

Microbit Robotics Beginner Level 2

Lesson 1

Running box

Presented by Advanced Superlogic Team

Today's Topic

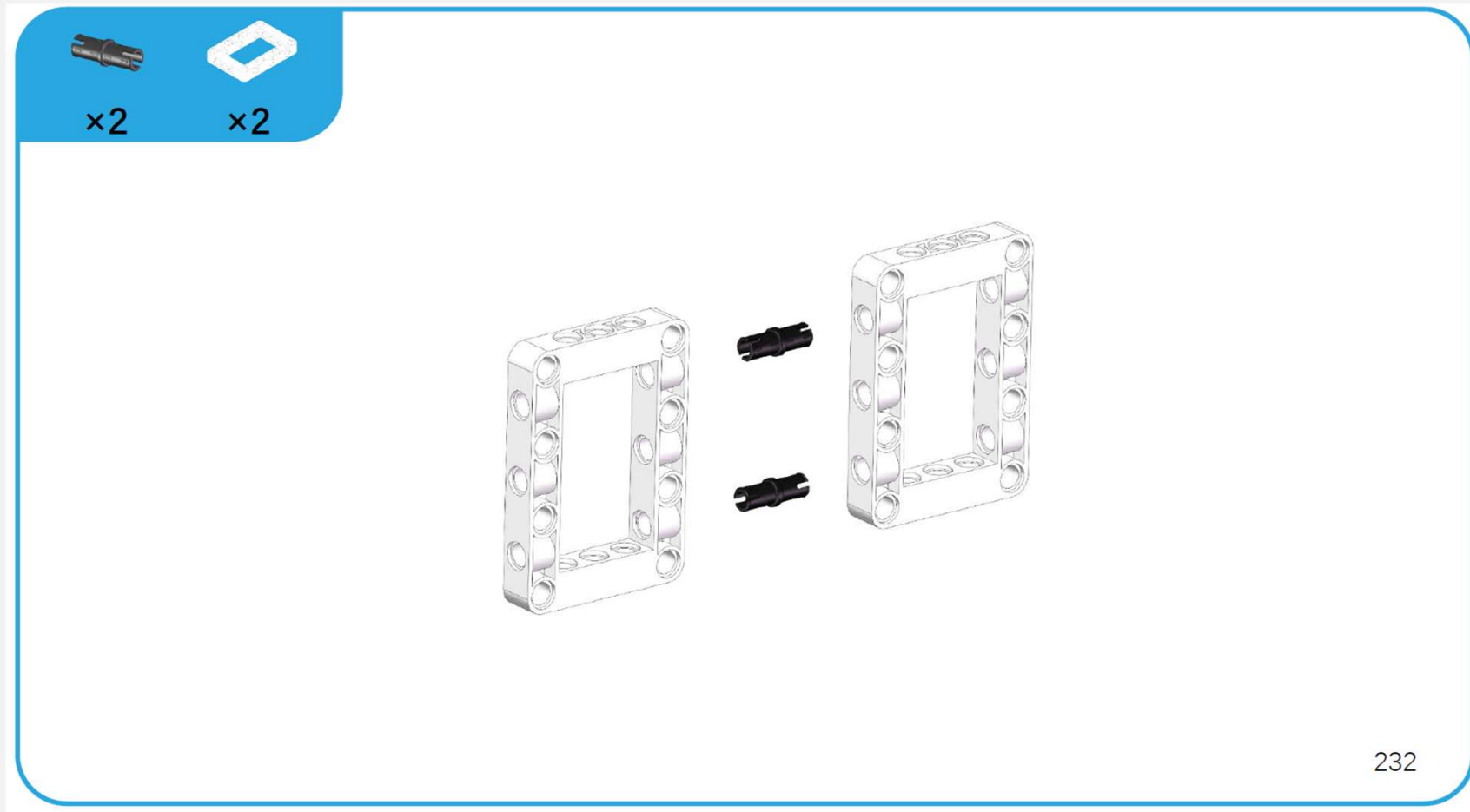
1. **Build a Running Box with World of Modules**
2. **Servo programming**
3. **Functions programming**
4. **Running Box programming**

Learning Outcome

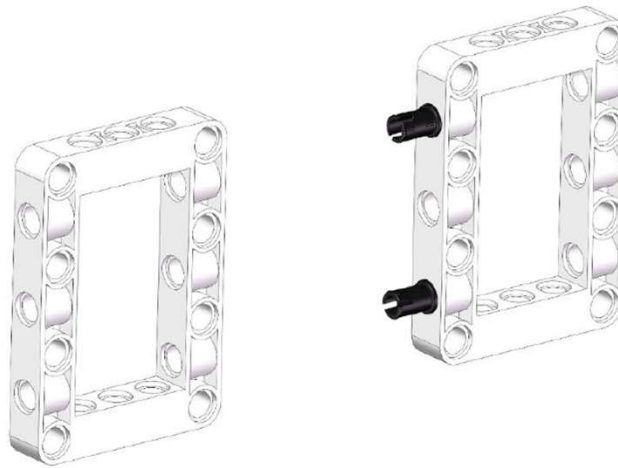
- 1. Able to build Running Box with instruction manual**
- 2. Understand Servo programming work**
- 3. Understand Functions programming work**
- 4. Able to program Running Box**

Let's build a Running box

Step 1

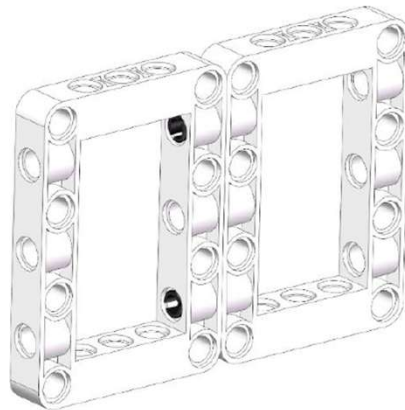


Step 2



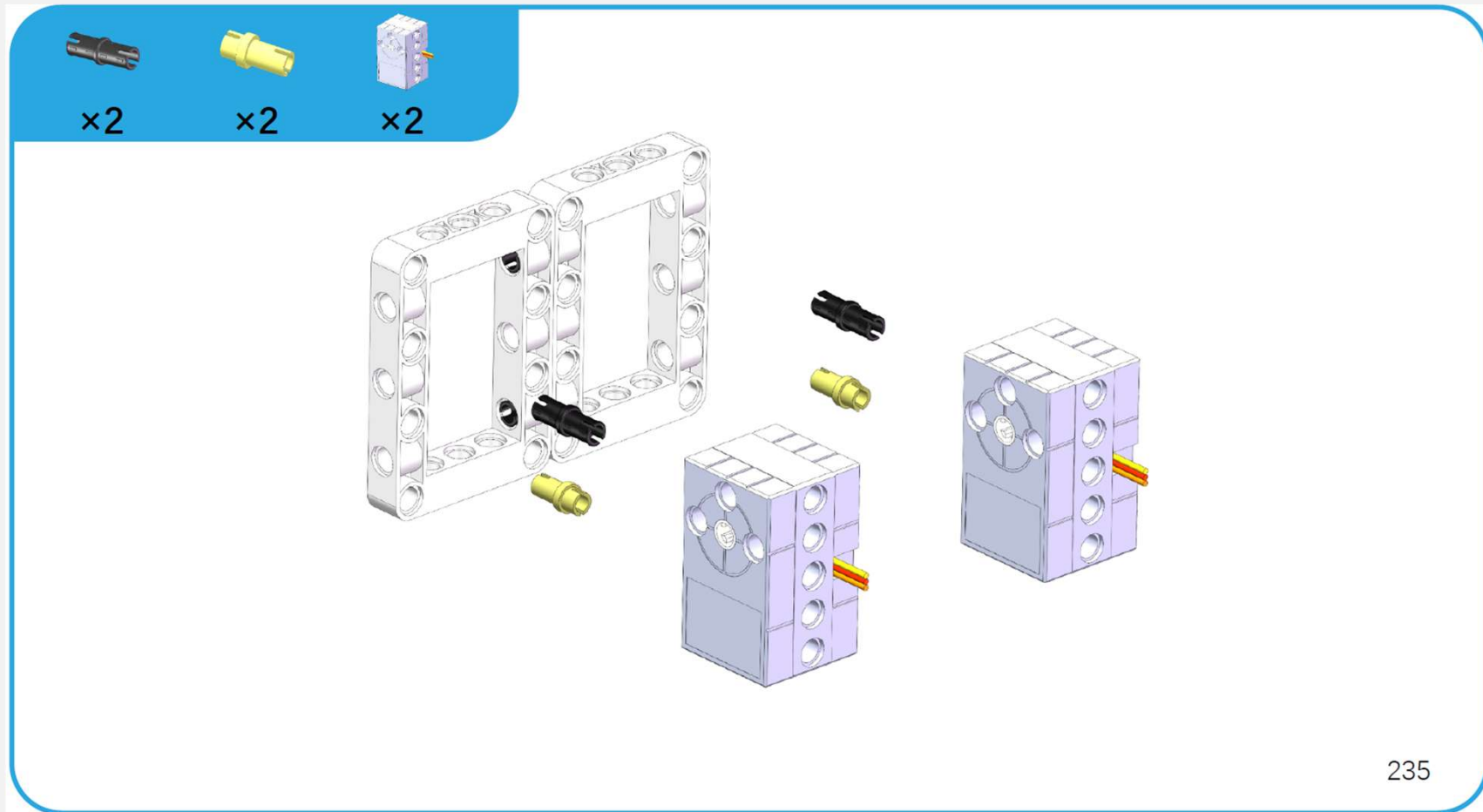
233

Step 3

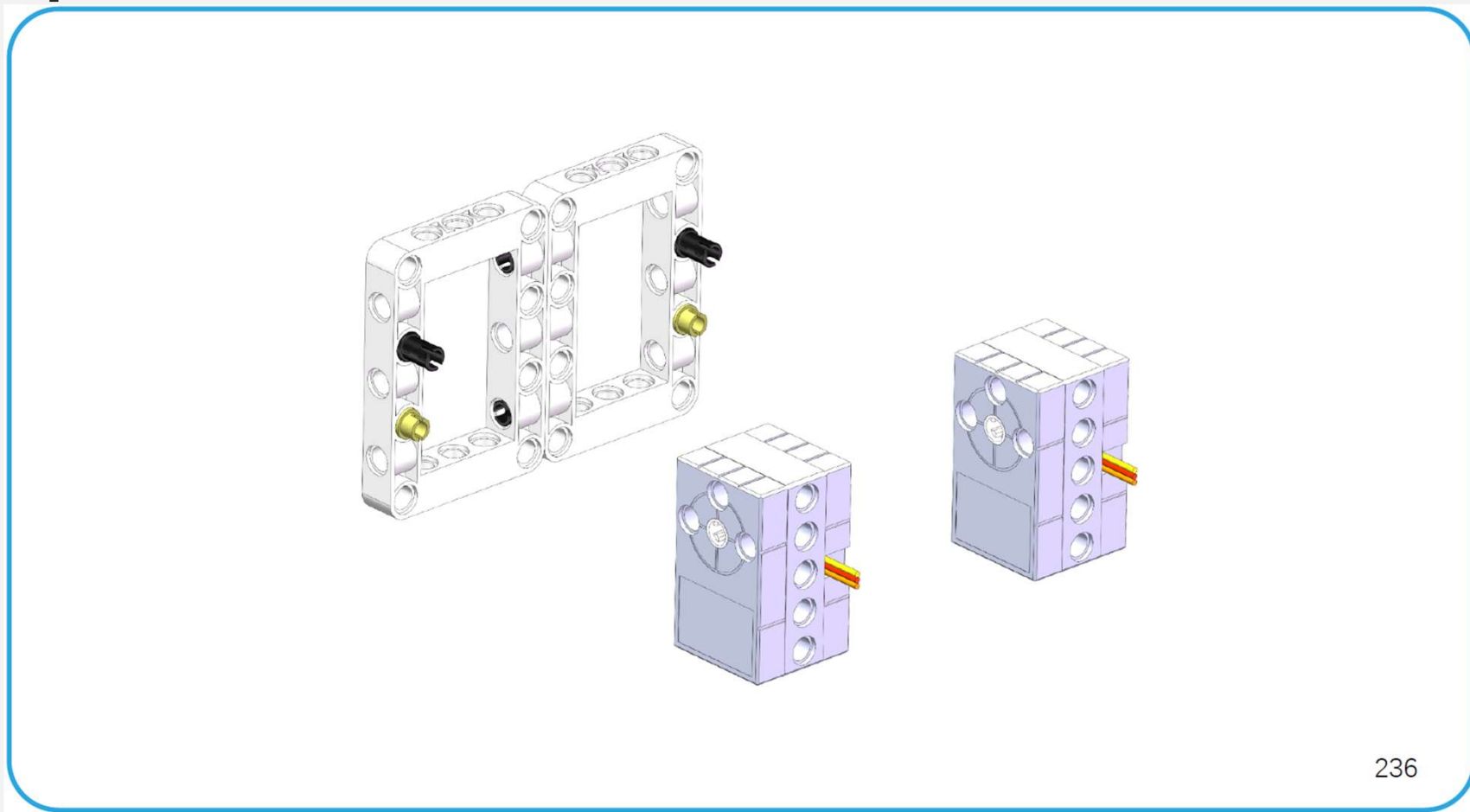


234

Step 4

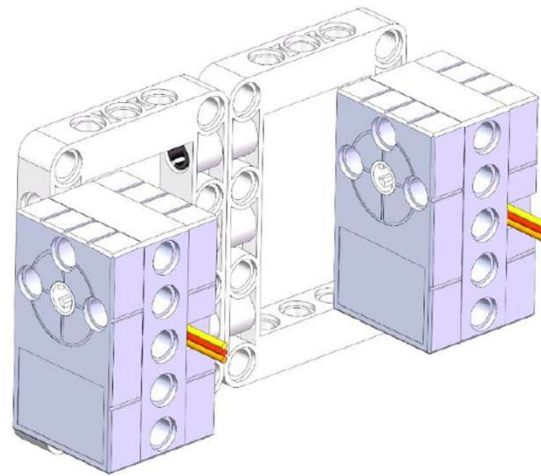


Step 5



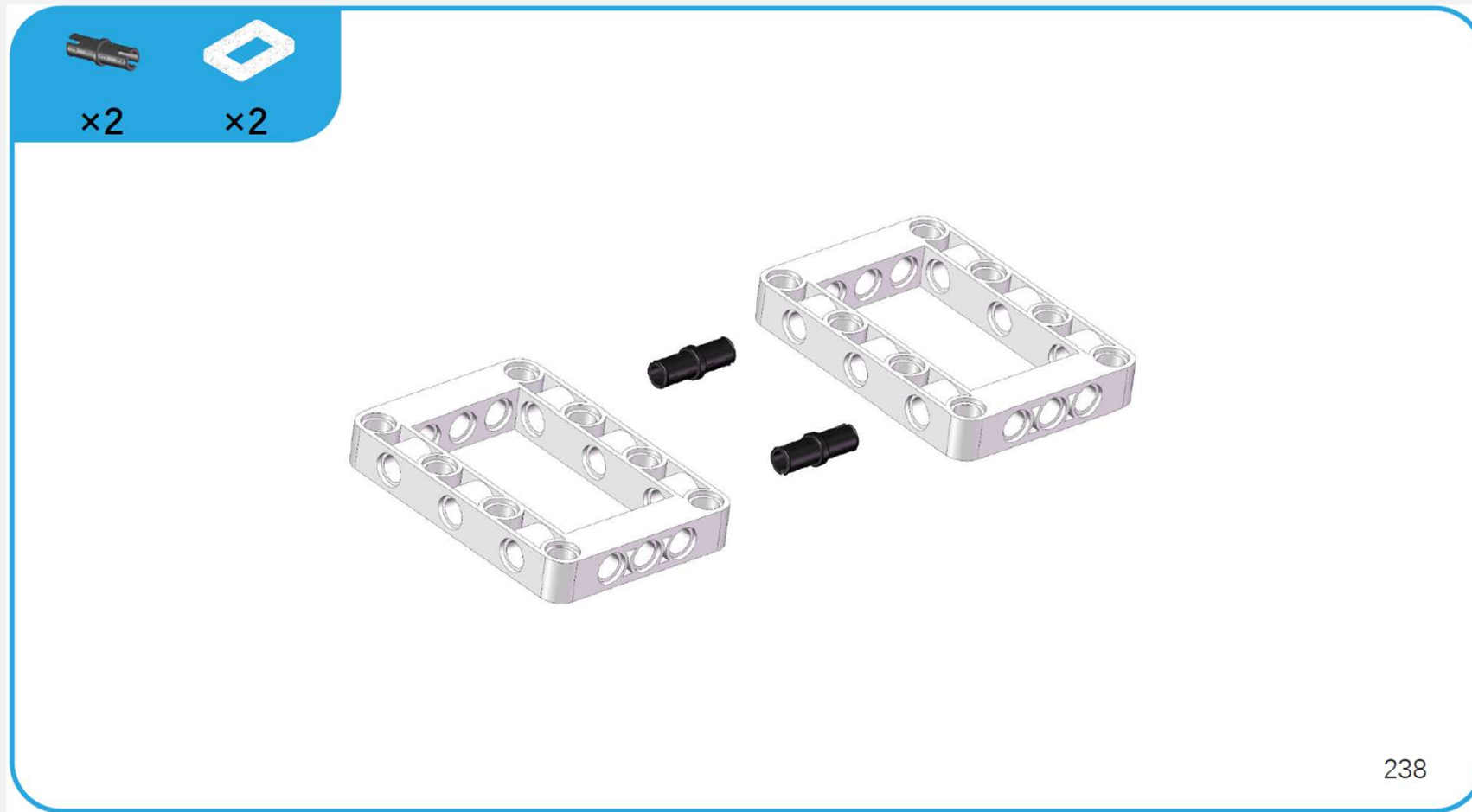
236

Step 6



237

Step 7



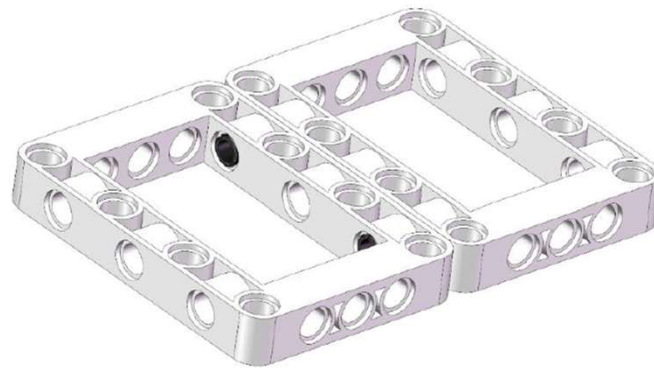
238

Step 8



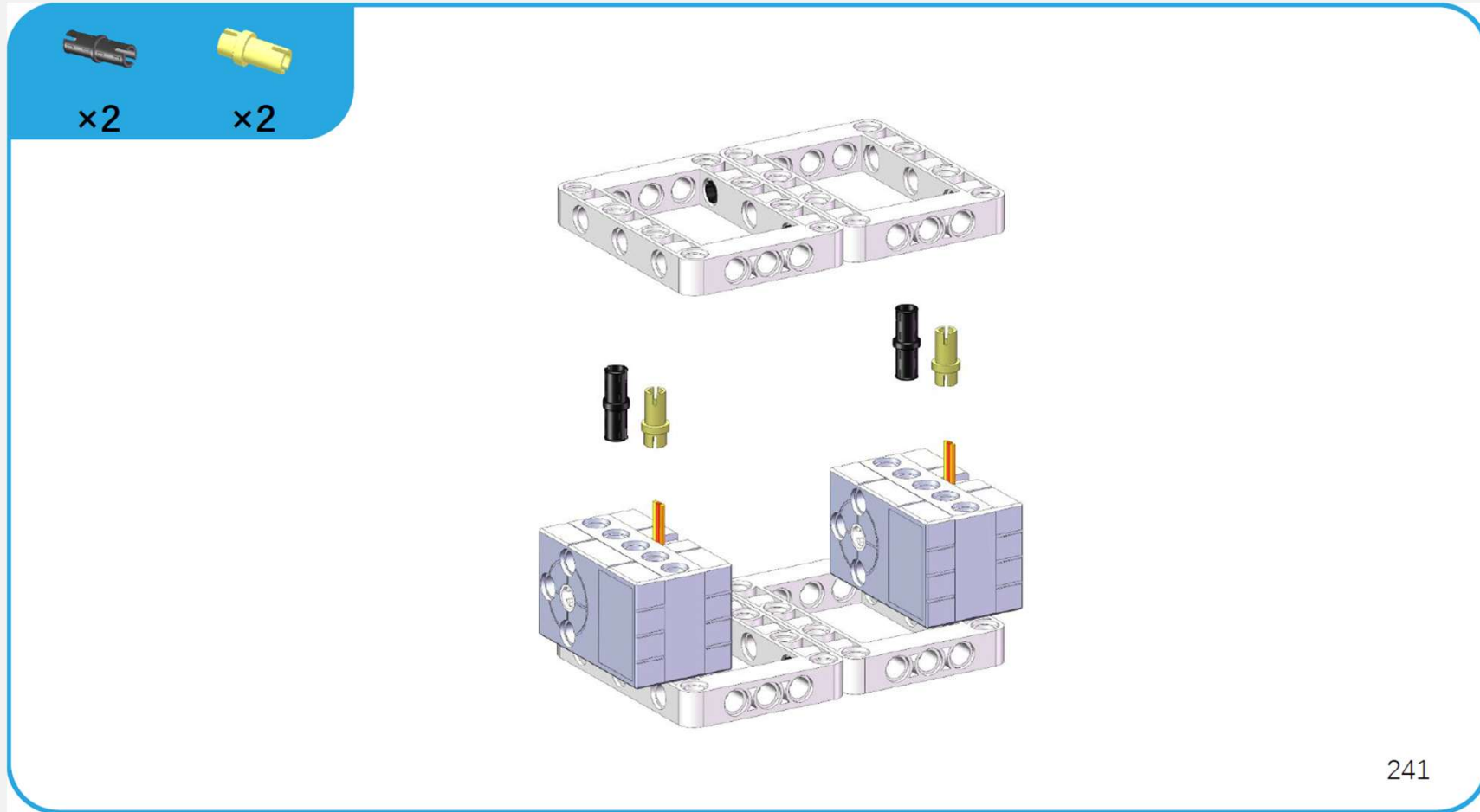
239

Step 9

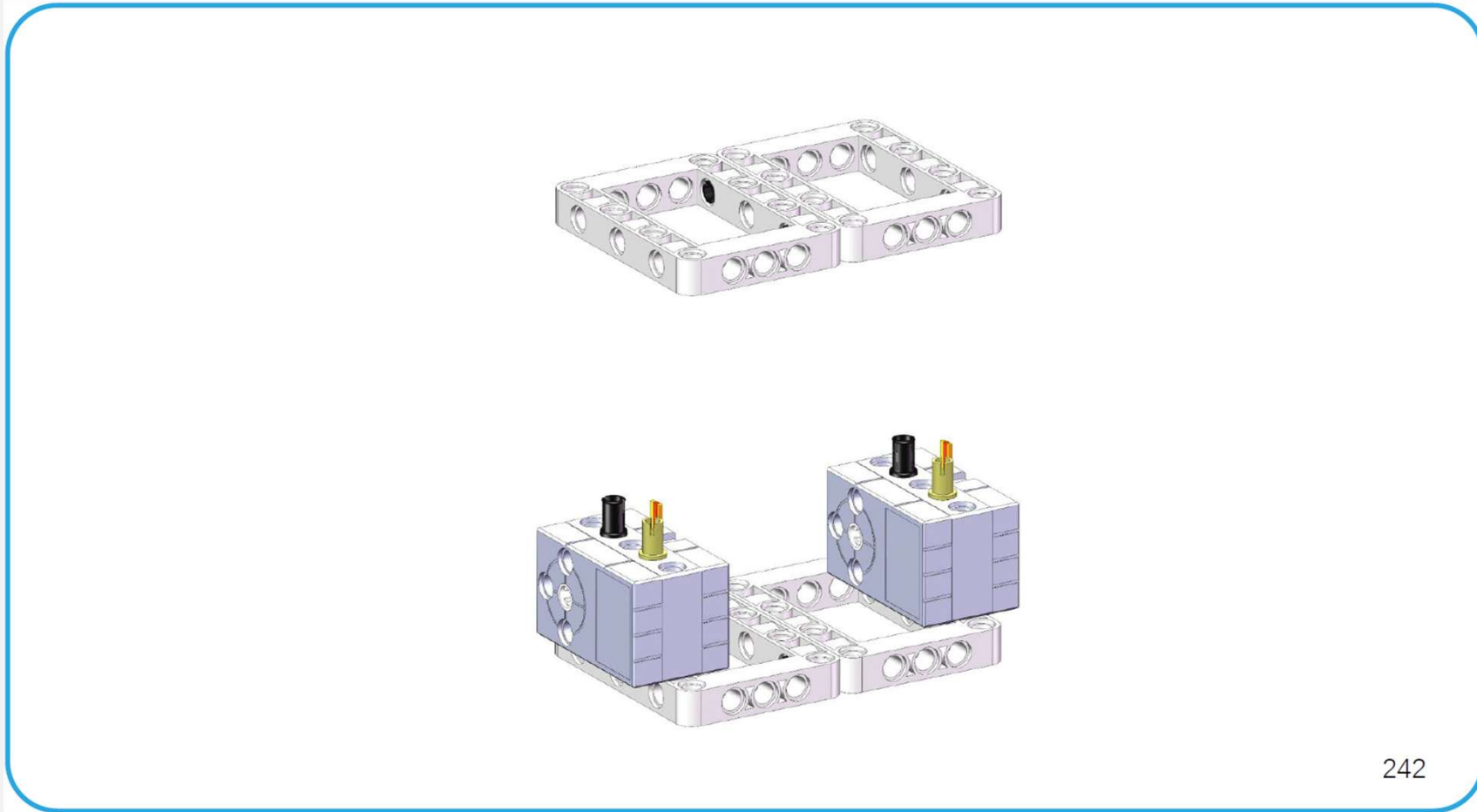


240

Step 10

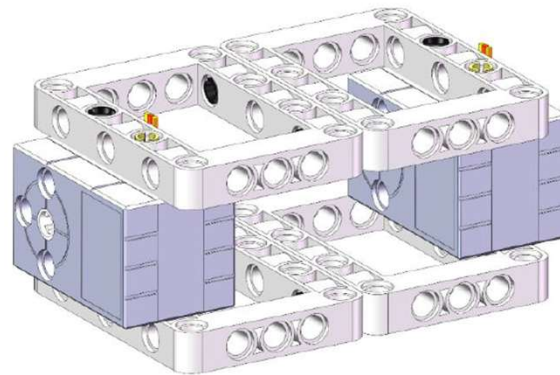


Step 11



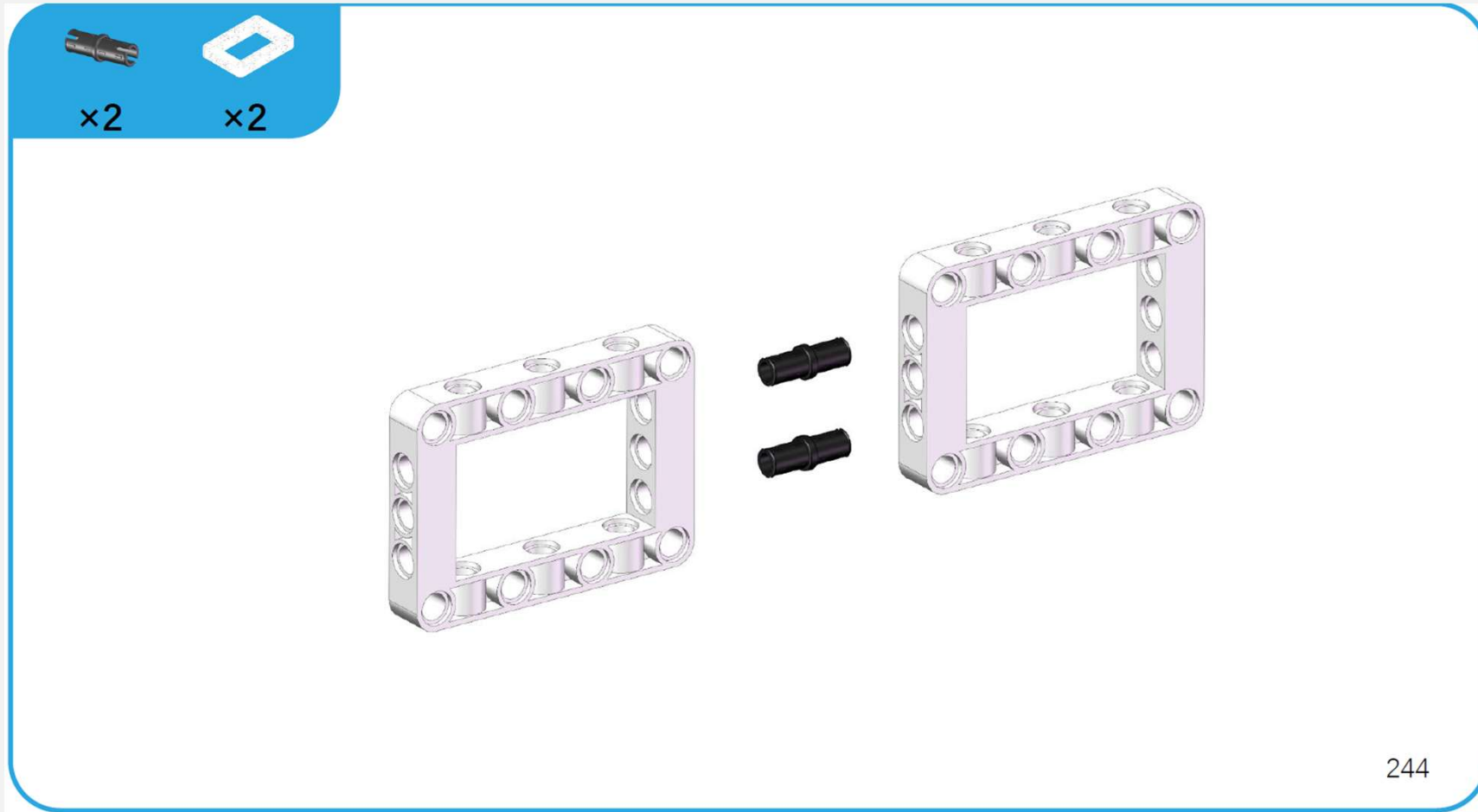
242

Step 12



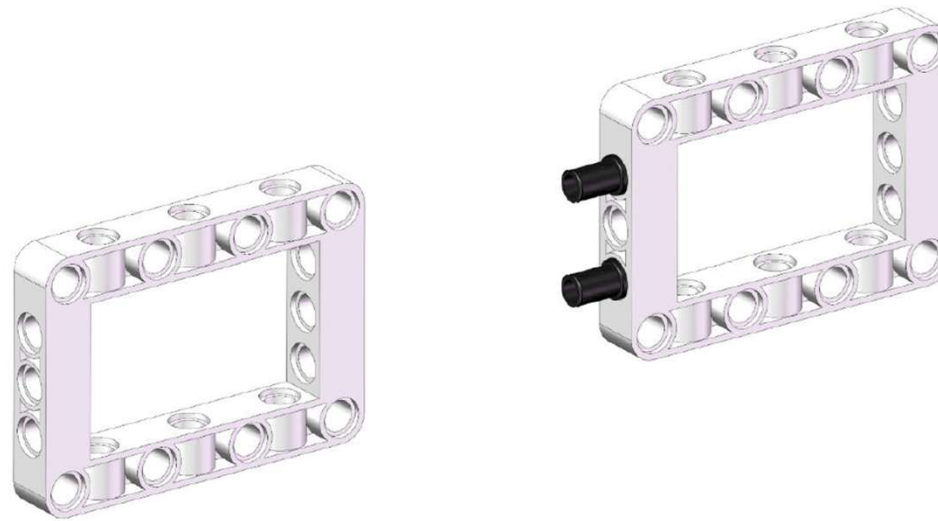
243

Step 13



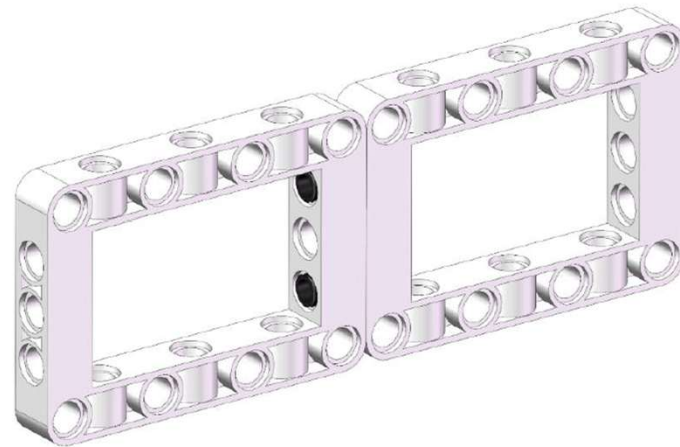
244

Step 14



245

Step 15

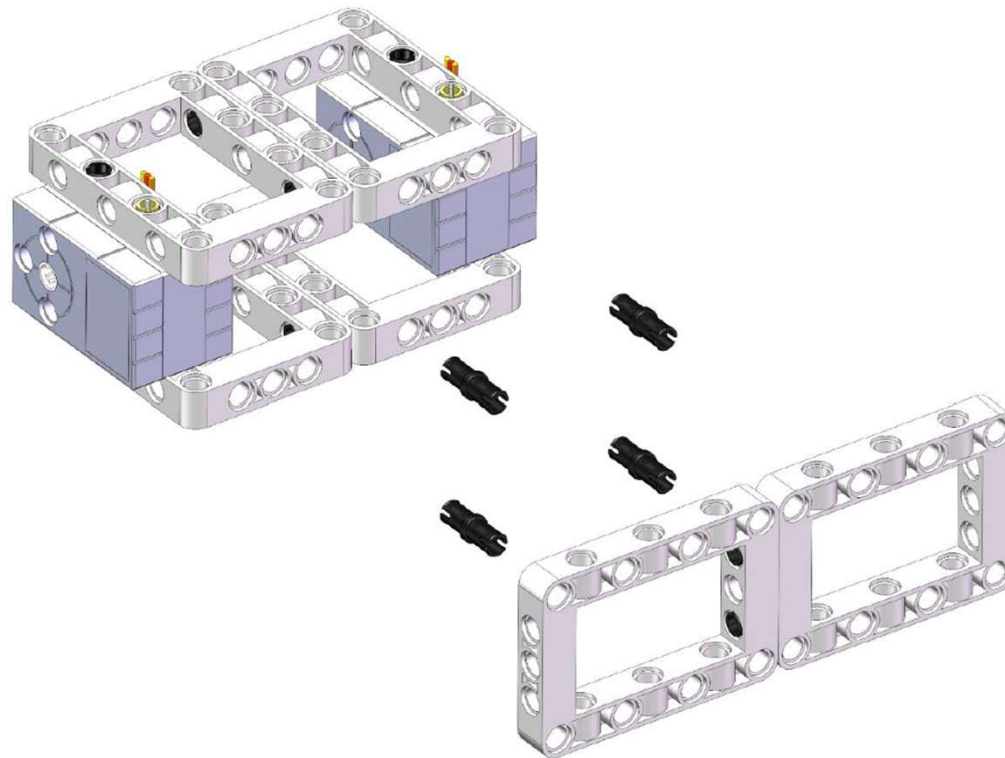


246

Step 16

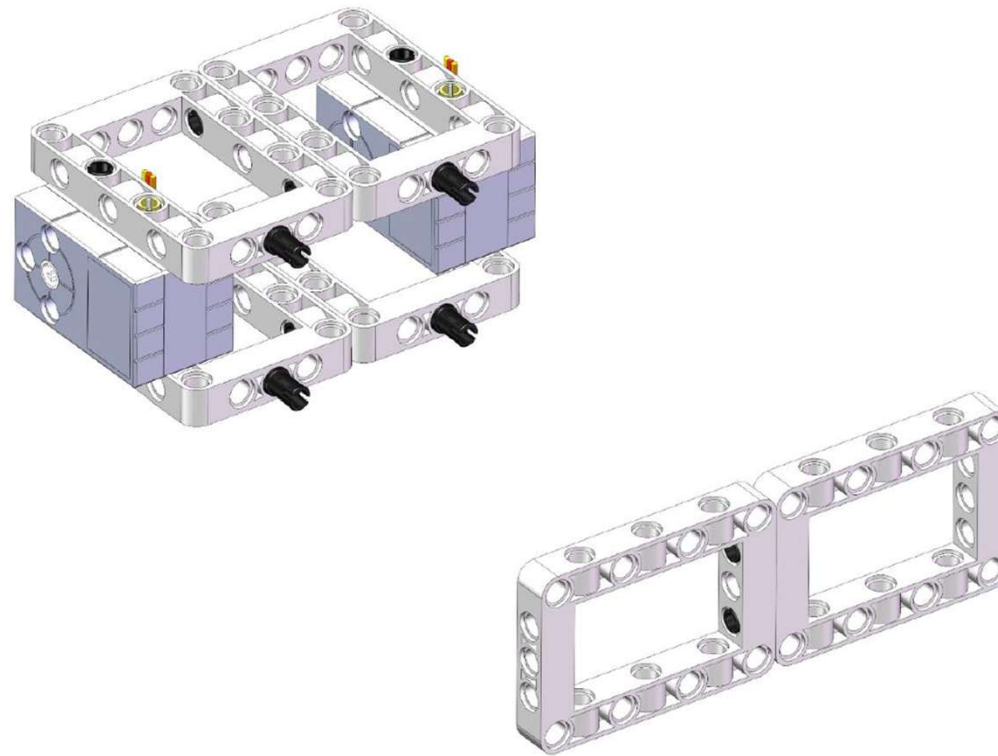


x4



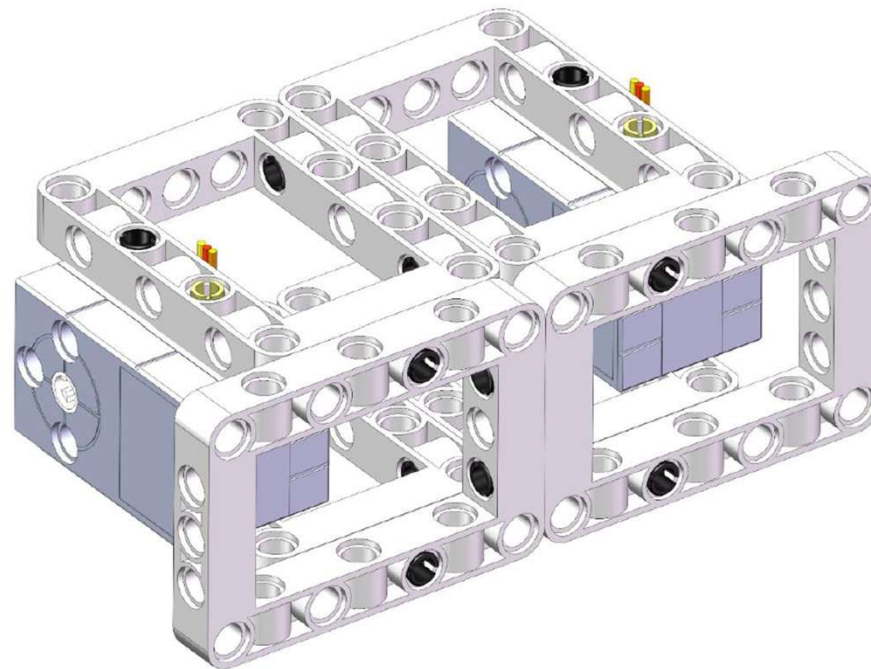
247

Step 17



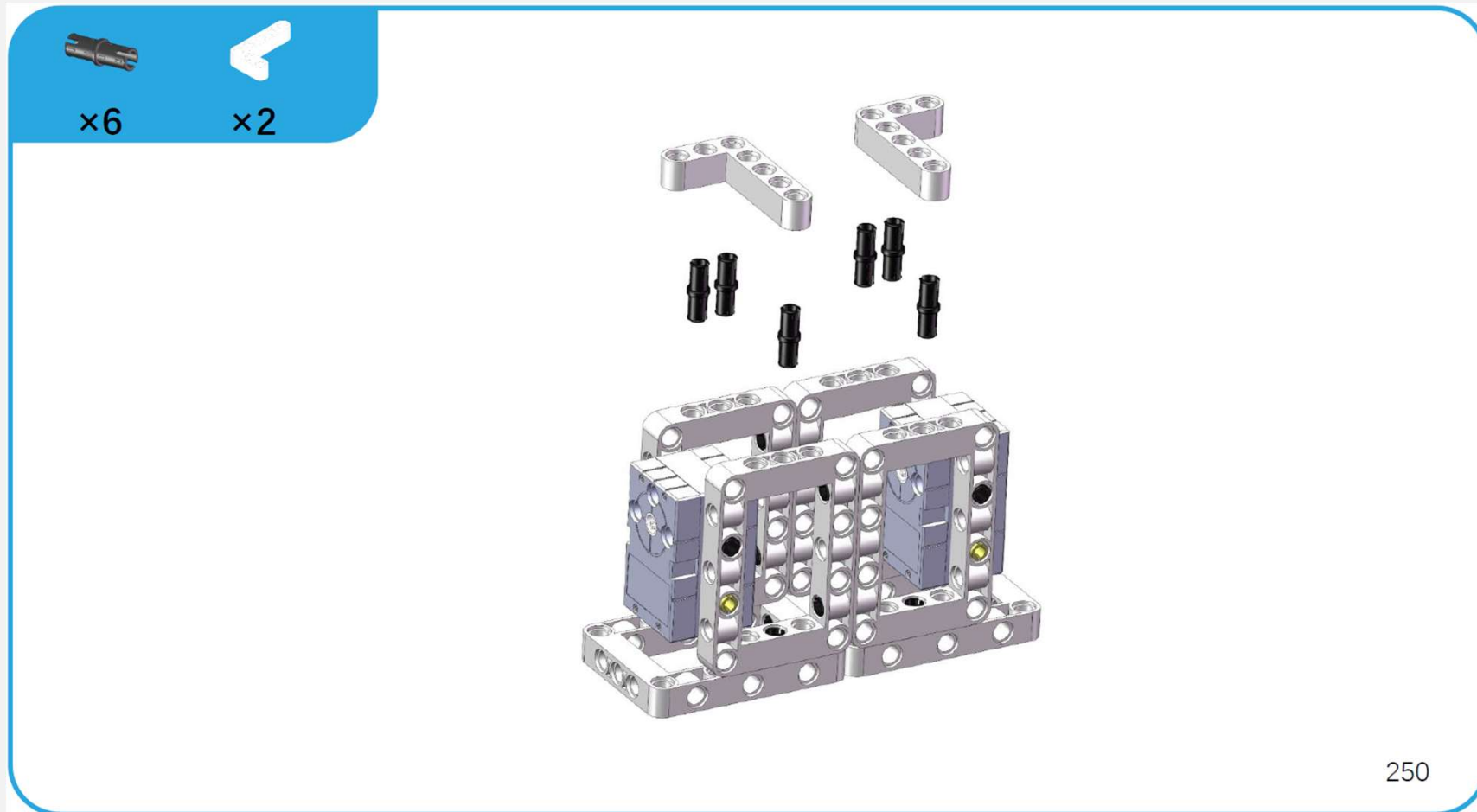
248

Step 18

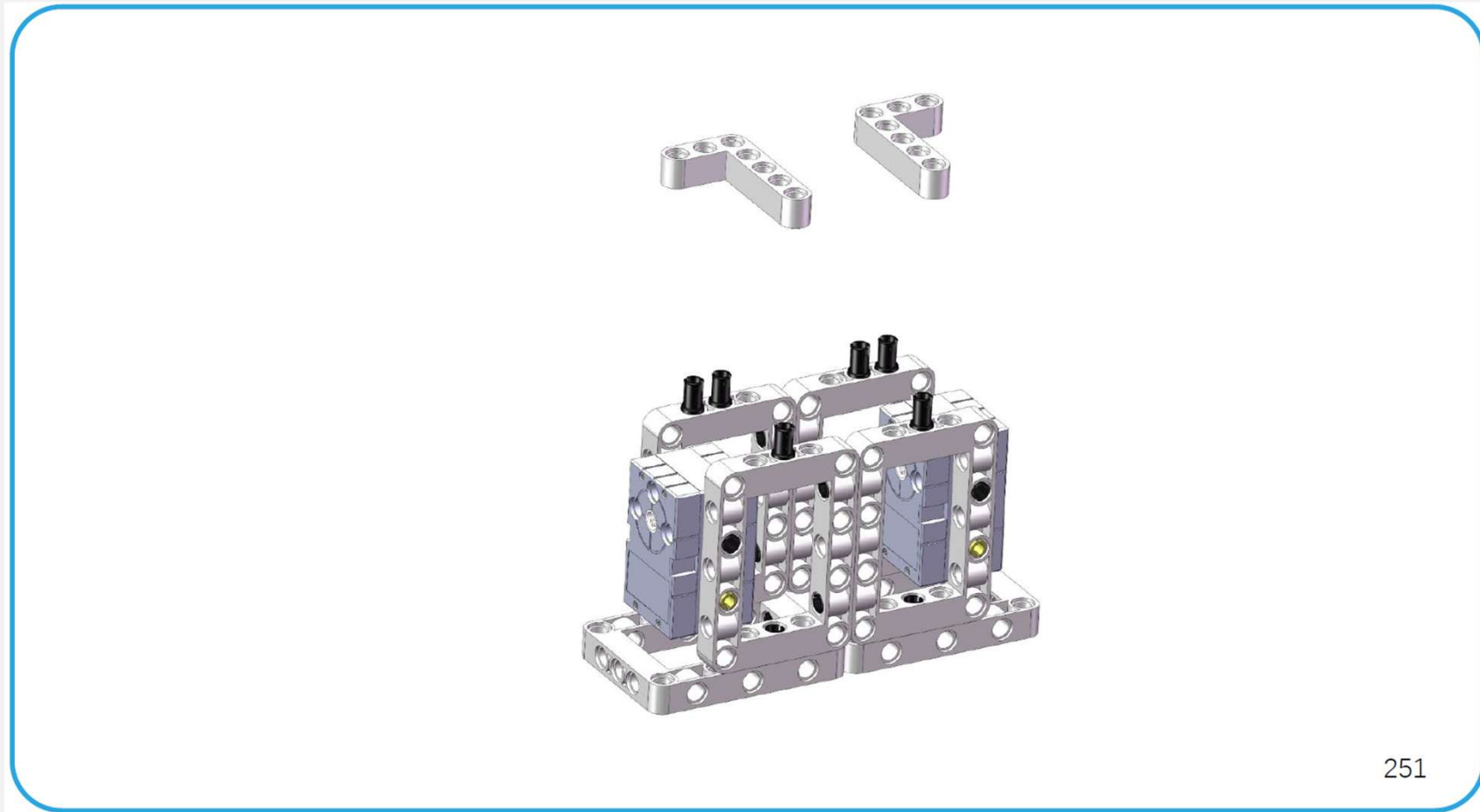


249

Step 19

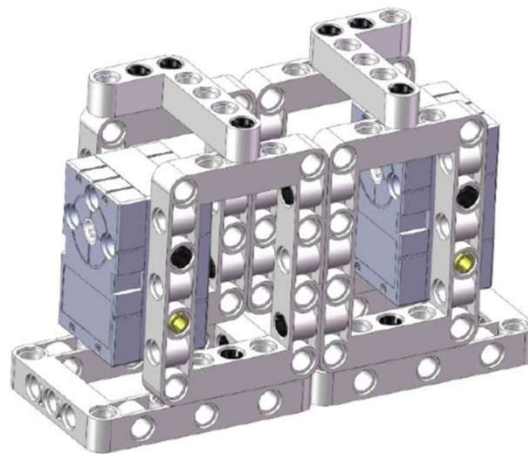


Step 20



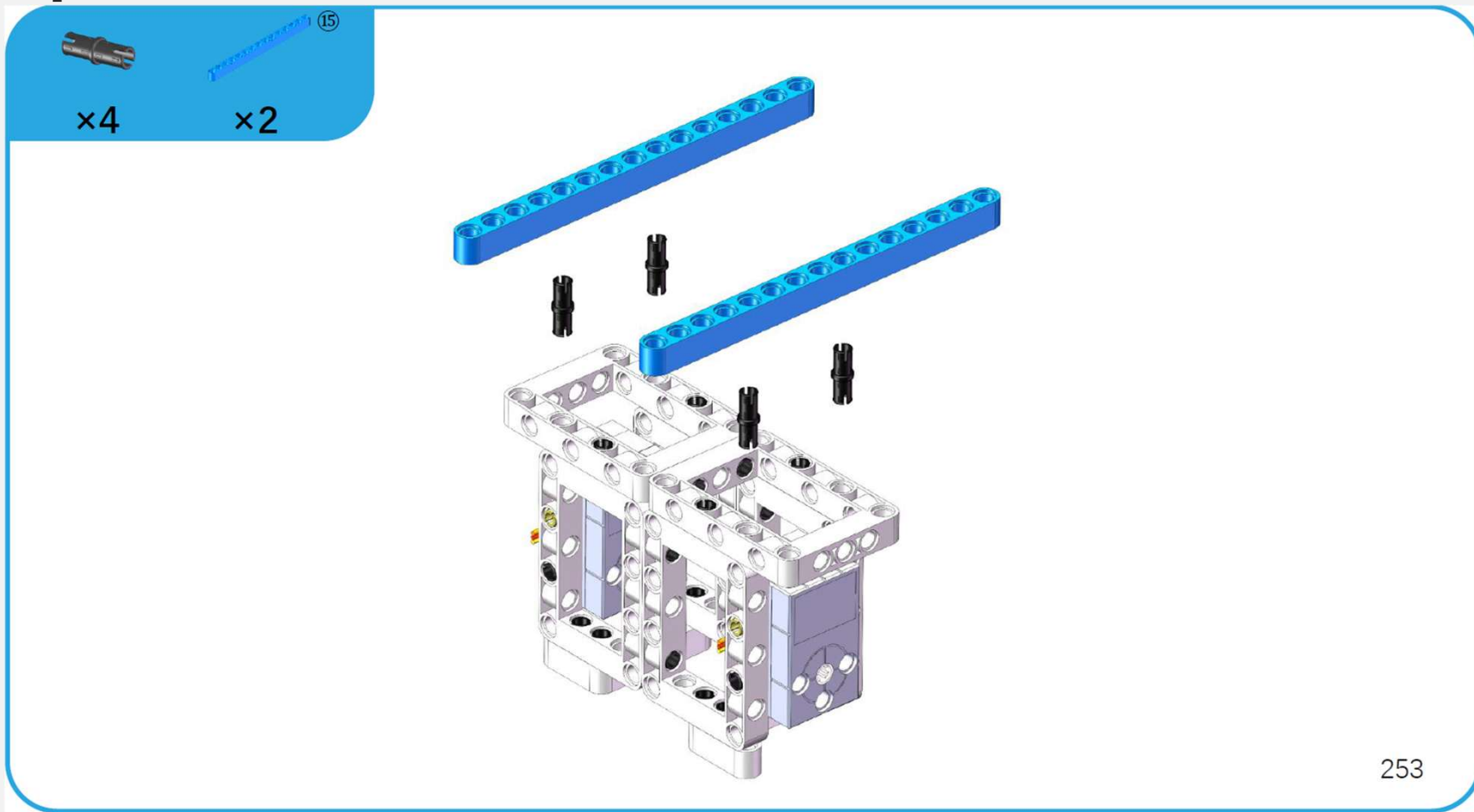
251

Step 21

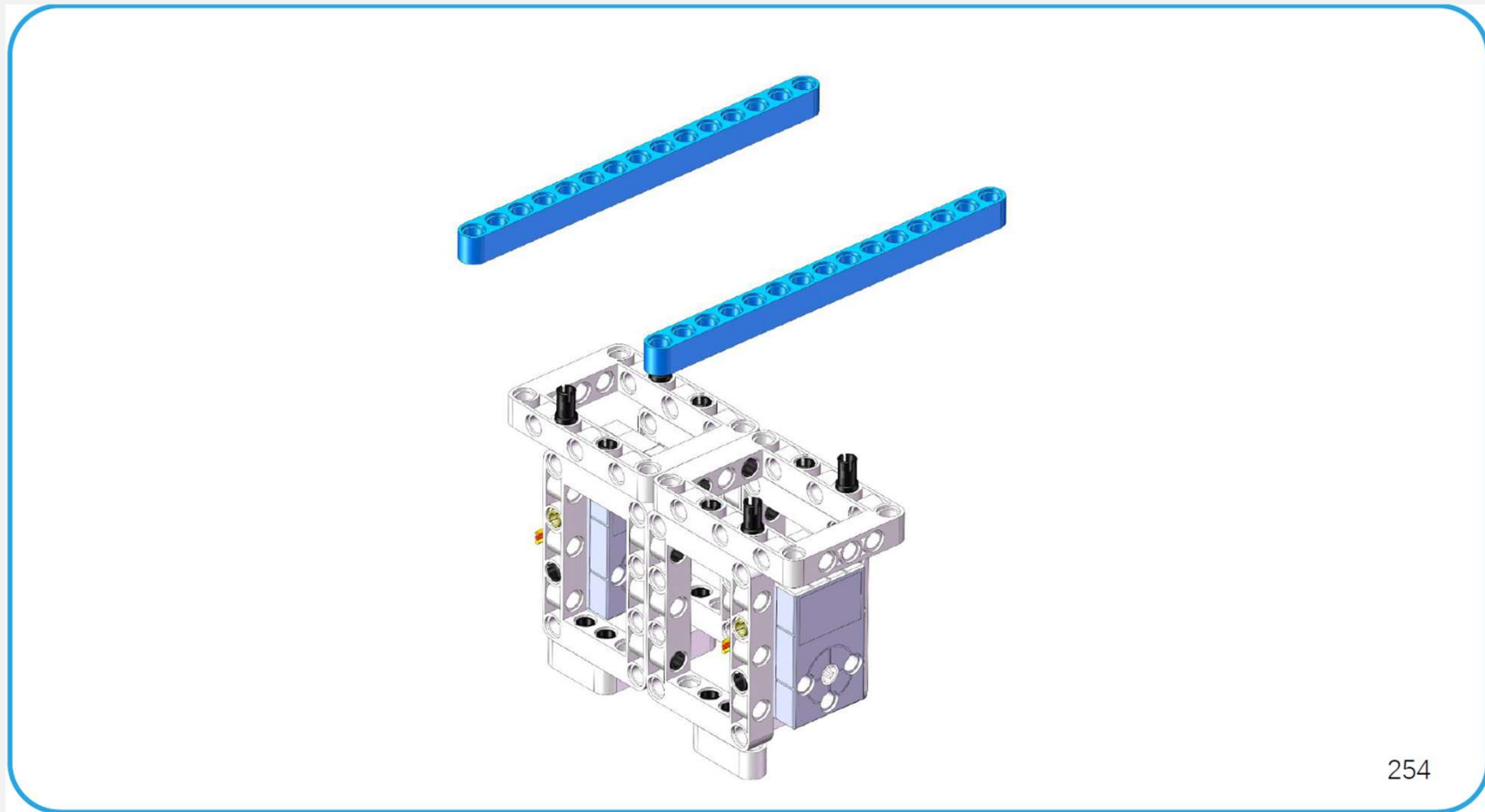


252

Step 22

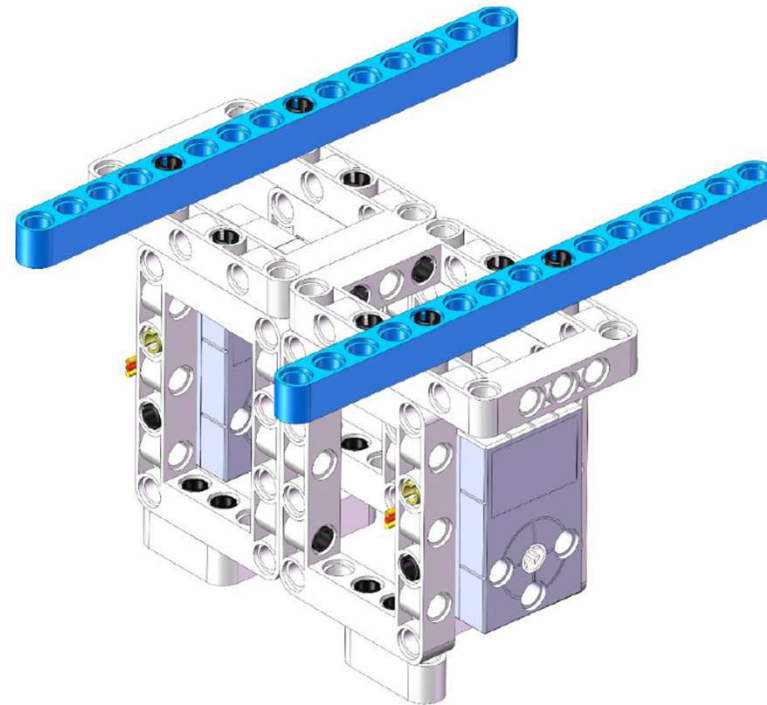


Step 23



254

Step 24



255

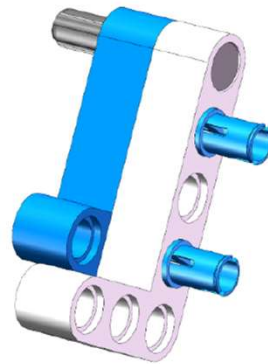
Step 25

A blue header bar contains four icons representing the parts for this step: a blue L-shaped Technic brick (1x3 with hole), a white L-shaped Technic brick (1x3 with hole), a blue Technic pin, and a brown Technic axle with a small circle and the number 3 next to it. Below each icon is a quantity: 'x1', 'x1', 'x2', and 'x1' respectively.

The main assembly area shows a 3D perspective view of the parts. On the left is a blue L-shaped Technic brick (1x3 with hole). In the center is a white L-shaped Technic brick (1x3 with hole). On the right are two blue Technic pins and one brown Technic axle (3).

256

Step 26



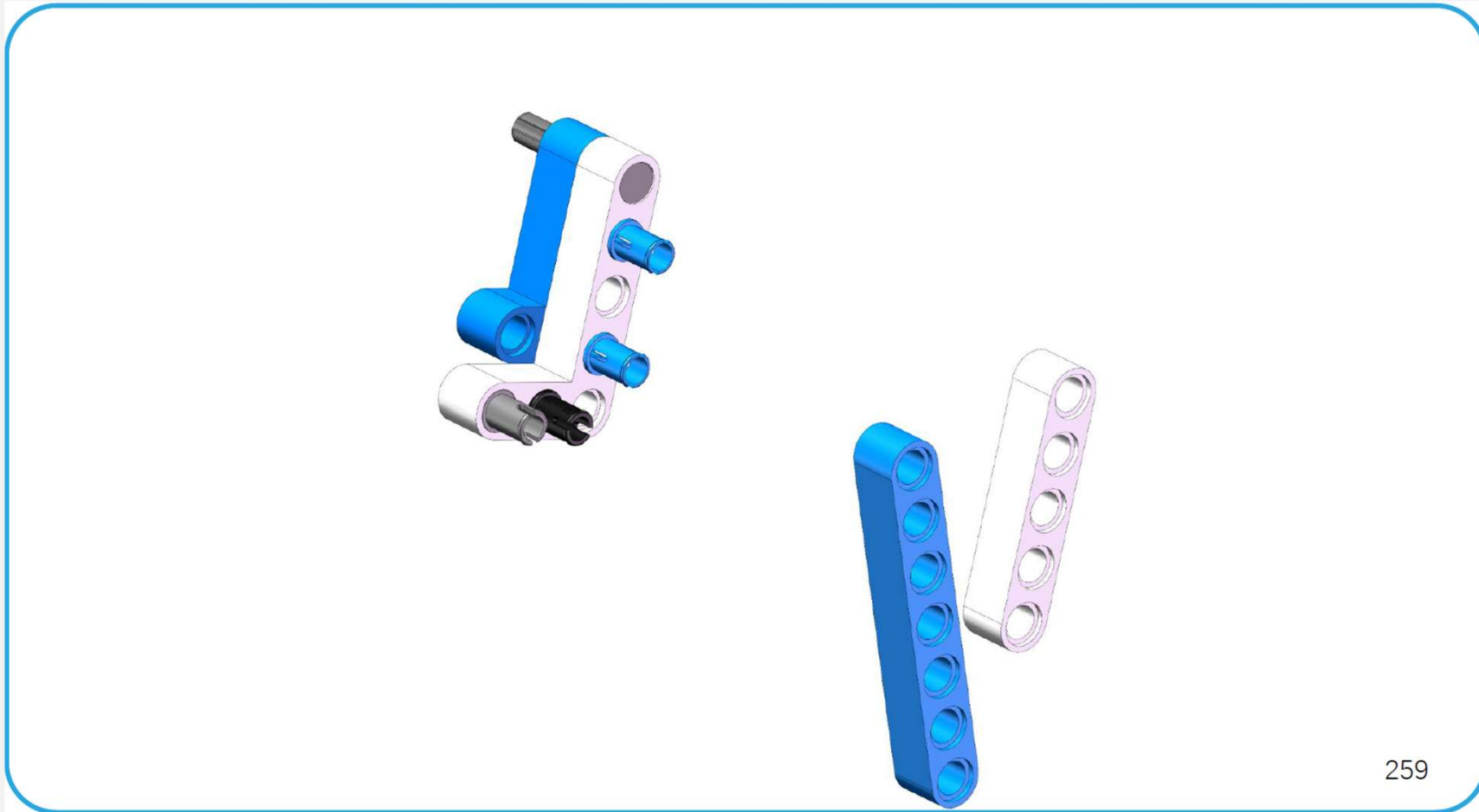
257

Step 27



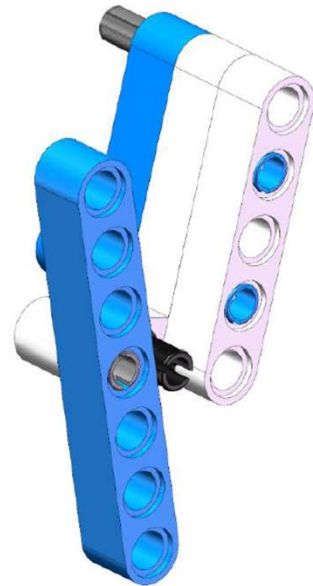
258

Step 28



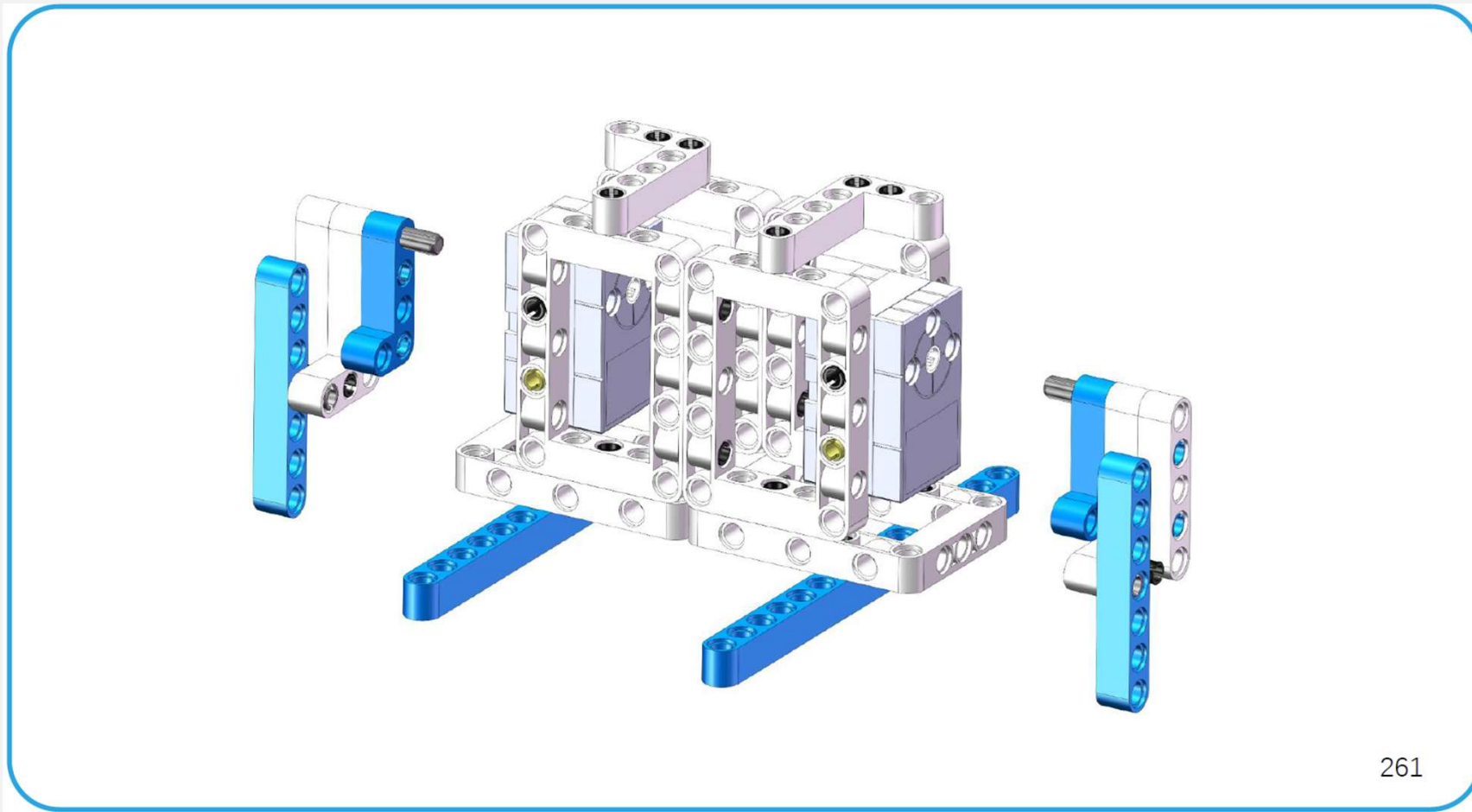
259

Step 29



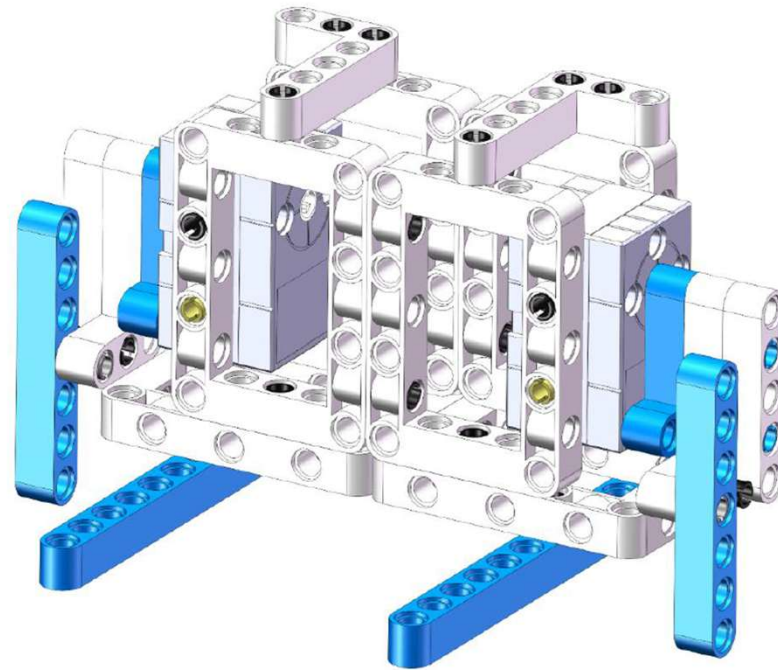
260

Step 30



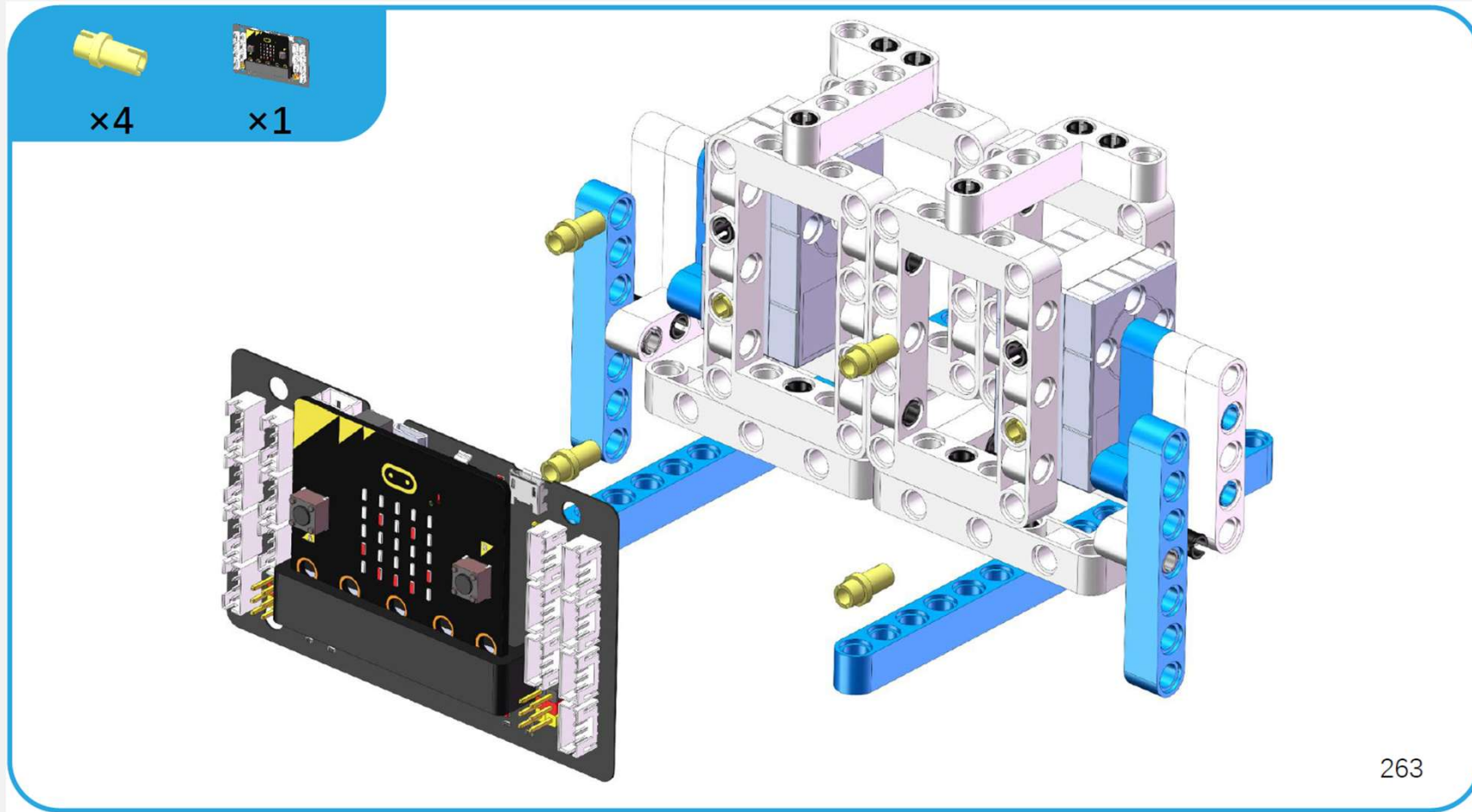
261

Step 31

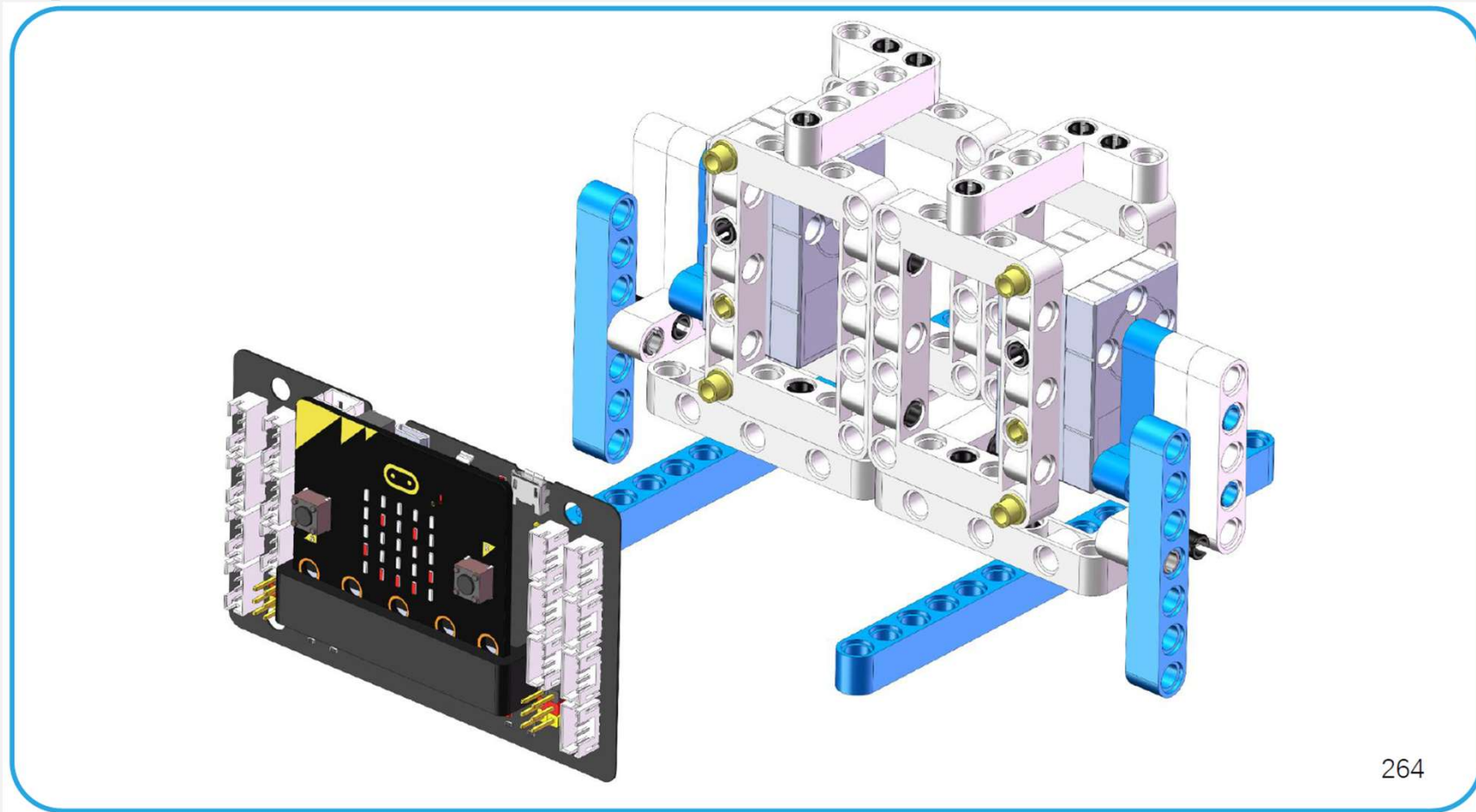


262

Step 32

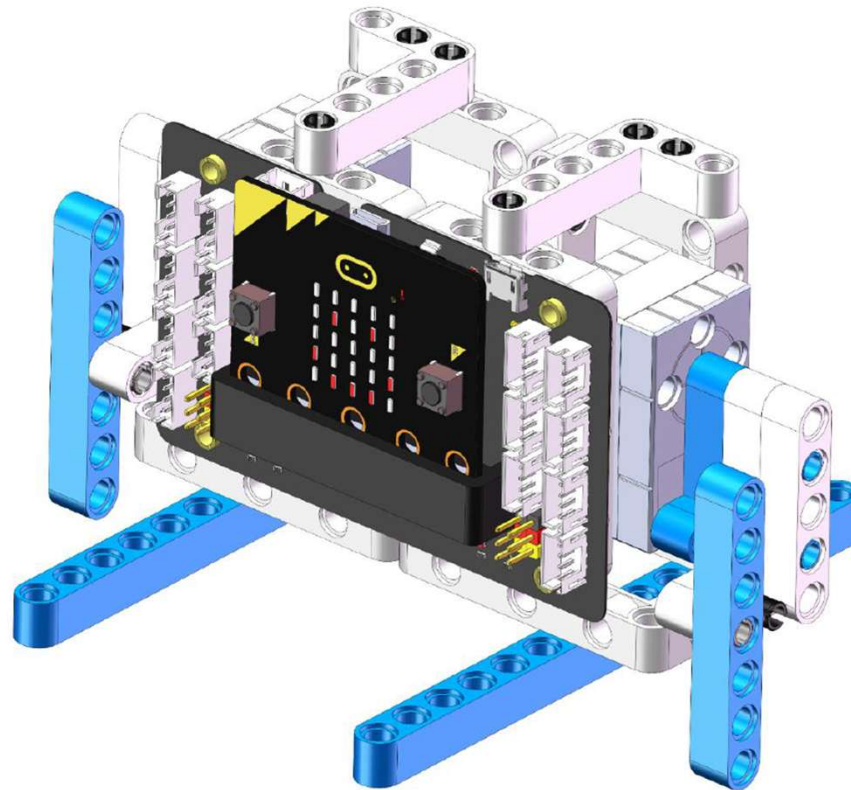


Step 33



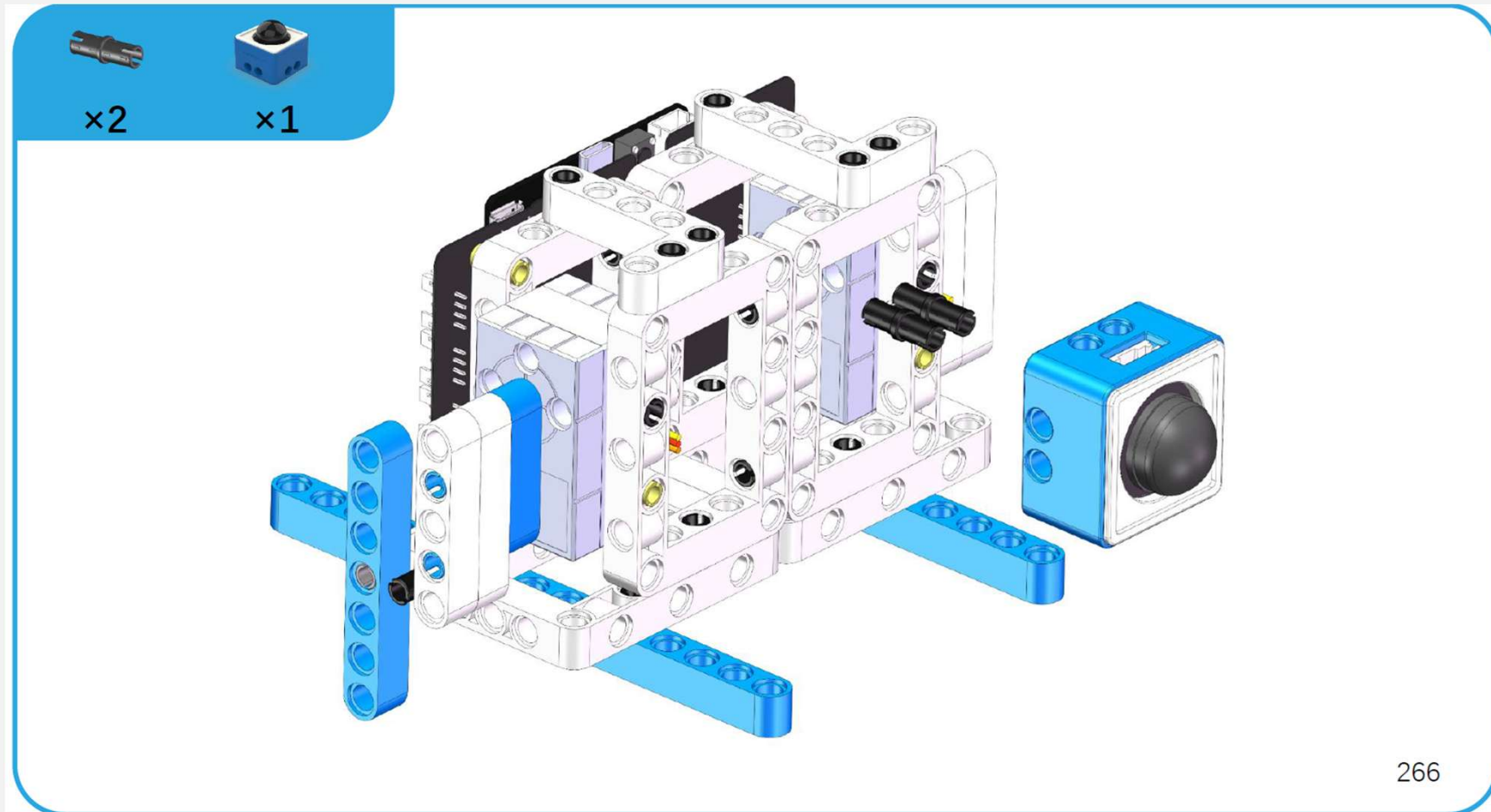
264

Step 34

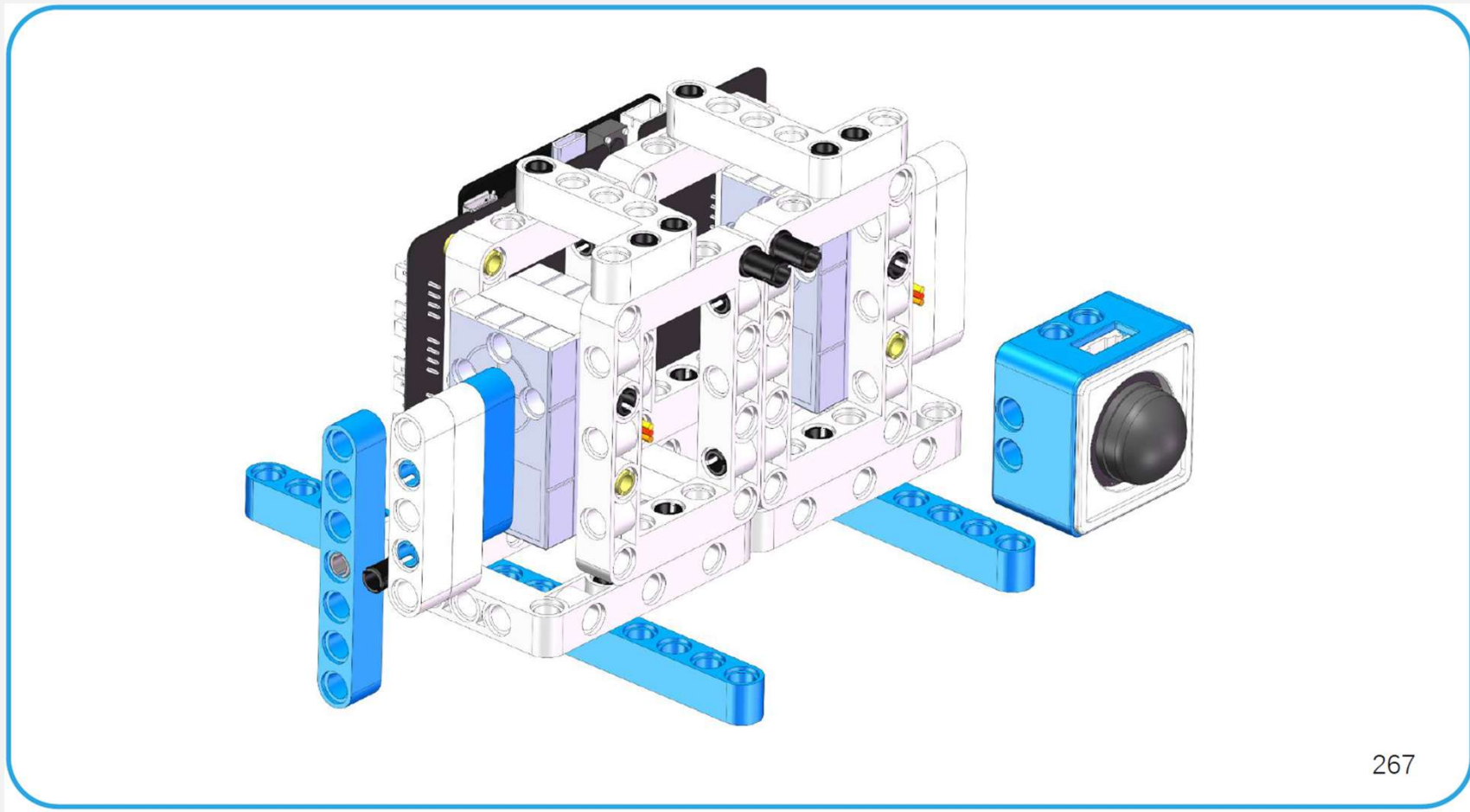


265

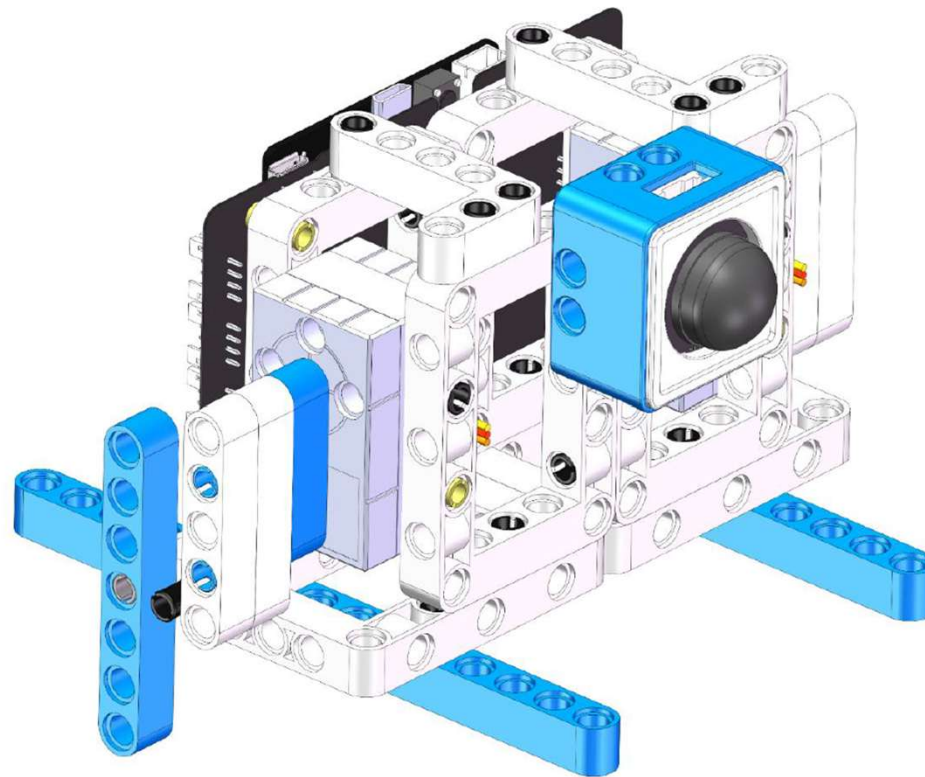
Step 35



Step 36

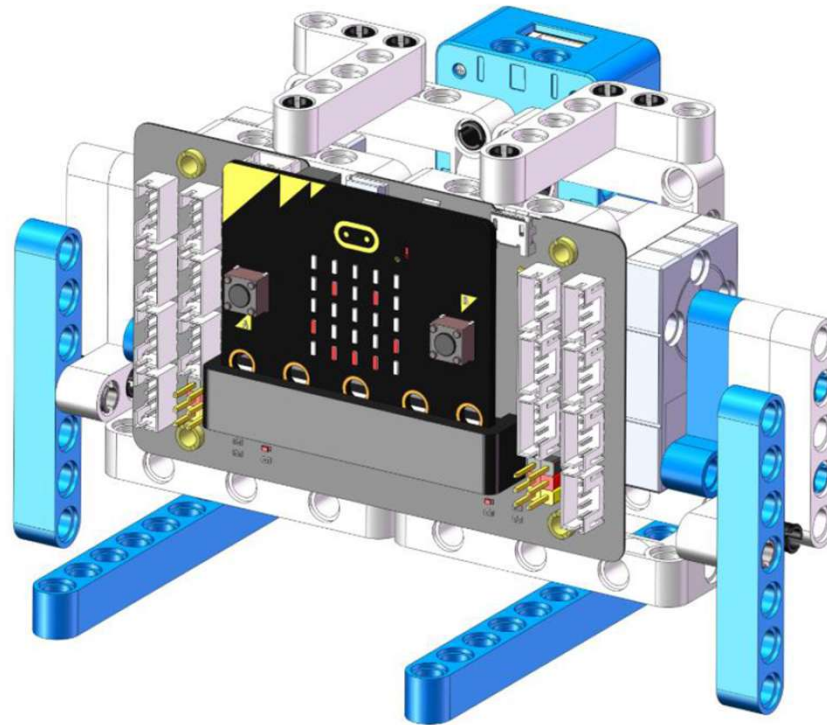


Step 37



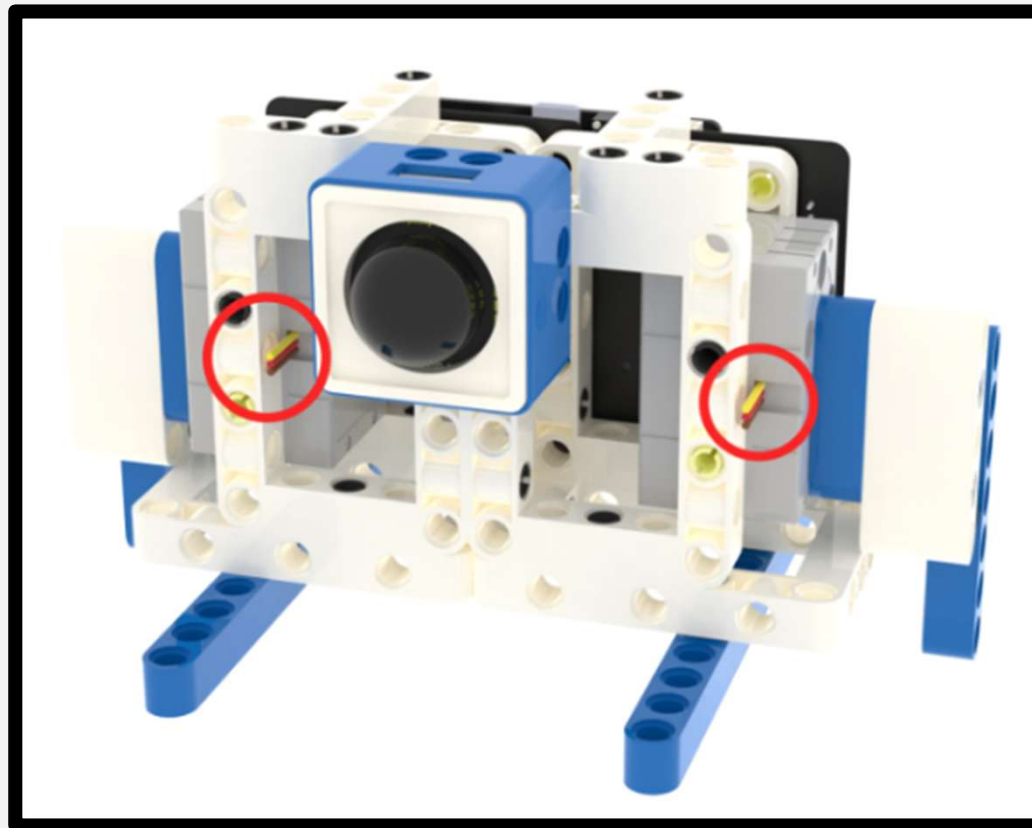
268

Step 38



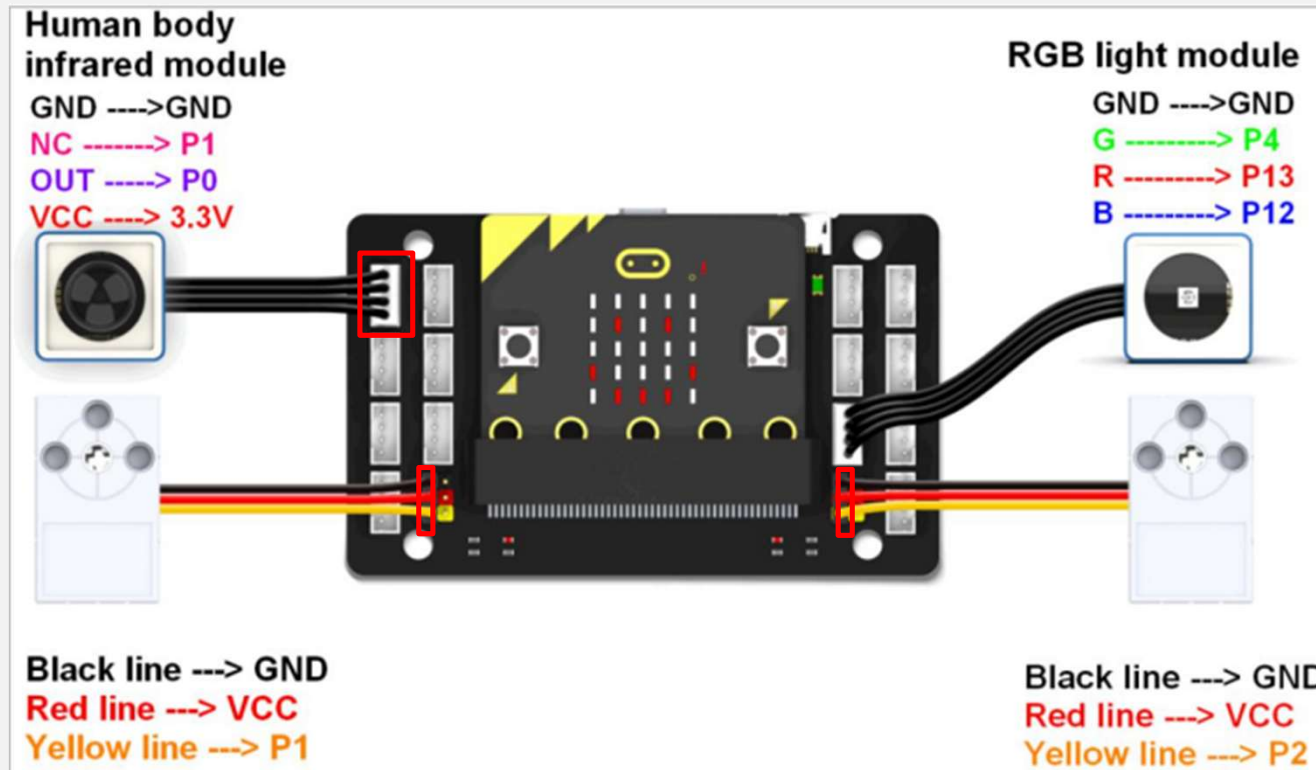
269

After the assembly is completed, please check the wiring of the servo as shown below.



Wire Connection

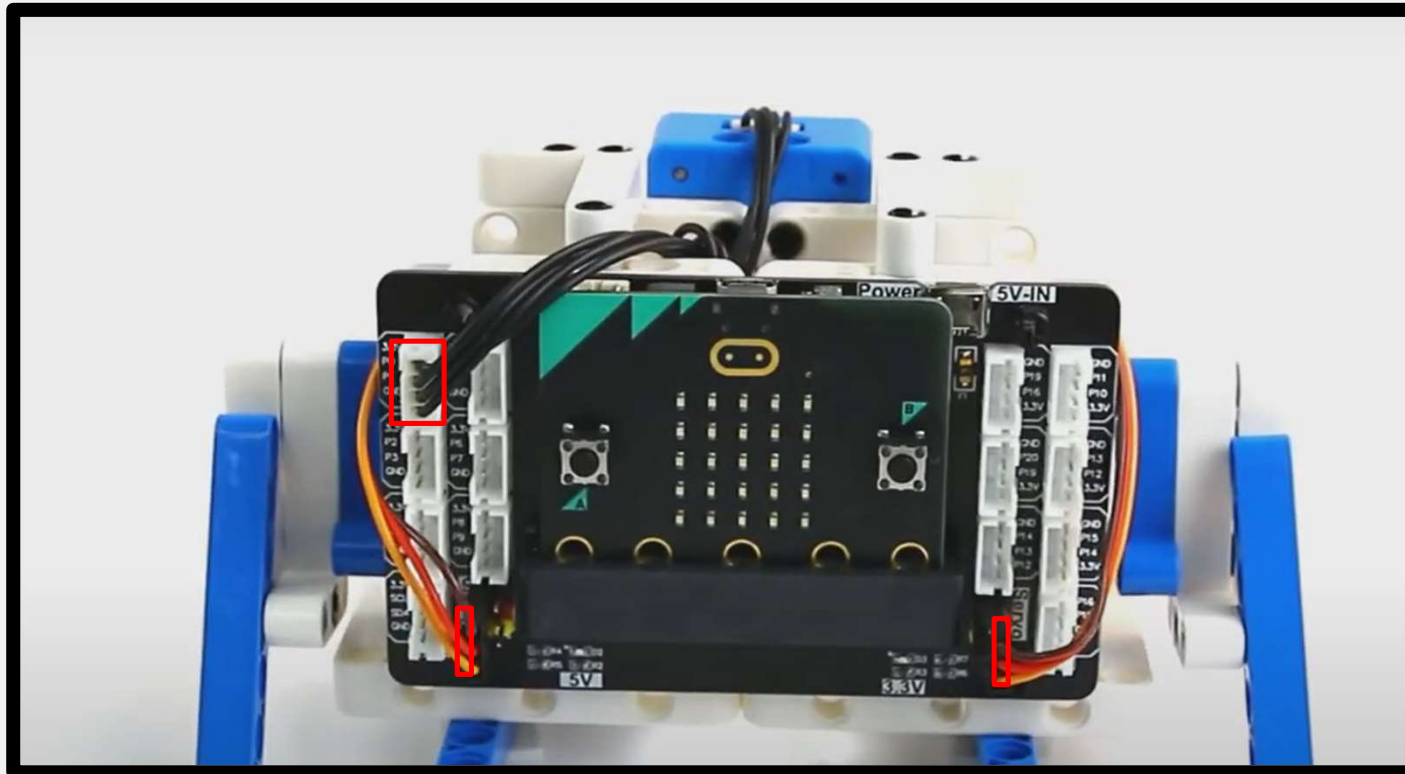
Connect the modules



*RGB is not needed

Let's **connect** the module like this.

Connect the modules



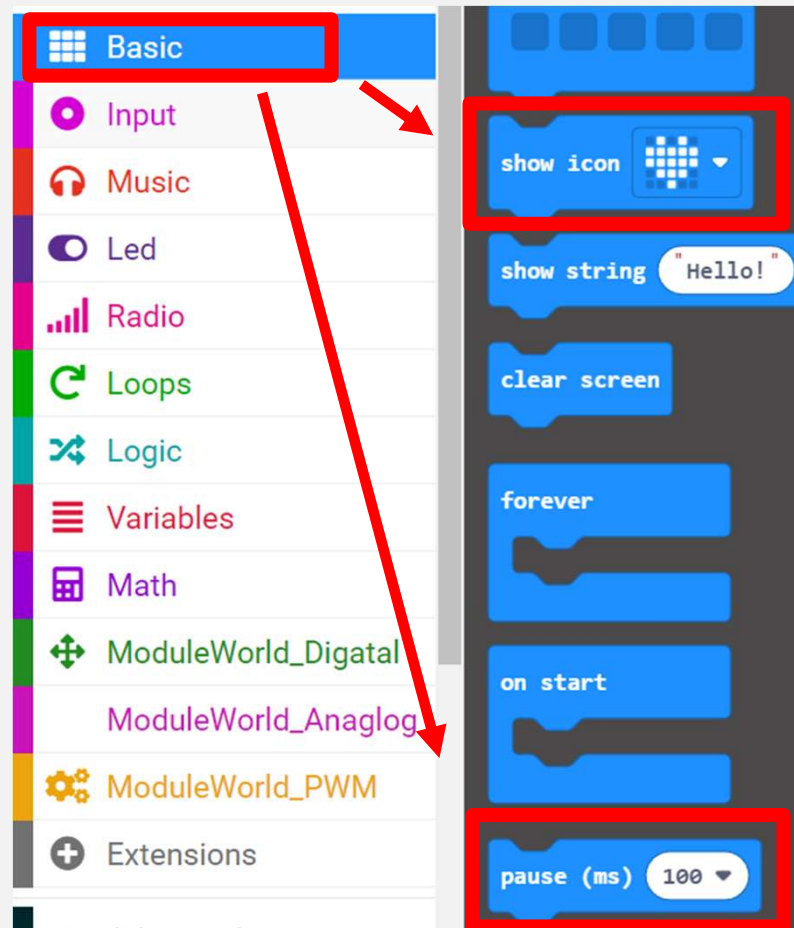
Let's **connect** the module like this.

MakeCode Programming

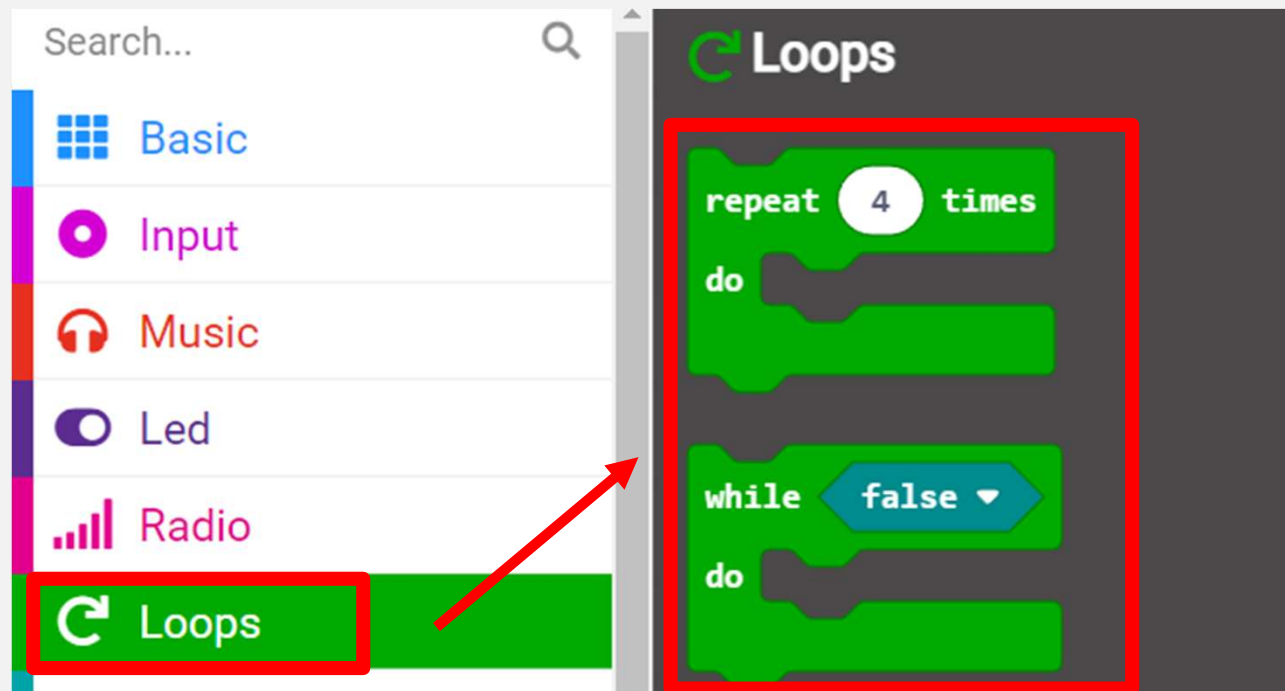
<https://github.com/YahboomTechnology/Module-World>.

Or search **YahboomTechnology/Module-World** in the extension block

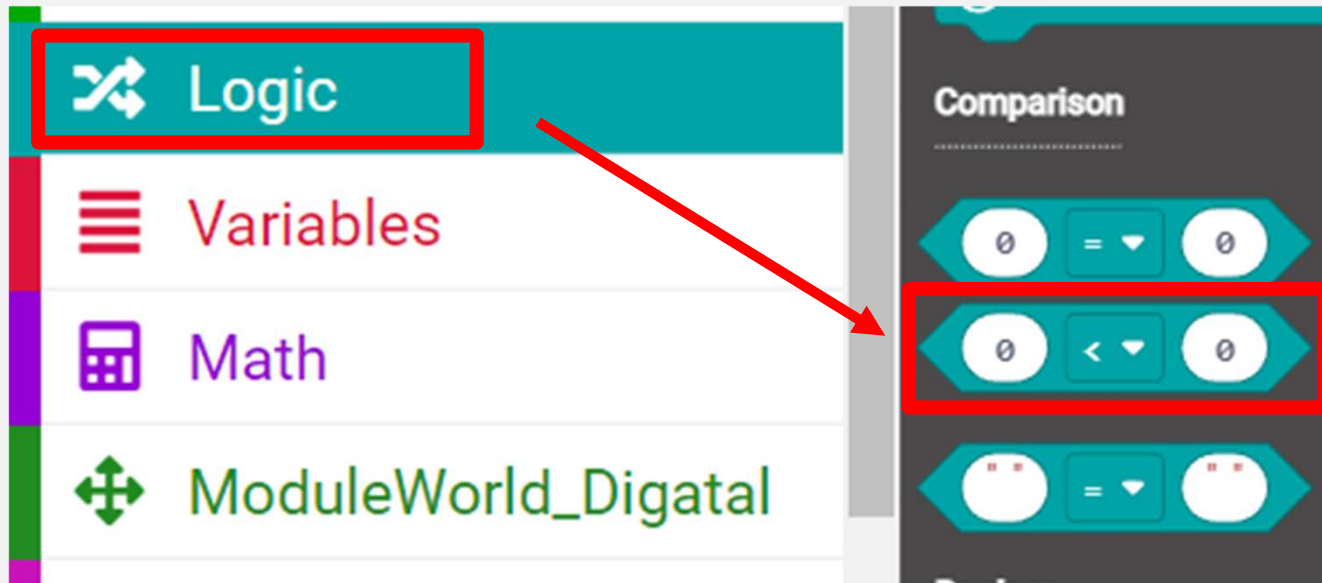
Coding – Basic



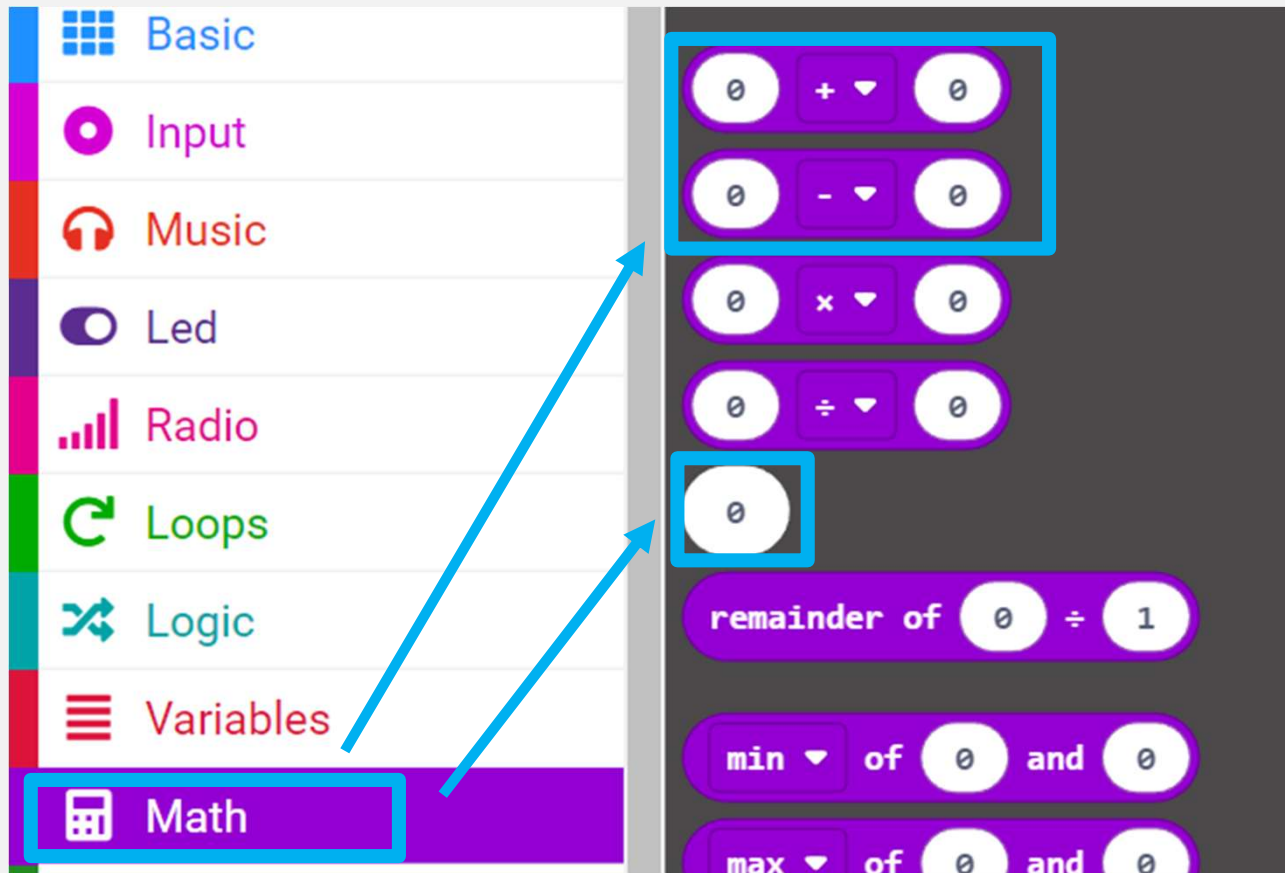
Coding – Loops



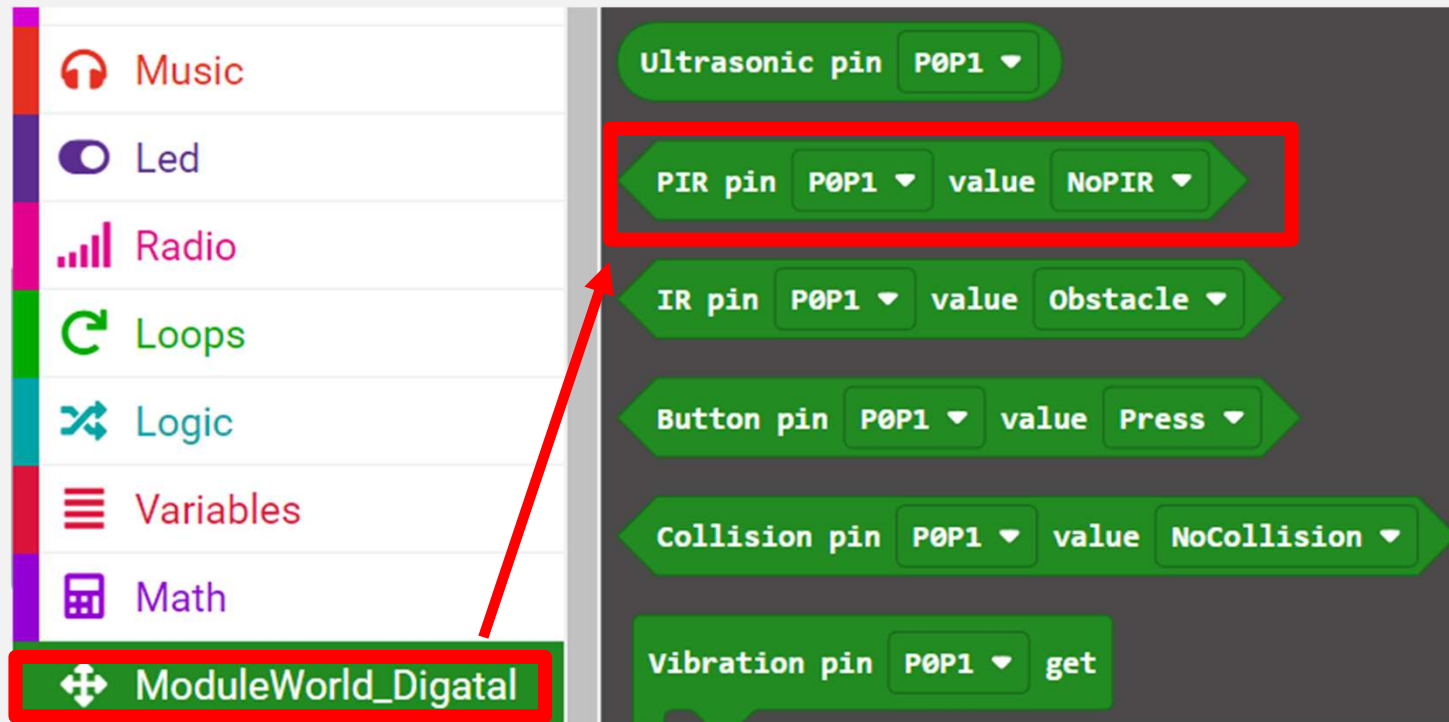
Coding – Logic



Coding – Math



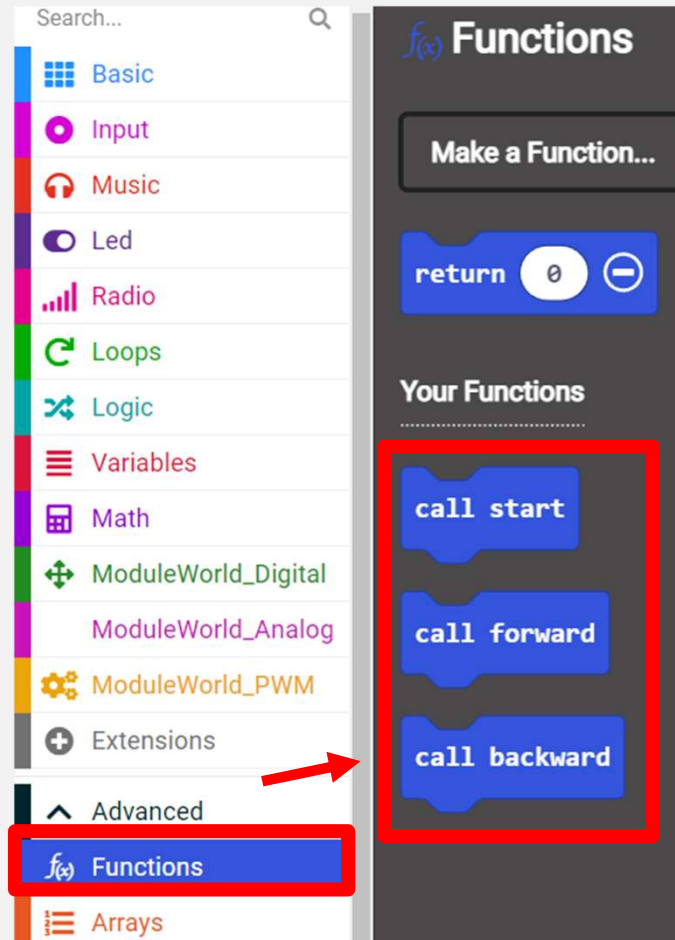
Coding – ModuleWorld_Digatal

A screenshot of the Microbit coding environment. On the left, a sidebar contains various modules: Music, Led, Radio, Loops, Logic, Variables, Math, and ModuleWorld_Digatal. The ModuleWorld_Digatal module is highlighted with a red box. A red arrow points from this box to the main workspace. In the main workspace, several green blocks are visible: Ultrasonic pin (P0P1), PIR pin (P0P1) value (NoPIR), IR pin (P0P1) value (Obstacle), Button pin (P0P1) value (Press), Collision pin (P0P1) value (NoCollision), and Vibration pin (P0P1) get. The PIR pin block is also highlighted with a red box.

Coding – ModuleWorld_PWM



Coding – Advanced > Functions



Coding - Overview



```
on start
  call start
  show icon [grid icon]
  pause (ms) 2000

forever
  while PIR pin P0P1 value OPIR
  do
    repeat 2 times
    do
      call backward
      pause (ms) 500
      call forward
      pause (ms) 500
    end
  end
  call start

function start
  Servo(360) pin P1 value 180
  Servo(360) pin P2 value 180

function forward
  Servo(360) pin P1 value 220
  Servo(360) pin P2 value 220

function backward
  Servo(360) pin P1 value 130
  Servo(360) pin P2 value 130
```

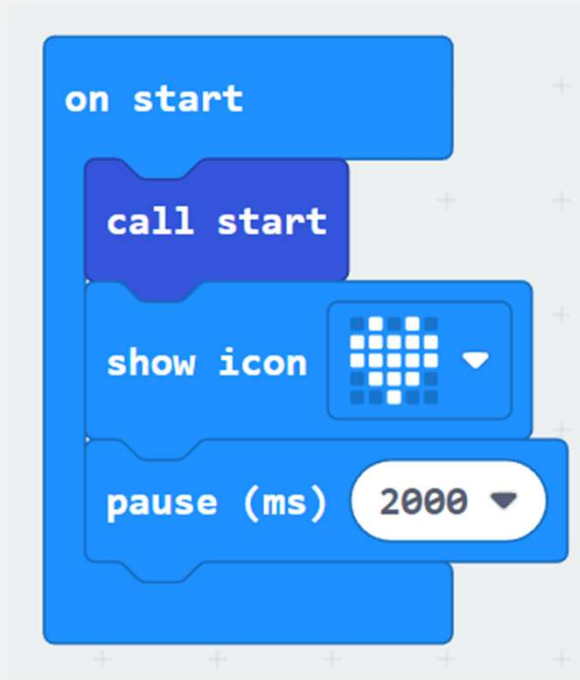
Coding – Functions

```
function start  
  Servo(360) pin P1 value 180  
  Servo(360) pin P2 value 180
```

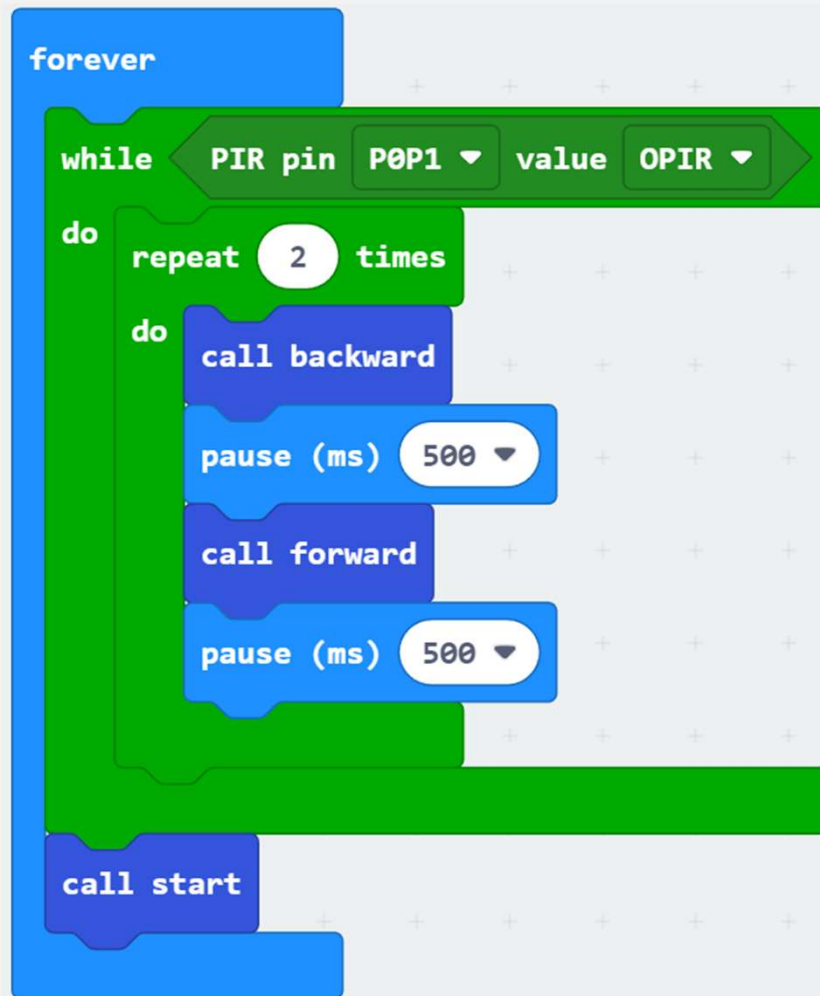
```
function forward  
  Servo(360) pin P1 value 220  
  Servo(360) pin P2 value 220
```

```
function backward  
  Servo(360) pin P1 value 130  
  Servo(360) pin P2 value 130
```

Coding – On start



Coding – forever

A Scratch code block for a "forever" loop. The loop contains a "while" block with the condition "PIR pin P0P1 value OPIR". Inside the "while" block is a "do" block containing a "repeat 2 times" block. The "repeat" block contains two "do" blocks: the first contains "call backward" and "pause (ms) 500"; the second contains "call forward" and "pause (ms) 500". Below the "while" block is a "call start" block.

```
forever
  while PIR pin P0P1 value OPIR
  do
    repeat 2 times
    do
      call backward
      pause (ms) 500
    do
      call forward
      pause (ms) 500
    call start
```

Coding - Overview



```
on start
  call start
  show icon [grid icon]
  pause (ms) 2000

forever
  while PIR pin P0P1 value OPIR
  do
    repeat 2 times
    do
      call backward
      pause (ms) 500
      call forward
      pause (ms) 500
    call start

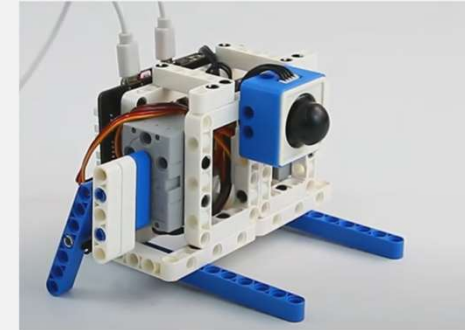
function start
  Servo(360) pin P1 value 180
  Servo(360) pin P2 value 180

function forward
  Servo(360) pin P1 value 220
  Servo(360) pin P2 value 220

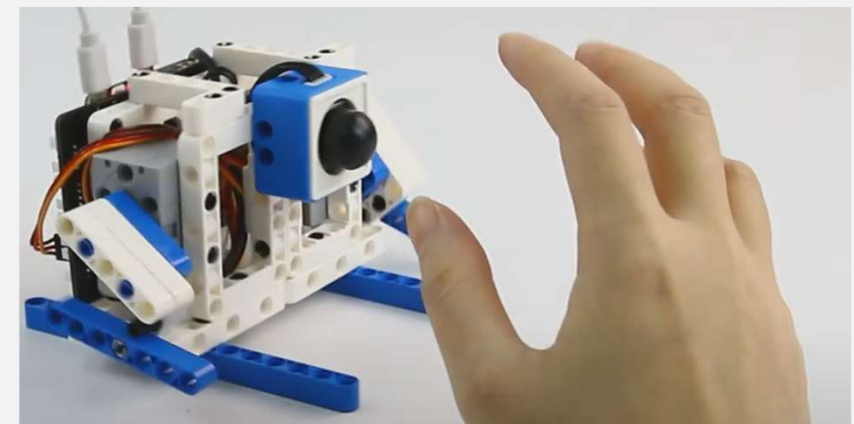
function backward
  Servo(360) pin P1 value 130
  Servo(360) pin P2 value 130
```

Phenomenon

After the program is downloaded successfully.
 We can see a love heart displayed on the Micro:bit board, indicating that it is still being **initialized**, and the human infrared module **cannot** be triggered at this time.

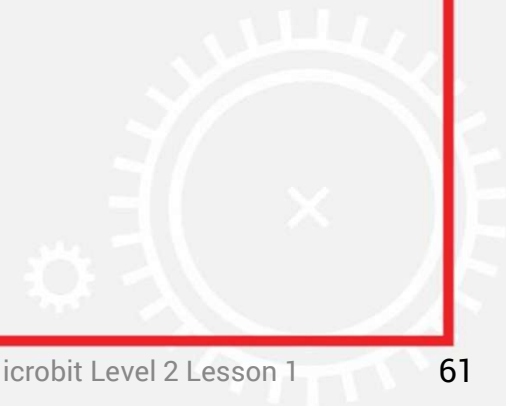


When the human body infrared module **detects** that there is a **human body** moving, the escape box will swing its arms and **move forward slowly**.



CHALLENGE

for : Lesson 1

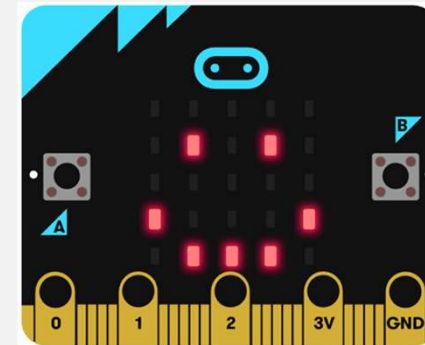
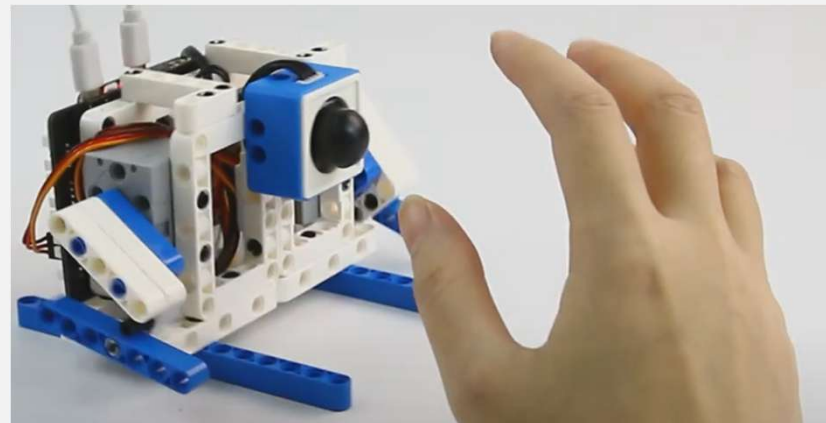
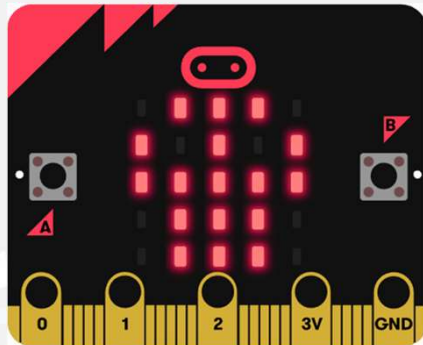


L1 – Challenge

After the program is downloaded successfully.

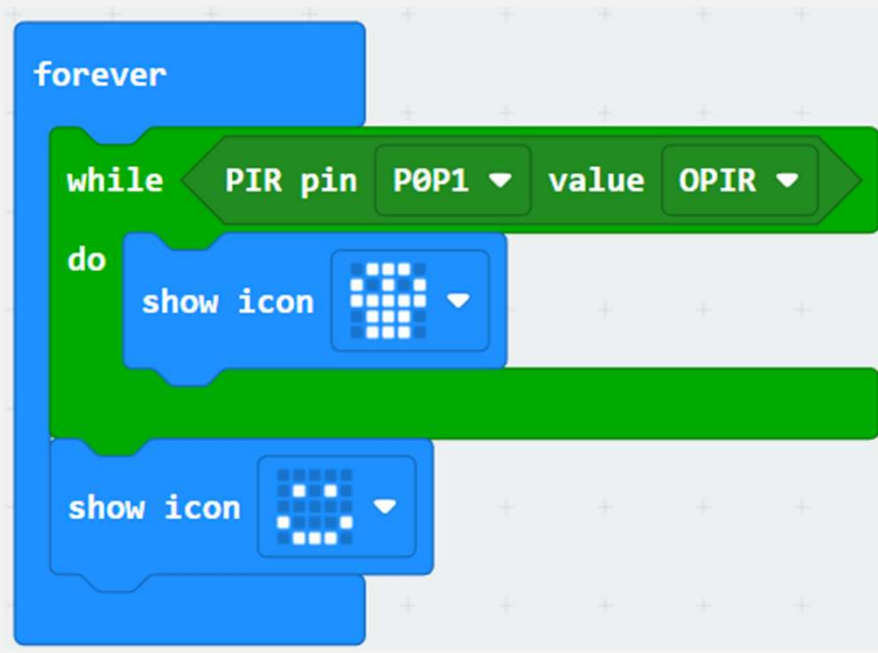
When it **detect** movements, the dot matrix will switch to an **alarm pattern** and the escape box will swing its arms and move forward slowly.

If there is no movement, shows a **smile pattern**, it means that the module is in **normal** working condition



Example

Coding – forever

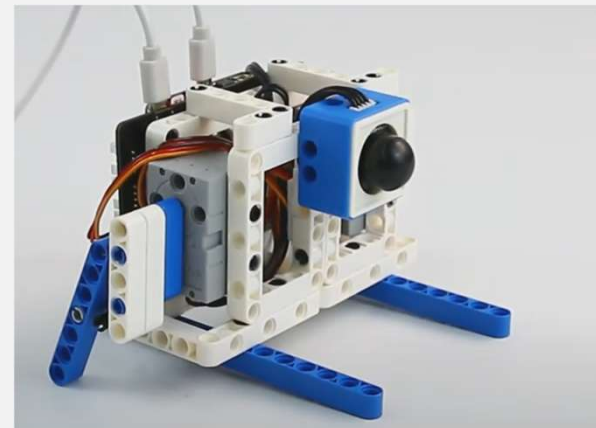
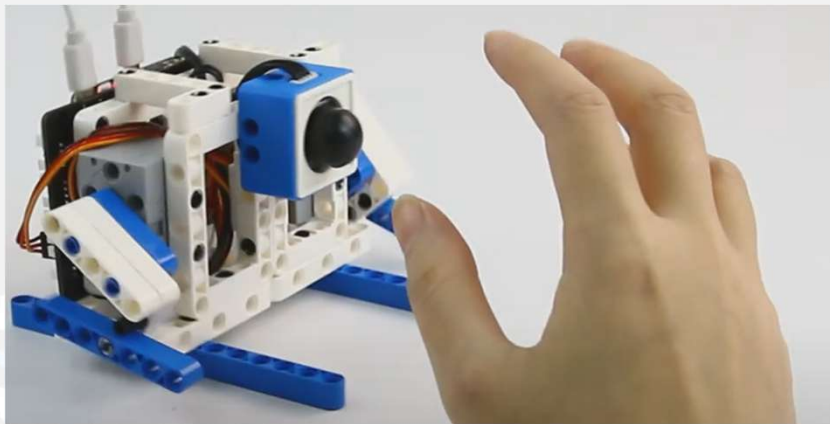
A Scratch code block for a Microbit. It consists of a blue "forever" loop block containing a green "while" loop block. The "while" block has "PIR pin" set to "P0P1" and "value" set to "OPIR". Inside the "while" loop is a blue "do" block containing a "show icon" block with a grid icon. Below the "while" loop, outside the "do" block, is another blue "show icon" block with the same grid icon.

```
forever
  while PIR pin P0P1 value OPIR
  do
    show icon [grid icon]
  show icon [grid icon]
```

L1 – Challenge 2

After the program is downloaded successfully.

When it **detect** movements, **instead** moving forward, the escape box will swing its arms and move **backward** slowly.

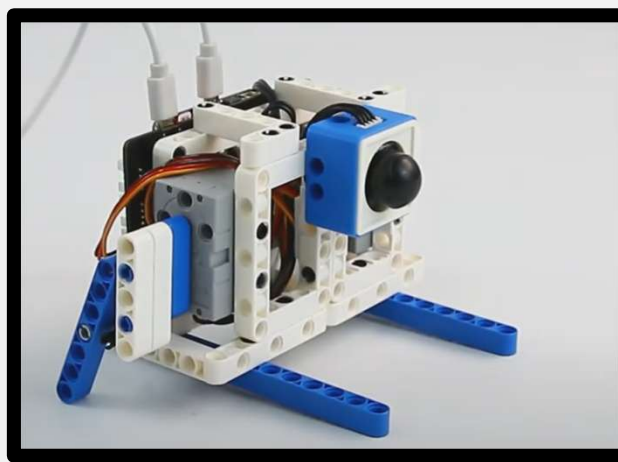
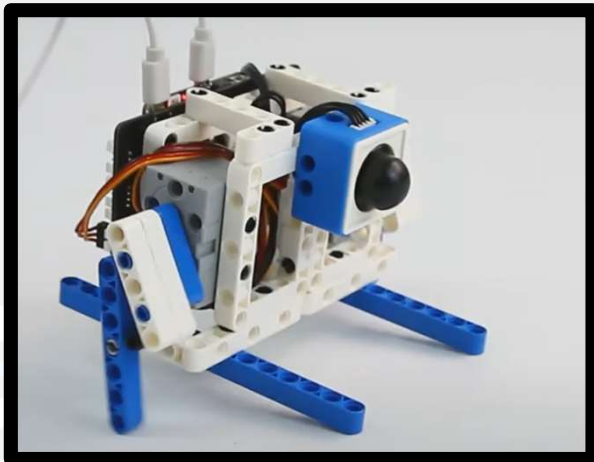


L1 – Challenge 3

After the program is downloaded successfully.

When it **detect** movements, buzzer will also **play an alarm sound**:

- The volume of the sound will **rise** at first
- The volume of the sound will **fall** at last

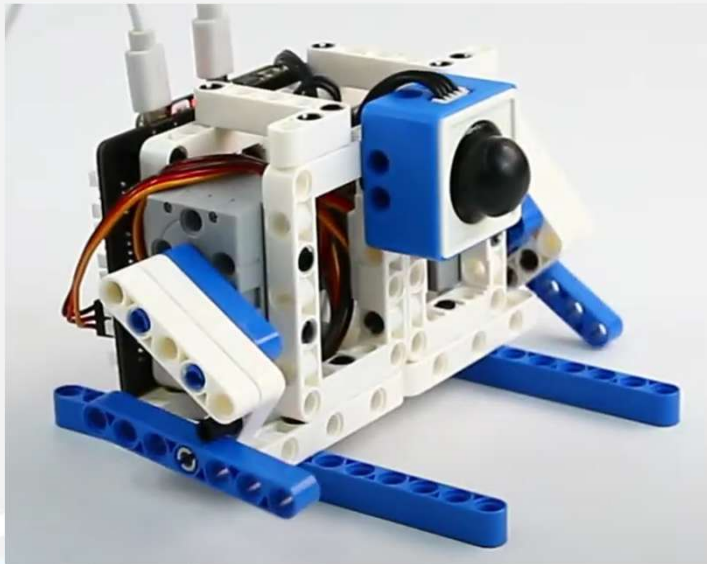


Example Coding – forever

the **buzzer** will play the alarm sound

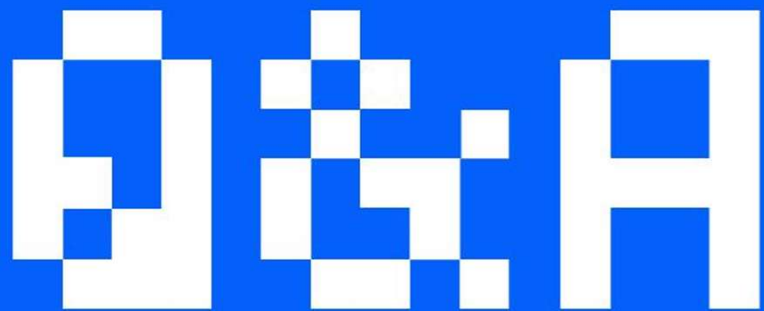
```
forever
  set X to 200
  while PIR pin P0P1 value OPIR
    do
      set volume 255
      while X < 800
        do
          set X to X + 1
          play tone X for 1
      while X > 200
        do
          set X to X - 1
          play tone X for 2
  set volume 0
```

L8 - Mission



After the program is downloaded successfully.

When it **detect** movements, the running box will swing its arms and move **forward/backward** , but if there is an obstacle, it will **stop**.



Any
Questions?

Thank you :)