mBlock

kids maker rocks with the robots

Authors Liao Yuqiang
Zhao Tongzheng

It's the most attractive and amazing educational robot so far I have try.
—— LIE LENG BOON

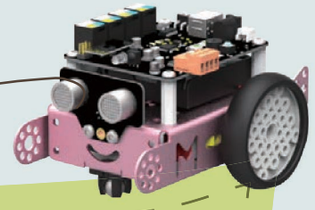
You did what I was looking for so many months: the best of two world, graphical and code.
—— Cant Sébastien, STEM teacher in France

This little robot has a lot of features to use: leds, IR, buzzer, and so on… I would like to use it more and more. I want two for my daughters firstly!!!
—— Vincent MARECHAL from Lycee Stendhal Milan

mBot in combination with mBlock is probably the best thing you and your team have done. With mBlock you can look at the code, which is very important for us teachers.
—— Christian Prim from Switzerland High School Zurich North

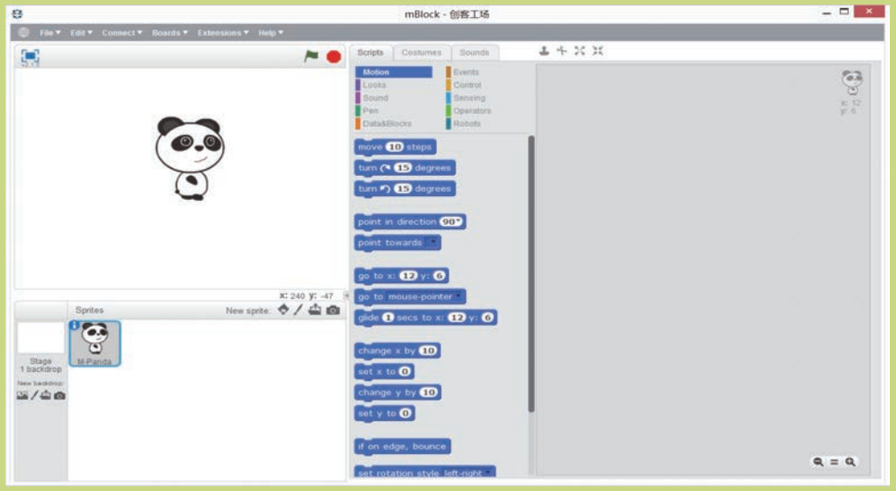
mBot is much more than a robot. I wish it were the "key to a great world" in the "small hands of kids".
—— Teddy Donat, physicist MSC, STEM teacher in Spain

Pre- Learning Preparation

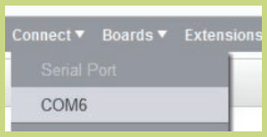


Hello everyone, welcome to the world of mBot robot!

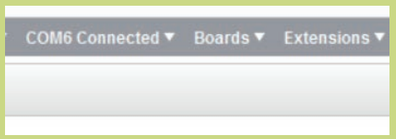
We will experience the interaction between the physical world and software through learning the course. The course consists of three parts: mBot robot, mCore main board and mBlock (You could download them with following address: http://mblock.cc/download). mBlock is a software developed on the basis of Scratch 2.0. It could control mCore main board of mBot and achieve the corresponding functions. Here is the main interface of mBlock:



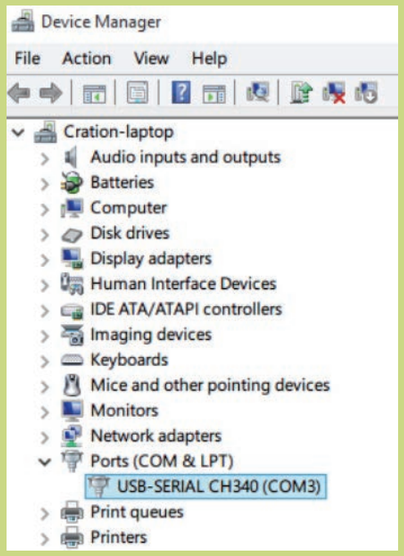
Therefore, only when mBlock and mCore are connected together, can we control mBot by mBlock. Then, how could we connect mBlock to the robot mBot? First, we connect the USB cable to both the computer and the mCore main board. Open mBlock software and select the appropriate COM port:



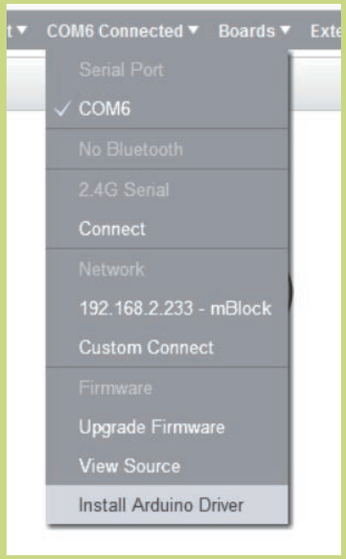
It will display "COMx Connected" after clicking:



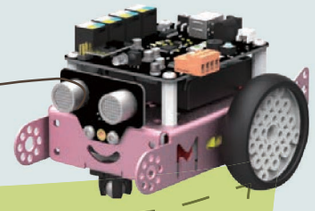
COM ports vary in computers. How could we find the COM port number of mBot? Open Device Manager and expand Ports (COM and LPT) to view:



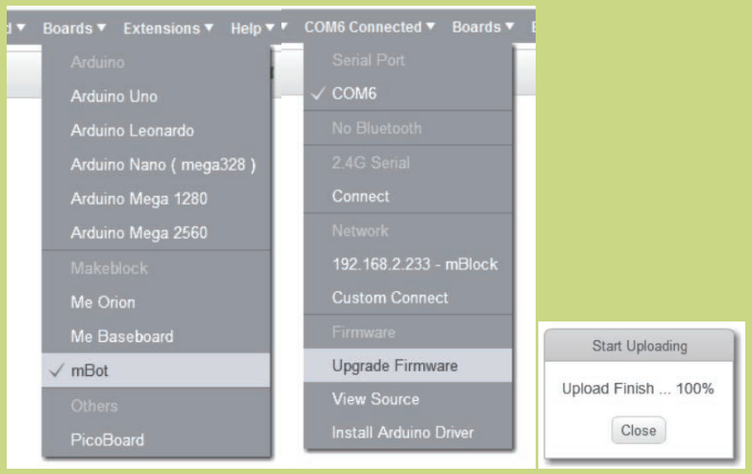
If you can't find the port number in the Device Manager, please try to install Arduino driver. Then find and select the port:



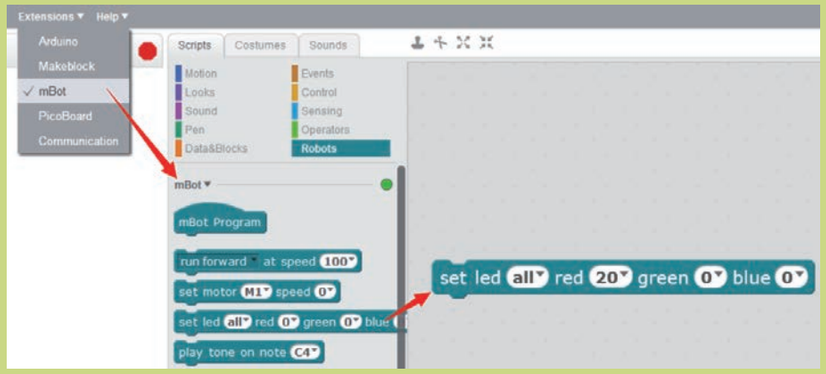
Pre- Learning Preparation



mBlock is successfully connected to mBot! But they are not able to communicate with each other yet, because communication demands the communication protocol as a basis. Let us install this protocol on the motherboard! First, select the correct main board and then select "Upgrade Firmware":



After installation, mBlock is able to communicate with mBot! Come on! Test it! Turn on the small lamp on the robot mBot:

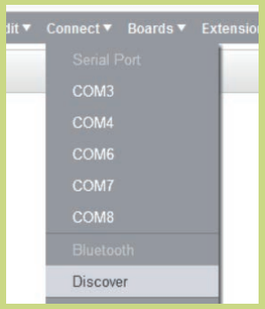


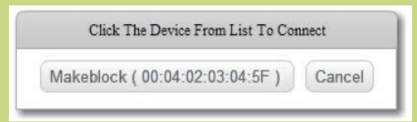
Click this block to set LED color. The little light of mBot will turn red. If all are set to 0, the light will go out.

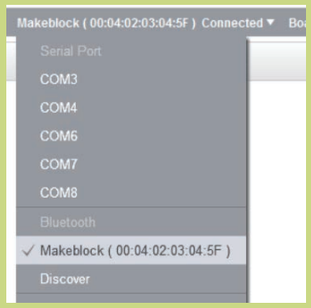
What should we improve if we want the car to move on the ground? Yes, the USB cable is the answer. It restricts the travel distance of mBot. We can solve this problem with the Bluetooth module or 2.4G module.

Bluetooth

First, open the computer's Bluetooth switch (or use a Bluetooth adapter) and disconnect the previous COM (just click on COM6). Enter mBlock and select Discovery. Then the Bluetooth list pops up. Then, you select the corresponding Bluetooth:



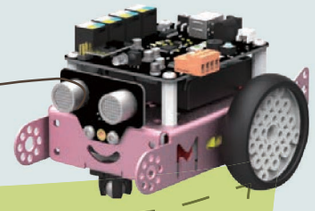




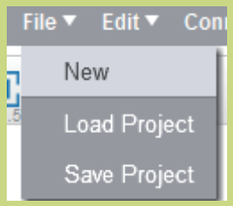
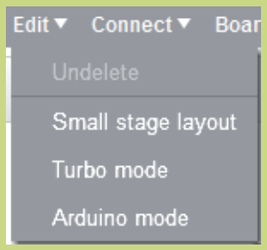
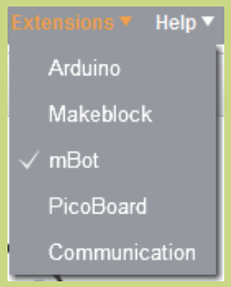
2.4G

It is even easier for 2.4G. First, insert the 2.4G adapter (receiver) into your PC, pair it with the 2.4 module of mBot according to the user manual. After successful pairing, click "Connect".

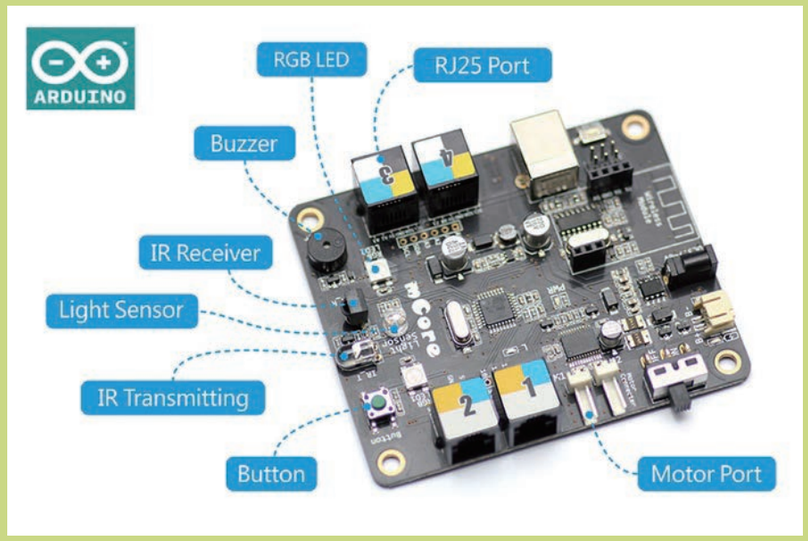
Pre- Learning Preparation



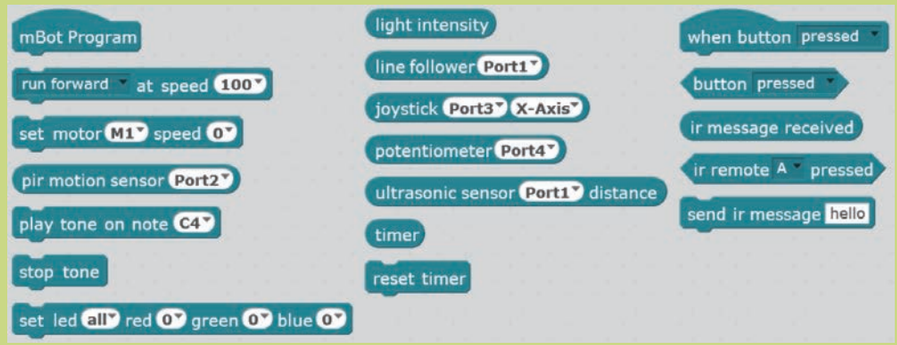
Now let's see the common-used functions in the menu.

Option	Function
	New : create a new project; Load Project: Open an existing project; Save project: Save the current project to local file. (Note: the filename extension is ".sb2")
	Undelete: restore the previous deleted script; Small stage layout: Hide the stage, hence to make the editing area of script bigger; Turbo mode: Accelerate the redraw speed of mBlock stage Arduino mode: Convert the block script of mBlock to Adruino program and upload to the Arduino main board to realize off-line operation.
	Extension menu is related tomodule. The option you choose in Extension will affect the blocks shown in the Robots module. Which means, different option represents different blocks in Robot module. Arduino: including blocks compatible with Arduino platform; Makeblock, mBot: Specific main board from Makeblock; PicoBoard: Control and testing board from Scratch; Communication: Provides the function of LAN communication.

Now, Let's check what sensors that mBot's main board contains :



Choose "mBot" from the Extension menu, then you will see various blocks which we will be using to control mBot:



OK! Everything is ready, just save it. Begin your mBot journey!

Chapter 1 The Mouse Loves Apples

Mouse, the small mouse, likes to eat apples very much. Today it breaks into the mBlock world and there is a sweet apple just in front of it. Can you help Mouse eat that apple? Come on, help Mouse with the mCore baseboard at your hand .

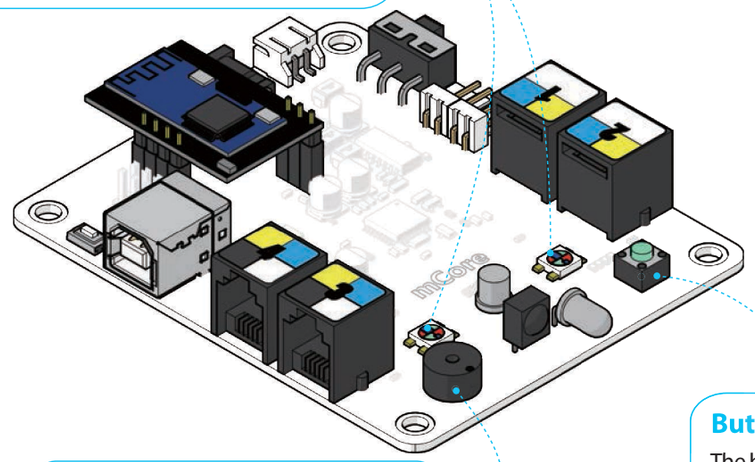
Learning Objectives

1. Learn to control LED lights
2. Learn to control the buzzer tone

Electronic Moduless

RGRGB LED lights

The RGB LED light is a output light source device. It can output light of any color through the tri-color of the red, the green and the blue.



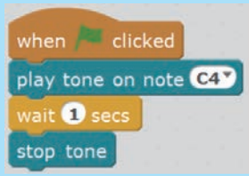
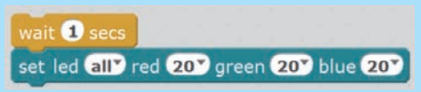
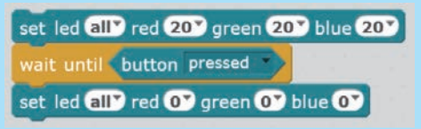
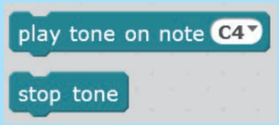
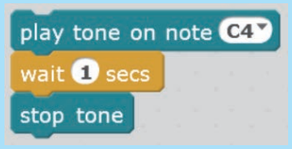
Button

The button is an input device. It usually defines different functions depending on the application scenario.

Buzzer

The buzzer is an audio device, but also an output device. It can control the different output frequencies so as to give a different tone.

Knowledge Points

Building Blocks	Instructions	Example
	The starting point of event and triggering ways of the program. The function of the script in the left figure is to start the program when you press the green flag.	
	Wait one second. The number can be an integer or a decimal number.	
	Before "button is pressed", always wait. The blue blocks in the left figure can be replaced by other hexagonal blocks.	
	Set the color of RGB LED lights. Each color range is from 0 to255. If you want to turn off the small lights, set all three colors to zero.	
	Buzzer can output the tone from C2 to D8. It needs to add the stop playing block in the left figure to make the buzzer stop playing.	

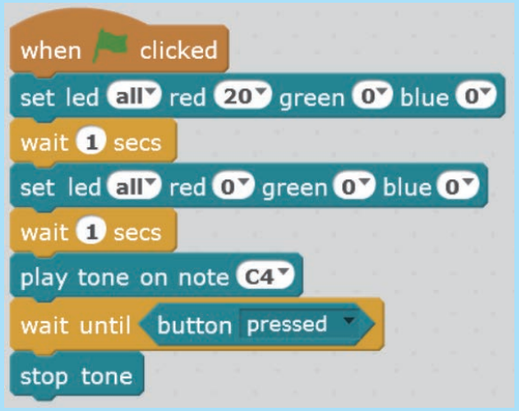
Programming Structure

Programming structure	Thinking Process
Sequence structure description: The script begins running from the first block, followed by all the other blocks executing in order. This is the sequential structure. The diagram on the right is a standard sequence structure. After the program starts, it implements three blocks in turn and finally ends. The sequential structure is the basis for the running way of the program.	Begin ↓ Instruction block A ↓ Instruction block B ↓ Instruction block C ↓ End

Tips

The value set in the blocks alters the state of the electronic module. For example, when setting the tone as C4, the buzzer will continue to make a sound and even if you stop the program, the sounding state won't be changed. The correct usage is to add a stop-playing block with playing time when sound is not required.

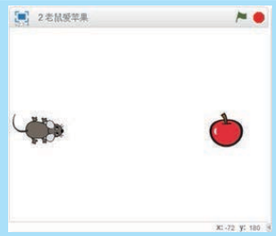
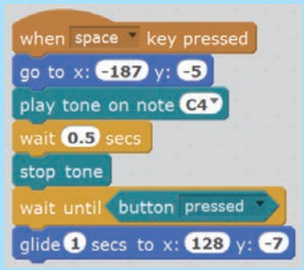
Try it

Try it	Flow chart
	Challenge learning flow chart and please write down the flow order in the left figure. 1.Start the program (when the green flag is clicked) 2.Red light is turned on one second 3.Light is turned off one second 4. 5.

Example Expansion

Let's make the first mBlock program now! In this program the mouse wants to eat the apple so it contains two sprites while achieving interactive control effects through mCore baseboard.

When the program starts, mCore utters a sound and wait for you to press the button on the mCore. Once you press the button, the mouse moves toward th e apple. The mCore sounds again when the mouse hits the apple.

The mouse loves apple		**Instructions** Mouse continue to move forward. When it comes to Apple, the mCore makes a sound.
Mouse sprite		**Let the mouse move towards Apple's position** Press the space bar to start the program->let the mouse move to the initial position-> Play tone C4->wait0.5 seconds-> Stop Playing ->before pressing button on the mCore, wait->let the mouse move to the specified location
Apple sprite		**Make a sound when the mouse eats the apples** Press the space bar to start the program-> Wait before you hit the mouse-> Play mCore buzzer tone with C4->wait0.2 seconds-> Change the tone as F2->wait0.2 seconds-> Stop playing sound

Exercise

1. Use the buttons on the mCore to change the color of the small LED lights, hence to create a gorgeous lighting effect. Try to make it emit purple, yellow and white light.

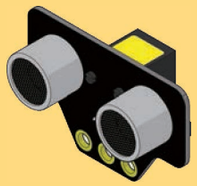
Chapter 2 Pick Apples

Little mouse Mice found a fruitful apple tree today . Now is the ripe season. Apples on the trees constantly fall when blowing in the wind. Mice hopes to find a container to hold these apples. Can you help Mice collect more apples?

Learning Objectives

1 . Learn how to control and apply the ultra sensor

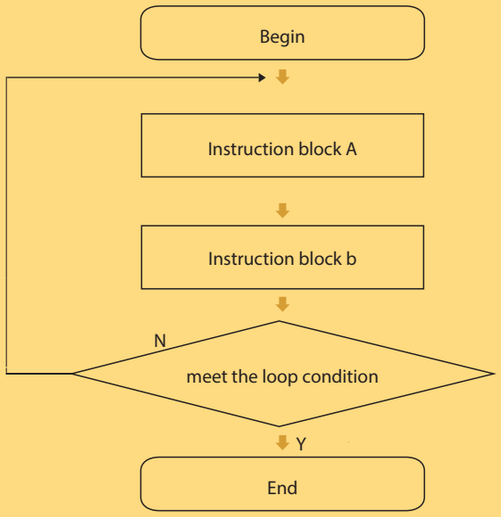
Electronic Modules

Category Name	Function	Instructions
Ultrasonic sensors 	Ultrasonic sensor is an input device for detecting distance. It has two "eyes", one of which is to emit ultrasonic waves while the other one is responsible for receiving the signal bounced back after detecting an obstacle. Hence, it realizes the goal of measuring distance. Detection range: 3cm-400cm; Detection angle: 30 °.	Ultrasonic module is marked with a yellow label so you need to connect it to the interface with a yellow logo on the motherboard.

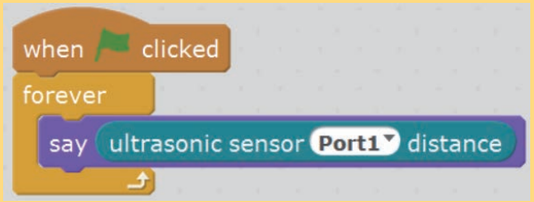
Knowledge Points

Building Block	Instructions	Example
Control repeat 10	Repeat encased script for specified times, and then continue to perform the following script.	repeat 3 play tone on note C4 wait 1 secs stop tone play tone on note E4 stop tone
repeat until	Repeat until the condition is satisfied (the condition is within the space) When the condition is not satisfied, the wrapped script will constantly run. If met, the program will move on to the following scripts.	repeat until button pressed play tone on note C4 wait 1 secs stop tone
forever	**Constantly repeat:** The encased script keeps running and cannot be terminated.	forever set led all red 10 green 10 blue 10 wait 0.1 secs set led all red 0 green 0 blue 0
Robots ultrasonic sensor Port1 distance	Ultrasonic sensor block: to give feedback on the distance between the ultrasonic sensor and any obstacles in front of it.	forever say ultrasonic sensor Port1 distance

Programming Structure

Programming Structure	Thinking Process
Loop Structure (Cycle Structure): Loop structure is the structure which repeats its contained execution script. As the right flow chart shows, instruction block A and B are called as loop body. If the loop condition is false, re-execute the loop, otherwise the loop ends. When you need to repeat the same script, we often use the loop structure in programming.	

Try it

Scripting	Flow Chart
	Challenge yourself to learn about flowchart. Write the flow order in the left figure and indicate which building blocks are the loop body.

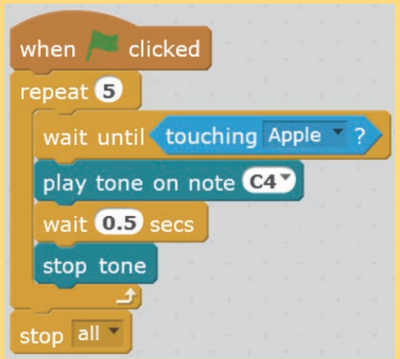
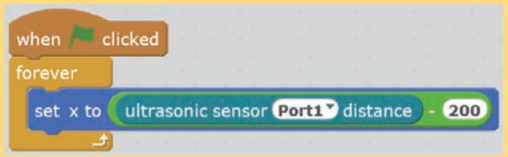
Scripting	Flow Chart
	Try to execute the script on the left side and record the changes of mCore.

Example Expansion

We have learned the loop structure. Can you help Mice catch apples through combining the sequential structure learned before?

We just need to take advantage of the characteristics of repeating to control the movement of Mice through ultrasonic information.

Catch the apple		**Instructions** When the mouse moves forward and hits the apple, mCore utters a sound.

Apple sprite		**Keep apples falling** Repeatedly execute its contained script -> Display Apple -> Specify the location that the apple appears -> Repeatedly move the apple down until it hit the edge -> After it hits the edge, hide the apple
Mouse sprite		**Stop the program after the mouse collect 5 apples** Wait for the mouse to hit the apple -> play a tone for 0.5 second -> stop the program after 5 apples are collected.
Mouse sprite		**Use mCore together with the ultrasonic sensor to control the moving of the mouse.** You must use values that are through constantly repeating and updating the ultrasonic sensor to make the position of the mouse dynamically change.

Exercise

1. Try to use loop structure to control RGB light on the baseboard. Let them flash once every second and indicate which building blocks are the loop body.

2. Try to make the buzzer sound as C4 in 10 times, E5 in 20 times, B6 in 30 times, and then keep repeating. Can you distinguish these loop bodies inside it? Try it.

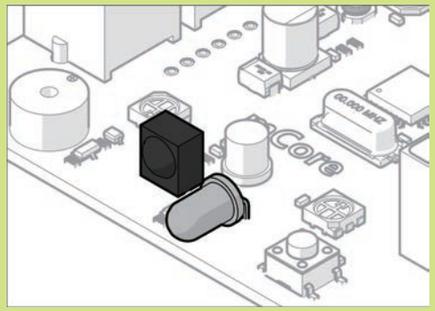
Chapter 3 The Arithmetic Challenge

Little Monkey randomly displays a number. It required the bat to collect the same number of points within 30 seconds. Can you help the bat complete the little monkey's task?

Learning Objectives

1. Use of mathematic operation
2. Use variables

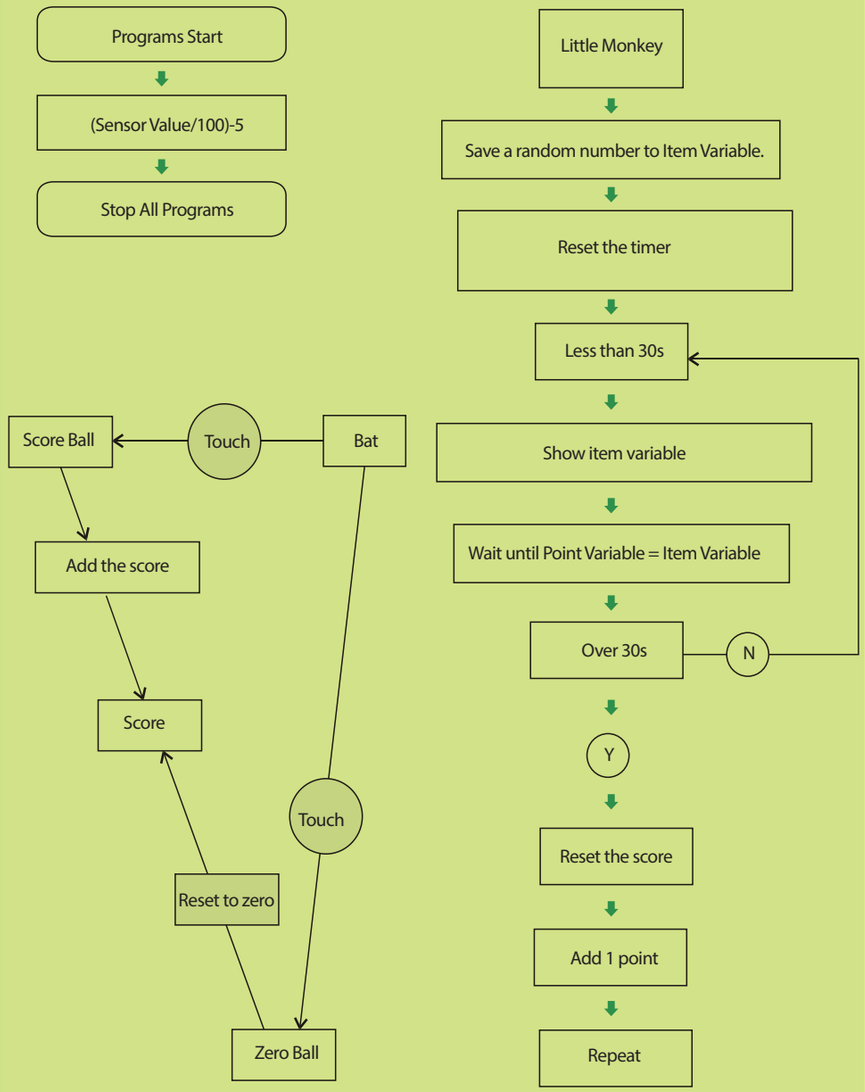
Electroni

Category Name	mCore Baseboard	Instructions
Infrared remote control		Send information via infrared to the infrared receiver module. Then process the received date in the program.
Infrared Transceiver Module		The mBot infrared transceiver module is able to receive and send information. The transmitted information can be numbers and text.

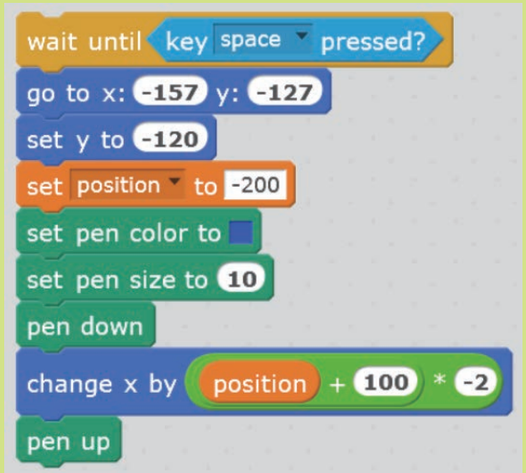
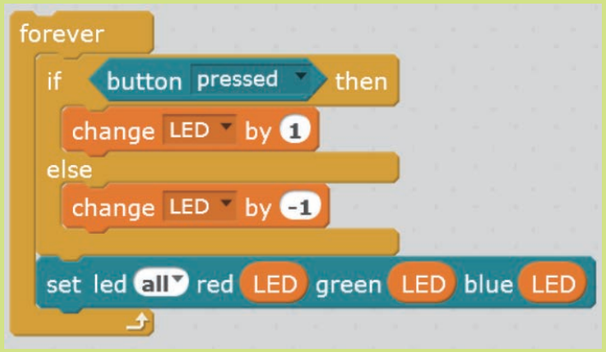
Knowledge Points

Building Block	Instructions	Example
Operators + - * / < = >	Four operations include addition, subtraction, multiplication, division. You can fill in the values or variables. Comparison operators can be used to compare the value with the variable, the variable with the variable, the value with the value. The input of the sensor can also be used as a variable.	joystick Port3 X-Axis / 10 The value of the joystick is divided by 10 wait until ultrasonic sensor Port1 distance < 30 Wait until the distance that the ultra sensor detects is less than 30cm, then move on to the following building blocks wait until point = problem Comparison of the two variables are done to check whether they are equal. Wait if they are not equal
pick random 1 to 10	Random number blocks. You can fill in the values or variables.	forever point in direction pick random 0 to 359 wait 1 secs Face to a random direction once every 1 second
Data&Blocks Make a Variable variable set variable to 0 change variable by 1 show variable variable hide variable variable	Variable is a container for storing data. Its value can be freely modified according to the needs.	set variable to ultrasonic sensor Port1 distance Sensor data will be saved in the variable set variable to 0 repeat 10 set variable to variable + 1 wait 1 secs show variable variable The variable is incremented by 1 once every second
if then	If <condition> then If the condition is satisfied, the internal script is executed	if ir remote → pressed then point in direction 90 If you press the right key on the infrared remote controller, the sprite will move towards the right direction

Programming Ideas

The idea Description	Thinking Process
Math games. It collects comput-er-generated points. Four sprites 1. Score ball, to increase score 2. Bat, operate by infrared remote controller 3. Small monkeys, compare scores collected by bats Zero ball, set score as zero	

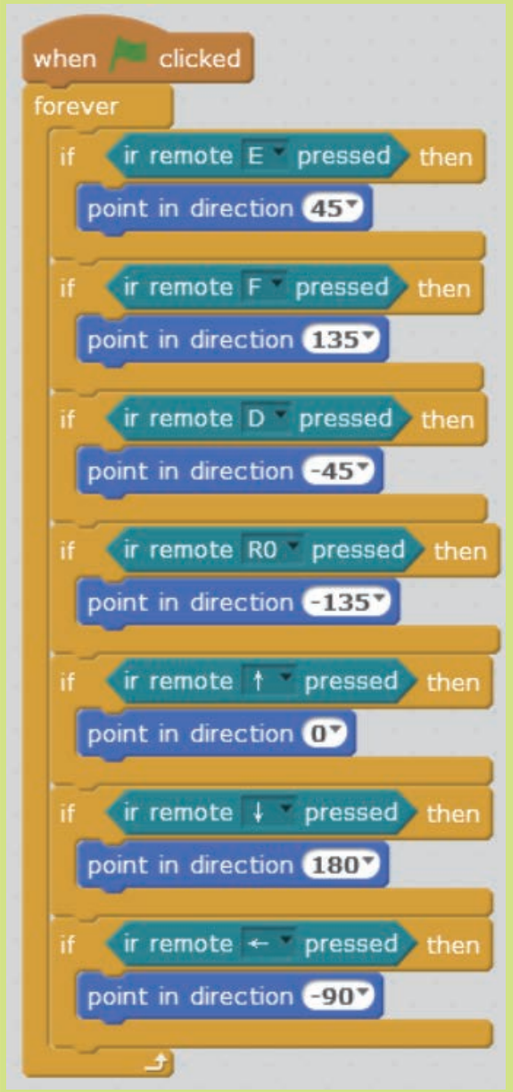
Try it

Scripting	Flow Chart
	Try to learn the flowchart and write the process in the left figure.
	Try to run the script on the left side and record the change of the mCore.

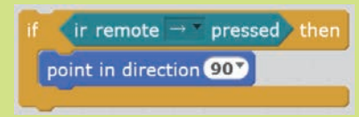
Example Expansion

The mathematic operation is very important for the program. The chapter will be very helpful for those learners who are not really good at math! Come and learn together!

Use an infrared remote controller to operate the bat

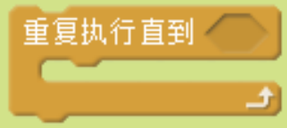


The bat automatically flies forward



Use If..Then building blocks to change the direction of flight

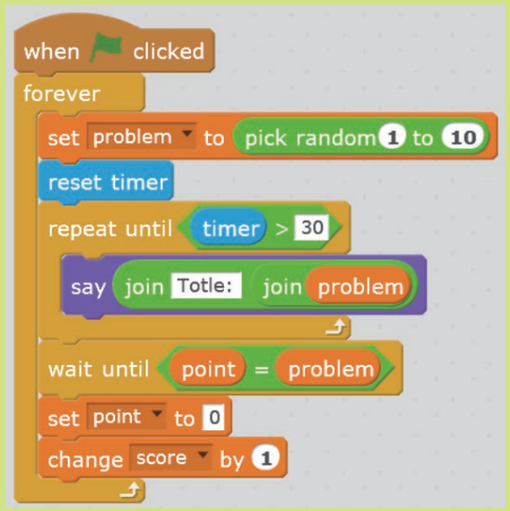
The score ball randomly appears on the stage and moves. If it hits the bat, it will modify the responding variables.。

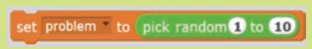


Randomly-appear ball will make the game more interesting

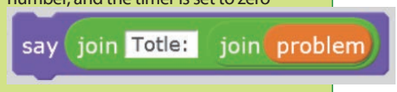
Different balls represents different points

Compare once every 30 seconds

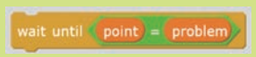




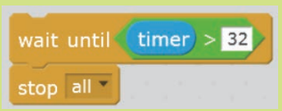
The title variable is set to a random number, and the timer is set to zero



Within 30 seconds, the little monkey will always say the number of points to be

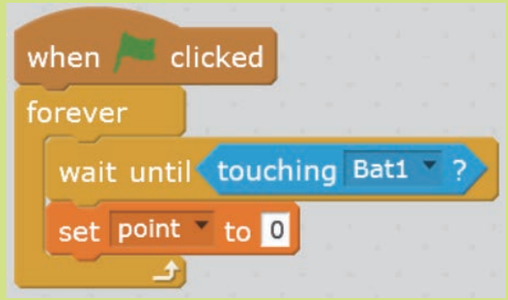


If the values of the two variables are equal, the execution number of points is set to zero and you score 1 point



If the time exceeds 32 seconds, it indicates that the subject is not the same number of points. It will stop all programs and the game is over

The zero ball can set the point number to zero and collect points from the beginning



The point number becomes zero when the bat hits it

Tips

If the turning mode of the sprite is set as 

, it will affect turning of the sprite. Test it by yourself.

Exercise

1. Try to make a small ball that can reduce the number of points.
2. Let zero ball move and it has a function of comparing the number of points.

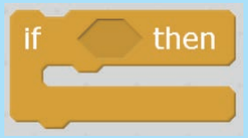
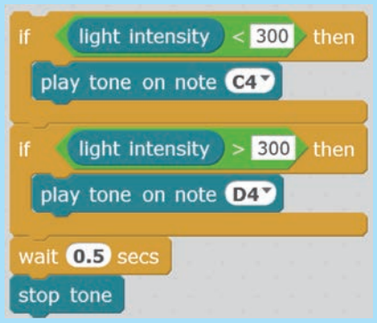
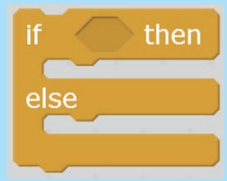
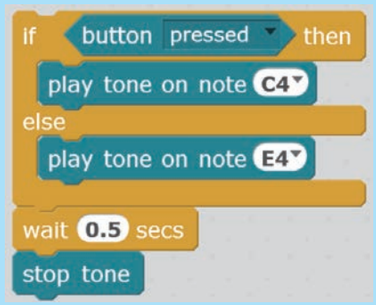
Chapter 4 Guess

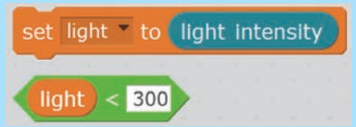
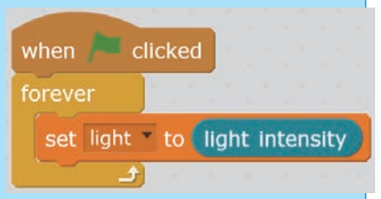
We often play game of guessing coin with friends. So how do we use mCore to play this game? In this chapter, we'll do it .Let's see who is the best guesser!

Learning Objectives

1. Learn the branch structure
2. Usage and comparison of variables

Knowledge Points

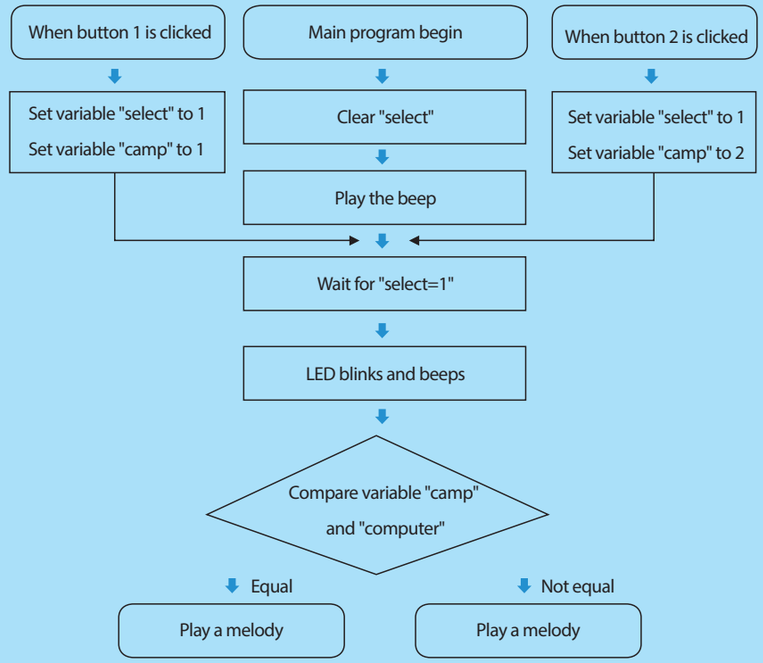
Build Blocks	Instructions	Example
	The "if .. then block" is a single branch structure. It needs to be filled with the hexagon blocks. The contained script will be executed if the condition is met. If not, the script will be skipped . If we judge a number of conditions in the program, we need to use several "if .. then "blocks.	
	The "if .. then otherwise block" is a double branch structure. The upper part will run if the conditions are met. If not, the lower part will run.	

	Typically, the value of the sensor will be saved into a variable. So the following script will not misjudge the sensor values by the time difference caused by the sequence structure.	

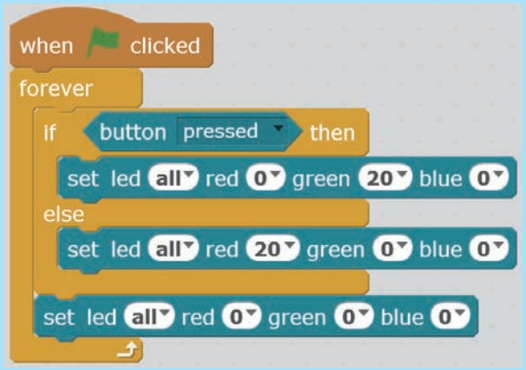
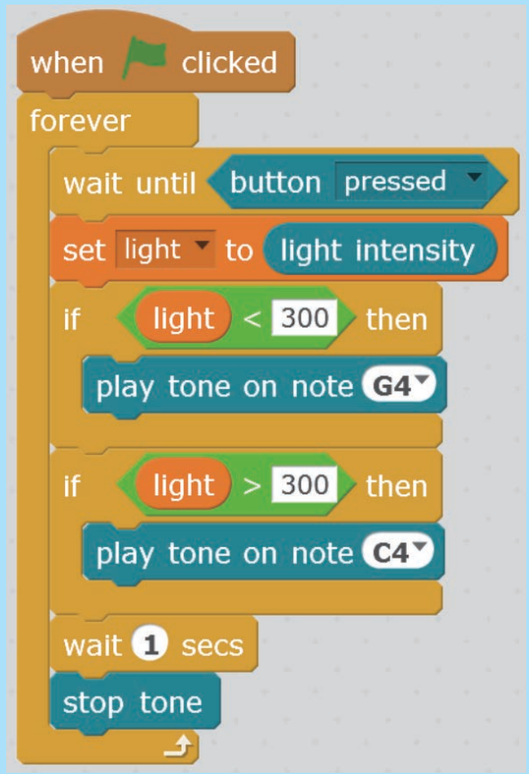
Tips

Because it is often for the programs to repeatedly judge whether the value of the sensor satisfies certain conditions, branch structure and loop structure are often used together.

Programming Ideas

The Idea Description	Thinking Process
Branch structure Branch structure is used for determining a given condition and controlling the flow of the program based on the judgment result.	

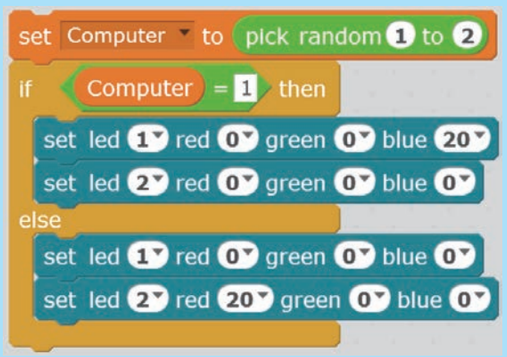
Try it

Scripting	Flow Chart
	Challenge yourself to make a flowchart. Write the flow chart of selection structure on the left graph.
	Try to run the script on the left side. Record change of mCore.

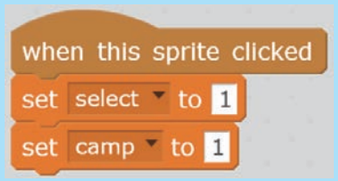
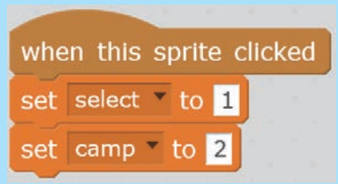
Example Expansion

Let's play a game: Click the button on the stage. Two RGB lights on the main board will flicker back and forth. One of them will light in the end. Let's guess and see who will be the best guesser!

Guess		**Operating Instructions** First, select LED lamp button 1 or 2 on the screen. After waiting for a while, computer will compare if the master control board is identical to the selection of player. If it is identical, it means the guess is right and a melody will be uttered; if it is not identical, it means the guess is false and a short music will be given off.
Stage Background		**Start the main program** If the variable select is set at 0, the player will be deemed that no selection has been made. If a tone is uttered, it means that it is ready for player to select.
		It is ready for player to select button 1 or 2 on the stage. After selection, with the variable select=1, the block will not obstruct the script any longer.

Stage Background		LED lamp will flash for 10 times and utter different tones.
		Make the computer generate a random number and store it in the variable computer, then light up the lamp selected by the computer with a branch structure.

Stage Background		After waiting for 2s, the program will compare the selection of player with that of computer. If camp =computer, it means the player has a right guess and a melody will be uttered; otherwise, LED lamp will be put out and a tone indicating a error will be uttered.

Stage Background		**Script of button 1** Set the variable "select" to 1 (start the main program) Set the variable "camp" to 1 (player's choice)
		**Script of button 2** Set the variable "select" to 1 Set the variable "camp" to 2

Tips

In the button sprite, we use the building blocks of "when the sprite is clicked". It means that no matter whether the green flag is started, the script below it will be executed when the sprite is clicked. Sometimes this design is not appropriate. We can solve this problem later after we learn the logical operator.

Exercise

1. Use the branch structure to display the number of pressing the mCore button within 5 seconds on the stage.
2. Use branch structure and values of light sensors on the mCore to switch the different stage background.

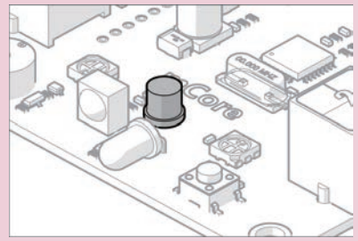
Chapter 5 Defend the Island

The treasure left by the head of the pirates is hidden on a small island. When the pirates get the news, many of them want to snatch the treasure and attack the island. Great heroes, come to the island to defend it!

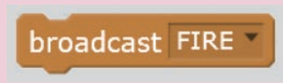
Learning Objectives

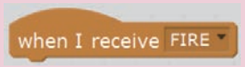
1. Learn the broadcast command
2. Learn to use clones

Electronic Modules

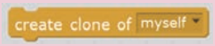
Category Name	Legend	Instructions
Light sensor		It can generate value by sensing the brightness of ambient light and.

Knowledge Points

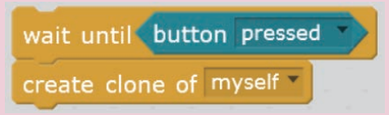
Building Blocks	Instructions	Example
	The broadcast blocks broadcast a sender's message to all the sprites (including the sprite itself). Thereby it reminds the recipient to perform certain actions. The Figure on the right is the menu displayed when right- clicking the block. Among them, "show sender" and "show recipient" can show the associated sprites. So it allows users to quickly find the corresponding relationship.	



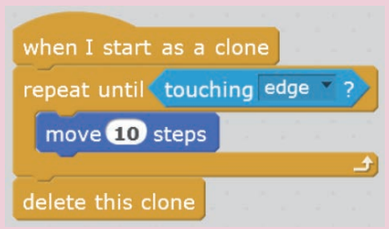
This building block is the recipient of the message. It is also the triggering block of script.

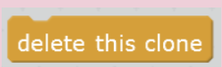


Clones can reproduce sprite that is exactly same as the original sprite. Its script is same as the original .This feature prevents the production of many sprites with the same script.



After its creation, the clone starts by this building block. It is different from starting of the original sprite.





Delete the unnecessary clones. Excessive clones will lead to the program delay and affect the execution speed.

Programn

The Idea Description

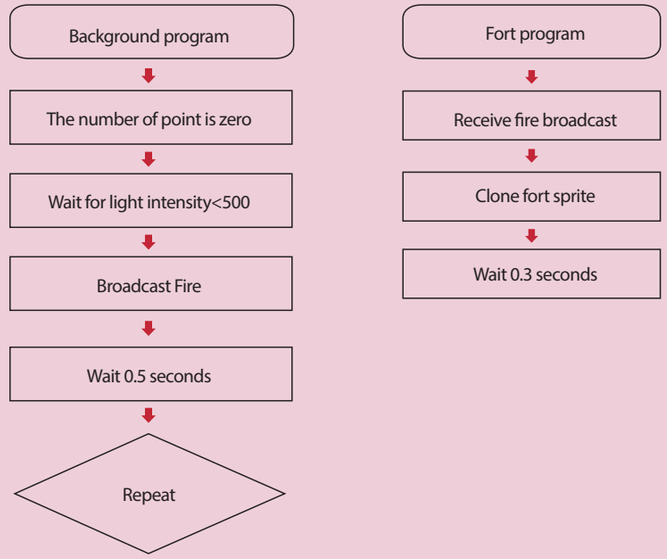
Broadcast Function

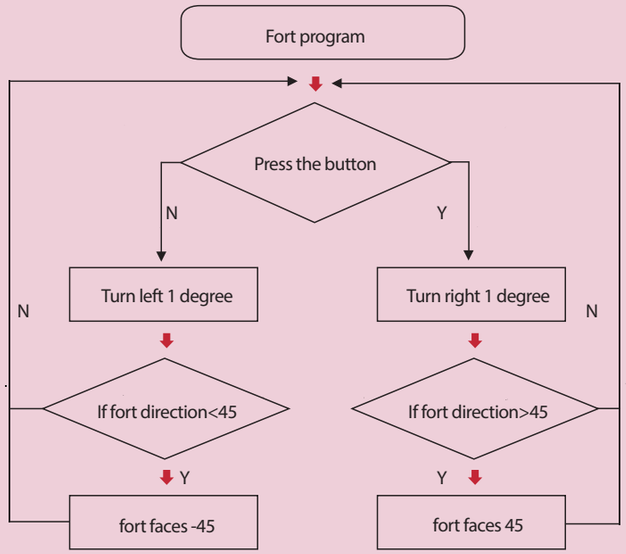
After broadcasting a message, the message will be sent to all the sprites (including the sprites themselves). The sprites use "when receiving" building blocks to receive messages. If messages sent and received match, the "when receiving" building blocks are triggered to run.

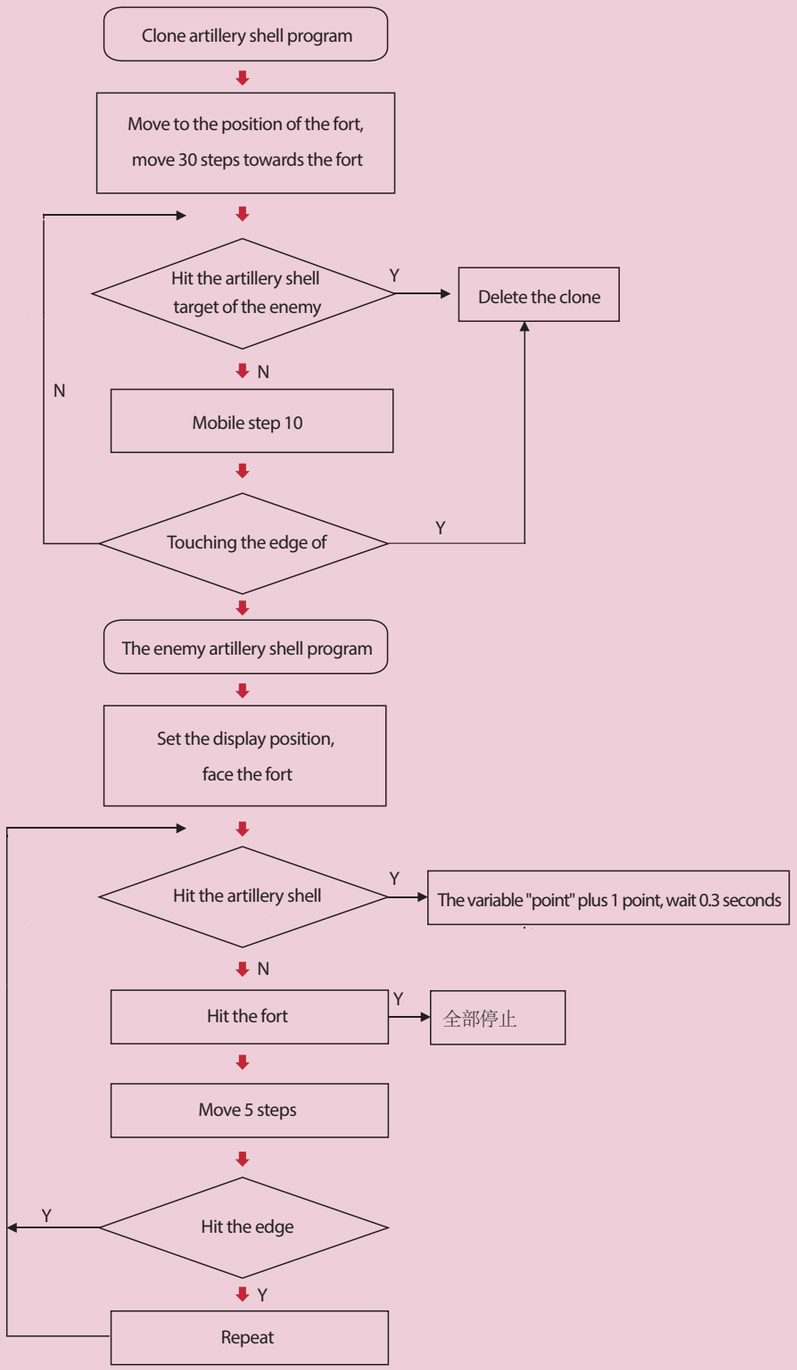
Cloning Function

Clone can reproduce the sprite with the same script and attribute while the program is running. It can solve the problem that the number of sprites is undetermined before the program runs.

Thinking Process

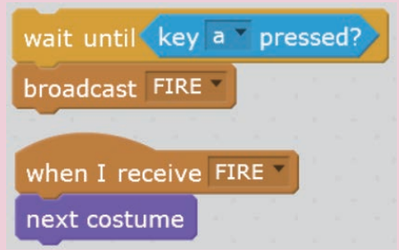


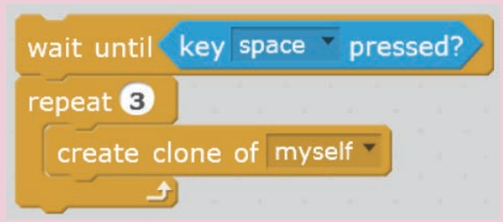


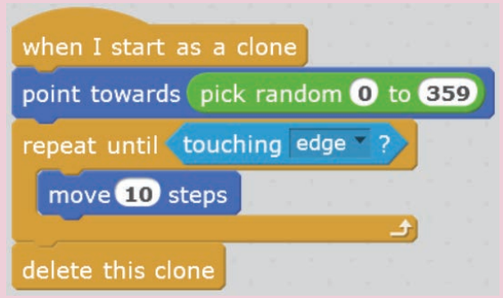
The Idea Description	Thinking Process
Same as above	

Try it

Scripting







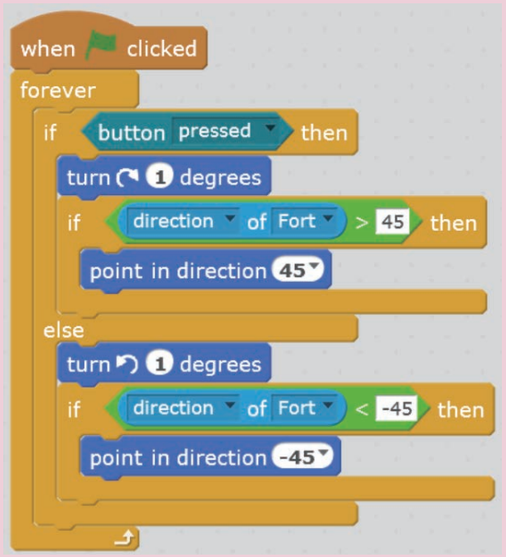
Flow Chart

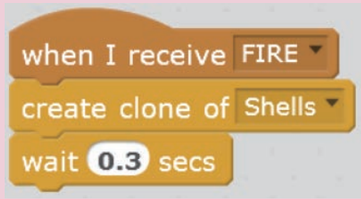
Challenge yourself with the learning flowchart. Please write the flow sequence stated on the left figure.

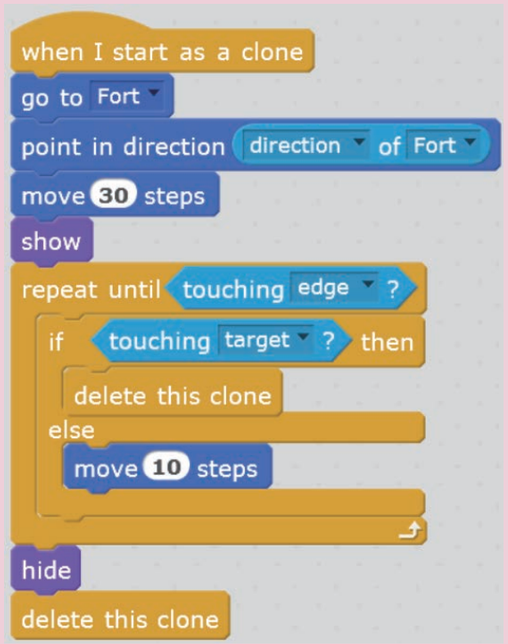
Try to run the script on the left side.
Record change of mBlock.

Example Expansion

Defend the Island		**Operating Instructions** Use the buttons on the mCore to control rotation angle of the fort. Use the optical sensor on the mCore to control emission of the artillery shell. If the fort is hit, the game is over.

Fort Sprite		Use the buttons on the mCore to adjust the angle of the fort. Branch structure is used to control the rotation magnitude that limits the angle of the fort. This makes the rotation of the fort more reasonable.
Stage Background		Control emission of the artillery shells in the background program. Broadcast FIRE message when light gets dimmed.

The Sprite of Enemy Targets		As the sprite continues to appear, the program needs to repeat running. The program sets that enemy target randomly appears in the sky and flies facing the fort. There are two situations before enemy target hits the edge, that is, hitting the fort or being hit by the artillery shell. We can use branch structures to determine: if enemy target hit the artillery shells, it will hide itself and wait 0.03 seconds after adding one point; otherwise, if it is hit by the artillery shells, the game is over.
Shells Sprite		Upon receiving the FIRE message, one clone is made.

Shells Sprite		**Produce clones** Each clone will execute this script. First, the artillery shell is moved near the muzzle and faces the muzzle in the direction. Before it hits the edge, the clones will be deleted if it hits the target, or moves 10 steps forward. It hides itself and deletes the clone after it hits the edge.

Tips:

1. The resulting clones will inherit the Hide/Show attributes of the original sprite: if the original sprite hides, the clones will hide, and vice versa.
2. Broadcast is also an event. Its "receiving the message" blocks start upon receiving a matching message.

Exercise

1. Try to use cloning feature to make a program with raining effect.
2. Use the optical sensor on the mCore to change the stage background.
3. Modify the game and enable 3 to 5 enemy artillery shells to appear simultaneously.

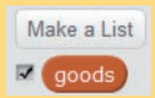
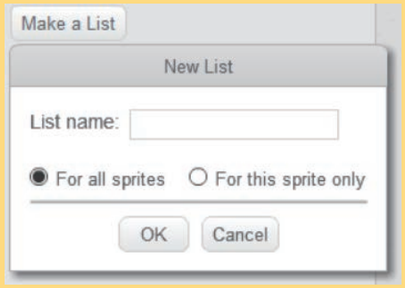
Chapter 6 The Card Reader

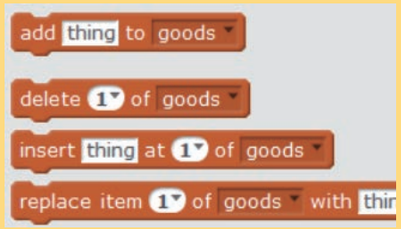
Every commodity in the supermarkets is with a barcode. Every barcode corresponds to a specific commodity. Can you design a machine to read the identity? Come on and realize it!

Learning Objectives

1. Understand and learn features and usage of the linked list

Knowledge Points

Building Blocks	Instructions	Example
	New linked lists can generate a set of variable form. It can store several variables and also take out a variable within the table.	



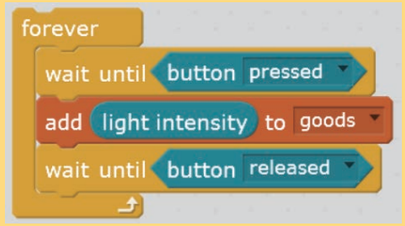
You can enter the "thing" in the blocks in the left figure manually. You can also fill in the variables. The filled data is automatically generated into the sequence number. It is the basis to obtain data in the linked list.

1.The data are stored at the end of the linked list

2.Delete data at the specified position from the linked list

3.Insert data into the specified location

Replace data at the specified position in the linked list

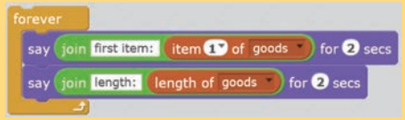


The functions of the blocks in the left figure are as follows:

1.The building block takes out of data in the linked list by position number

2.Get the number of data stored in a linked list

Find whether the linked list contains the data



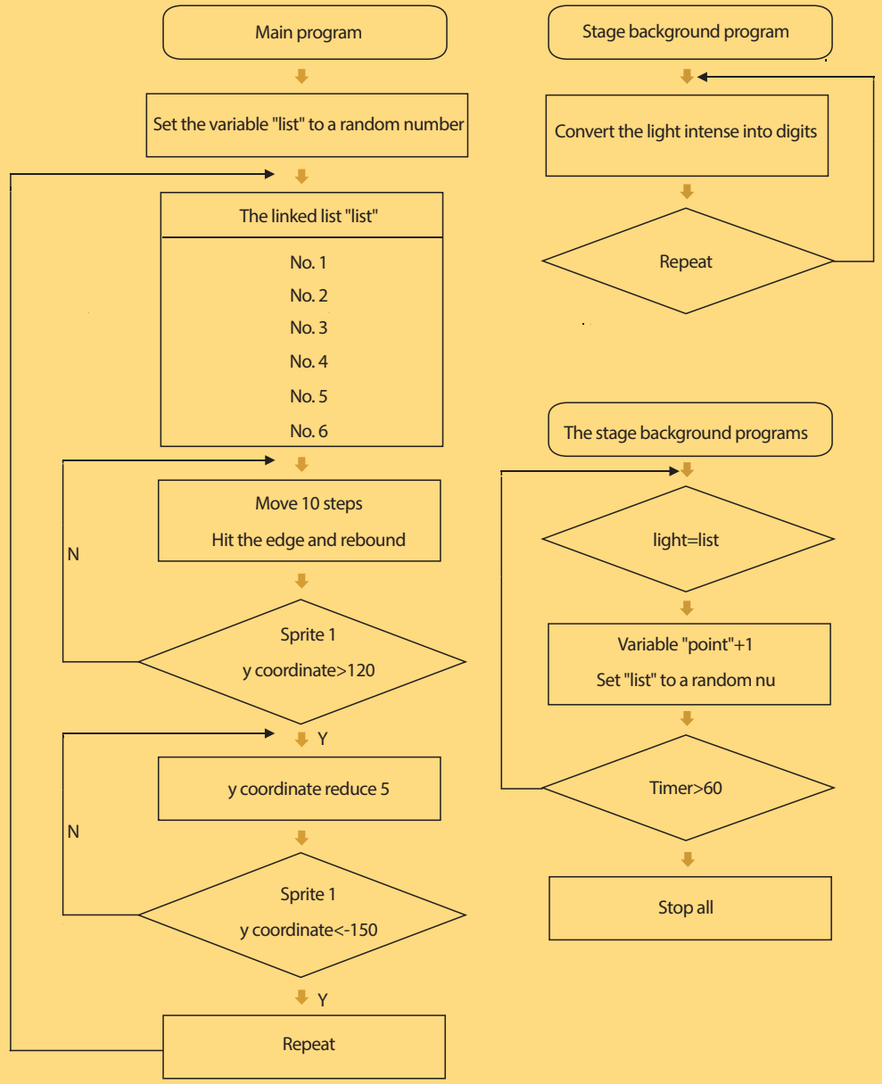
Programming Ideas

The Idea Description

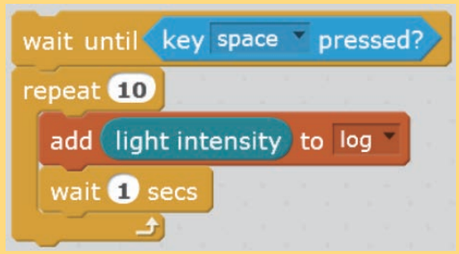
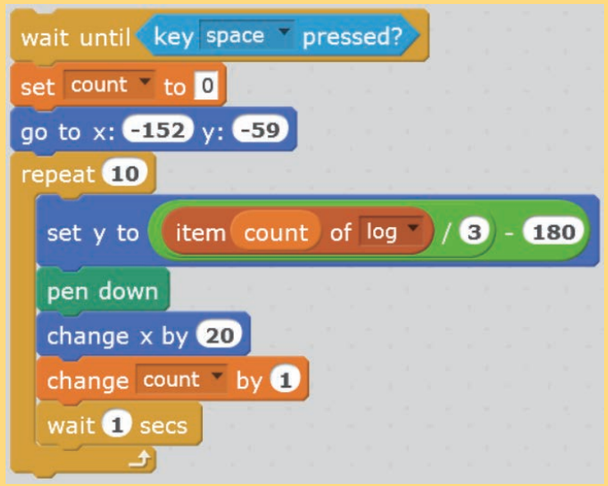
Thinking Process

Linked list Block Description

When the amount of data needed to be recorded is more, we usually use linked list. The linked list can store numeric, string or Chinese text. The data stored in the linked list can be checked and modified at any time.

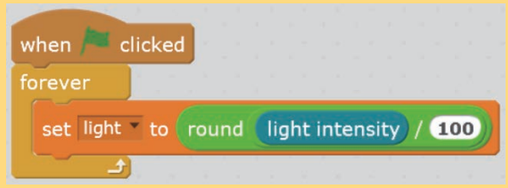
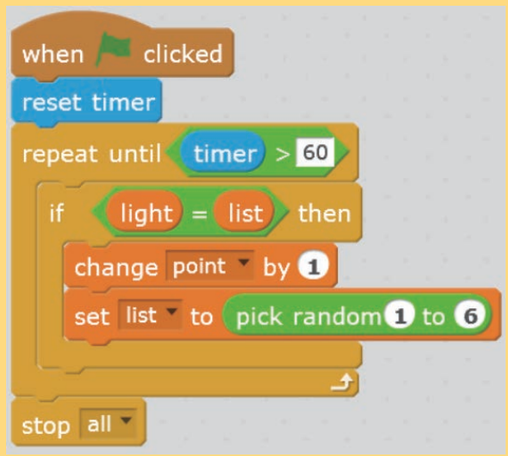


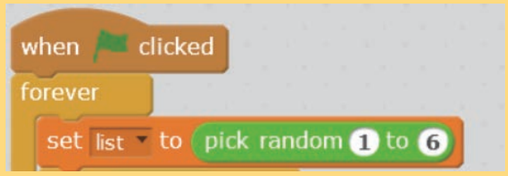
Try it

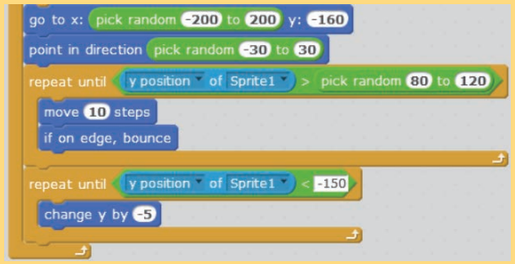
Scripting	Flow Chart
	Challenge yourself with learning flowchart. Please write the flow sequence stated on the left figure.
	Try to run the script on the left side. Record change of mBlock.

Example Expansion

The linked list is a good helper for recording massive amount of data. With it, you can save data (either digital or text) and find or modify data if necessary. It is a commonly used tool for high-level programming!

Card Reader		**Operating Instructions** Use the optical sensor to scan color card Find the corresponding commodity through the linked list
Background Program		The value of the light sensor is processed through arithmetic operation. The value representing the color characteristics is left.
		**Countdown timer of 60 seconds** Within a given period of time, the values in the light sensor and the linked list are compared to validate whether they are same. Wherein the variable "light" is the value of the light sensor and the variable "list" is the value of the linked list.

Item Sprite		Start the program and set the variable "list" to a random number between 1 and 6
		According to the value of the variable "list", change modeling and text of the sprite

Item Sprite		After the modeling is changed, the sprite moves to a region. Keep moving upwards before reaching the 120 y-axis position. Keep moving downward before reaching -120 y-axis position. Repeat at last and randomly set the value of "list" again.

Tips

Right-click the linked list, you can import or export the linked list data. So you can save time for inputting or outputting data.

Exercise

1. Try to use the linked list to record the value of the light sensor (recording data 20 times in 20 seconds) and export it to a text file.
2. Try to use two linked lists to record time and the sensor data.
3. Combining with brush function, try to plot data in the previous topic into a line chart or bar graph.

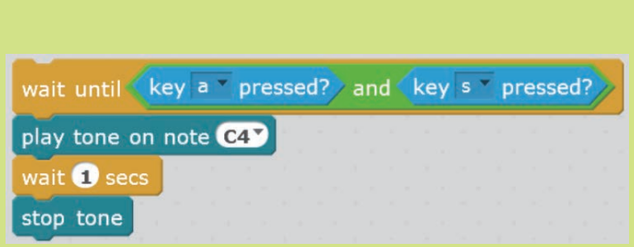
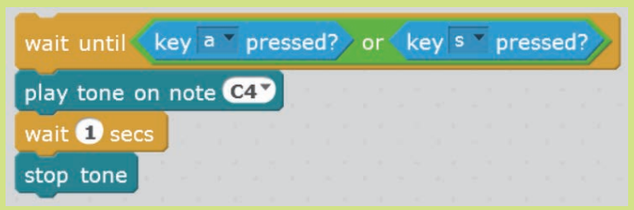
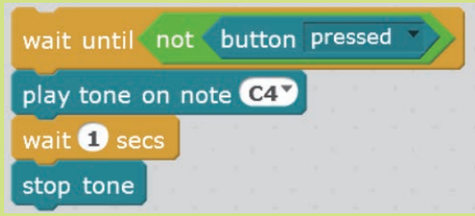
Chapter 7 Rhythm Game

Party time! Happy hour demands music. Let's play a rhythm game and see who has better sense of rhythm. This game is focused on Hand-eye coordination!

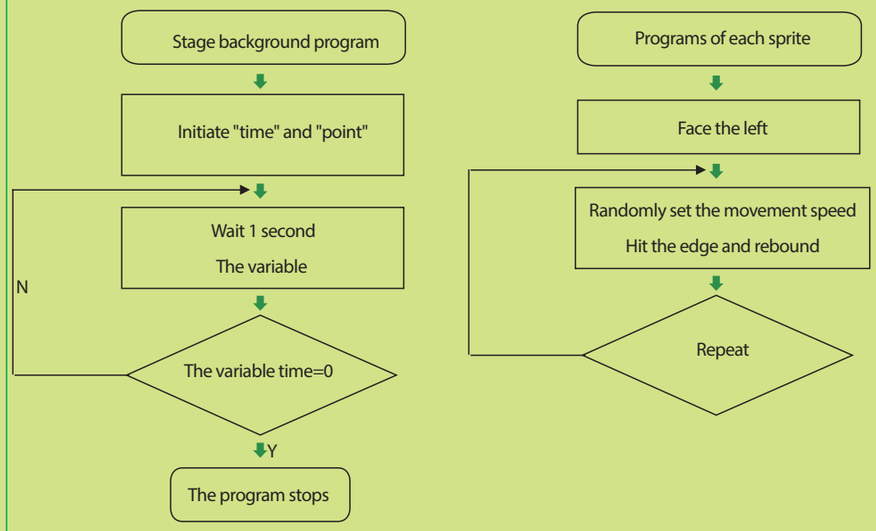
Learning Objectives

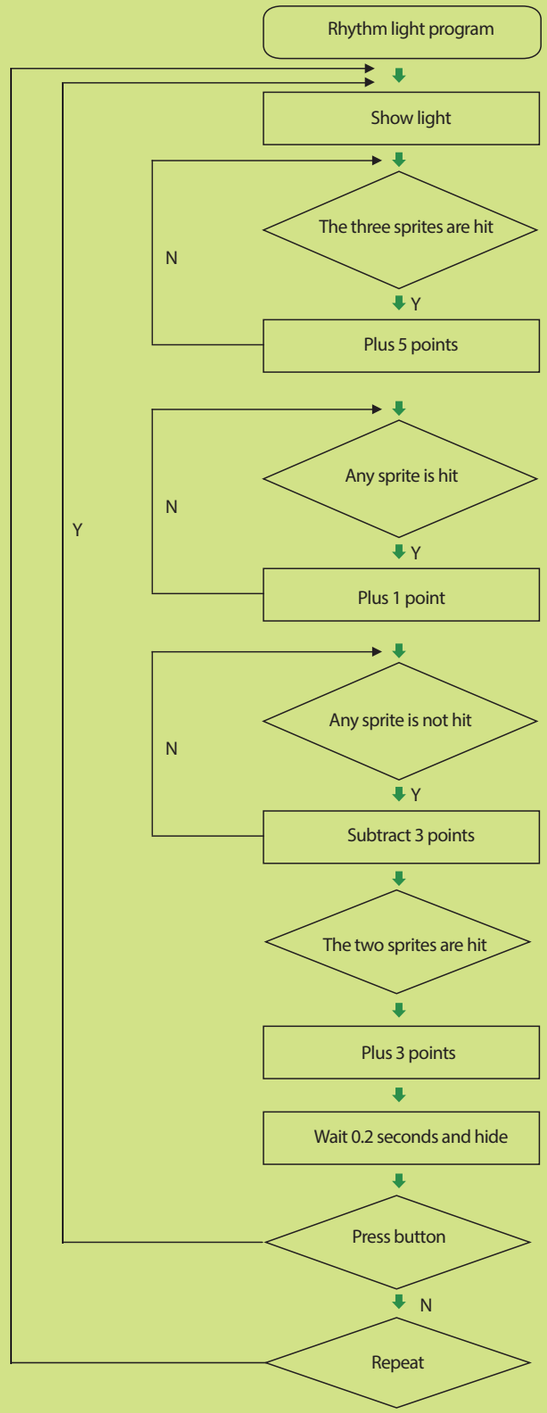
1. Learn logical operators

Knowledge Points

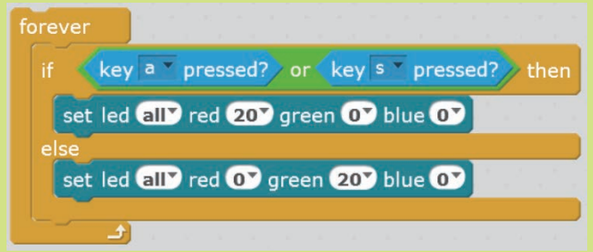
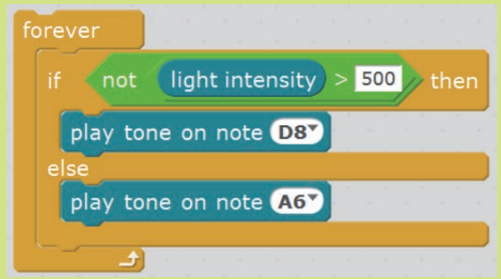
Building Blocks	Instructions	ExampleExample
and	When both internal conditions are true, "and" building block will be true. Otherwise it is false.	
or	When either one of the two internal condition is true, "or" building block is true. Otherwise it is false.	
not	When the condition inside the Block is true, "not" building block is false. When the internal condition is false, the building block is true.	

Programming Ideas

The Idea Description	Thinking Process
Logical Operators When there are two or more conditions needed to be judged, we usually use logical operators to combine conditions. Operation varies in conditions.	

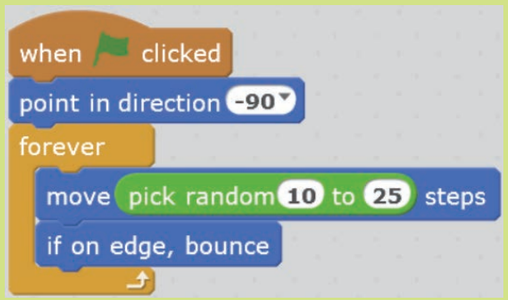
The Idea Description	Thinking Process
Same as above	

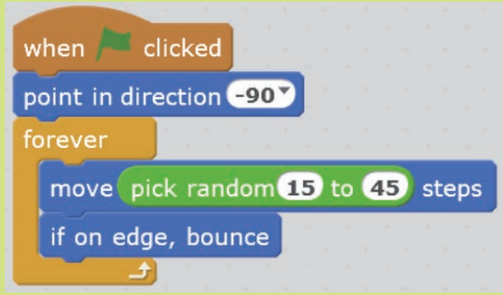
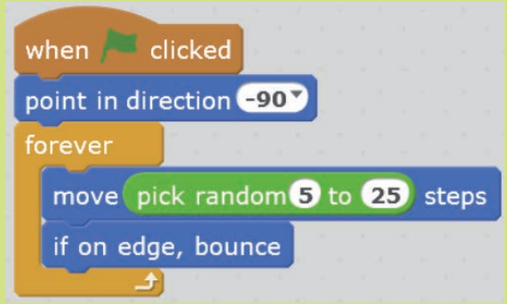
Try it

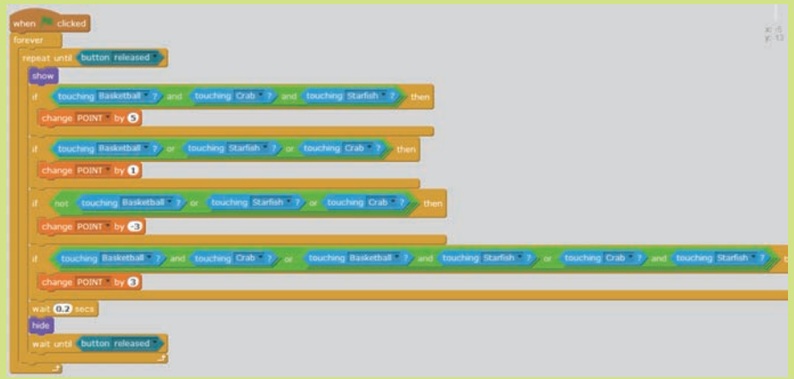
Scripting	Flow Chart
	Challenge yourself with learning flowchart. Please write the flow sequence stated on the left figure.
	Try to run the script on the left side. Record change of mCore. Which tone will mCore play if the light intensity is exactly equal to 500?

Example Expansion

Rhythm game is very simple: When one or more sprites are caught by the middle light beam (light beam is controlled by the button), points are obtained; If any sprite is not caught, points are reduced. Use "and", "or" and "not" to judge points obtain or deduction. Keep your eyes open and concentrated. Let's play this small but tight game!

Rhythm Game		**Operating Instructions** The game is only 60 seconds long. Crabs, starfish and basketball on the stage move around at different speeds. The program uses buttons to control the beam showing or hiding. Points are obtained according to the sprite captured by the beam. Come and try! Let's see how many points you could get in one minute!
Stage Background		**Calculate the game time** The variable "time" represents total time of the game. The variable "point" represents the obtained points. After the timer returns to zero, the game starts. While the program is in progress, the value of the variable "time" continuously diminishes. When it is equal to 0, it indicates that the game is over and the program stops.
Crab Sprite		The sprite moves back and forth on the stage at random speed

Starfish Sprite		The sprite moves back and forth on the stage at random speed
Basketball Sprite		The sprite moves back and forth on the stage at random speed

Beam Sprite		Logic judgment of beam The script constantly checks the contact between the light beam and the other three sprites. Points obtained vary in situation.

Exercise

1. Use "and" building block to design an effect: enlarge the sprite if the condition is met. Otherwise, narrow the sprite.

2. Use "not" building block to design a program that makes LED light blinking .

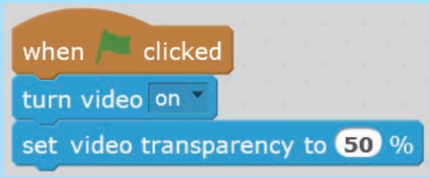
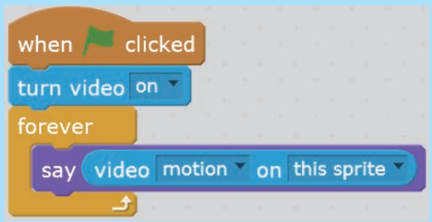
Chapter 8 The Video Ball

In this chapter, we will learn how to make an interactive video game. The game tests your hand-eye coordination. Are you ready?

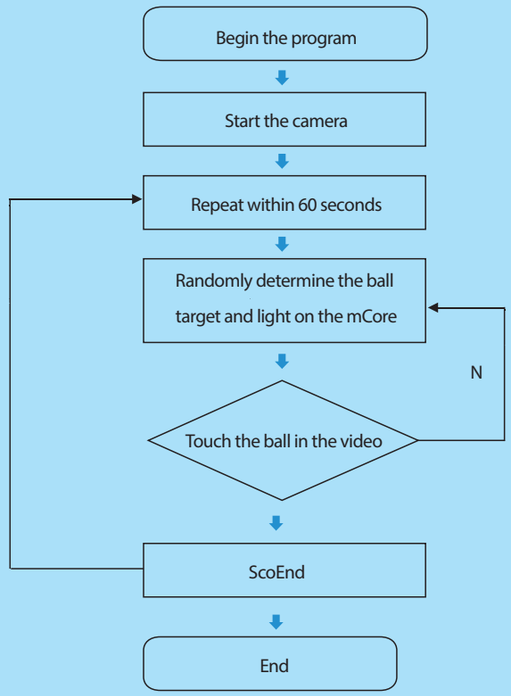
Learning Objectives

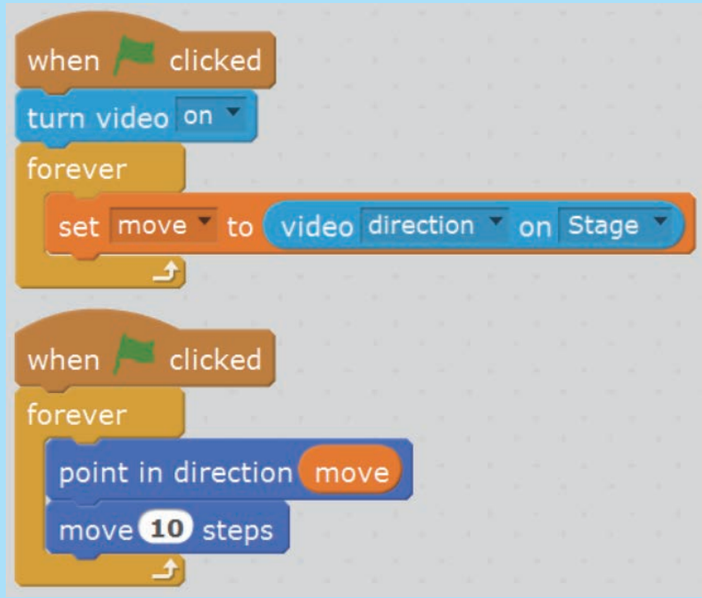
1. Learn to use the video-related building blocks
2. Learn to use variables

Knowledge Points

Building Blocks	Instructions	Example
turn video on	Start the camera connected to the computer	
set video transparency to 50 %	Set the transparency of the video. 0 is opaque and 100 is completely transparent.	
video motion on this sprite	Detect sprite's movement in the video.	

Programming Ideas

The Idea Description	Thinking Process
Use Video: Use the camera to make an interactive game. The flow is shown in the right figure. The target balls for touching are randomly generated by the computer. Then the mCore turn on LED lights with the corresponding color. So players know what color ball they need to touch. The program uses the branch structure for scoring. When 60 seconds end, the score is calculated.	



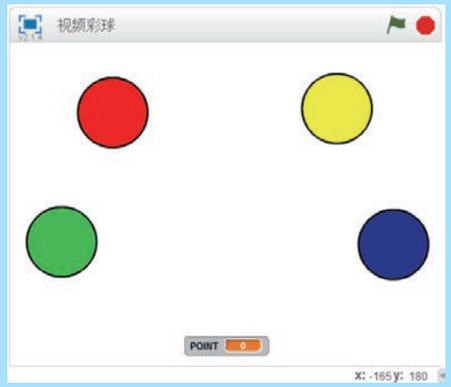
Try to run the script on the left side. Record change of mCore.

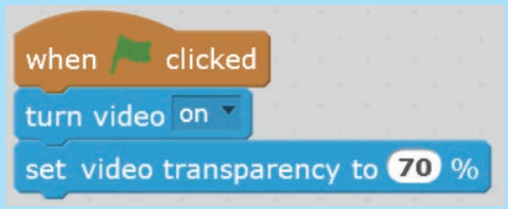
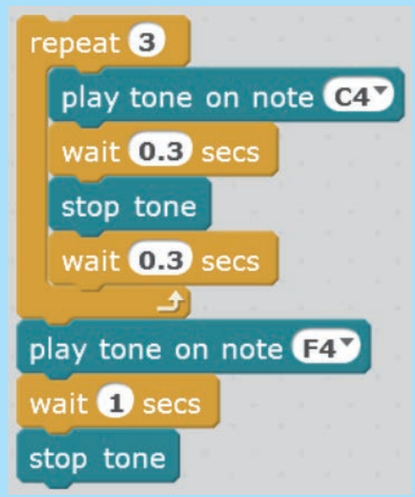
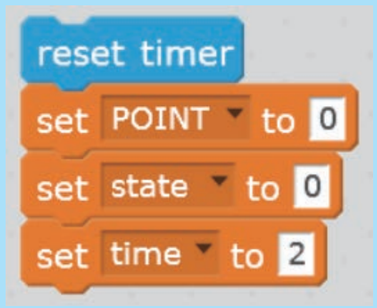
Try it

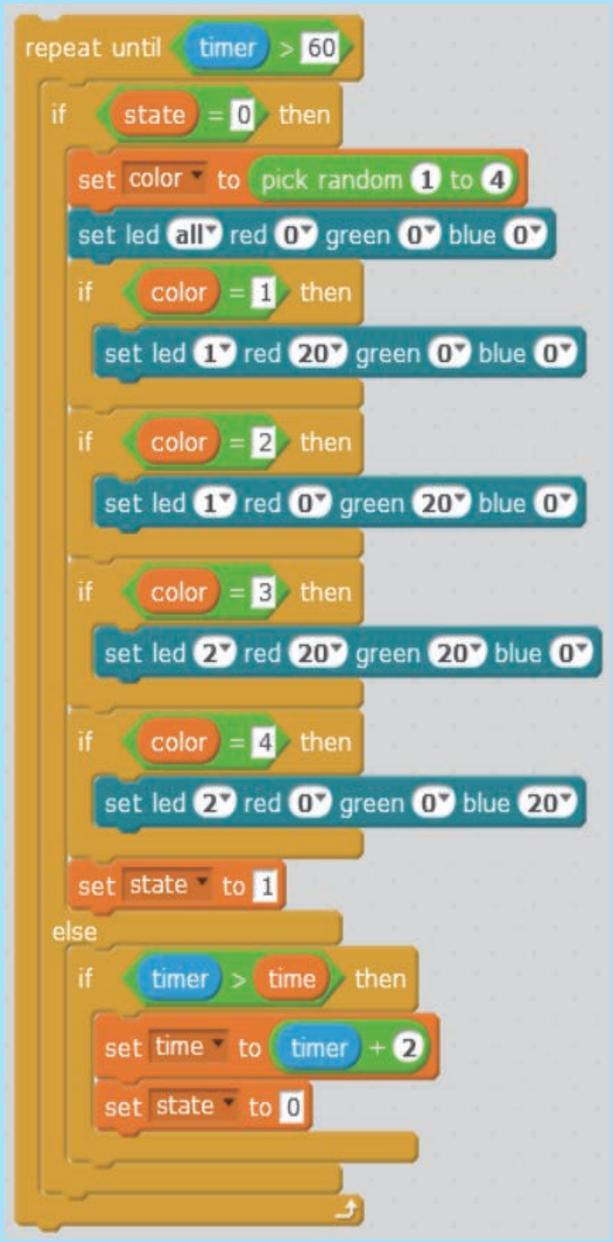
Scripting	Flow Chart
	Challenge yourself with learning flowchart. Please write the flow sequence stated in the left

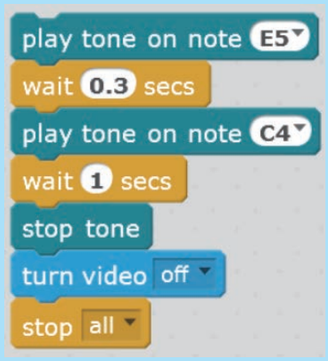
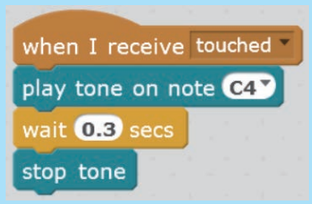
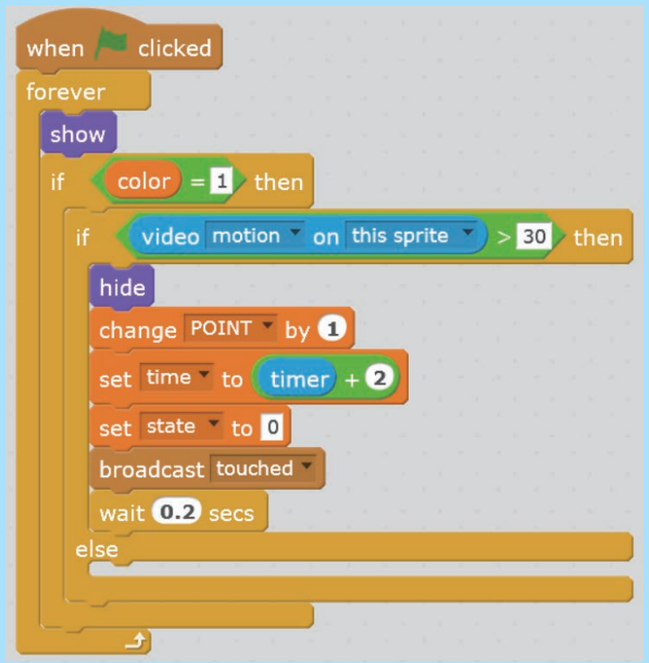
Example Expansion

It is a new feature of Scratch 2.0 that uses the camera as a sensor. The mBlock also inherits this feature. When the video features are combined with mCore, it can create lively, interesting, funny and interactive games. Let's experience it now!

Video Ball		**Operating Instructions** When the game starts, LED on the mCore will constantly change color. Players must touch the ball of the same color on the stage within certain time. You will obtain points if you touch the balls within the specified time. The game time is totally 60 seconds. You need to connect the camera to your computer.

Stage Background		You should firstly start the camera and set the transparency when the program starts.
		The beeper sounds before the game starts.
		**Initialize variables and timers** The timer display returns to zero. The variable POINT represents player's points. The variable "state" represents whether players answer. 0 indicates that they have answered. 1 indicates that they have not answered. The variable "time" represents the next timeout time point. have answered. 1 indicates that they have not answered. The variable "time" represents the next timeout time point.

Same as above		The game is 60 seconds long. Program constantly switches the color of LED (1 is red, 2 green, 3 yellow and 4 blue). Then the status of "state" is set to 1. It indicates that the player did not answer. If the players answer it within two seconds, "state" will be set to 0 and the program re-picks the color. And then the game enters into the next round; otherwise, if the answer is not made within two seconds, the variable "time" is set to the next time-out point. At same time, the "state" is set to 0 so that the next round of color conversion is done.

Stage Background		The game is over. Play a beep tone and turn off the camera.
		The buzzer sounds when point is obtained.
Ball Sprite		Four ball sprites constantly judge whether they are the same color as the one displayed by the mCore and be hit by the video. When the ball is hit correctly, it will disappear for 0.2 seconds. The variable "color" is set in the stage background, which represents the color selected by the game.

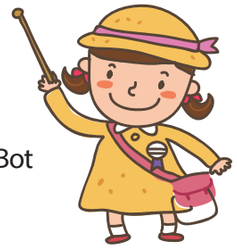
Tips

Why will the ball disappear for few seconds when it is hit? This is because the continuous display can cause incorrect scoring for continuous score. So it must disappear for few seconds.

Exercise

1. Test relationship between the camera and its transparency.
   Record the values obtained through  .
2. Compare the difference between  and  .

Chapter 9 Run! Robot

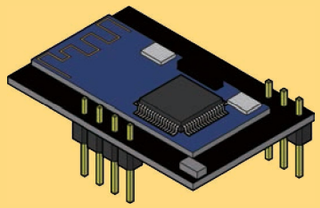


We can apply the mBlock programming knowledge we learned to control mBot robots. If you play a chasing game with other children, it will be very interesting.

Learning Objectives

1.Learn how to use mBlock to control traveling of robot

Electronic Modules

Category Name	Legend	Instructions
Bluetooth Module		Bluetooth can help us be free from the data line and enjoy wireless programming

Knowledge Points

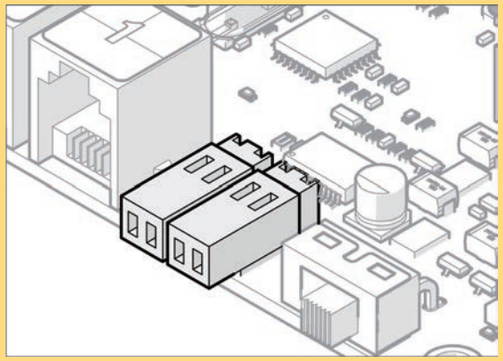
Module	Building Blocks	Instructions
Robots	set motor M1 speed 0	Set the motor interface and implement rotation of the motor. The value range from 255 to -255 .255 represents the maximum forward rotation value, while -255 indicates the maximum reverse rotation value

Programming Ideas

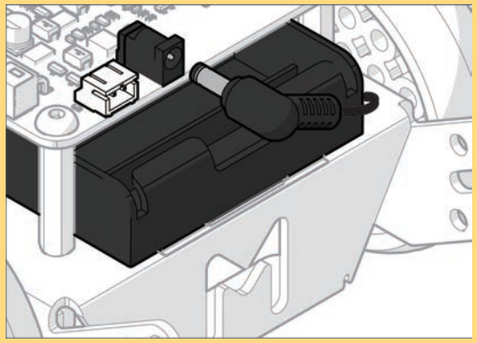
The Idea Description	Thinking Process
Use mBot Bluetooth to receive mBlock keyboard commands and control the movement direction	Event of pressing the key → The motor runs Event of releasing the key → The motor stops

Try it

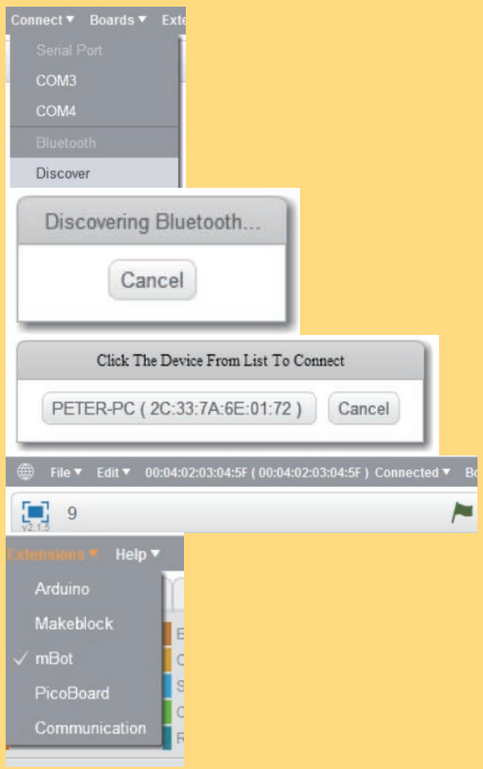
1.Connect the Motor

Legend	Instructions
	At the side of the main board, there are two orange ports for connecting the motor.

2.Power by Battery

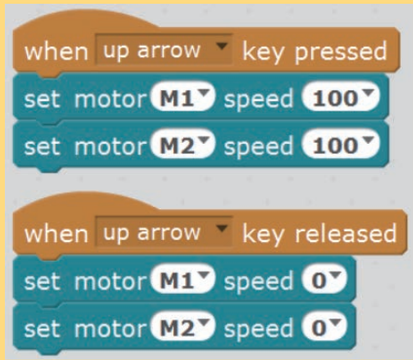
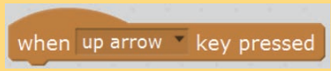
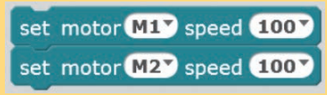
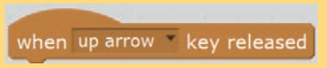
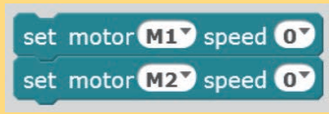
Legend	Instructions
	Before you use Bluetooth, please make sure to use the battery to power and switch on.

3.3.Create Bluetooth Connections

Legend	Instructions
	Click Discover and wait for the search results of Bluetooth devices. Find the Bluetooth connection named "Makeblock" When it is successfully connected, the toolbar will change, as shown on the left figure. Finally, make sure to select MBot in the connection.

4.Write a Script

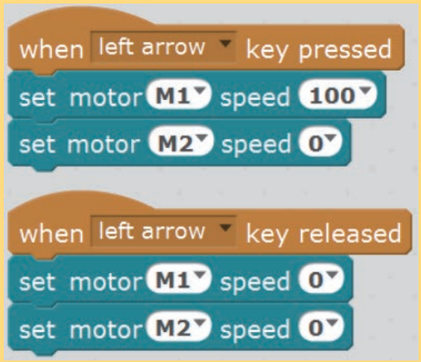
Achieve the mobile mBot

Idea	Script	Script Description
Press the move-up key on the keyboard to move mBot forward. Release the move-up key on the keyboard to move mBot backward		 Response only by pressing the move-up key.  Set speed of two motor to 100.  Response by releasing the move-up key.  Set speed of two motors to 0 and the motors stop.

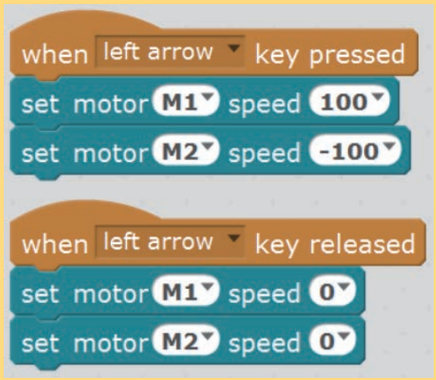
Tips

Motor speed of 100 is not a real speed value but a "range value" for easy operation. Within the range, speed varies in voltage and motor. This "range value" is in the range of 0-255. To be specific, 0 indicates the motor stops while 255 indicates the motor reaches the maximum speed at the current voltage.

How will you operate it if you want to steer the mBot ?

Idea	Script	Script Description
If a wheel rotates forward and a wheel does not turn, what happens?		When you press the move-left button, set the left motor to 0 and the right motor to rotation speed. Then the car will turn left. (If your test is opposite to the above effect, it is still correct. You just need to remember the motor interface number and corresponding motor.

If you think mBot turns too slowly, you can also set it as follows

Idea	Script	Script Description
If a wheel turns forward and a wheel turns back, what happens?		If the left wheel turns back and the right wheel turns forward, the car will quickly turn left.

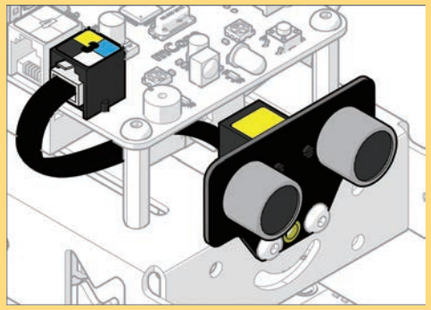
Please program by yourself with the knowledge learned above. Then you can control the car with the up , down, left and right keys on the keyboard.

	Control the mBot program with the arrow keys on the keyboard.

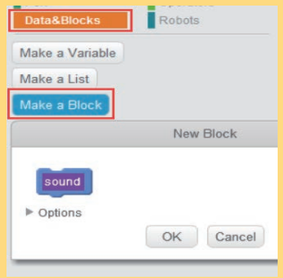
Example Expansion

Judge if there is an obstacle ahead in the way.

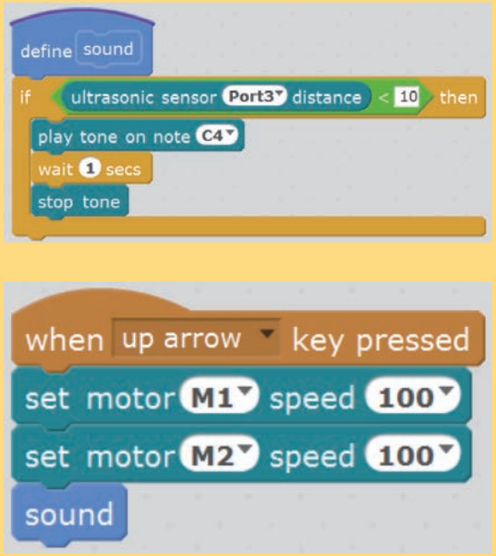
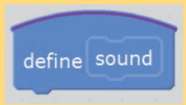
1.Ultrasonic Sensors

	Install the ultrasonic sensor on the front of the car. Connect the data cable to the 3rd port

1.Define Module Commands

Idea	Module	Module Description
Create a module command named "sound"		Click the data and commands. Create a new module command named "sound"

2.Write a Script

Idea	Script	Script Description
When mBot moves, the "sound" program will run. If there is an obstacle ahead detected by the ultrasonic sensors, the buzzer will alarm.		 is the beginning of the program definition. If the ultrasonic sensor(connected to the interface 3) senses that the obstacle ahead is less than 10 cm from it, the buzzer will sound for one second. When the mBot moves (such as pressing the up/down key), the "sound" program runs.

Exercise

1.Let us think, is there any problem for the above program? Can we use the spacebar to control the car's alarm?

More interesting games are waiting for your development!

Chapter 10 Dodging Master

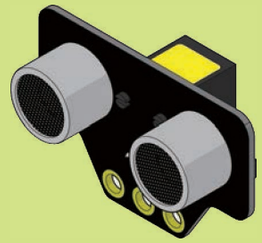
The robot mBot wants to learn walking. But it always hits the obstacle ahead. Look! mBot hits the children in front of it. It is very frustrating. Let's help it now. In this chapter, we will learn how to use mBlock programming to enable mBot to avoid obstacles ahead.

Learning Objectives

1.Understand the usage of ultrasonic sensors

2.Learn how to use mBlock programming to enable mBot to avoid obstacles

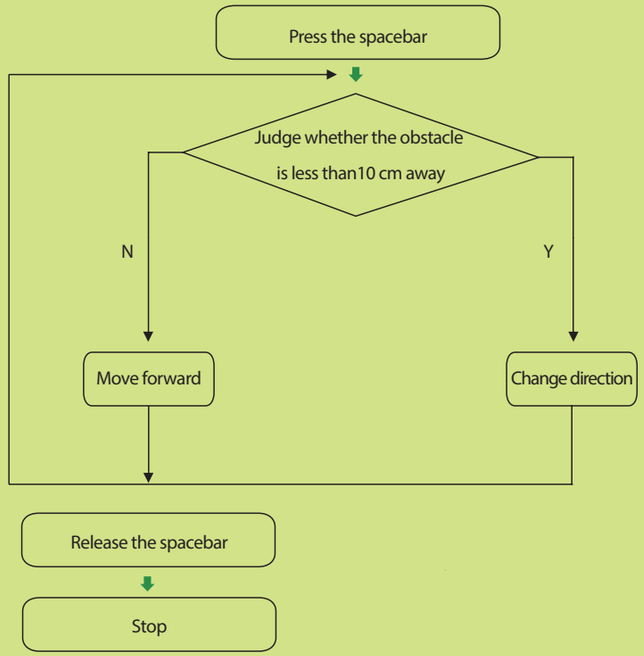
Electronic Modules

Category Name	Legend	Instructions
Ultrasonic Sensor Module		Ultrasonic sensors can detect the distance from the object ahead

Knowledge Points

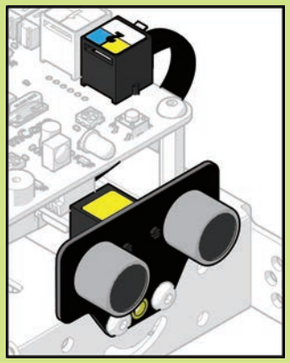
Module	Building Blocks	Instructions
Robots		The distance from the object ahead detected by the ultrasonic sensor will be measured in centimeter.

Programming Ideas

The Idea Description	Thinking Process
Hold down the spacebar and mBot move forward. If the ultrasonic sensor detects that it is closed to the object ahead (such as less than 10 cm), the mBot will turn to avoid obstacles and move on. If you release the spacebar, mBot stops.	

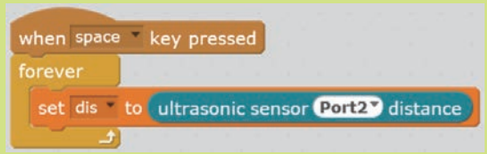
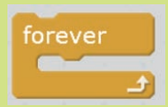
Try it

1.1.Install Ultrasonic Sensors

Legend	Instructions
	Install the ultrasonic sensor on the front of mBot. Connect the data line to the No. 2 port on the main board.

2.Write a Script

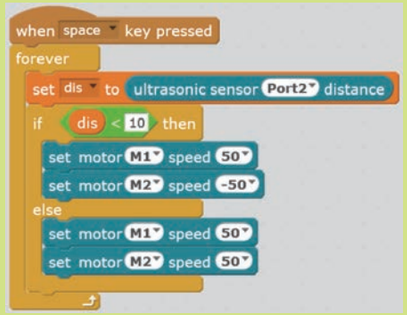
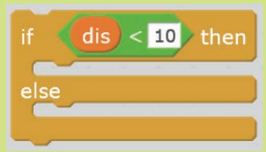
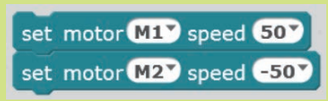
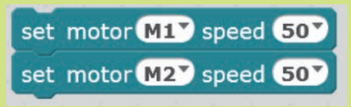
Acquire the sensor data

Idea	Script	Script Description
Create a variable (e.g. dis) to hold the distance data detected by the ultrasonic sensor.		 Press the spacebar (do not release) to trigger the program  Because the ultrasonic data is constantly acquired, it should use the loop block.  It assigns the variable "dis" to the distance from the object ahead detected by the ultrasonic sensors . In this way, when you click the little green flag, the current distance detected by the ultrasonic sensor will be displayed in the upper left corner. 

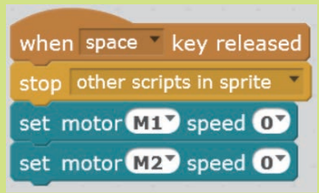
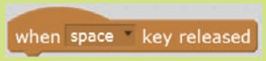
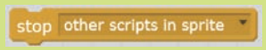
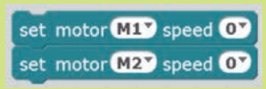
Tips

Normally the human ear cannot hear sound of 20 kilohertz (kHz) or higher. Such sound wave is called ultrasonic wave. Bats flying at night can make such type of ultrasound. Bat is with amblyopic eyes, so its locating and foraging rely on ultrasound. Bat emits sound waves ranging from 20 and 120 kHz. The sound wave come from bat's mouth or nose and is accepted by its ears. Makeblock ultrasonic sensor is also divided into the transmitting end and the receiving end. The end marked with T is transmitter and the end with R is the receiver.

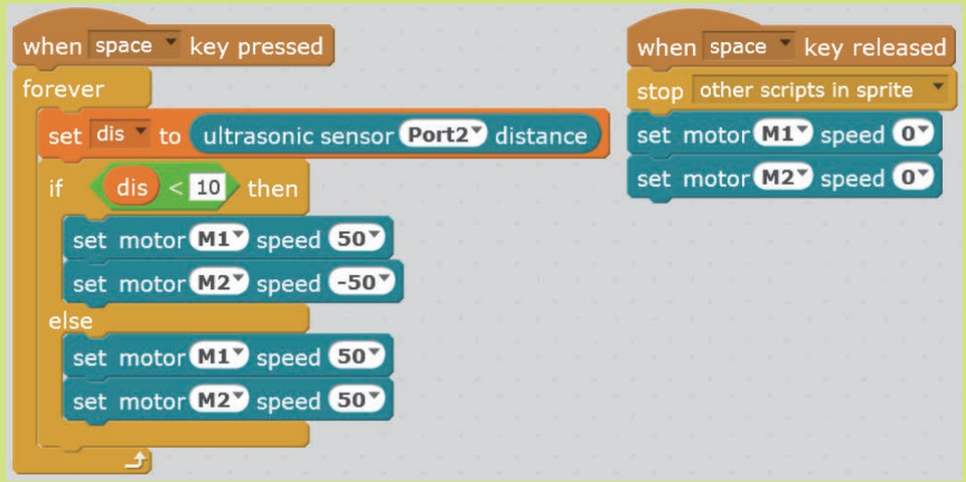
Add a judgment and enable mBot's function of avoidance

Idea	Script	Script Description
If the "distance value" is" less than 10, the two motors rotate in the opposite direction to make mBot turn. Otherwise, the two motors rotate forward and mBot moves forward.		 Judge whether the distance value is less than 10.  Two motors rotate in reverse direction and mBot turns.  Two motors rotate in the same direction and mBot moves forward.

Set mBot stopping event

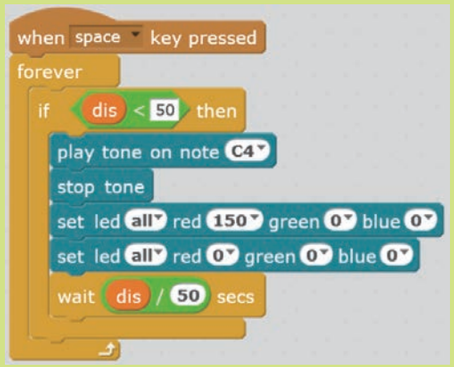
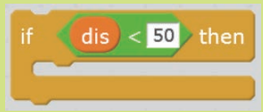
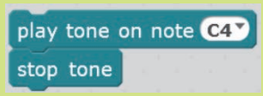
Idea	Script	Script Description
Set the motor speed to 0 and stop other scripts.		 It will trigger event by releasing the spacebar.  Firstly, stop other script programs (because the other program keeps judging whether there are obstacles ahead).  Set the motor speed to 0 and stop turning.

All programs

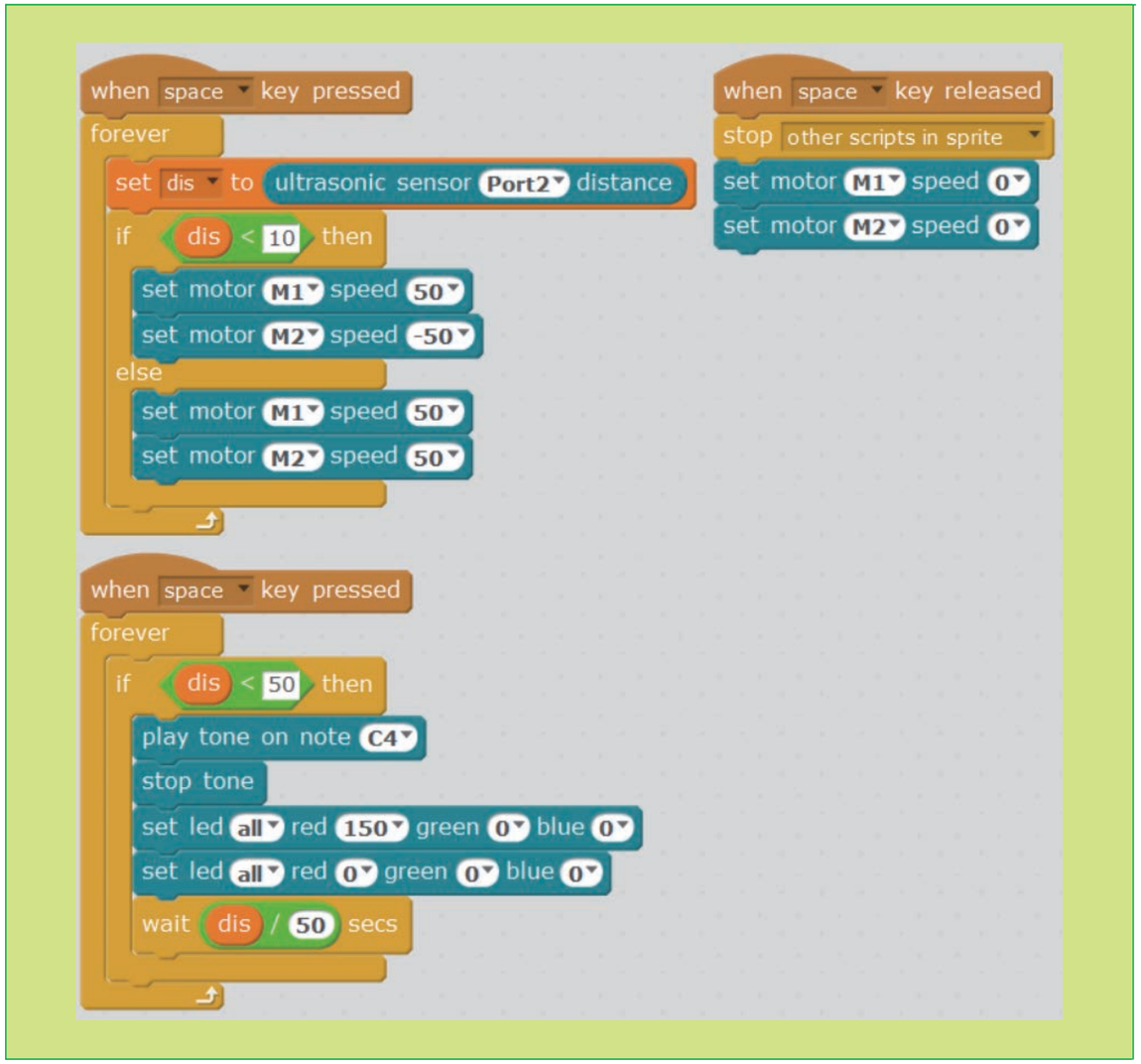


Example Expansion

Could you achieve following effect? When mBot moves forward, if there is an obstacle ahead (eg 50 cm away), mBot will be alerted and turn on alarm light. As obstacle is getting closer, alarm and light frequency will gradually accelerate until mBot turn.

Idea	Script	Script Description
Add the program of "when you press the spacebar". Write effects of tone and small LED light		 Add new events.  Due to the continuous detection, we use the loop block.  The variable "dis" in the previous program is called to check whether dis value is less than 50 (if less than 50, it indicates that there are obstacles ahead within the distance of 50 cm).  Stop immediately after playing a tone.  Turn on the small LED (red) and turn off it immediately.  As it is getting closer, the waiting time is gradually reduced.

All programs



Exercise

1.Can we use the recording function to record a piece of sound and play it when mBot encounters an obstacle?

2.Can we set mBot to turning randomly when it encounters an obstacle?

Chapter 11 The Success Road

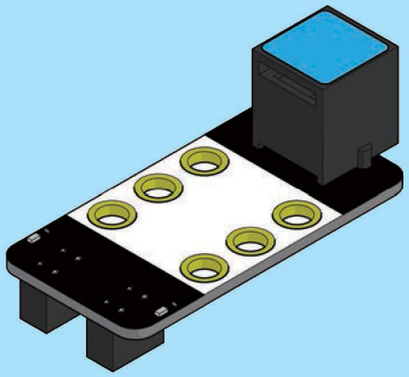
The robot mBot sees the train moves along the track. It hopes it could be with a train-like track. In this chapter, we will learn the line-patrolling feature of mBot!

Learning Objectives

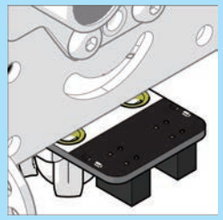
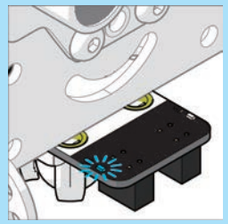
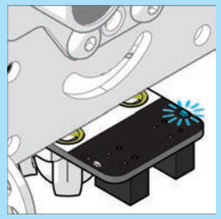
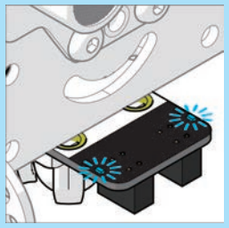
1.Understand the characteristics of the line-patrolling sensor

2.Learn to patrol line with the line-patrolling sensor.

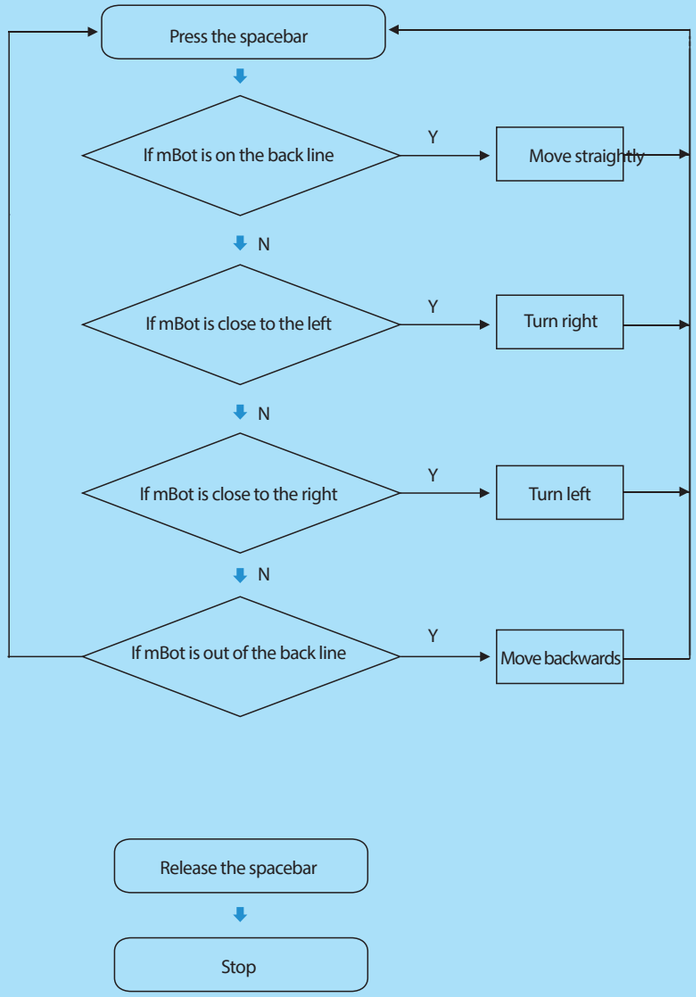
Electronic Modules

Category Name	Legend	Instructions
Line-patrolling Sensor Module		Line-patrolling Sensor module can help the robot move along the black line on the ground. There are two detectors on its front. Through the reflected light projected on the ground by the LED, it detects the offset of the module against the black line.

Knowledge Points

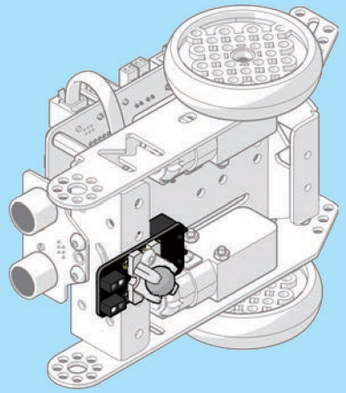
Module	Building Blocks	Instructions
Robots	line follower Port1	Get offset value of the line-patrolling sensor. They are a total of four values, namely 0,1,2,3. Among them, 0 represents that the sensor is completely along the black line and 3 represents that the sensor is completely off-track. 1 and 2 illustrate that one side of the sensor is away from the black line, while the other side is along the black line. As shown in the figure:  line follower Port1 The value is 0  line follower Port1 The value is 1  line follower Port1 The value is 2  line follower Port1 The value is 3

Programming Ideas

The Idea Description	Thinking Process
Hold down the spacebar and then mBot runs the program. Judge whether mBot is completely along the dark line. if it is, it moves straightly. Otherwise, it continues to judge whether mBot is on the left (right). If it is, it turns to the right (left) and make mBot return to the black line. If mBot is away from the black line, it will draw back to the black line. Repeat the cycle as this until you release the spacebar.	

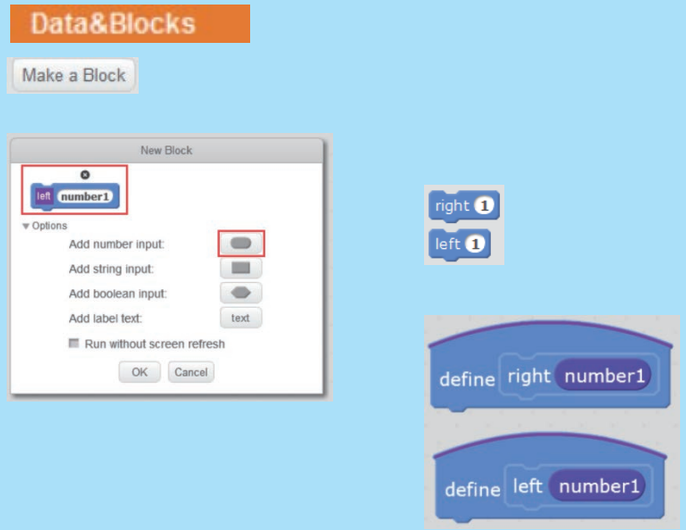
Try it

1.Install Line-patrolling sensor

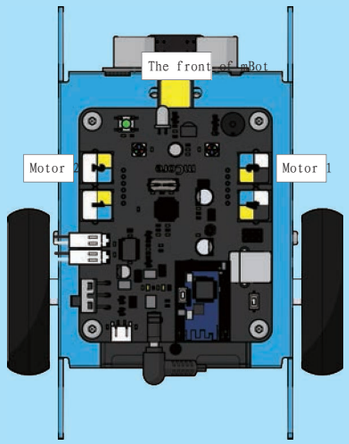
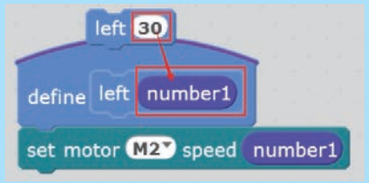
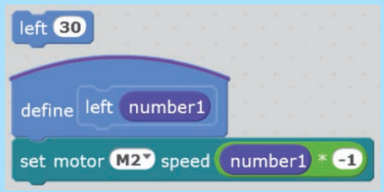
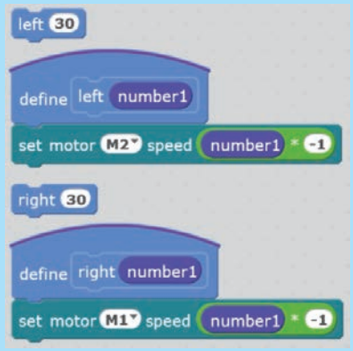
Legend	Explanation
	The line-patrolling sensor is attached to one end of mBot. Please make sure that two detectors are downwards. The data cable is connected to the port 3. Note: For easy learning, we regard the end with the line-patrolling sensor as the front of mBot. So, if you want to drive mBot forward, you should set the motor to turning backward during programming (example: -50).

2.Set Custom Module Motor

In this lesson, we regard the end with the line-patrolling sensor as the front of mBot. For easy learning and operation, we set the motor operation with the custom module, so as to facilitate the follow-up operation.

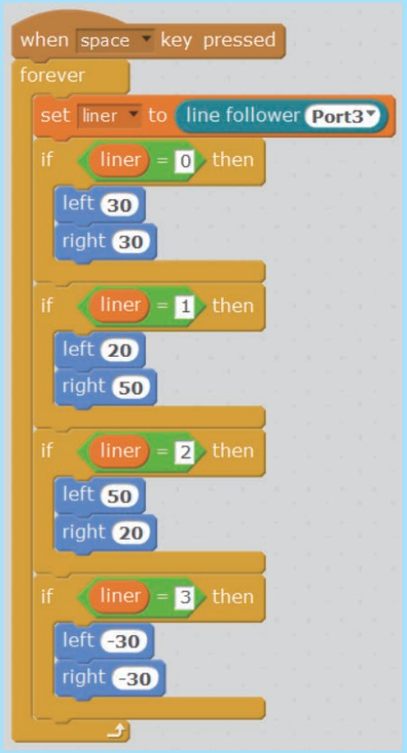
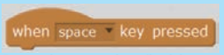
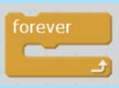
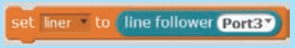
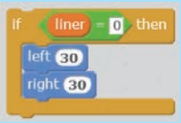
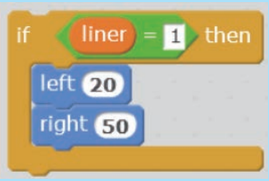
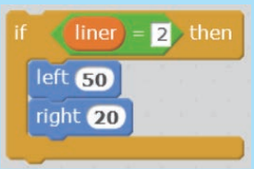
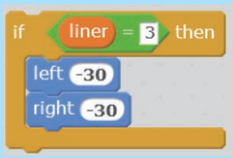
Legend	Explanation
	In "Data and Blocks" module, click "Make a Block" Select "Add number input" in the open form(This parameter can be referred as motor speed) and rename the module as "left" (It means the custom function would operate the motor on the left side of mBot) Create a custom module "right" for operating the motor on the right side of mBot

3.Debug mBot's movement

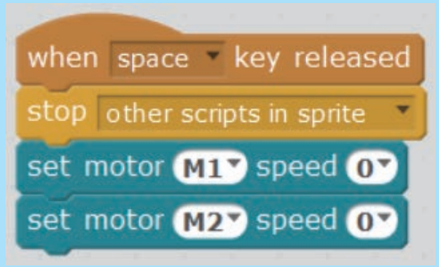
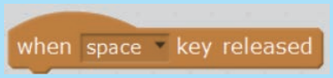
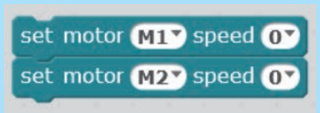
Legend	Explanation
	The left wheel of mBot corresponds motor 2 and the right wheel corresponds motor 1. Proceed as follows to set custom modules:  The left wheel of mBot corresponds motor 2 and the right wheel corresponds motor 1. Proceed as follows to set custom modules: As shown in the figure, when we run command "left", commands in the custom module are called. The value 30 in the "left" command will replace the value number1. In this example, the rotation speed of motor 2 will be 30. Does the motor rotate forward or backward? In this lesson, the front of mBot is the end of the previous lessons. Therefore, you should set the value to a negative value if you want the motor to rotate forward (In the previous lessons, it rotates backward). We can finish this step in the custom module:  The value of number1 will be multiplied by -1. Finally, prepare custom modules "left" and "right": 

4.Write the Script

Set the Line-patrolling function of mBot

Idea	Script	Script Description
When you press the spacebar (hold down), mBot begins to judge the state of the line patrolling sensors. The rotation of the motor depends on the offset, which facilitate the direction adjustment of mBot.		 Press the spacebar (hold pressing) to trigger the program  Because it will constantly acquire the data of line-patrolling sensor, it is necessary to use a loop block.  The value detected by the sensor is assigned to the variable "liner".  If the liner is 0, it indicates that mBot is just along the black line and rotation speed of left and right motors is consistent with other.  If the liner is 1, it indicates that mBot is right to the black line. It should turn left. At that time, the left motor decelerates and the right motor accelerates.  If the liner is 2, it indicates that mBot is left to the black line and should turn right. At that time, the right motor decelerates and the left motor accelerates.  If the liner is 3, it indicates that mBot has been completely off-track the black line .It should immediately be back to the black line.

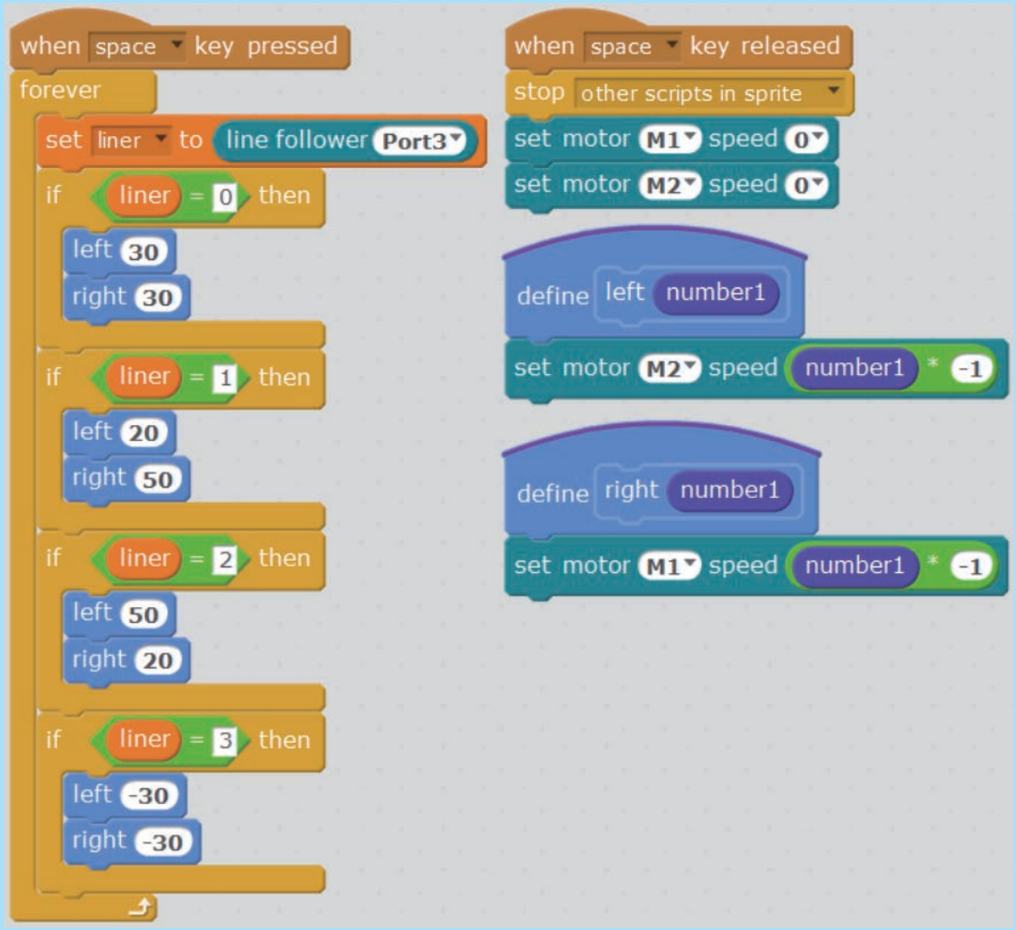
Stop the line-patrolling function of mBot

Idea	Module	Instruction Description
When you release the spacebar, mBot stops.		 Release the spacebar to trigger event.  Stop other script programs (because the other program is continuously judging the value of the line-patrolling sensors).  Set the motor speed to 0 and stop rotation.

Tips:

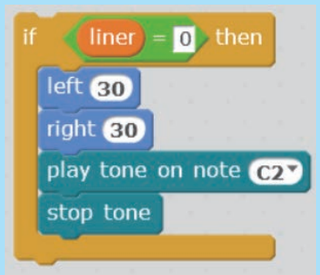
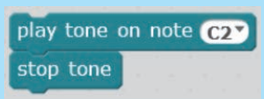
It should make sure that mBot is along the black line when the program starts. If the line –patrolling of mBot doesn't function well, for example, often off-line, you can appropriately increase the width of the black line.

All programs



Example Expansion

The robot mBot learned how to move along the track! It is very happy. It was humming a song and walking on the "success road". Please write programs for the robot and add songs according to different values detected by line –patrolling sensor

Idea	Script	Script Description
The buzzer plays different sound according to various values detected by line –patrolling sensor		 If your play tone and immediately stop playing, it will emit short and rapid voice. That is the effect we expect! As a result, mBot will play melody when it patrol the lines (you could also try to take different tones).

Exercise

1.Write a program to control the LED lighting effect and make mBot movement more cool.

Chapter 12 The Obedient Robot

mBot is a smart obedient robot. There is an infrared receiver on its body. We can remotely c control mBot's movement with this receiver. Is it so cool, ah? In this lesson, we will learn how to control the robot with the infrared remote controller.

Learning Objectives

1.Learn how to control the robot with the infrared remote controller.

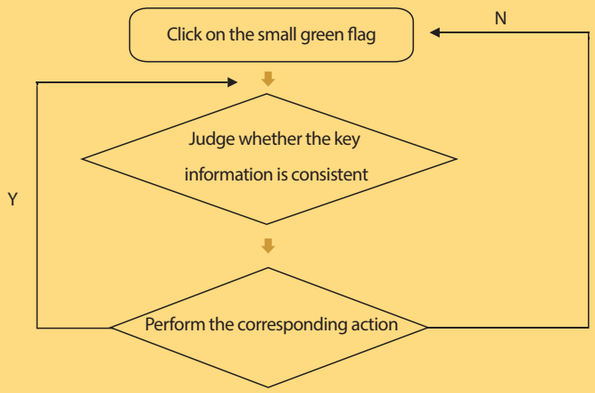
Electronic Modules

Category Name	Legend	Instructions
Infrared Remote Controller		Transmit the coded infrared signal. It is received and processed by the infrared receiver of mBot master controller, which realize the related program action .

Knowledge Points

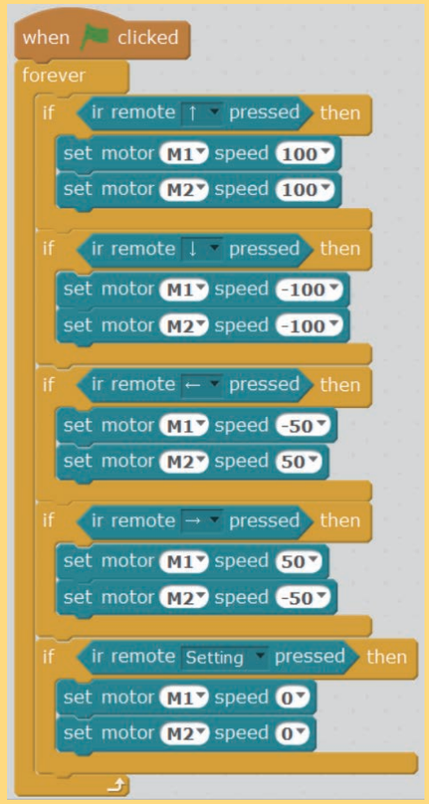
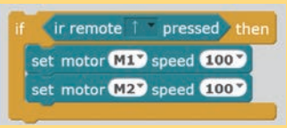
Module	Building Blocks	Instructions
Robots	ir remote A pressed	Get the button Information for the infrared remote controller

Programming Ideas

The Idea Description	Thinking Process
Click the green flag to start the loop program. It will constantly judge whether the received button information of the infrared remote controller is consistent with the preset information (such as the A key). If yes, corresponding action is executed. Otherwise, the loop continues.	

Try it

Use arrow keys to control the traveling mBot

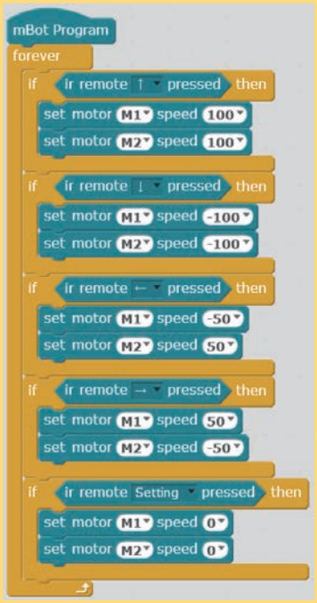
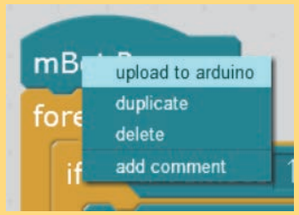
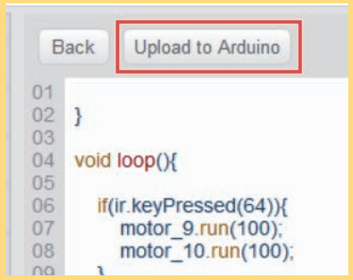
Idea	Script	Script Description
Click the green flag to execute the loop structure. It will continuously detect the received key information of the remote controller. If it receive the arrow, mBot will move or turn.		 Click the green flag and execute the program  The loop program is used for continuously detecting data information  Analyze the key information of the remote controller and execute the corresponding program.

Tips

You can set the keypad of the remote controller according to your preferences, for example, press A and mBot play a piece of sound.

Example Expansion

We are able to use the arrow keys on the remote controller to control mBot's movement. However ,I have to remind your of a problem : mBot moves very hard. When we control it with an infrared remote controller , the signal is firstly received by the infrared receiver of the main controller and then fed back to mBlock of the computer through Bluetooth. After the signal is processed by mBlock, the motor rotation of mBot is controlled by Bluetooth. This process is very complicated, isn't it? Is there a way to disconnecting mBot from computer? In other words, the program is directly burned into the main controller. It is definitely feasible. Let's learn this function now.

Idea	Script	Script Description
Use "mBot main program" in the "robot module" to replace "when the green flag is clicked". Upload the code to the main controller.		 Allow program code to be uploaded to mBot's main controller and make the robot operate off-line Then right-click, select "Upload Arduino program", as shown below  Click the "Upload to arduino" in the opened window  Attention! When you upload Arduino program, please make sure that your USB cable is connected to mBot's main controller and the corresponding port number is

Tips

When the program is successfully uploaded, you could control mBot with an infrared remote controller , If you write a very complicated program (such as using the Scratch brush, recording and other functions), it may not generate the corresponding code. Reminder page will be prompted and indicate where goes wrong. Therefore, It is recommended that you should not write the program for complex functions.

Exercise

1.Upload code to the main controller. It can control mBot's movement with the remote controller . It can also control the buzzer sound and LED flash effects and so on.

Chapter 13 The Versatile Robot

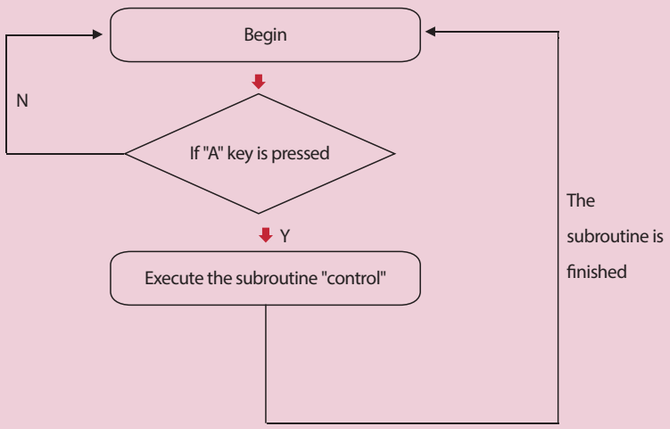
mBot is a smart robot. It has learned a lot of skills. For example, it is able to deftly dodge obstacles ahead. It can also perform line- patrolling and operate the infra-red remote controller. In this lesson, we will have mBot shown its all skills!

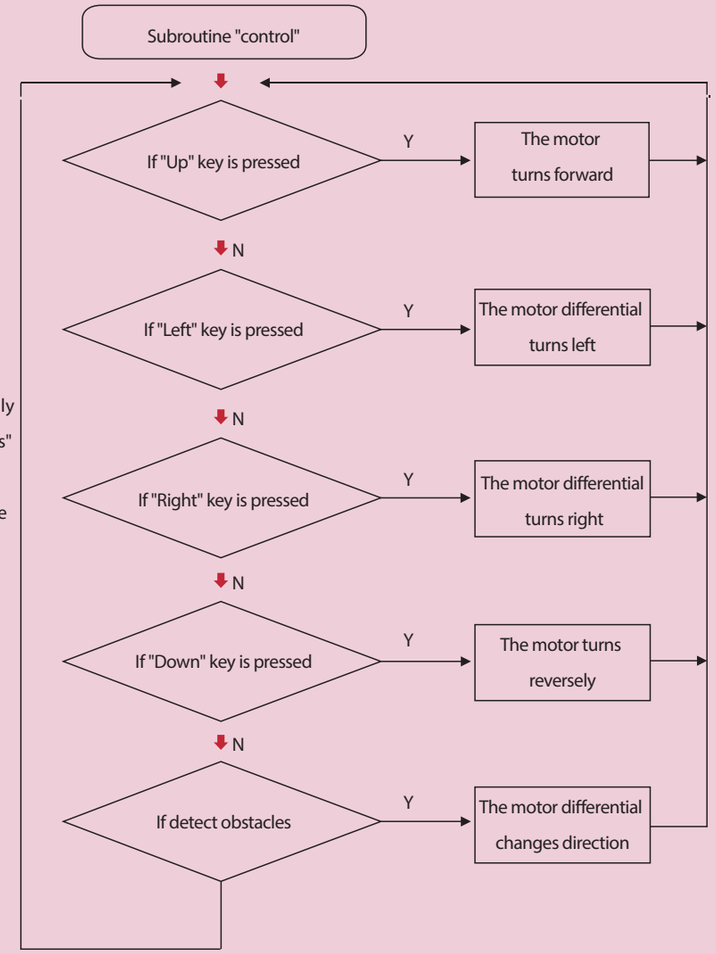
Learning Objectives

1.Press "A" key of the remote controller. mBot enters the remote control mode. Use the arrow keys to control mBot to move forward, backward and turning. However, if there is any obstacle 10 cm ahead, mBot will automatically turn.

2.If you press "control key" on the remote controller or the line-patrolling sensor hits the black line,

Programming Ideas

The Idea Description	Thinking Process
The program is burned into the controller. When it is executed, it will continuously judge whether button "A" is pressed. If pressed, the subroutine "control" is executed. The subroutine "control" repeatedly executes the program until the key "Set" is pressed or the line sensor detectes the black line. In the	

The Idea Description	Thinking Process
"control" program, it controls the movement direction of mBot with the button of the remote controller. However, if there is any obstacle ahead (within 10 cm), mBot will automatically turn. Run repeatedly until "Settings" button is pressed or the line sensor detects the back line	

Try it

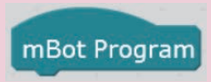
1.Write a script. Judge whether "A" on the remote controller is pressed

Idea

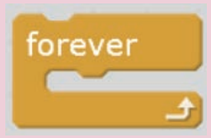
In the loop body ,mBot's main program judge whether "A" key on the remote controller is pressed. If yes, the subroutine "control" will run.

Script

Script Description



The program can be burned directly into the controller to achieve mBot running without the computer.



The program that is burned into the main controller must be executed in a loop

Judge whether "A" key on the remote controller is pressed

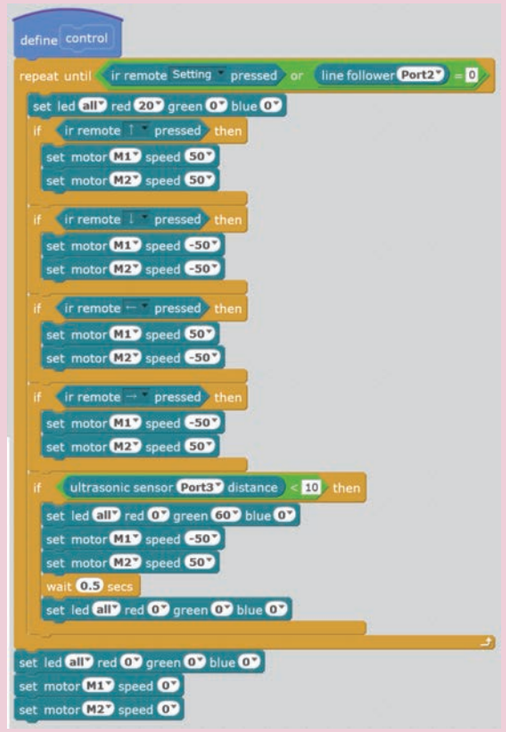
To execute the subroutine "control", you should create a subroutine in "Data and Blocks" module, shown as below:

2.Write Subroutine "control"

Idea

The program runs repeatedly in the subroutine "control" until "Settings" button on the remote controller is pressed or the line sensor detects the black line. In the loop, it will detect the state of keys of the remote controller, and judge whether there are obstacles 10 cm ahead.

Script

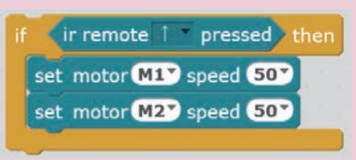


Script Description

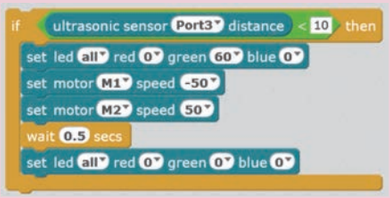
Execute the subroutine "control".

Repeatedly execute the program until "Settings" button on the remote controller is pressed or the line sensor detects the black line (the value of the sensor is 0). Then turn off the small LED lights and stop rotation of the motor.

Set LED light to red color, which indicates that mBot enters the control mode.



According to different keys of the remote controller, the movement direction of mBot. varies.



If there are obstacles 10 cm ahead, the LED light is in green and robot turns. The small light will be off in 0.5 seconds

Exercise

1.Apart from making mBot enter the control mode by pressing "A" key of the remote controller, you can also set other modes. For example, press "B" key and enter the line-patrolling mode, and then mBot travels along the black line, etc.

Chapter 14 The Horse Race Tycoon

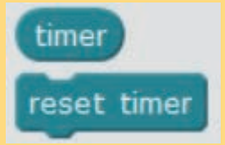
Today, we come to the racetrack. Each contestant brings mBot. They will calibrate their beloved mBot by themselves, so that it achieves the best results in this racing game. Before the game starts, players have plenty of time to train the mBot. Game time, all go to the track please!

Learning Objectives

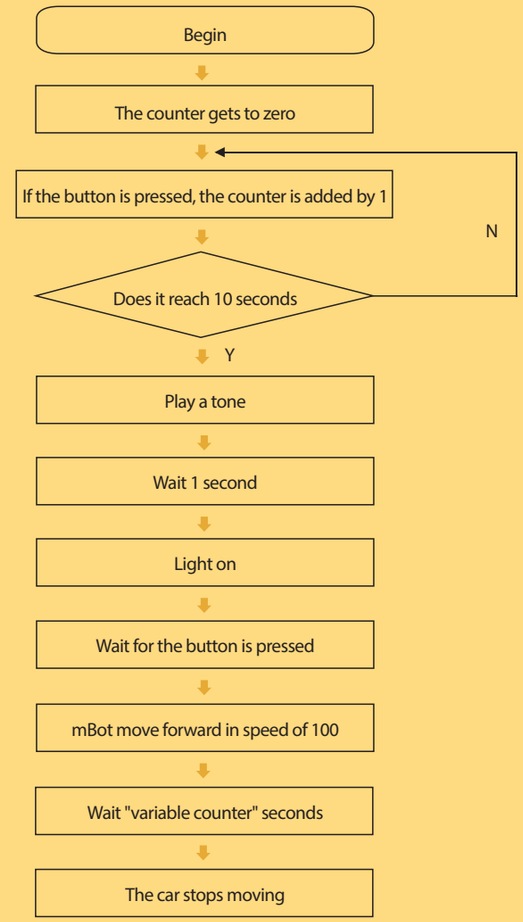
1.Upload the control program to mBot in offline mode and finish the racing game

Knowledge Points

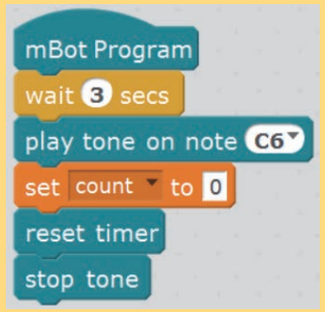
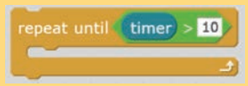
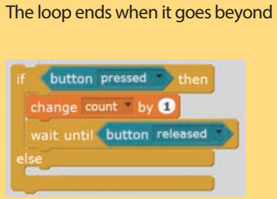
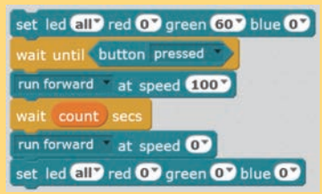
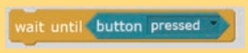
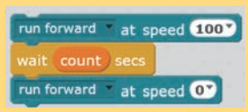
Building Blocks	Instructions	Example
Robots	Right-click the main program building blocks of mBot	
mBot Program	Upload the program to the mCore and off-line run mBot.	
		Select "Upload Arduino Program" and the Arduino code appears on the right of software.

Building Blocks	Instructions	Example
	Achieve timing function with the timer blocks,	Timer is a built-in function of Arduino chip, which is equivalent to mBlock timer.
	Set blocks for motor turning and speed	It could control turning and speed of two motors simultaneously.

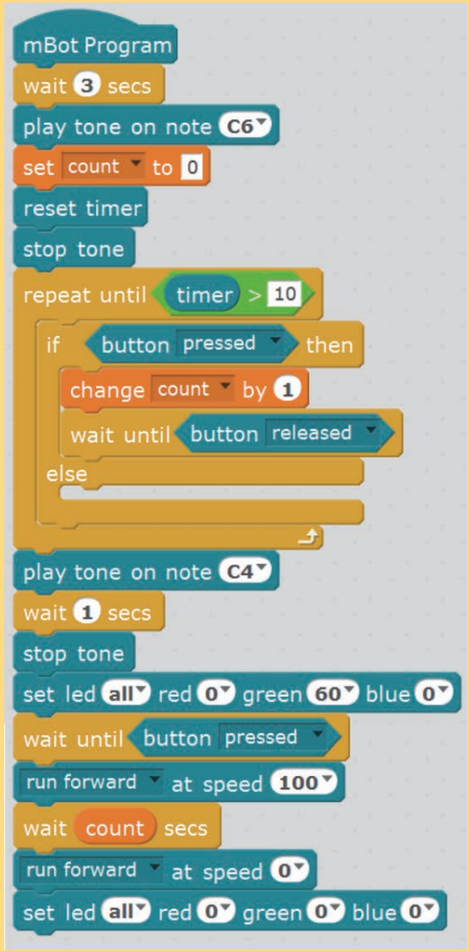
Programming Ideas

The Idea Description	Thinking Process
mBot determines the travelling distance according to the numbers of button pressing, so it must run mBot offline. How does our program run? 1.First set the initial state. 2.Record the numbers of pressing mBot button within 10 seconds. 3.When the green light is on, put mBot at the starting pint of runway. Press the button again and then mBot will move forward and stop in the end.	

Try it

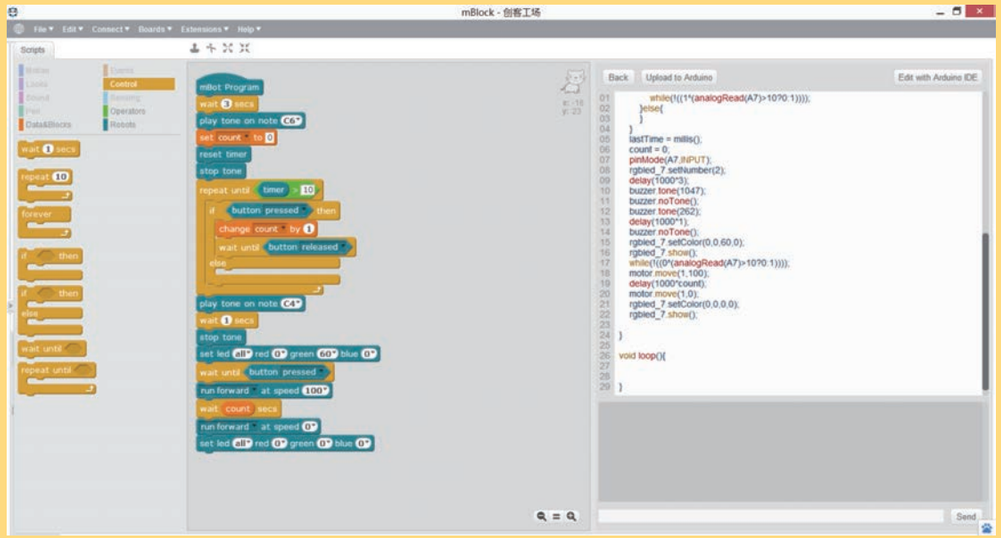
Idea	Script	Script Description
Initialize the variable applied to mBot before it sets off		 The variable "count" records the numbers of pressing button. If the numbers does not get to zero before the program starts, the following program will add the previous numbers. The program will run abnormally.  Start recording time again.
Detect the number of times that the button is pressed within 10 seconds. Emit a tone in 10 seconds.		 The loop ends when it goes beyond 10 seconds.  Record the number of times that the button is pressed.
The green light turns on when time is up. Players put mBot to the specified location. Press the button again and mBot starts moving.		 Wait for pressing the button.  Forward time is equivalent to the number of times that the button is pressed by the player.

The completed Arduino source code when the Arduino program is uploaded

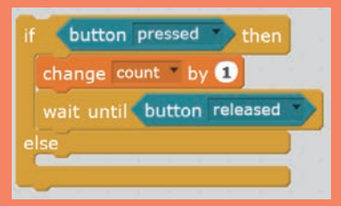


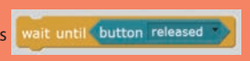

```
#include <Arduino.h>
#include <Wire.h>
#include <Servo.h>

#include "mBot.h"
#include "MePort.h"
MeBoard myBoard(mBot);
#include "MeBuzzer.h"
#include "MeRGBLed.h"
#include "MeDCMotor.h"

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double count;
MeBuzzer buzzer;
double currentTime = 0;
double lastTime = 0;
MeRGBLed rgbled_7(7);
MeDCMotor motor(0);

void setup(){
   while(!((currentTime) > (10)))
 {
      if((0^(analogRead(A7)>10?0:1))){
         count += 1;

while(!((1^(analogRead(A7)>10?0:1))));
      }else{
      }
   }
   lastTime = millis();
   count = 0;
   pinMode(A7,INPUT);
   rgbled_7.setNumber(2);
   delay(1000*3);
   buzzer.tone(1047);
   buzzer.noTone();
   buzzer.tone(262);
   delay(1000*1);
   buzzer.noTone();
   rgbled_7.setColor(0,0,60,0);
   rgbled_7.show();

while(!((0^(analogRead(A7)>10?0:1))));
   motor.move(1,100);
   delay(1000*count);
   motor.move(1,0);
   rgbled_7.setColor(0,0,0,0);
   rgbled_7.show();
 }
void loop(){
   }
```

Click the "Upload to Arduino" button and upload the programs to the mBot mainboard.



Tips

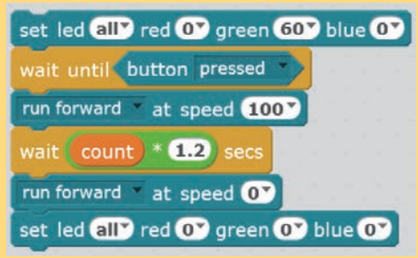


Why is 

added to if .. then block? Because it is only countered as successful pressing when the button is pressed and then released. If you do not add this building block, it will be considered as repeated pressing when you hold down it. It is not the effect that we want.

Will there be any difference if we use if..then block in the script? Try it. Try again, will mBot react if the time you press the button is more than 10 seconds?

How could you change the moving distance per second for mBot?

Idea	Script	Script Description
Modify the waiting time in the program		 Extend the waiting time by 20%. Try  and see the difference.

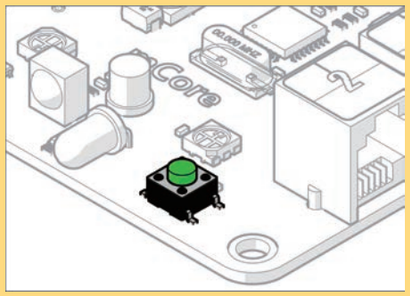
If you think mBot moves too fast, how could you set it?

Idea	Script	Script Description
Modify the motor speed		 You can change the moving speed by modifying this number

Example

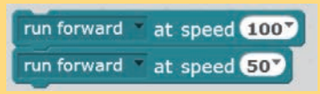
Touchdowns game: Let mBot start from a fixed starting point and move to the specified range with the button.

1.Playground Configuration

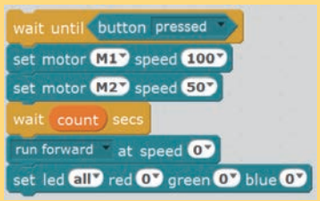


Determine the number of times of button –pressing through user evalutaion. Make mBot enter the specified range.

2.The Modifiable Motor Speed Block

Idea	Script	Script Description
If the target range is not in front but a route with radian		It is a good solution that we set motor at different speed. Maybe you have better idea ~

3.Write the Script

Idea	Script	Script Description
Move in a curve way		The motor speed setting is adjusted to the best travelling curve of mBot

Exercise

1.If there is an obstacle within target range, how would you solve this problem? More interesting games are waiting for your development!