NASHENBOT

Nashenbot Training

--- MPBOT 609

Micro-brain programming

Key-screen interaction

MPC Chip

Core patents

Mpbot 609: No computer programming is required, one set of machines can meet the needs of class.

1. Core parts parameter

NASHENBOT

Model	MPBOT 609
Scenario	Standard kit
Image	
Controller (with lithium battery)	1
Motor	2
Spindle box	2
Infrared sensor	1
Sound sensor	1
Red light	1
Green light	1
Blocks	109

1. Core parts parameter

NASHENBOT

Multiple sensors and motors

Controller

Motor

Spindle box

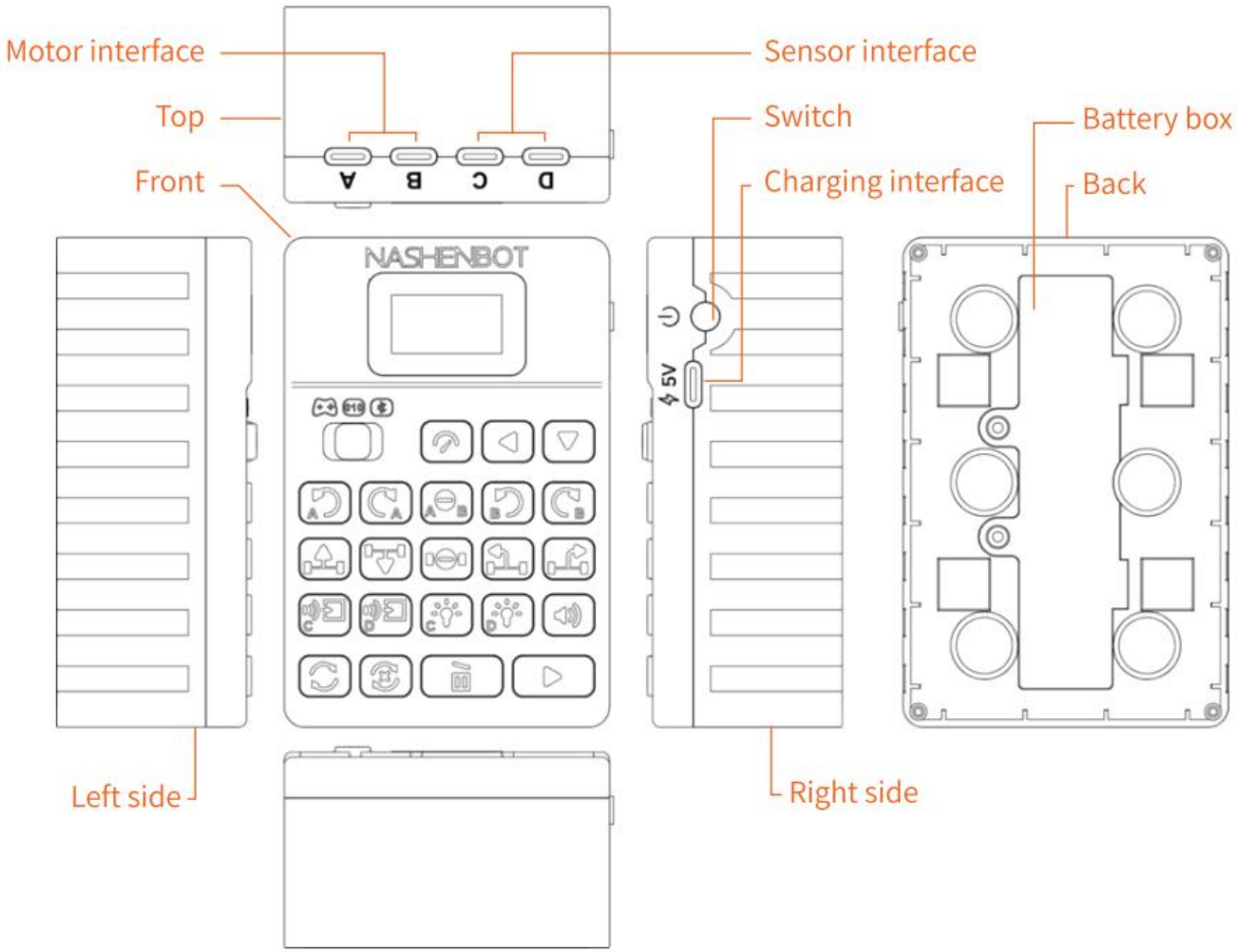
Red light

Green light

Infrared sensor

Sound sensor

2.1 MPC controller



2.1 MPC controller

NASHENBOT

2.1 Top surface

Icon		
Name	Sensor interface (Shared by input and output)	Motor interface
Parameter	USB Type-C; Support infrared obstacle avoidance, sound intensity detection, one-way follow, collision switch and other sensor input control; Support red light, green light and other actuator output control	USB Type-C; Support control of motors

2.1 MPC controller

NASHENBOT

2.2 Right side

Icon	⏻	ϟ 5V
Name	Switch	Charging interface
Parameter	Press and hold for about 1 second to turn on or off	USB Type-C interface; Support 5V1A charging; Built-in charging status indicator

2.3 Back side

Icon	
Name	Battery compartment
Parameter	3.7V lithium battery; Support charging through the interface ϟ 5V

2.2 Mode

After powering on, you need to select the programming mode first.

The mode is radio:

Real-time Mode	All the above keys are valid;
	Command Display the currently pressed key in the command area.

2.2 Mode

NASHENBOT

010

Key screen

programming

In this mode, the first-level interface is displayed for one second and automatically enters the second-level interface (the key screen programming state);

All buttons on the front are valid;

The program example is as follows:

01	01 A1 A A B7 02 A9 C C	03 A1 A A B7 04 A9 C C B2

01 The number here represents the number of program lines, and the program capacity of different lines is supported according to the robot model;

A1 This number represents the parameter of the program (seconds or times);

A The letters here represent the parameters (ports) of the program;

C The horizontal bar here indicates that the program is selected (you can delete it or insert a new program backwards).

2.2 Mode

Bluetooth connection	This is the Bluetooth connection mode, which supports the interconnection of two Micro-brain programming controllers (follow the NASHENBOT Official Account for detailed usage methods). 

The power is relative value: 0~100% corresponds to a voltage of 3.5~4.2V±0.2V.

When the power is lower than 5%, the buzzer will send out a low-battery alarm prompt of "Di Di Di Di", and at the same time, it will automatically stop the motor and other program modules, and automatically shut down after 5 seconds. At this time, you need to use the charging cable that comes standard with the product for compliance charging (5V1A).

During charging, the charging status light is red, and the microcomputer automatically turns into the shutdown state; the theoretical charging time is about 2 hours.

2.2 Mode

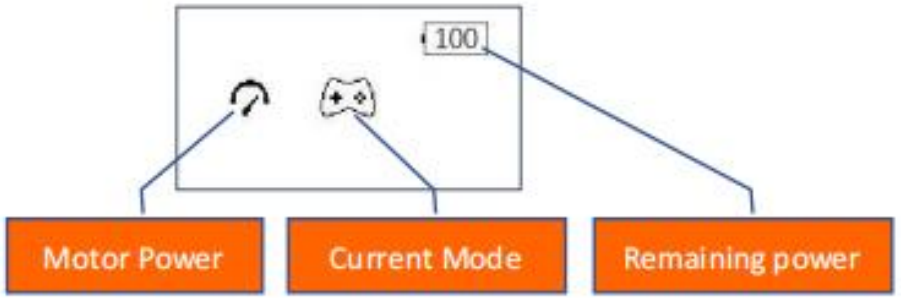
Bluetooh mode: how to connect between two controllers?

No.1 controller: Remotely controlled controller

Step 3

No.2 controller: as the remote control

2.3. Power switch



The motor power is a global command, divided into two types: high speed and low speed . It can be used to control the rotational power of A and B motors.

Power switch
Supports two powers: 1. High speed 2. low speed All three modes are valid

2.4. Button operation

◀	▼
Caret	Turn pages
Support for inserting program modules; Only valid in 010 key screen programming mode	Support page down; Only valid in 010 key screen programming mode

🗑	▶
Delete program	Run the program
Short press to delete the currently selected program	Short press to run or stop the program

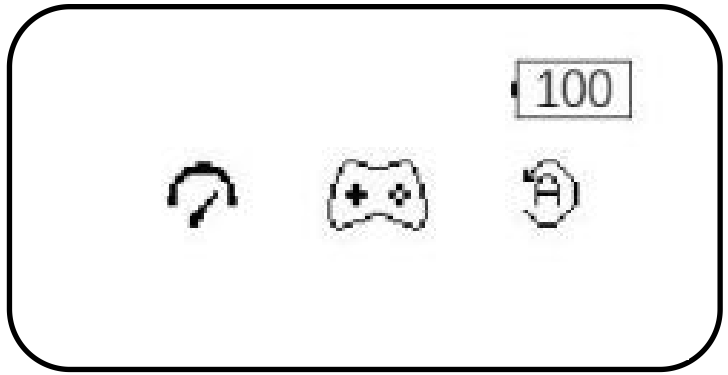
In the key screen programming mode, clicking ▶ **Run"** will execute the completed programming statements, but the program will not automatically exit when it finishes running. If you want to change the program, you need to click ▶ "**Stop"** again before making changes.

2.5. Single motor control programming

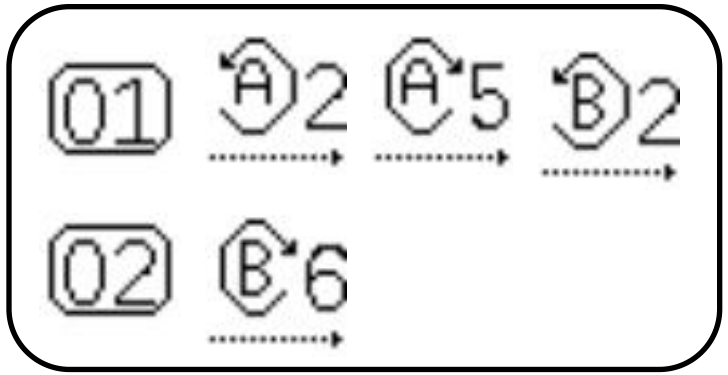
Pay attention to the one-to-one correspondence of the motor control interface. Generally, infinite motion is used in conjunction with sensors.

Icon					
Name	A motor reverses	A motor rotates forward	A or B motor stops	B motor reverses	B motor rotates forward
	Control A motor to rotate in reverse	Control A motor to rotate forward	Control A or B motor to stop	Control B motor to rotate in reverse	Control B motor to rotate forward

2.5. Single motor control programming example

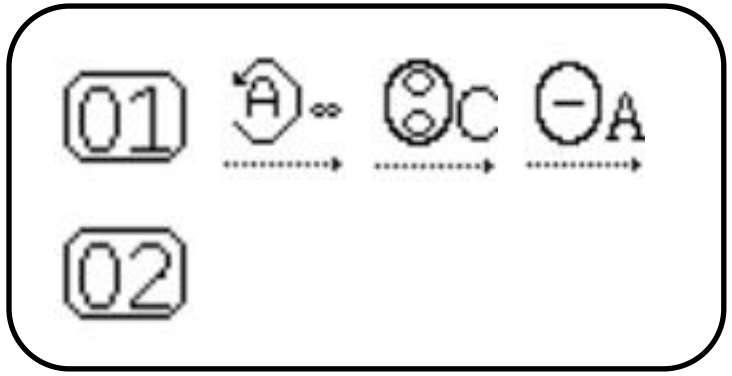


Press to turn, release to stop turning



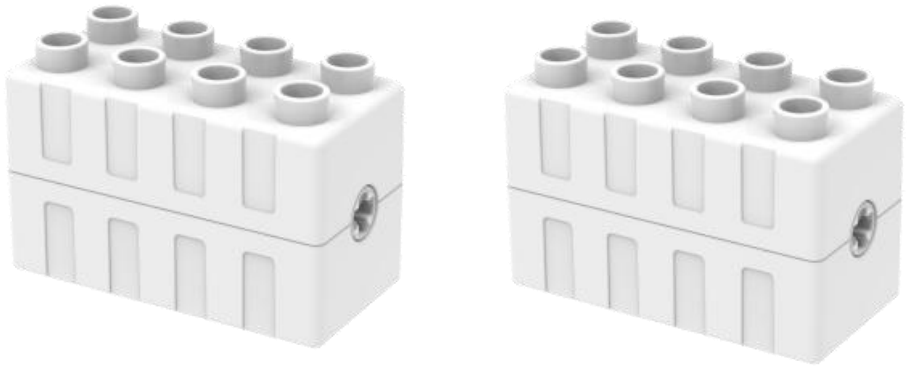
The two motors rotate alternately

A motor keeps turning



Motor A keeps turning until the sensor is triggered and stops.

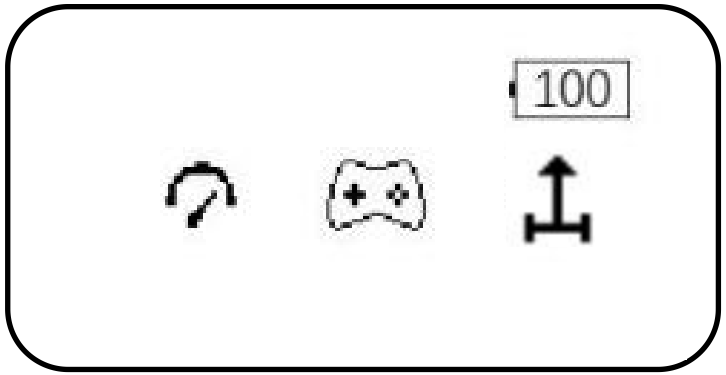
2.6. Dual motors control programming



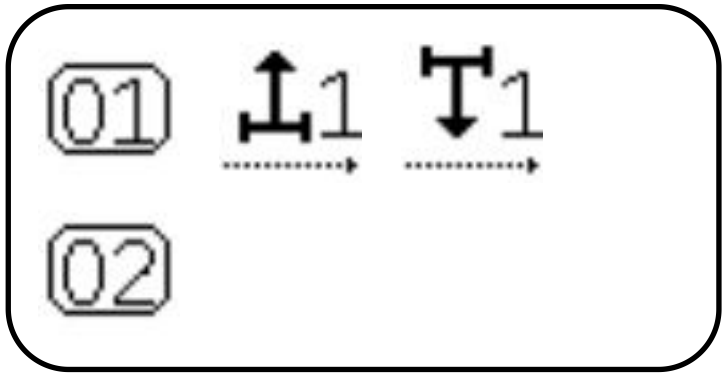
The dual motors are sometimes used separately, but more often they are used to control the car's shape and movement. This can vary depending on the build and wiring. If the forward button causes the car to move backward, swap the motor connectors or change the direction of the car's front.

Icon					
Name	Forward	Back	Stop	Turn left	Turn right
Parameter	Control the A motor to reverse and the B motor to forward at the same time	Control the A motor to rotate forward and the B motor to rotate reversely at the same time	Control A and B motors to stop at the same time	control the A motor and the B motor to rotate forward at the same time	control the A motor to reverse and the B motor to reverse at the same time
	010 Key screen programming mode: short press to set 1~9 seconds and unlimited time		-	010 Key screen programming mode: short press to set 1~9 seconds and unlimited time	

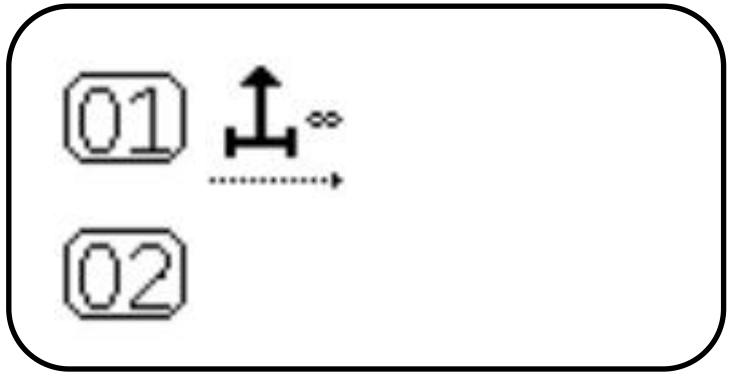
2.6. Dual motors control programming example



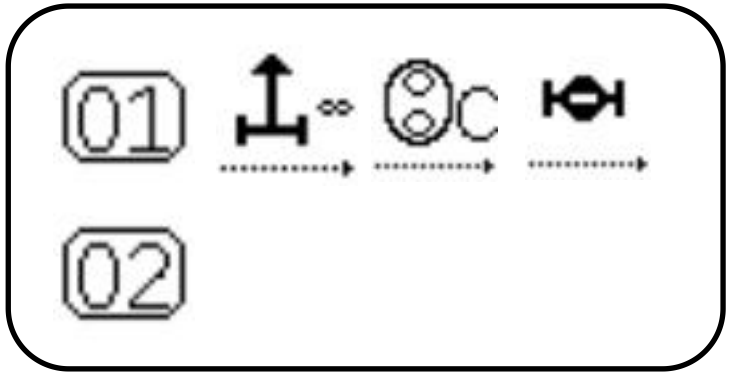
Press to go forward, release to stop



Move forward one second and then stop



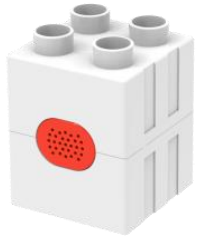
Keep moving forward



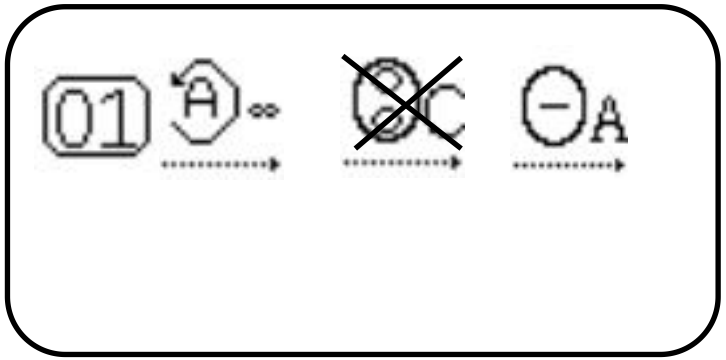
Keep going until the sensor is triggered and stops.

2.7. Sensor programming

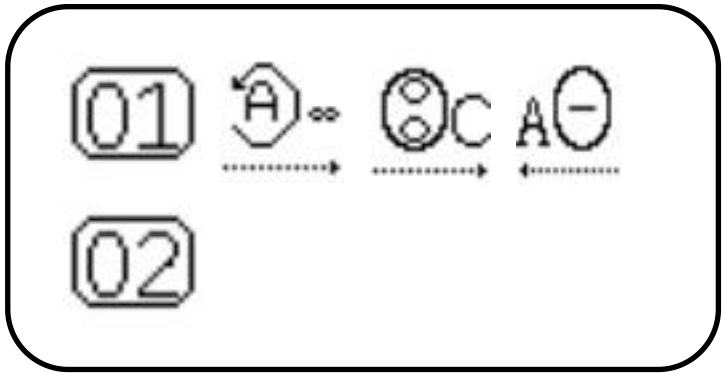
Icon	C	D
Name	C interface sensor detection	D interface sensor detection
Parameter	Short press to select infrared or sound sensor for logical judgment Only valid in 010 key screen programming mode	



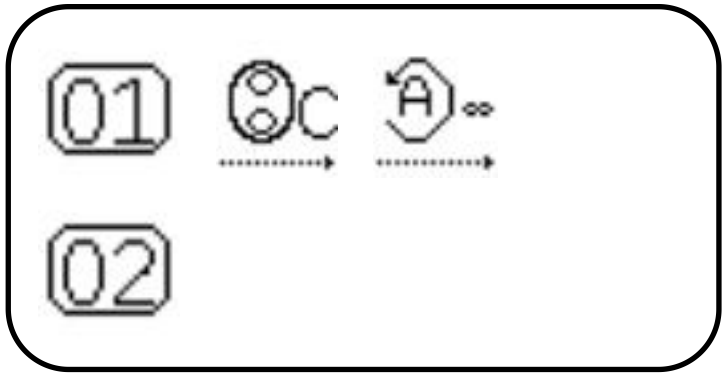
Sensor identification is based on the wiring interface. The program will wait until the sensor is identified before continuing. If a sensor is marked with a cross, it could mean the wrong sensor was selected or the wiring is not securely connected. Placing the sensor trigger in different locations will produce different effects.



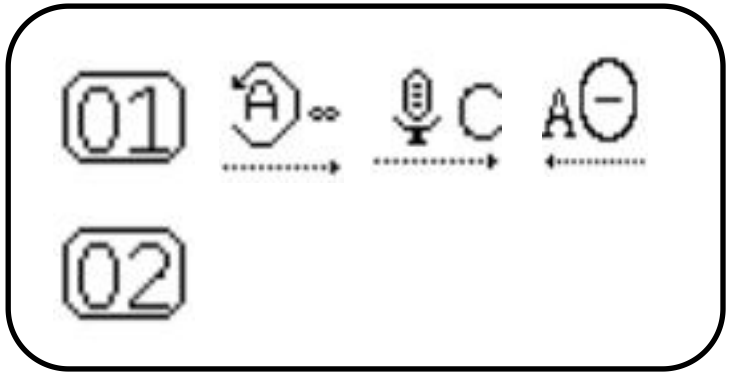
2.7. Sensor programming example



Keep going forward and stop when you encounter an obstacle.



Don't move until an obstacle is in front of you. It doesn't matter whether there is a still icon in front of you or not.



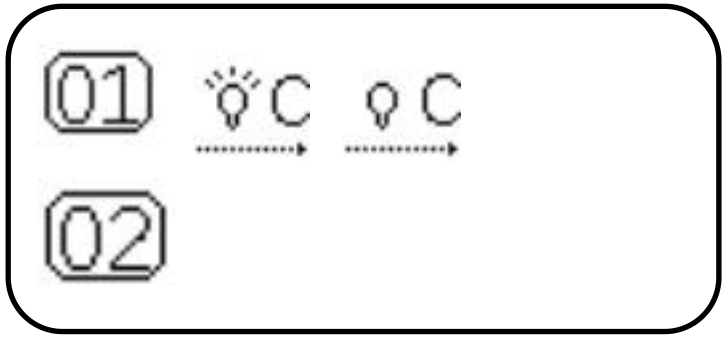
Keep going and stop when you hear a loud noise

2.8. Lights

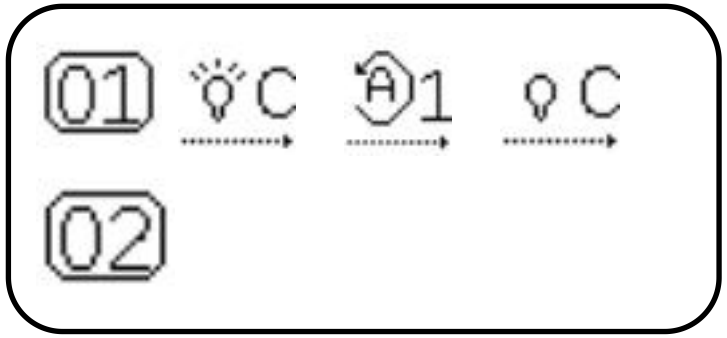
C	D
C interface light switch	D interface light switch
Short press to turn on or off the C interface lights	Short press to turn on or off the D interface lights

The red and green lights are used in the same way. They are controlled on and off individually, with the timing determined by the program execution time in between. To achieve blinking, alternate the two lights on and off.

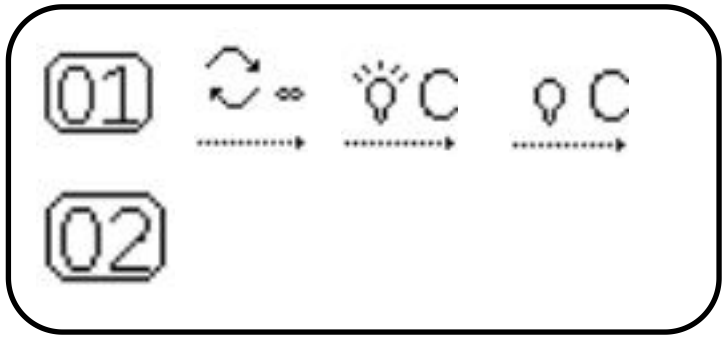
2.8. Lights - example



The light turns off after 0.01 seconds



The light turns on, the motor rotates for 1 second, and then the light turns off



Light flashes quickly

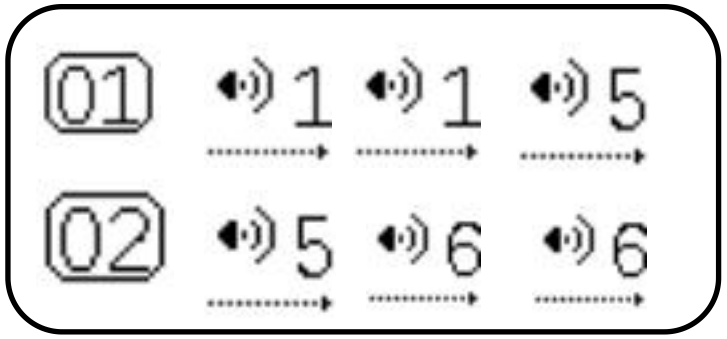
2.9. Buzzer

Turn on the buzzer
010 key screen programming mode: short press to set 1~7 audio

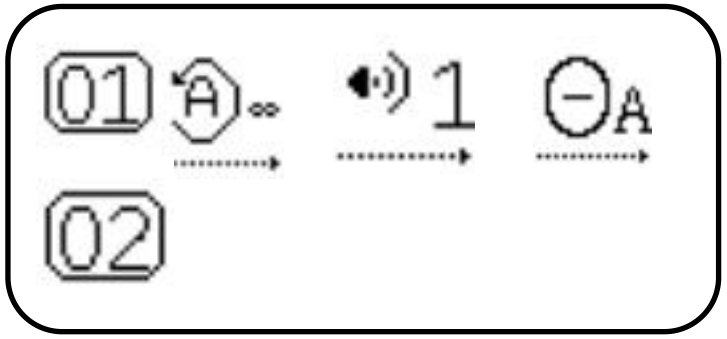
Buzzer button

The buzzer is built into the controller and does not require a separate connection. The buzzer sound has a specific duration. The minimum unit of motor movement time is 1 second. If you want to move for 0.1 seconds, you can use the buzzer as the interval. If you want to move for 0.01 seconds, you can use the light as the interval. This interval is the time it takes for the program to take effect.

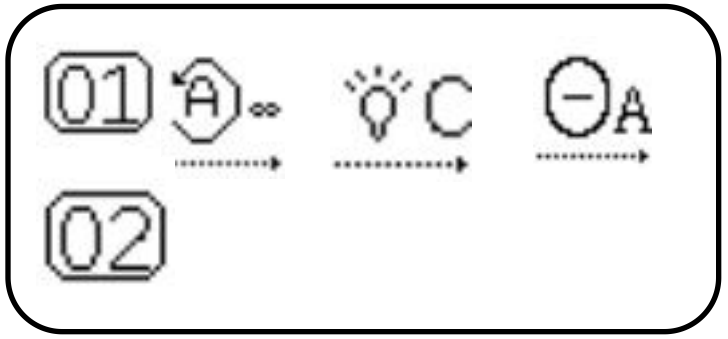
2.9. Buzzer - example



Write a song: Twinkle Twinkle Little Star



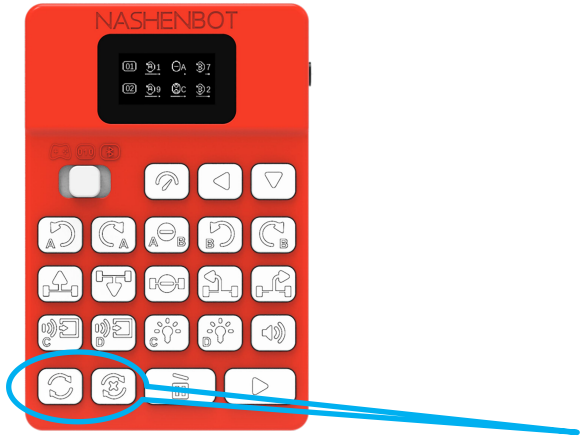
The buzzer makes an interval and the motor stops after rotating for 0.1 seconds.



The light can also be set to intervals, and the motor will stop after 0.01 seconds.

3.0. Loop

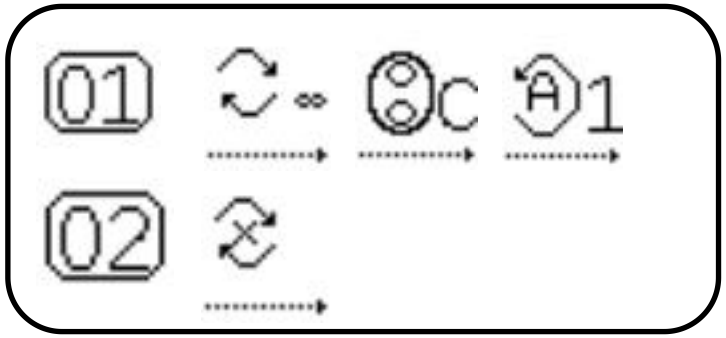
Icon		
Name	Start loop	End loop
Parameter	Short press to set infinite and 2~9 cycles	-



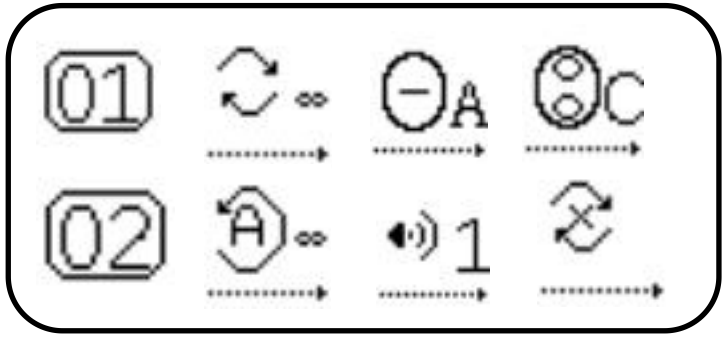
Loop button

Loops are usually added before a program to be executed repeatedly. It's a good idea to pair the start and end loops. This won't affect a single loop, but you need to be careful when your program has multiple loops.

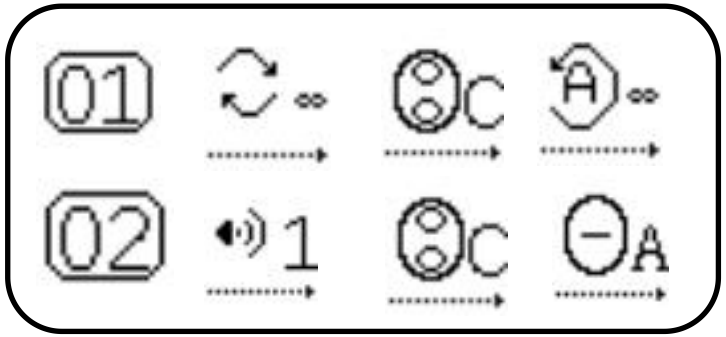
3.0. Loop - example

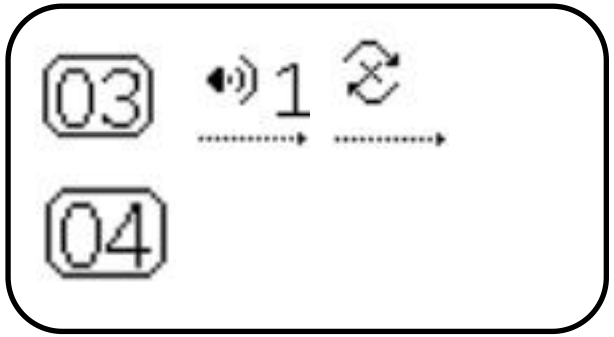


Every time there is an obstacle, turn for one second and then stop.



Each time there is an obstacle, it moves for 0.1 seconds





It keeps moving every time there is an obstacle, and stops when there is another obstacle.

The 5th Generation Educational Robot

— Artificial Intelligence Programming Robot

Thanks !