Starter Kit

Project Guide

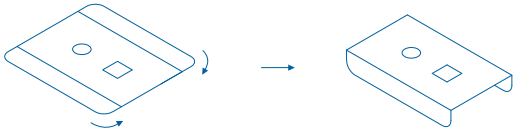
Grove Zero

Table of Contents

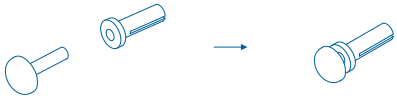
Tips

1 Before you start a project, try to find the project sheet first (except for project #1). Some of the projects have more than one parts – Part A, Part B and others, find them all and start the project.

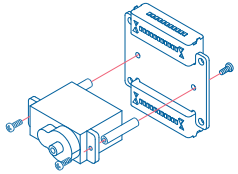
2 **Fold lines** can be found in every project sheet. The key to a good construction is to crease firmly along the fold lines. Always check where the fold lines are and follow the instructions to fold them in proper order.



3 Rivets can be used to connect different parts, sometimes you'll need to take apart the rivet to use it.



4 Mount Servo module



Now let's get started to make!

#1 Burglar Alarm

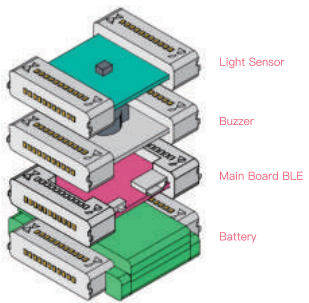
Description:

Let's make a Burglar Alarm to protect and secure your snacks in the bag, your coins in the saving pot, and your secret dairy in the drawer! Open your bag(or drawer) or moving your saving pot will set off the alarm, then you'll hear the buzz and catch the 'bad' guy right away!

How it works:

Main Board BLE: Receives sensor data from the light sensor and compares it with a reference value, then decides whether or not to trigger the buzzer.
Light Sensor: Measures the nearby light intensity and converts it into digital signals, then sends the signal to Main Board BLE.
Buzzer: Makes annoying sound when it receives signals from Main Board BLE.
Power: Provides electricity to the circuit.

Build circuit like this



Project Type
Snap & Play

Level
Easy

Time
10 minutes

You need

- Main Board BLE
- Battery
- Light Sensor
- Buzzer

Tips

Make sure the light sensor is able to catch the light and trigger the buzzer when someone open your bag or drawer.

#2
Music Visualizer

Watch those beats! Let's invent a device that can visualize the nearby sound intensity. Put it next to your music player to see the music beats, or snap your fingers around it with your friends, see who can create the loudest clicking sound.

Project Type
Snap & Play

Level
Easy

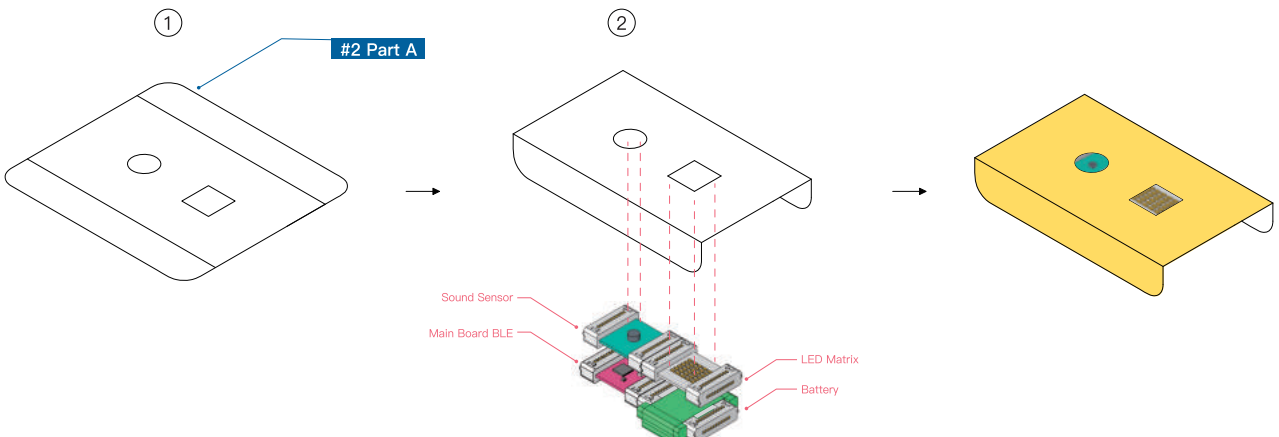
Time
10 minutes

You need

#2 Part A &

- Main Board BLE
- Battery
- Sound Sensor
- LED Matrix

#2 Music Visualizer



How it works:

Main Board BLE: Receives sensor data from the Sound Sensor (range from 0 to 1023), then maps it to LED Matrix (range from 0 to 18) as a LED Bargraph.

Sound Sensor: Measures the nearby sound intensity and converts it into digital signals that range from 0 to 1023, then sends the signal to Main Board BLE.

LED Matrix: Displays and updates the received LED Bar Level which represents the current sound intensity, 20 times per second.

Power: Provides electricity to the circuit.

#3
Night-light

A little bit of light in a dark room can always make people feel safe, comfort and less anxious. Make a cool night–light that can automatically light up in the dark area. Place it in in your bedroom, kitchen or anywhere you need to avoid turning on the main light in late night, stripping over obstacles or pets, or simply to relieve the fear of dark.

Project Type
Snap & Play

Level
Easy

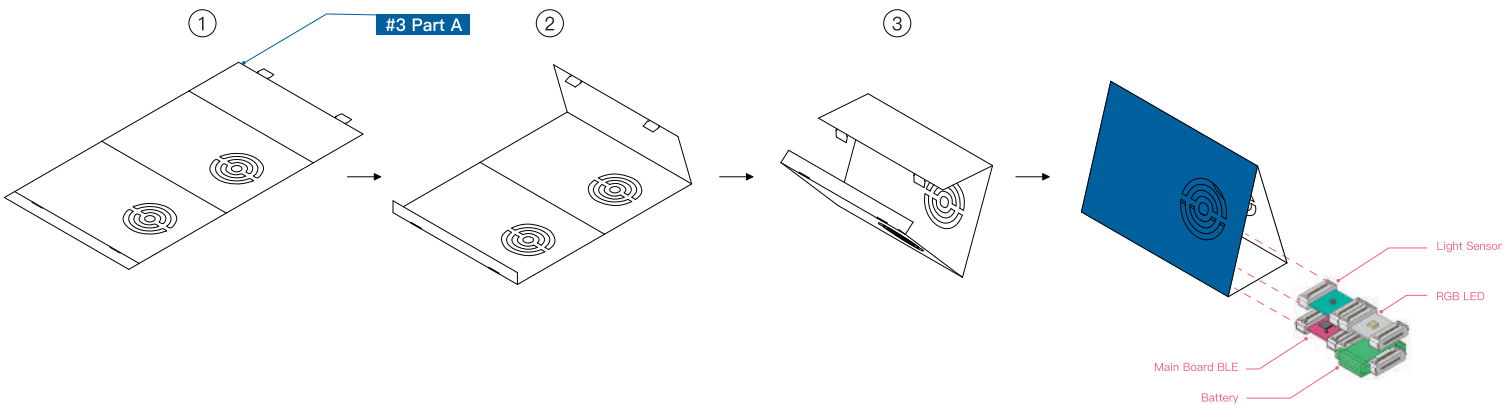
Time
10 minutes

You need

#3 Part A &

- Main Board BLE
- Battery
- Light Sensor
- RGB LED

#3 Night-light



How it works:

Main Board BLE: Receives sensor data from the light sensor and compares it with a reference value, then decides whether or not to trigger the RGB LED.

Light Sensor: Measures the nearby light intensity and converts it into digital signals, then sends the signal to Main Board BLE.

RGB LED: Continuously generates red, green and blue light at different intensities to show rainbow light.

Power: Provides electricity to the circuit.

#4
Sound Meter

Noise can be exciting, silence can be relaxing. Too much noise can be stressful, and too much silence can be boring! We are going to make a Sound Meter, a device that will show you the sound level at any place you are at. Then you are able to record the sound levels from different places in your lives, and compare them with friends.

Project Type
Snap & Play

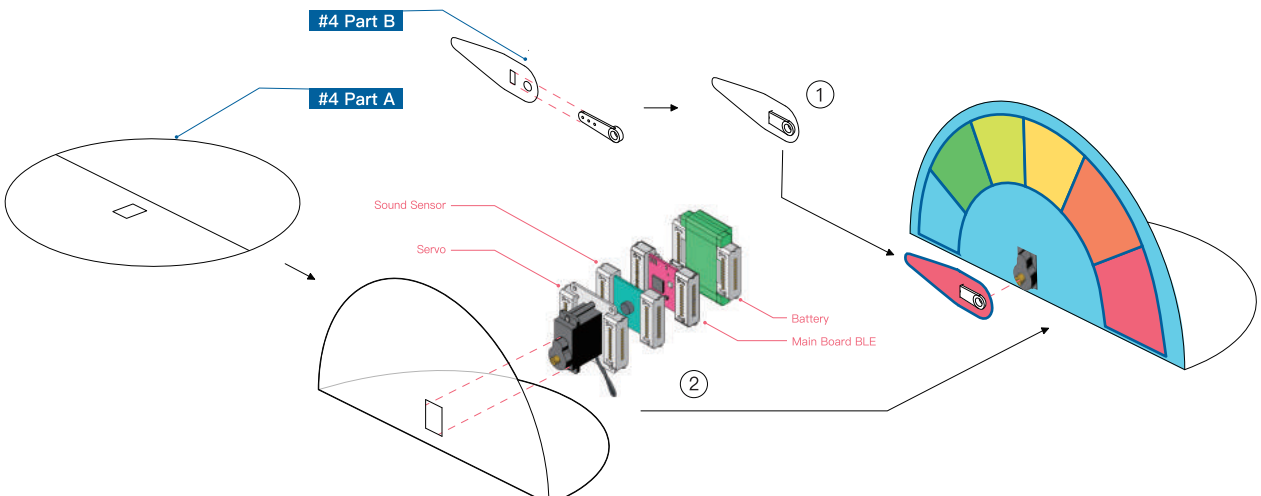
Level
Easy

Time
15 minutes

You need

#4 Part A
#4 Part B
&
- Main Board BLE
- Battery
- Sound Sensor
- Servo

#4 Sound Meter



How it works:

Main Board BLE: Receives sensor data from the Sound Sensor (range from 0 to 1023), then maps it to the angular position of Servo (range from 0 to 180).

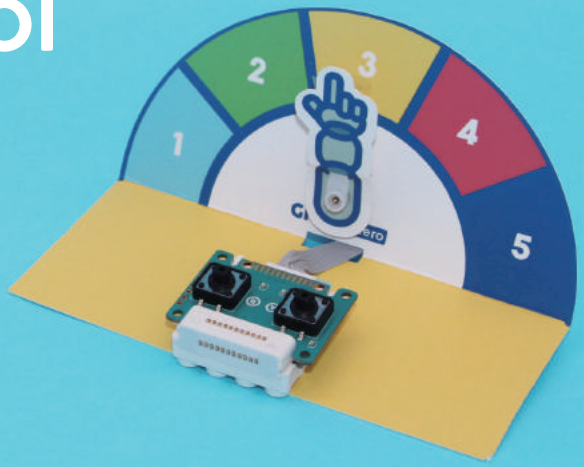
Sound Sensor: Measures the nearby sound intensity and converts it into digital signals that range from 0 to 1023, then sends the signal to Main Board BLE.

Servo: Rotates its shaft and the pointer attached on it. The louder the voice, the wider angle it moves.

Power: Provides electricity to the circuit.

#5
Robot Arm Control

Servo motors are the most essential parts of robotics, every cool robot starts from the servo control. Now let's make a robot arm and a controller. Use the controller to move the robot arm to specific positions.



Project Type
Snap & Play

Level
Easy

Time
20 minutes

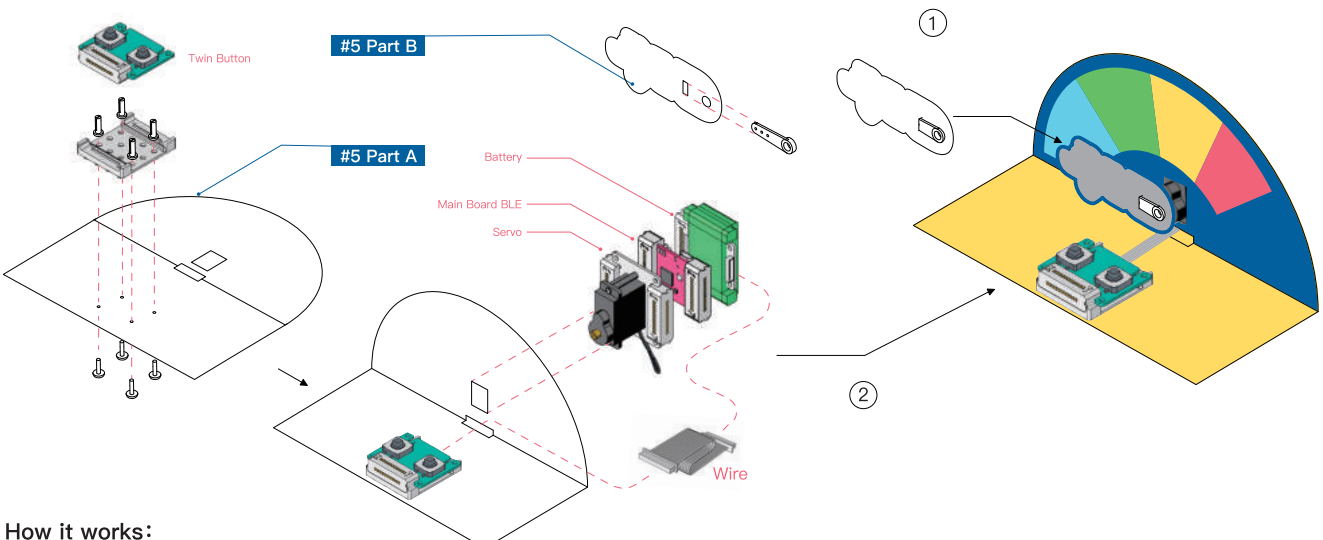
You need

#5 Part A
#5 Part B

&

- Main Board BLE
- Battery
- Twin Button
- Servo
- Wire

#5 Robot Arm Control



How it works:

Main Board BLE: Receives button events from Twin Button, then sends the control signals to Servo.

Twin Button: Detects button events and reports to Main Board BLE.

Servo: Rotates its shaft and maintain it at a specific angular position. The arm rotates along with the shaft.

Power: Provides electricity to the circuit.

#6
One-eyed Monster

Make a creepy little ghost that sleeps at day and awake at night. Put it at the bedside as a night-light, or hide it somewhere in the house to trick your friends.

Project Type
Blockly Programming (Codecraft)

Level
Medium

Time
20 minutes

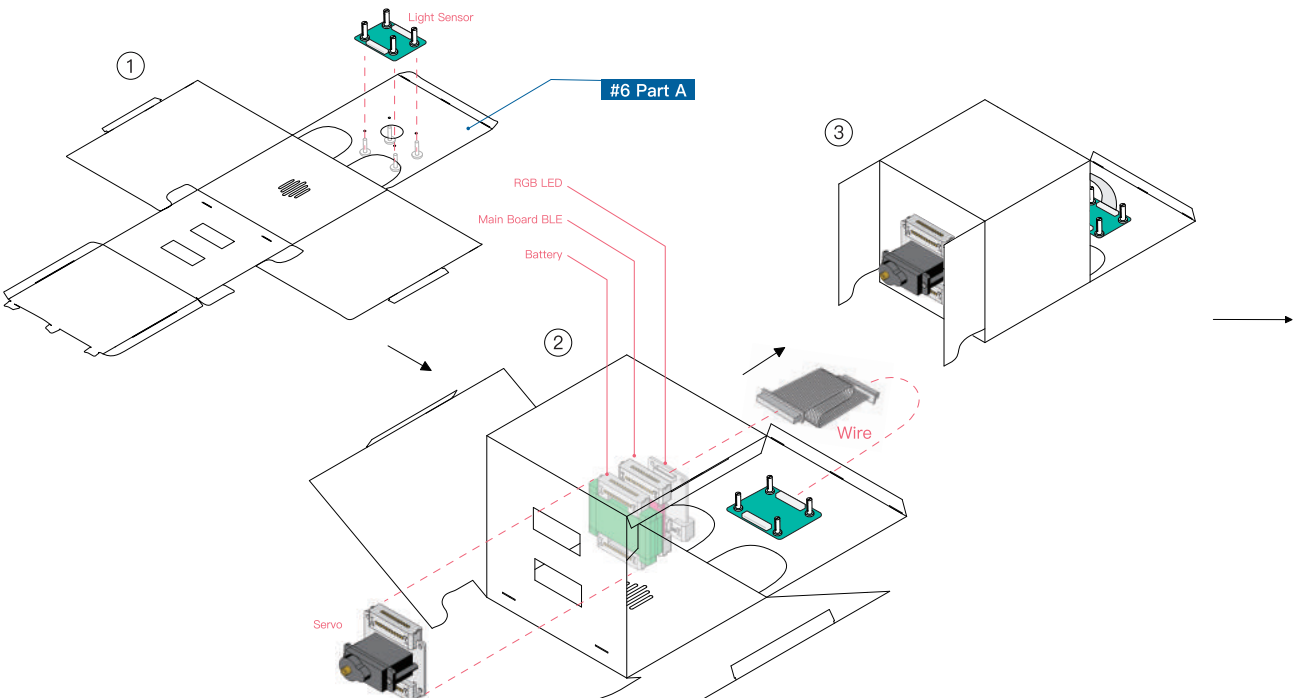
You need

- #6 Part A
- #6 Part B
- #6 Part C

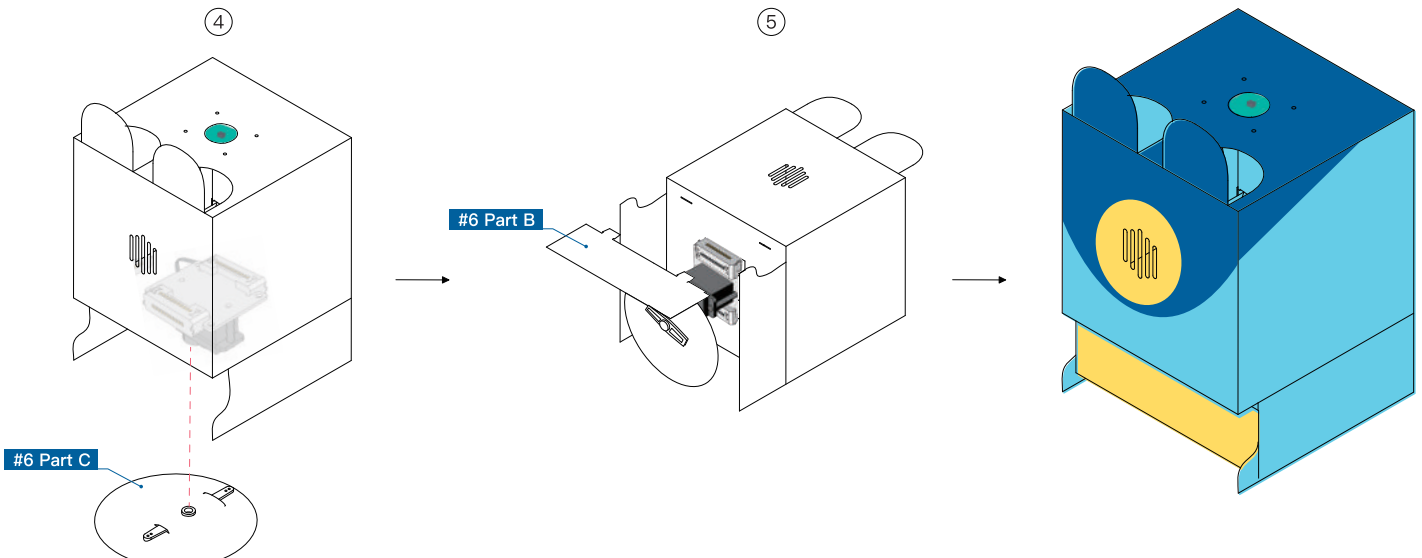
&

- Main Board BLE
- Battery
- Light Sensor
- Servo + RGB LED
- Wire

#6 One-eyed Monster



#6 One-eyed Monster



#6 One-eyed Monster

How it works:

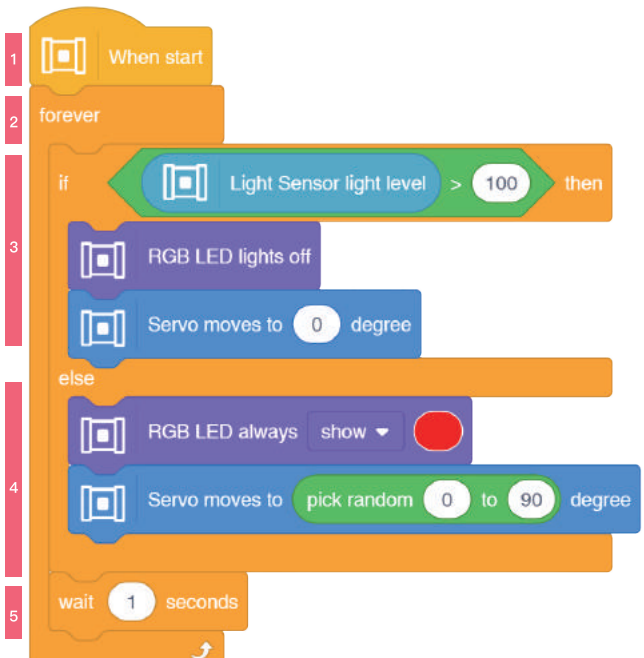
1 The code below the 'When start' starts running from top to bottom when the control board is powered on.

2 The code in the forever loop runs again and again until power off.

3 Gets light level measured by Light Sensor and compares it with the reference value '100 lux'. In this case, if the light level is larger than 100 lux, then it's daytime, otherwise it's nighttime. In daytime, the monster is sleeping, so the RGB LED is off and the Servo is still (at 0 degree).

4 When darkness comes, the monster opens its red eye and keeps moving, so the RGB LED shows red light and the Servo moves.

5 The code in the forever loop runs once every 1 second.



#7
Space Dream

Spacemen are well known to have strong body and mighty heart, but in fact, when they are kids they are just like everyone else, they also grow up with pleasure, anger, sorrow and joy. Let's make a baby spaceman that express his/her bad moods when you make noises.

Project Type
Blockly Programming (Codecraft)

Level
Medium

Time
20 minutes

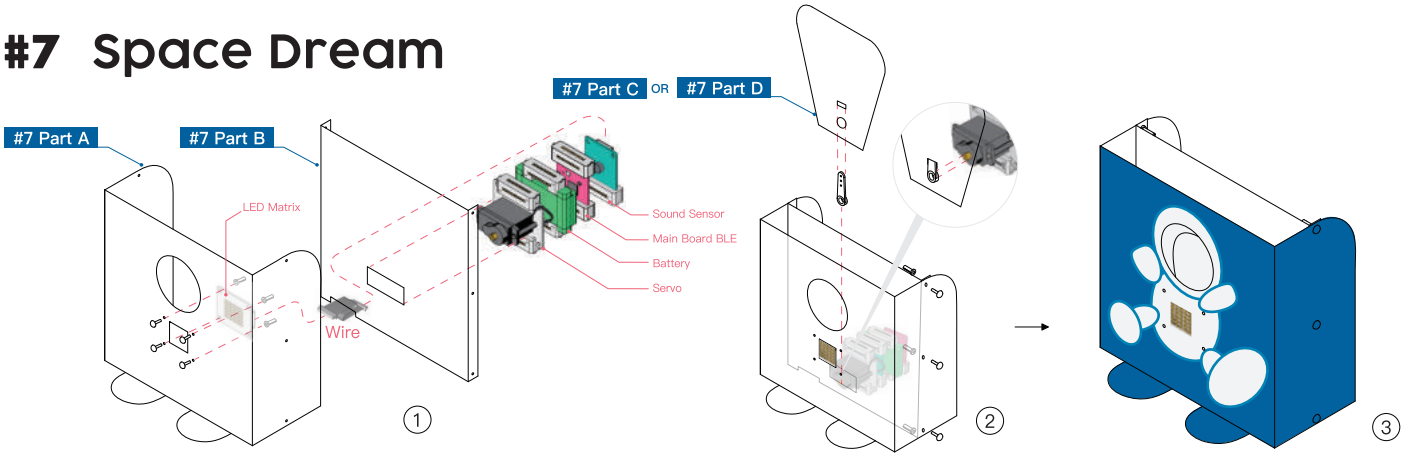
You need

- #7 Part A
- #7 Part B
- #7 Part C
- #7 Part D

&

- Main Board BLE
- Battery
- Sound Sensor
- Servo + LED Matrix
- Wire

#7 Space Dream



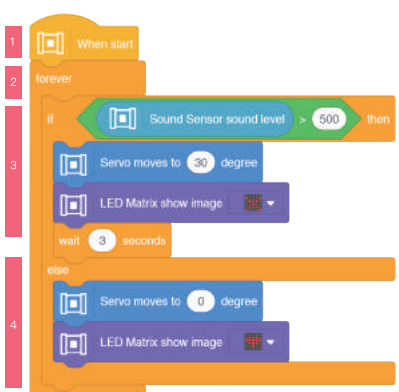
How it works:

1 The code below the 'When start' starts running from top to bottom when the control board is powered on.

2 The code in the forever loop runs again and again until power off.

3 Gets sound level measured by Sound Sensor and compares it with the reference value '500' (you can test and change this value). In this case, if the sound level is higher than 500, then the environment is loud, otherwise it's quiet. When it's loud, the Servo move to 30 degrees to show emotion no.2(probably angry) and the LED Matrix shows a cross for 3 seconds.

4 When it's quiet, the Servo stays at 0 degree to show emotion no.1(probably happy) and the LED Matrix shows a heart.



#8
The Dancing Octopus

Make a dancing octopus that waves its soft arms up and down and dances with the music. When the music gets louder, the octopus will get more excited and speed up its movement.

Project Type

Blockly Programming (Codecraft)

Level

Hard

Time

40 minutes

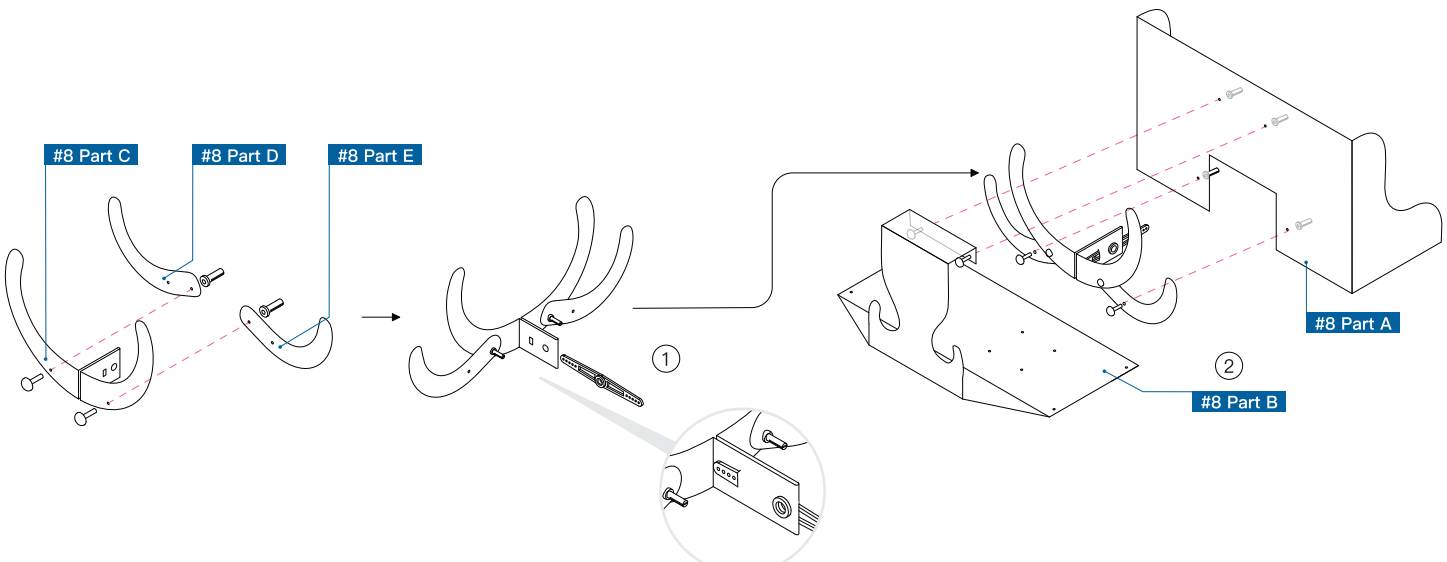
You need

- #8 Part A
- #8 Part B
- #8 Part C
- #8 Part D
- #8 Part E
- #8 Part F

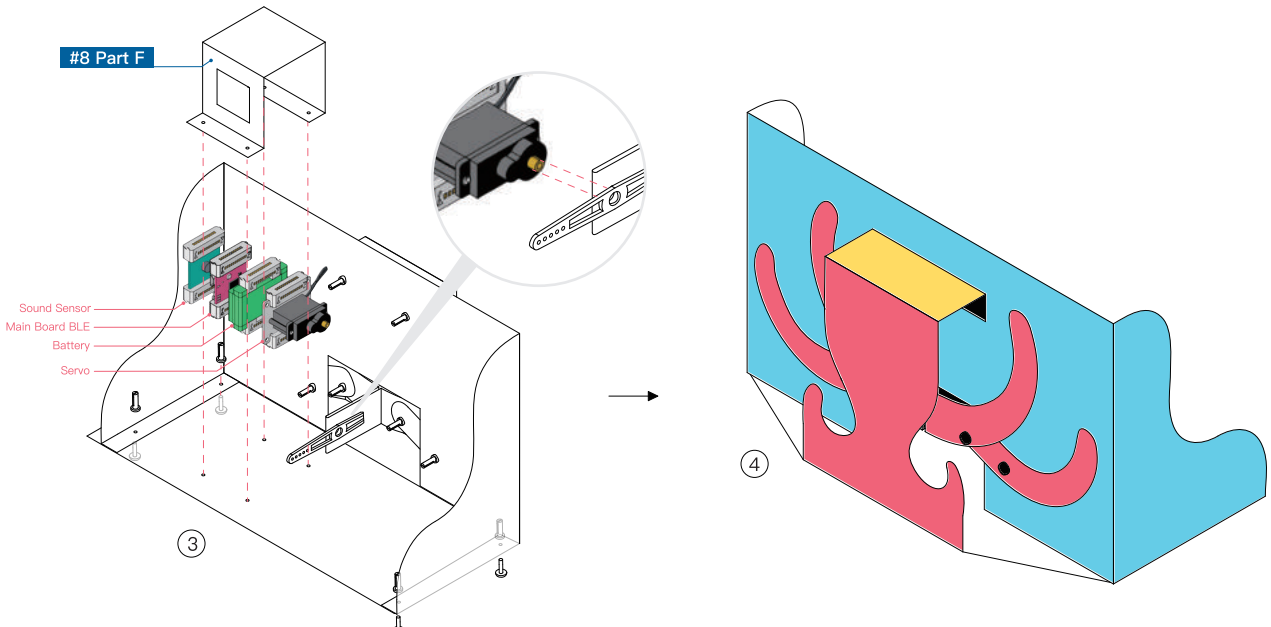
&

- Main Board BLE
- Battery
- Sound Sensor
- Servo

#8 The Dancing Octopus



#8 The Dancing Octopus



#8 The Dancing Octopus

How it works:

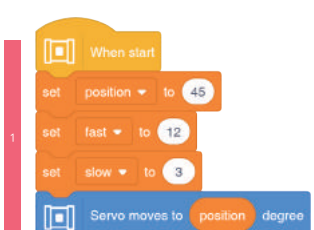
1 The initial states

In the initial state, we set the speed for different levels, and move the servo to 45 degrees.

Hereby we create **3** variables:

position – represents the current angular position of servo.

fast/slow – represents the speed of the servo rotation, which controls the movement of the octopus's arms.



2 The code below the 'When timer>3' runs in 3 seconds after power on.

3 If the sound level is below 300, then change the current position by 3 degrees(move slowly) , otherwise if the sound level is higher than 300, then change the current position by 12 degrees(move fast).

4 As the arms of octopus move in a specific range, we need to limit the range of rotation between 45° to 90° in case the servo breaks the arms.

Meanwhile, to make the arms wave up and down, we can reverse the direction of shaft rotation by changing the values of 'fast/slow' to negative. If the 'position' is above 90° or blow 45°, then it starts rotating in opposite direction.

5 Change the 'position' every 0.1 second.

#9
Wheel of Fortune

One of the most fun part in playing a tabletop game is the dice rolling, you'll never know if you are having bad luck or good luck. Let's make a cool electronic wheel of fortune, and try it in your favorite tabletop games with your friends.

Project Type

Blockly Programming (Codecraft)

Level

Hard

Time

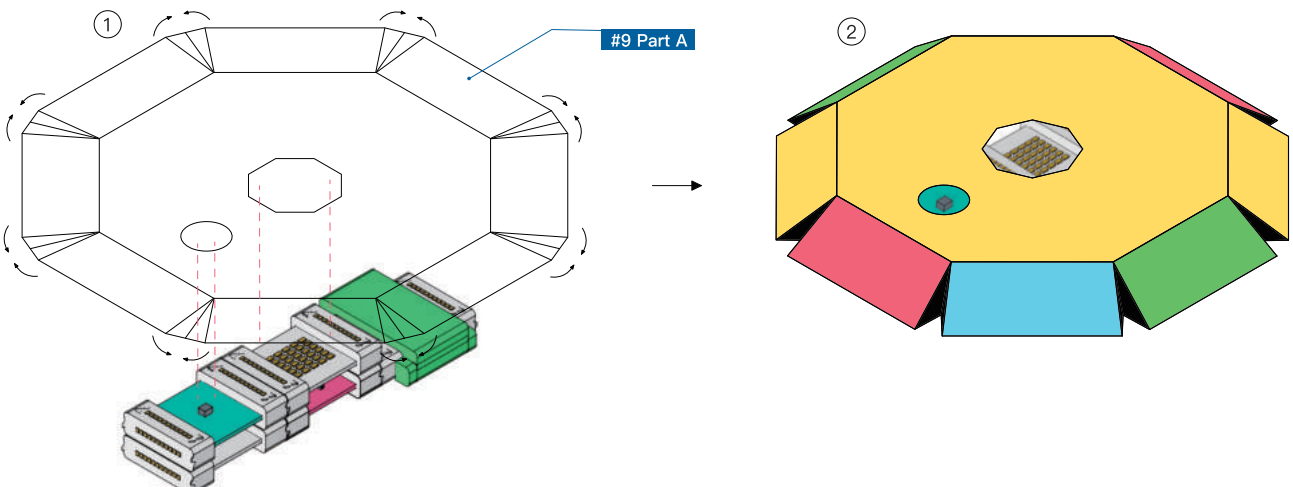
40 minutes

You need

#9 Part A &

- Main Board BLE
- Battery
- Light Sensor
- LED Matrix + Buzzer

#9 Wheel of Fortune



#9 Wheel of Fortune

How it works:

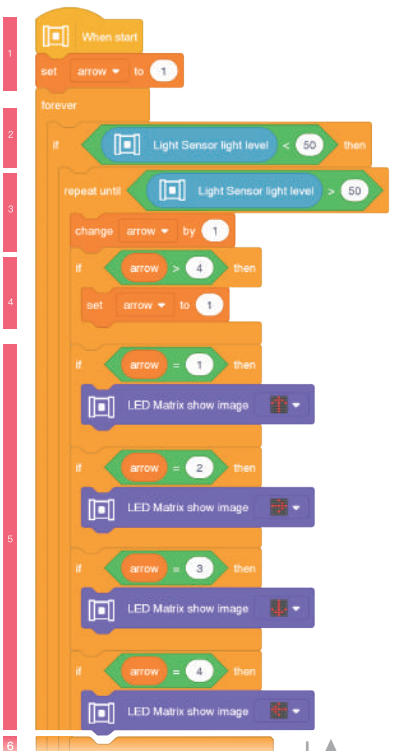
1 The initial state
In the initial state, we create 1 variable:
arrow – represents the current direction of the arrow. 1 is up, 2 is right, 3 is down, 4 is left.

2 Spinning
When you cover the light sensor with your hand, the light sensor detects a very low light intensity, which is lower than '50 lux' (you can change this value if the LED Matrix acts weird) in this case.

3 The wheel starts to 'spin' clockwise until you take your hand off.
Change the value of 'arrow' to make the wheel 'spin'.

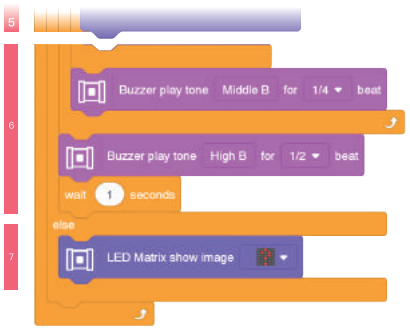
4 When 'arrow' is above 4, set it back to 1 so it can start over.

5 The value of 'arrow' represents the direction.
1 = > Up
2 => Right
3 => Down
4 => Left



6 Add some sound effects 'Middle B' when the wheel 'spins'. The wheel stops 'spinning' along with a sound effect 'High B', and the LED Matrix shows the last arrow for 1 second.

7 When the wheel is not used, the LED Matrix shows a 'question' mark to create a feeling of unknown.



#10
Game Boy

The Game Boy can be a very good choice when you feel bored and need something to play with. In this project, you'll make a Game Boy and a simple game about fighter jet. Enemies will be randomly showing up on the screen, and your mission is to pilot the figher plane to shoot as many opponents as you can.

Project Type

Blockly Programming (Codecraft)

Level

Challenging

Time

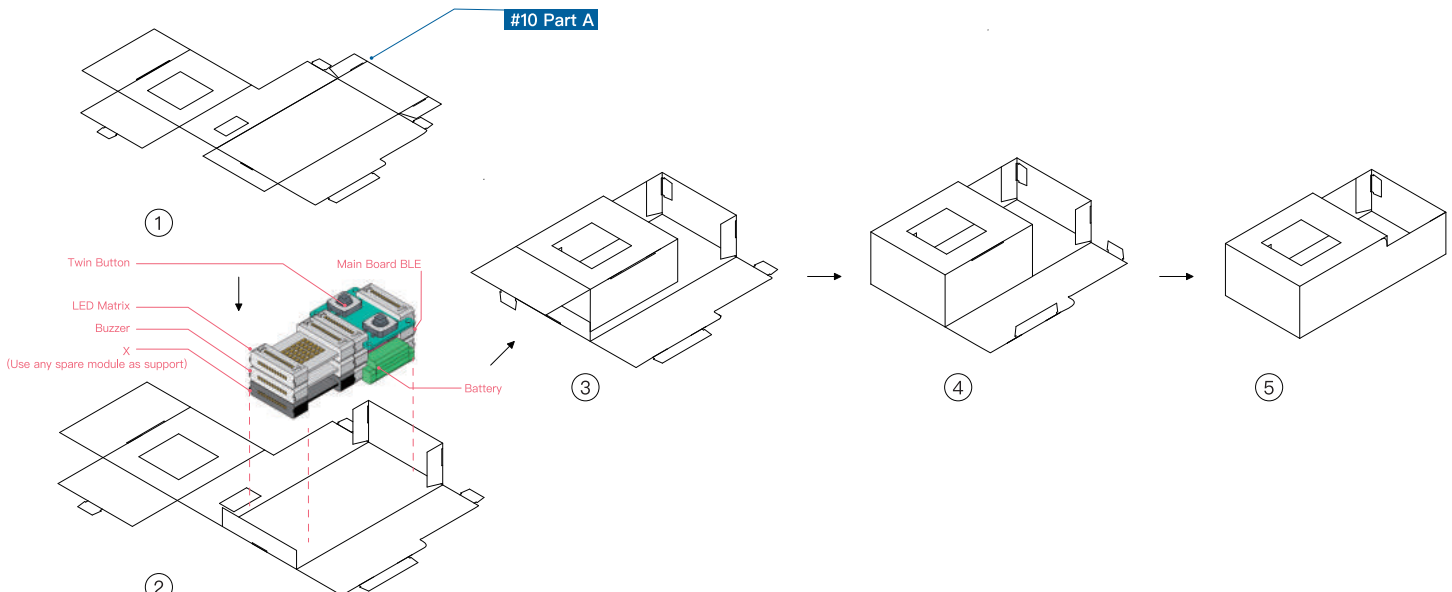
40 minutes

You need

#10 Part A &

- Main Board BLE
- Battery
- Twin Button
- LED Matrix + Buzzer

#10 Game Boy


#10 Part A
1
Twin Button
LED Matrix
Buzzer
X
(Use any spare module as support)
Main Board BLE
Battery
2
3
4
5

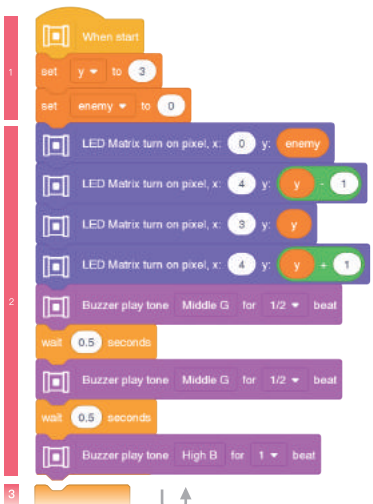
#10 Game Boy

1 The initial states

Variable 'y' is the position of the fighter plane and likewise the variable 'enemy' represents the target to be destroyed.

2 The opening scene

The fighter and enemy show up on the screen at the beginning, and the game is on after the buzz.



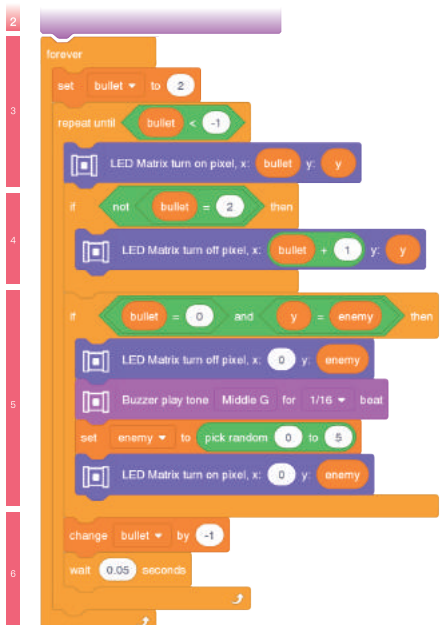
3 Game on

Bullets show up in front of the fighter plane once it's 'launched'.

4 Eliminate the 'track' of bullet.

5 If the bullet 'hits' the enemy, then the enemy will be 'destroyed' and a new enemy will appear at a random position.

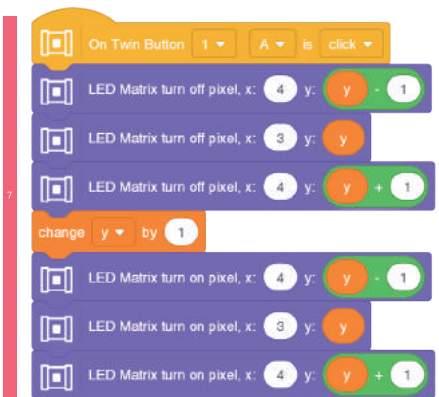
6 The bullet flies forward by changing its position. The 'wait time' sets the bullet's flying speed, a smaller number means a faster speed.



#10 Game Boy

7 Control – move to left

When button 'A' is clicked, move the fighter plane to its left by changing its position by '1'.



8 Control – move to right

When button 'B' is clicked, move the fighter plane to its right by changing its position by '–1'.