor_10(10);
9 void move(int direction, int speed)
10 {
11 int leftSpeed = 0;
12 int rightSpeed = 0;
13 if(direction == 1){
14 leftSpeed = speed;
15 rightSpeed = speed;
16 }else if(direction == 2){
17 leftSpeed = -speed;
18 rightSpeed = -speed;
19 }else if(direction == 3){
20 leftSpeed = -speed;
21 rightSpeed = speed;
22 }else if(direction == 4){
23 leftSpeed = speed;
24 rightSpeed = -speed;
25 }
send encode mode
binary mode char mode
recv encode mode
binary mode char mode
Send
12:58 PM
3/17/2017

mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Disconnected - Not saved

File Edit Connect Boards Extensions Language Help

stage 0

x: 240 y: -180

Sprites New sprite:

Stage
1 backdrop

New backdrop:

M-Panda

Scripts Costumes Sounds

Motion Events
Looks Control
Sound Sensing
Pen Operators
Data&Blocks Robots

mBot

mBot Program

run forward at speed 0

set motor M1 speed 0

set servo Port1 Slot1 angle 90

set led on board all red 0 green

set led Port1 all red 0 green 0

set led strip Port1 Slot2 all red

play tone on note C4 beat Half

show face Port1 number: 0

show face Port1 x: 0 y: 0 characters

show time Port1 hour: 10 : min:

show drawing Port1 x: 0 y: 0 draw

set 7-segments display Port1 number

set light sensor Port3 led as On

set camera shutter Port1 as Press

set mini fan Port1 blow clockwise

light sensor light sensor on board

x: -23
y: 3

mBot Program
set motor M1 speed 100
set motor M2 speed 100

ENG 1:30 PM 3/17/2017

mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Disconnected - Not saved

File Edit Connect Boards Extensions Language Help

stage 0

x: 240 y: -180

Sprites New sprite:

Stage
1 backdrop

New backdrop:

M-Panda

Scripts Costumes Sounds

Motion Events
Looks Control
Sound Sensing
Pen Operators
Data&Blocks Robots

wait 1 secs

repeat 10

forever

if then

if then
else

wait until

repeat until

stop all

when I start as a clone

create clone of myself

delete this clone

x: -23
y: 3

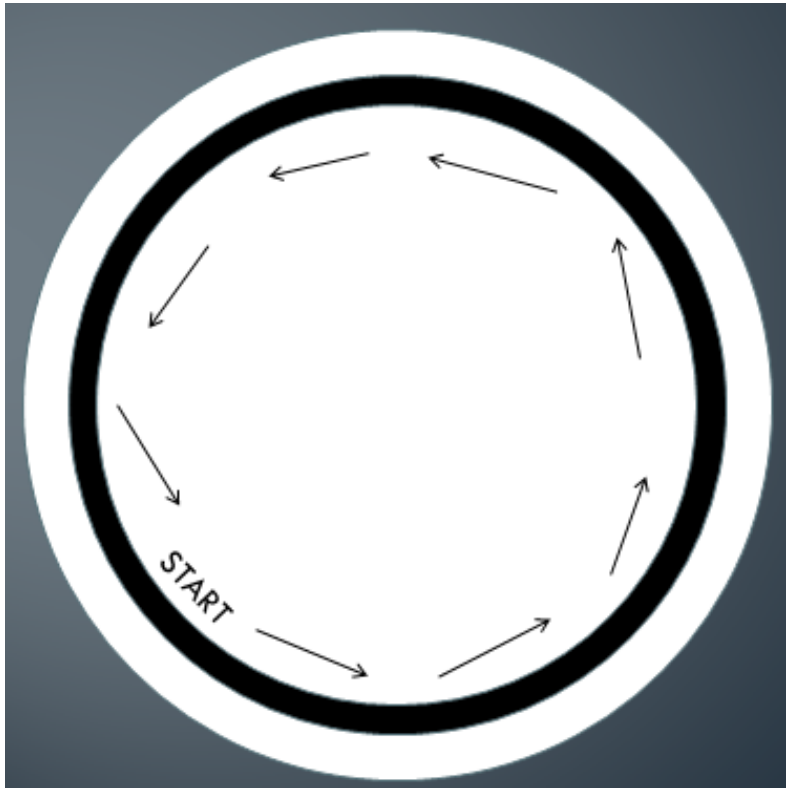
mBot Program
run forward at speed 100
wait 1 secs
turn right at speed 100
wait 1 secs
run backward at speed 100
wait 1 secs
run backward at speed 0

ENG 1:21 PM 3/17/2017

สั่งงานให้หุ่นยนต์เคลื่อนที่

- โปรแกรมหุ่นยนต์ mBot ให้
  1. เดินหน้า
  2. ถอยหลัง
  3. เลี้ยวแบบหุ่นยนต์อยู่กับที่
  4. เลี้ยวแบบหมุนหนึ่งล้อ (อีกล้อหยุด)
  5. เลี้ยวแบบวงกลมใหญ่
  6. หยุด

เดินเป็นวงกลมใหญ่



บทที่ 2 – หุ่นยนต์รถบังคับ

Project Lab Robotics Education

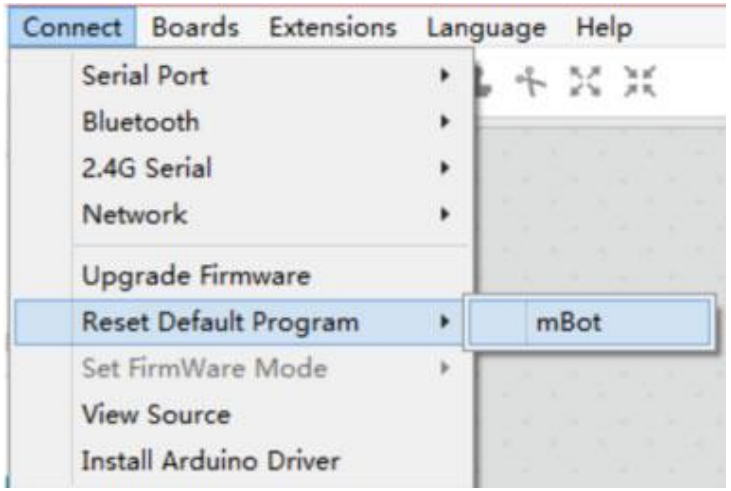
เข้าใจการทำงาน mode ‘Scratch’

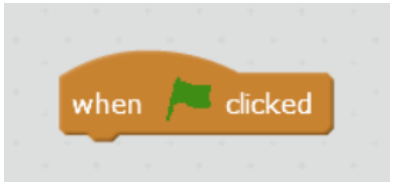
- ปกติแล้ว mode ‘Scratch’ นั้นมีไว้สำหรับการโปรแกรมบนคอมพิวเตอร์ โดยที่โปรแกรมจะทำงานผ่านหน้าจอที่มีหมีแพนดา
- เมื่อเขียนโปรแกรมใน mode ‘Scratch’ เราสามารถดูและควบคุมการทำงาน mBot ผ่านคอมพิวเตอร์ได้
- สามารถเชื่อมโยงการทำงานของหุ่นยนต์ผ่าน input ของ computer เช่น mouse และ keyboard
- สามารถใช้เซนเซอร์ของหุ่นยนต์ ควบคุมโปรแกรมใน ‘Scratch’ ได้

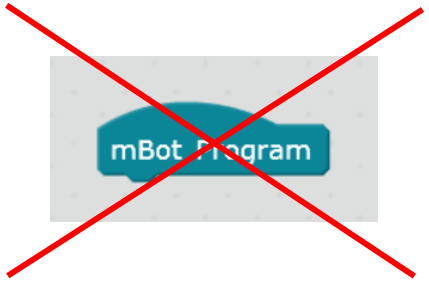
- ในคอร์สนี้เราจะไม่ได้โปรแกรมเกมส์ใน mode ‘Scratch’ เราจะเน้นที่หุ่นยนต์ ‘mBot’

Reset Default Program

- เพื่อให้หุ่นยนต์ส่งข้อมูลให้โปรแกรม **mBlock** ผ่านสาย **USB** เราต้อง **Reset Default Program**



- ทีนี้เราจะโปรแกรมหุ่นยนต์ร่วมกับหมีแพนดาได้
- เพื่อเริ่มโปรแกรมใช้คำสั่ง 

แทนที่ 

mBlock - Based On Scratch From the MIT Media Lab(v3.4.0) - Disconnected - Not saved

File Edit Connect Boards Extensions Language Help

Serial Port
Bluetooth
2.4G Serial
Network
Upgrade Firmware
Reset Default Program
Set FirmWare Mode
View Source
Install Arduino Driver
Connect

Scripts Costumes Sounds

Motion Looks Sound Pen Data&Blocks
Events Control Sensing Operators Robots

when clicked
when space key pressed
when space key released
when this sprite clicked
when backdrop switches to backdrop1
when loudness > 10
when I receive message1
broadcast message1
broadcast message1 and wait

when clicked
set motor M1 speed 100
set motor M2 speed 100

x: -23
y: 3

x: -79 y: 180

Sprites
New sprite:
Stage
1 backdrop
New backdrop:
M-Panda

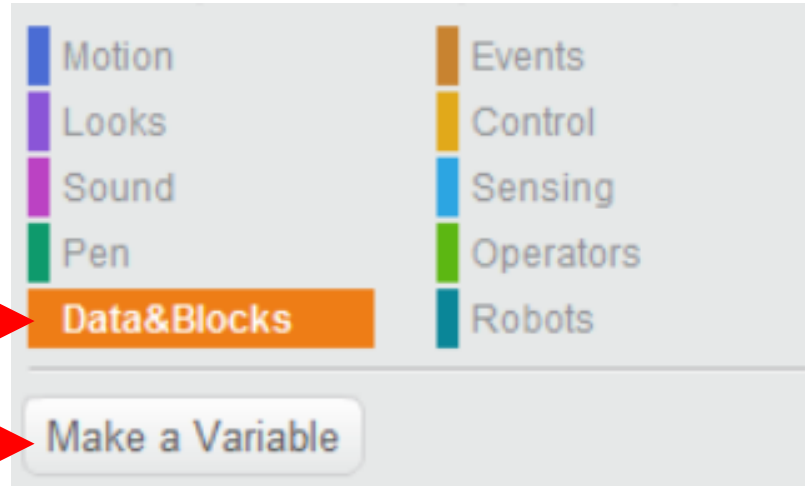
2:14 PM
3/17/2017

Variable (ตัวแปร)

- **ตัวแปร** ก็คือการ **แทนค่าอะไรซักอย่างด้วยตัวอักษร**
- เช่นในสมการ x + 2 = 7 เรามีหนึ่งตัวแปร, ในสมการนี้ ตัวแปร x เท่ากับ _____
- ในสมการ x + y + z = (a x b) + c, เรามีกี่ตัวแปร? _____
- ในการเขียนโปรแกรมเราจะใช้ตัวแปรบ่อยมาก
- สร้างตัวแปร

บล็อกเกี่ยวกับ ตัวแปร

สร้างตัวแปรใหม่



Variable (ตัวแปร)

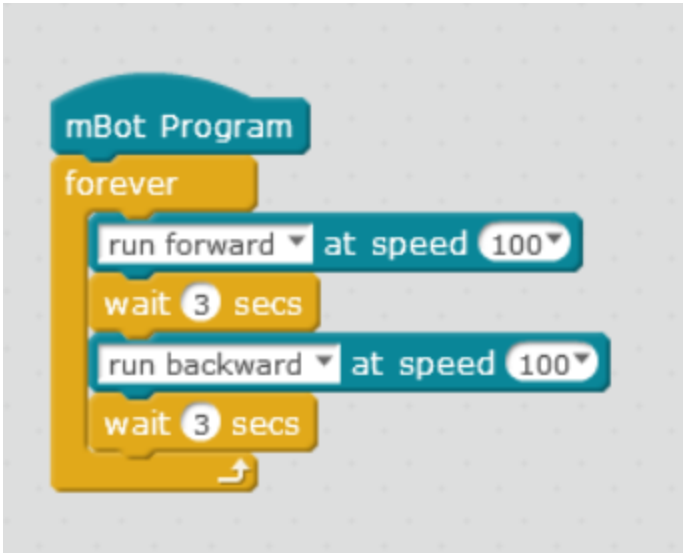
- ในการเขียนโปรแกรม การใช้ **ตัวแปร** มีประโยชน์อะไรบ้าง?
  - ใช้เก็บสถานะของหุ่นยนต์
  - ใช้ช่วยในการตัดสินใจ
  - ใช้ทำการคำนวณสูตรคณิตศาสตร์
  - ช่วยรวบรัดในการแก้ไขโปรแกรม แก้จุดๆเดียวแทนที่การแก้หลายจุด
  - ช่วยแบ่งแยกสัดส่วนของโปรแกรม ที่จะเชื่อมโยงกันผ่านตัวแปรที่ส่งต่อ

Loop Forever

- **Loop Forever** จะทำให้โปรแกรมทำงานแบบวนไปเรื่อยๆ ไม่มีวันสิ้นสุด

<u>การทำงานของโปรแกรม</u>

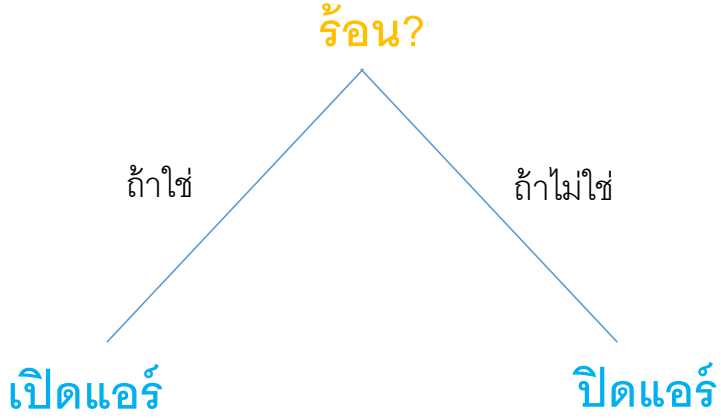
1. เริ่มโปรแกรม
2. เดินหน้า
3. รอ 3 สามวินาที
4. ถอยหลัง
5. รอ 3 วินาที
6. เดินหน้า
7. รอ 3 สามวินาที
8. ถอยหลัง
9. รอ 3 วินาที
10. เดินหน้า
11. รอ 3 สามวินาที
12. .....



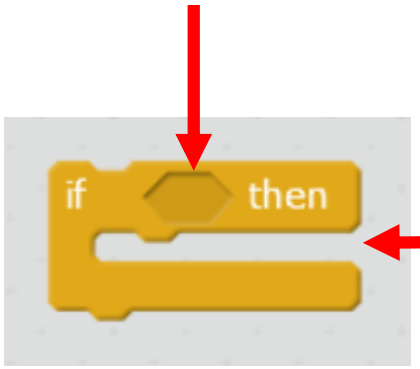
If Statement

- หุ่นยนต์สามารถตัดสินใจโดยใช้คำสั่ง If Statement
- If แปลว่า **ถ้าเกิดจริง**
- If / Else แปลว่า **ถ้าเกิดจริง / ถ้าเกิดไม่จริง (เท็จ)**

การตัดสินใจด้วย if statement

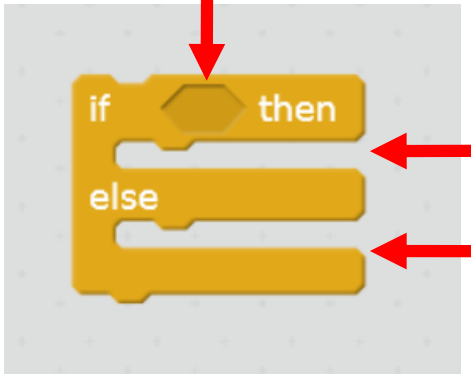


กรณีใช้ตัดสินใจ



ทำในนี้ถ้าเกิด **กรณีนั้นเป็นจริง**

กรณีใช้ตัดสินใจ

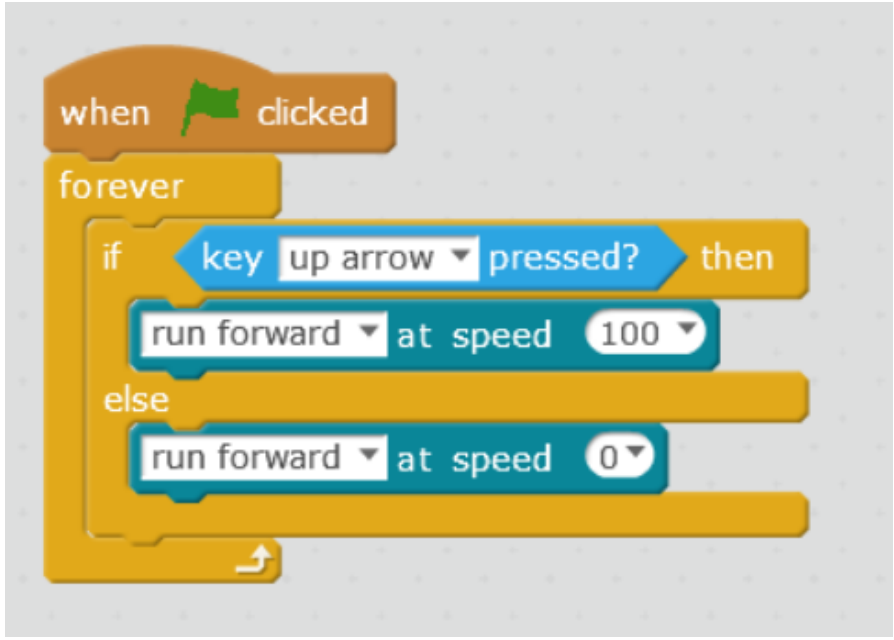


ทำในนี้ถ้าเกิด **กรณีนั้นเป็นจริง**

ทำในนี้ถ้าเกิด **กรณีนั้นไม่จริง**

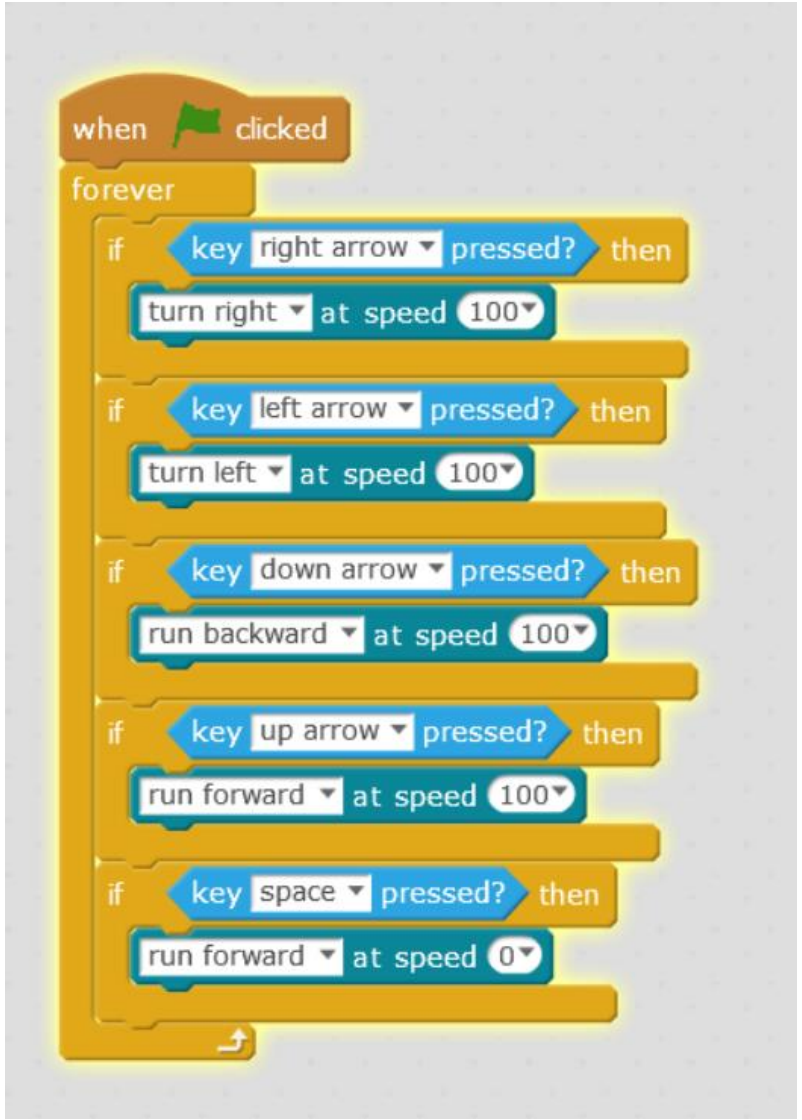
หุ่นยนต์รถบังคับ กดปุ่มเดียว

- ใช้ keyboard ทำหน้าที่เป็น remote control
- ถ้ากดแต่ละปุ่มจะเกิดแต่ละเหตุการณ์
- ลองทำตัวอย่างนี้ดู



หุ่นยนต์รถบังคับ กดหลายปุ่ม

- ใช้ keyboard ทำหน้าที่เป็น remote control
- ถ้ากดแต่ละปุ่มจะเกิดแต่ละเหตุการณ์
- ลองทำตัวอย่างนี้ดู



หุ่นยนต์รถบังคับ

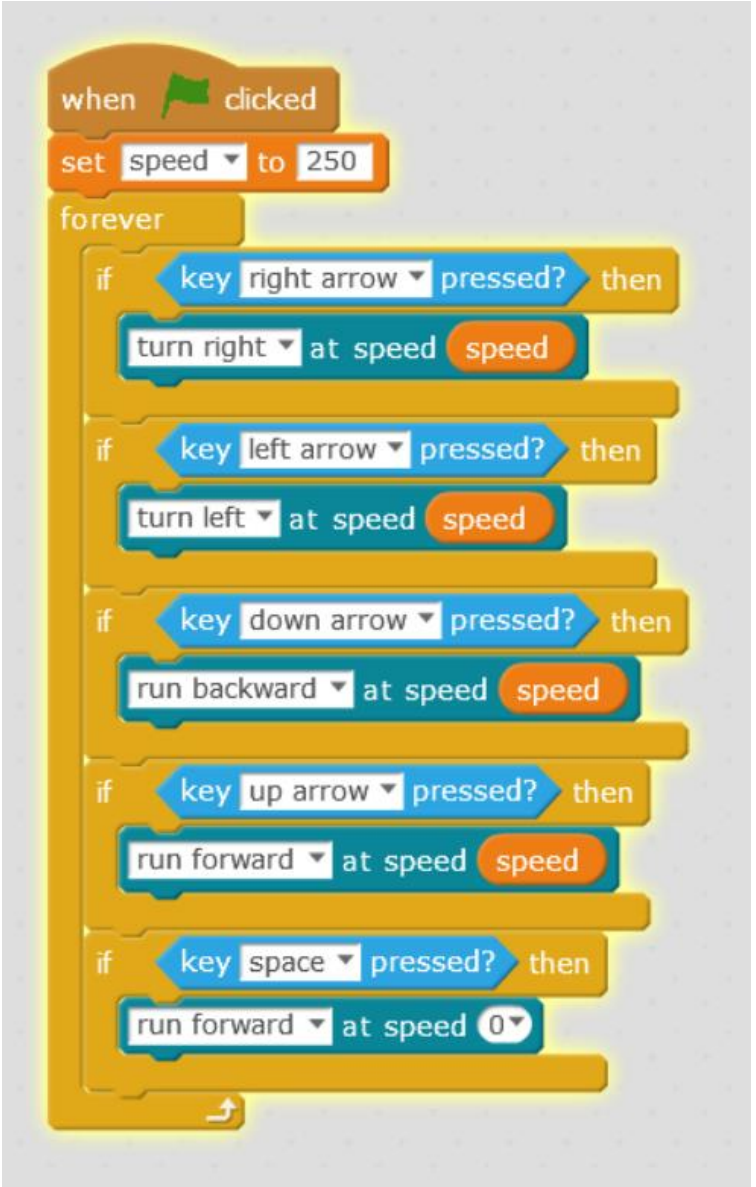
- ถ้าเราจะเปลี่ยนความเร็วของหุ่นยนต์ที่จุดเดียวทำได้โดยการใช้ตัวแปร
- ทำให้หุ่นยนต์หยุดได้โดยการใช้อีกปุ่ม

ตัวแปร ขีดเส้นใต้

Loop

Start
Set speed to 250
Forever Loop

If key right then, turn right at speed speed
If key left then, turn left at speed speed
If key up then, run forward at speed speed
If key down then, run backward at speed speed
If key space then, run forward at speed 0



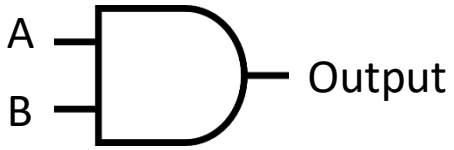
Logic Operation (ลอจิก ออปเปอเรชั่น)

ลอจิก Operation มีสามแบบหลักๆ AND, OR, NOT

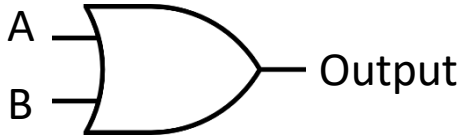
Input หมายความว่า **สัญญาณเข้า**

Output หมายความว่า **สัญญาณออก (ผลลัพธ์)**

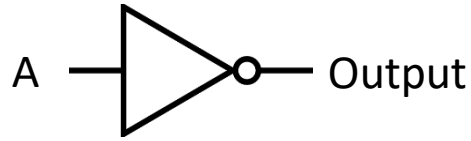
ภาษาคอมพิวเตอร์ **True = จริง = 1** **False = เท็จ = 0**



AND Operation (และ)		
A	***B***	***Output***
0	0	0
1	0	0
0	1	0
1	1	1



OR Operation (หรือ)		
A	***B***	***Output***
0	0	0
1	0	1
0	1	1
1	1	1

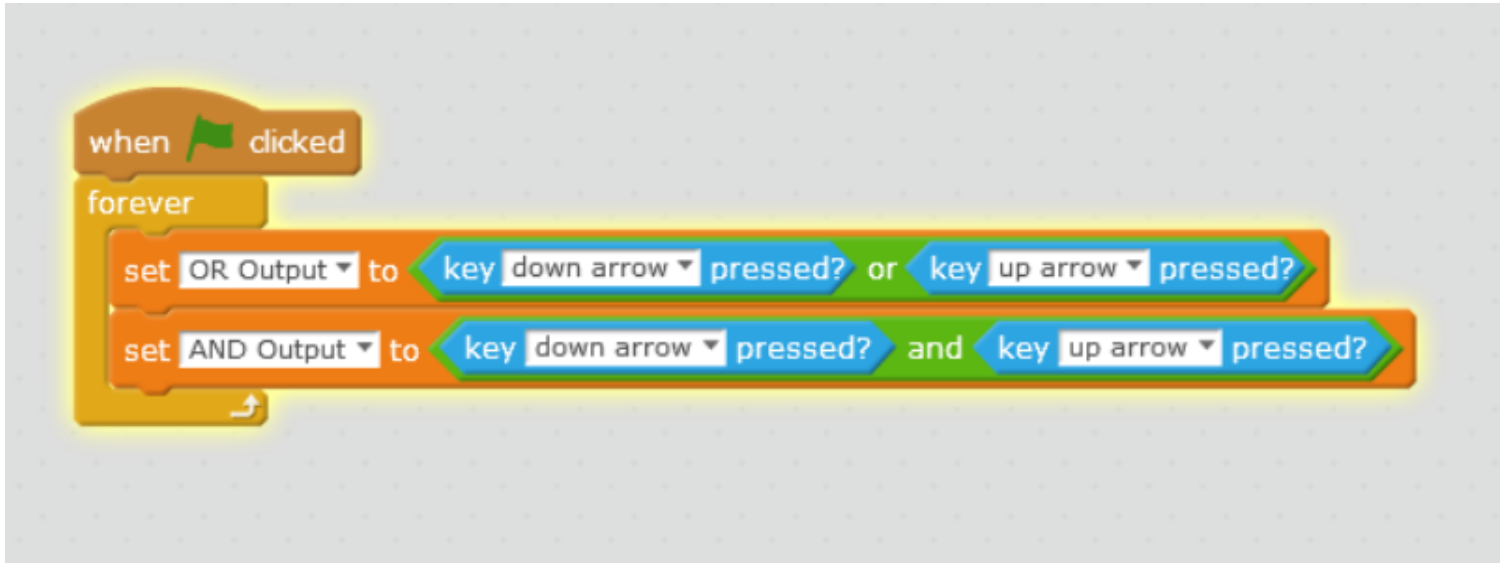


NOT Operation (ไม่)	
A	***Output***
0	1
1	0

Logic Operation (ลอจิก ออปเปอเรชั่น)

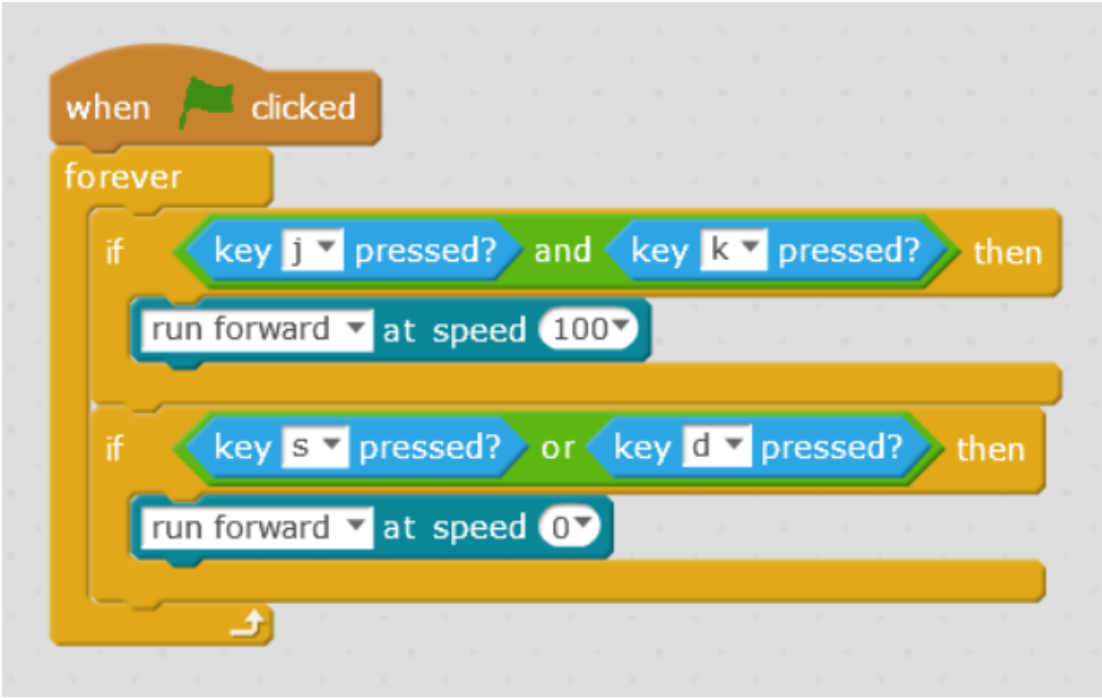
ลองมาดูการใช้งานของลอจิกกัน

- OR: เมื่อกด up arrow **หรือ** down arrow จะได้ผลลัพธ์เป็น TRUE
- AND: เมื่อกดทั้ง up arrow **และ** down arrow จึงจะได้ผลลัพธ์เป็น TRUE



Logic Operation (ลอจิก ออปเปอเรชั่น)

- เมื่อกด **j** และ **k** พร้อมกัน หุ่นยนต์จะ เดินไปข้างหน้า
- ถ้ากด **s** หรือ **d** หุ่นยนต์จะ หยุด



หุ่นยนต์บังคับ

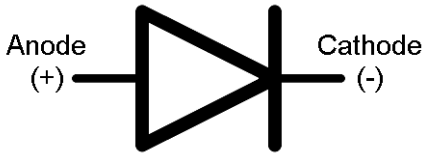
- ทำอย่างไรให้หุ่นยนต์หยุดเองได้โดยไม่ต้องกดเพิ่มอีกปุ่ม ตัวอย่างเช่นเมื่อไม่กดปุ่มอะไรหุ่นยนต์ก็จะหยุดเอง ให้เขียนโค้ดด้านล่างนี้ (ใบ้ – ให้ใช้บล็อก If, Else, Logic Operation)

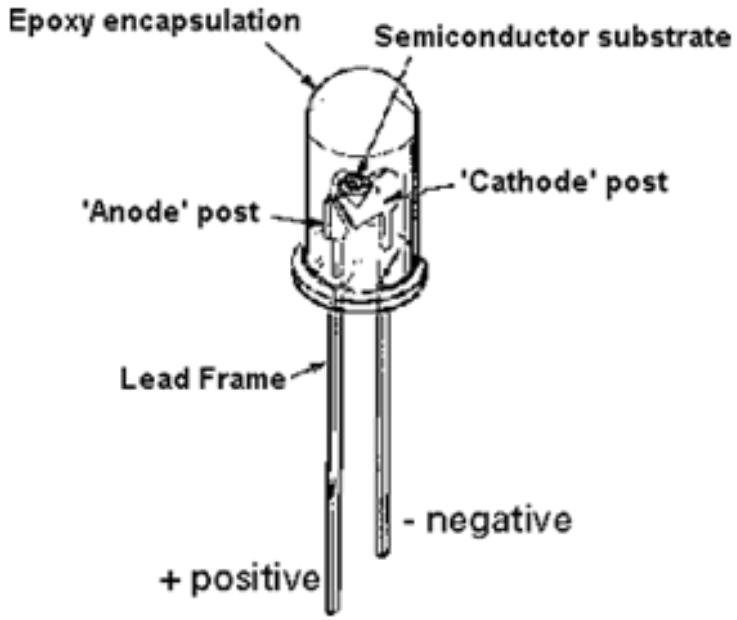
บทที่ 3 – RGB LED และ While Loop

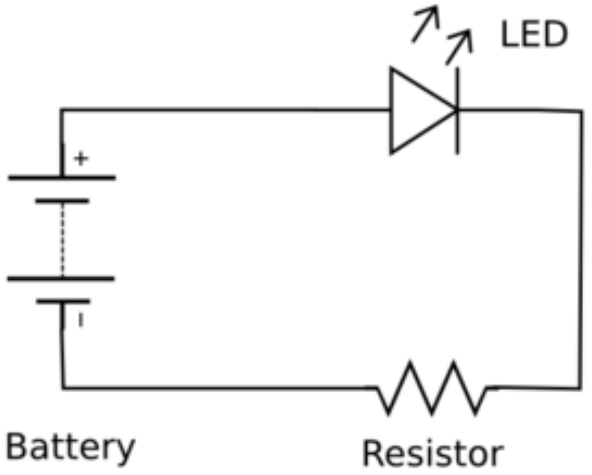
Project Lab Robotics Education

LED

- หลอดไฟ LED ย่อมาจาก Light Emitting Diode
- เป็นอุปกรณ์ประเภท Diode ซึ่งแปลว่าจะนำไฟฟ้าได้ในเพียงทิศทางเดียว
- ทำจากวัสดุประเภท Semiconductor เช่น silicon (ซิลิคอน) ที่นำไฟฟ้าได้ในบางกรณีเท่านั้น

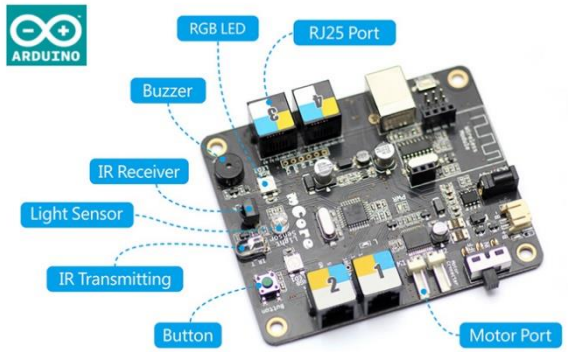


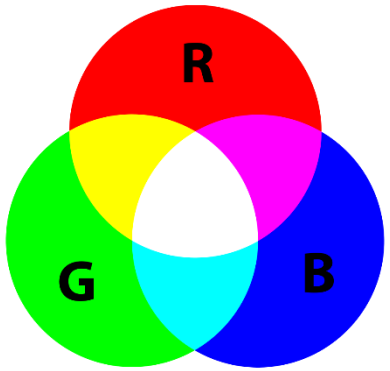


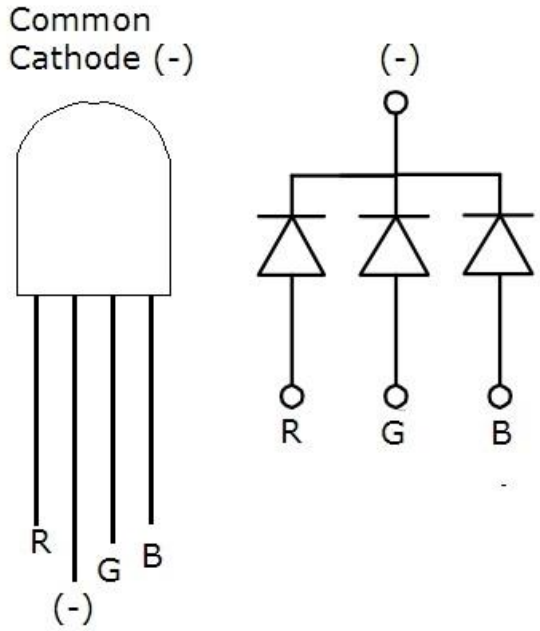


RGB LED

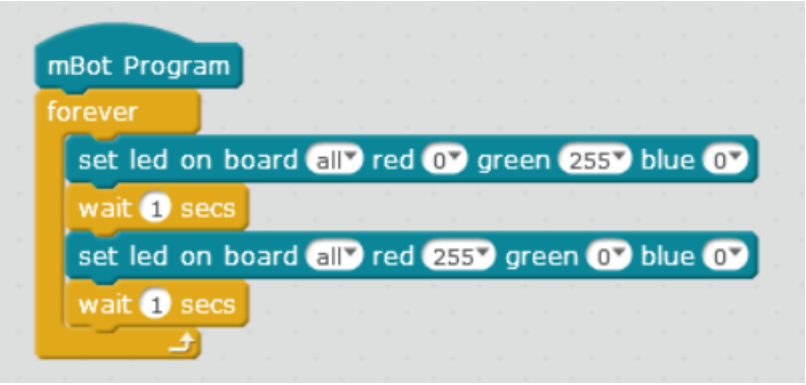
- บนบอร์ดมีไฟ **RGB LED**





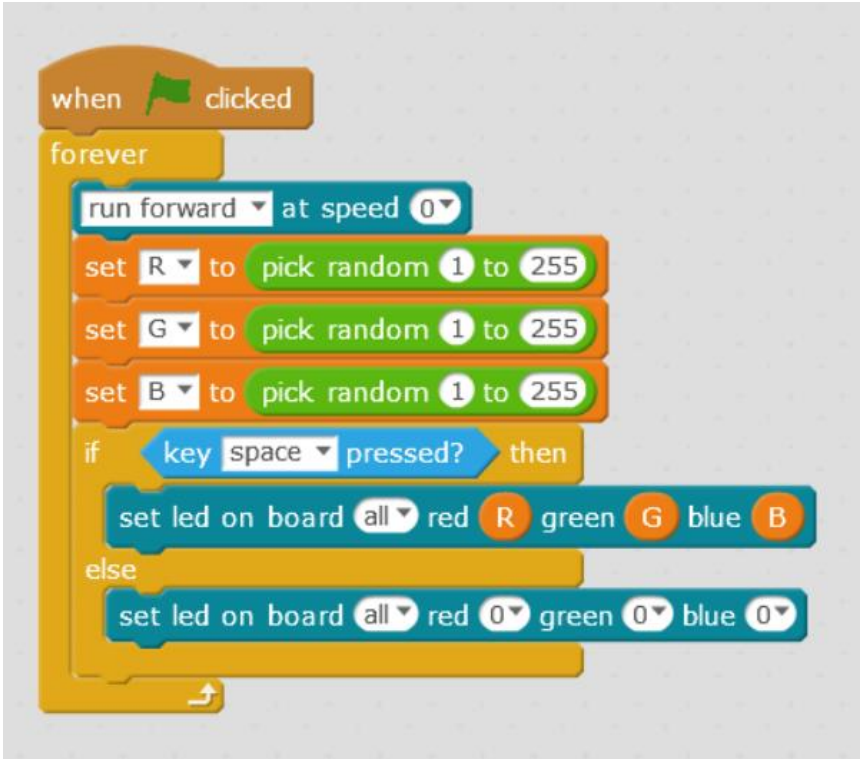


- ลองสั่งหุ่นยนต์ให้ทำแต่ละสี
- สั่งหุ่นยนต์ทำไฟกระพริบ **แดง** / **น้ำเงิน**
- สั่งหุ่นยนต์ทำไฟกระพริบ **เหลือง** / **ม่วง**



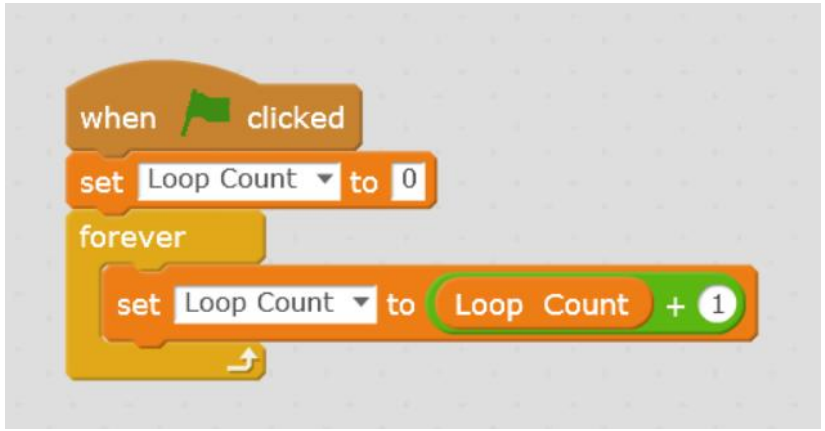
Random

- คอมพิวเตอร์ สามารถ**generate** (สร้าง) เลข **random** (สุ่ม) ได้โดยใช้สูตรคณิตศาสตร์
- ในโปรแกรมเราจะสร้างเลข **Random** ด้วยคำสั่ง **pick random**
- บล็อก **pick random** จะเลือกตัวเลข ระหว่าง 1-255 ตามตัวอย่าง
- ความสว่างสูงสุดของ **LED** คือ 255



Loop Count

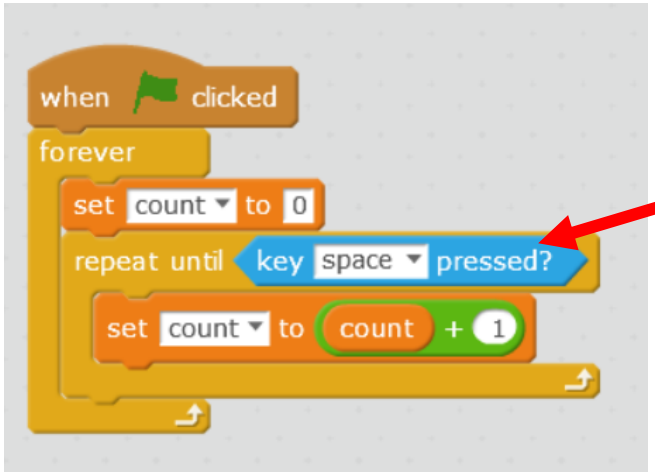
- ดูว่าหุ่นยนต์วน **Loop** เร็วแค่ไหน?
- ใน **10** วินาที หุ่นยนต์ทำงานไป _____ Loop
- จากการทดลอง หนึ่ง **Loop** ใช้เวลาเท่ากับ : $\frac{10}{\text{จำนวนลูป}}$ = _____วินาที



Loop ที่	Loop Count	Loop Count + 1
1	0	0 + 1 = 1
2	1	1 + 1 = 2
3	2	2 + 1 = 3
4	3	
5		
6		

Repeat Until Loop

- **Repeat until** คือ **Loop** ที่จะทำงานวนไปเรื่อยๆ จนกว่า กรณีจะเป็นจริง



เมื่อกรณีนี้เป็นจริง
ถึงจะมีการออกจากลูป

- **Count** จะเริ่มที่ 0 และมีการนับไปเรื่อยๆ จนกว่าจะมีการกด **space**
- เมื่อกด **Space** จะทำให้ตัว **count reset** กลับไปที่ 0 ใหม่และเริ่มนับไปเรื่อยๆ อีกรอบ

LED ที่ไล่เฉดสี

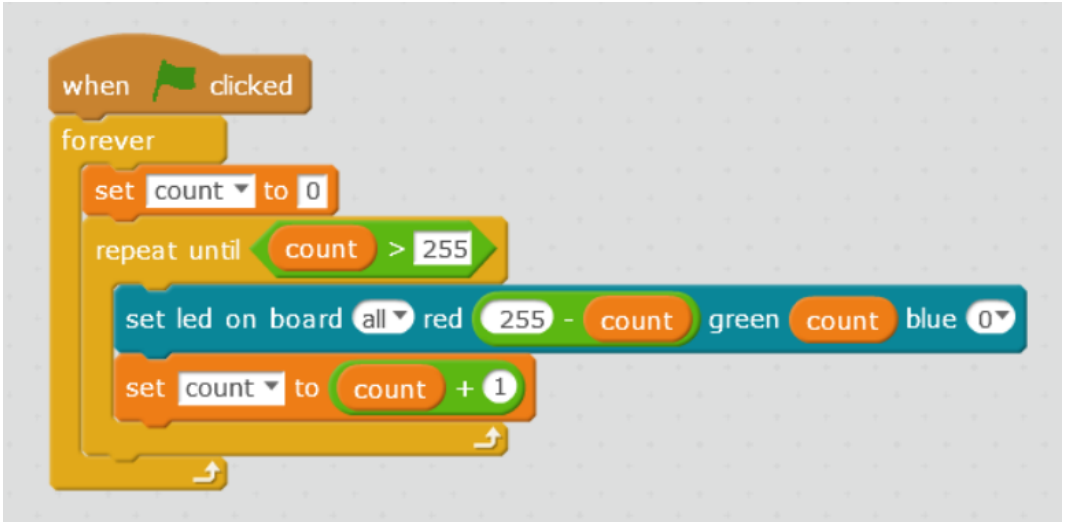
- นี่คือโปรแกรมที่จะไล่สีไฟจากสีแดงเป็นสีเขียว
- ตอนที่เข้า repeat until loop (While Loop) ตอนแรก ตัวแปร count = 0
- ด้วยการวน repeat until loop แต่ละครั้งตัวแปร count จะเพิ่มขึ้นทีละหนึ่ง
- เมื่อตัวแปร count เพิ่มจนมากกว่า 255 กรณีสำหรับการออกจาก repeat until loop จะเป็นจริง และมีการออกจาก repeat until loop

ตัวอย่างการเขียน

Start
Loop Forever

Set count = 0
Repeat until loop

Set LED, R= 255 – count, G = count, B = 0
Set count = count + 1



LED ไล่เฉดสี (โจทย์เก็บคะแนน)

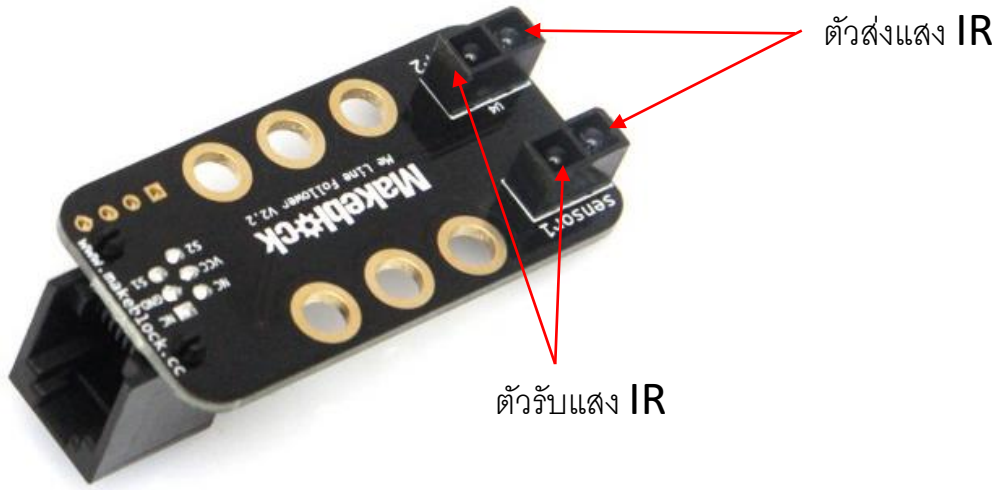
- ทำให้ LED ไล่เฉดสี จาก แดงไปเขียว จากเขียวไปน้ำเงิน จากน้ำเงินไปแดง

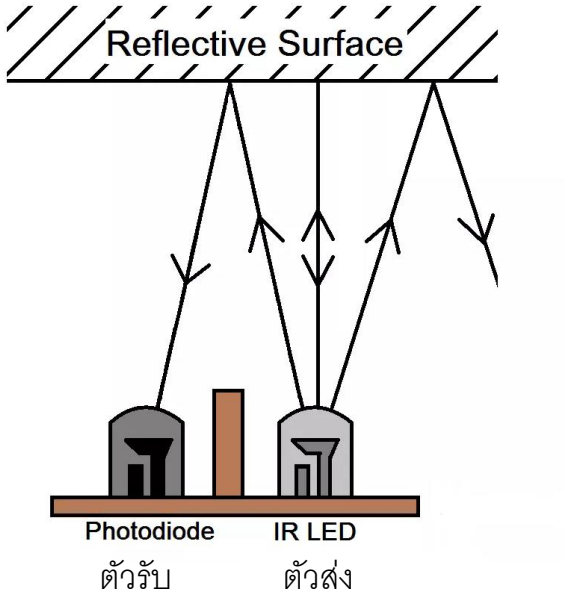
บทที่ 4 — หุ่นยนต์เดินตามเส้น

Project Lab Robotics Education

หุ่นยนต์เดินตามเส้น – Line Follower Sensor

- สำหรับการเดินตามเส้นเราจะใช้ Line Follower Sensor ซึ่งประกอบด้วยเซนเซอร์สองชุดคือ Sensor 1 และ Sensor 2
- แต่ละชุด Sensor ประกอบด้วย ตัวส่ง และตัวรับ แสง Infrared (IR)

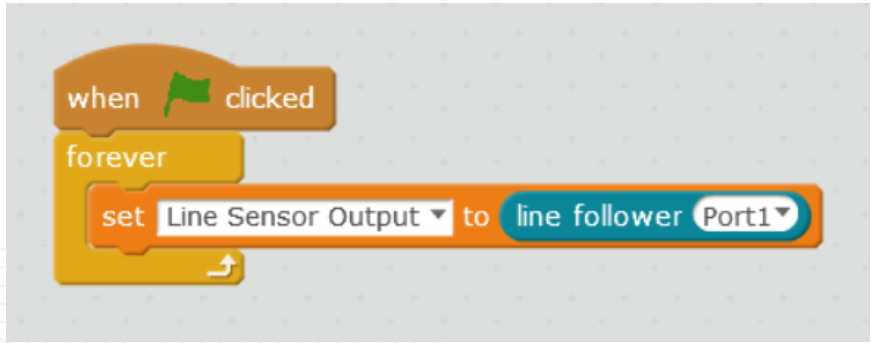




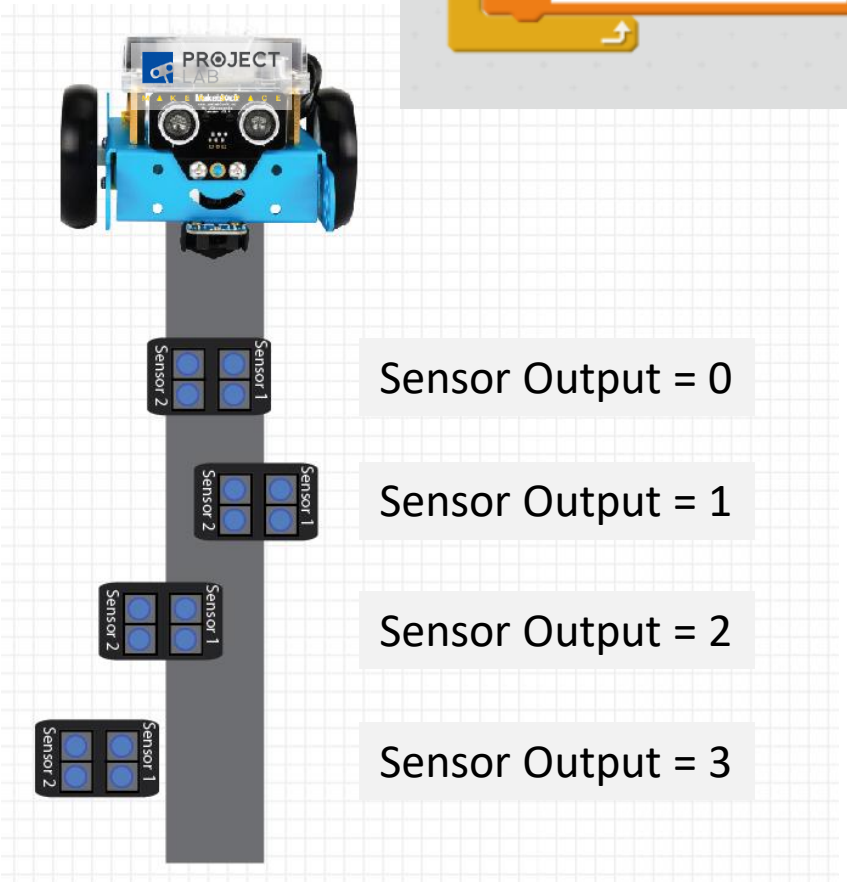
- สีอะไรสะท้อนแสง Infrared ได้มากหรือน้อย? สีดำจะสะท้อนแสง _____ สีขาวจะสะท้อนแสง _____

ทดลอง Output ของ Line Follower Sensor ใน Scratch Mode

- ตัวอย่างการใช้ตัวแปรอ่านค่า **Line Follower Sensor**



Sensor 2	Sensor 1	Output
ดำ	ดำ	0
ดำ	ขาว	1
ขาว	ดำ	2
ขาว	ขาว	3



หุ่นยนต์เดินตามเส้น – กลยุทธ์

- เราจะทำให้หุ่นยนต์เดินตามเส้นได้อย่างไร?

- ถ้าค่าเซนเซอร์เท่ากับ 3 (เจอสีขาวทั้งคู่) จะทำอย่างไรดี?

หุ่นยนต์เดินตามเส้น

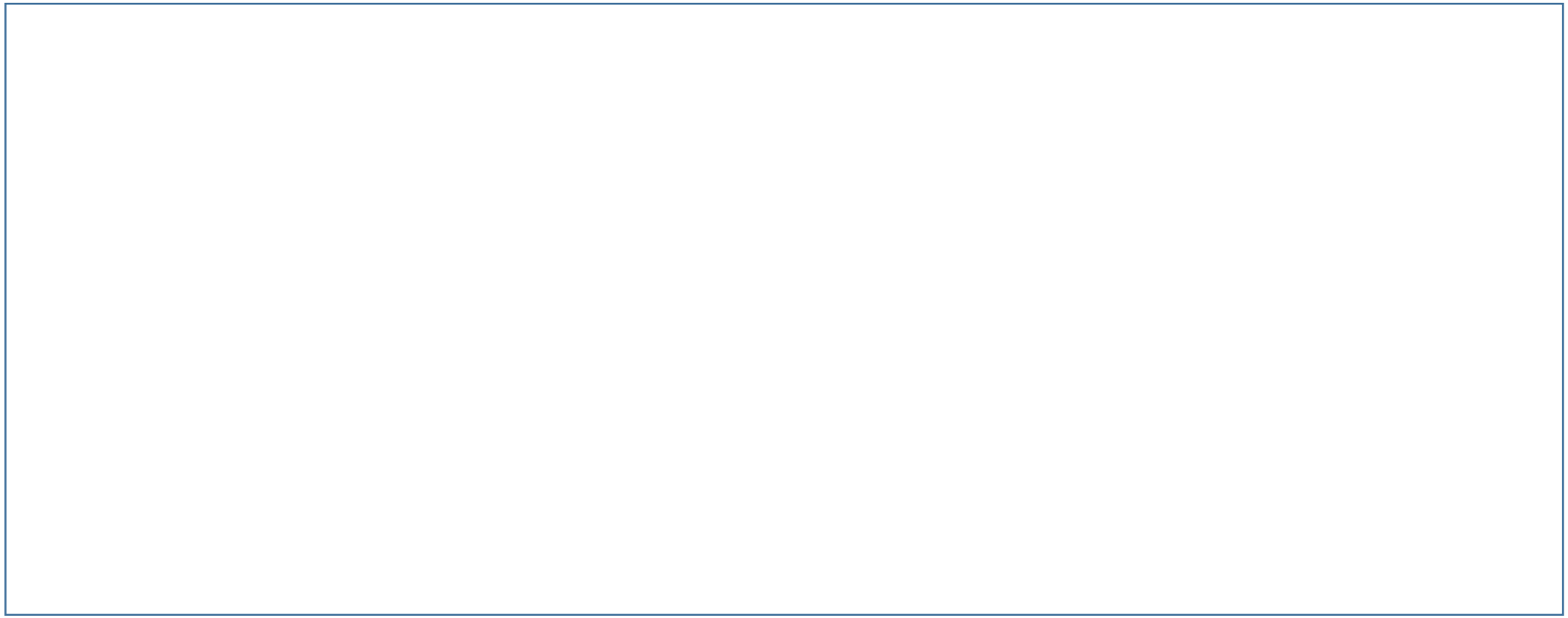
- เขียนโค้ดคำตอบ

โจทย์เก็บคะแนน: หุ่นยนต์เดินตามเส้น + ไฟกระพริบ

- ทำอย่างไรให้ไฟของหุ่นยนต์กระพริบเป็นสองสีพร้อมกับเดินตามเส้น
- ใบ้ – ไม่สามารถใช้ wait (เพราะ wait จะทำให้โปรแกรมค้างการทำงานที่คำสั่ง wait และจะเลิกอ่านค่าเซ็นเซอร์ และทำการควบคุมให้หุ่นยนต์เดินตามเส้นไปชั่วขณะ)
- ใบ้ – ใช้ตัวแปร Loop Count
- ใบ้ – ใช้ If Statement
- ใบ้ – ใช้เครื่องหมายคณิตศาสตร์ มากกว่า น้อยกว่า

หุ่นยนต์เดินตามเส้น + ไฟกระพริบ (โจทย์เก็บคะแนน)

- ทำอย่างไรให้ไฟของหุ่นยนต์กระพริบเป็นสองสีพร้อมกับเดินตามเส้น

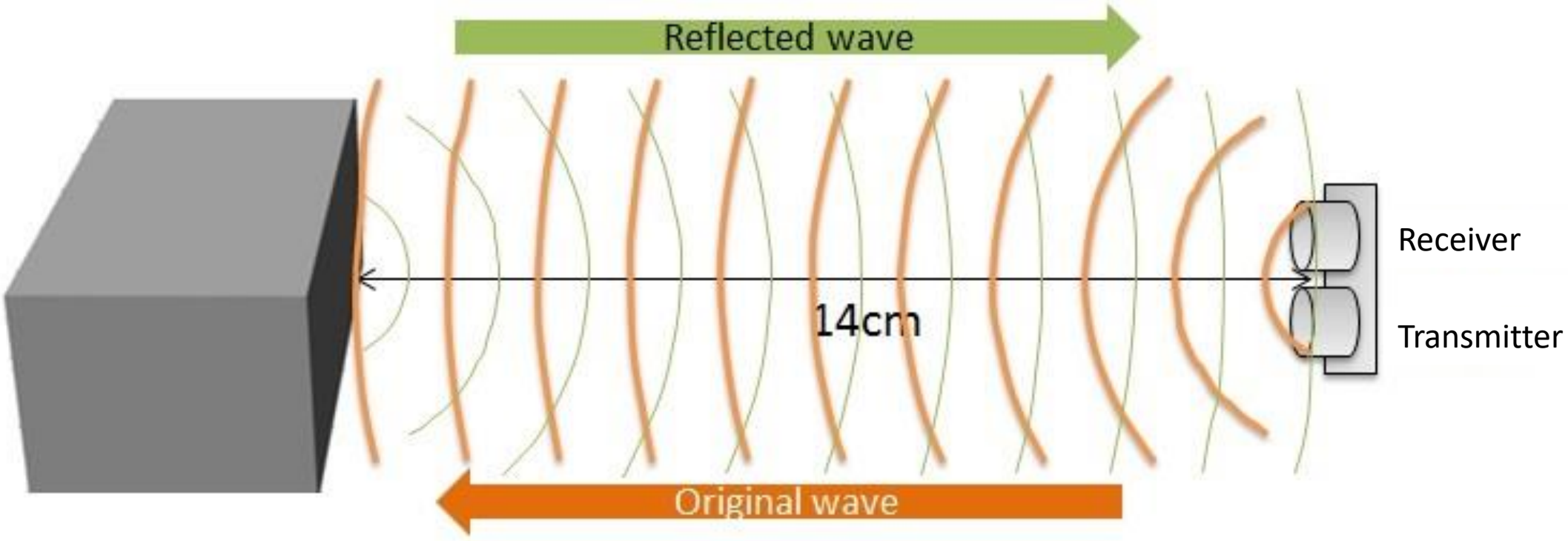


บทที่ 5 — หุ่นยนต์หลบหลีกสิ่งกีดขวาง

Project Lab Robotics Education

Ultrasonic Sensor

- จากคลื่นที่สะท้อนกลับ **Ultrasonic Sensor** จะใช้วัดระยะทางได้อย่างไร?



Measure Ultrasonic Sensor

- ในระดับพื้นฐานสิ่งที่เราได้จาก ultrasonic sensor คือเวลาของการปล่อยคลื่นจาก transmitter และเวลาที่ได้รับคลื่นสะท้อนจาก receiver
- จากเวลานี้เราจะคำนวณระยะทาง โดยความสัมพันธ์ระหว่าง ความเร็ว ระยะทาง และ เวลา คือ

$$\text{ความเร็ว} = \frac{\text{ระยะทาง}}{\text{เวลา}} \qquad \textbf{หรือ} \qquad \text{ความเร็ว} \times \text{เวลา} = \text{ระยะทาง}$$

- คลื่น ultrasonic เป็นคลื่นเสียงชนิดหนึ่งที่มีความถี่สูงจนมนุษย์ไม่สามารถได้ยิน โดยทั่วไปเสียงเดินทางในอากาศด้วยความเร็วประมาณ 340 m/s

$t_1 =$ เวลาที่ปล่อยคลื่น

$t_2 =$ เวลาที่รับคลื่น

$d =$ ระยะทางถึงวัตถุ

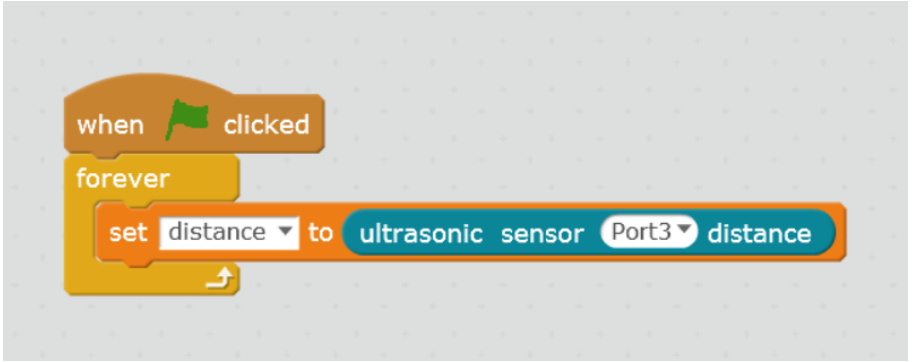
$$d = \frac{v(t_2 - t_1)}{2}$$

1. ยิงคลื่นที่เวลา 26.00 วินาที รับคลื่นสะท้อนได้ที่เวลา 26.05 วินาที วัตถุอยู่ห่างจากเซนเซอร์เท่าไหร่? อย่าลืมใส่หน่วยระยะทางที่ถูกต้อง

2. ทำไมถึงต้องมีการ หารด้วยสอง (/2) ในสูตรนี้

Measure Ultrasonic Sensor

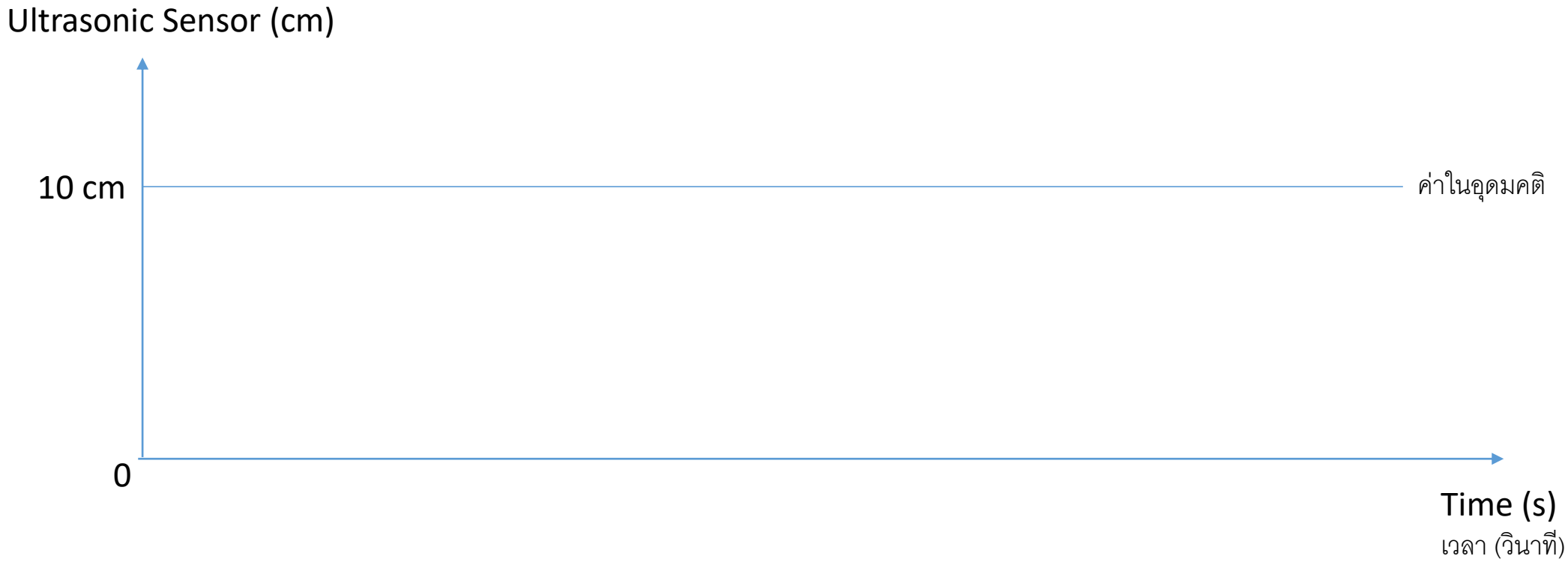
- วัดและแสดงผลระยะทางโดยใช้ **Ultrasonic Sensor**



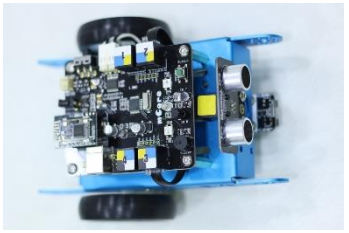
- สังเกตอะไรได้บ้างจากค่าที่อ่านได้? ค่านั้นนิ่งไหม? (ขนาดไหน) ค่านั้นแม่นยำไหม? (จะตรวจสอบได้อย่างไร)

Measure Ultrasonic Sensor

- ลองทำการทดลอง ให้ **Ultrasonic Sensor** วัดระยะห่างจากวัตถุที่อยู่นิ่งๆ ที่ระยะ 10 **cm** ลองวาด **graph** ดูว่าค่าที่อ่านได้จะเป็นอย่างไร?



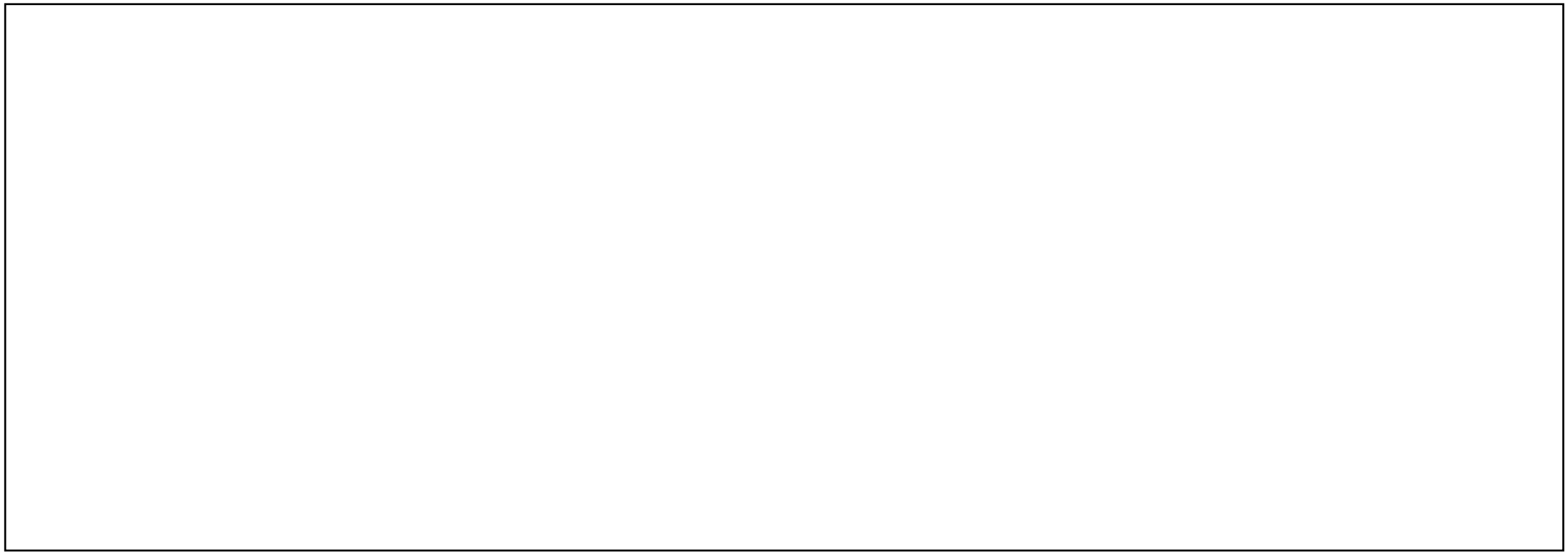
Obstacle Avoidance



สิ่งกีดขวาง

Obstacle Avoidance

- เขียนโปรแกรมสำหรับหุ่นยนต์ที่สามารถหลบสิ่งกีดขวาง หุ่นยนต์ตัวนี้จะวิ่งตรงไปเรื่อยๆจนกว่าจะเจอสิ่งกีดขวางแล้วจึงทำการเลี้ยวหลบ เมื่อเลี้ยวหลบแล้วก็จะวิ่งตรงต่อไป



Random Obstacle Avoidance

- เขียนโปรแกรมให้หุ่นยนต์มีบุคลิก หุ่นยนต์ตัวนี้จะมีการสุ่มทิศทางการเดิน และการเลี้ยว
- การสุ่มทิศทางเดินขณะที่ไม่มีสิ่งกีดขวางอยู่ข้างหน้า

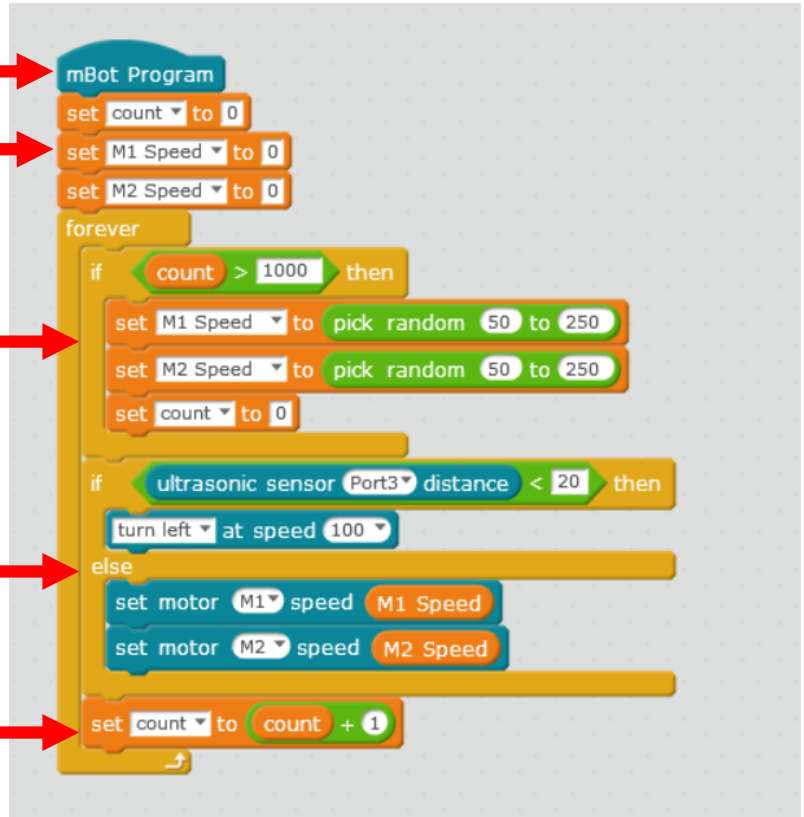
mBot โหมด

กำหนดค่าตั้งต้น ทำนี่ครั้งเดียว

เลือกความเร็วใหม่ทุกๆครั้งที่ **count** มากกว่า 1000

เจอสิ่งกีดขวางแล้วเลี้ยวซ้ายหลบ

เพิ่ม **count** ทีละ 1



Random Obstacle Avoidance

จากโปรแกรมหุ่นยนต์ในหน้าที่แล้ว จงตอบคำถามเหล่านี้ อย่าลืมใช้ mBot โหมด เพราะหุ่นยนต์จะทำงานได้เร็วกว่าใน Scratch โหมด

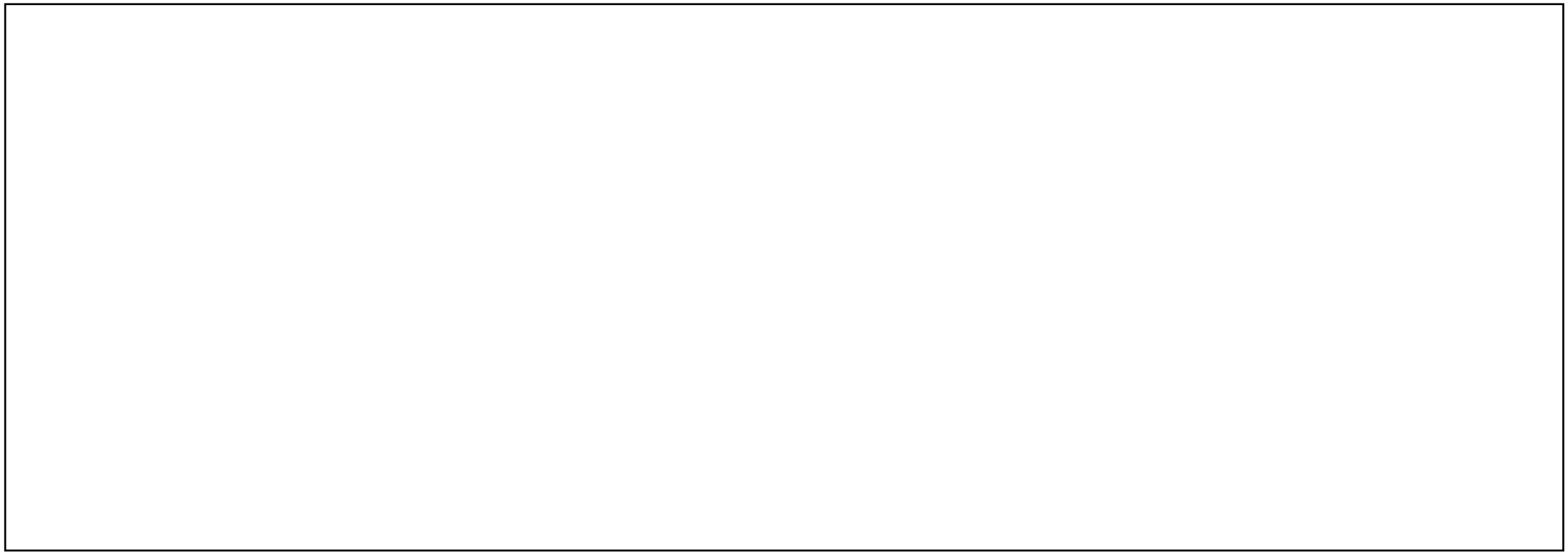
1. หุ่นยนต์จะเริ่มเลี้ยวเมื่อห่างจากวัตถุเป็นระยะทางเท่าไหร่?

2. ถ้าหนึ่งลูปของโปรแกรมใช้เวลา 0.001วินาที หุ่นยนต์จะเลือกทิศทางเดินทุกๆกี่วินาที

3. ถ้าเราให้หุ่นยนต์สุ่มทิศทางใหม่ทุกๆลูป จะเกิดอะไรขึ้น? หุ่นยนต์จะเคลื่อนไหวอย่างไร? ทำไม?

Random Obstacle Avoidance (โจทย์เก็บคะแนน)

- ตอนนี้หุ่นยนต์เราสุ่มทิศทางเดินไปข้างหน้าได้แล้ว โจทย์ของเราตอนนี้ก็คือต่อยอดโปรแกรมสุ่มทิศทางที่จะเลี้ยวแต่ละครั้ง เขียนโปรแกรมที่สำเร็จด้านล่างนี้

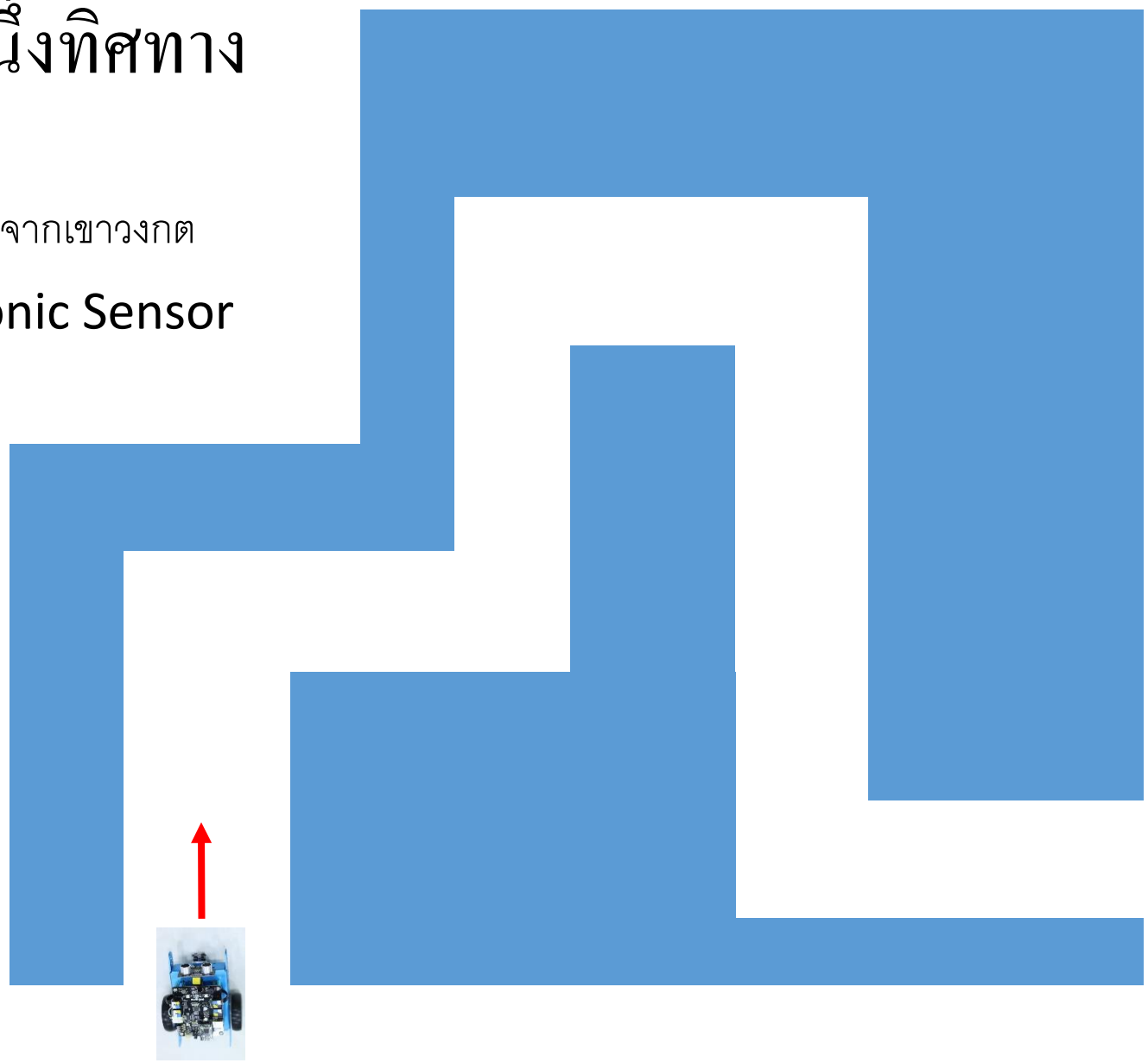


บทที่ 6 – หุ่นยนต์เขาวงกต

Project Lab Robotics Education

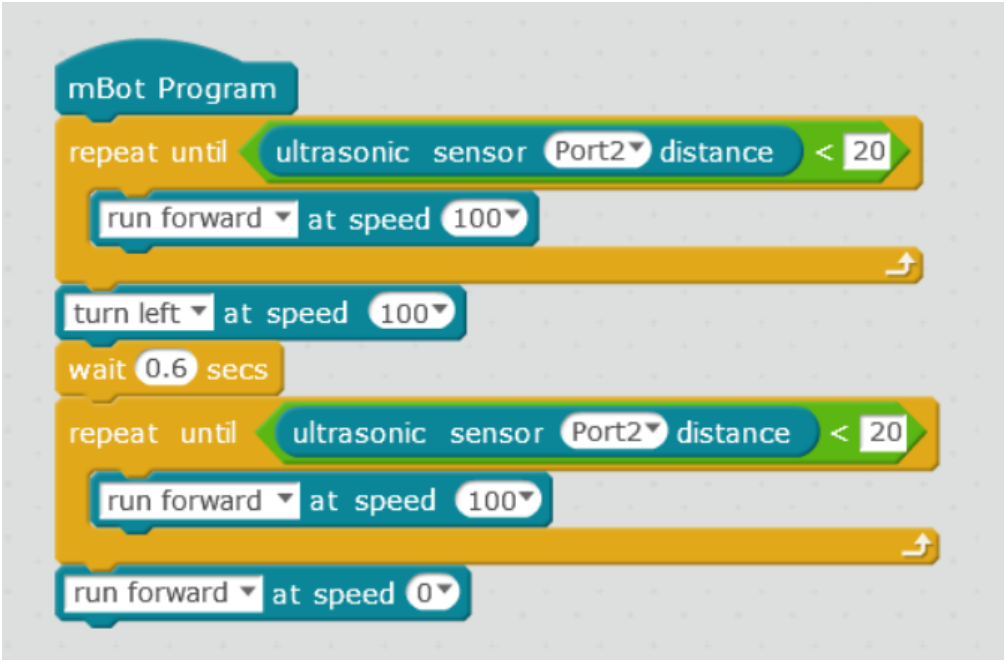
เขาวงกตหนึ่งทิศทาง

- ทำให้หุ่นยนต์ออกจากเขาวงกต
- โดยใช้ Ultrasonic Sensor



เขาวงกตหนึ่งทิศทาง (Repeat Until)

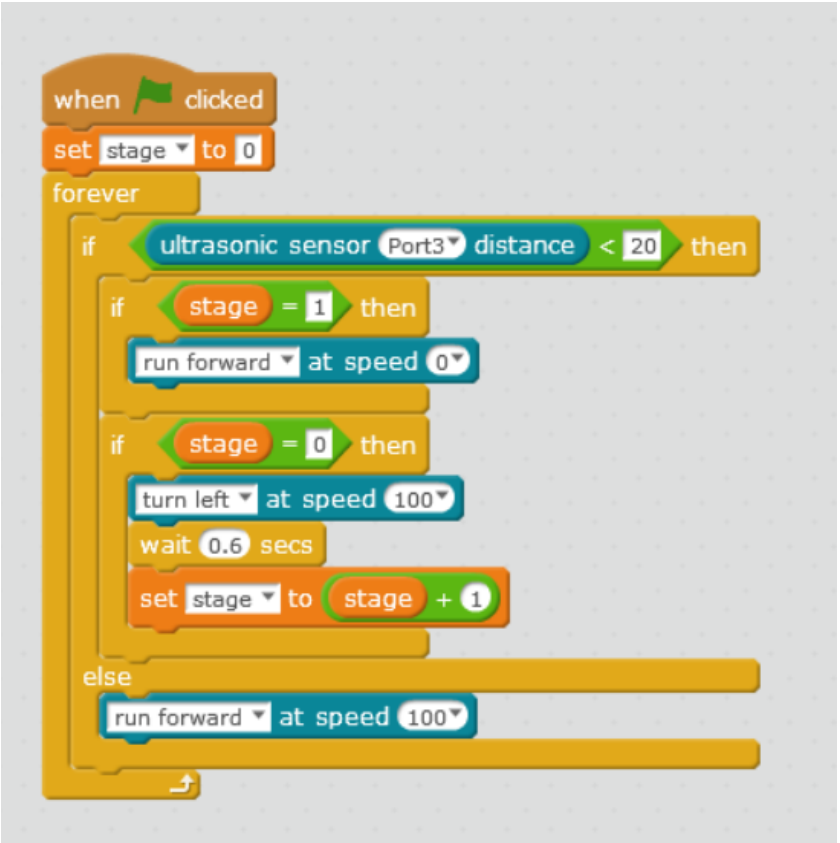
- เราสามารถสั่งงานหุ่นยนต์ให้ทำงานเป็นขั้นเป็นตอนได้โดยการใช้ **Repeat Until** และเราจะแบ่งโปรแกรมเราเป็นสัดส่วน
- ตัวอย่าง
  1. หุ่นยนต์เดินตรงเรื่อยๆจนเจอกำแพง
  2. หุ่นยนต์เลี้ยวซ้าย 0.6 วินาที
  3. หุ่นยนต์เดินตรงเรื่อยๆจนเจอกำแพง
  4. หุ่นยนต์หยุด



เขาวงกตหนึ่งทิศทาง (Variable and If Statement)

- เราสามารถสั่งงานหุ่นยนต์ให้ทำงานเป็นขั้นเป็นตอนได้โดยการใช้ ตัวแปร **(Variable)** และ **(if Statement)** เพื่อที่จะแบ่งโปรแกรมเราเป็นสัดส่วน สำหรับการทำงานในแต่ละส่วน

1. หุ่นยนต์เดินตรงเรื่อยๆจนเจอกำแพง
2. stage = 0 หุ่นยนต์เลี้ยวซ้าย 0.6 วินาที
3. ปรับให้ stage = 1
4. หุ่นยนต์เดินตรงเรื่อยๆจนเจอกำแพง
5. stage = 1 หุ่นยนต์หยุด



Start
Set **stage** = 0
Forever Loop

```
If ultrasonic sensor < 20 then
        if stage = 1 then
                forward
        if stage = 0 then
                turn left
                wait 0.6s
                set stage = stage+1
Else
        run forward
```

เขาวงกตหนึ่งทิศทาง (โจทย์เก็บคะแนน)

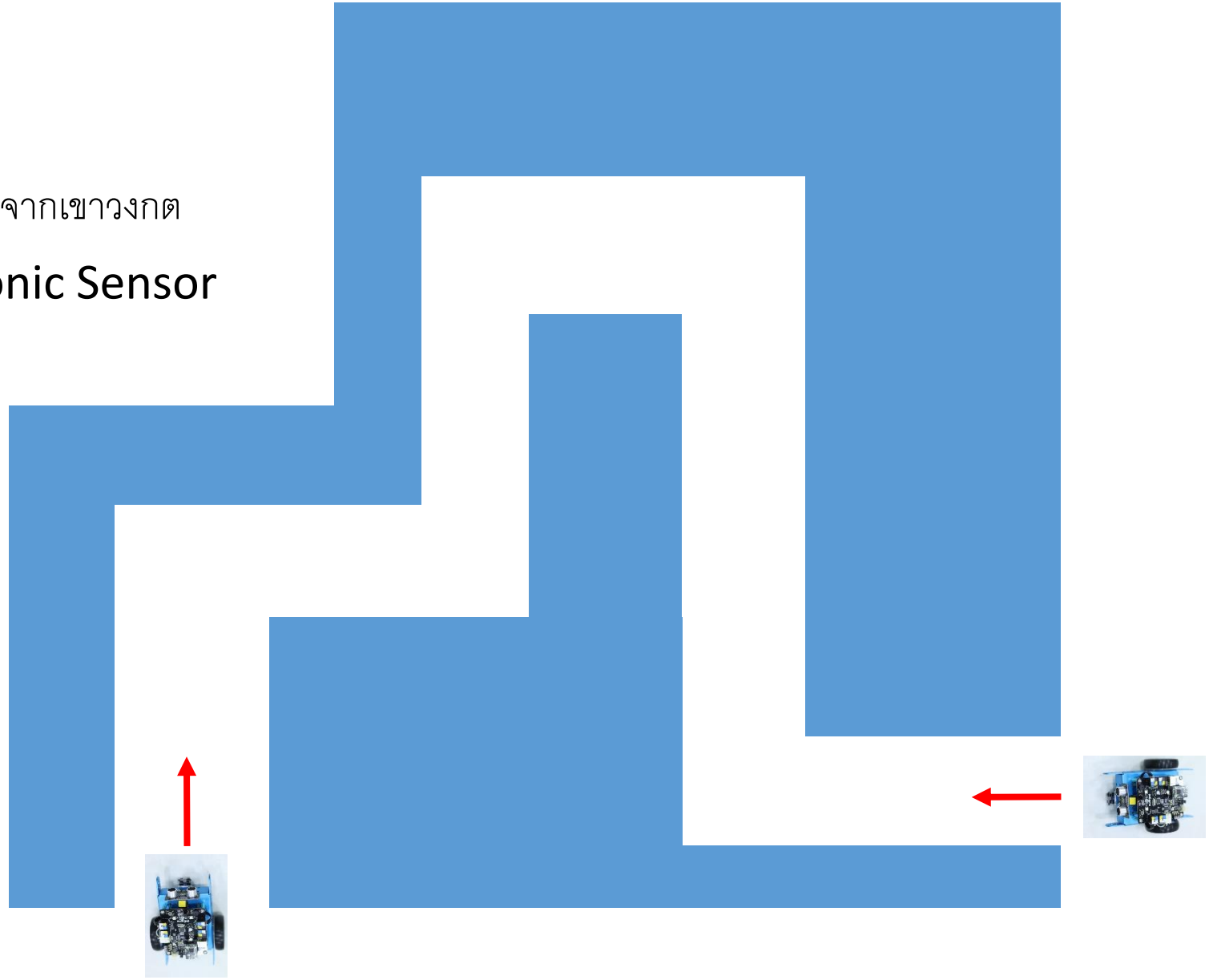
- ใช้กรรมวิธีใช้ ตัวแปร (Variable) และ if statement เพื่อให้หุ่นยนต์เดินออกจากเขาวงกตได้สำเร็จ เขียนโปรแกรมที่เสร็จแล้วด้านล่างนี้

บทที่ 7 — หุ่นยนต์เขาวงกต เลือกทิศทางเอง

Project Lab Robotics Education

เขาวงกต

- ทำให้หุ่นยนต์ออกจากเขาวงกต
- โดยใช้ Ultrasonic Sensor



เขาวงกต - หุ่นยนต์เลือกทางเลี้ยวด้วยตัวเอง

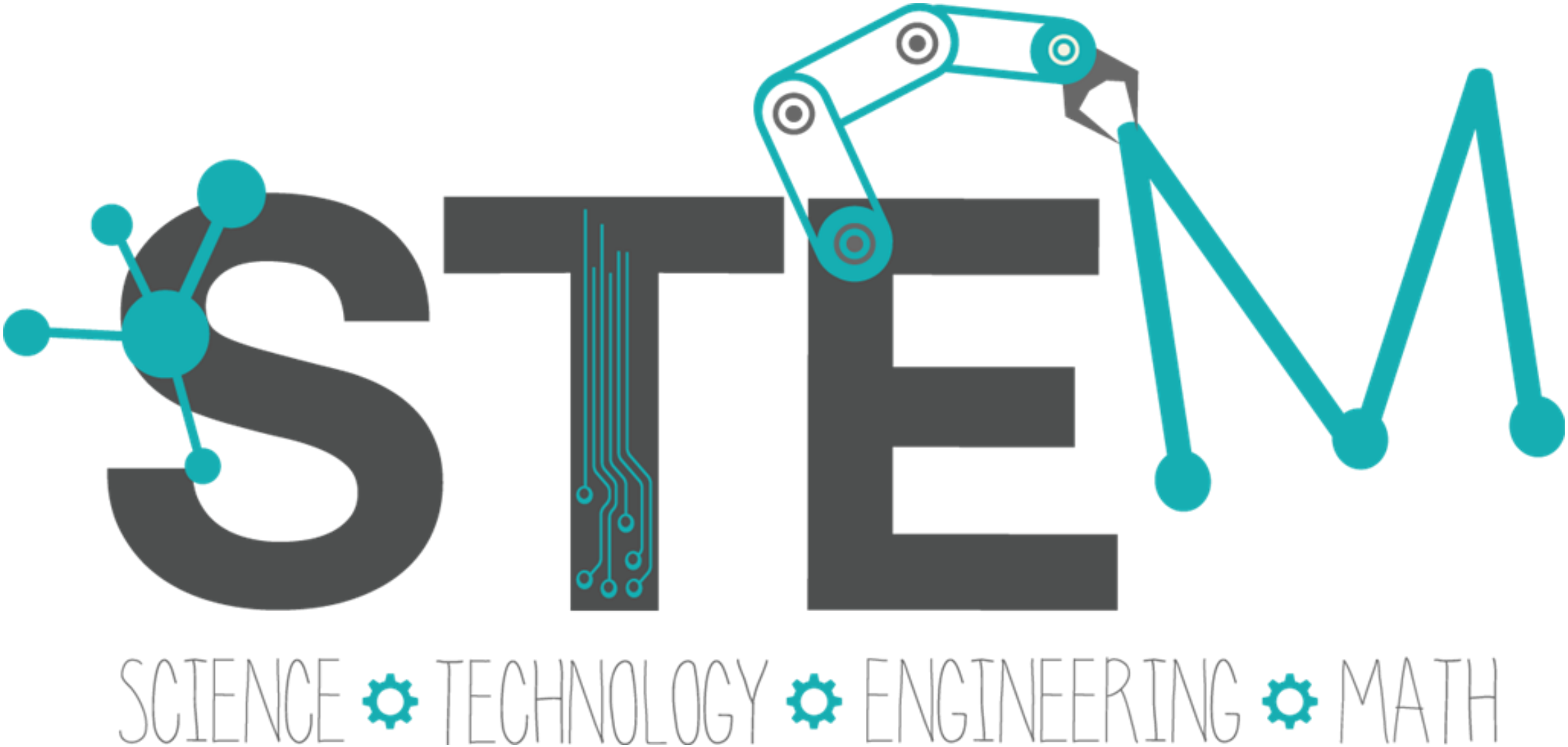
- เขียนโปรแกรมให้หุ่นยนต์ออกจากเขาวงกตได้ไม่ว่าจะเข้าจากทางไหนก็ตาม
- จะทำอย่างไรถ้าเราไม่รู้ว่าทางโล่งอยู่ด้านซ้ายหรือขวา?
- หุ่นยนต์จะต้องตัดสินใจเอง
- ให้หุ่นยนต์ต้องตรวจสอบว่าทางไหนโล่งและให้เดินไปทางนั้น

เขาวงกดหนึ่งทิศทาง (โจทย์เก็บคะแนน)

- โจทย์ข้อนี้จะเก็บคะแนนจากการปฏิบัติของหุ่นยนต์ในสนาม
- Basic Score: ทำให้หุ่นยนต์สามารถเดินออกจากเขาวงกดได้ไม่ว่าจะเข้าจากด้านไหนก็ตาม
- Bonus Score: หุ่นยนต์เดินออกจากเขาวงกด แล้วหันกลับ แล้วเดินกลับมาที่จุดเริ่มต้นได้

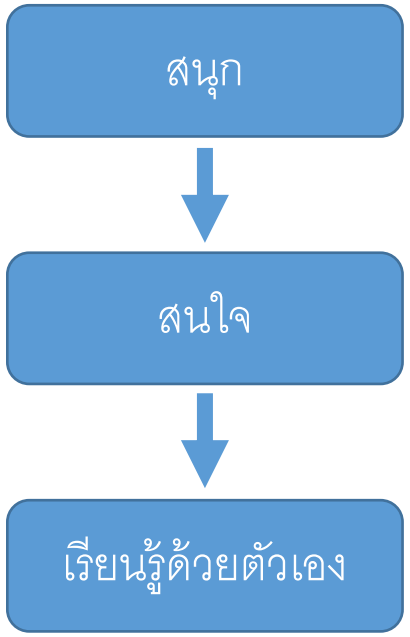
บทที่ 8 – สรุปการเรียน การสอน หุ่นยนต์

Project Lab Robotics Education


STEM
SCIENCE TECHNOLOGY ENGINEERING MATH

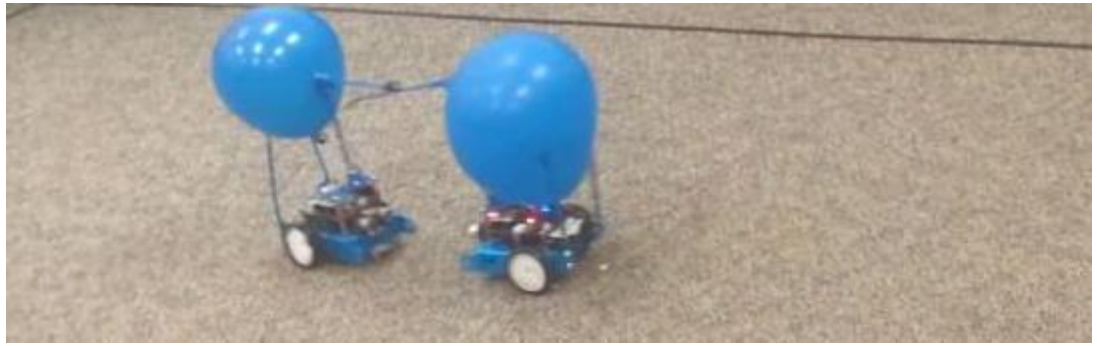
สอนให้สนุก

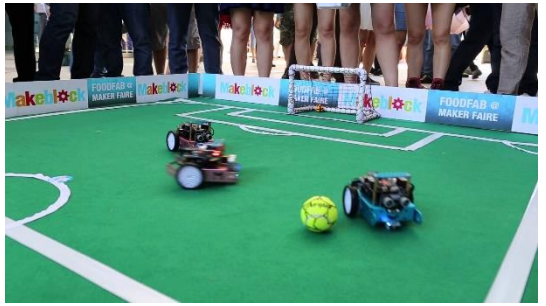
1. ทำให้นักเรียนชอบ ให้สนุก
2. ความชอบจะทำให้มีความสนใจ
3. สนใจแล้วก็จะขยันใฝ่หาความรู้ เรียนรู้ด้วยตัวเอง

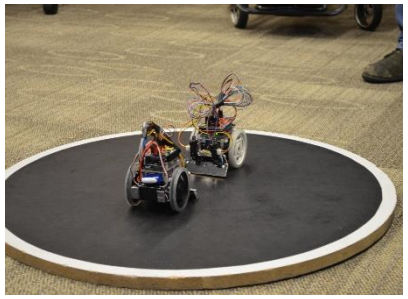


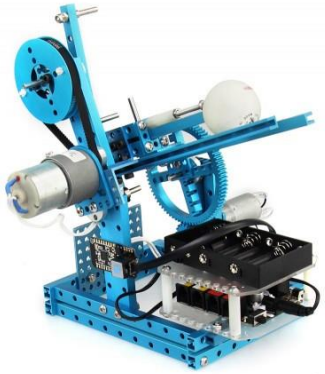
จัดกิจกรรมที่สนุก

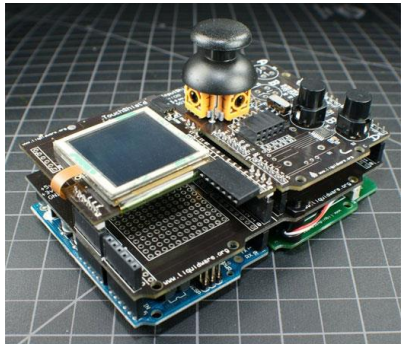
- หุ่นยนต์บังคับ แบบ balloon fight
- หุ่นยนต์บังคับ แบบเตะฟุตบอล
- หุ่นยนต์ ซูโม่ อัตโนมัติ
- หุ่นยนต์ ยิงลูกปิงปอง
- หุ่นยนต์ส่งของ
- หุ่นยนต์ ปลูกต้นไม่
- หุ่นยนต์ลบกระดานอัตโนมัติ
- สร้างเครื่องเล่น เกมกด
- สร้างเกมส์ Scratch

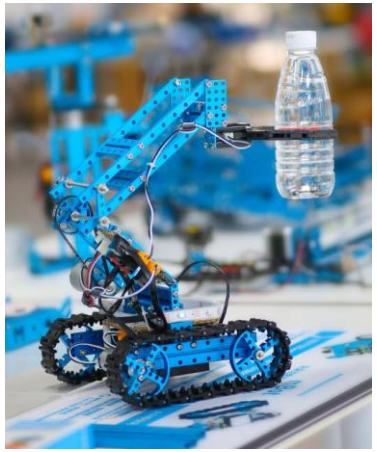








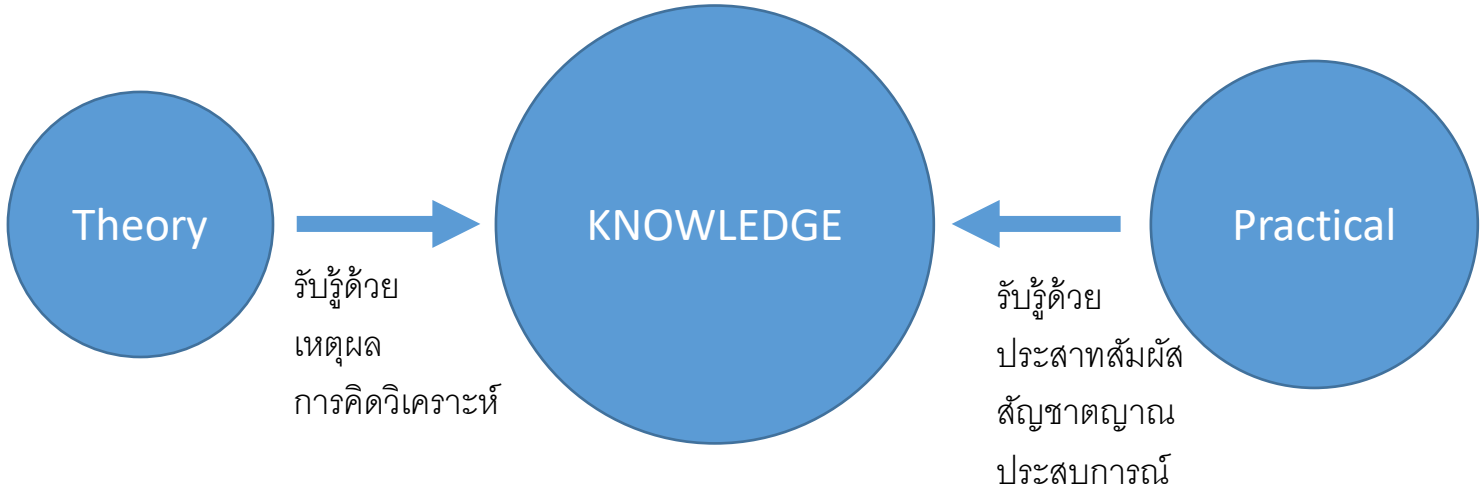


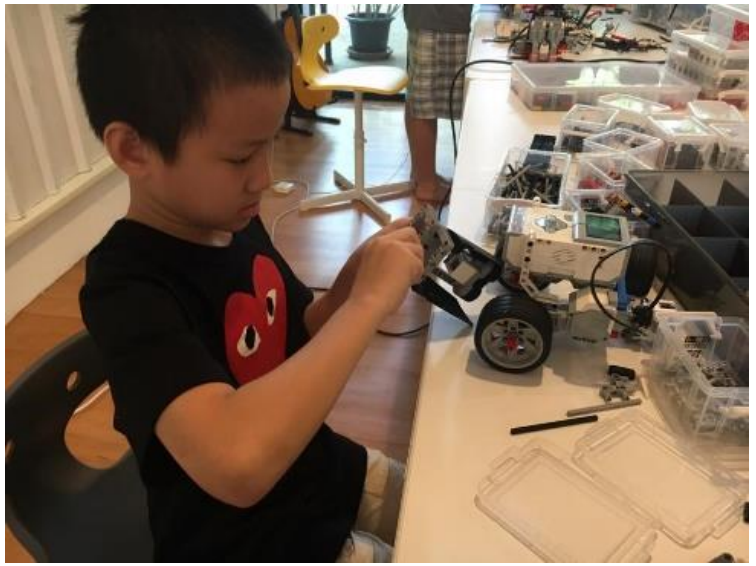


เน้นการลงมือทำ

“Tell me and I forget, teach me and I may remember, involve me and I learn.”
- Benjamin Franklin

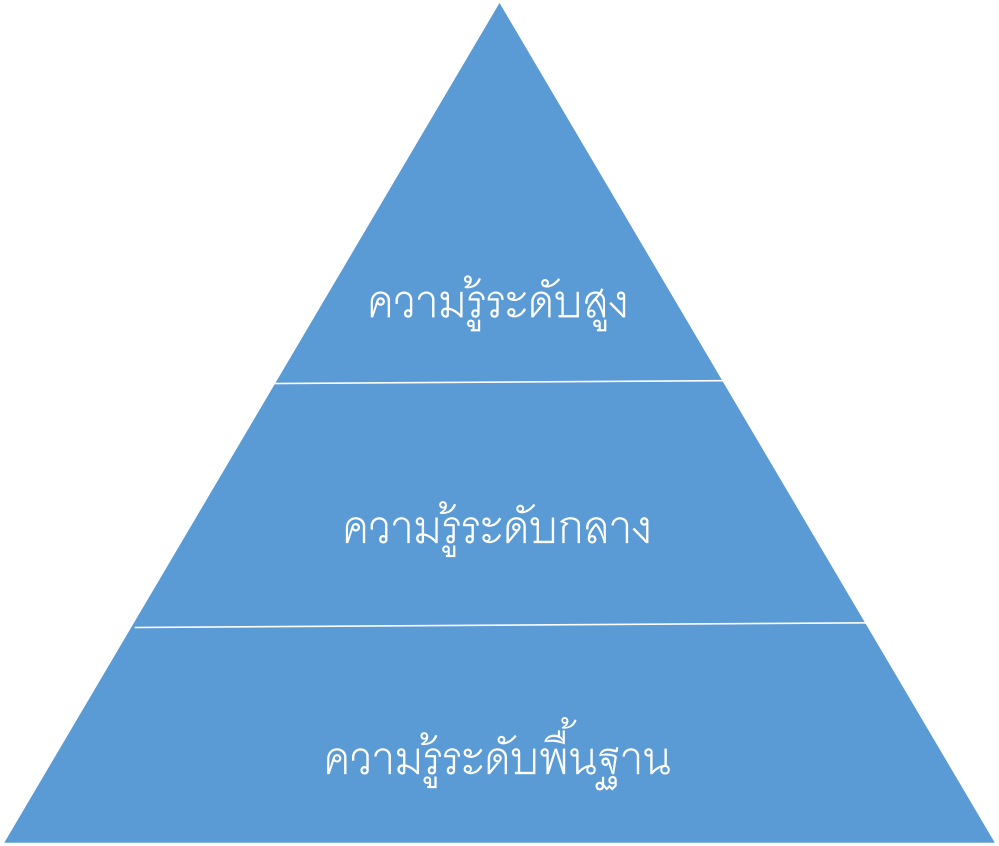
- เรียนรู้ผ่านการลงมือทำ
- มีประสบการณ์จริง เชื่อมโยงทฤษฎีและการทำงานจริงเข้ากันได้
- เข้าใจ hardware จริงๆ เช่น sensor, motor, computer
- ถ้าไม่ลงมือทำจริงๆ นักเรียนจะได้ภาพที่ไม่ครบองค์ประกอบ





ใจความสำคัญของเนื้อหาคือ Concept

- ความรู้ต่างๆก็คือ การเข้า **concept** (ประเด็นสำคัญ) นี่แหละ
  - เช่น concept ของ ความเร็ว คือ ระยะทาง เทียบกับเวลา
  - เช่น concept ของ การคูณ คือ การบวกซ้ำกันหลายๆรอบ
  - เช่น concept ของ ลูป คือการวนทำคำสั่งโดยคอมพิวเตอร์
  - เช่น concept ของ ตัวแปร คือการเก็บค่าบางอย่าง สำหรับที่เราจะนำมาใช้ได้อีก
- เข้าใจ **concept** คือรากฐานสำหรับการขยายต่อยอดความรู้
- ถ้าความเข้าใจระดับพื้นฐานไม่ดี ไม่ถูกต้อง ก็จะต่อยอดได้ยาก



The Key is to…

Understand What & Why & How

เข้าใจเนื้อหา | เข้าใจเหตุผล | เข้าใจกรรมวิธี

พัฒนาความเข้าใจ Concept ด้วยตัวอย่าง

- Concept นั้นมักจะเป็นเรื่องที่ **นามธรรม**
- ซึ่งไม่ง่ายที่จะ **เห็นภาพ** และ **ทำความเข้าใจ**
- เพื่อเปลี่ยน Concept เป็น **รูปธรรม** เราจะใช้ **ตัวอย่าง** และ **การลงมือทำ**
- พอลงมือทำบ่อยๆเข้า เราจะเห็นภาพที่ชัดเจนขึ้น
- เมื่อเข้าใจ concept ของการเขียนโปรแกรมไม่ได้ทำให้เขียนโปรแกรมเป็น **เราต้องฝึกปฏิบัติด้วย**

พลิกแพลงใช้ความรู้ Applied Sciences

- ความรู้เฉยๆไม่ได้ก่อให้เกิดประโยชน์
- ประโยชน์เกิดขึ้นเมื่อนำความรู้มาใช้ตอบโจทย์ในชีวิตจริง เช่น
  - ✓ประโยชน์ของการคำนวณโครงสร้าง เกิดขึ้นเมื่อสร้างเป็นสะพานให้คนข้ามแม่น้ำ
  - ✓ประโยชน์ของการสร้างหุ่นยนต์ เกิดเมื่อสร้างเครื่องจักรที่เพิ่มผลผลิตให้โรงงาน
  - ✓ประโยชน์ของการเขียนโปรแกรม เกิดเมื่อเราเขียนโปรแกรมที่ช่วยในการทำงานแบบ Microsoft Word
- เราต้องฝึกให้นักเรียนรู้จักประยุกต์ใช้ความรู้ที่มี เพื่อตอบโจทย์ที่แตกต่างกันออกไป
- เพราะโจทย์ในโลกนั้นมีมากไม่รู้จักจบ เราจึงไม่สามารถที่จะฝึกสำหรับทุกๆโจทย์ทุกๆสถานการณ์ได้ สิ่งที่เราฝึกได้คือการรู้จักการพลิกแพลง และประยุกต์ใช้ความรู้เพื่อแก้ปัญหาต่างๆนาๆ

เราต้องการฝึกให้นักเรียน **คิดโจทย์เป็น** และ **ประยุกต์ใช้ความรู้** เพื่อตอบโจทย์

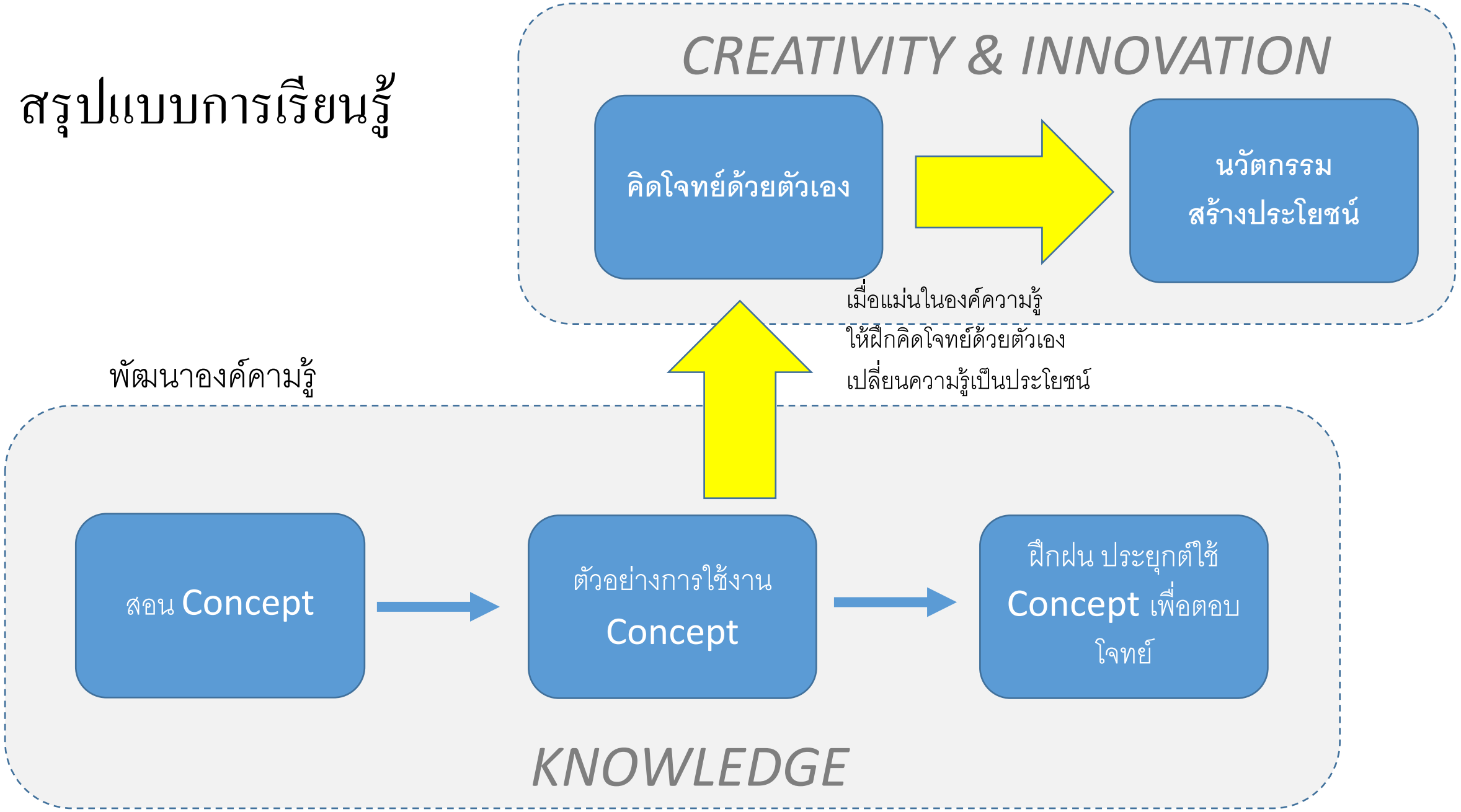
ตัวอย่างการประยุกต์ความรู้

ขั้นที่1 สอน Concept เกี่ยวกับ ตัวแปร If Statement และ Logic Operations

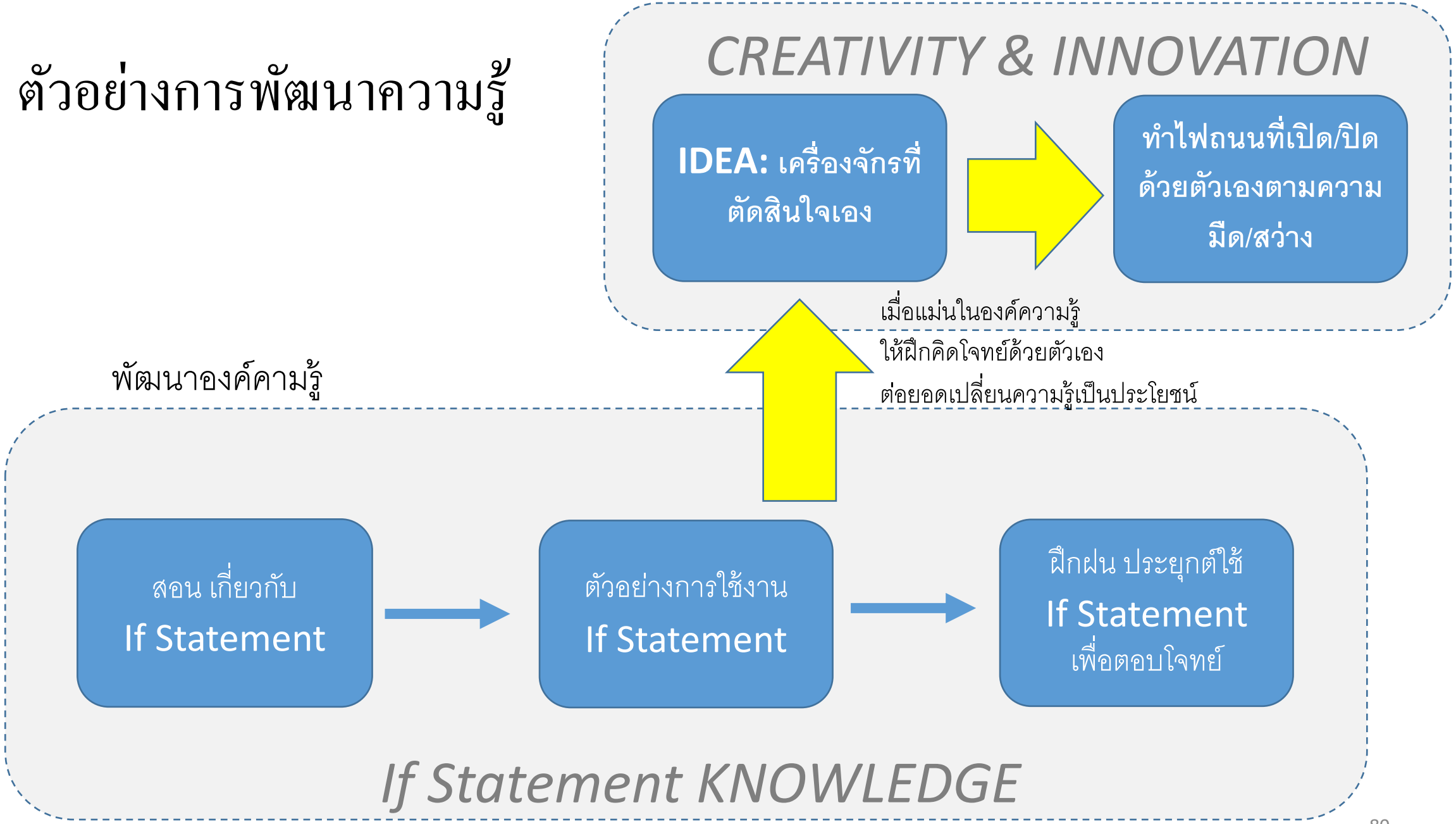
ขั้นที่2 สอนการใช้ ตัวแปร If Statement และ Logic Operations ด้วยตัวอย่างต่างๆ

ขั้นที่3 ให้นักเรียนประยุกต์ใช้ความรู้ concept เหล่านี้ ในการตอบโจทย์ใหม่ เช่นโปรแกรมรถบังคับที่หยุดเองเมื่อไม่กดปุ่ม

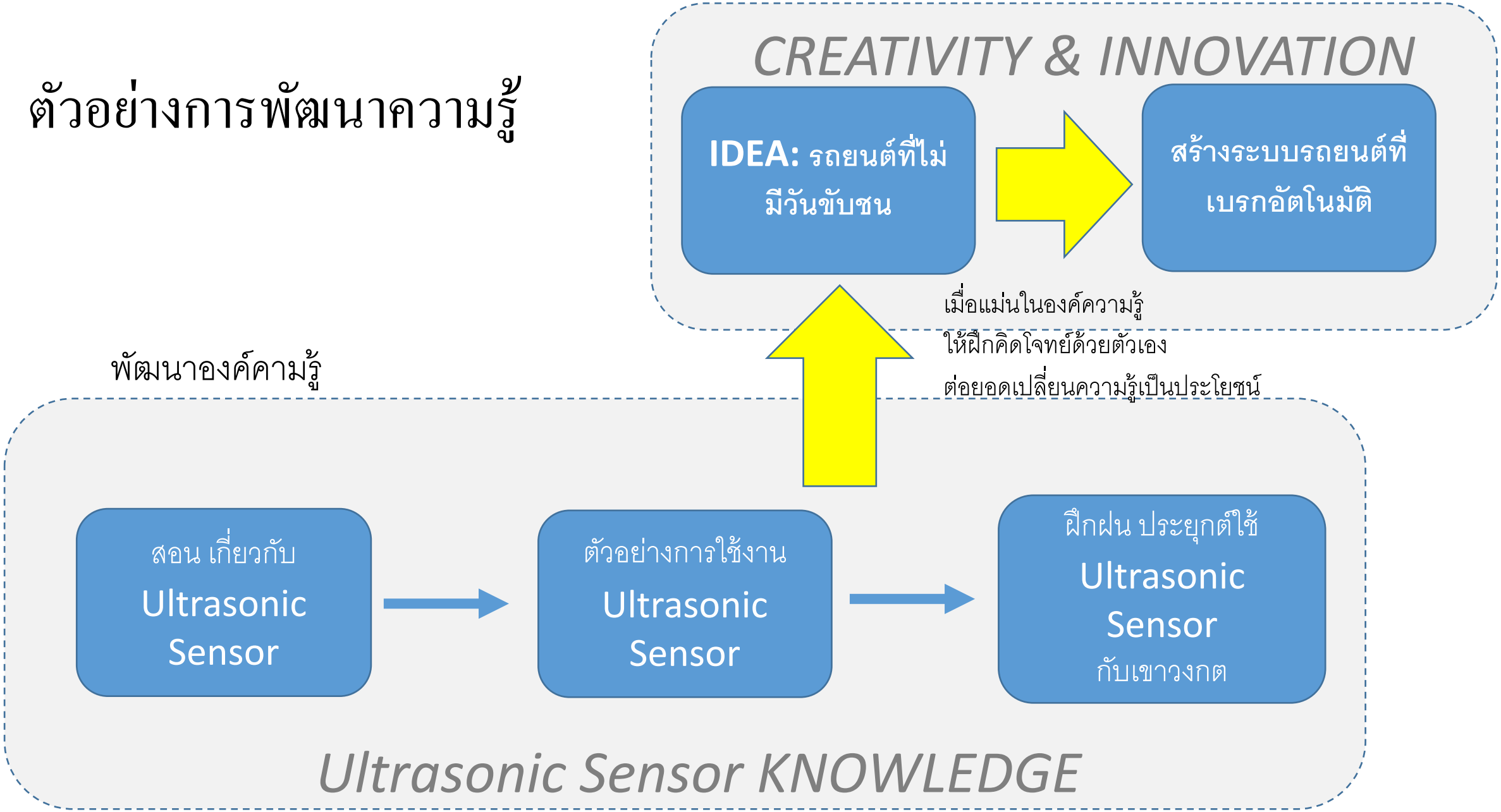
สรุปแบบการเรียนรู้



ตัวอย่างการพัฒนาความรู้

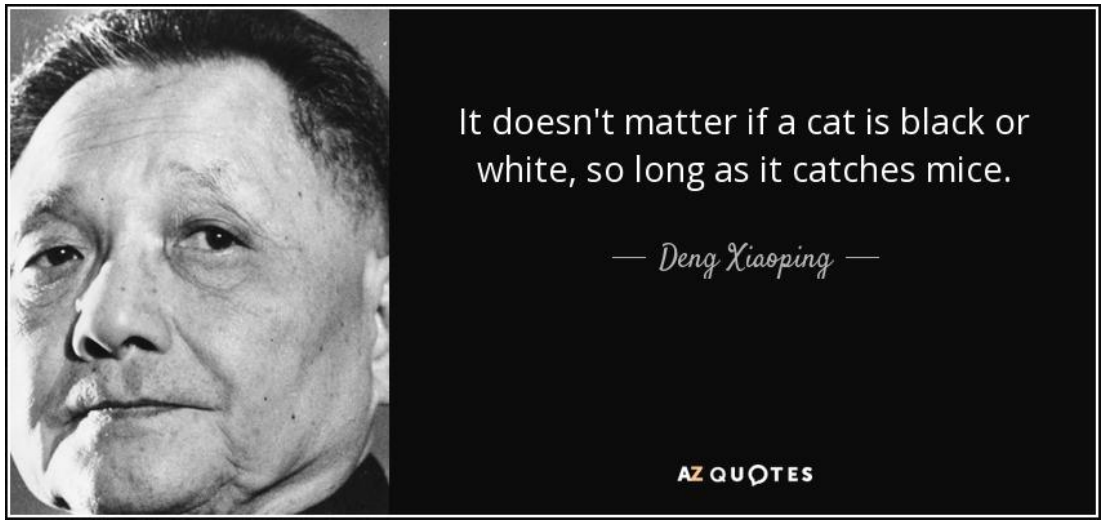


ตัวอย่างการพัฒนาความรู้



ไม่มีคำตอบที่ถูกต้องแบบเดียว

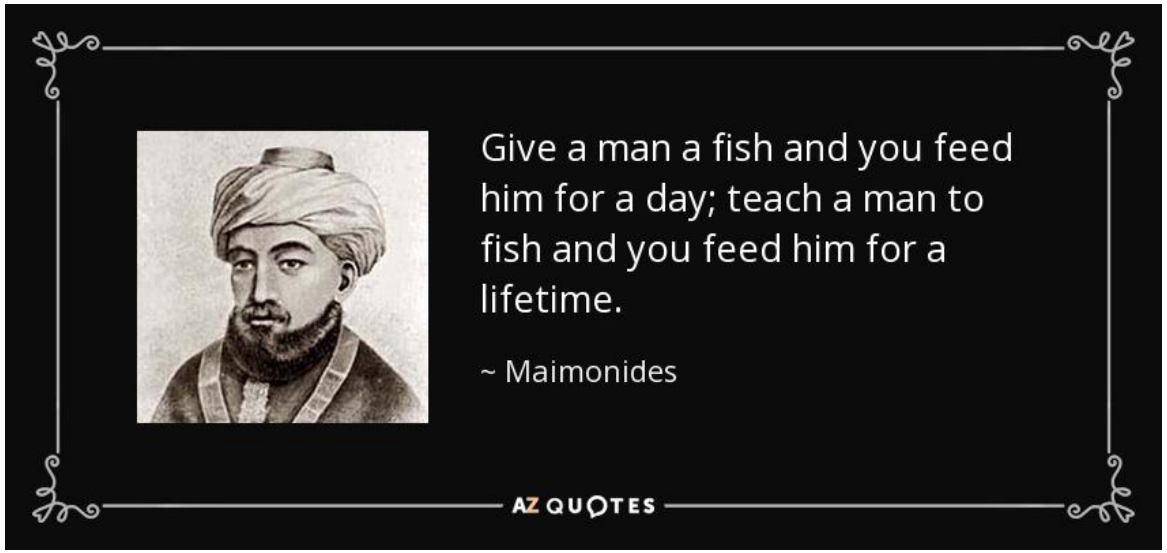
- ไม่มีสูตรสำเร็จสำหรับคำตอบที่ถูกต้อง
- การออกแบบ หุ่นยนต์ เครื่องจักร หรือ โปรแกรม นั้นสามารถทำได้หลากหลายรูปแบบ
- หลายครั้งการที่จะบอกว่าวิธีในดีที่สุดนั้น **วัดได้ยาก** และก็ขึ้นอยู่กับ **กรณีการใช้งาน**
- หลายครั้งการที่จะบอกว่าอะไรดีกว่ากันเป็นเรื่อง **นามธรรม** ที่ขึ้นอยู่กับความชอบของบุคคล
- แต่บางครั้งเราก็สามารถบอกได้ว่า **วิธีนี้ดีที่สุด** เรื่องเหล่านี้เราต้องวิเคราะห์ตามกรณี



“อย่าตีกรอบให้นักเรียนเพราะว่าคำตอบนั้นมีได้หลากหลาย ให้โอกาศนักเรียนค้นหาคำตอบของเขา”

สอนให้คิดเป็น

Key: ถ้าหมั่นให้นักเรียนฝึกคิดและแก้ปัญหา นักเรียนก็จะเป็นผู้มีความสามารถที่จะหาทางออกได้ด้วยตัวเอง



- สุดท้ายแล้วสิ่งที่เราต้องการสอนคือให้นักเรียน **คิดเป็น**
- **เพราะเมื่อนักเรียนคิดเป็นแล้ว ก็จะสามารถเรียนรู้เพิ่มเติมได้ตัวเองได้ทั้งชีวิต**
- การ **คิดเป็น สำคัญกว่า ความรู้**
- ให้เวลากับนักเรียนที่จะ ค้นหา และ คิดค้น คำตอบ นี่คือการฝึกคิดให้เป็น
- เพื่อไม่ให้นักเรียนย่อท้อ ช่วยให้คำตอบก่อนที่นักเรียนจะหมดสนุก