# **Smarthon Documentation**

Release 1.0

**Smarthon Limited** 

Sep 03, 2022

# CONTENTS

1 Tutorial guide

1

#### CHAPTER

ONE

# **TUTORIAL GUIDE**

# **1.1 Smarthon Smart House Kit for micro:bit**

1.1.1 Chapter 1 Know More About Smart Home

What is Smart Home

Advantages of Smart Home

# Steps of Building Smart Home Model





























My Projects View All	
<b>Order</b> New Project	
Create a Project	O
Give your project a name.	
	Create 🗸
🄅 📕 Microso	
🖨 Extensions	
•<₽ Pair device	

### Prepare Micro:bit Programming MakeCode: Add Extensions

Extensions				
https://github.com/stem	hub/pxt-StemhubCity	٩		
	stemhubCity User-provided extension, not endorsed by Microsoft. Learn more			
C Led				
Stemhub:City				
I Radio				
C Loops				
🗙 Logic				
Variables				
🖩 Math				
I2C_LCD1602				

# 1.1.2 Chapter 2 Smart Human Body Induction Lamp

Background

#### Preparation

Learn About Smart Body Induction Lamp

Learn About Human Sensor Module and the Red and Green Light Module

#### Human Body Sensor Module



Fully automatic sensing: high level is output when a person enters its sensing range, and low level is output when the person leaves the sensing range with automatic delay to turn off high level.

Repeatable triggering method: After the induction output is in high level, if a person is sensed in the induction area during the delay time period, the output will remain high level until the person leaves and then the high level will be changed to low level (the induction module will automatically delay a delay time period after detecting each human activity, and the time of the last activity will be the starting point of the delay time HC-SR505). The small human body sensor module has three pins, G for GND ground, V for VCC high level or 5v, S is the signal pin.

### White LED light module





### Installation of Human Body Sensor Light

Program Design





#### **Hardware Connection**



# Sample Program

Makecode program

on start led enable fa	Lse 🔻							
forever								
if Get motio	n (trig	gered	lor n	ot) a	t Pin	P1 -		then
Turn White LED pause (ms) 20	to 10	23 a	it P2					
else								Θ
Turn White LED	to 0	at	P2 -	3.	+	+	+	+
$\odot$								

#### Conclusion

# 1.1.3 Chapter 3 Music Doorbell

Background

#### Preparation

Learn AboutTouch Sensors and Passive Buzzers

#### **Touch Sensor**





#### Passive buzzer module

Power	
 ∩ <	
· · ·	
	Passive Buzzer Module

Buzzer	BLE-UNO Main Board
G	GND
V	VCC · 5V · 3.3V
s	D0-D13

#### Installation of Doorbell





Program Design



**Hardware Connection** 

forever						
if Touch	detec	ted a	t Pin	P1 -		then
play tone	Middl	e C	for	1 <b>-</b> b	eat	+
$\odot$						

Conclusion

## **1.1.4 Chapter 4 Smart Temperature Control Fan**

Background

Preparation

Learn About Temperature And Humidity Sensors And DC Motor Fan Modules

Temperature and Humidity Sensor Module



Temperature and Humidity Sensor	Arduino BLE-UNO	
G	GND	
v	VCC \ 5V	
s	A0-A5	

#### **DC Electric Motor Wind Fan Model**



# 直流电动机工作原理



INA	INB	Power Status
0	0	Release
1	0	Turn
0	1	Reverse
1	1	Stop (Brake)

### Installation of Temperature-Controlled Fan



# Program Design

**Algorithm Design** 



#### **Hardware Connections**

Sensors and Actuators | Main Control Board :- | :- Temperature and Humidity Sensor | P1 DC Motor Fan Module | P15P16



#### Sample Program


## Conclusion

# 1.1.5 Chapter 5 Smart Access Control

Background

Preparation

Learn About Smart Locks

Learn About the Matrix Keyboard Sensor Module



Keyboard Value | Hexadecimal value | Keyboard Value | Hexadecimal value | Keyboard Value | Hexadecimal value :- | :- | :- | :- | :- | :- | :- 1|0xFFFEl7l0xFEFFIDI0x7FFF 2l0xFFFDl8l0xFDFFlCl0xF7FF 3l0xFFFBl9l0xFBFFlBl0xFF7F 4l0xFFFF|\*l0xEFFF|Al0xFFF7 5l0xFFDFl0l0xDFFF| 6l0xFFBF|#l0xBFFF|

## Installation of Smart Access Control

















## **Program Design**

## Algorithm Design

#### Hardware Connections

Sensors and Actuators | Main Control Board :- | :- Matrix Keyboard Sensor Module |P15(SCL)P16(SDO) Servo |P2



## Sample Program





if not ite		is en	npty		and	• <		umber	D	= -	2		then
set password3 🔻	to	item	•										
set number 🔻 to	0 3		+	+									
pause (ms) 200 🔹		+											
clear screen	-												
set item ▼ to	•••												
$\odot$													
if number • - • 3	then						-	· ·					
if password1 • • •	.1.	and -		passwor	rd2 🔻	(	·2·	and <del>•</del>		assword:		• (3)	then
show icon pause (ms) 500 -													
set password1 🕶 to 🎌													
set password2 - to													
set number - to 0													
Turn Servo to 93 degree a	t P2 🔻												
pause (ms) 5000 -													
clear screen	it p2 •												
else	Ċ												Θ
show icon													
set password1 - to													
set password2 🔹 to 🎌													
set password3 ♥ to set number ♥ to 0													



Conclusion

## 1.1.6 Chapter 6 Light-controlled Automatic Windows

Background

Preparation

Learn About the Principle of Light-controlled Automatic Windows

Learn About Photoreceptors and Digital Tube Displays

Photosensitive sensor



Photosensitive sensor	Arduino BLE-UNO
G	GND
v	VCC \ 5V
s	A0-A5

Installation Of Light-controlled Automatic Window

















Program Design



**Hardware Connections** 

# Sample Program

forever							
if Get	light value	(percenta	age) at Pi	in P1 🔻		80	then
Turn Servo	to 91 de	gree at P	2 👻	+	+ +	+	· ·
else							Θ
Turn Servo	to 🕜 de	gree at P	2 🔻				
$\odot$							
show string	Get light	value (per	centage)	at Pin	P1 🔻	+	+ +
pause (ms)	1000 🔻	+ +	+ +	+	+ +		
	-						

#### Conclusion

# 1.1.7 Chapter 7 Smart Rain Control Windows

## Background

Preparation

Principle of Smart Rain Control Window

# Learn About Raindrop Sensor



Raindrop sensor	Arduino BLE-UNO
G	GND
v	VCC \ 5V
A	A0-A5
D	D0-D13



#### Construction of Automatic Smart Window

Program Design



#### **Hardware Connections**

# Sample Program

orever		+	+	+ +	+	+	+	+	+	+	+
if	Get raind	lrop va	lue (p	ercenta	ge) at	Pin	P1 🔻	•	- 8	0	then
Turn Se	ervo to 🤆	91 deg	gree at	P2 🔻	+	+	+	+	+	+	+
else											Θ
Turn Se	ervo to	0 deg	gree at	P2 🔻							
•											
show stri	ng Get	raindro	op valu	ie (perc	entage	e) at	Pin P	1 🔻			
pause (m	5) 1000	•									
		1									

Conclusion

# 1.1.8 Chapter 8 Environmental Monitoring System

Background

Preparation

Learn About Environmental Monitoring Systems

Learn About LCD Displays





Installation of Environmental Monitoring System



### **Program Design**

**Algorithm Design** 



#### Hardware Connections

Sample Program



how string	join Temperature: DHT11 Read temperature - at p	oin P1 •	JΘ	$\odot$	at x	0 y	0
how string	join <sup>*</sup> Humidity: <sup>*</sup> DHT11 Read humidity <b>•</b> at pin P1	•		at x (	0 y	1	
	1000 -		· ·		+	1	
ause (ms)							
ause (ms) ( f DHT	11 Read temperature ▼ at pin P1 ▼ >▼ 27 the	en					
f DHT Fan Contro	11 Read temperature → at pin P1 → > → 27 the 1 On → , Input A P15 → Input B P16 →	en					
ause (ms) ( f DHT Fan Contro	11 Read temperature • at pin P1 • > • 27 the 1 On • , Input A P15 • Input B P16 •	en D					

#### Conclusion

# 1.2 Smart City IoT Starter Kit

# 1.2.1 Know More About Smart City

#### What is a Smart City?

#### **Advantages of Smart City**

- More effective, data-driven decision-making
- Enhanced citizen and government engagement
- Safer communities
- Reduced environmental footprint
- Improved transportation



# 1.2.2 Automated Smart Street Lamp

Goal

Background

What is a smart street lamp?

#### Smart street lamp operation



#### Part List

Assembly step





Step 2





# Step 4








## Hardware connect



## Programming (MakeCode)

### Step 1. Disable micro:bit LED.

- Snap led enable false to on start
- Note that P3 is used as LED in default setting, LED need to be disable

on start	+	
led enable fal	se 🔻	Disable micro:bit LED

### Step 2. Turn on LED by light sensor and obstacle Infrared avoidance sensor

- Drag forever from Basic
- Snap if statement into forever
- Set get light value (percentage) at P1 <40 and get motion (triggered or not) at P0 = true, into if statement that says motion is triggered, someone passes by.
- Then, turn white LED to 1023 at P2 as turning on white LED and pause 10 seconds.
- Else, turn white LED at P2 to 0 as turning off.

orever				+	-	-		+	
if	Get light valu	e (percentage)	at Pin	P1 •		< • (	40	th	en
if	Get motion (tr	iggered or not	) at Pir	n PØ		then		+	
Tur	n White LED to	1023 at P2 -		+	+	+	+		
pau	se (ms) 2000 🔻		+						
else						Θ			
Turr	n White LED to	0 at P2 🔻	+	+	-	+			
$\odot$								-	
else								(	Э
Turn	White LED to 🛛	at P2 🔹	+	+	+	+	÷	+	
$\odot$									

Result

# Think

# 1.2.3 Car Speed Monitoring



Goal

background

What is car speed monitoring?

Car speed monitor operation







## Part List

Assembly step





























## Hardware connect



## Programming (MakeCode)

### Step 1. Initialize OLED screen

- Drag Initialize OLED with width:128, height: 64 to on start
- Set distance1, distance2 and speed to 0 from variables



#### Step 2. Set up function (calculate\_Speed)

- Set up a new function calculate\_Speed from Advanced > Functions.
- Set distance1 to get distance unit cm trig P14 echo P15 (distance from the car to the distance sensor before 0.5 second) Drag Pause to wait 500ms and set distance2 to get distance unit cm trig P14 echo P15 (distance from the car to the distance sensor after 0.5 second)
- By the equation of speed = distance / time. We get the speed of the moving car to (distance1-distance2)/0.5 (unit: cm/s)

function calculate_Speed	a 🔗 🖉 👘	+ +	+	+	+ +	+	+
set distance1 🕶 to	Get distance	unit cm 🔻	trig	P15 🔻	echo	P16 🔻	
pause (ms) 500 💌	+ + +	+ +	+	+	+ +	+	
set distance2 🕶 to	Get distance	unit cm 🕶	trig	P15 🔻	echo	P16 🔻	
set speed 🔹 to dis	stance2 🔹 🕒	• distan	ice1 🔹	) ֥	0.5		+
		+ +		+			

### Step 3. Calculate car speed

- In block forever, call function calculate\_Speed from Advanced > Functions to get the speed of the moving car
- Snap If statement into the loop
- Snap clear OLED display from OLED to avoid overlap
- Snap show string and show value of variables distance1, distance2 and speed



### Step 4. buzzer

- Snap If statement into the loop
- If speed 40, then snap play tone Middle C for 1 beat from music



Result

Think

# 1.2.4 Unloading Alert System



Goal

Background

What is an Unloading alert system?

# Unloading alert system operation



# Part List

# Assembly step











## Hardware connect



## Programming (MakeCode)

### Step 1. Set variable and initialize multi-colour LED

- Inside on start, snap set variable distance to 0 from variables
- Snap set colorLED to color pin setting....
- Snap pause to wait 5 seconds

on sta	irt												
set	distance 🔻	to	0										+
set	colorLED -	to	Color	led	pin s	setting	Blue	P0 -	Red	P1 •	Green	P2 🔻	
Daus	e (ms) 5000	<b>P</b>	+	+	+	+	+	+	+	+	+ +	+	+

### Step 2. Get distance value

- Inside block forever. Set distance to get distance unit cm trig P15 echo P16, that's say get the distance value by connecting the distance sensor to P15 and P16
- Snap if statement into forever, set distance 10 into if statement
- Snap Pause to the loop to wait 1 second for next checking

forever		+	+ +	+	+	+	+	+	+
set distance •	• to Get	distance	unit	cm 🔻	trig	P15 🔻	echo	P16	•
if distant	ce 🔻 🖾 🔻	10	then	+	+	+	+	+	-
		+	a a						
else			Θ						
0									
pause (ms) 100	0 🔹 👘	+	+ +						

### Step 3. Show indicating colours with distance value

• If distance 10, then strip show color green, else strip show color red



Result
Think



# 1.2.5 Smart Car Park Access Barrier 1: Car Park Monitoring System

Goal

Background

What is an Smart car park monitoring systems?



## Smart car park monitoring systems operation

### Part List

Assembly step























ſ	G1

#### Hardware connect

#### Programming (MakeCode)

#### Step 1. Set variables and initialize multi-colour LED and OLED screen

- Drag Initialize OLED with width:128, height: 64 to on start
- Inside on start, snap set variable distance to 0 and set number to 0 from variables.
- Snap set colorLED to color led pin setting ... set colorLED to color led pin setting ...

on sta	rt												
init	ialize OLED	with	width	128	heigh	<b>t</b> 64							
set	distance 💌	to	0	+	+	+							
set	colorLED 🔻	to	Color	led p	in set	ting	Blue	P0 🔻	Red	P1 -	Green	P2 🔻	
set	number 🔻	to 🕜		*	+	+	+	+	+	+	+ +		-
			-										

#### Step 2. Get distance

• Drag get distance to distance unit cm trig P15 echo P16, store the value to variable distance.

foreve	r									
set	distance 🔻	to	Get	distance	unit	cm 🔻	trig	P15 🔻	echo	P16 🔻

#### Step 3. Show indicating colours and count the number of vacancies

- Snap if statement into forever, set variable distance > 10
- If distance >10, then colorLED shows color green, else colorLED shows color red
- Snap change number by 1 if distance>10



#### Step 4 display on OLED

- Snap clear OLED display from OLED to avoid overlap
- Snap show number and show value of variables number
- Snap Pause to the loop to wait 1 second for next checking
- Reset number to 0 before next checking



Result

Think

# 1.2.6 Smart Car Park Access Barrier 2: Car Park Access Barrier



## Goal



## Background

What is a smart car park access barrier?

## Car park access barrier operation



### Part List

Assembly step

















### Hardware connect

### Programming (MakeCode)

#### Step 1. Set variables, initialize OLED screen and servo at start position

- Drag Initialize OLED with width:128, height: 64 to on start
- Inside on start, snap set variable distance1 to 0, set variable distance2 to 0 and set number to 0 from variables.
- Snap set colorLED to color led pin setting
- Snap Turn Servo to 0 degree at P0.
- Snap pause to wait 5 seconds

on start	+ +											
initialize OLED	with widt	h 12	B he	ight (	64							
set distance1	🔹 to 🧕		+	+	+	+						
set distance2	🔹 to 🧕											
set number 🔻	to 🕜	+										
set colorLED •	to Col	or led	l pin	setti	ng Bl	lue Pi	∍ •	Red	P1 🔻	Green	P2 -	•
Turn Servo to	0 degree	at	P0 🔻	+	÷	+	+	+	+	+	+	-
pause (ms) 500	•	-		+								

### Step 2. Get distance

• Drag set distance1 to distance unit cm trig P15 echo P16, store the value to variable distance1.

forev	rer		+		-				-	+ +
set	distance1	• to	Get d	listance	unit	cm 🔻	trig	P15 🔻	echo	P16 🔻

### Step 3. Show indicating colours and count the number of vacancies

- Snap if statement into forever, set variable distance1 > 10
- If distance1 >10, then colorLED shows color green, else colorLED shows color red
- Snap change number by 1 if distance1>10



#### Step 4 display on OLED

- Snap clear OLED display from OLED to avoid overlap
- Snap show number and show value of variables number



#### Step 5. Open/close gate with distance value

- Snap if statement into forever, set variable number>0
- Drag get distance2 to distance unit cm trig P8 echo P12, store the value to variable distance2.
- Snap if statement into forever, set variable distance 2 < 10
- Snap Pause to the loop to wait 1 second for next checking
- Reset number to 0 before next checking

if	number 👻	> •	0		then						
set	distance2 🕶	to	Get	dista	nce unit	t cm 🔻	trig	P8 💌	echo	P12 🔻	
if	distance	D	< •	10	the	n 	+			+	
$\odot$					+	+					
pause	(ms) 1000 •										
set	number 🗸 to										

## Step 6. Set servo position

- Snap Turn Servo to 90 degree at P3 as the gate is opened.
- Snap pause to the loop to wait 5 seconds
- Snap Turn Servo to 0 degree at P3 as the gate is closed.

	+			
Turn Servo to 90 degree at P3 •				
pause (ms) 5000 💌				
Turn Servo to 0 degree at P3 -				

Result

Think

1.2.7 Broken Car



Goal

Background

What is a smart traffic system ?

## Smart traffic system Operation



### Part List

Assembly step













## Hardware connect



## Programming (MakeCode)

## Sender

## Step 1. Set radio set group at start position

• Drag radio set group 6 to on start



## Step 2. Get light and distance value

- Snap if statement into forever, set get light value (percentage) at pin P4 < 40
- If get light value (percentage) at pin P4 < 40, and else if get distance unit cm trig P15 echo 16 < 20

forever			-	-		-		-				
if	Get 1	ight val	lue (pe	rcenta	ge) at	: Pin	P4 -		< -	40	then	+
if	Get	distanc	e unit	cm 🔻	trig	P15	•	echo	P16 •			then
else											Θ	
												+
			+	-		-	+	+	+	+		

### Step 3. Control the car by sending radio number

- Drag radio send number to 2 into if
- Drag radio send number to 0 into else

forever			+	-		÷	-		+	+	-	-+-		
if	Get li	ght value	e (per	centa	ge) at	Pin	P4 -		< •	40	the	n		
if	Get a	listance	unit	cm 🕈	trig	P15	• •	echo	P16		•	20	the	en
radi	o send i	number (	2											
$\odot$														
else											6	Э		
radio	send nu	mber 🧿												
$\odot$														

### Receiver

#### Step 1. Set radio set group at start position

- Drag radio set group 6 to on start
- Initially, the car moves forward by default

on star	t						
radio	set gr	oup 6					
Motor	M1 🔻	speed	150	M2	• s	peed	150
Motor	мз 👻	speed	150	M4 -	• s	peed	150
			+	+	+	-	+

## Step 2. Control car by receiving different number

- Snap if statement into on radio received receivedNumber
- Set receivedNumber =2 and make the car turn left
- Set receivedNumber=0 and make the car move forward

on radio received receivedNumber
if receivedNumber = • 2 then +
Motor M1 - speed 150 M2 - speed 0
Motor M3 • speed 0 M4 • speed 150
if receivedNumber = • 0 then
Motor M1 - speed 150 M2 - speed 150
Motor M3 • speed 150 M4 • speed 150

Result

Think

# 1.2.8 Crosswalk



## Goal



## Background

What is a smart crosswalk system?

## Smart crosswalk system operation



### Part List

Assembly step


















### Hardware connect



# Programming (MakeCode)

Sender:

## Step 1. Set radio set group at start position

• Drag radio set group 6 to on start



#### Step 2. When trigger a motion

- Snap if statement into forever, set get motion (triggered or not) at pin P4
- Snap pause to the loop to wait for 0.1 second for next checking

orever										
if	Get n	notion	(trig	gered	or n	ot) a	t Pin	P4 🔻		then
	+		+	-	-	-	-		-	-
pause	(ms) (	100 🔻								

#### Step 3. Keep the status for 5 second

- set variable second to 50
- While second > 0, snap pause to 0.1 second and change second by -1.



### Step 4. Get distance value

• Snap if statement into while loop, set get distance unit cm trig P15 echo P16 < 20

<pre>hile second ▼ &gt; ▼ 0 lo if Get distance unit cm ▼ trig P15 ▼ echo P16 ▼ &lt; ▼ 20 t  f pause (ms) 100 ▼ change second ▼ by -1</pre>	while second $\checkmark$ $\checkmark$ $\circ$ $\circ$ do if Get distance unit cm $\sim$ trig P15 $\checkmark$ echo P16 $\checkmark$ ( $\checkmark$ 20) $\bigcirc$ pause (ms) 100 $\checkmark$ change second $\checkmark$ by $-1$	set second •	to 50		+	-	+	4	-					
<pre>do if Get distance unit cm ▼ trig P15 ▼ echo P16 ▼ &lt; ♥ 20 1</pre>	do if Get distance unit cm ♥ trig P15 ♥ echo P16 ♥ < ♥ 20 pause (ms) 100 ♥ change second ♥ by -1	while secon	nd 🔻 🔉 🔻	0										
Depause (ms) 100 ▼         A the second se	Pause (ms) 100 ▼     change second ▼ by -1	do if Ge	t distance	unit	cm 🔻	trig	P15	• e	cho P	16 🔻	<	•	0	t
pause (ms) 100 ▼ change second ▼ by -1	pause (ms) 100 ▼ change second ▼ by -1	$\odot$												
change second - by -1	change second • by -1	pause (ms)	100 🔻											
		change sec	ond 🔻 by (	-1										
							+-		-					
		pause (ms)	100 • cond • by (	-1	-		+++++++++++++++++++++++++++++++++++++++	-	-					

## Step 5. Control the car by sending radio number

• Drag radio send number to 1 into if

forever	÷ ÷							
if Get motio	on (triggered	or not) a	t Pin	P4 🔻	the			
set second -	to 50	+ +	+	+	+ +			
while seco	and 💌 💌	0						
do if G	et distance u	nit cm 🕶	trig	P15	🔹 echo	P16 🔻	20	then
radio se	nd number 🚺			+				
•								
pause (ms)	100 🔻							
change se	cond 👻 by 🕒	1						
radio send num	ıber 😐	+ +						
•								
pause (ms) 100	•	+ +	+	+		+		
	+ +							

#### Receiver

## Step 1. Set radio set group at start position

- Drag radio set group 6 to on start
- Initially, the car moves forward by default

n start	:						
radio	set gr	oup 6					
Motor	M1 🔻	speed	150	M2	•	speed	150
Motor	мз 👻	speed	150	M4	•	speed	150
			+	+	+	-	+

## Step 2. Control car by receiving different number

- Snap if statement into on radio received receivedNumber
- Set receivedNumber =1 and make the car stop
- Set receivedNumber=0 and make the car move forward



Result

## Think

# **1.2.9 Smart Traffic Lights**



Goal

Background

What is a smart traffic light?

### Smart traffic light operation



## Part List

## Assembly step









#### Hardware connect



### Programming (MakeCode)

### Sender

### Step 1. Set up a new function (TurnRed)

- Snap pause to wait 1 second
- Control traffic light yellow on
- Snap pause to wait 1 second

• Control traffic light red on

	÷ +									
function TurnRed										
pause (ms) 1000 🔻										
trafficLight •	Control	traffic	: light Re	d OFF	Ye	11ow	ON	Gree	n OFF	
pause (ms) 1000 🔻					+		+	+		
trafficLight •	Control	traffi	: light Re	d 🧹 🔮	N Ye	11ow 🧹	)FF	Gree	n OFF	
					+		+	+		-

#### Step 2. Set up a new function (TurnGreen)

- Snap pause to wait 1 second
- Control traffic light yellow on
- Snap pause to wait 1 second
- Control traffic light green on

	1.1																
function	n TurnGree	• 📀															
pause	(ms) 1000	•		+	+	+	+	+	+	+	+	+	+	+	+	+	-
	trafficLig	ht 🔹	Contr	rol t	raffic	light	t Red	OFF		Yel	low		ON	Green	OFF		
pause	(ms) 1000	•	÷	÷	+	÷	+	+	÷	÷	÷	+	÷	+	+	+	
	trafficLig	ht •	Contr	rol t	raffic	light	t Red	OFF		Yel	low (	OFF		Green		ON	
				+	+	+	+	+	-		-		+	+	+	+	

#### Step 3. Disable micro:bit LED.

- Snap led enable false to on start
- Note that P3 is used as LED in default setting, LED need to be disable



## Step 4. Initialize the program similar as last lesson

- Drag set variable trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 to on start
- Control traffic light green on
- Drag radio set group 6 to on start

on star	rt	-													+
set	trafficLig	ht •	to	Traffic	: lig	nt pin	set	ting R	ed F	P0 🔻	Yellow	P1 -	Green	P2 🔻	
	trafficLi	ght 🔻	Con	trol tr	affic	ligh	t Red	011		Ye	11ow 🤇	OFF	Gree	n	ON
radio	set group	6					1	+					+ +		1
-															

forever	+ +							
if Get motio	on (triggered	or not) a	t Pin	P4 👻	the	n - **		
set second -	to 50	+ +	+	+	+			
while seco	and 💌 🔊 🔻	0						
do if G	et distance u	unit cm 🕶	trig	P15	▼ echo	P16 -	20	then
radio se	nd number 🚺							
•								
pause (ms)	100 🔻							
change se	cond 👻 by 🤇	-1						
radio send num	ıber 🕜	+ +						
pause (ms) 100	•	+ +	+	+	+			
	* *							

## Step 5. Call function

- Snap function TurnRed into if get motion (triggered or not) at pin P4 case
- Drag function TurnGreen after the while loop



#### Receiver

#### Step 1. Set radio set group at start position

- Drag radio set group 6 to on start
- Initially, the car moves forward by default

n start	:						
radio	set gr	oup 6					
Motor	M1 🔻	speed	150	M2	•	speed	150
Motor	мз 👻	speed	150	M4	•	speed	150
			+	+	+	-	+

#### Step 2. Control car by receiving different number

- Snap if statement into on radio received receivedNumber
- Set receivedNumber =1 and make the car stop
- Set receivedNumber=0 and make the car move forward



Result

## Think

# 1.2.10 Smart Pedestrian Lights 2



Goal

Background

What is a smart pedestrian light?

### Smart traffic light operation



#### Part List

Assembly step

















#### Hardware connect

Programming (MakeCode)

#### **Traffic light 1**

### Step 1. Set up a new function (TurnRed)

- Snap pause to wait 1 second
- Control traffic light yellow on
- Snap pause to wait 1 second
- Control traffic light red on

function TurnRed 🚫								
pause (ms) 1000 V								
trafficLight •	Control	traffic	light Red	OFF	Yello	 ON Gr	een OFF	
pause (ms) 1000 🔻								
trafficLight •	Control	traffic	light Red		> Yello	Gr	een OFF	

#### Step 2. Set up a new function (TurnGreen)

- Snap pause to wait 1 second
- Control traffic light yellow on
- Snap pause to wait 1 second
- Control traffic light green on

function TurnGreen	<b>)</b>														
pause (ms) 1000 🔻		+	+	+	+	+	-	+	-	+	+	+	+	+	
trafficLight •	Con	trol	traffic	ligh	ıt Red	07		Yel	11ом		ON	Green	0	F	
pause (ms) 1000 🔻	+	+	+	+	+	+	÷	+	+	÷	÷	+	÷	+	-
trafficLight •	Con	trol	traffic	ligh	nt Red	07		Yel	110w	OFF		Green		ON	
			+	-		-			-			+	÷	+	-

### Step 3. Initialize the program

- Drag set variable trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 to on start
- Control traffic light green on
- Drag radio set group 6 to on start
- In forever, snap function TurnRed into if get motion (triggered or not) at pin P4 case
- Drag function TurnGreen after the while loop

on sta	art	+	+	+	+	+	+	+	+	+	-	+	+	+	+	+	
set	trafficLig	ht •	to	Traffi	c lig	nt pir	ı seti	ting R	ed F	•0 •	Yello	91 ·	• Gr	een	P2 🔻		
	trafficLig	ght 🔻	Con	trol t	raffic	ligh	rt Red	011		Ye	llow 🤞	OFF		ireen		ON	
radi	io set group	6						+						+	*		



## Step 4. Control traffic light 2 by sending radio string

- Drag radio send string "R" before TurnGreen
- Drag radio send string "G" after TurnRed


## **Traffic light 2**

### Step 1. Set up new functions

unctio	n TurnRed 🔿															
pause	(ms) 1000 •															
	trafficLight 🔻	Cont	trol	traffic	light	Red				Yellow		ON	Green			
pause	(ms) 1000 💌															
	trafficLight -	Cont	trol	traffic	light	Red		ON		Yellow	•		Green	•		
pause	(ms) 1000 -															
unctio	n TurnGreen	0														
pause	(ms) 1000 -															
	trafficLight •	Cont	trol	traffic	light	t Red	05			Yellow			Gree	• <	177	
pause	(ms) 1000 💌								-							
	trafficLight •	Con	trol	traffic	light	t Red			Þ	Yellow	•	Ŧ	Green	• <	0	r)
pause	(ms) 1000 ¥															

#### Step 2. Initialize the program

- Drag set variable trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 to on start
- Drag radio set group 6 to on start
- Control traffic light green on
- Set a variable green=0

on start												
radio set group   6								-			÷	
set trafficLight 🔻	to	Traffi	ic light	t pin s	etting	Red P	9 🔻	Yellow	P1 •	Green	P2 🔻	
trafficLight •	Con	trol t	raffic	light (	Red			1	155		1	
						0.0	a ter	TOM		Greet	1	
set green 🔻 to 🛛	T			+		Crit	e tel	104		Greet		

## Step 3. Control traffic light by receiving different number

- Snap if statement into on radio received receivedString
- Set receivedString ="R" and call TurnRed
- Set receivedString ="G" and call TurnGreen
- Change variable green depend on the light



#### Step 4. Play sound effect depend on the light status

- Snap if statement into forever
- Play melody with different tempo

ſ	forever					
	if gree		0	then		
	play melody	J 1000		at tempo	500	(bpm)
+	else			Θ	+	+
	play melody	, <b>,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,</b>		at tempo	999	(bpm)
+- 	•	+ +	+	+ +	+	+

## Receiver

## Step 1. Set radio set group at start position

- Drag radio set group 6 to on start
- Initially, the car moves forward by default

n start	£						
radio	set gr	oup 6					
Motor	M1 🔻	speed	150	M2 -	• •	peed	150
Motor	МЗ 🔻	speed	150	M4 -	• s	peed	150
			+	+	+	+	+

## Step 2. Control car by receiving different number

- Snap if statement into on radio received receivedNumber
- Set receivedNumber =1 and make the car stop
- Set receivedNumber=0 and make the car move forward

on radio received receivedNumber
if receivedNumber = 1 then
Motor M1 - speed 0 M2 - speed 0
Motor M3 🔻 speed 📀 M4 🔻 speed 🥥
if receivedNumber = • 0 then
Motor M1 <b>v</b> speed 150 M2 <b>v</b> speed 150
Motor M3 <b>v</b> speed 150 M4 <b>v</b> speed 150

Result

Think

# 1.2.11 Intersection



Goal

Background

What is a smart traffic light?

### Smart traffic light operation





## Part List

Assembly step





















#### Hardware connect

Programming (MakeCode)

#### **Traffic light 1**

#### Step 1. Set up a new function (TurnRed)

- Snap pause to wait 1 second
- Control traffic light yellow on
- Snap pause to wait 1 second
- Control traffic light red on

function TurnRed 🚫										
pause (ms) 1000 🔻										
trafficLight •	Control	traffic	light Red	OFF	Yell	•	ON	Green	OFF	
pause (ms) 1000 🔻	÷ ÷	+ +		+	+ +		+			+
trafficLight •	Control	traffic	light Red		🕨 Yell	•	P	Green	OFF	

#### Step 2. Set up a new function (TurnGreen)

- Snap pause to wait 1 second
- Control traffic light yellow on
- Snap pause to wait 1 second
- Control traffic light green on

function TurnGreen	a												
pause (ms) 1000 🔻	+ +	+	+	+	+	+	+ +	+	+	+	+	+	÷
trafficLight -	Contro	l traffic	light	Red	OFF		Yello	•	ON	Green	OFF		•
pause (ms) 1000 🔻	+ +	+	+	÷	÷	÷	+ +	+	÷	+	+	+	
trafficLight •	Contro]	l traffic	light	Red	077		Yello			Green		ON	
	÷	+	+	÷	+	÷				+	+	÷	

#### Step 3. Initialize the program

- Drag set variable trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 to on start
- Drag radio set group 10 to on start
- Control traffic light red on
- Pause for 5s

on sta	art				+	+ +	+	+	+	-	+ +	+	
set	trafficLight -	to 🔽	raffic	light	pin se	etting	Red P0	- Y	ellow	P1 -	Green	P2 🔻	а. — н
radi	io set group 10		+	-	+	+ +	+	÷	+	+	+ +	+	+
	trafficLight -	Contr	ol tra		ldaha n	-			1				
		Conte	UL CIA	TTIC 1	LIGHT R	ea	UN	Yell			Green	1 OFF	
paus	se (ms) 5000 🔻	Conci	+		Hight K	ea		Yell	ы 🦿		Green	n OFF	

#### Step 4. Keep the green light status for 5 second

- Call function TurnGreen
- Pause 1.5s before TurnGreen
- Pause 3.5s after TurnGreen



#### Step 5. Keep the red light status for 5 second

- Call function TurnRed
- set variable second to 50
- While second > 0, snap pause to 0.1 second and change second by -1.



### Step 6. Get distance value

call Turn	Red at to	50											
while	second •	30	0										
do if	Get di	stance (	unit	cm •	trig	P15	• ec	ho P	16 🕶	<	- 6	20	the
	+	+ +	+	+	-	-	-	+	-	+	-		-
pause	(ms) 10	••		+						+	-	+	
chang	e second	🔹 by (	-1										

• Snap if statement into while loop, set get distance unit cm trig P15 echo P16 < 20

### Step 7. Control traffic light 2 and car by sending radio number

- Drag radio send number 1 into if
- Drag radio send number 0 after TurnGreen
- Drag radio send string "R" before TurnGreen
- Drag radio send string "G" before TurnRed



**Traffic light 2** 

nction TurnRed 🔗											
pause (ms) 1000 🔻											
trafficLight •	Control	traffic	light	Red		×	ellow		ON	Green	•
pause (ms) 1000 💌											
trafficLight -	Control	traffic	light	Red	0	) v	ellow	•		Green	
pause (ms) 1000 -	· ·										
nction TurnGreen 📀											
pause (ms) 1000 🔻	a										
trafficLight •	Control	traffic	light	Red		Y	ellow		ON D	Green	011
pause (ms) 1000 💌											
trafficLight •	Control	traffic	light	Red		D Y	ellow	077		Green	
pause (ms) 1000 🔻											
	1.1										

## Step 1. Set up new functions

## Step 2. Initialize the program

- Drag set variable trafficLight to Traffic light pin setting Red P0 Yellow P1 Green P2 to on start
- Drag radio set group 10 to on start
- Control traffic light green on
- Pause for 5s

on star	t																
set	trafficLig	it •	to	Traff	fic li	ght p	oin se	tting	; Red	P0 •	Yell	ow	P1 -	Green	P2	•	
radio	set group	10		÷	+	+	+	+	+	+	+	+	+	+	+	+	
	trafficLig	ht 🔻	Cor	trol	traff	ic li	ght R	ed 🤇	OFF	<b>Y</b> e	llow	•••		Gree	n	0	
pause	(ms) 5000	••	÷	+	+	+	+	+	+	+	+	+	+	+	+	+	-
		1															

#### Step 3. Control traffic light and car by receiving different number

- Snap if statement into on radio received receivedString
- Set receivedString ="R" and call TurnRed
- Set receivedString ="G" and call TurnGreen



### Step 4. Keep the red light status for 5 second

- After TurnRed, set variable second to 50
- While second > 0, snap pause to 0.1 second and change second by -1.
- Before TurnGreen, pause 1.5s
- After TurnGreen, pause 3.5s



## Step 5. Get distance value and control the car

- Snap if statement into while loop, set get distance unit cm trig P15 echo P16 < 20
- Drag radio send number 3 into if 1 if

on radio received receivedString	-									
if receivedString = •	e"	then								
call TurnRed	1									
set second - to 50										
while second • > • 0										
do if Get distance unit c		trig	P15	• ec	ho P	16 🕶	••	20	then	ľ.
radio send number 3										
pause (ms) 100 💌										
change second ▼ by -1										
<b>⊙</b>			r.							
if receivedString = - Co	i')	then								
pause (ms) 1500 🗸										
call TurnGreen										
radio send number 2										
pause (ms) 3500 •										
$\odot$			Ľ.							

## Car 1:

### Step 1. Set radio set group at start position

• Drag radio set group 10 to on start



### Step 2. Control car by receiving different number

- Snap if statement into on radio received receivedNumber
- Set receivedNumber =1 and make the car stop
- Set receivedNumber=0 and make the car move forward

on radio received receivedNumber		
if	receivedNumber = 1 then	
Motor	M1 • speed 0 M2 • speed 0	
Motor	M3 🔻 speed 0 M4 🔻 speed 0	
$\odot$		
if 🧹	receivedNumber =	
Motor	M1 - speed 150 M2 - speed 150	
Motor	M3 <b>v</b> speed 150 M4 <b>v</b> speed 150	
$\odot$		

Car 2:

## Step 1. Set radio set group at start position

• Drag radio set group 10 to on start



### Step 2. Control car by receiving different number

- Snap if statement into on radio received receivedNumber
- Set receivedNumber =3 and make the car stop
- Set receivedNumber=2 and make the car move forward

on radio received receivedNumber		
if receivedNumber = - 3 then +		
Motor M1 - speed 0 M2 - speed 0		
Motor M3 <b>v</b> speed <b>0</b> M4 <b>v</b> speed <b>0</b>		
if receivedNumber = • 2 then +		
Motor M1 - speed 150 M2 - speed 150		
Motor M3 - speed 150 M4 - speed 150		

## Result

Think

## 1.2.12 Intelligent Road Signs

Goal

Background

What is intelligent road sign?

## Intelligent road sign operation





#### Part List

Assembly step



















### Hardware connect

Programming (MakeCode)

## **Install library**

- Click extensions
- Enter StemhubCity

Search Q	
Basic	
Input	
Music	
Led	
B Stemhub:City	
III Radio	
C Loops	stembubcity
🔀 Logic	Lights and Display Software 3156 Robotics Gaming Networking   Image: Software Networking Image: Software Networking   Image: Software Software Software Networking   Image: Software Software Software Networking   Image: Software Software Software Software   Image: Software Software Software Software
<b>≣</b> Variables	
🖬 Math	
I2C_LCD1602	
😧 Neopixel	
Extensions	

## **Update Time**

- Pause 1 second
- Then increase time by 1
- Set time to 0 when time is equal to 25



#### **Initialize Time**

• set time to 0



### Checking time and changing road signs

- Call function TimeChange to update the time
- Drag out two turn servo to 0 degree at P0 from the stemhubcity library
- Then place them inside the if condition and else condition
- Change turn servo to 0 degree at P0 as turn servo to 0 degree at P3 and turn servo to 180 degree at P3

Search Q	Stemhub:City
Basic	
Input	Turn White LED to 0 at P0 ▼
Music	Turn Servo to 0 degree at P0 -
Led	
_ 👖 Stemhub:City	Get noise level (dB) at Pin P0 ▼
••• SmartCity	
••• SmartHome	


Result

### Think

# 1.3 Micro: bit M1 SMART CAR\_Beginner

## 1.3.1 Lesson 1



### Introduction

### **Teaching Objectives**

### Understand micro: bit expansion Circuit Board



#### **Understand the Mecanum Wheel Car**

- 4 Motors
- 4 Mecanum Wheels
- Micro bit
- Micro bit Expansion Circuit Board



• Ultrasonic Sensor



• Infrared Line Follower



. micro:bit		🕋 Home		•	Microsoft
	<b>1</b>	show Leds		2)	
My Projects View All					± Import
New Project	new car t	test 6 days ago	new	1ed 20 1	2 days ago
Tutorials					
New? Start Here!	Name Tan S	miley Buttons	all on the second	Love Meter	
Google Play	Search				٩
88 Apps	Categories 🗸 Hom	ne Top charts New	v releases		
My apps					
C Games Kids Editors' Choice	C micro:bit	Micro:bit Micro:bit Education	t nal Foundation Educatio	on ★★★	± ± 1,013 ≛
Account Payment methods My subscriptions		Add to Wishli	st		Install

## Preparing Micro: bit Programming Software Makecode

Ś	Mac	iPad	iPhone	Watch	тν	Music	Support	Q	Ô
App S	tore Previev	V							
			This app is av	ailable only on the	App Store for if	hone and iPad.			
m	; icro:t	bit	micro:bit ( Micro:bit Educa	44) Itional Foundat	ion				

### **New Extensions**





## Connect micro: bit to the computer

Pair device for one-click downloads	G
Connect the micro:bit to your computer with a USB cable Use the micro:USB port on the top of the micro:bit	Pair your micro:bit Click 'Pair device' below and select BBC micro:bit CMSIS-DAP or DAPLink CMSIS- DAP from the list
	? Pair device +€+
makecode.microbit.org wants to connect	
"BBC micro:bit CMSIS-DAP"	



### Save program file.hex) to micro: bit





understand Block Programming

### PAUSE Block module:



### **IF Logic Block Module:**



- "else if" can be used to set additional condition
- "else" is used to set the action to be taken when condition of both "if" and "else if" did not fulfilled.



### Comparison block module:



Mathematical operation block module:



### "AND,OR" block module



" Variable"



## 1.3.2 Lesson 2



Introduction

**Teaching Objectives** 

Control of RGB light (LED) on expansion board





Exercise 1: design a program to flash the LED on the expansion board in order.



Exercise 2: design a program to change the brightness of the LED on expansion board start from dim to bright and then bright to dim. Breathing light effect





### Buzzer



## Delay of the word display



Exercise: Play a song, design a program to play the song below by your car

1	1	5	5	5	5	5
G	G	D	D	E	E	D
5	5	1	1	1	1	1
Twin	- kle	twin -	kle	lit -	tle	star
4	4	3	3	2	2	1
C	C	B	B	A	A	G
2	2	3	3	4	4	5
how	I	won -	der	what	you	are
5	5	4	4	3	3	2
D	D	C	C	B	B	Α
ĩ	1	2	2	3	3	4
up	a -	bove	the	world	50	high
5	5	4	4	3	3	2
D	D	C	C	B	B	A
1	ĩ	2	2	3	3	4
like	a	dia -	mond	in	the	sky
1	1	5	5	5	5	5
G	G	D	D	E	E	D
5	5	ĩ	ĩ	ĩ	ĩ	ĩ
Twin	- kle	twin -	kle	lit -	tle	star
	1 G 5 Twin 4 C 2 how 5 D 1 up 5 D 1 like 1 G 5 Twin	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	1       1       5       5       5       5         G       G       D       D       E       E         5       5       1       1       1       1         Twin-kle       twin-kle       lit-       tle         4       4       3       3       2       2         C       C       B       B       A       A         2       2       3       3       4       4         how       I       won-       der       what       you         5       5       4       4       3       3         D       D       C       C       B       B         1       1       2       2       3       3         up       a-       bove       the       world       so         5       5       4       4       3       3         up       a-       bove       the       world       so         5       5       4       4       3       3       like         1       1       2       2       3       3       like         1



James Pierpont Arranged by Julie A. Lind

**Right hand only** 

E E E - E E E - E G C D E - - -Jin- gle bells, jin - gle bells, jin - gle all the way.

F F F F F E E EE Oh what fun it is to ride in a

E D D E D-Gone horse op - en sleigh

E E E - E E E - E G C D E - - - Jin- gle bells, jin-gle bells, jin-gle all the way.

F F F F F E E EE Oh what fun it is to ride in a

G G F D C one horse op - en sleigh!

Copyright © 2011 Julie A. Lind www.PianoSongDownload.com

#### Answers

### Exercise 1:



### **Exercise 2: Breathing light**

a.	Changing	the	color	of	all	the	light	into	orangebrightness	of	the	light	will
⇔change	as 'value'	cha	inge.										
b.	Stop for	0.01	seco	nd									
с.	'value' A	DD	5										

a. Changing the color of all the light into orangebrightness of the light will\_ →change as' value' change. b. stop for 0.01 second c. 'value' SUBTRACT 5 d. Program will end when' value' drop to 0.

rever	· · · · ·								
set value 🔻	to 🕘								
while valu	e 🔹 🗹	255							
do On-board L	ight all	- cole	or or	ange	• br	ightn	ess	value	•
pause (ms)	10 🔻								
change val	lue 🔻 by	5							
set value 🔻	to 255								
while valu	e 🔹 🔰	0							
do On-board L	ight all	- cole	or or	ange	✓ br	ightn	ess	value	
pause (ms)	10 🔻								
change val	lue 🔻 by	0	- 🔻	5					

## 1.3.3 Lesson 3



Introduction

**Teaching Objectives** 

## Understanding the Mecanum wheels



## Principal of the wheels







## 1.3.4 Lesson 4

### **Movement of Car**



### Block module that control motor



### Forward

on start	:	+					
Motor	M1 🔻	speed	100	M2 🔻	speed	100	
Motor	МЗ 🔻	speed	100	M4 🔻	speed	100	microbitcar
		+	+	+	+	+ +	

### Backward



### Turn to the left



Turn to the right



### Spinning in anti-clockwise direction



## Spinning in clockwise direction



### Conclusion

### Exercise1: Horizontal movement of the car





## Exercise 2: Diagonal movement of the car

Exercise 3: Different ways to drive the car

Moving around in a square:



## Moving around in a circle:



## M shape route driving



### Answer

Exercise 1:





### Exercise 2:

Moving toward right upper corner:



Moving toward left upper corner:



### Moving toward right lower corner:

forever	+				1000
Motor M1 - sp	eed 0	M2 🔻 sp	eed	-60	
Motor M3 🕶 sp	eed -60	M4 🔻 s	peed	0	
	+	+ +	+	+	

### Moving toward left lower corner:

forever	+				
Motor M1 💌 sp	eed -60	M2 🔻	speed	0	
Motor M3 🔻 sp	eed 0	M4 👻 s	speed	-60	
	+	+ +	÷	+	

Exercise 3:

Moving around in a square:



Moving around in a circle:





### M shape route driving





## 1.3.5 Lesson 5 and Lesson 6



Introduction

**Teaching Objectives** 

### **Ultrasonic Obstacle Avoidance**



Principal of ultrasonic module


Trig	Transmitter Objectiver	t
Stemhub: bit Ultrasonic		
Hath	Ultrasonic	
Stemhub:bit	Ultrasonic Distance(cm)	
Neopixel	Ultrasonic Distance(cm) pin P0 ▼	
Advanced	Ultrasonic Light all V show color red V	Ultrasonic Distance(cm
Ultrasonic Dist	tance(cm) pin P0 🔻	
Ultrasonic Light	all ▼ show color red ▼	







## **Exercise 3**





Exercise 6





## Think about it



Answer

forever	-						
On-board Light	one 👻	color	blue •	bright	iness	255	2
pause (ms) 100		+		· ·	+	+	-
On-board Light	one 🔻	color	blue •	bright	iness	0	+
On-board Light	two 🔻	color	blue •	bright	iness	255	
pause (ms) 100		+	+ •	r	÷	÷	1
On-board Light	two 🔻	color	blue 🖣	bright	ness	0	+
On-board Light	three	• colo	or blue	🝷 brig	htnes	s 25	5
pause (ms) 100							
On-board Light	three	• colo	or blue	👻 brig	htnes	s ()	
On-board Light	four •	color	blue	• brig	tness	255	
pause (ms) 100	•					+	
On-board Light	four 🔻	color	blue	🕶 brigt	tness	0	9
Ultrasonic Ligh	t left	▼ sho	w color	blue •			
pause (ms) 100	•	+	+		+		
Ultrasonic Ligh	t left	▼ sho	w color	black	•		
Ultrasonic Ligh	t righ	t 👻 sł	now colo	or blue	•		
pause (ms) 100		+	+	н. н. Н	+		
Ultrasonic Ligh	t righ	t 🔻 sł	now colo	black	•		
	-	+	-	e e	+		







forever		+ forever		
if Ultras	onic Distance(cm) < 🔹 15 t	hen if <mark>Ultras</mark>	onic Distance(cm) < 🔻 15	then
Motor M1 👻	speed 0 M2 - speed 60	Motor M1 -	speed 60 M2 <del>-</del> speed 0	
Motor M3 🔻	speed 0 M4 - speed 60	Motor M3 -	speed 60 M4 🗸 speed 0	a a.
else		⊖ else		Θ
Motor M1 -	speed 60 M2 - speed 60	Motor M1 -	speed 60 M2 - speed 60	
Motor M3 -	speed 60 M4 - speed 60	Motor M3 🔻	speed 60 M4 - speed 60	· ·
	+ + + + + +			
			* * * * *	



When obstaclehandand the car distanced 10cm above, the car will move forward When obstaclehandand the car distanced 10cm less, the car will move backward When obstaclehandand the car distanced 10cmthe car will stop there to avoid\_  $\rightarrow$  distancing furthermore.





forever	+ +	+	+ +	+ +	+
if Ultras	onic Distanc	e(cm)	> • 10	then	
Motor M1 🔫	speed 150	M2 🔻	speed 15	0	+
Motor M3 🔹	speed 150	M4 🔻	speed 15	0 +	+
else if Ult	rasonic Dist	tance(cm)		10 t	hen Θ
Motor M1 🔻	speed 100	M2 🔻	speed -1	00 +	+
Motor M3 🔻	speed 100	M4 🔻	speed -1	00	
pause (ms) 2	999 🔹 👘	+	+ +	+ +	+
$\odot$					



on start	
set detecteddistance • to 12	
forever	
if Ultrasonic Distance(cm) <  v detecteddistance  v then	
Motor Stop All	
Ultrasonic Light all - show color red -	
Motor M1 - speed 150 M2 - speed -150	
Motor H3 - speed 0 H4 - speed 0	
pause (=s) 1000 -	
Motor Stop All	
pause (ms) 500 -	
if Ultrasonic Distance(cm) < • detecteddistance • then	if Ultrasonic Distance(cm) < v detecteddistance v then
Motor Stop All	Profession
Ultrasonic Light all - show color red -	Netor Stop All
Motor M1 - speed -150 M2 - speed 150	else
Motor N3 + speed 0 N4 + speed 0	Ultrasonic Light all - show color green -
pause (ms) 2000 -	Motor M1 - speed 50 M2 - speed 50
Motor Stop All	Motor M3 - speed 50 M4 - speed 50
pause (ms) 500 •	

Appendix



# Programming for Ultrasonic Sensor

function ultrasonic 🔗						
digital write pin P15 🔻 to 🥚						
wait (µs) 2						
digital write pin P15 🔻 to 🔳						
wait (µs) 10						
digital write pin P15 🔹 to 🕜						
set distance ▼ to pulse in (µs) pin	P14 🔻	pulsed	high	•	÷ 🔻	58
show number distance 🔻						

# 1.3.6 Lesson 7 and Lesson 8



#### Introduction

## **Teaching Objectives**

## Movement through line tracking

• Infrared Line Tracking sensor



• Line following Basic Module













**Exercise 3** 



# Exercise 4Maze



## Challenge

- Build Variable "BW\_value", think about how to store both digital signal of both left and right sensor at the same time.
- Build Variable "last\_value" to store the signal when it is out of bounds and decide the turning direction when out of bounds.

## Answer

### Exercise 1: Design a program to let the car move following the black line



set L - to digital read pin P13 - set R - to digital read pin P14 - if L 0 and - R 0 then Motor M1 - speed 60 M2 - speed 60 Motor M3 - speed 60 M4 - speed 60 else if L 1 and - R 0 then Motor M1 - speed 80 M2 - speed 0	-
set R • to digital read pin P14 • if L • = • 0 and • R • = • 0 then Motor M1 • speed 60 M2 • speed 60 Motor M3 • speed 60 M4 • speed 60 else if L • = • 1 and • R • = • 0 then Motor M1 • speed 80 M2 • speed 0	
if L ▼ = ▼ 0 and ▼ R ▼ = ▼ 0 then Motor M1 ▼ speed 60 M2 ▼ speed 60 Motor M3 ▼ speed 60 M4 ▼ speed 60 else if L ▼ = ▼ 1 and ▼ R ▼ = ▼ 0 then Motor M1 ▼ speed 80 M2 ▼ speed 0	
Motor       M1 *       speed       60       M2 *       speed       60         Motor       M3 *       speed       60       M4 *       speed       60         else       if       L *       *       1       and *       R *       *       0       then         Motor       M1 *       speed       80       M2 *       speed       0	
Motor M3 - speed 60 M4 - speed 60 else if L 1 and - R - 0 then Motor M1 - speed 80 M2 - speed 0	
else if L • • • 1 and • R • • • 0 then Motor M1 • speed 80 M2 • speed 0	
Motor M1 - speed 80 M2 - speed 0	Θ
Motor M3 - speed 80 M4 - speed 0	
else if $\begin{bmatrix} \mathbf{v} & \mathbf{v} & 0 \end{bmatrix}$ and $\mathbf{v} & \begin{bmatrix} \mathbf{v} & \mathbf{v} & 1 \end{bmatrix}$ then	Θ
Motor M1 - speed 0 M2 - speed 80	
Motor M3 - speed 0 M4 - speed 80	
else if $\begin{bmatrix} \mathbf{v} & \mathbf{v} & 1 \end{bmatrix}$ and $\mathbf{v} & \begin{bmatrix} \mathbf{v} & \mathbf{v} & 1 \end{bmatrix}$ then	Θ
Motor M1 - speed -60 M2 - speed -60	
Motor M3 <b>v</b> speed -60 M4 <b>v</b> speed -60 m m m m	

Exercise 2:



<pre>set L • to digital read pin P13 • set R • to digital read pin P14 • if L • = • 0 and • R • = • 0 th Motor M1 • speed 60 M2 • speed 60 Motor M3 • speed 60 M4 • speed 60 else if L • = • 1 and • R • = • 0</pre>	en
set R • to digital read pin P14 • if L • = • 0 and • R • = • 0 th Motor M1 • speed 60 M2 • speed 60 Motor M3 • speed 60 M4 • speed 60 else if L • = • 1 and • R • = • 0	en
if <b>l</b> • • • 0 and • <b>R</b> • • • 0 th Motor M1 • speed 60 M2 • speed 60 Motor M3 • speed 60 M4 • speed 60 else if <b>L</b> • • 1 and • <b>R</b> • • 0	en
Motor       M1 • speed       60       M2 • speed       60         Motor       M3 • speed       60       M4 • speed       60         else if       L • • • 1       and •       R • • • 0	then 🗭
Motor M3 - speed 60 M4 - speed 60	then 🖨
else if L V - V 1 and V R V - V 0	then 🖨
Motor M1 - speed 80 M2 - speed 0	
Motor M3 - speed 80 M4 - speed 0	
else if L V = V 0 and V R V = V 1	then Θ
Motor M1 - speed 0 M2 - speed 80	
Motor M3 - speed 0 M4 - speed 80	
else if $\begin{bmatrix} v & v \\ v & v \end{bmatrix}$ and $\overrightarrow{R}$ $\overrightarrow{R}$ $\overrightarrow{r}$ $\overrightarrow{r}$ $\cancel{1}$	then Θ
Motor M1 - speed -60 M2 - speed -60	
Motor M3 - speed -60 M4 - speed -60 marked	
•	





# Challenge

on start	forever		
call line_follow	call line_follow		
function line_fo	llow		
set BW_value ▼	to digital read pin P	13 🔹 💌	• 18
change last_val	ue 🔻 by digital read pin	n P14 🔻	
if BW_valu	e 🕶 💉 11 then		
Motor M1 - s	peed 80 M2 - speed 8	0	
Motor M3 🔻 s	peed 80 M4 - speed 8	• • •	
set last_val	ue 🔻 to (BW_value 🔹)		
else	Θ		
if last_	value 🔻 💷 🚺 the	en	
Motor M1 -	speed -40 M2 - speed	80	
Motor M3 🔻	speed -40 M4 - speed	80	
else	•	Э	
Motor M1 -	speed 80 M2 - speed	-40	
Motor M3 🖛	speed 80 M4 - speed	-40	
$\odot$			
pause (ms) 10			

# 1.3.7 Lesson 9 and Lesson 10



Introduction

**Teaching Objectives** 

# Controller



# Radio and our daily life



# Radio programming module

← 返回	1	鏡展		?
(III	尋玩是直读输入闲让		٩	Í
Restort	radio-broadcast			
Bluetooth services	Adds new blocks for message communication in the radio	A micro-servo library	AdaFruit NeoPixel driver 進一步眼解	
https://github.com/lzty6	34158/GHBit		٩	
	GHBit Extension for YahBo GHBit_V1/V2 V3.0.3	om		
	User-provided extense endorsed by Microso	sion, not oft. Learn more		



# Example 1: Two-way radio



Program module for controller and the car



# Programming for controller:(GHBit)

star	t		+			forever + + + + + + + +
radio	set	group	1			if Rocker value Up ▼ then
radio	set	trans	mit	power	7	radio send string "Up"
•			÷	+	+	else if Rocker value Down ▼ then
						radio send string "Down"
						else if Rocker value Left - then
						radio send string "Left"
						else if Rocker value Right  then
						radio send string "Right"
						also if Button pum B1 = value Drace = then
						eise in Button num Bi + Value Press + then
						radio send string Bi
						else if Button num B2 ▼ value Press ▼ then
						radio send string "B2"
						else if Button num B3 ▼ value Press ▼ then
						radio send string "B3"
						else if Button num B4 ▼ value Press ▼ then
						radio send string "B4"
						+ + + + + + +

B2: Display a heart shape
B3: The car moves toward left
B4: The car move toward right



Exercise 2

Answers



on	star	t				forever
F	adio	set g	roup (	1		if Rocker value Up - then
r	adio	set t	ransmi	it pow	ier 7	radio send string "Up"
						else if Rocker value Down - then $igodot$
						radio send string "Down"
						else if Rocker value Left  then
						radio send string 'Left'
						else if Rocker value Right - then \ominus
						radio send string "Right"
						else if Button num B1 ▼ value Press ▼ then ⊖
						radio send string "B1"
						else if Button num B2 → value Press → then —
						radio send string "B2"
						else if Button num B3 → value Press → then $\bigcirc$
						radio send string "B3"
						else if Button num B4 → value Press → then ⊝
						radio send string "B4"
						else \ominus
						radio send string "Stop"
						$\odot$



# 1.4 Micro:bit M1 SMART CAR\_Intermediate

# 1.4.1 Lesson 1



## Introduction

## Learning Target

## **Reviewing the structure of Mecanum Wheel Car**

- 4 Motor(M1, M2, M3, M4)
- 4 Mecanum Wheel
- Micro bit
- Micro bit expanding board


### micro:bit expansion board



### Sensor

• Ultrasonic Sensor



• Infrared Receiver and Sender



# Reviewing of advanced block

Search	Q	fe Functions
Basic		
<ul> <li>Input</li> </ul>		Make a Function
O Music		
C Led		
I Radio		
C Loops		
🔀 Logic		
🔳 Variables		
🗮 Math		
Advanced		
$f_{(x)}$ Functions		
∃≣ Arrays		
T Text		
😎 Game		
🖾 Images		
Pins		

Edit Function		(.) Music						0
Add a parameter	T Text	≍ Boolean	🖩 Number	i≣ Array	✓ LedSprite	🖬 Image		
			function	doSomething				
							Done	*
forever		+						



#### Meet the Mechanical extension tools

### Beetle



### Loader



# Forklift



# Bulldozer with ultrasonic



#### Components in extension package



# Meet the Micro:bit extension tools - beetle





# Principle and function of Mechanical beetle





#### Install of beetle

















# Stemhub:bit - Servo Motor block module

Stemhub:bit	
Servo	
Servo S1 🔻 degree 🛛 0	
Servo(270°) S1 ▼ degree 0	Servo S1 - degree 0









- Refer to Beginner, Lesson 10, receive text as operation condition
- Apply basic >> Show of indicate light using block module

#### Answer

Angle value←	Beetle movement↩	<⊐ <
Increase	Close←	
Decrease←	Open←	
on button A • press Servo S1 • degree	sed for the serve S1	B ▼ pressed ▼ degree 110

# Program of Remote

on start			forever + + + + + + + +
radio se	et grou	p 1	if Rocker value Up ▼ then
radio se	et grou	p 7	radio send string "Up" + + + + + +
		+	else if Rocker value Down ▼ then ⊖
			radio send string "Down"
			else if Rocker value Left - then $igodot$
			radio send string "Left"
			else if Rocker value Right ▼ then ⊖
			radio send string "Right"
			else if ⊂ Button num B1 🕶 value Press 🕶 then ⊝
			radio send string "B1"
			else if ⊂ Button num B2 🕶 value Press 🕶 then ⊝
			radio send string "B2" + + + + + +
			else $igodot$
			radio send string "Stop"
			$\odot$



#### **Program of the Car**

# 1.4.2 Lesson 2



Introduction

Learning Tutorial



#### Install of Ultrasonic sensor:











**Exercise 3** 



- Built variable "goal" to mark the destination, and use it in Conditional blockUse text "start" and "end" indicate start point and end point
- Beware of the sequence of the condition of ultrasonic, line patrol and "goal" condition

#### Answer



forever	+ + +	+ + +	+ +	+	+
if Ultras	onic Distance	(cm) > •	4 th	en	
Servo S1 🔻	degree 🛛	Clip open	+ +	+	+
pause (ms)	100 🔹 👘	+ + +	+ +		
Motor M1 -	speed 80 M	2 🔻 speed 🚷	80		
Motor M3 -	speed 80 M	4 🔹 speed 🛛	80	+	+
else if Ul	trasonic Dist	ance(cm) ≤	• 4	the	n Θ
Motor Stop A	<b>11</b> + +	+ + +	+ +	+	+
Servo S1 🔻	degree 110	Clip close	+ +	+	+

forever	on start
set L 💌 to digital read pin P13 💌	set goal • to fend Default destination at end
set R 🔻 to digital read pin P14 👻	point
call ultrasonic	
	function LineFollow ⊘
function ultrasonic 🔗 🔹 Program to end point	if <b>L•</b> -• 0 and • <b>R•</b> -• 0 then
if goal • - • • end • then	Motor M1 - speed 60 M2 - speed 60 manual and a second seco
if Ultrasonic Distance(cm) > ▼ 5 then	Motor M3 - speed 60 M4 - speed 60
Servo S1 - degree 0	else if $l \cdot \cdot \cdot 1$ and $\cdot R \cdot \cdot 0$ then $\Theta$
call LineFollow	Motor M1 - speed 80 M2 - speed 0
pause (ms) 2000 -	Motor M3 - speed 80 M4 - speed 0
else if Ultrasonic Distance(cm) ≤ ▼ 5 then ⊖	else if $\left[ \cdot \cdot - \cdot \right] = \left[ \cdot \cdot \right]$ and $\left[ \cdot \cdot \right] = \left[ \cdot \cdot \right]$ then $\Theta$
Motor Stop All	Motor M1 - speed 0 M2 - speed 80
Motor M1 - speed 100 M2 - speed -100	Motor M3 v speed 0 M4 v speed 80
Hotor M3 - speed 100 M4 - speed -100	
pause (ms) 2000 - Rotate 180 degree	
set goal + to (start) Change toward starting p	Gint goal - · (start) then
$\odot$	Motor Stop All When excess black line.
else if $\bigcirc$	Servo S1 - degree 110 destination is end point.
call LineFollow	else $\bigcirc$ close the beetle
	Motor Stop All
Program to start point	

# 1.4.3 Lesson 3



Introduction

**Learning Target** 



# Understand Micro:bit expansion tool – Loader



# Principle and function of loader



#### **Install of Loader**













+ 返回			A IR		?
		(國際成是直接输入網站		٩	
E	) 🧖				
Bluet	tooth soth services	radio-broadcast Adds new blocks for message communication in the radio	Servo A micro-servo library	neopbxel AdaFruit NeoPixel driver 進一步眼斜	
L	Dis E	7			
https://	github.com/lzty63	34158/GHBit		٩	
		GHBit Extension for YahBoom GHBit_V1/V2 V3.0.3			
		User-provided extension, not endorsed by Microsoft. Learn n	nore		

- B1 and B4 bottom control the horizontal movement of left and right
- B2 and B3 bottom control the load or unload of loader
- Control rods control the back and forward and turning left or right
- Car stops when pending of remote
- When load or unload, DoReMi and SoFaMi will play respectively ( or any other melody



- Refer to Beginner, Lesson 10, Receive text as operation condition
- Use of sound effect >> perform melody speed (bpm) in block module

#### Answer

Angle value 🛛 🖓	Loader movement⇔	ς
Increase←	Unload	<b>E</b>
Decrease	Load←	<b>ČEDÕAT</b>
on button A • pr Servo S3 • deg	ree 80 Ser	utton B • pressed •vo S3 • degree 150

# Program of remote

on start		forever + + + + + + + +
radio set group	1 + +	if <b>Rocker value Up →</b> then
radio set group	7	radio send string "Up"
	+ + +	else if Rocker value Down → then $igodot$
		radio send string "Down"
		else if Rocker value Left → then 🕞
		radio send string "Left"
		else if Rocker value Right - then 🕞
		radio send string "Right"
		else if ⊂Button num B1 → value Press → then ⊖
		radio send string "B1"
		else if Button num B2 ▼ value Press ▼ then ⊖
		radio send string "B2" and a second second second
		else if Button num B3 ▼ value Press ▼ then ⊖
		radio send string "B3" + + + + +
		else if Button num B4 ▼ value Press ▼ then ⊖
		radio send string "B4" + + + + + + + +
		else
		radio send string "Stop"
		+ + + + + + + + +

#### **Program of Car**



# 1.4.4 Lesson 4



Introduction

**Learning Target**


### To install the ultrasonic sensor













Answer



forever	+ +	+	+	+ +
if Ultrasonic Distance(cm)	> • 4	the	n	
Servo S3 - degree 150 Un	load	· +	+	
pause (ms) 100 🔻	+ +	+		
Motor M1 • speed 80 M2 • sp	eed 80	+		
Motor M3 ▼ speed 80 M4 ▼ sp	eed 80	+	+	+ •
else if Ultrasonic Distance(cr	n) <	4	then	Θ
Motor Stop All	+ +	+	+	+ -
Servo S3 - degree 80 Load Load L	Jp	+	+	+ +



## The program of the car

<pre>radio set group 1 set mode * to `avoid'  forever set L * to digital read pin P13* set R * to digital read pin P13* set R * to digital read pin P13* set R * to digital read pin P14* call Ultrasonic  function Ultrasonic Distance(cm) &gt;  \$ then     Servo S3 * degree 159     call LineFollower     if Ultrasonic Distance(cm) &gt;  \$ then     Servo S3 * degree 159     call LineFollower     if Ultrasonic Distance(cm) &gt;  \$ then     Servo S3 * degree 159     call LineFollower     if Ultrasonic Distance(cm) &gt;  \$ then     Servo S3 * degree 159     call LineFollower     if Ultrasonic Distance(cm) &gt;  \$ then     Servo S3 * degree 159     call LineFollower     if Outr M3 * speed 60     H2 * speed 60</pre>
<pre>set mode * to 'uouid' set mode * to 'Load' Mode: avoid &gt;&gt; load forever set L * to digital read pin P13 * set R * to digital read pin P14 * call Ultrasonic function Ultrasonic if Ultrasonic Distance(cm) &gt;  \$ then serve 33 * degree 159 call LineFollower else if Ultrasonic Distance(cm) ≤  \$ then if unction LineFollower if unction Ultrasonic Distance(cm) ≤  \$ then if unction Ultrason</pre>
forever set L * to digital read pin P13 * set R * to digital read pin P14 * call Ultrasonic function Ultrasonic © if Ultrasonic Distance(cm) >  \$ then set s if Ultrasonic Distance(cm) >  \$ then function LineFollower else if Ultrasonic Distance(cm) ≤  \$ then © if Distance(cm) ≤  \$ then © if Distance(cm) ≤  \$ then © if Ultrasonic Distance(cm) ≤  \$ then © if Distance(cm) ≤  \$ \$ then © if Distance(cm) ≤  \$ \$ \$ then © if Distance(cm) ≤  \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$
forever         set L + to digital read pin P13 +         set R + to digital read pin P14 +         call Ultrasonic         function Ultrasonic         if       Ultrasonic Distance(cm) >
<pre>set L * to digital read pin P13 * set R * to digital read pin P14 * call Ultrasonic function Ultrasonic O servo S3 * degree 150 call LineFollower else if Ultrasonic Distance(cm) &gt; * 5 then if Ultrasonic Distance(cm) &gt; * 5 then call LineFollower else if Ultrasonic Distance(cm) &gt; * 5 then if unction LineFollower else if Ultrasonic Distance(cm) &gt; * 5 then if unction Distance(cm) &gt; * 5 then if unc</pre>
<pre>set R * to digital read pin P14 * call Ultrasonic function Ultrasonic @ if Ultrasonic Distance(cm) &gt;  \$ then Servo S3 * degree 150 call LineFollower else if Ultrasonic Distance(cm) ≤  \$ then @ if @ dad @ d</pre>
call Ultrasonic         function       Ultrasonic Distance(cm) >> 5         tif       Ultrasonic Distance(cm) >> 5
function       Ultrasonic @         if       Ultrasonic Distance(cm) > < \$ then         Servo S3 + degree 150       call LineFollower         else if       Ultrasonic Distance(cm) ≤ + \$ then         if       Dirasonic Distance(cm) ≤ + \$ then         or M3 + speed 100       M4 + speed -100         unction       Line Follower         else if       Ultrasonic Distance(cm) ≤ + \$ then         if       mode x - x @ or id         if       mode x - x @ or id
<pre>function Ultrasonic Oistance(cm) &gt; • 5 then function LineFollower else if Ultrasonic Distance(cm) ≤ • 5 then if mode * - * avoid then function LineFollower else if Ultrasonic Distance(cm) ≤ • 5 then if mode * - * avoid then function LineFollower function Line</pre>
if Ultrasonic Distance(cm) > * 5 then Servo S3 * degree 150 call LineFollower else if Ultrasonic Distance(cm) ≤ * 5 then ⊖ if mode * - * avoid then Avoid mode
Servo $S3 \bullet$ degree 150 call LineFollower else if Ultrasonic Distance(cm) $\leq \bullet$ 5 then $\bigcirc$ if mode $\star$ - $\star$ avoid then Avoid mode
call LineFollower         else if       Ultrasonic Distance(cm) ≤ + 5 then ⊖         if       mode + - + (avoid) then         Avoid mode
else if Ultrasonic Distance(cm) $\leq \bullet$ 5 then $\Theta$ if mode $\bullet$ - $\bullet$ "avoid" then Avoid mode
if mode (avoid) then Avoid mode
Avoid mode else if $(\mathbf{l} \cdot \mathbf{r} - \mathbf{r} \cdot \mathbf{l})$ and $\mathbf{r} \cdot (\mathbf{R} \cdot \mathbf{r} - \mathbf{r} \cdot 0)$ then $\Theta$
Motor Stop All
pouse (ms) 500 ▼ Motor M3 ▼ speed 80 M4 ▼ speed 0
call avoid Avoid
else if mode V = V (load) then O Load mode
Hotor Stop All service
Servo S3 • degree 80 Load up object
pause (ms) 1000 $\bullet$ then $\Theta$ else if $L \bullet = \bullet (1)$ and $\bullet (R \bullet = \bullet (1))$ then $\Theta$
⊕     Motor M1 ▼ speed 60 M2 ▼ speed -60
Motor M3 + speed 60 H4 + speed 60 Avoid and back on track

# 1.4.5 Lesson 5



Introduction

Learning Target

## Meet the Micro:bit Expansion Tool - Forklift





## The principle and function of forklift

# Install the forklift



















#### **Exercise 2**

Find items suitable for forklift handling and calculate their angles

(The highest height that the forklift can lift is 6cm, pay attention to the size of the object selected)

#### **Exercise 3**



#### Answer

## Exercise 1:

on button 🗛 🔻 pressed		on button <b>B -</b> pressed
call Open 🕌 🖡 💡		call Close
+ + + + +		+ + + + + +
function Open 🔿 👘	+	function Close
Servo S2 - degree 0		Servo S2 - degree 100

Exercise 3:

#### Car program



## Remote programm

on s	tart				forever + + + + + + + +
rad	dio set	group	1	+ +	if Rocker value Up 🔹 then
rac	dio set	group	7	+ +	radio send string "Up" + + + + + + + +
			+	+ +	else if Rocker value Down ▼ then
					radio send string "Down"
					else if Rocker value Left - then
					radio send string "Left"
					else if Rocker value Right 🔻 then
					radio send string "Right"
					else if Button num B1 ▼ value Press ▼ then 🤆
					radio send string "B1" + + + + + +
					else if Button num B2 ▼ value Press ▼ then (
					radio send string "B2" + + + + + +
					else if Button num B3 ▼ value Press ▼ then 🤆
					radio send string "B3" + + + + + + +
					else if Button num B4 ▼ value Press ▼ then 🤆
					radio send string "B4" + + + + + + + + +
					else
					radio send string "Stop"
					•
					+ + + + + + + +

# 1.4.6 Lesson 6



Introduction

**Learning Target** 

Use of forklift tools

To install the ultrasonic sensor













### Answer

forever	+ + +					
Servo S2 🔻 d	legree 100					
if Ultras	sonic Distanc	e(cm)	<ul><li>≤ ▼ (</li></ul>	12	then	
Motor Stop	<b>All</b> + +	+	+ +	+	+	+
Servo S2 🔻	degree 10	+				
pause (ms)	1000 🔻	+	+ +	+		
Motor M1 🔻	speed 100	M2 🔻	speed	100		
Motor M3 🔻	speed 100	M4 🔻	speed	100	+	ł
else					Θ	
Motor M1 🔻	speed 100	M2 🝷	speed	100		
Motor M3 🔻	speed 100	M4 🔻	speed	100		
$\odot$	+ + +	+	÷ +	+	+	J

<pre>set L * to digital read pin P13 * set R * to digital read pin P14 *  if L * = * 1 and * R * = * 1 tH Motor Stop All Servo S2 * degree 0 pause (ms) 1000 * Motor M1 * speed 80 M2 * speed 80 Motor M3 * speed 80 M4 * speed 80 pause (ms) 2000 * else Servo S2 * degree 100</pre>												er	orev
<pre>set R • to digital read pin P14 •  if</pre>					+		P13	pin	read	digital	] to (	L •	set
<pre>if</pre>	+				+		P14	pin	read	digital	to (	R 🕶	set
Motor Stop All Servo S2 • degree () pause (ms) 1000 • Motor M1 • speed 80 M2 • speed 80 Motor M3 • speed 80 M4 • speed 80 pause (ms) 2000 • else	nen	th	1	•	-	R	•<	and		= 🔹 1	Ð		if
Servo S2 • degree 0 pause (ms) 1000 • Motor M1 • speed 80 M2 • speed 80 Motor M3 • speed 80 M4 • speed 80 pause (ms) 2000 • else Servo S2 • degree 100	+	+	+	+	+	+	+	+	+	<b>11</b> *	Stop /	tor	Мо
<pre>pause (ms) 1000 ~ Motor M1 ~ speed 80 M2 ~ speed 80 Motor M3 ~ speed 80 M4 ~ speed 80 pause (ms) 2000 ~ else Servo S2 ~ degree 100</pre>									0	degree	S2 💌	rvo	Se
Motor M1 • speed 80 M2 • speed 80 Motor M3 • speed 80 M4 • speed 80 pause (ms) 2000 • else Servo S2 • degree 100									+	1000 🔻	(ms)	use	ра
Motor M3 • speed 80 M4 • speed 80 pause (ms) 2000 • else Servo S2 • degree 100					30	ed 8	spee	M2 🔻	80	speed	M1 🔻	tor	Мо
pause (ms) 2000 ▼ else Servo S2 ▼ degree 100					30	ed 8	spee	M4 🔻	80	speed	M3 🔻	tor	Мо
else Servo S2 • degree 100					+	+	+	+		2000 🔻	(ms) (	use	ра
Servo S2 - degree 100	Θ	(										2	else
		+	+	+	+	+	+		100	degree	S2 🔻	rvo	Se
Motor M1 - speed 80 M2 - speed 80					30	ed 8	spee	M2 🔻	80	speed	M1 -	tor	Мо
Motor M3 - speed 80 M4 - speed 80					30	ed 8	spee	M4 🔻	80	speed	мз 🝷	tor	Мо
$\odot$													lacksquare

function Ultrasonic	⊘
if Ultrasonic Di	stance(cm) ≤ ▼ 5 then
Motor Stop All Servo S2 • degree pause (ms) 1000 •	Stop Car lift the fork
else	Θ
call LineFollower	Call Line tracking
•	
Initialize	+ + + + + +
forever call Ultrasonic	on start Servo S2 – degree 80

#### Line tracking module



# 1.4.7 Lesson 7



Introduction

**Learning Target** 

Meet the Micro:bit Expansion Tool - Bulldozer





## Principle and function of mechanical bulldozer





### To install the bulldozer





Components – with an ultrasonic sensor:









- First adjust the installation angle of the servo so that the angle value of the servo module is 85 as the front
- Only using the servo module 0-180 degrees cannot make the servo ultrasonic sensor look to the left and right
- Negative numbers and values above 180 degrees must be tried







# 1.4.8 Lesson 8



Introduction

**Learning Target** 

Use of bulldozer tools
# Shoot the gantry



## **Football match**

- 1. The car  $\mathbf{or}$  extension tool cannot exceed the line  $\mathbf{in}$  front of the gantry 2. The speed cannot exceed 150
- 3. Do  ${\bf not}$  touch the car during the race



## Answer

Exercise 1:

#### Car program



# Remote program

on start		forever + + + + + + + +
radio set group	1 + +	if <b>Rocker value</b> Up <b>▼</b> then
radio set group	7	radio send string "Up" + + + + + + + + +
	+ + +	else if Rocker value Down → then $\bigcirc$
		radio send string "Down"
		else if Rocker value Left → then $\bigcirc$
		radio send string "Left"
		else if Rocker value Right - then 🕞
		radio send string "Right"
		else if ⊂ Button num B1 ▼ value Press ▼ then ⊖
		radio send string "B1"
		else if Button num B2 ▼ value Press ▼ then ⊝
		radio send string "B2" and a set and a
		else if ⊂ Button num B3 ▼ value Press ▼ then ⊖
		radio send string "B3" + + + + + + +
		else if Button num B4 ▼ value Press ▼ then ⊖
		radio send string "B4" + + + + +
		else $\bigcirc$
		radio send string "Stop"
		+ + + + + + + +

# Appendix



# 1.4.9 Lesson 9



Introduction

Learning Target

Liftable clip

To install the lift bracket

To attach the clip to the lift bracket





#### **Exercise 2**

- The length of the clip will increase after adding the bracket, sufficient distance is needed when arranging the site.
- Pay attention to whether it is necessary to pause when operating every tool.
- Create a variable "destination" to record the destination, and use the destination to determine the action when the car goes out of bounds

#### Muti – function engineering car (Clips and Forks)

#### **Exercise 3**

- The forklift is in front of the car and the clip is behind the car
- Directly adjust the program in exercise 1
- Create variables to record the usage status of the two tools, and use this status to determine the next action of the tool

#### Answer

# Remote program

on sta	rt				forever + + + + + + + +
radi	o set	group	1	+ +	if Rocker value Up ▼ then
radio	o set	group	7	+ +	radio send string "Up"
			+	+ +	else if <b>Rocker value Down →</b> then
					radio send string "Down"
					else if Rocker value Left ▼ then
					radio send string "Left"
					else if ≮Rocker value Right ▼ then
					radio send string "Right"
					else if (Button num B1 • value Press • ) then 👄
					radio send string "B1"
					also if Rutton num R2 = value Press = then
					radio cond stains "P2"
					raato sena string B2
					else if Button num B3 • Value Press • then •
					radio send string B3 + + + + +
					else if Button num B4 ▼ value Press ▼ ) then ⊝
					radio send string "B4"
					else 🖯
					radio send string "Stop"

# Car program

on star	t							on radio received receivedString
radio	set	gro	up 🚺		Defa	ault		if <b>receivedString ▼</b> = ▼ <b>*Up*</b> then
Servo	S1 •	d	egree	110	Bee	tle (	Close	Motor M1 - speed 255 M2 - speed 255
Servo	S2 •	d	egree	100	Brad	cket	down	Motor M3 - speed 255 M4 - speed 255
								else if < receivedString ▼ = ▼ *Down* then ⊖
								Motor M1 - speed -255 M2 - speed -255
								Motor M3 → speed -255 M4 → speed -255
								else if < receivedString ▼ = ▼ ('Left') then ⊖
								Notor M1 + speed -85 N2 + speed 255
								Notor M2 - speed 25 M4 - speed 255
								MUCUP HSV Speed of HHV Speed 255
								else if receivedString • = • Right then 👄
								Motor M1 - speed 255 M2 - speed -85
								Motor M3 - speed 255 M4 - speed 85
								else if receivedString $\bullet = \bullet$ "B1" then $\Theta$
								Servo S1 - degree 0
								else if receivedString • = • "B2" then $\Theta$
								Servo S1 - degree (110)
								else if receivedString = • "B3" then
								Servo S2 degree 0
								else if receivedString 'B4' then $\Theta$
								Servo S2 - degree 100
								else if receivedString $\bullet$ = $\bullet$ 'Stop' then $\Theta$
								Motor Stop All

on start	function LineFollow
Servo S1 - degree Ø Default:	if $\mathbf{L} \cdot = \cdot $
Servo S2 • degree 100 Bracket Down	Motor M1 - speed 60 M2 - speed 60
set destination ▼ to <b>'end'</b>	Motor M3 - speed 60 M4 - speed 60
Destination: end point	else if $\mathbf{L} \mathbf{v} = \mathbf{v}$ 1 and $\mathbf{v}$ $\mathbf{R} \mathbf{v} = \mathbf{v}$ 0 then $\Theta$
forever	Motor M1 - speed 80 MZ - speed 0
set L • to digital read pin P13 •	Motor M3 → speed 80 M4 → speed 0
set R ▼ to digital read pin P14 ▼	else if $L \bullet = \bullet 0$ and $\bullet R \bullet = \bullet 1$ then $\Theta$
call LineFollow	Motor M1 - speed 0 M2 - speed 80
· · · · · · · · · · · ·	Motor M3 → speed 0 M4 → speed 80
	else if $L \bullet - \bullet 1$ and $\bullet R \bullet - \bullet 1$ then $\Theta$
	if destination ▼ = ▼ "end" then
	Motor Stop All Car stops
	Servo S1 - degree 110 Beetle Clip object
Dragger when destination is and point	pause (ms) 500 -
Program when destination is end point	Servo S2 - degree 🕜 - Bracket lift up
	Motor M1 - speed 80 M2 - speed -80 180 degrees
	Motor M3 - speed 80 M4 - speed -80 turn right
	pause (ms) 1500 -
	set destination • to 'start' Destination: start point
	else if destination • = • start then •
Program when destination is start point	Notor Stop All Car stops Serve St z degree Beetle release object
	$\odot$

# Exercise 3

# **Remote Program**

adi	o set	grou	p 1	·	if Rocker value Up - then
adi	o set	grou	p 7	·	radio send string "Up"
			-	· ·	else if Rocker value Down ▼ then
					radio send string "Down"
					else if Rocker value Left → then
					radio send string "Left"
					else if Rocker value Right → then
					radio send string "Right"
					else if Button num B1 - value Press - the
					radio send string "B1"
					pause (ms) 200 🔹
					else if Button num B2 ▼ value Press ▼ the
					radio send string "B2"
					pause (ms) 200 •
					else if Button num B3 - value Press - the
					radio send string ("B3")
					else if Button num B4 ▼ value Press ▼ the
					radio send string "B4")
					else
					radio send string "Stop"
					• • • • • • • • • • • • • • • • • • •

#### Car program



# 1.4.10 Lesson 10



Introduction

**Learning Target** 

Two-way movable fork

## Installation steps:



## Exercise 1



### Task

Control the car to transport the objects in the left to the middle, then transport the objects in the right to the left, and finally transport the objects in the middle to the right.

#### Car relay race

#### Introduce

#### Game rules

The relay race is divided into two stages, Stage 1: AB (task: transport all three\_ →objects from point A to point B), stage 2: BC (task: transport three objects from\_ →point B to point C, and Overlap these objects in the order of 1, 2, 3). After\_ → completing the second stage, only two cars reach point C (point A is the starting\_ → point, point B is the relay point, and point C is the end point) to be considered\_ → successful.

During the game, players cannot touch the car. Whoever completes all tasks in the shortest time wins.



Answer

Car program	(the	horizontal	moving	part	of the	fork)
-------------	------	------------	--------	------	--------	-------

on	star	t					on radio received receivedString
ŀ	adio	set g	Iroup	1			if <b>receivedString ▼ = ▼ 'B1'</b> then
s	et a	ngle •	• to	85			Servo S2 - degree 0 Fork up
S	iervo	S1 •	deg	ree	ang	le 🔻	else if receivedString ▼ = ▼ *BZ* then G
S	iervo	S2 🕶	deg	ree	100		Servo S2 - degree 100 Fork down
							else if <pre>receivedString ▼ = ▼ *B3*</pre> then
							if angle ▼ > ▼ -20 then
							change angle - by -1
							Servo S1 - degree angle - Fork turns left
							else if receivedString • = • B4 then
							if angle ▼ < ▼ 190 then
							change angle → by 1
							Servo S1 - degree angle - Fork turns right
							else if receivedString Front then
							set angle - to 85
							Servo S1 ▼ degree angle ▼
							else if receivedString • = • ("Stop") then
							Motor Stop All

#### **Remote Program**



# Car relay race

on st	tart								on radio received receivedString
rad	io s	et g	roup	1	-				if receivedString = "Up" then
rad	io s	et g	roup	7	4				Notor M1 - speed 255 M2 - speed 255
set	ang	le 🔻	to	90	÷	+	+		Notor N3 - speed 255 N4 - speed 255
Ser	vo S	1 -	degr	ee	angl	.e •			else if receivedString = • 'Down' then \ominus
									Motor M1 - speed -255 M2 - speed -255
									Motor M3 - speed -255 M4 - speed -255
									else if <
									Motor M1 - speed -85 M2 - speed 255
									Motor M3 - speed 85 M4 - speed 255
									else if receivedString = Tright then $\Theta$
									Hotor M1 + speed 255 M2 + speed -85
									Hotor M3 • speed 255 M4 • speed 85
									else if receivedStrings - * [R] then
									if angle > 10 then
									Servo S1 • degree angle •
									change angle v by -3
									pause (ms) 100 ♥
									else if receivedString = - 'B2' then ⊖
									if goale <b>x</b> (170) then
									change angle thy 3
									$\odot$
									else if receivedString = • (B3) then $\Theta$
									Motor M1 - speed 150 MZ - speed -150
									Motor M3 - speed -150 M4 - speed 150
									else if receivedString = - 'B4' then \ominus
									Motor M1 - speed -150 M2 - speed 150
									Motor M3 - speed 150 M4 - speed -150
									else if receivedString▼ = ▼ "Stop" then ⊖
									Motor Stop All
									$\odot$
94 st	art					+	+	+	on radio received receivedString Chapter 1. Tutorial qui
rad	io se	t gr	oup 2						if receivedString = 'Up' then
rad	io se	t ar	oup 7						

		+		+				forever
0	n butt	on	A 🔹 þ	ores	ssed			
	radio	set	grou	p (	1			if Rocker value Up T then
	radio	set	grou	p	7			radio send string ('Up'
	$\sim$							else if <b>Rocker value Down ▼</b> then ⊙
								radio send string 'Down'
			+		+	+		else if Rocker value Left ▼ then ⊖
0	n butt	on	B 🔻	pre	ssed	+		radio send string 'Left'
	radio	set	grou	IP (	2			else if Rocker value Right • then $igodot$
	radio	set	grou	IP (	7	+		radio send string "Right"
						+		else if Button num B1 • value Press • then $igodot$
								radio send string 'B1'
								else if Button num B2 ▼ value Press ▼ then ⊖
								radio send string 'B2'
								else if ⟨Button num B3 ▼ value Press ▼ ) then ⊖
								radio send string "B3"
								else if Rutton num R4 value Press v then
								radio send string B4
								else $igodot$
								radio send string Stop

# 1.5 Micro:bit M1 SMART CAR\_Advanced

# 1.5.1 Lesson 1



#### Introduction

Objective



# Introduction of HuskyLens

#### Method



- Dialing the function button to the left/right to switch functions/modes
- Click on the function button to call out the function menu
- After long pressing the function button, you can set whether to learn multiple targets.
- Click the function button again to drag the scroll bar to the right to enable "Learn More".



- Click the Learn button to learn a new object, and the object ID will be displayed on the screen.
- Press and hold the learning button to learn to recognize new object from different angle, and distance.
- After studying the new object, it is required to press the learning button again for further study within 4s.Otherwise, stop studying after 4 seconds, you must first forget the target and then re-study



• After terminating the study, press learning button twice for forgetting the learned objects.

## Installing HuskyLens in Smart Car













## HuskyLens expansion pack



HuskyLens building block module





#### **Exercise 1**



1When the program starts, it automatically switches to the object classification mode. 2Use HuskyLens to learn 5 kinds of objects (ID) 3Press the button A: display the number of learning objects - (a) 4Press the button B: clear learning object data 5Press the button A again to display the number of learning objects - (b)

HuskyLens forget all learning data of the current algorithm

## Exercise 2:

	Extensions	
https://github.com/stemhub/p	xt-Stemhubbit	٩
	STEM C Hub C C C C C C C C C C C C C C C C C C C	
	Learn more	
1When the program starts, 2Use HuskyLens to learn mu 3If there is a learned obj →object ID closest to the →otherwise lights up red.	it automatically switches to the ltiple objects. ect within the screen, the Mic center of the screen and light	the object classification mode. cro:bit board displays the_ nts all onboard LEDs green,_

#### Answer

Location←	Number←
(a)<⊐	
(b)<⊐	

start																					
luskyLens	initia	olize I	2C unt:	il sud	ccess																
luskyLens	switch	algor:	ithm to	o Obj	ject (	lassi	ificat	ion 🔻													
luskyLens	forget	t all 1	earning	g dat:	of 1	the c	urrent	algo	rithm												
		-							1	on bu	tton	B 👻	presse	d							
			÷.							Husi	yLens	; for	get all	lear	ning	data	of 1	the d	current	algo	rithm
n button		presse																		+	-
HuskyLe	ns requ	est dat	ta once	and	save	into	the r	esult		-	-										
show nu	mber 🚺	luskyLe	ns get	a tot	tal nu	umber	of le	arned	IDs	from	the r	esult									

			-																				
m	start																						
н	luskyL	ens i	initia	lize	e I2C unti	il succe	55																
н	luskyL	ens :	switch	al	gorithm to	o Object	clas	sific	ation	•													
				-	_		4			-													
					forever																		
					Huskyl	ens requ	Jest d	lata o	nce a	ind sa	we in	to th	e resi	lt			+	-					
					if	HuskyLe	ns ch	eck i	ffr	ane 🔻	is	on sci	reen f	from 1	the re	sult	the	•					
					On-t	oard Li	;ht o	ne 🔻	show	colo	r gr	een 🔻											
					show	number	Husk	cyLens	; get	ID -	of	frame	close	est t	o the	cente	er of	scree	n from	n the	resul	lt	
					else												e	) 					
					On-b	oard Lig	tht o	ne 🔻	show	colo	r re	a 🔹											
					show	number	Husk	cyLens	get	10 -	of	frame	close	est to	o the	cente	er of :	scree	n from	n the	resul	lt	
					•																		
																		-					

# 1.5.2 Lesson 2



#### Introduction

Objective

#### HuskyLens facial recognition function

What is facial recognition?

#### Applications of facial recognition

- Access right system: assess the right for entering protected areas such as prisons, detention centers, communities, schools, and residences by using facial recognition.
- Surveillance systemIt can be used to monitor crowds in public places, such as banks, airports, stadiums, shopping malls and supermarkets.
- Internet application: use face recognition to assist online payment, prevent others from stealing credit cards and protect society.

# facial recognition with Husky lens

# Face Detection:





# face learning:





#### facial recognition:



#### **Exercise 1**

• In HuskyLens >> More. There are blocks that change name by ID

HuskyLens name ID	1 of the current algorithm as	"DFRobot "

- If you don't have an idea for giving a name, try the following names: Peter, Sam, Mary
- Since it takes time for HuskyLens to change objects, changing the names of multiple objects at one time may fail
- Therefore use "forever"



#### **Restore changed object names**

#### **Exercise 2**

- Create variable "ID" to store each face ID
- Use Loop >> Repeat Judgment... Execute Blocks to change face names one by one

- Assume HuskyLens sees only one face at a time
- Since the angle of the lens cannot be adjusted in real time, it is better to place different photos in front of the car in turn, or hold the car and place the lens on the face.
- The blocks for controlling the car's motor are in the Stemhub:bit expansion pack:https://github.com/stemhub/pxt-Stemhubbit





#### Answer


start	+ +										
HuskyLens init	ialize I2C u	until su	iccess	+							
HuskyLens swit	ch algorithm	n to Fa	ice Red	ogniti	on 👻						
	+ +		+	+	+	+					
+ +											
orever	+ +		+	+	-+-	+					
HuskyLens req	uest data or	nce and	save :	into th	ie re	sult		+	+	+	+
HuskyLens req	uest data on HuskyLens	nce and ; get a	save : total	into th number	ie re ∙of∶	sult learne	+ d IDs	from	+ the	result	
HuskyLens req set ID • to while ID	uest data on HuskyLens	nce and ; get a 0	save : total	into th number	of	sult learne	d IDs	+ from	+ the	result	
HuskyLens req set ID • to while ID do HuskyLens	uest data or HuskyLens	ace and get a o v of	save : total	into th number	of	sult learne	d IDs	from	+ the +	result	+
HuskyLens req set ID • to while ID do HuskyLens change II	uest data or HuskyLens >>> name ID II D = by -1	nce and get a o of	save : total	into th number + current	of algo	sult learne orithm	d IDs	+ from	+ the + +	result	+ + +
HuskyLens req set ID • to while ID do HuskyLens change I	uest data or HuskyLens I I I name ID II	ace and get a o of	save : total	into th number current	of : algo	sult learne orithm	as	+ from + +	+ the + + +	result	

	forever
	HuskyLens request data once and save into the result
	if HuskyLens check if ID 1 frame - is on screen from the r
	Motor M1 - speed 150 M2 - speed 150
	Motor MB - speed 150 M4 - speed 150
	show number 1
	else if HuskyLens check if ID 2 frame 💌 is on screen from th
	Motor M1 - speed 150 M2 - speed -150
	Motor M3 - speed -150 M4 - speed 150
	show number 2
	else if HuskyLens check if ID 3 frame - is on screen from th
	Motor M1 - speed -150 M2 - speed 150
	Motor M3 - speed 150 M4 - speed -150
	show number 3
	else if HuskyLens check if ID 4 frame • is on screen from th
on start	Motor M1 - speed -150 M2 - speed -150
	Motor M3 - speed -150 M4 - speed -150
HuskyLens initialize I2C until success	show number 4
HuskyLens switch algorithm to Face Recognition 👻	else
	Motor Stop All



# 1.5.3 Lesson 3



Introduction

Objective

HuskyLens Color Recognition Function

What is Color Recognition

# Applications of Color Recognition



### Color Recognition of Huskylens

### Detecting color



### Learning color



#### Identifying color



#### Adjusting the threshold of the recognition border







HuskyLens get a total number of ID 1 frame 🔻 from the result

• Use the following Huskylens blocks to get the height and width of each color block. Area = (length) height × width



- Build "number" store "HuskyLens get atotal number of ID 1 frame from the result"
- Build "area". Store the increased area and use the loop to increase the area of the color block gradually

- ID 1 light up red
- ID 2 light up yellow
- ID 3 light up blue
- ID 4 light up green



```
1When Huskylens detects that red color is closest to the center of the screen, the_

→car starts to move forward

2Until Huskylens detects that the purple squre is closest to the center of the screen,

→ the car stops briefly and starts to reverse

3Keep backing up until Huskylens detects the red square closest to the center of the_

→screen again and the car stops.
```

- First, adjust the angle of Huskylens and look at the red square
- Create variables "direction" Record the moving direction of the car to determine the action of the car
- Create variables "finsih" Determine if the car has returned to red square
- if returned, play mid-range C for one second, and set finish as "true"
- Stop playing the tone when finish is "true"

#### Answer

#### Exercise 1

on start					forever												
HuskyLens initiali	ze I2C unti	il success			HuskyLens r	equest d	ata once	and s	ave i	nto th	ie resi	ult					
HuskyLens switch a	lgorithm to	Color Re	cognition	•	show number	HuskyL	ens get	a tota	1 num	ber of	F ID	1	frame	•	from t	he re	sult

on start		on but	ton A .	press	ed										
HuskyLens initialize I	2C until success	Husky	Lens re	quest d	ata onc	e and	save in	to the	resul	t					
HuskyLens switch algor:	ithm to Color Recognition 👻	set	area 🕶	to 0											
function totalArea	Program is used to calculate the total area of all color patches싄	set while	number	▼ to ( mber ▼	HuskyL	ens ge	t a tota	il numi	ber of	ID 1	fri	ame 🔻	from t	he res	ult
set height - to 0 set width - to 0	Initialize the height and width for obtaining the height and width of the sub-function	do do	all tot hange	alArea	by		Keep o the ini	chan form	ging Iatio	the r n of e	numb each	er by color	/ -1 to bloc	o get k∉	
call getHeight call getWidth	Sum up the total area of the current color block and the area of the new color block	show	nuttber	area 🔻											
set area 🕶 to area	+      height             x      width		1												
function getHeight 🔿	Obtain the height of the color block∉ and a second														
set height ♥ to Hus	kyLens get height • of the ID 1 No. number • frame from the	result													
function getWidth 🔗	Obtain the width of the color block⇔														
set width 🕶 to Husk	yLens get width • of the ID 1 No. number • frame from the r	esult													
· · · · · ·															
* * * * *															+

00.5	tart											forever the second s
R	ikyLen	s ini	tiali	e 120	unti	1 500	cess					Nuskylens request data once and save into the result
Hus	skyLen	s swi	tch al	gorit	hm to	Col	or Re	cogni	tion	•		If NuskyLens check if frame - is on screen from the result then
5							1	1				If NuskyLens get ID * of frame closest to the center of screen from the result ** 1 then
												call backward
func	tion	back	ard (	$\overline{\mathbf{a}}$								On-board Light one + show color red +
	tor R	1 -	speed	-60	) HZ	• •	peed	-60	Ľ.			else if HuskyLens get ID - of frame closest to the center of screen from the result = - 2 then $\Theta$
10	tor A	a <del>•</del>	speed	-60	84	• 5	peed	-60				call backward
												On-board Light one - show color yellow -
												else if Nuskylens get ID * of frame closest to the center of screen from the result = * 3 then 🔿
												On-board Light one - show color blue -
												also if Hushulans not TD * of frame classes to the center of screen from the result . *
												OF COMPANY LIGHT ON THIS COLOR BREEN
												else $\Theta$
												Notor Stop All
												On-board Light one - show color black -



### **Forward function**

HuskyLens get ID +	of frame	closest 1	to the	0.0011.00	r or s	creer	1 from	the	resu			4	- UII2	0	
if HuskyLens get ID	• of fram	e closest	t to ti	he cen	ter of	scre	en fr	on th	e res	ult	= •	1	) t	hen	
Motor M1 - speed 30	M2 <del>v</del> spec	ed 30													1
Motor M3 - speed 30	M4 <del>v</del> spec	ed 30													
On-board Light one - sh	ow color	red 🔻													
ring tone (Hz) Middle C															
else if HuskyLens get	ID 🔻 of f	frame clo	sest t	o the	center	of	screen	from	the	resul		••	2	the	n (
On-board Light one 🛩 sh	ow color	yellow 👻	1.												
ring tone (Hz) Middle E	- A.														
ring tone (Hz) Middle E	ID 🕶 of f	frame clo	sest t	o the	center	of :	screen	from	the	resul	•		3	the	n (
ring tone (Hz) Middle E else if HuskyLens get On-board Light one - sh	ID - of f	frame clo green •	sest t	o the	center	of :	screen	from	the	resul	•	•	3	the	n (
ring tone (Hz) Middle E else if HuskyLens get On-board Light one - sh ring tone (Hz) Middle G	ID - of f ow color	frame clo green <del>-</del>	sest t	o the	center	of	screen	from	the	resul			3	> the	n (
ring tone (Hz) Middle E else if HuskyLens get On-board Light one   sh ring tone (Hz) Middle G ⊕	ID - of f	frame clo green 👻	sest t	o the	center	of :	screen	from	the	resul	•		3	> the	•• (-
ring tone (Hz) Middle E else if HuskyLens get On-board Light one V sh ring tone (Hz) Middle G tse	ID v of f	frame clo green 👻	sest t	o the	center	of s	screen	from	the	resul			3	> the	:n (-
ring tone (Hz) Middle E else if HuskyLens get On-board Light one → sh ring tone (Hz) Middle G ⊕ Ise On-board Light one → show	ID v of f ow color	frame clo green •	sest t	o the	center	of	screen	from	the	resul			3	> the	m (-
ring tone (Hz) Middle E else if HuskyLens get On-board Light one → sh ring tone (Hz) Middle G On-board Light one → show ring tone (Hz) High C	ID v of f	frame clo green •	sest t	o the	center	of s	screen		the	resul			3	> the	m (
ring tone (Hz) Middle E else if HuskyLens get On-board Light one → sh ring tone (Hz) Middle G On-board Light one → show ring tone (Hz) High C Motor Stop All	ID v of f	frame clo green •	sest t	o the	center	of	screen	from	the	resul			3	) the	•n (
ring tone (Hz) Middle E else if HuskyLens get On-board Light one → sh ring tone (Hz) Middle G On-board Light one → show ring tone (Hz) High C Motor Stop All pause (ms) 1000 ▼	ID v of f	frame clo green •	sest t	o the	center	of	screen	l from	the	resul			3	) the	n (

### **Backward function**

	false	the				1				1	1					
1f Huskyl	ens get ID 🕶 d	f frame c	losest t	o the	cente	r of :	scree	n from	the	resul	•	•	1	<b>the</b>	n	
Motor M1 -	speed -30 M2	<ul> <li>speed</li> </ul>	-30												1	
Motor M3 -	speed -30 M4	<ul> <li>speed</li> </ul>	- 30													
La Ruck	ul and get TD =	of from	closet	***									G	Ν.	han	
	yreis ger 10 +	or maile	ciosest		e cen		sure			- 1-55			-	/ "	nen	
On-board Li	ght one <del>-</del> show	color b	lack 🔻													
ring tone (	Hz) High C															
else if	luskyLens get II	• of fr	ame clos	est to	the	cente	r of :	scree	n from	the	resul		•	3	the	• 🖂
On-board Li	ght one <b>+</b> show	color b	lack 🔻													
ring tone (	Hz) Middle G		a													
00015	husbad and set T		ana silas	oct. **	abo						normal.					- 0
else II	uskytens get 11		alle Clus	est to	cille	cente		scree		- une	resul		Ę,	2	Cille	
On-board Li	ght one 🕈 show	color b	Lack 🔻													
ring tone (	Hz) Middle E															
ring tone ( €	Hz) Middle E										-					ļ
ring tone ( ⊕ else	Hz) Middle E													e	9	-
ring tone ( • else On-board Lig	Hz) Middle E	color bla	ck 🕶											e	9	
ring tone ( • else On-board Ligh Motor Stop Al	Hz) Middle E	olor bla	ck •	-										e	9	
ring tone ( € else On-board LigH Motor Stop Al ring tone (Hi	Hz) Middle E	olor bla	ck 🕶											e	9	
ring tone ( 	Hz) Middle E nt one - show of 11 e) Middle C 1800 -	olor bla	<b>ck 🕶</b>											e		
ring tone ( € else On-board Ligh Motor Stop Al ring tone (Hu pause (ms) ( set finish	Hz) Middle E nt one - show o 11 2) Middle C 1890 - 100 (true)	olor bla	<b>ck -</b>											e	9	
ring tone ( € else On-board Ligh Motor Stop Al ring tone (Hz pause (ms) set finish €	Hz) Middle E t one - show o H Middle C 1000 - to true	olor bla	<b>at y</b>											C	•	

# 1.5.4 Lesson 4



Introduction

Objective

HuskyLens tag recognition function

what is tag recognition

# Applications of tag recognition



### Huskylens tag recognition

# **Detecting tag**



# Learning tag



# Identifying tag



# AprilTag









Exercise 3: Self-service supermarket cash register

# Mission 1Identify goods



Mission 2Start and end scan

### **Mission 3goods settlement**



#### Answer



HuskyLe	ens check if ID	is learned from	the result	then							Husky	Lens 1	nitial	lze 120	until	succi	155	
f Husky	Lens check if fra	me • is on scree	n from the	result t	hen						Husky	Lens s	witch a	algorit	the to	Tag F	lecognit	ion •
set heig	ht 🕶 to HuskyLen	is get ID 🕶 of f	rame closes	t to the c	enter of :	icreen 1	from the	result			-		1					
34 <b>b</b>	eight - s	0 then																
14	HuskyLens get ID	• of frame close	st to the o	center of s	creen fro	m the r	esult	)	1	then								
Motor	R1 - speed 60	M2 • speed 60																
Motor	R3 → speed 60	M4 - speed 60	a															
else if	HuskyLens get	ID • of frame c	losest to t	he center (	of screen	from ti	e resul		2	the	Θ							
Motor	R1 - speed -60	N2 - speed -6	0		a													
Motor	N3 • speed -60	M4 + speed -6	0															
else if	HuskyLens get	ID • of frame c	losest to t	he center (	of screen	from ti	e result		- 3	the	Θ							
Motor	R1 - speed 0	M2 - speed 60																
Motor	M3 - speed 0	M4 - speed 60	1 - 1 - I															
else if	HuskyLens get	ID * of frame c	losest to t	he center (	of screen	from ti	e result		- 4	the	Θ							
Motor	R1 - speed 60	N2 - speed 0	a															
Motor	N3 - speed 60	M4 - speed 0	a															
else											Θ							
Hotor	Stop All																	
$\odot$																		
<u>•</u>																		
Č																		

forever		on start
if Dutton A - is pressed then		HuskyLens initialize I2C until success
set total + to 💿 Creat	e a variable to store the status of	HuskyLens switch algorithm to Tag Recognition -
set B pressed + to "false" butto	n B. The default is not pressed	
while B pressed • • • * false*	When the status of the variable if false, k	keep checking
do if button B + is pressed then	whether the button B has been pressed of	or not⇔
HuskyLens request data once and save into t	the result	
if HuskyLens check if ID 1 frame -	is on screen from the result then	ther there are learned tags in the
show string "bottle"	recognized data	results. Each learned tag has a
change total - by 15	corresponding I	D←
	Each time a pr	oduct is identified the variable "total price"
if (HuskyLens check if ID 2) frame +	is on screen from the result then will be added t	to the corresponding selling price of the
show string "biscuit"	producted	the corresponding sening price of the
change total - by 2		
$\odot$		
if HuskyLens check if ID 3 frame -	is on screen from the result then	
show string "knife"		
change B pressed • by 8	the second s	
	a a a a a	
er a pressed ↓ to true		
Δ	fter press the button B, the status of the v	ariable "B
show number total -	THE MILLION DURING AND A LICE AND A LICE V	
n	ressed" will be changed to (true) and brea	ak the loon ←

# 1.5.5 Lesson 5



Introduction

Objective

HuskyLens object classification

what is object classification

HuskyLensDifferences between object classification and other functions

### Huskylens' object classification



### learning objects



### Identification of labeled objects



### Exercise 1Symbol classification



### Exercise 2Location Recognition





**Exercise 3Line tracking** 

Answer

Exercise 1

### The main body of the program

start																							
skyLens initial	ize I2C until s	uccess																					
skyLens switch	algorithm to O	bject Cl	assific	ation 🗸																			
ver																							
skyLens request	data once and	save int	to the r	esult																			
HuskyLens	check if ID 📑	is lea	arned fr	on the	result	t) th	ien																
													16.70										
HuskyLe	ns check 1f ID	2 fr	ane 💌	15 00 5	screen	from 1	the ru	esult	or -	·	skyLen	s check	( 1 <b>f</b> ID	3	frame	• 15	on :	screen	1 fro	n the	resul	"/	
if Husky	yLens get ID →	of fra	e close	st to 1	the ce	nter o	of scr	een fr	on the	result		2	the	n		1	1	1	1	1	1	1	
if Husky call shake_	rlens get ID →	of fra	me close	st to 1	the ce	nter o	of scr	een fro	on the	result	) 	2	the	n	-	-		-		- - -			
if Husky call shake_ show icon	/Lens get ID +	of fram	me close	est to 1	the ce	nter o	of scr	een fr	on the	result		2	the	n		-				-	-	•	
if Rusky call shake_ show icon else if H	vLens get ID +	of fran	ne close frame cl	st to t	the cer	nter o	er of	een fro	om the	result	alt	2	the	n then (	Ð	-	*			-	•		
if Rusky call shake_ show icon else if R call shake_	vLens get ID +	of fram	e close frame cl	est to f	the cer	nter o	er of	een fro	on the	result	alt	2	the 3	n then (	Э	*	-			•	•	- - -	
if Husky call shake_ show icon else if H call shake_ show icon	rlens get ID • • • • • • • • • • • • • • • • • •	of fram	e close	est to t	the cer	nter o	er of	screen	on the	result	ult	2	3 the	n then (	Ð	-	-			- - - -	•	-	
if Husky call shake_ show icon else if H call shake_ show icon	rlens get ID • • • • • • • • • • • • • • • • •	of fran	ne close	est to t	the cer	nter o	er of	screen	on the	result	ault		3	n then (	Ð	-	•			•	•	-	
if Husky call shake show icon else if H call shake show icon else	vlens get ID • vlens get II uskylens get II us	of fra	ne close	est to 1	the cer	nter o	er of	screen	n from t	result	alt		3	n then (	Ð								
if Pusky call shake_ show icon else if H call shake_ show icon else else	rlens get ID • skylens get II uskylens get II	of fra	e close	losest	the cer	nter o	er of	screen	on the	result			3 <b>3</b>	n then (	2								
if Busky call shake_ show icon else if B call shake_ show icon else Potor stop Al e	rlens get ID • s uskylens get II uk 1	of fran	ne close	losest	the cer	nter o	er of	screen	on the	result		2	3 )	n then (	9								





start																	
uskyLens initi	alize I2C unt	il suc	cess														
uskyLens switc	h algorithm t	o Obje	ect Clas	sifica	tion <del>•</del>												
ever																	
					- 14												
iskyLens reque	st data once	and sa	ve into	the re	SULT												
F HuskyLen	s check if ID	4	is lear	ned fro	m the	resu	lt t	hen									
if Husky	/Lens get ID	• of	frame c	losest	to the	e cen	ter of	scr	een fr	om th	e res	ult	= -	2		then	
Motor M1 -	speed 0	M2 🔻	speed (	70													
Motor H3 -	speed 70	M4 <del>-</del>	speed (	0													
														_		\	
else it	uskylens get	10 •	OF TEAL	ne cros	est to	the	cente	• OT	screer	Tron	i the	resul	9		<u> </u>	the	'n
Motor Stop	<b>all</b>																
else if 🛛	uskyLens get	ID 🔻	of fram	ne clos	est to	the	center	of	screer	n from	the	resul	t		4	the	n
Motor M1 -	speed 70	M2 🔻	speed			1		1			1						
MOTOR N3 +	speed	M4 👻	speed	70													_
else								_						-			
else Motor Stop	411																
else Motor Stop	<b>11</b>																

skyLens initia	alize I2C unti	il succ	ess 👘														
skyLens switch	algorithm to	o Obje	ct Cla	ssifica	tion 🔻												
	+ +																
ever																	
uskyLens reque	st data once	and sa	ve into	the r	esult												
HuskyLen	s check if ID	4	is lea	ned fr	om the	resu	lt	then		1	ine is	s stra	ight↩				
if Husky	Lens get ID	▼ of	frame	closest	to th	e cen	ter o	f scr	een fi	rom ti	he res	sult		2	$\mathbf{x}$	then	
Motor M1 -	speed 70	M2 👻	speed	70	Mov	ve fo	rward	Ļ									
Motor M3 🕶	speed 70	M4 👻	speed	70						Lin	e cha	nge t	to top	) left	с »		-
else if 🖉 H	uskyLens get	ID 🔻	of fra	me clos	est to	the	cente	r of	scree	n fro	m the	resu	lt	• •	3	the	an 🗩
Motor M1 -	croad	M2 -	<b>.</b>														
	speeu -40	- m2 - •	speed	60									1				
Motor M3 -	speed 40	MA -	speed	60	Mo	ve to	op lef	t←					e e	*		4. 4.	
Motor M3 🗸	speed 40	M4 👻	speed	60	Mo	ve to	op lef	t←			Line	chan	ge to	topı	right	ę	
Motor M3 🕶 else if 🛛 🖁	speed 40 uskyLens get	M4 -	speed of fra	60 60 me clos	Mo	ve to	op lef	t⊬ r of	scree	n fro	Line (	chang resu	ge to 1t	top	right 4	دیا the	en 😑
Motor M3 v else if H Motor M1 v	speed 40 uskyLens get	M4 - ID - M2 -	speed of fra speed	60 60 me clos	Mo sest to	the	op lef cente	t≓ rof	scree	n fro	Line ( = the	chanı resu	ge to 1t	top	right 4	ह्य the	en 🗩
Motor M3 • else if H Motor M1 • Motor M3 •	speed 48 uskyLens get speed 68 speed 68	M4 -	speed of fra speed speed	60 60 me clos -40 40	Mo sest to Mov	o the	op lef cente p righ	t⊬ rof t⊬	scree	n fro	Line (	chanı resu	ge to 1t	top I	right 4	دا the	en 🗩
Motor M3 - else if H Motor M1 - Motor M3 -	speed 48 uskyLens get speed 68 speed 68	M4 -	speed of fra speed speed	60 60 me clos -40 40	Mo sest to Mov	o the	op lef cente p righ	t↩ r of t↩	scree	n fro	Line (	chan resu	ge to	top I	right 4	دِا the	en 🗩
Motor M3 - else if H Motor M1 - Motor M3 - else	speed 48 speed 48 speed 68 speed 68	M4 -	speed of fra speed speed	60 60 -40 40	Mo sest to Mov	o the	op lef cente	t⊬ rof t	scree	n fro	Line (	chan resu	ge to	top	right	دِي the	en 🗩
Motor M3 • else if H Motor M1 • Notor M3 • else Motor Stop /	speed 40 uskyLens get speed 60 speed 60	M4 -	speed of fra speed speed	60 60 -40 40	Mo sest to Mov	o the	op lef cente p righ	t← r of t←	scree	n fro	Line (	chan, resu	ge to	top I	right 4	دا the	en 🗩
Motor M3 - else if H Motor M1 - Motor M3 - else Motor Stop J O	speed 40 speed 40 speed 60 speed 60 All	M4 -	speed of fra speed speed	60 60 -40 40	Mo sest to Mov	ve to	op lef cente p righ	tç≓ r of tç≓	scree	n fro	Line (	chang resu	ge to	top I	4	← the	en 🗩
Motor M3 - else if H Motor M1 - Motor M3 - else Motor Stop J pause (ms) 1	speed 40 speed 40 speed 60 speed 60 All	M4 -	speed of fra speed speed	60 60 -40 40	Mo est to Mov	ve to	op lef	t≓ r of t	scree	n fro	Line (	chan resu	ge to	top I	right 4	دِا the	

### 1.5.6 Lesson 6



Introduction

Objective

HuskyLens line tracking function

what is line tracking function

#### **Application of line tracking**

- For navigation in public places such as shopping malls or museums
- · For industrial transportation, save a lot of space compared to traditional conveyor belts
- For restaurants, deliver food to guests

# Huskylens' line tracking function

# learning



### Line tracking





• Use different X beginning to decide whether the car should go forward, turn left or turn right



**Exercise 3** 


Exercise 4Autonomous line tracking cars:



Answer

nuskyLens s	switch algori	+	+ +	+					
rever		+	+ +	+	+ +				



#### Program



When the car de	tects the obstacle or reach
function stop	ecute stop function ← Huskylens initialize 12C until success
Motor Stop All	HuskyLens switch algorithm to Line Tracking *
if Ultrasonic Distance(cm) ≤ ▼ 5	then When the car detects the obstacle a set played? • to false
Ultrasonic Light all ▼ show color red ▼	
if played? • • • 'false' the	■ When the variable "played" is Does not play any sound by default
play tone Low A# for 1 ♥ beat	false, play Low A# once 🗠
set played? • to true	After playing the sound, set
if HuskyLens get Y endpoint • of ID	1) arrow from the result 2 - 120) then
Ultrasonic Light all 💌 show color yello	1 Million the series the end esint
if played? 'false' the	• (The tip of the arrow is in the lower half of the screen)
play melody 🞵 🚺 at tempo 1	20 (bpm)
set played? • to 'true'	
$\odot$	<u></u>
· · · · · ·	
forever	<u> </u>
HuskyLens request data once and save into th	e result
if HuskyLens check if ID 1 is learned	from the result then When the car hasn't detected the obstacle
1f HuskyLens check 1f ID 1 arrow •	is on screen from the result then
if HuskyLens get Y endpoint + of	ID 1 arrow from the result < • 120 and • Ultrasonic Distance(cm) > • 5 then
Ultrasonic Light all - show color gr	Set variable "played" as false↔
set played? + to 'false'	
if HuskyLens get X beginning *	of ID 1 arrow from the result > • 190 then
Motor M1 + speed 60 M2 + speed	-40 Turn right↔
Motor M3 • speed 60 M4 • speed	48
else if HuskyLens get X beginning	• of ID 1 arrow from the result < • 130 then •
Motor M3 • speed 40 M4 • speed	<sup>™</sup> Turn left⇔
Motor M1 + speed -40 M2 + speed	60
else	
Notor N1 - speed 80 N2 - speed	Move forward
pause (ms) 100 💌	
else	Θ
call stop V	When the car detects the obstacle or reach the end point, call stop function $\!\!\!\!\!\!\!\!\!\!\!\!\!\!$
A2A Potor Stop All	Chapter 1 Tuteriel suide
+0+ ●	
$\odot$	

# 1.5.7 Lesson 7



Introduction

Objective

HuskyLens object tracking function

What is object tracking function

Application of object tracking

**Event detection** 

**Directing traffic** 

Human-computer interaction

# **Virtual Reality**

# Huskylens' object tracking

HuskyLens's object tracking function can be divided into two parts: learning objects and tracking objects.

# Learning objects



# Tracking objects



# The method of turning off tracking and learning at the same time

# **Exercise 1**







- Use "pause" blocks to control how long the car takes to turn in each direction
- Use the X center to represent the object position



- How do we usually know if we are near an object by our eyes?
- The learning frame of HuskyLens object tracking is preset to 70\*70 square units.
- Learning objects at longer distances is more effective



#### Answer

### Exercise 2

Ì	on st	tart									forever
	Hus	skyLer	ns in:	itial	ize I	2C unt	til s	uccess			HuskyLens request data once and save into the result
	Hus	skyLer	ns sw:	itch	algor	ithm t	to O	bject	Track	ing 🔻	if HuskyLens check if ID 1 is learned from the result and • HuskyLens check if ID 1 frame • is on screen from the result the
×.								÷			if (HuskyLens get: X center • of ID 1) frame from the result > • 180) then Object's X-center located
											Noter M1 + speed 40 M2 + speed 40
											Notor N3 ▼ speed 40 M ▼ speed 40
											pause (ms) 100 -
											Notor Stop All
											else if Huskylens get X center + of ID 1 frame from the result (+ 140) then 💬 Object's X-center located
											Note: MI + speed -40 M2 + speed 40 on the left side of the
											Rotate to left slowly screen
											pause (#s) 100 →
											Noter Stop All
											Object's X-center in the of
											Roter Step All Stop← the screen←
											else
											Notor Stop All

HuskyLen	s initia	alize	12C u	ntil	succes	is		HuskyLens request data once and save into the result	
HuskyLen	s switc	h algo	rithm	to	Object	Track	ing 👻	if HuskyLens check if ID 1 is learned from the result then	
<u> </u>								if HuskyLens check if ID 1 frame • is on screen from the result then	
								set height • to HuskyLens get height • of ID 1 frame from the result	
								if height - < 30 then	
								Motor M1 - speed 40 M1 - speed 40	
								Motor M3 - speed 40 M4 - speed 40	
								else $\Theta$	
								Motor Stop All	
								else	
								Motor Stop All	1



# 1.5.8 Lesson 8



Introduction

Objective

HuskyLens Object tracking function

HuskyLens Advanced Application of Object Tracking

Obstacle detection: HuskyLens object tracking v.s. ultrasonic sensor



### Exercise 2

- You can divide the screen into three areas by referring to Exercise 2 in Lesson 6.
- Adjust the direction of the car first, and then continue to move forward.

# **Exercise 3**

Object distance∈	Obje	ect in screen posi	ition←	Color
ustance	left↩	middle↩	right⇔	light⊄
Far away<⊐	ب ب	ب ب	ب ج	Green←
Moderate	<b>;</b>	<b>,</b>	Ç,	Yellow
Close←	<b></b>	<b>₿</b> <b>↓</b> <sub>4</sub>	¢	Red←

# Car's action when objects are in different positions

# Challenge





Answer

start																					
NuskyLens initialize I	2C until success																				
uskyLens switch algor	ithm to Object T	racking •																			
even																					
iskyLens request data	once and save in	to the re	sult																		
F HuskyLens check	if ID 🚺 is l	earned fr	rom the	e nesu	at	and	•	Huskyl	Lens o	heck	if H	1	fra	<b>ee</b> •	is o	n scr	een fi	rom t	he re	sult	> t
set height <del>+</del> to H	uskyLens get he	ight 👻	of ID (	1	franc	from	the	result													
if height -	- 60 then																				
Motor M1 - speed	30 M1 - spee	d 30																			
Notor M3 • speed	30 M4 - spee	d 30																			
else if neight •		then 😑																			
Motor M1 + speed	-30 M1 - spe	ed -30																			
Motor M3 - speed	-30 M4 - spe	ed -30																			
else		Θ																			
Motor Stop All																					
$\odot$																					
lse																					
Motor Stop All																					

on start																			
HuskyLens initialize I2C until success																			
HuskyLens switch algorithm to Object Tracking																			
orever																			
HuskyLens request data once and save into the re	sult																		
if HuskyLens check if ID 1 is learned fr	om th	e resu	lt	and	-	Husky	Lens	check	if I	1	fra	ame 🔻	is d	on scr	een f	rom t	he re	sult	t
set height ▼ to HuskyLens get height ▼ o	of ID	1	frame	from	the	result													-
if height V ( V 60) then																			
Motor M1 - speed 30 M2 - speed 30																			
Motor M3 - speed 30 M4 - speed 30																			
else if height 🔹 > 🔹 80 then $igodot$																			
Motor M1 - speed -30 M2 - speed -30																			
Motor M3 - speed -30 M4 - speed -30																			
else $igodot$																			
Motor Stop All																			
else																			
Motor Stop All																			
<b>A</b>																			

- The object is large and only one feature of the object is learned. Ensure that the sensor receives the bounce back signal
- It is important to include enough angles and distances when learning with HuskyLens.Since the angle of the lens on the mount is fixed, and there are differences in the appearance of objects seen at different angles and distances, it is important to ensure that HuskyLens can recognize objects in different positions.

# Unlimited execution of the program

HuskyLens check i	if ID 1 is	learned fro	the r	esult	•	4 -		skyLe	ns che	ck if	D (	1	fran	••)	is on	scree	en fro	m the	resul	th
if Ultrasonic Dis	tance(cm) >	• 12	then				n sta	rt												
call forwards	Distance(cm)		the	• 🖂			Husk	yLens	initi	alize	120	until	succ	ess	1					
call backwards							Husk	yLens	swite	h alg	prith	m to	Obje	ct Tr	ackin	e •				
else				Θ		1														
call adjust				1																
•																				
se																				6

forward

nction forwards 🔗												
Ultrasonic Light all	show c	olor gr	een 🔻									
if HuskyLens get	X cente	er 🔻 of	ID 1	frame	from	the res	ult	> -	190		then	
Motor M1 - speed 5	i0 M2 -	speed	30									
Motor M3 - speed 5	i0 N4 -	speed	30									
pause (ms) 200 🔻												
Motor Stop All												
else if HuskyLens g	et X c	enter 💌	of ID	1 f	rame fr	om the	resul		•	130	th	en (=
Motor M1 - speed 3	0 M2 -	speed	50									
Motor M3 - speed 3	10 M4 -	speed	50									
pause (ms) 200 🔻												
Motor Stop All												
else												e
Motor M1 - speed 4	10 M2 -	speed	40									
Motor M3 - speed 4	0 N4 -	speed	40									
$\odot$												

# backward

	ic Light all	. 🔹 st	now col	or re	d 🔻									
if	HuskyLens ge	t X	center	• of	10 1	frame	from	the r	esult	>	- (	190	> the	n
Motor	M1 🔻 speed	- 30	M2 🔻	speed	-50		+	+	-			-	-	
Motor	M3 <b>v</b> speed	- 30	M4 🔻	speed	-50	-								
pause	(ms) 200 🔻	- 4	+	+	+									
Motor	Stop All	+	-	-			+	+	+	-		+	+	÷
lse if	HuskyLens	; get	X cen	ter 🔹	of ID	1 fr	ame f	rom ti	he res	ult	< •	130		the
Motor	M1 🔻 speed	- 50	M2 🔻	speed	-30				1	1	1			
Motor	M3 🔻 speed	- 50	M4 🔻	speed	-30									
	(ms) 200 🔻													
pause														
pause Motor	Stop All													
pause Motor	Stop All													
pause Motor :lse Motor	Stop All M1 - speed	-40	M2 🔻	speed	4 -40		+							

# Move in Place

	Buc are	• sr	IOW CO.	Lor ye	ellow 🗢	+									
f Hu	skyLens ge	t X	center	• of	F ID 1	frame	e from	the	resul	t	• •	190	t	hen	
Motor M1	▼ speed	40	M2 🔻	speed	-40		+	-							-
Motor M3	<ul> <li>speed</li> </ul>	40	M4 👻	speed	-40										
pause (ms	200 🗸	+	+	+	+ +										
Motor Sto	<b>A11</b>														
lse if	HuskyLens	get	X cer	iter 🔻	of ID	1 f	rame f	from	the re	esult	<	• 1	30	the	• 6
Motor M1	▼ speed	-40	M2 🔻	spee	d 40			+							
Motor M1 Motor M3	<ul><li>speed</li><li>speed</li></ul>	-40 -40	M2 • M4 •	spee spee	d 40 d 40	+		+	+	-		-	-	+	
Motor M1 Motor M3 pause (ms	<ul> <li>speed</li> <li>speed</li> <li>200 </li> </ul>	-40 -40	M2 -	spee spee	d 40 d 40	+		+ + +	+ +	-	•	-	-	•	
Motor M1 Motor M3 pause (ms Motor Sto	<ul> <li>speed</li> <li>speed</li> <li>200 </li> <li>All</li> </ul>	-40 -40	M2 - M4 -	spee	d 40 d 40	+ + + +		+ + +	+ + +	•	•	- - -	+ + +	+ + +	
Motor M1 Motor M3 pause (ms Motor Stop	<ul> <li>speed</li> <li>speed</li> <li>200 </li> <li>200 </li> <li>All</li> </ul>	-40 -40	M2 -	spee	d 40 d 40	+ + +	•	+ + +	•			•		•	

# 1.5.9 Lesson 9



Introduction

Objective

Micro:bit Al Smart Car

#### **HuskyLens Advanced Applications**



#### Exercise 1Function combination: color recognition + line tracking

- If the tracking line is easy to go out of bounds, you can use the pause blocks to make the car brake after each movement.
- After pressing the A button to switch to color recognition mode, send two requests for data storage results.
- Initialize the data for the first time and update the data to the new learning result for the second time
- Create a variable id to store the color ID with the highest number of squares.
- Create variable mode to save the current mode of HuskyLens.







# Exercise 2Use different labels to represent different HuskyLens modes



# SD card save/load model



### Method 1 : Manual operation in the second-level menu of each algorithm function



### HUSKYLENS

- A 名称
  - ObjectClassification\_Backup\_1.conf

~

- ObjectClassification\_Backup\_0.conf
- FaceRecognition\_Backup\_0.conf
- ColorRecognition\_Backup\_2.conf

### Method two : use the block module operation



#### Take photos, take screenshots and save to SD card

Exercise 3



Exercise 4Sort the objects first, then switch to the corresponding mode + screenshot.



Answer

haskytens switch algorithm to color recognition 🗸		set mode	• to color mod	le" de	
wind are named data area and says into the area initialize the color resu	ılt⇔ * * *	set id 🔻	to 0		
Store the latest result		HuskyLens	initialize I2C unt	til success	
how making this had one and a total number of langed The form the partit		HuskyLens	switch algorithm 1	to Color Rec	ogniti
			· · ·	+ +	-
the HuskyLens get a total number of learned LUS from the result	n The second second				
if HuskyLens check if ID 1 frame • is on screen from the result and •	HuskyLens check if ]	ID 2 frame ▼ is	on screen from th	e result	then
if HuskyLens get a total number of ID 1 frame • from the result > •	HuskyLens get a total	l number of ID 2	frame 🔻 from the	result	then
set id • to 1		Compare	the number	of color	bloc
else					Θ
set id • to 2					_
●					
€	- a - a - a				
ause (ms) 3000 V					
f (id v = v (0) then Excluding the case where no color is	identified and	the			
set mode • to "line mode" • number of squares of both colors is	about the same	<u>ب</u>			
HuskyLens switch algorithm to Line Tracking	king mode whe	n id is 1 or 2∉			
€ Contraction of the tract	king mode whe				
reven					
	-vocuto this tu	nction only in I			
If <b>mode v</b> = <b>v</b> line mode then	LACCULE LINS TU	iction only in i	ine		
If mode $\checkmark$ = $\checkmark$ line mode then then t	racking mode		ine		
HuskyLens request data once and save into the result	racking mode		ine		
HuskyLens request data once and save into the result	racking mode		ine		
Ine mode     Ine m	arrow from the	result < •	120	then	
Ine mode     ■     Ine mode     Then       HuskyLens request data once and save into the result       if     HuskyLens get     Y endpoint      of the No. id	arrow from the	result < •	120 1	then	
If mode • = • line mode • then HuskyLens request data once and save into the result if HuskyLens get Y endpoint • of the No. id • if HuskyLens get X beginning • of the No. id •	arrow from the	result < •	120 t • 130	then then	
HuskyLens request data once and save into the result if HuskyLens get Y endpoint • of the No. id • if HuskyLens get X beginning • of the No. id •	arrow from the	result < •	120 T	then then	
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint • of the No. id • if HuskyLens get X beginning • of the No. id • Motor M1 • speed 40 M2 • speed 60	arrow from the	result < •	120 t	then	
Image: Intermode       Intermod	arrow from the	result < •	120 T	then then	
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint * of the No. id * if HuskyLens get X beginning * of the No. id * Motor M1 * speed 40 M2 * speed 60 Motor M3 * speed -40 M4 * speed 60	arrow from the	result < •	120 1 • 130	then then	
if       mode • = •       line mode •       then         HuskyLens request data once and save into the result         if       HuskyLens get Y endpoint •       of the No. id •         if       HuskyLens get X beginning •       of the No. id •         if       HuskyLens get X beginning •       of the No. id •         Motor       M1 •       speed       40         Motor       M3 •       speed       -40         Motor       M3 •       speed       -40         else if       HuskyLens get X beginning •       of the No.	arrow from the arrow from the arrow from	result < •		then then	en (•
Image: Intermode       Intermode       Intermode       Intermode       Intermode         HuskyLens request data once and save into the result         if       HuskyLens get Y endpoint * of the No. id *         if       HuskyLens get X beginning * of the No. id *         Motor       M1 * speed       40       M2 * speed       60         Motor       M3 * speed       -40       M4 * speed       60         else if       HuskyLens get X beginning * of the No.       10	arrow from the arrow from the arrow from	result < •	120 120 130 190	then then	en (•
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint * of the No. id * if HuskyLens get X beginning * of the No. id * Motor M1 * speed 40 M2 * speed 60 Motor M3 * speed -40 M4 * speed 60 else if HuskyLens get X beginning * of the No. Motor M1 * speed 60 M2 * speed 40	arrow from the arrow from the arrow from	result < • the result <	120 t • 130	then then	en (
f mode = Ine mode then HuskyLens request data once and save into the result if HuskyLens get Y endpoint + of the No. id + if HuskyLens get X beginning + of the No. id + Motor M1 + speed 40 M2 + speed 60 Motor M3 + speed -40 M4 + speed 60 Hotor M1 + speed 60 M2 + speed 40 Motor M1 + speed 60 M2 + speed 40 Motor M3 + speed 60 M2 + speed 40 Motor M3 + speed 60 M4 + speed -40	arrow from the arrow from the arrow from	result < •	120 t • 130	then then	en (
Ine mode       Ine mode       Then         HuskyLens request data once and save into the result         if       HuskyLens get Y endpoint * of the No. id *         if       HuskyLens get X beginning * of the No. id *         Motor       M1 * speed       40       M2 * speed       60         Notor       M3 * speed       -40       M4 * speed       60         else if       HuskyLens get X beginning * of the No.       10         Motor       M1 * speed       60       M2 * speed       60         else if       HuskyLens get X beginning * of the No.       10         Motor       M1 * speed       60       M2 * speed       40         Motor       M1 * speed       60       M2 * speed       40         Motor       M3 * speed       60       M4 * speed       -40	arrow from the arrow from the arrow from	result < •	120 120 130 > 190	then then	en (
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint * of the No. id * if HuskyLens get X beginning * of the No. id * Motor M1 * speed 40 M2 * speed 60 Motor M3 * speed -40 M4 * speed 60 else if HuskyLens get X beginning * of the No. Motor M1 * speed 60 M2 * speed 40 Motor M1 * speed 60 M2 * speed 40 Motor M3 * speed 60 M2 * speed 40 Motor M3 * speed 60 M4 * speed -40 else	arrow from the arrow from the arrow from	result < •	120 • 130 > • 190	then then	en (-
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint * of the No. id * if HuskyLens get X beginning * of the No. id * Hotor M1 * speed 40 M2 * speed 60 Hotor M3 * speed -40 M4 * speed 60 else if HuskyLens get X beginning * of the No. Motor M1 * speed 60 M2 * speed 40 Hotor M3 * speed 60 M2 * speed 40 Hotor M3 * speed 60 M4 * speed -40 else	arrow from the arrow from the arrow from	result < •	120 t 120 t 130	then then	en (
Ine mode       Ine mode       Then         HuskyLens request data once and save into the result         if       HuskyLens get Y endpoint * of the No. id *         if       HuskyLens get X beginning * of the No. id *         Motor       M1 * speed       40       M2 * speed       60         Notor       M3 * speed       -40       M4 * speed       60         else       if       HuskyLens get X beginning * of the No.       id *         Notor       M3 * speed       60       M2 * speed       60         else       if       HuskyLens get X beginning * of the No.       id *         Notor       M1 * speed       60       M2 * speed       40         Notor       M3 * speed       60       M4 * speed       -40         else       Motor       M1 * speed       60       M4 * speed       -40	arrow from the arrow from the arrow from	result < •	120 120 130 190	then then	en (
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint * of the No. id * if HuskyLens get X beginning * of the No. id * Motor M1 * speed 40 M2 * speed 60 Motor M3 * speed -40 M4 * speed 60 else if HuskyLens get X beginning * of the No. Motor M1 * speed 60 M2 * speed 40 Motor M3 * speed 60 M2 * speed 40 Motor M3 * speed 60 M4 * speed 40 else Motor M1 * speed 40 M4 * speed 40 Motor M1 * speed 40 M4 * speed 40 Motor M3 * speed 40 M4 * speed 40 Motor M3 * speed 40 M4 * speed 40	arrow from the arrow from the arrow from	result < •	120 • 130 • 190	then then	en (
HuskyLens request data once and save into the result HuskyLens request data once and save into the result if HuskyLens get Y endpoint * of the No. id * if HuskyLens get X beginning * of the No. id * Hotor M1 * speed 40 M2 * speed 60 Hotor M3 * speed -40 M4 * speed 60 else if HuskyLens get X beginning * of the No. Motor M1 * speed 60 M2 * speed 40 Hotor M3 * speed 60 M4 * speed 40 Hotor M3 * speed 60 M4 * speed 40 Hotor M3 * speed 40 M4 * speed 40 Hotor M1 * speed 40 M4 * speed 40 Hotor M1 * speed 40 M4 * speed 40 Hotor M3 * speed 40 M4 * speed 40	arrow from the arrow from the arrow from	result < • the result <	120 • 130 • 190	then then	en (
Ine mode       Ine mode       then         HuskyLens request data once and save into the result         if       HuskyLens get Y endpoint * of the No. id *         if       HuskyLens get X beginning * of the No. id *         Motor       M1 * speed       40       M2 * speed       60         Notor       M3 * speed       -40       M4 * speed       60         else if       HuskyLens get X beginning * of the No.       1         Notor       M1 * speed       60       M2 * speed       40         Notor       M1 * speed       60       M2 * speed       40         Notor       M3 * speed       60       M4 * speed       40         Notor       M1 * speed       60       M4 * speed       40         Notor       M1 * speed       40       M4 * speed       40         Notor       M1 * speed       40       M4 * speed       40         Notor       M3 * speed       40       M4 * speed       40	arrow from the arrow from the arrow from from	result < • the result < •	120 1 130 190	then then	en (
If mode + = + line mode + then HuskyLens request data once and save into the result if HuskyLens get Y endpoint + of the No. id + if HuskyLens get X beginning • of the No. id + Notor M1 • speed 40 M2 • speed 60 Notor M3 • speed -40 M4 • speed 60 else if HuskyLens get X beginning • of the No. Motor M1 • speed 60 M2 • speed 40 Notor M3 • speed 60 M2 • speed 40 Notor M3 • speed 60 M4 • speed 40 else Notor M1 • speed 40 M4 • speed 40 Hotor M1 • speed 40 M4 • speed 40 Hotor M3 • speed 40 M4 • speed 40 Hotor M3 • speed 40 M4 • speed 40 Else	arrow from the arrow from the arrow from from	result < • the result < •	120 120 130 > • 190	then then	en (
If mode + = + line mode then HuskyLens request data once and save into the result if HuskyLens get Y endpoint + of the No. id + if HuskyLens get X beginning • of the No. id • Notor M1 • speed 40 M2 • speed 60 Notor M3 • speed -40 M4 • speed 60 else if HuskyLens get X beginning • of the No. Motor M1 • speed 60 M2 • speed 40 Notor M3 • speed 60 M4 • speed -40 else Notor M1 • speed 40 M4 • speed 40 Notor M3 • speed 40 M4 • speed 40	arrow from the arrow from the arrow from from	result < •	120 • 130 > • 190	then then	en (-

on start	on button A - pressed
HuskyLens initialize I2C until success	if the second seco
HuskyLens switch algorithm to Tag Recognition -	if Huskylens get ID - of frame closest to the center of screen from the result - 1 then
HuskyLens request data once and save into the result	set mode - to 1
set mode 🕶 to 🕕	Huskytens switch algorithm to Color Recognition -
	else if HuskyLens get ID • of frame closest to the center of screen from the result - • 2 then $\Theta$
on button 🛛 💌 pressed	set mode v to 2
Motor Stop All	Huskytens switch algorithm to Face Recognition 👻
set mode - to 0	else if (HuskyLens get ID + of frame closest to the center of screen from the result 1) then $\Theta$
Huskylens switch algorithm to Tag Recognition +	set mode 🔹 to 🛐 and a second
	HuskyLens switch algorithm to Object Classification 🕶
	else if (HuskyLens get ID 🔹 of frame closest to the center of screen from the result) 📼 🕢 then $\Theta$
torever	set mode - to (1)
HuskyLens request data once and save into the result	Huskylens switch algorithm to Line Tracking •
call car_movement	

function car_movement	
If mode + + - 3 and + mod	then
if HuskyLens check if ID 1 frame + i	s on screen from the result then
if node 1 then	Color recognition↔
Motor N1 - speed 70 N2 - speed 70	forward↩
Notor N3 - speed 70 N4 - speed 70	
else if made v · · 2 then $\Theta$	Face recognition↔
Notor N1 • speed -70 N2 • speed -7	backward←
Notor N3 • speed •78 N4 • speed •7	
else	Θ
Motor Stop All	
eles if and a star Obj	ect classification
if thiskylens check if ID 2 frame v	s on screen from the nesult, then
Notor M1 + speed -70 M2 + speed 70	1.6
Motor N3 • speed 70 N4 • speed -70	leftward
else	Θ
Notor Stop All	
if HuskyLens check if ID 1 arrow -	s on screen from the result then
Motor H1 * speed 70 H2 * speed -70	
Motor N3 - speed -70 N4 - speed 70	rightward↩
else	Θ
Motor Stop All	
<ul> <li>⊕</li> </ul>	

start			on button A 💌 pressed
wskyLens initialize I2C until success			HuskyLens save - current algorithm data as No. 0 model of SD card
wskyLens switch algorithm to Tag Recognition -			HuskyLens forget all learning data of the current algorithm
uskyLens request data once and save into the result			
set model - to 1			
			on button B - pressed
prever			Huskylens save - current algorithm data as No. 1 model of 50 card
HuskyLens request data once and save into the result			HuskyLens forget all learning data of the current algorithm
if Huskylens check if ID 1 is learned from the	result then		
if HuskyLens check if ID 1 frame + is on sc	reen from the result	then a second	ter a ser a se
if sodel • • • 0 then			on button A+B + pressed
Notor H1 - speed 70 H2 - speed 70			if sodel • • • 0 then
Motor M3 - speed 70 M4 - speed 70			HuskyLens load - current algorithm data as No. 1 model of SD card
else if model • • • 1 then $\Theta$			set model + to 1
Motor M1 - speed -70 M2 - speed -70			else if model • • • 1 then $\bigcirc$
Mator M3 - speed -70 M4 - speed -70			Huskylens load - current algorithm data as No. (1) andel of 50 card.
else		Θ	
Motor Stop All			
•			
lse	Θ		
Motor Stop All			
●			

on start	funct	ion ob	ject cla	ssifi	cation	$\odot$	Function of object classification mode⇔							
HuskyLens clear all custom texts on screen	if	Hus	kyLens c	heck :	if ID	2 i	s learn	ed fro	1 the	result	ther	1		
HuskyLens initialize I2C until success	l	f H	JskyLens	check	k if I	0 2	frame	• is	on sci	reen fi	rom the	result	the	n
HuskyLens switch algorithm to Object Classification 🔻		set n	ode 💌 1	· · ·	ag moo	ie i	Swi	tch to	o tag	reco	ogniti	on m	ode←	1
set mode <b>v</b> to object mode Preset object		Huskyla	ms swit	ch als	vocith		Tag Reco	mitio						
set screenshot • to 'true' classification mode		Ð		cir azę	501 7 611		op neer	Purcro						
	•													
forever														
HuskyLens request data once and save into the result														
HuskyLens request data once and save into the result														
if and a call abject made that														
call object classification														
else if mode = • "tag mode" then 🕞														
call tag recognition														
function tag recognition														
if HuskyLens check if ID 2 is learned from the result then														
if HuskyLens check if ID 1 frame • is on screen from the	result	or 🔻	Husky	Lens o	theck i	if 10 (	2 fra	me • )	is on s	screen	from th	e resul	•	ien
Motor Stop All														
if screenshot v v then														
if HuskyLens get ID + of frame closest to the center of	screen	from the	result		1	the	n	·						
HuskyLens show custom texts ('Hello') at position x 150 y	30 on 1	screen				-								
else if [ HuskyLens get ID * of frame closest to the center	of scre	en from	the resu	lt		2	then Θ	1 - I						
HuskyLens show custom texts ('World') at position x 150 y	30 on :	screen												
•														
HuskyLens take screenshot • and save to 5D card														
set screenshot v to false														
else													(	9
Notor NI - speed 50 N2 - speed 50		-								-			-	2
Notor H3 - speed 50 H4 - speed 50														

# 1.5.10 Lesson 10



#### Introduction

#### Objective

### **Conclusion Micro:bit AI Smart Cart**

#### **Competition 1**

- Face: micro:bit board showing smile symbols
- Color: Lights up one RGB on-board light or ultrasonic light (six in total, each a different color)
## **Rules**

The competition is a score system, there are two rounds, the highest score.
→of the round will be the final score
when the car successfully identify the face, +5 points
Whenever the car successfully identifies a color, +2 points
If all RGB on-board lights and ultrasound lights are successfully lit, +6.
→point
If the car does not go to the end point within the time limit, the score.
→will be reduced by half

## **Competition 2**

## **Rules**

The competition is a score system, there are two rounds, the highest score.
of the round will be the final score
Whenever the car successfully identifies a tag, +2 points
Each time a single word in the HuskyLens screen is successfully linked into.
a phrase, an additional +5 points will be awarded.
If the car does not go to the end point within the time limit, the score.
will be reduced by half. The fastest group to complete will +6 points, the second.
group will +4 points, the third group will +2 points.

