



Microbit Robotics Beginner Level 2

Lesson 2

Lifting platform

Presented by Advanced Superlogic Team

Today's Topic

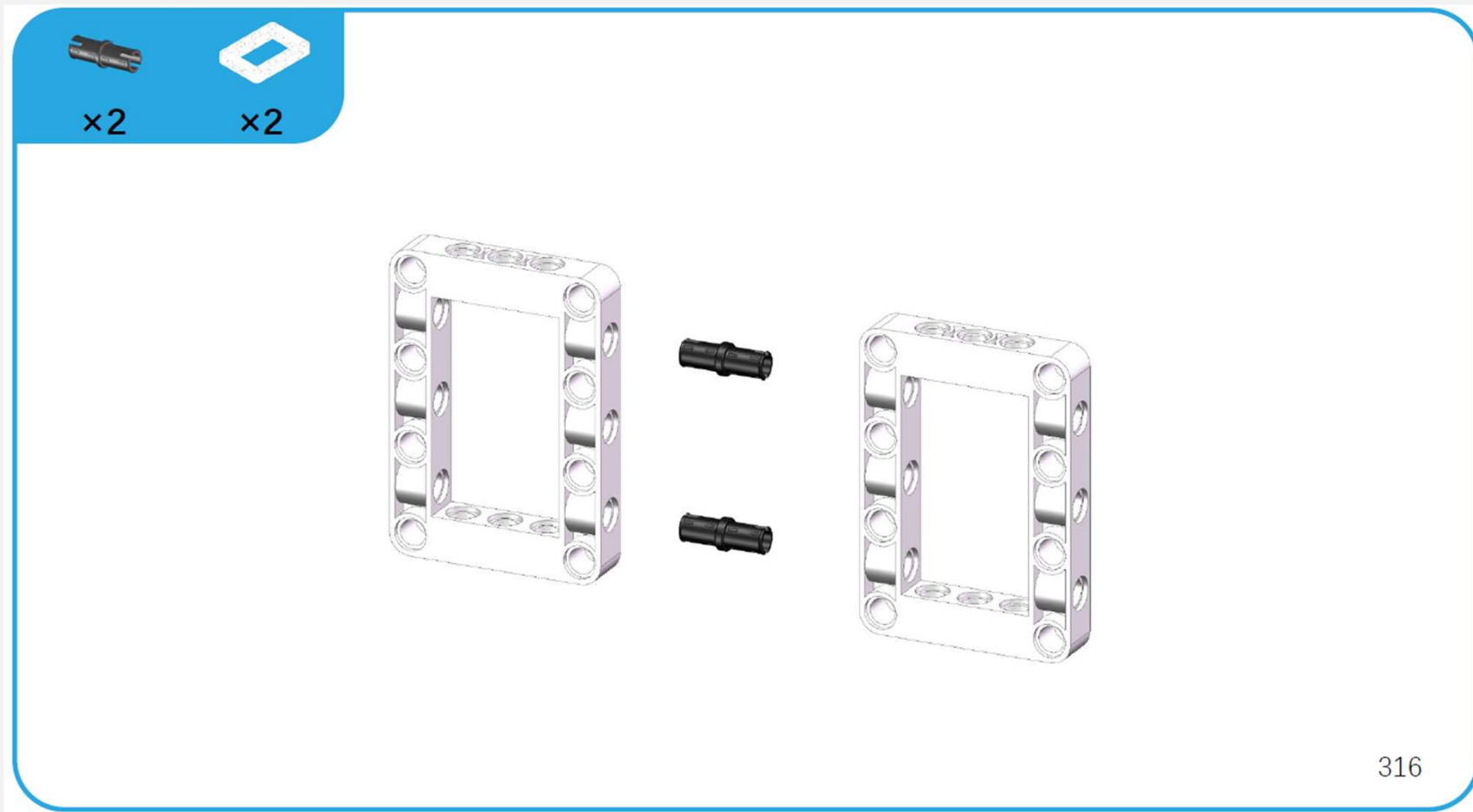
- 1. Build a Lifting platform with World of Modules**
- 2. Servo programming**
- 3. Function programming**

Learning Outcome

- 1. Able to build Lifting platform with instruction manual**
- 2. Understand Servo programming work**
- 3. Understand Function programming work**
- 4. Able to program Lifting platform**

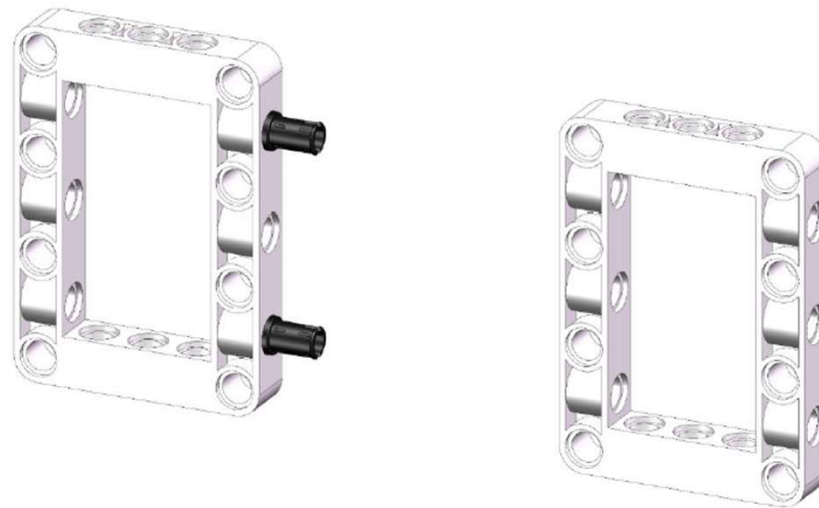
Lets build Lifting Platform

Step 1



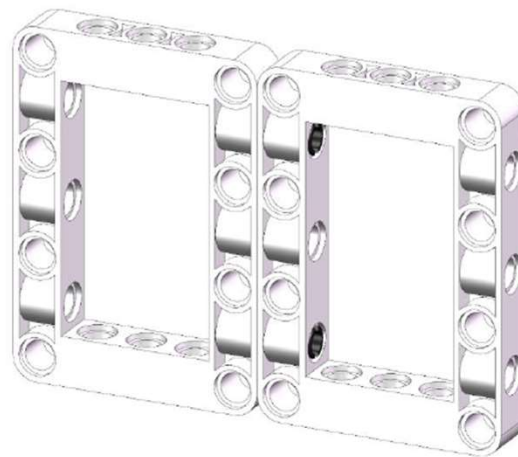
316

Step 2



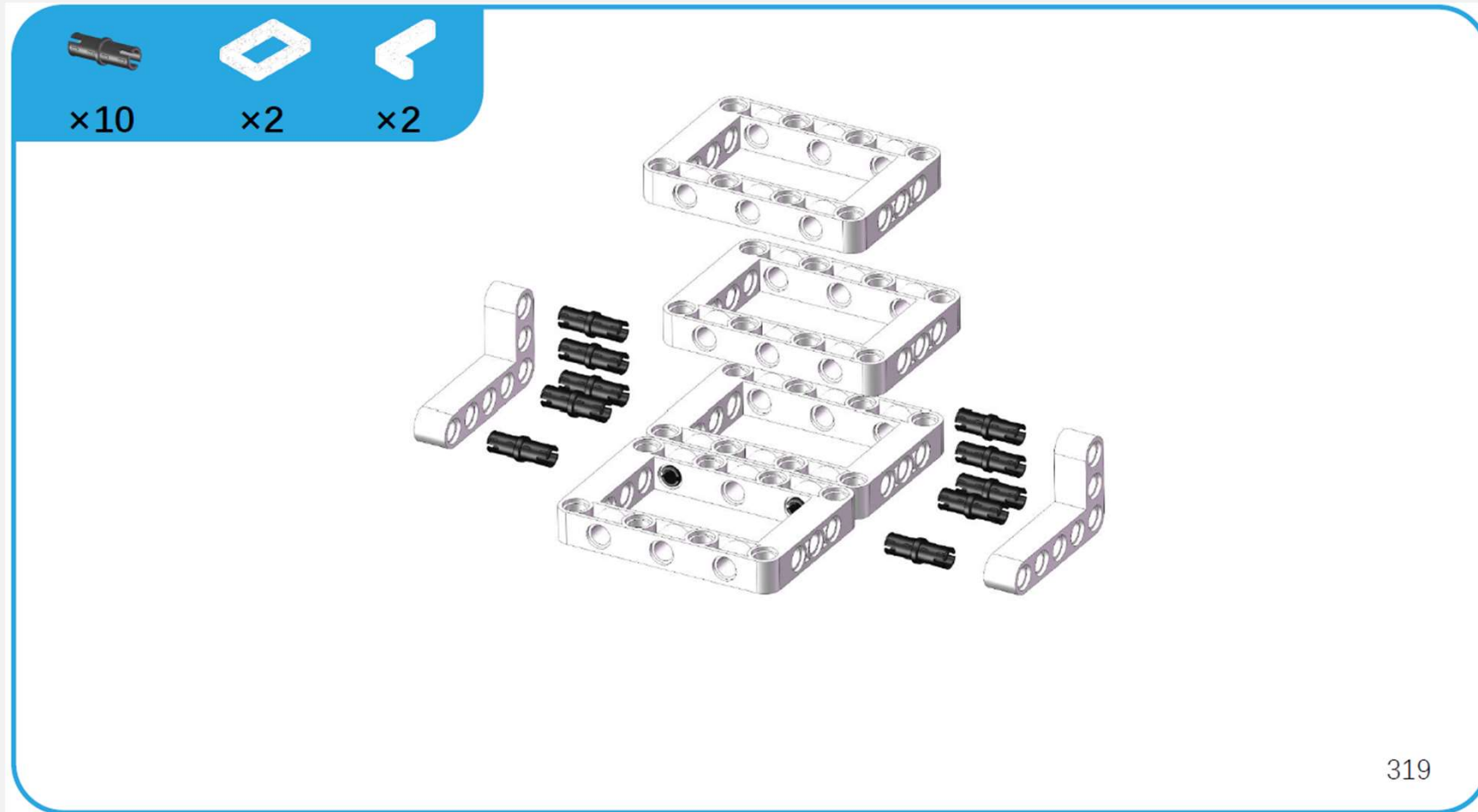
317

Step 3

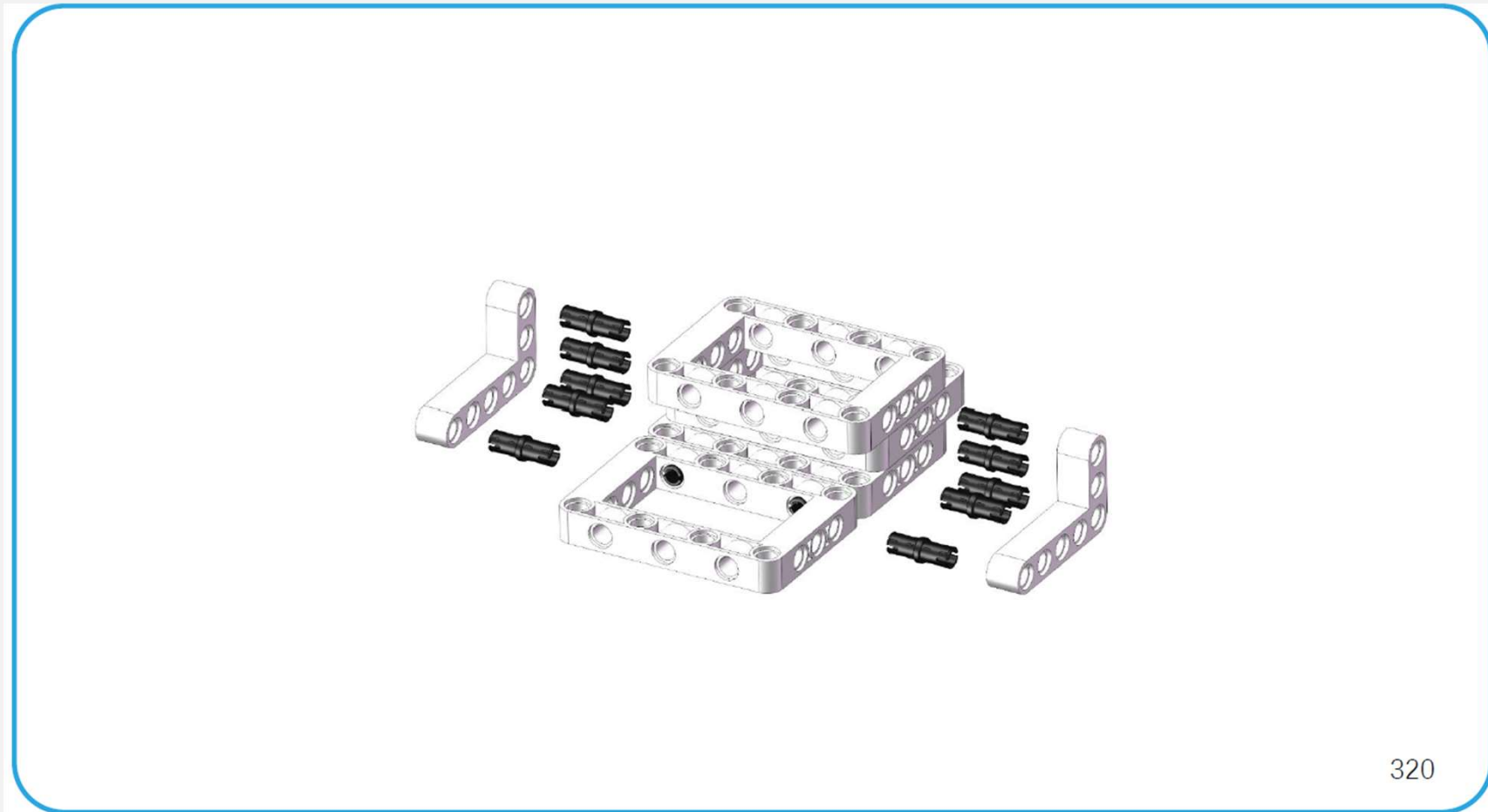


318

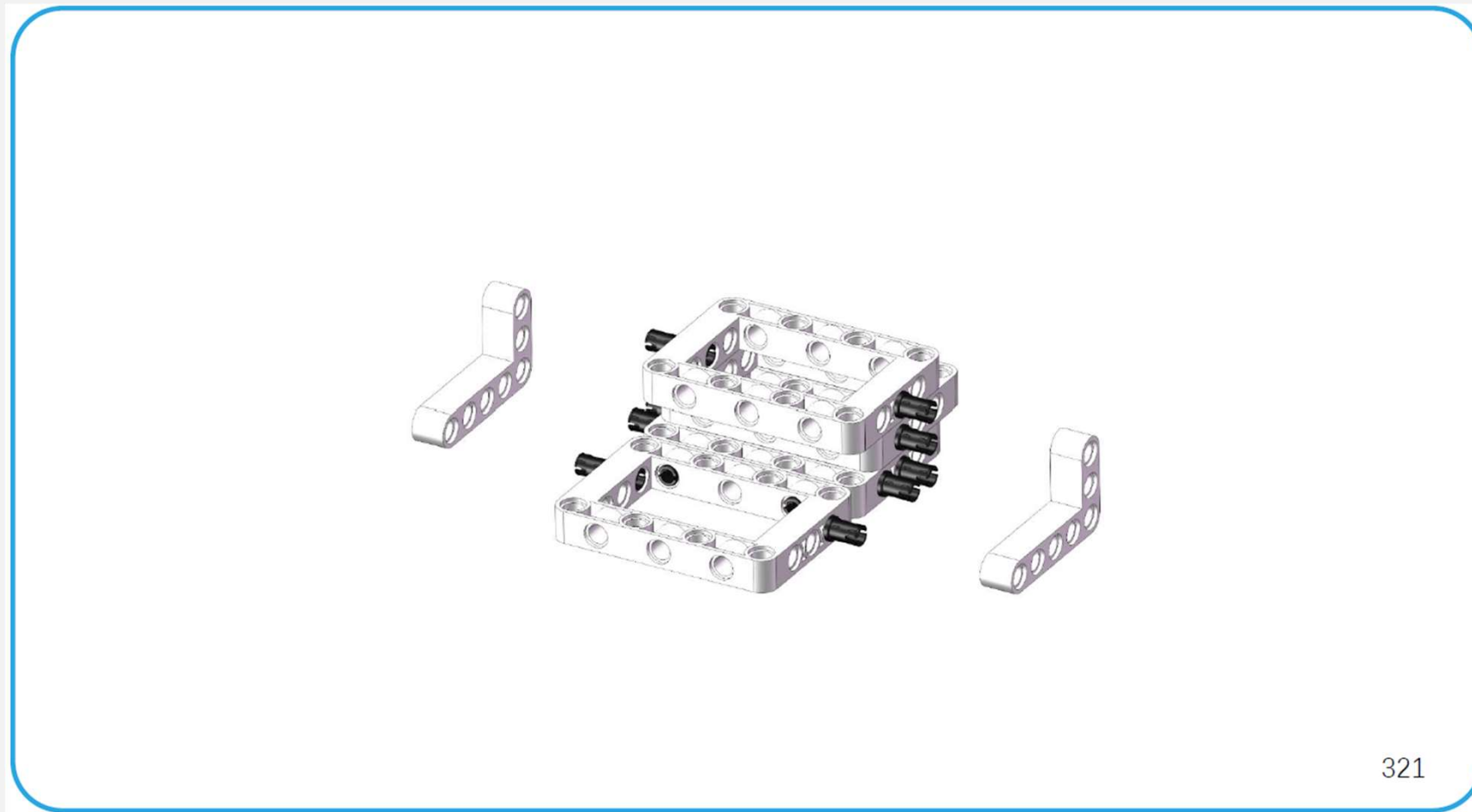
Step 4



Step 5

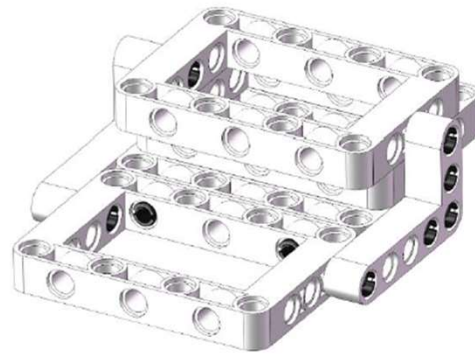


Step 6



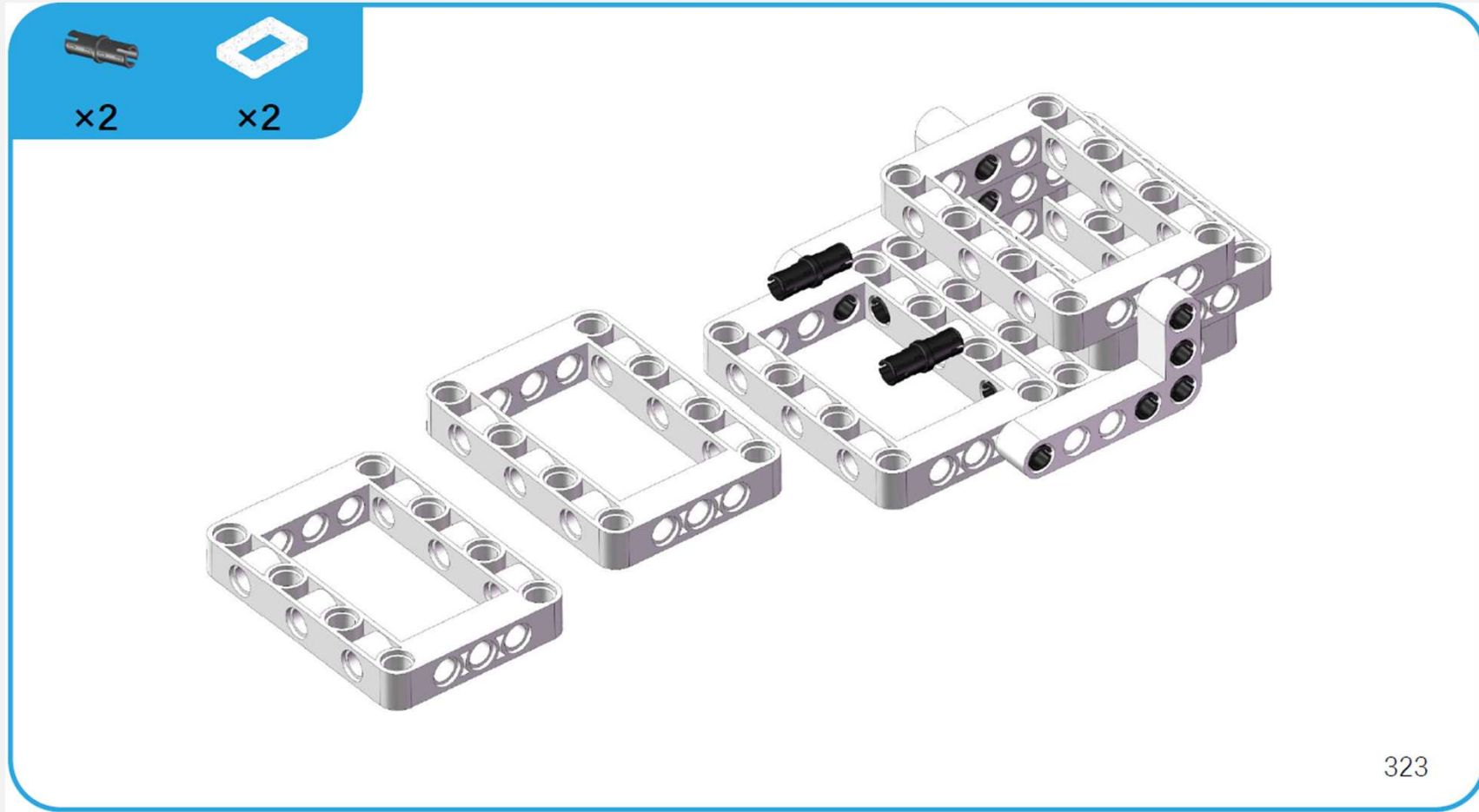
321

Step 7



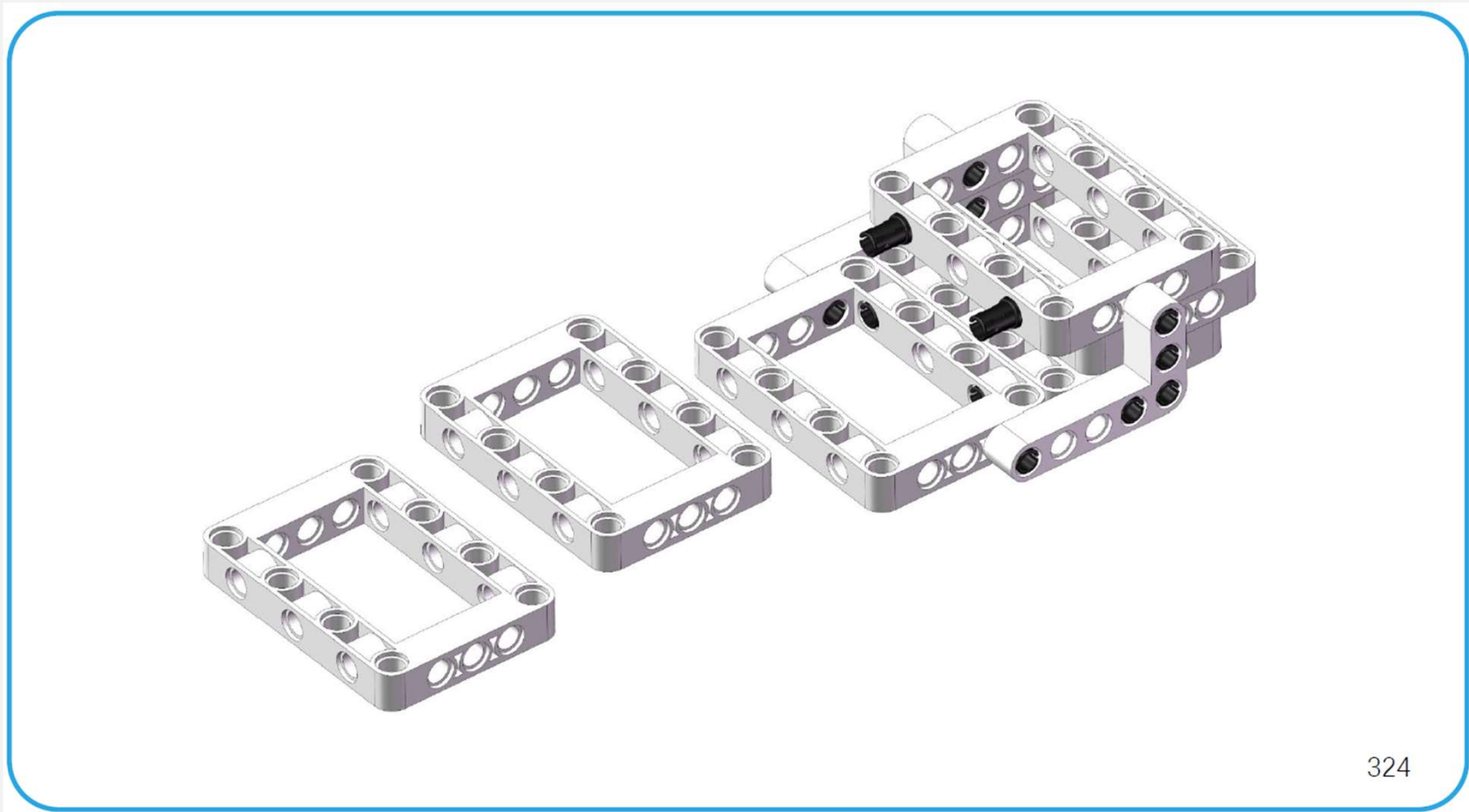
322

Step 8



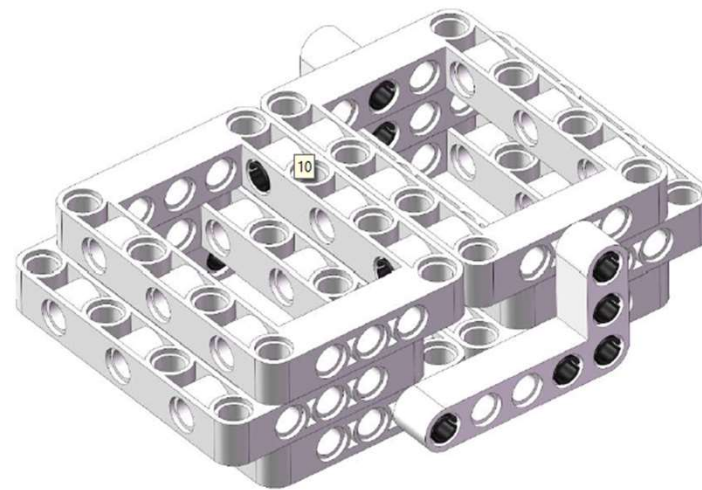
323

Step 9



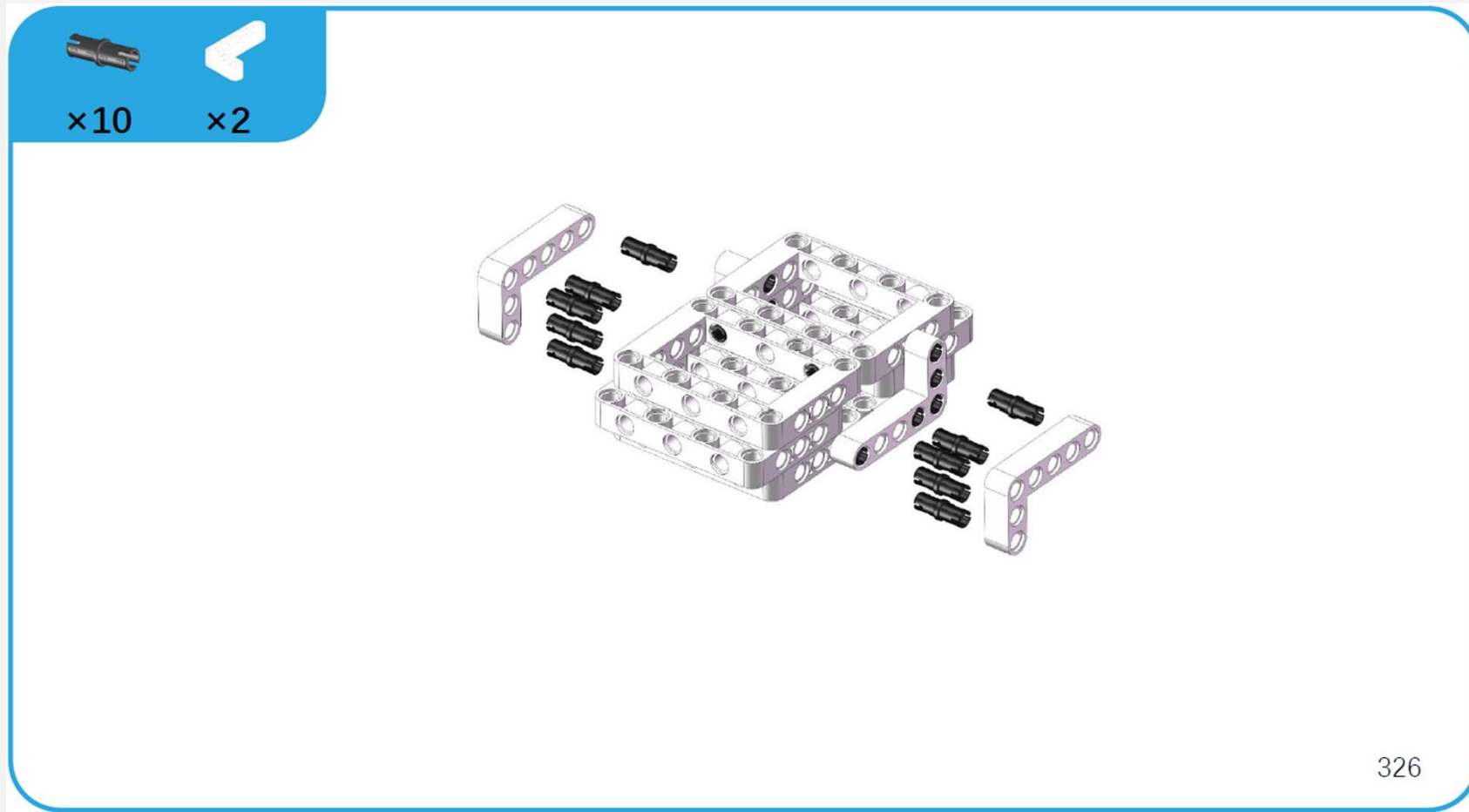
324

Step 10

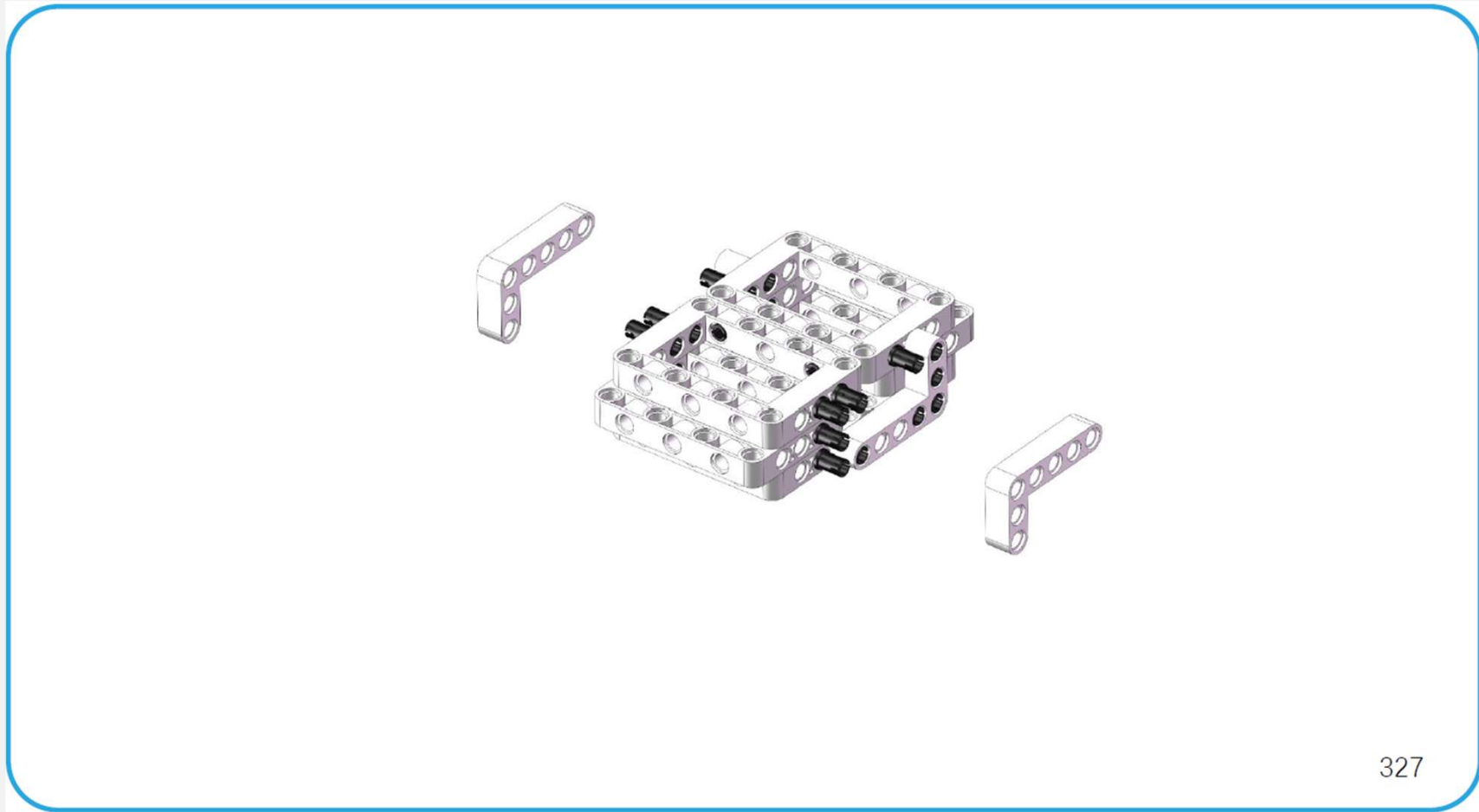


325

Step 11

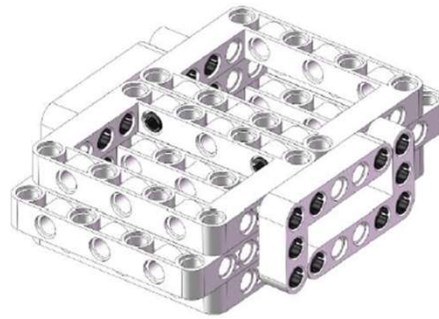


Step 12



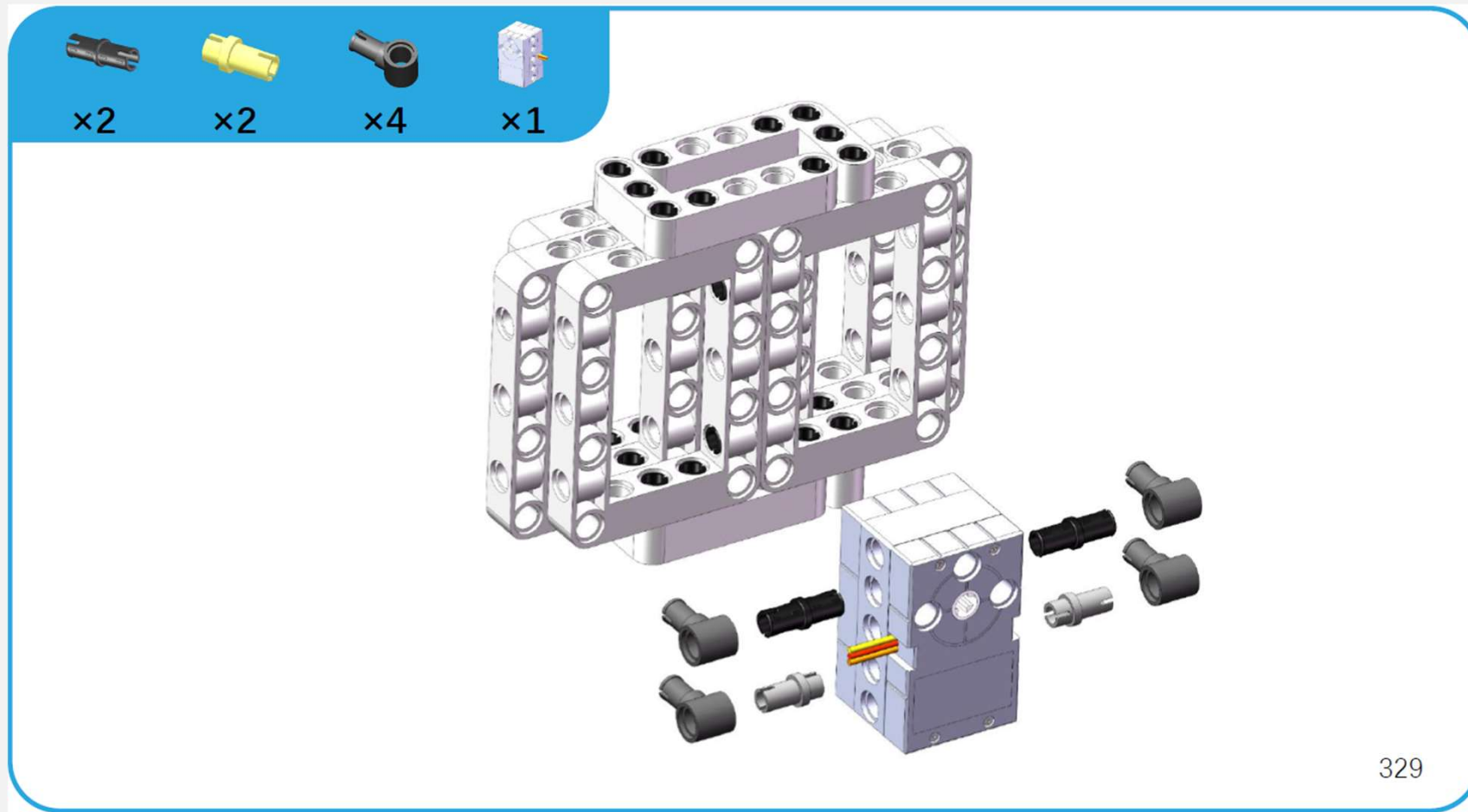
327

Step 13

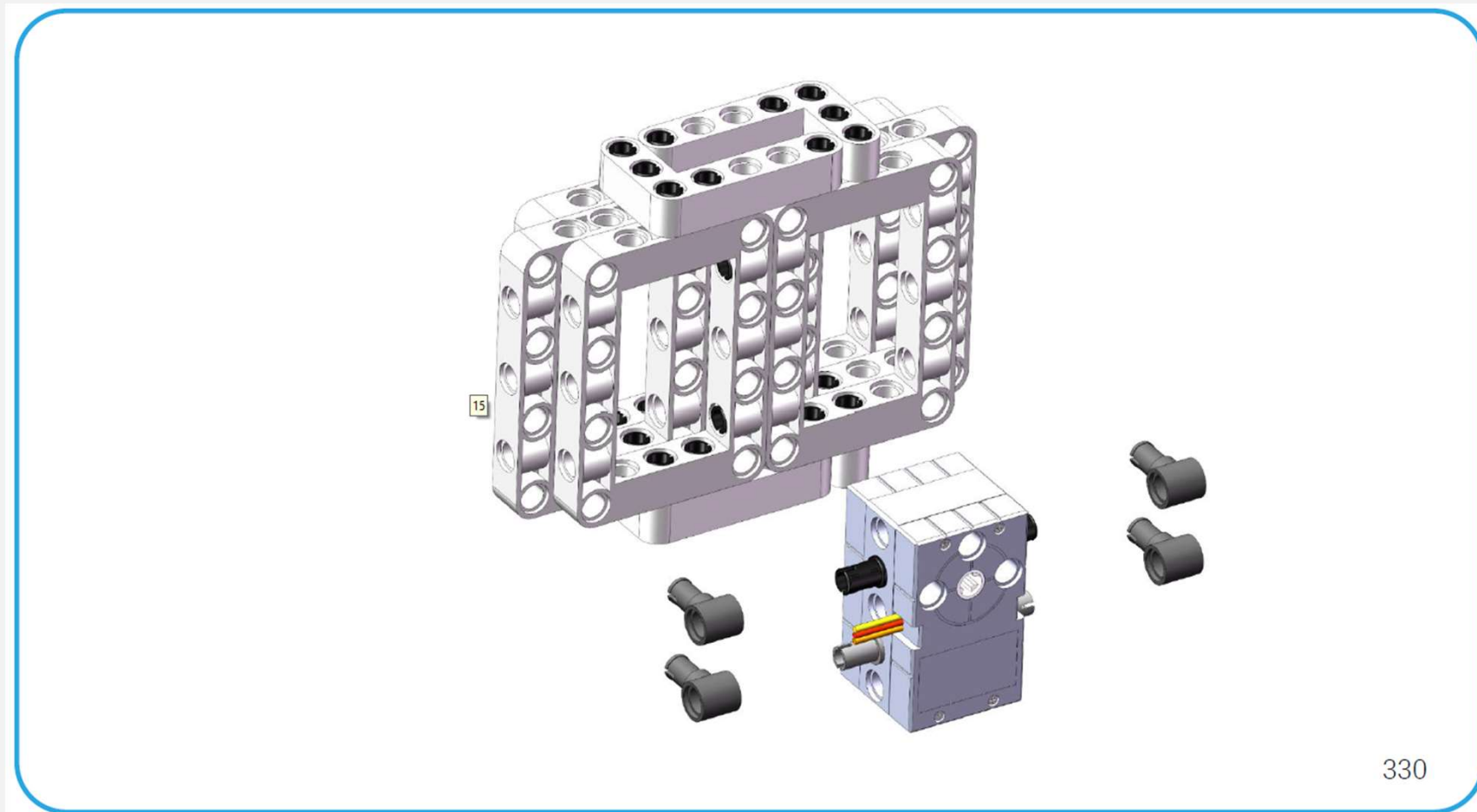


328

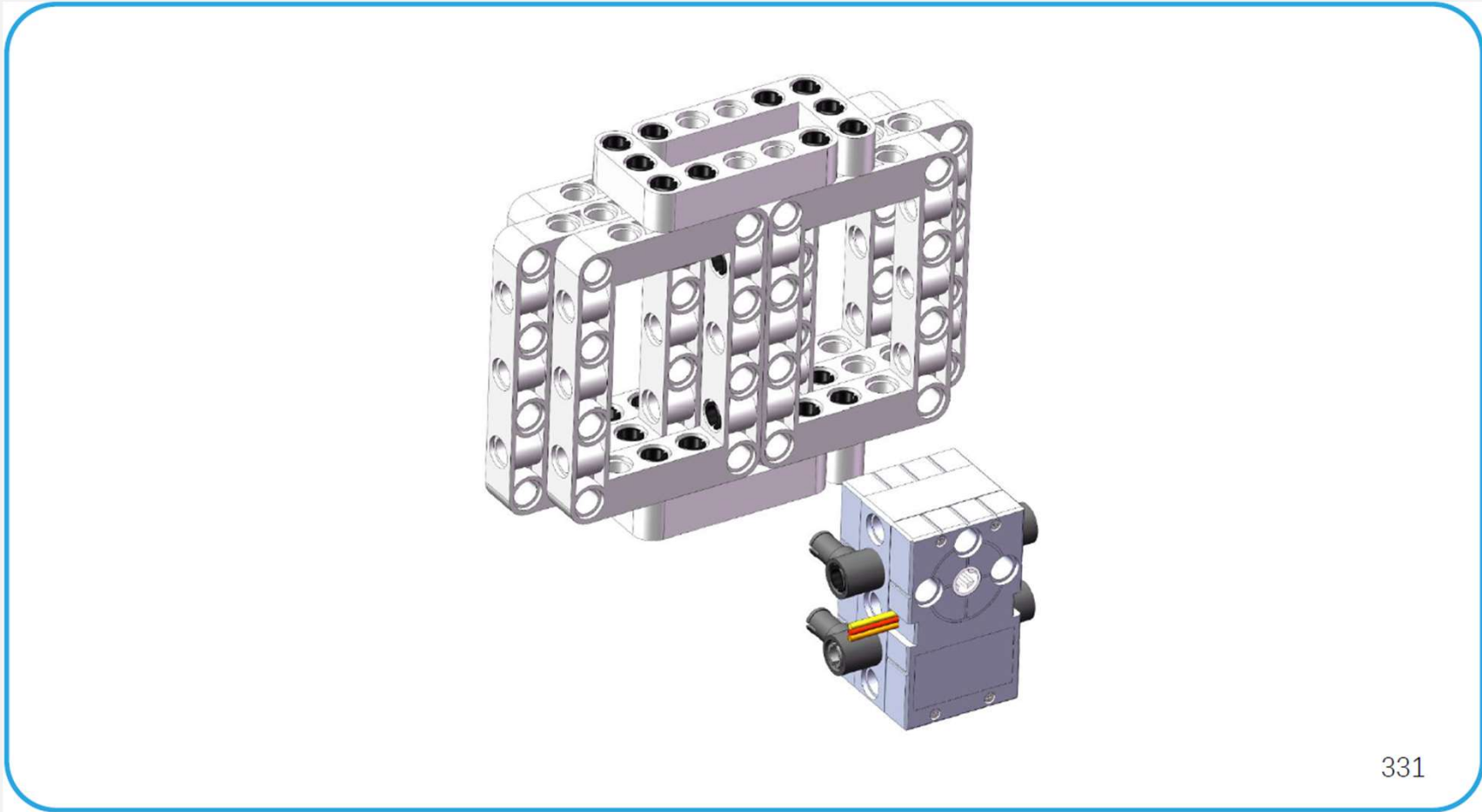
Step 14



Step 15

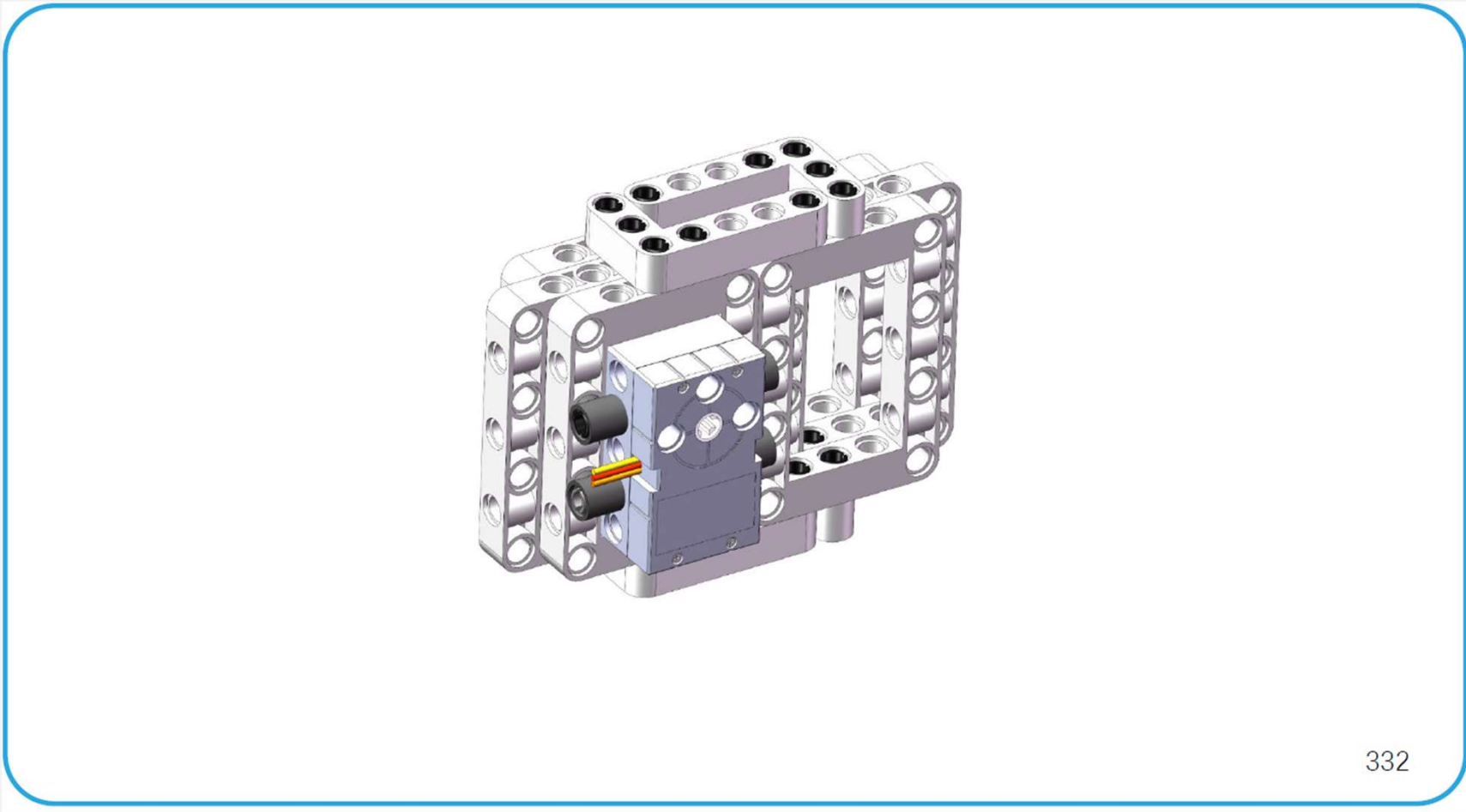


Step 16



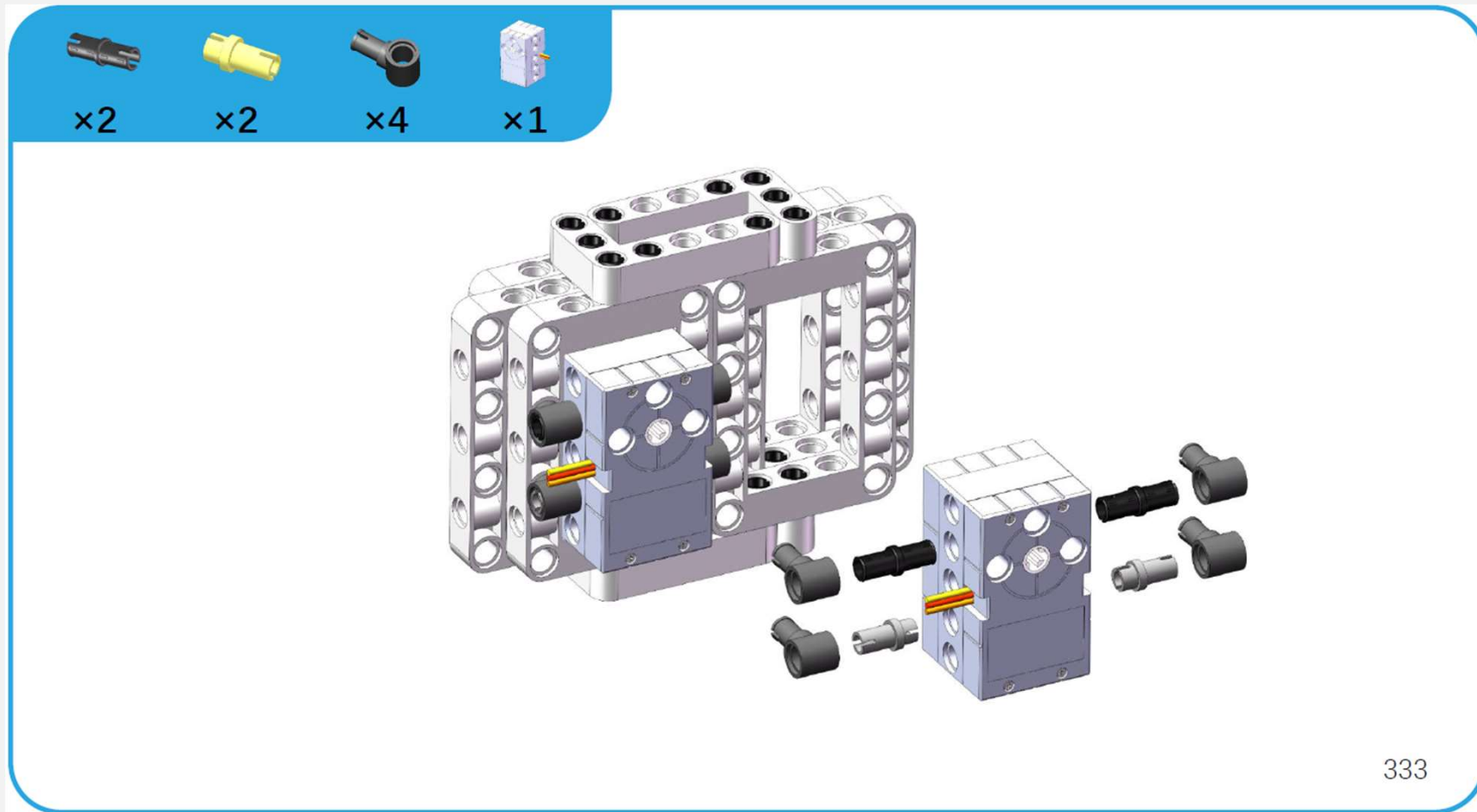
331

Step 17

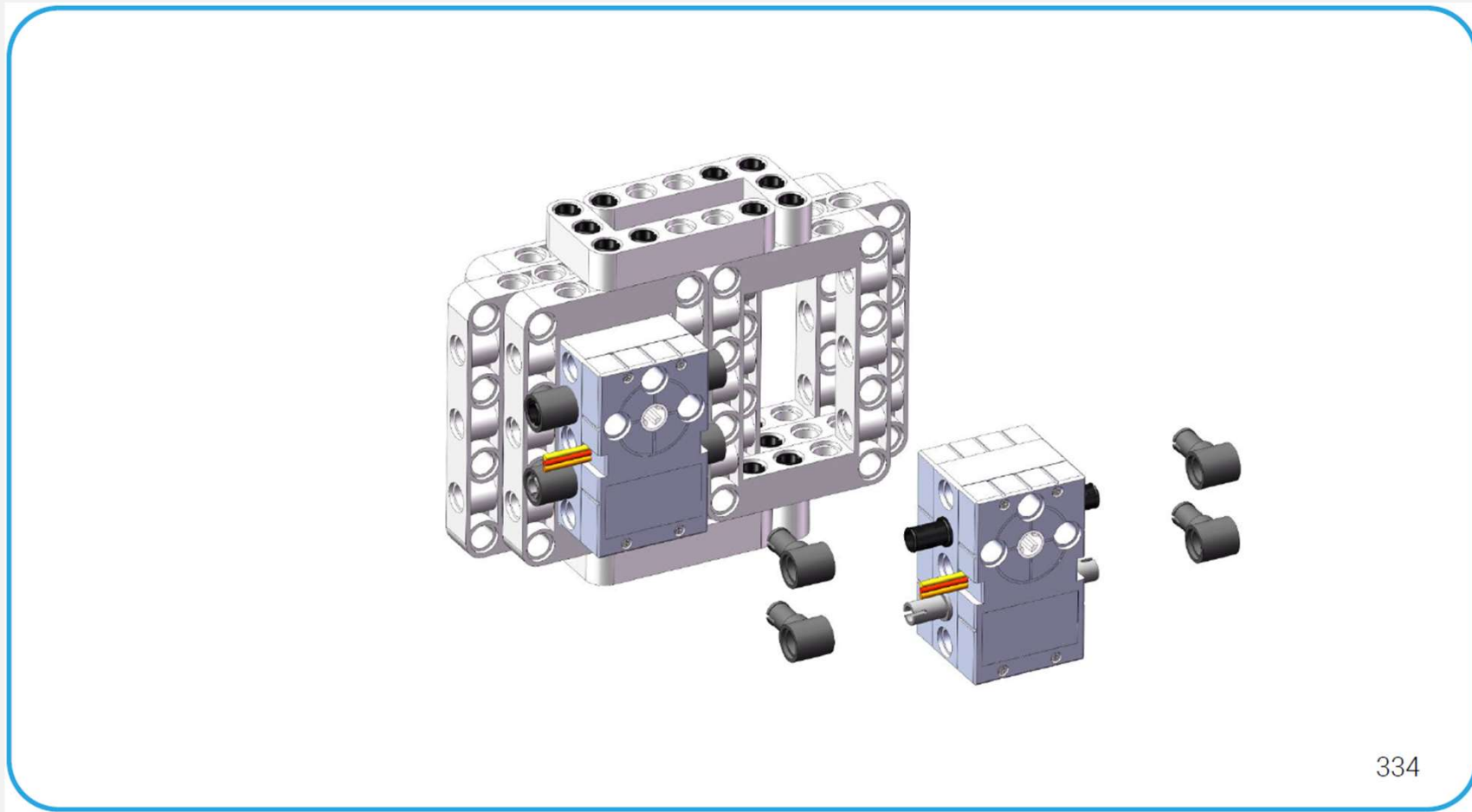


332

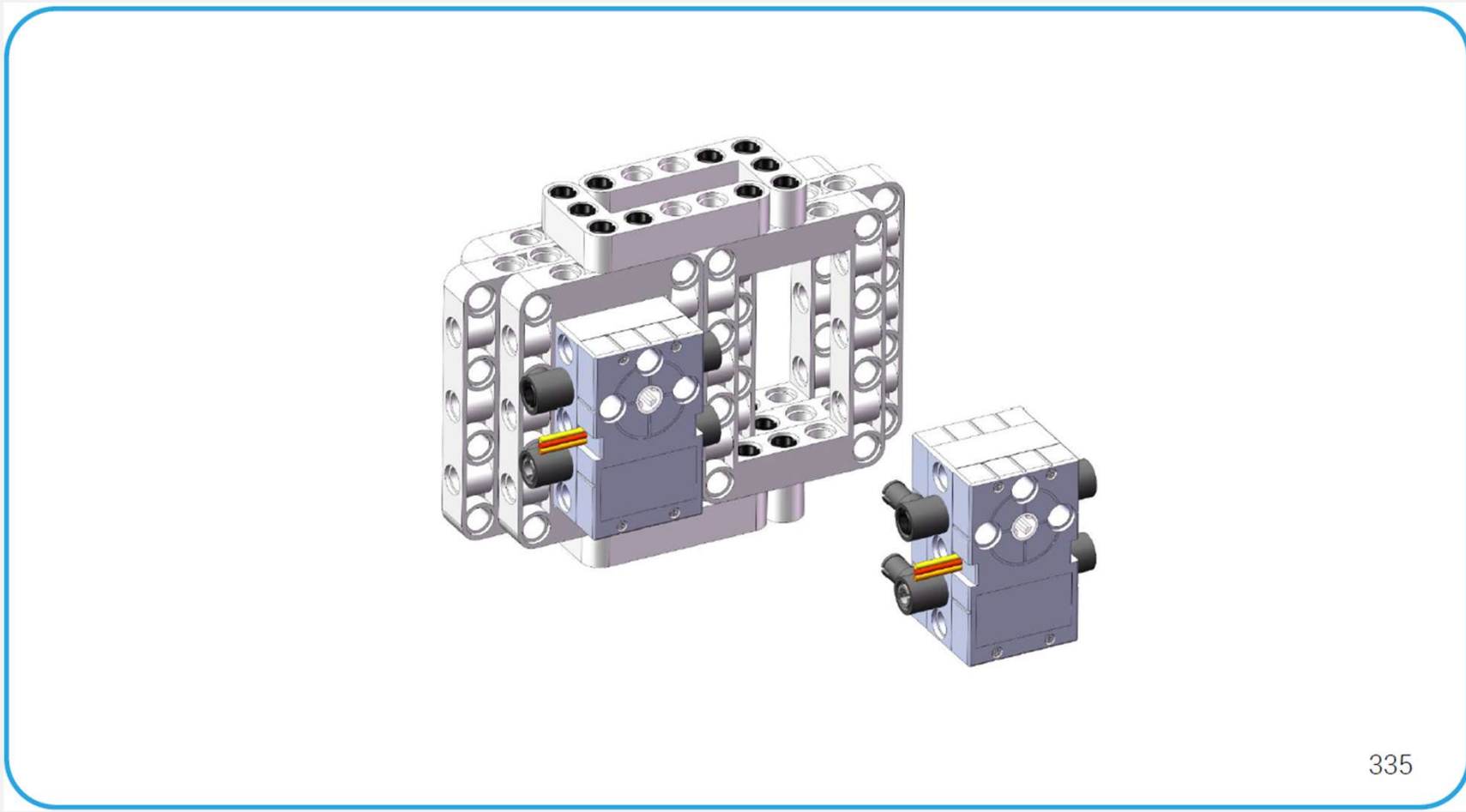
Step 18



Step 19

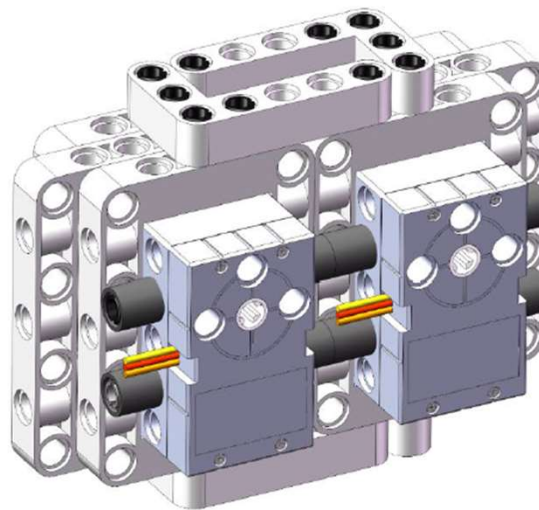


Step 20



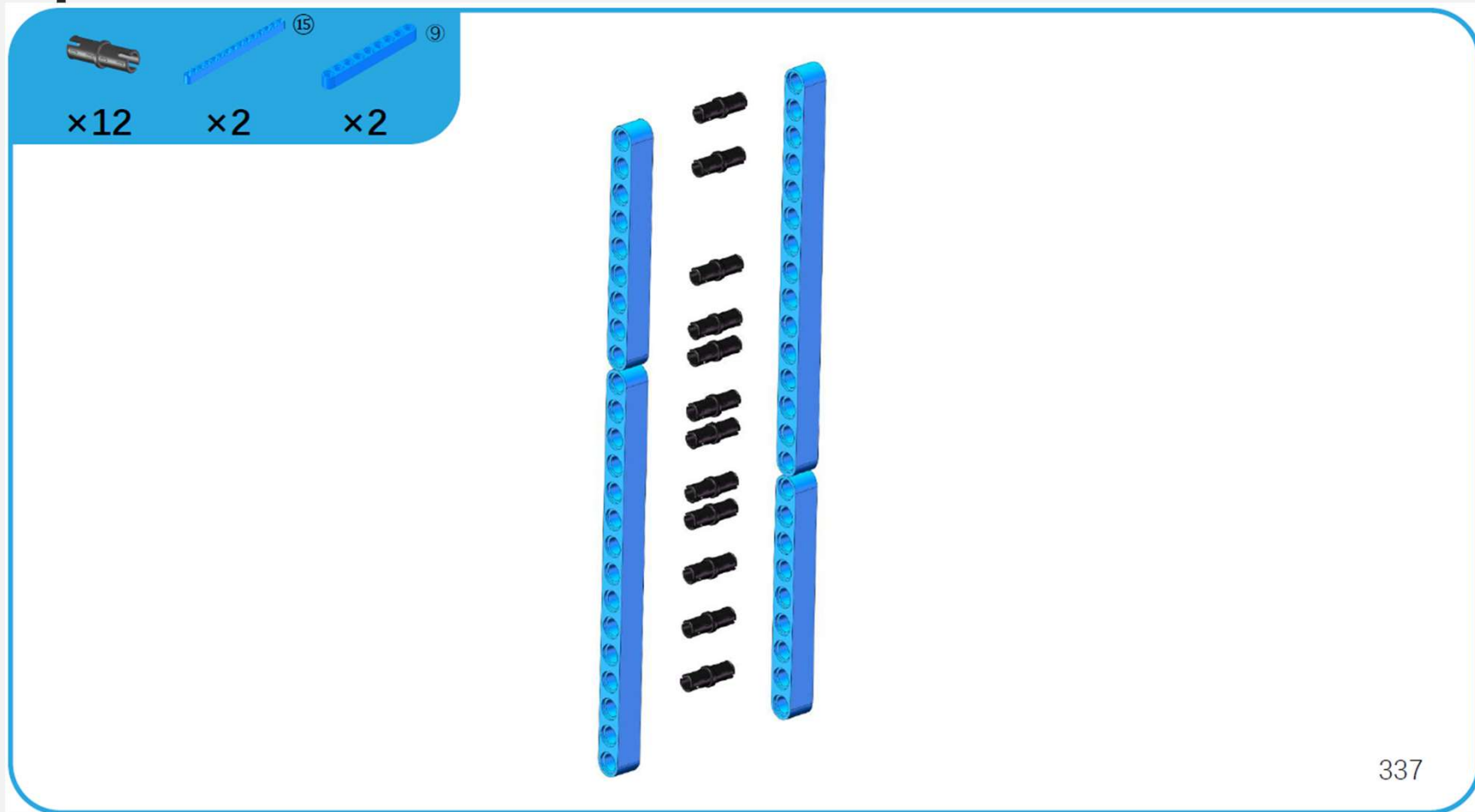
335

Step 21



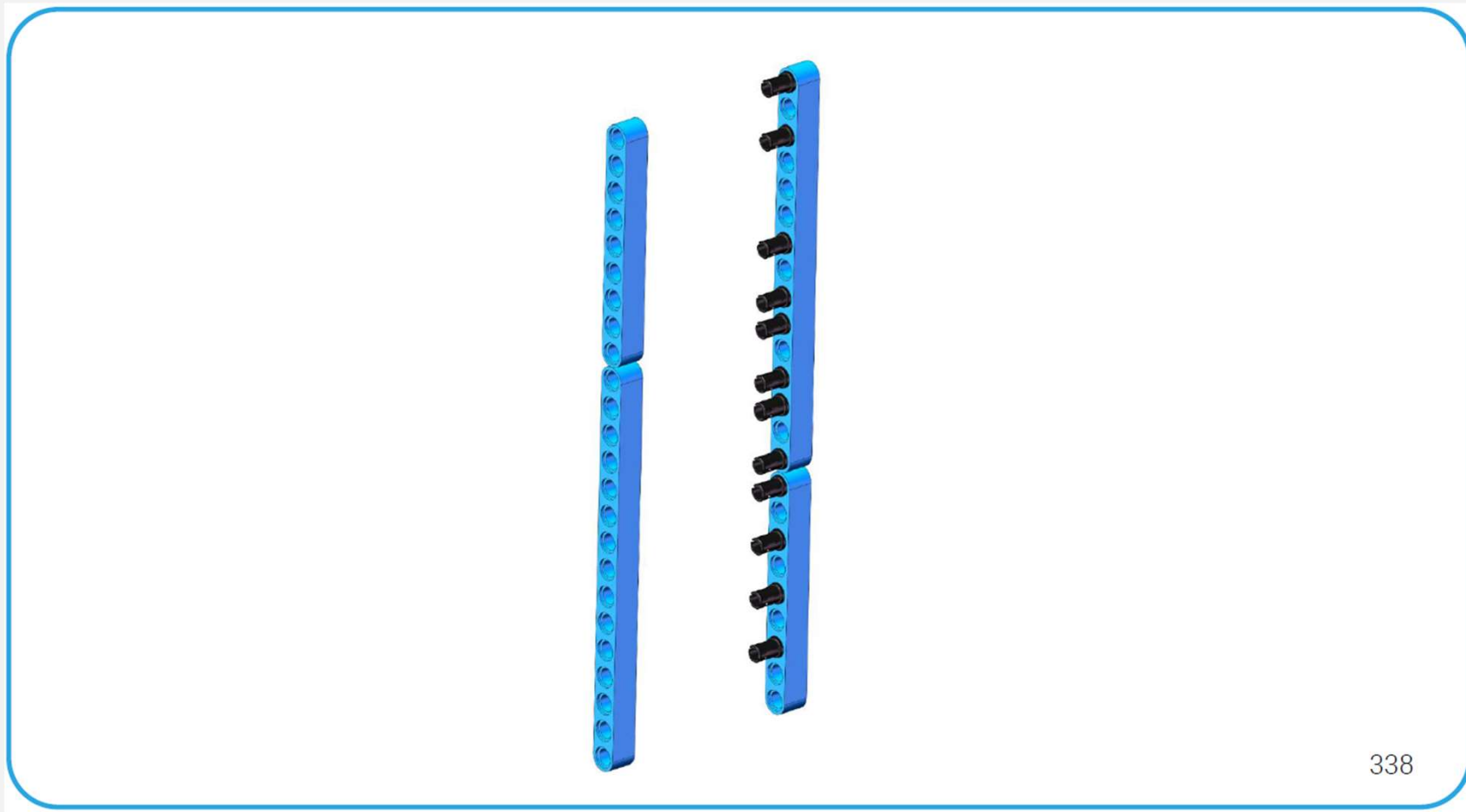
336

Step 22



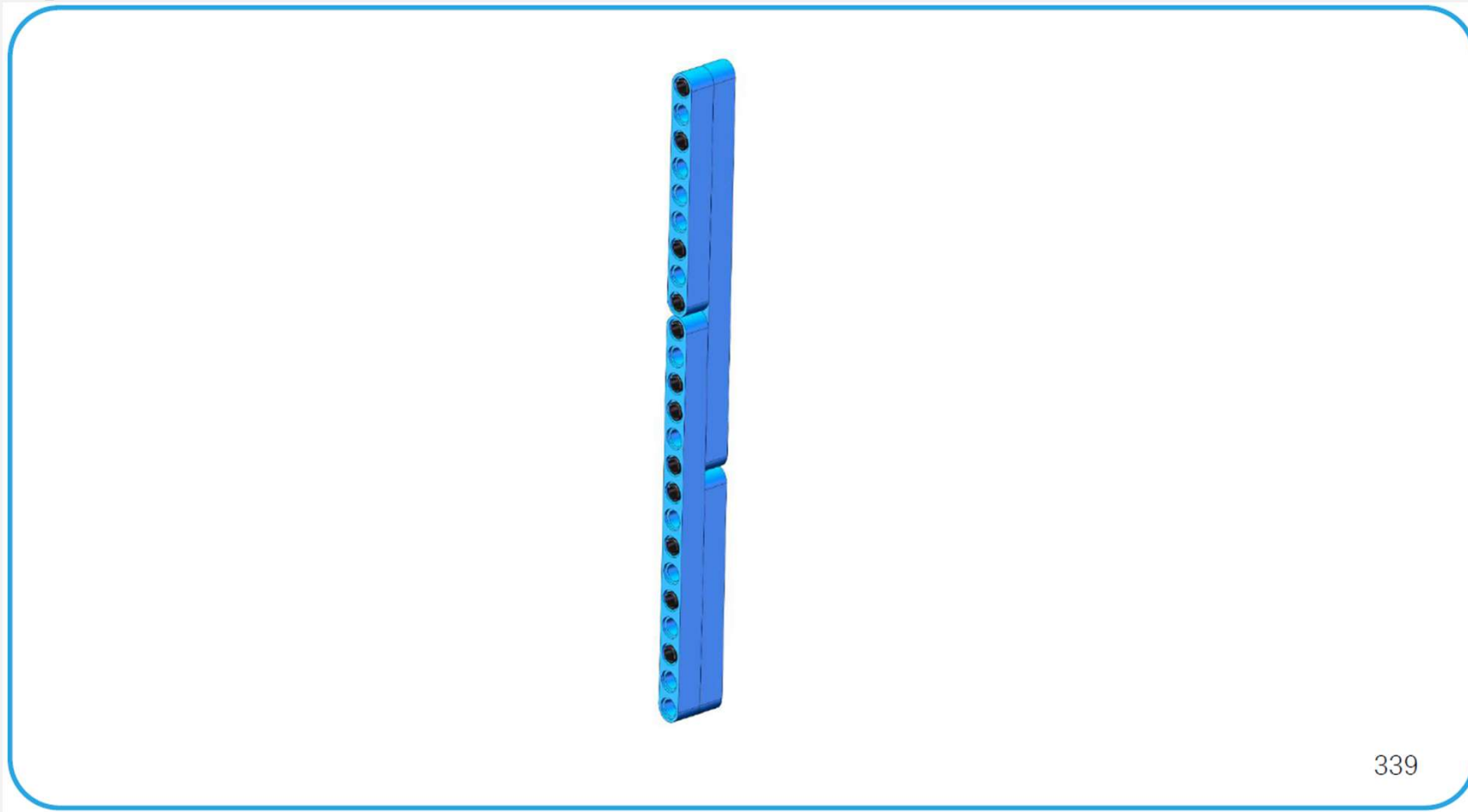
337

Step 23



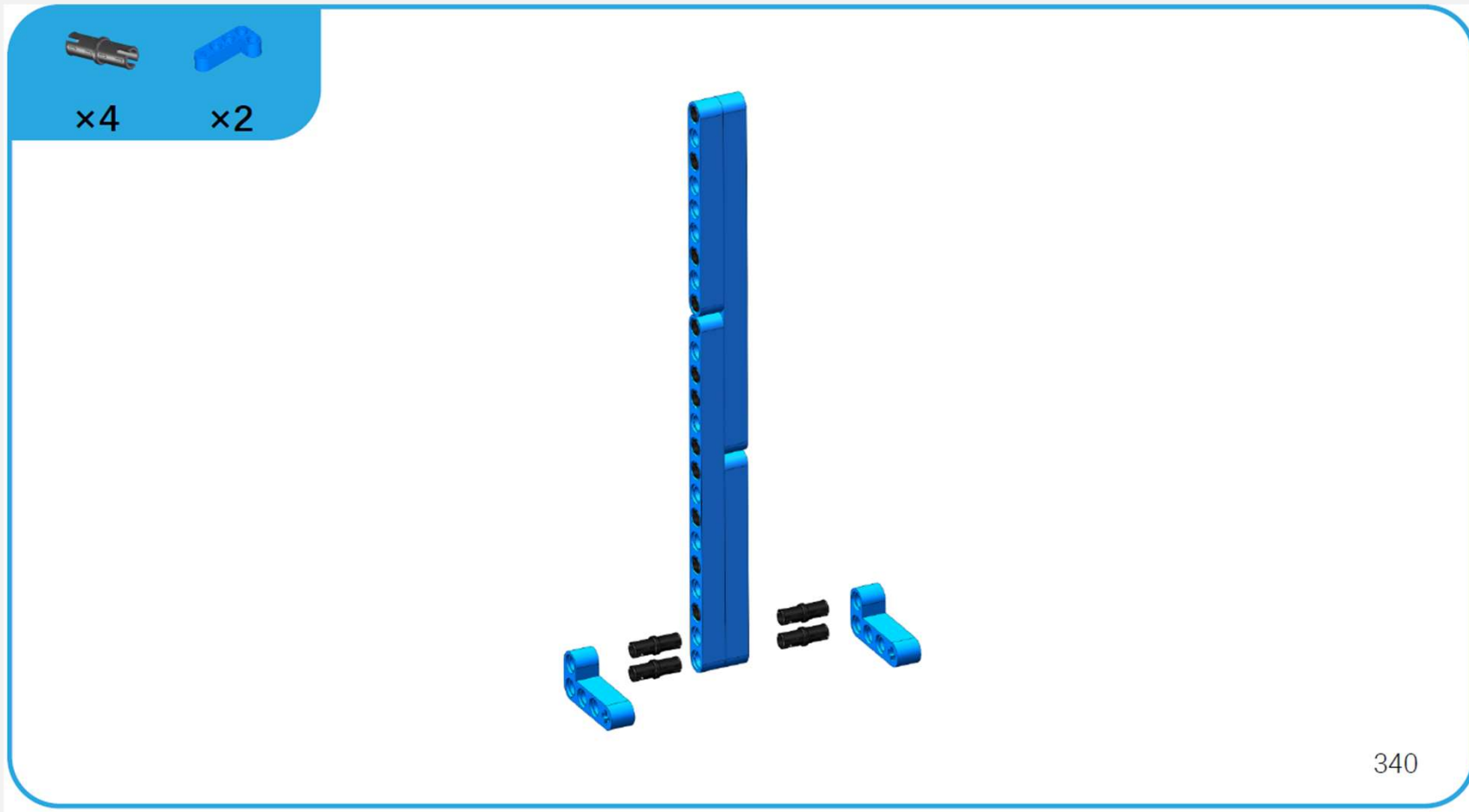
338

Step 24

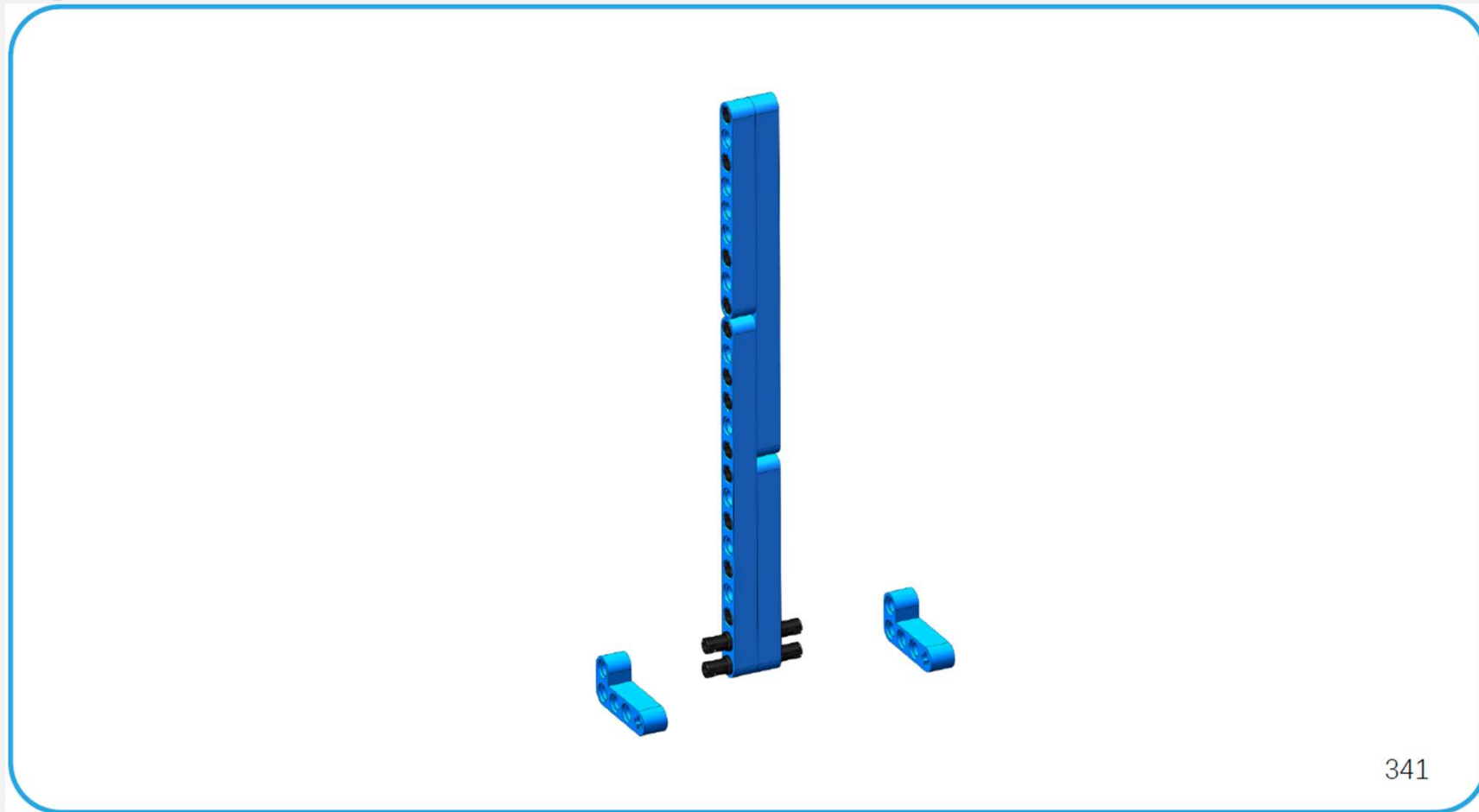


339

Step 25

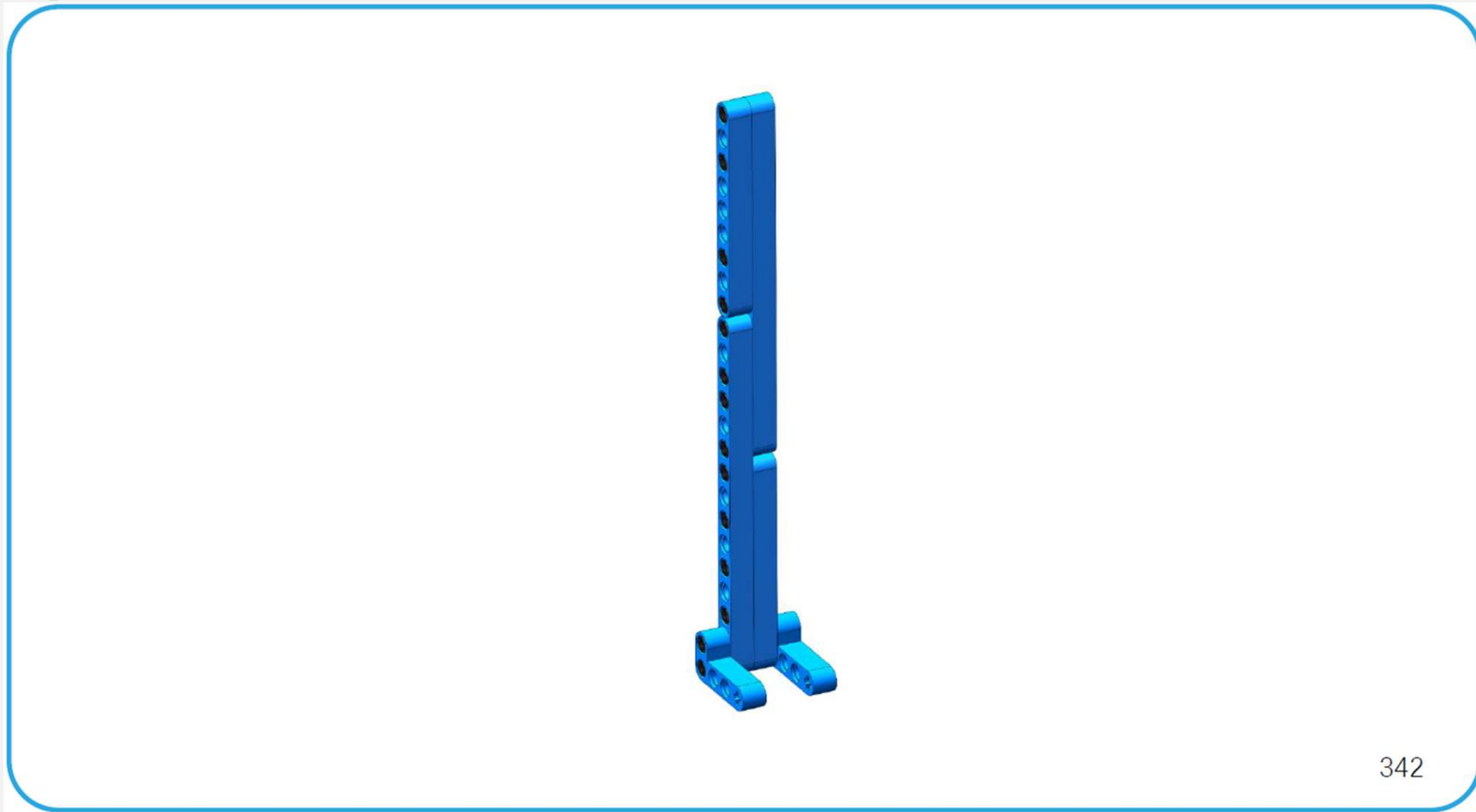


Step 26



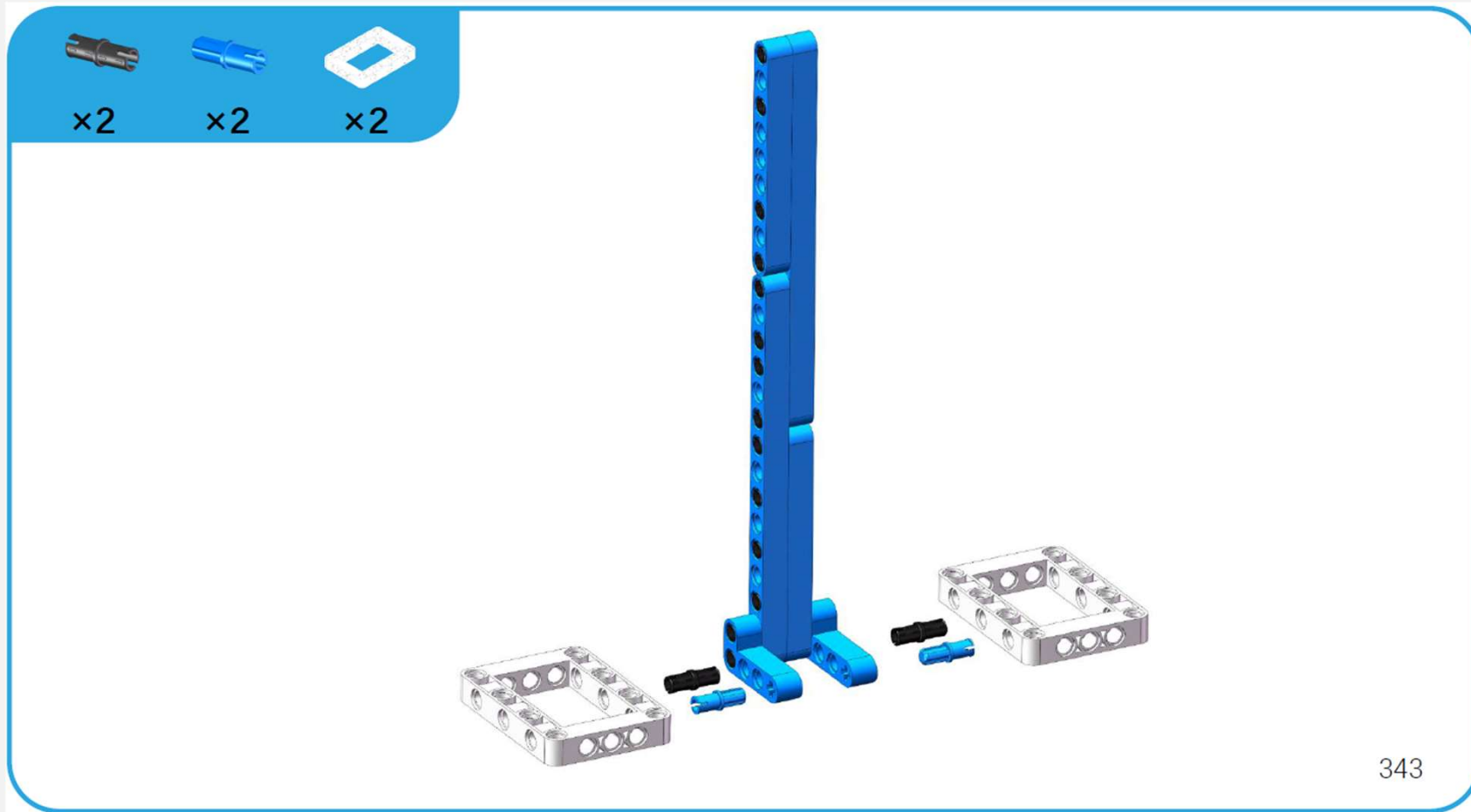
341

Step 27

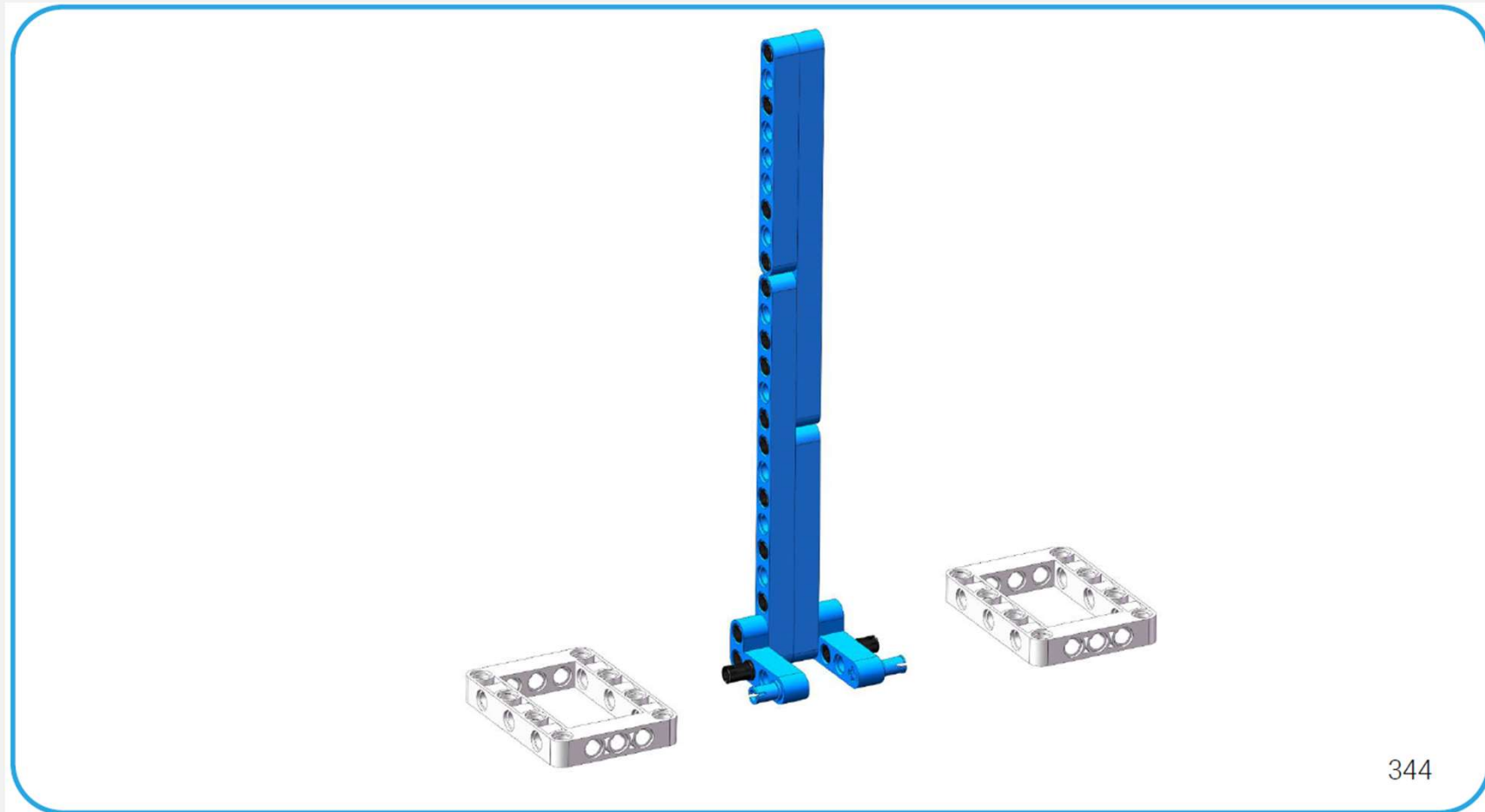


342

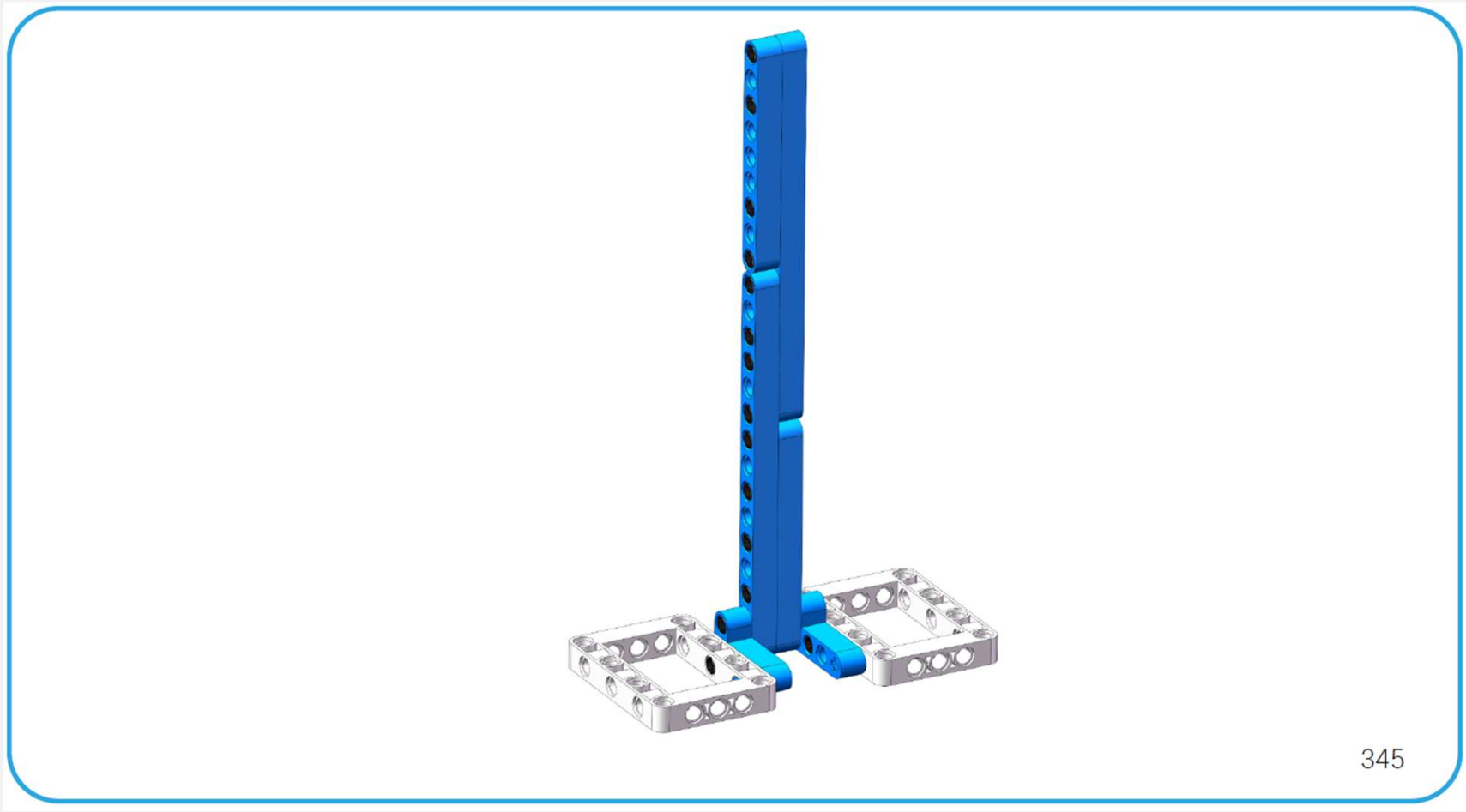
Step 28



Step 29

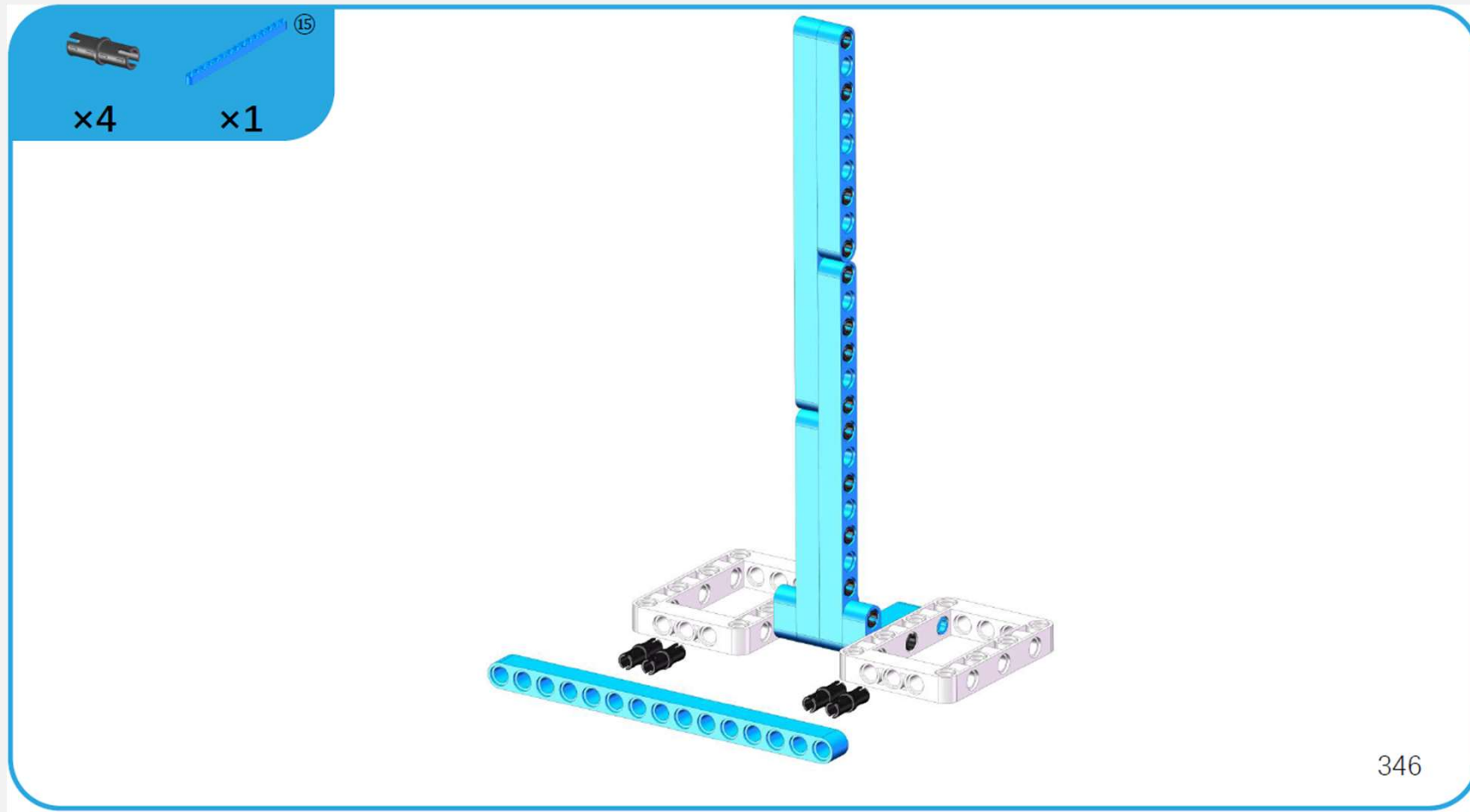


Step 30

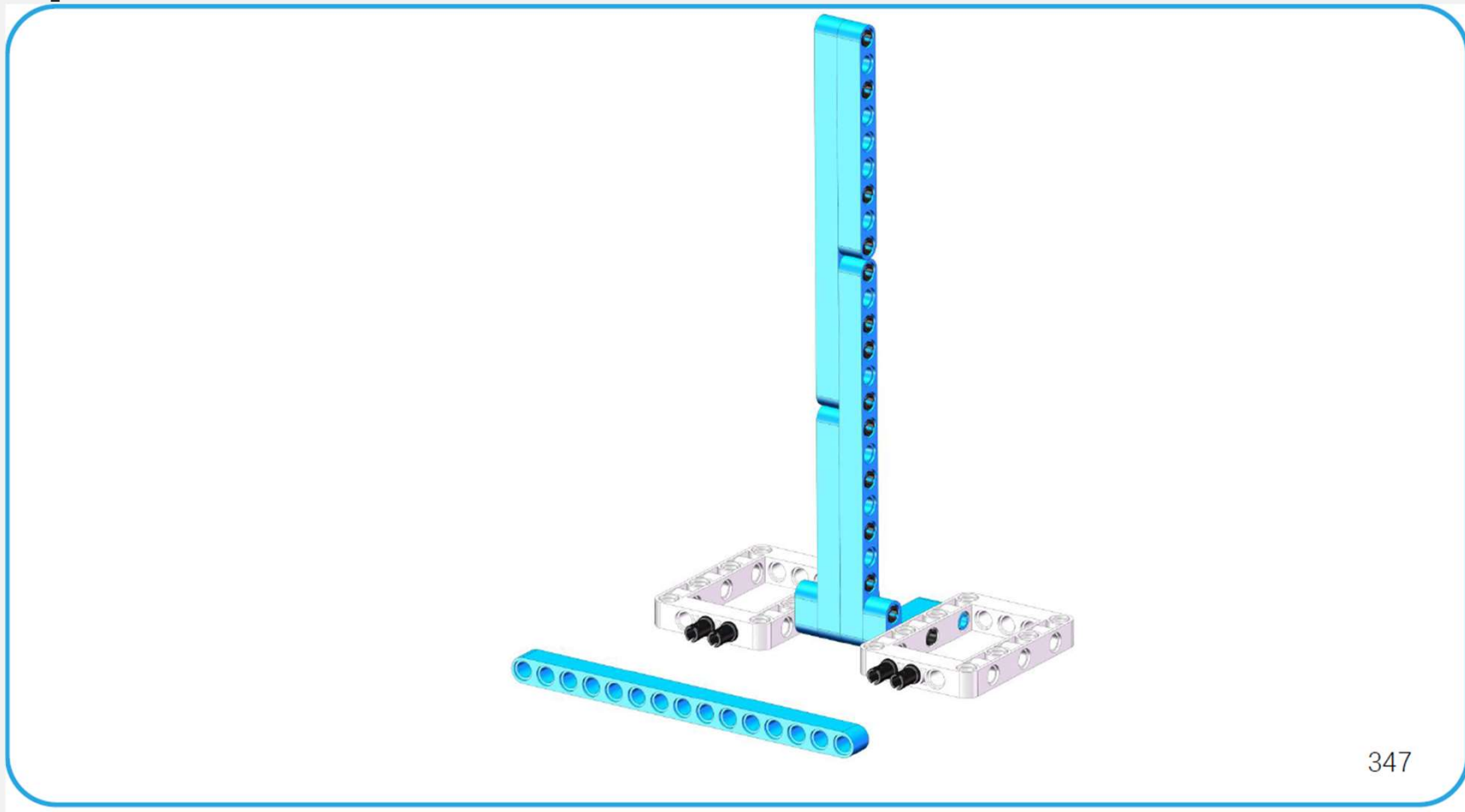


345

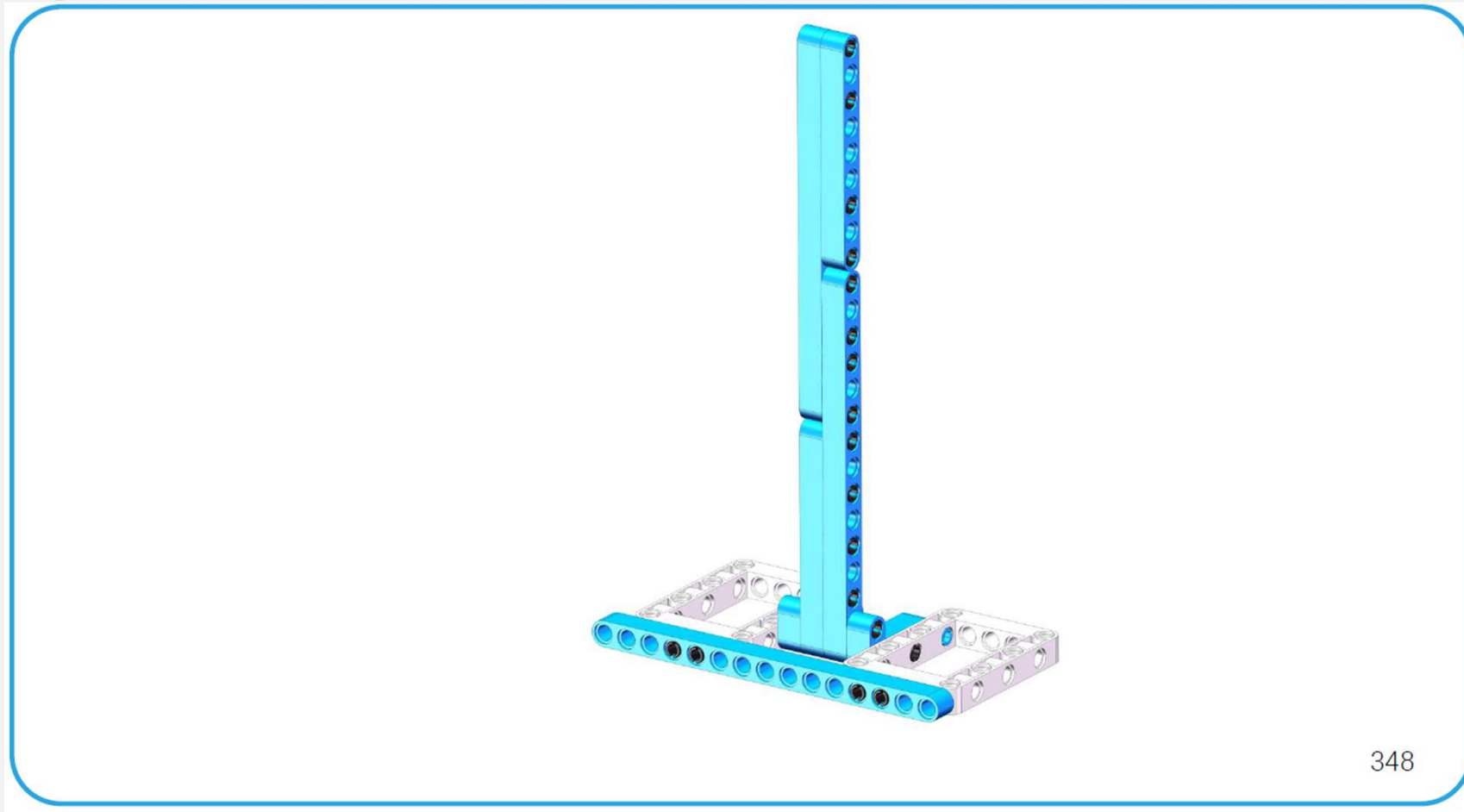
Step 31



Step 32

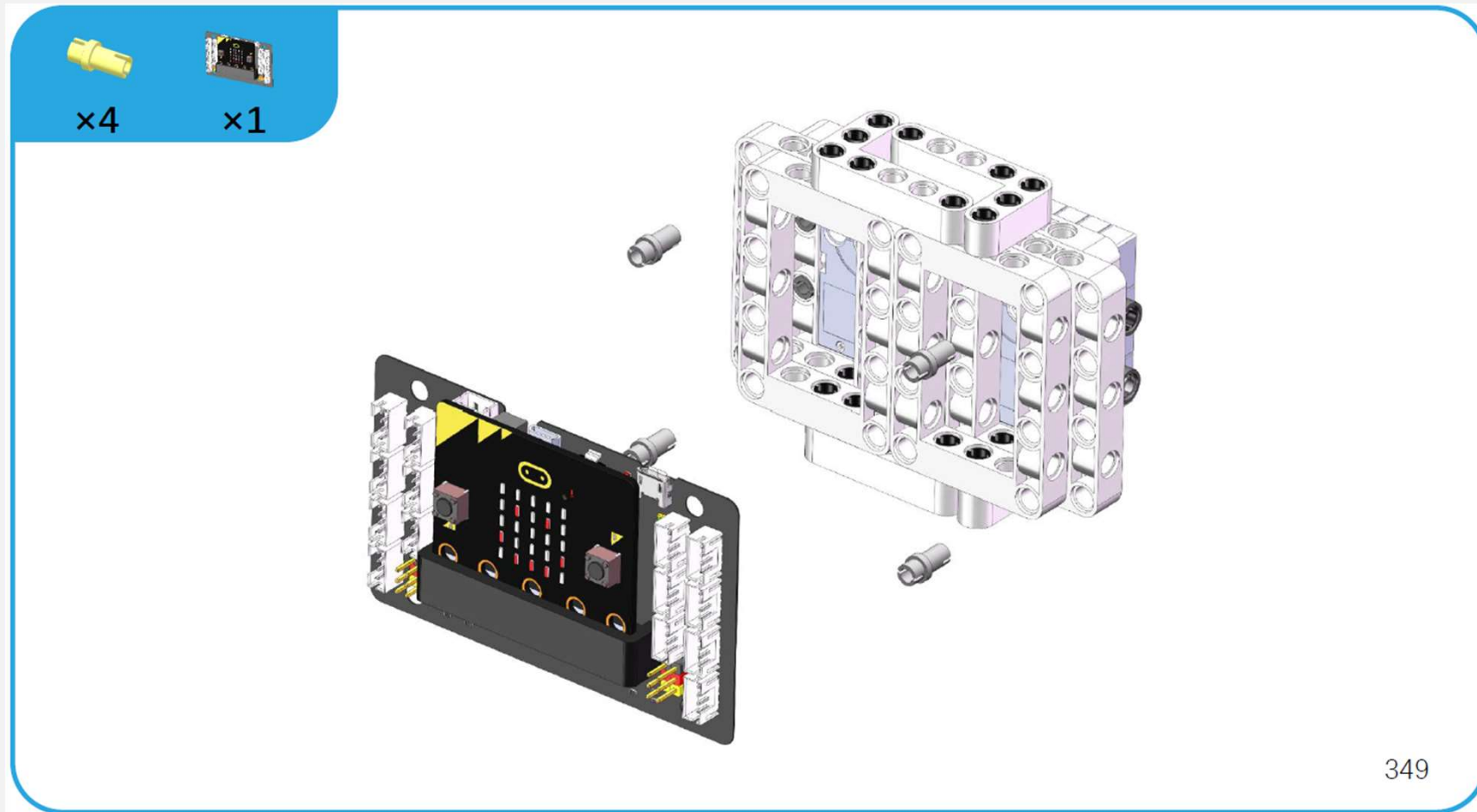


Step 33



348

Step 34



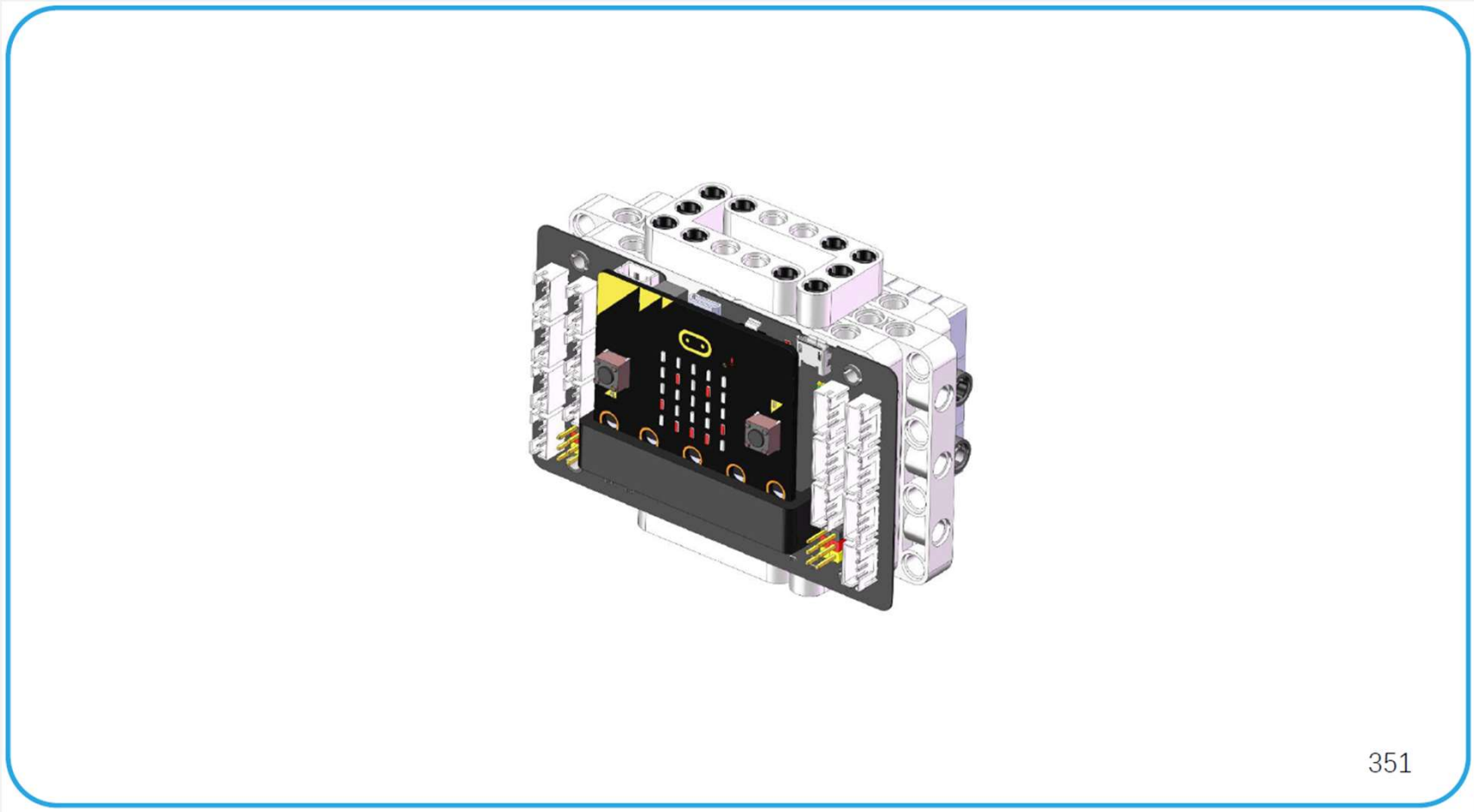
349

Step 35



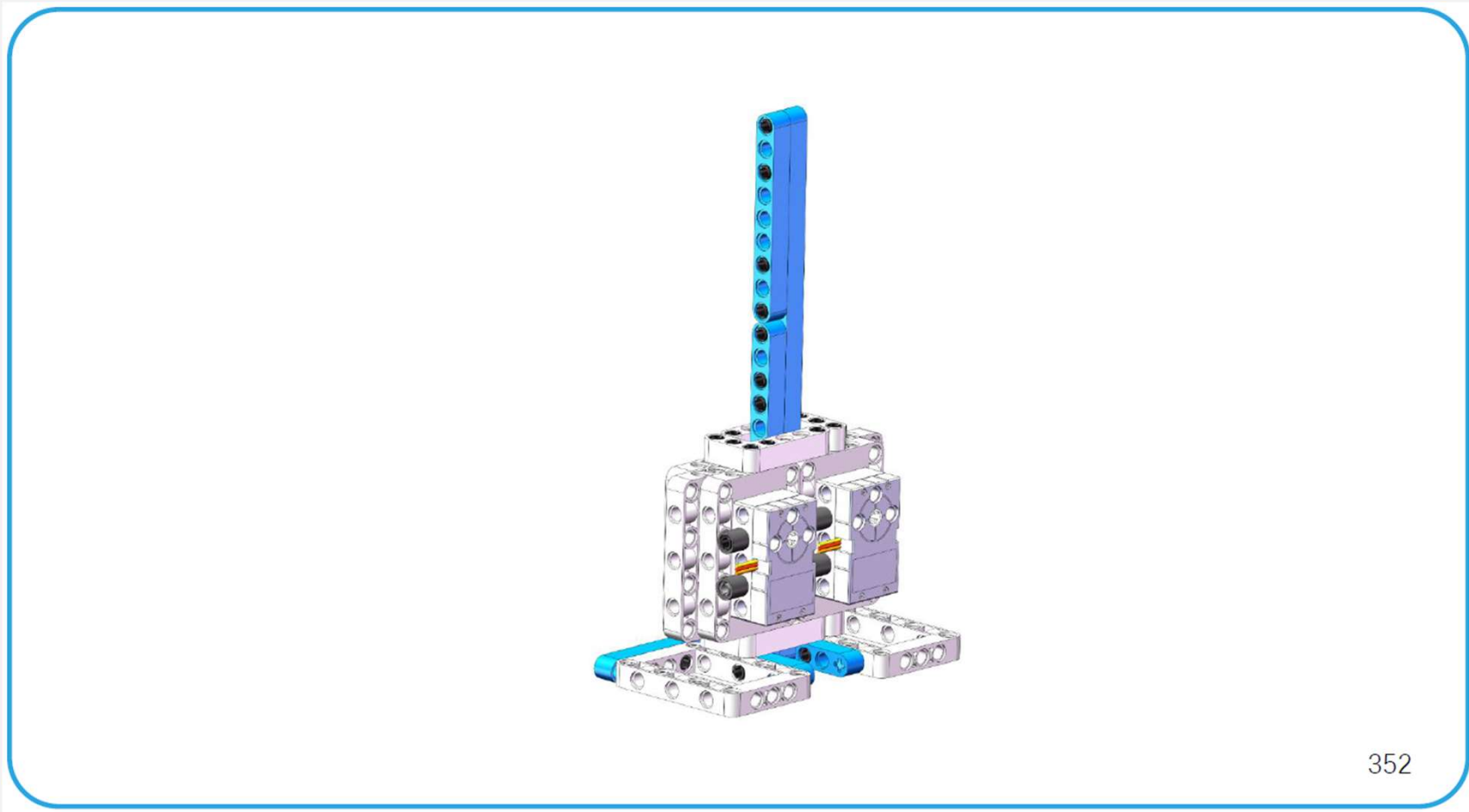
350

Step 36



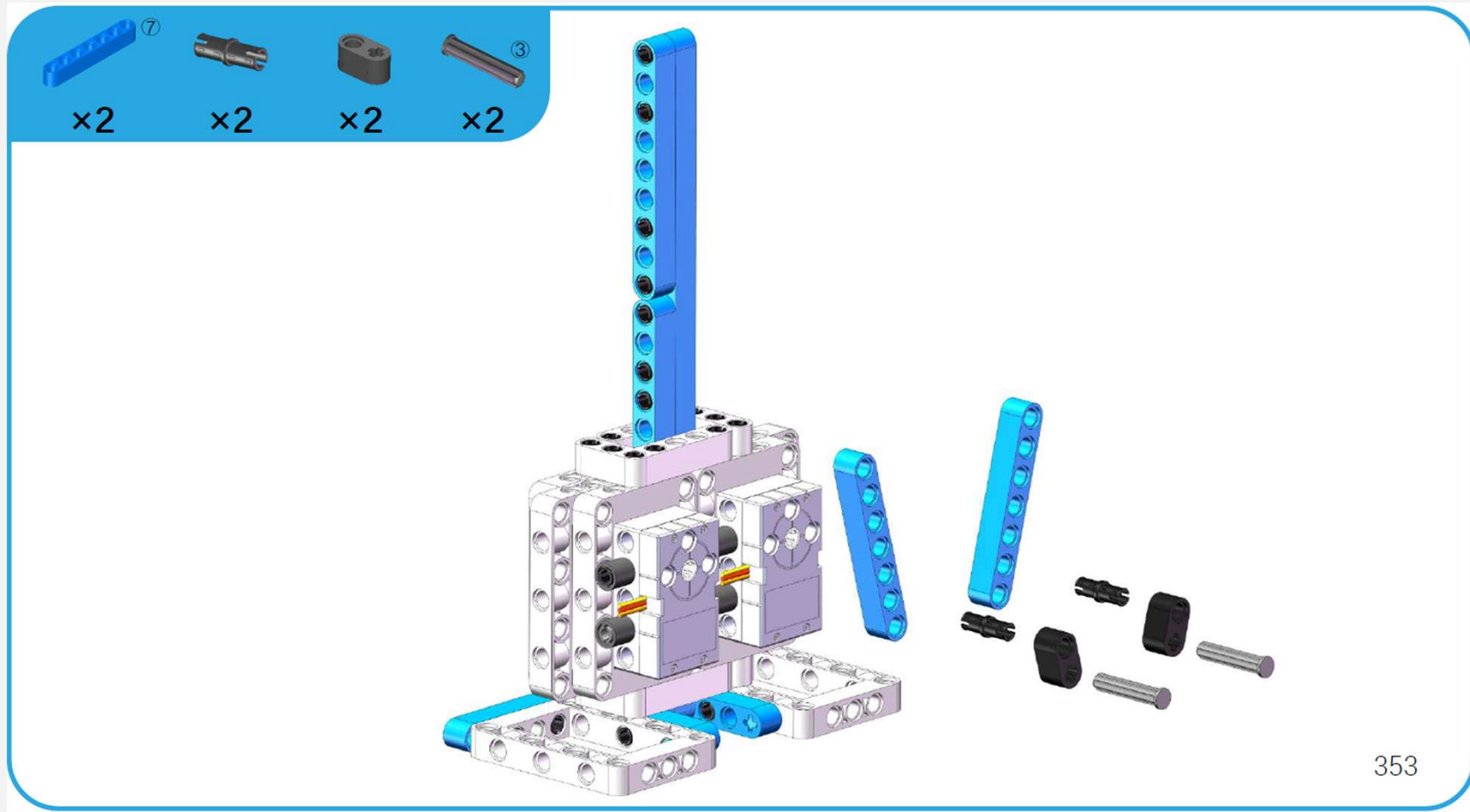
351

Step 37

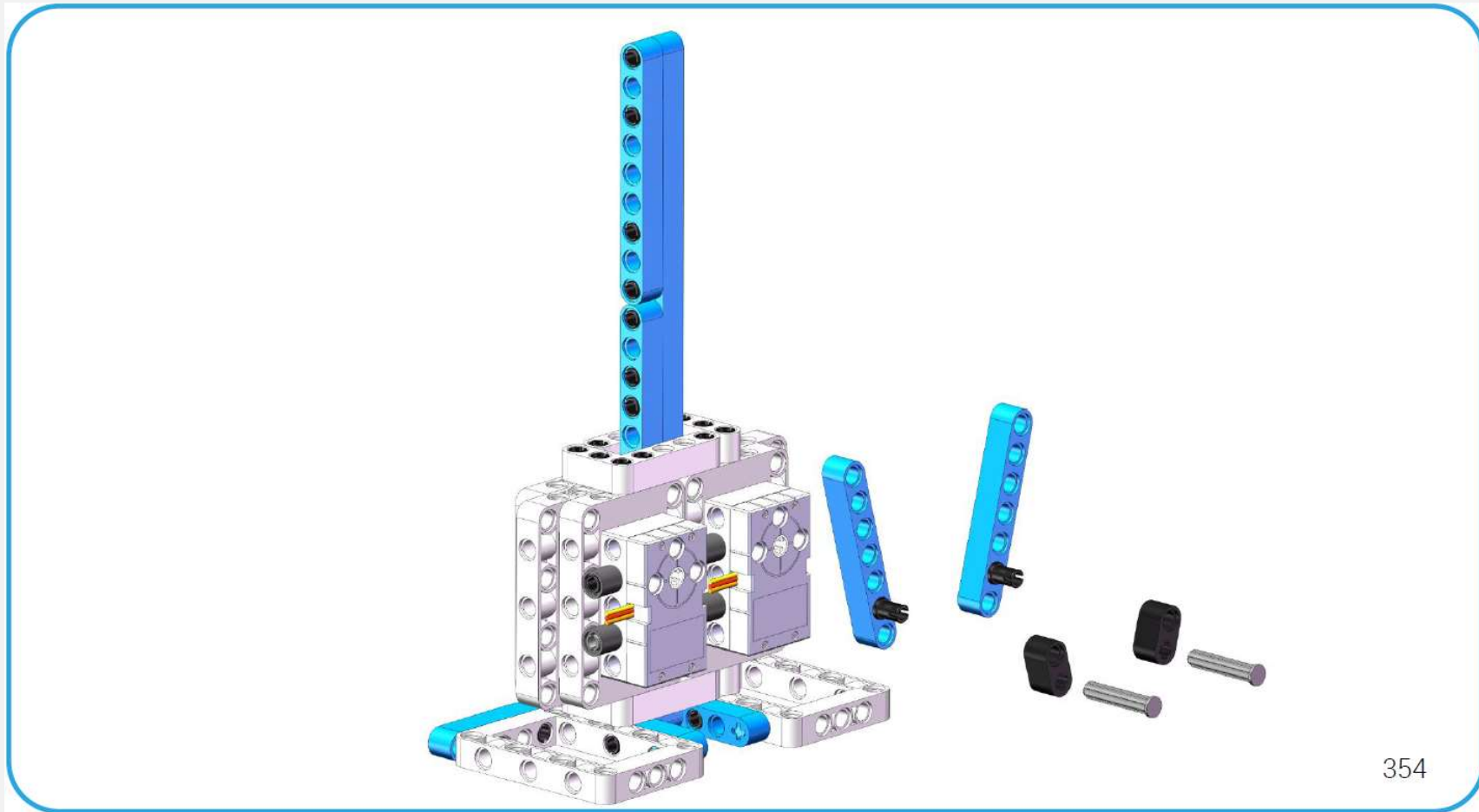


352

Step 38

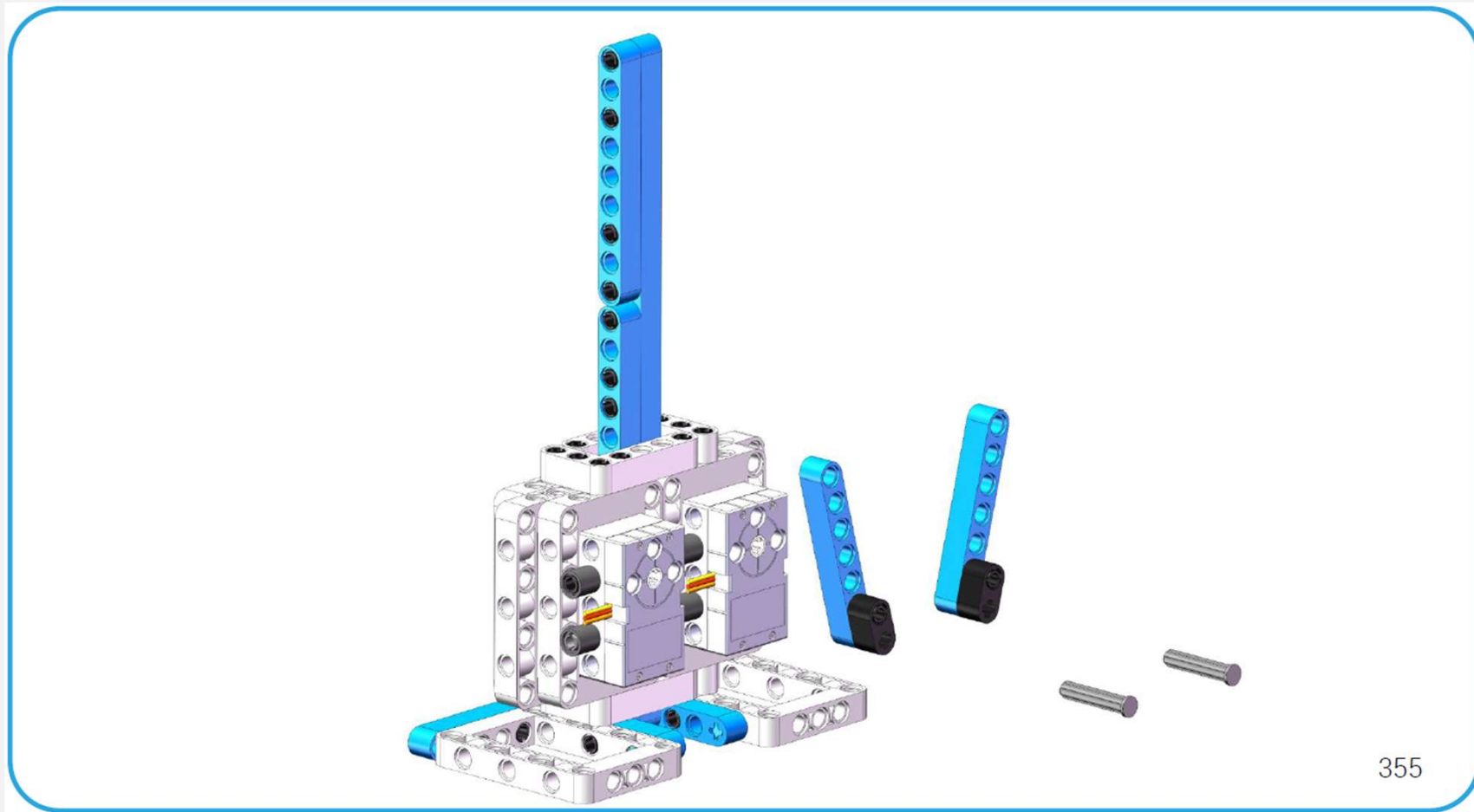


Step 39

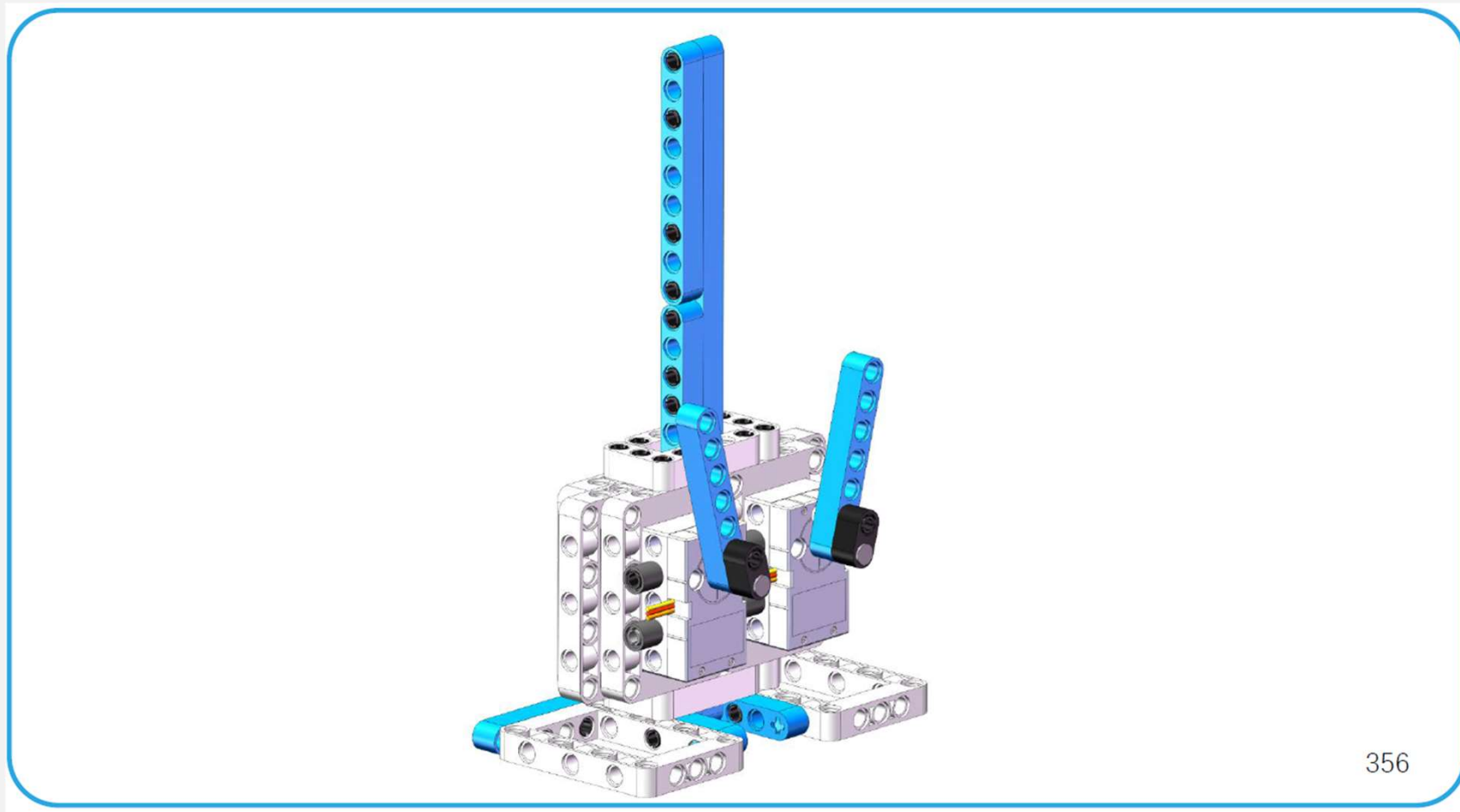


354

Step 40

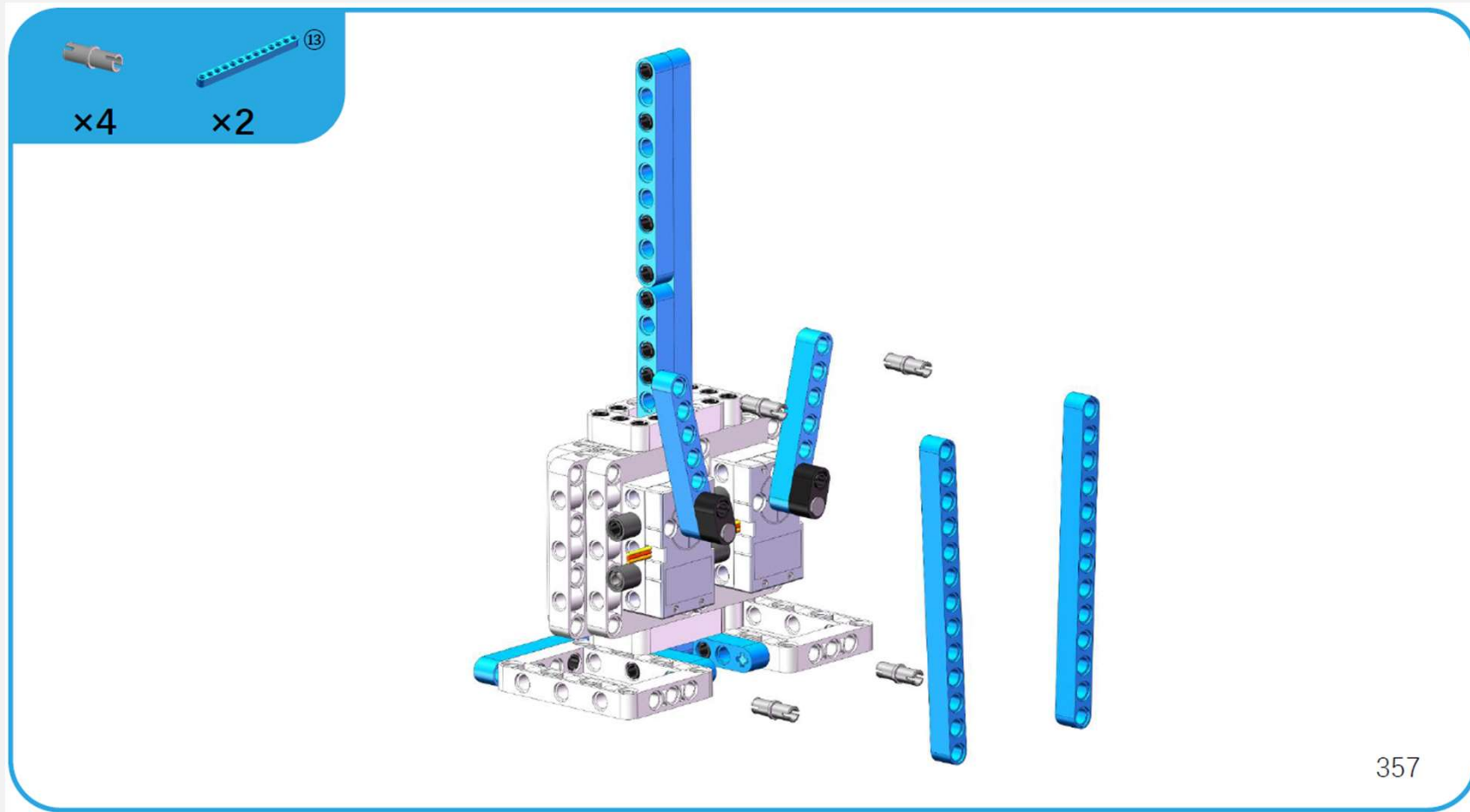


Step 41

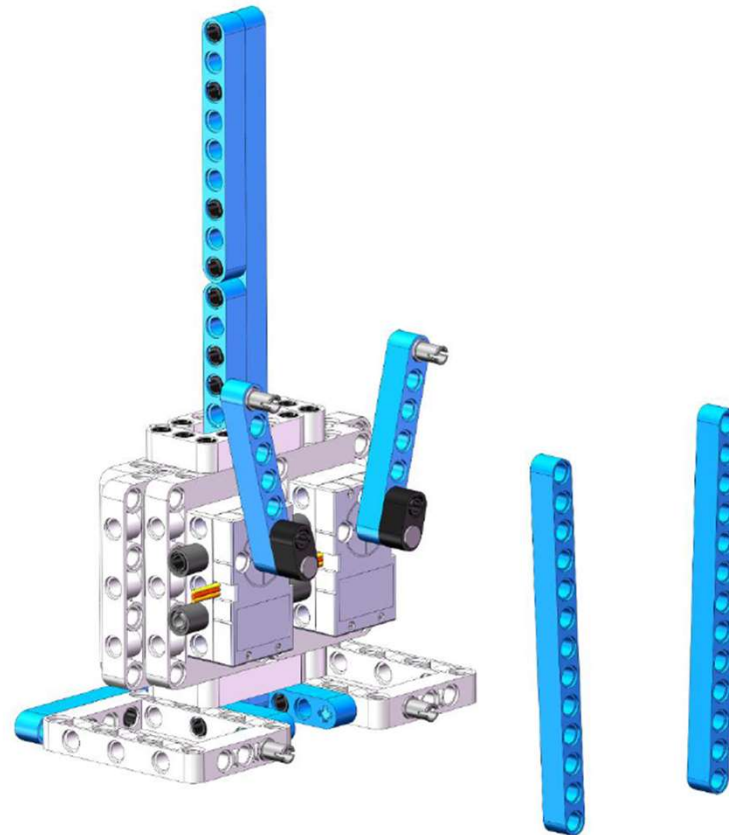


356

Step 42

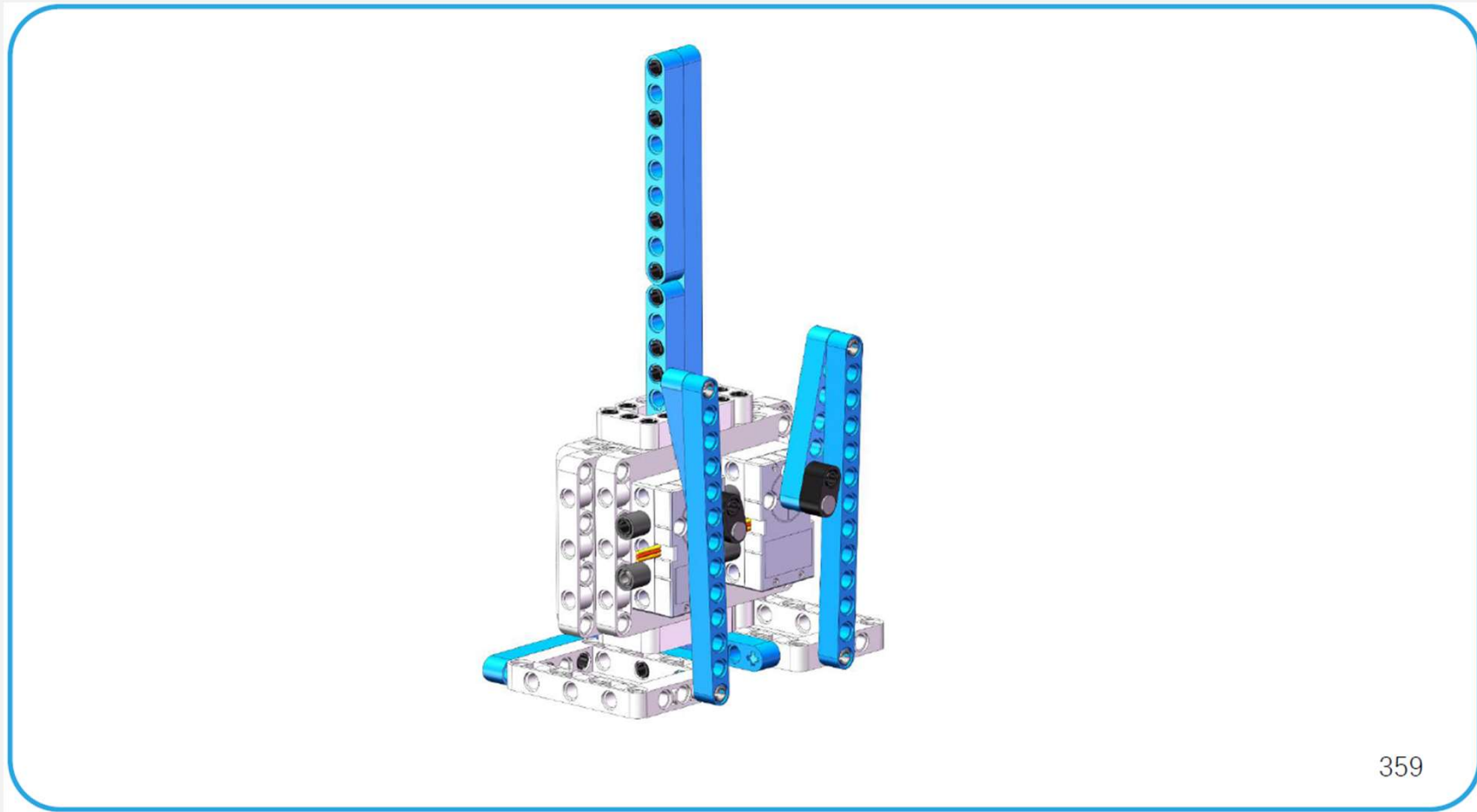


Step 43



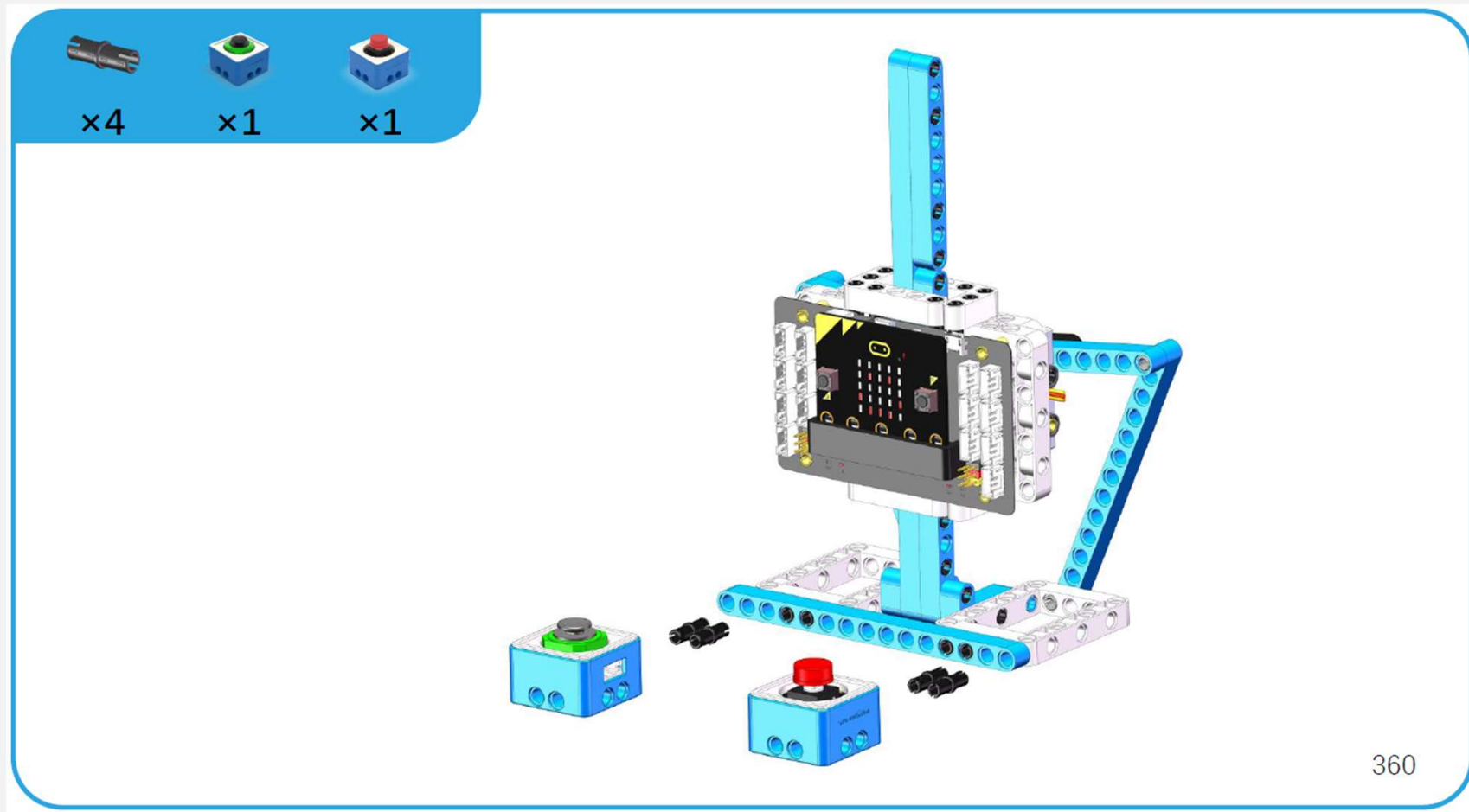
358

Step 44

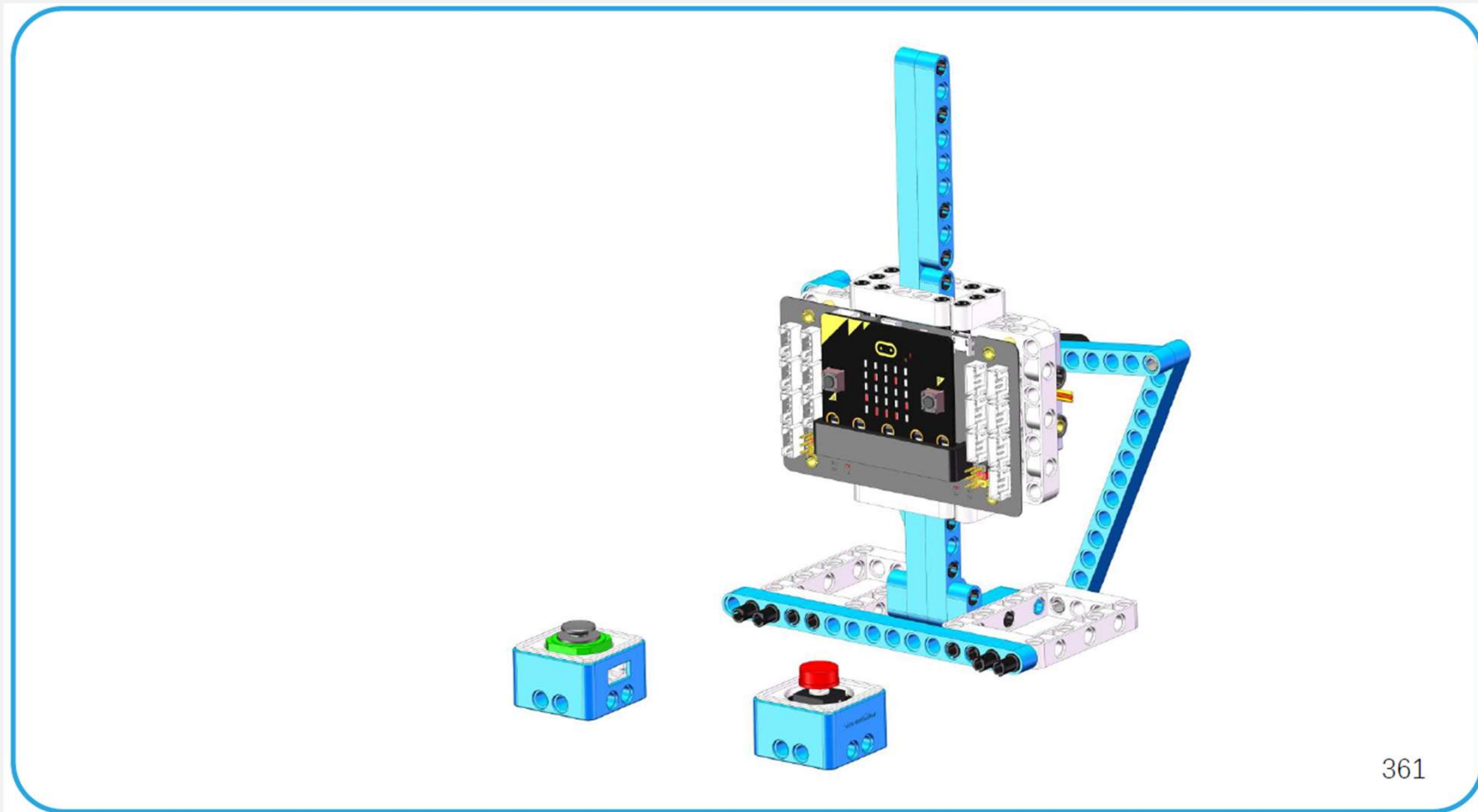


359

Step 45

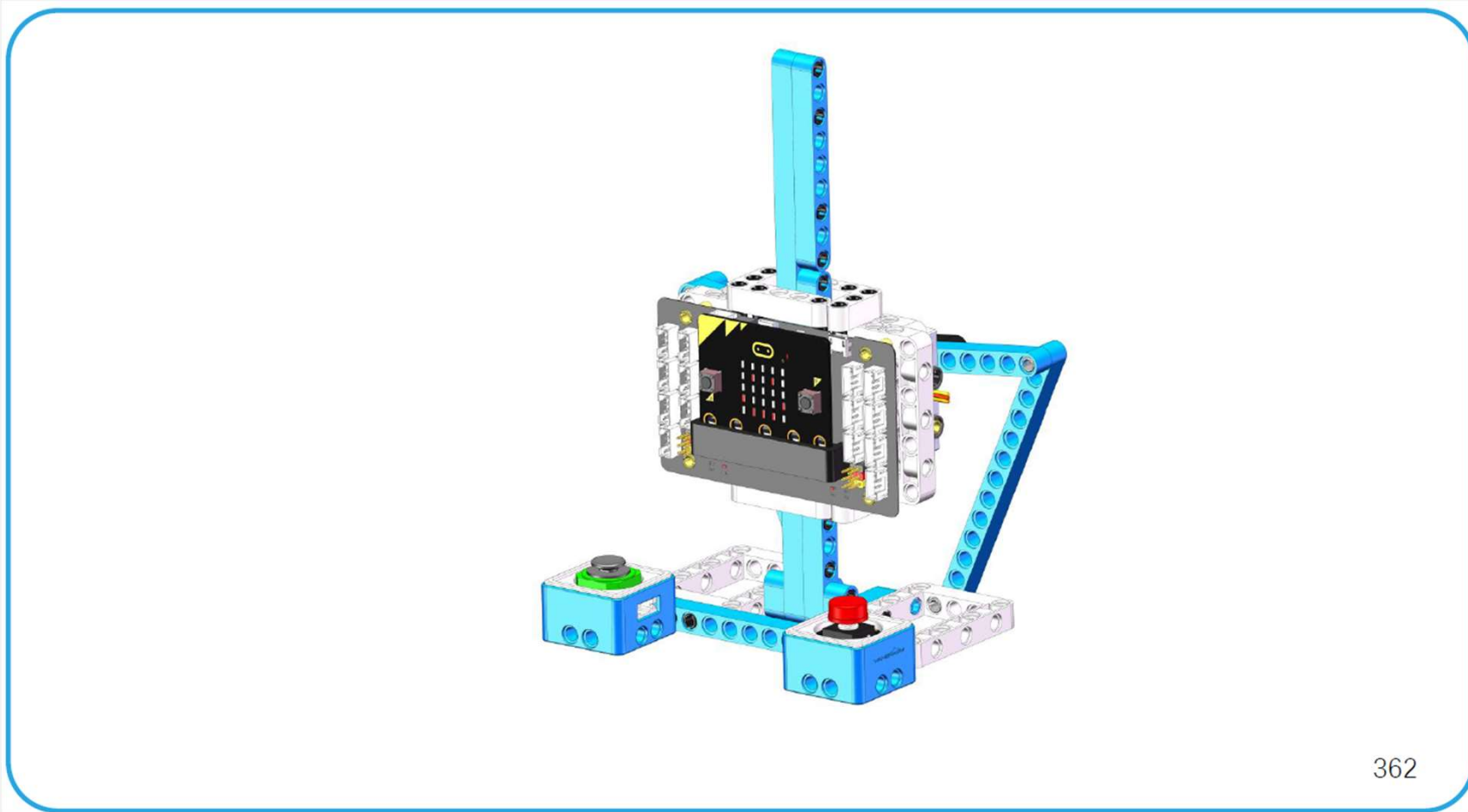


Step 46



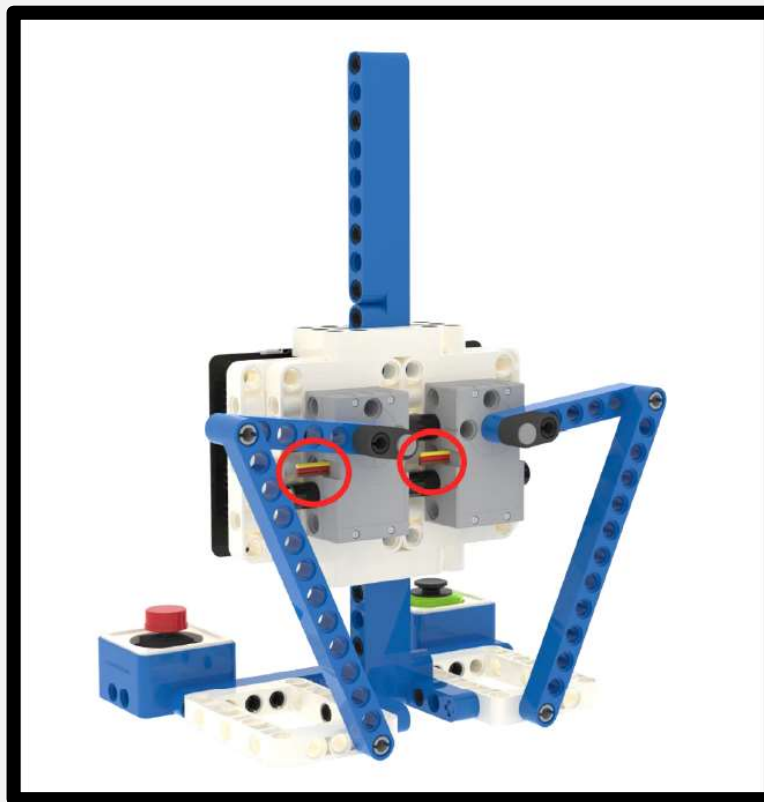
361

Step 47



362

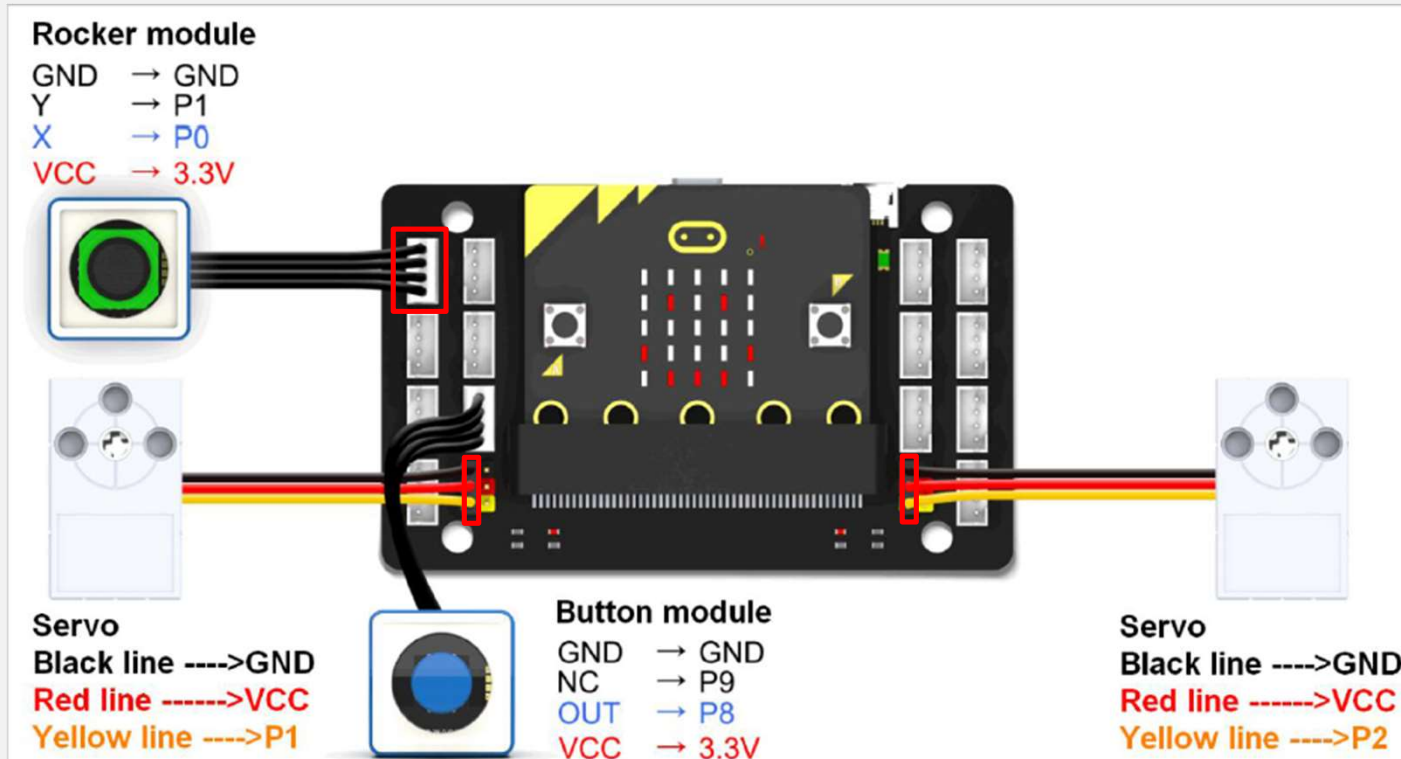
After the assembly is completed, please check the wiring of the servo as shown below.



30 Points

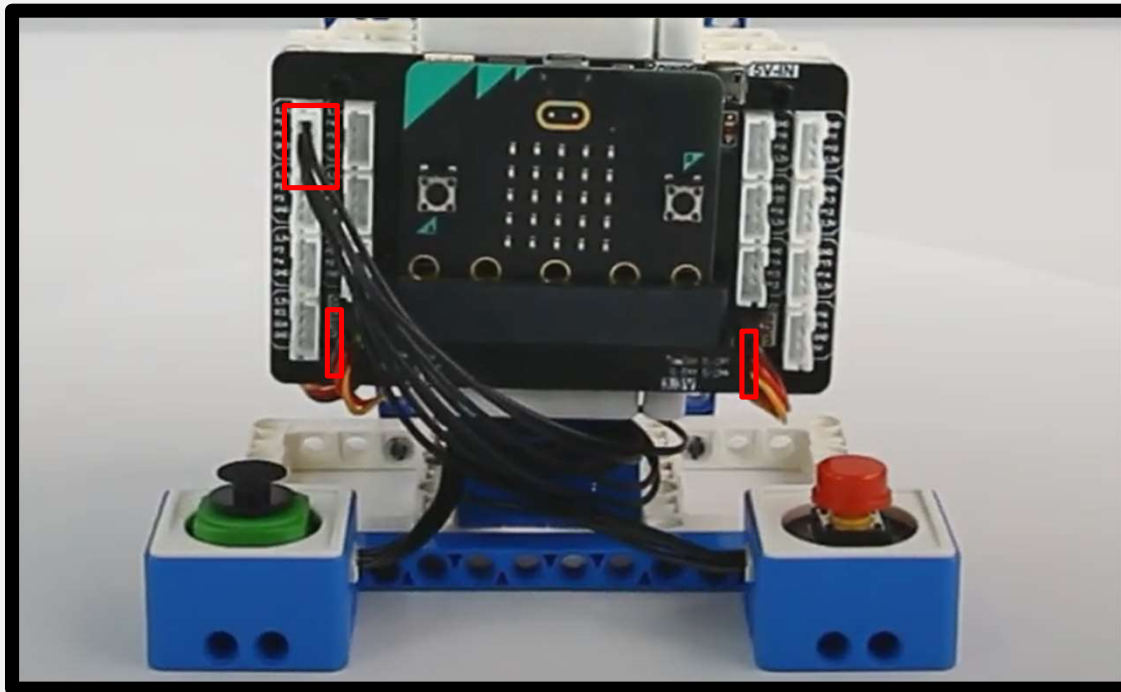
Wire Connection

Connect the modules



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

Connect the modules



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

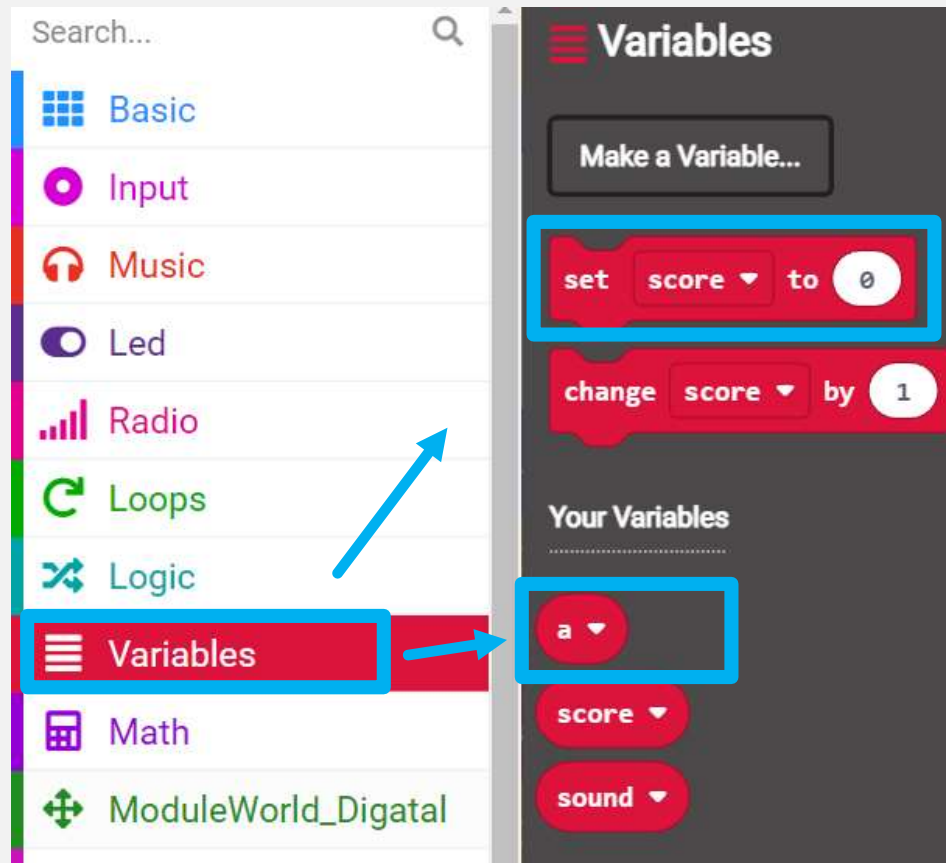
10 Points

MakeCode Programming

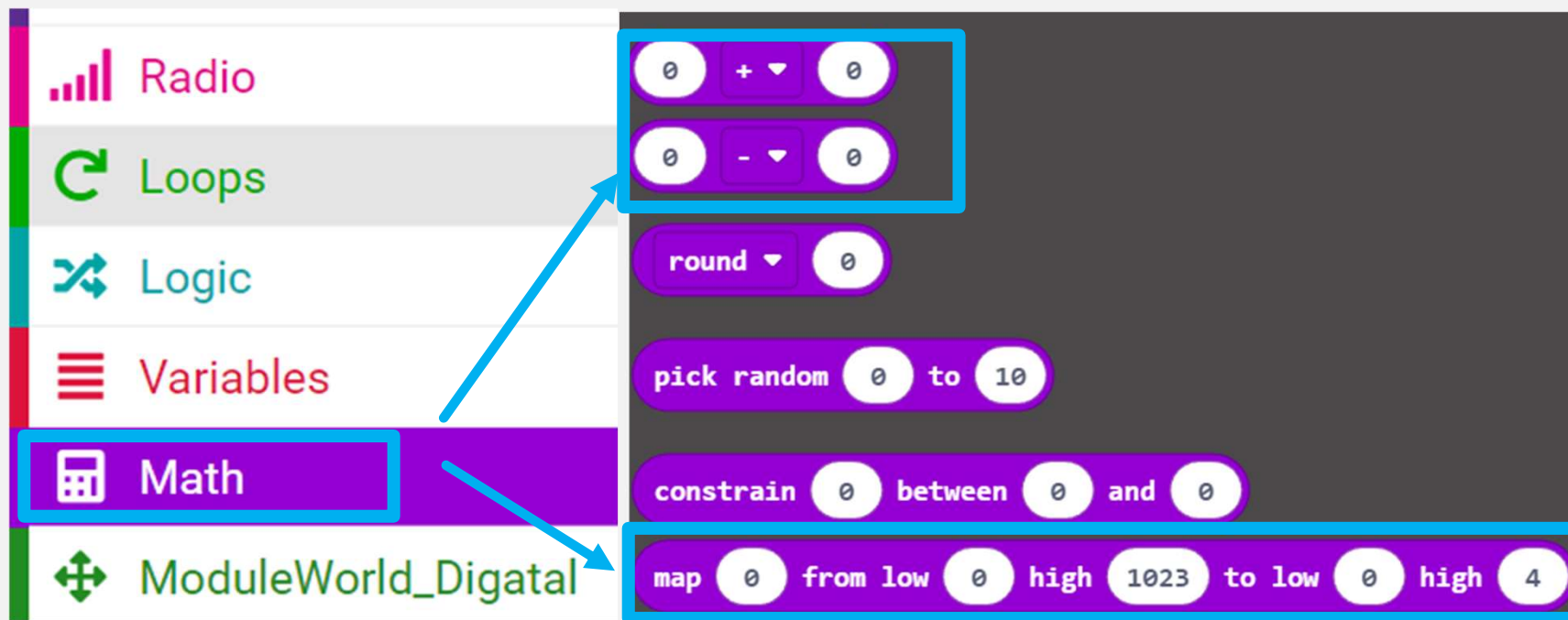
<https://github.com/YahboomTechnology/Module-World>.

Or search [YahboomTechnology/Module-World](https://github.com/YahboomTechnology/Module-World) in the extension block

Coding – Variables

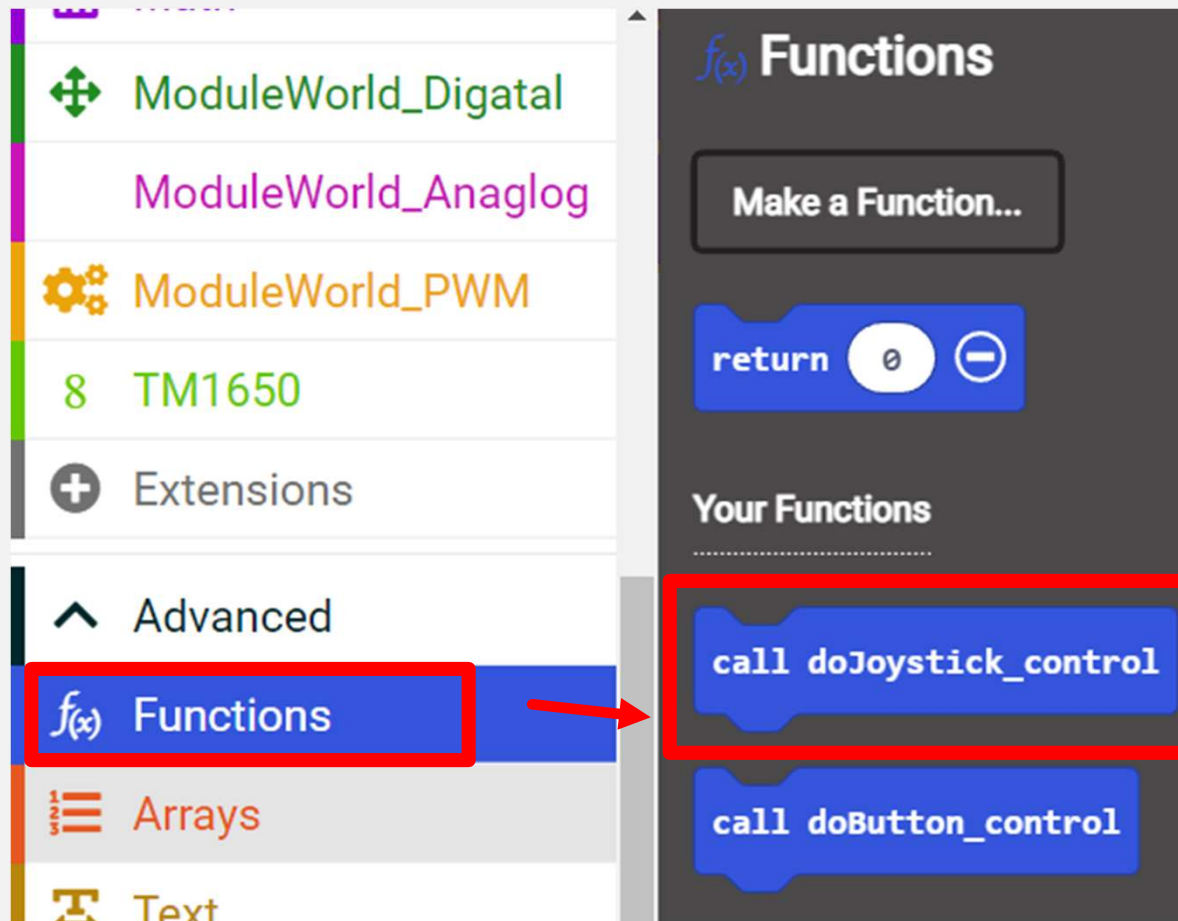


Coding – Math



The image shows a screenshot of the Microbit coding environment. On the left, a sidebar contains several categories: Radio, Loops, Logic, Variables, Math, and ModuleWorld_Digital. The 'Math' category is highlighted with a blue box. In the main workspace, several blocks are visible, including addition and subtraction blocks, a round block, a pick random block, a constrain block, and a map block. The 'Math' category in the sidebar and the 'map' block in the workspace are also highlighted with blue boxes. Two blue arrows point from the 'Math' category to the 'map' block and from the 'pick random' block to the 'Math' category.

Coding – Functions



The screenshot shows the Microbit coding environment. On the left is a palette of blocks, and on the right is the workspace. The 'Functions' block in the palette is highlighted with a red box, and a red arrow points to the 'call doJoystick_control' block in the workspace. The workspace also shows a 'call doButton_control' block below it.

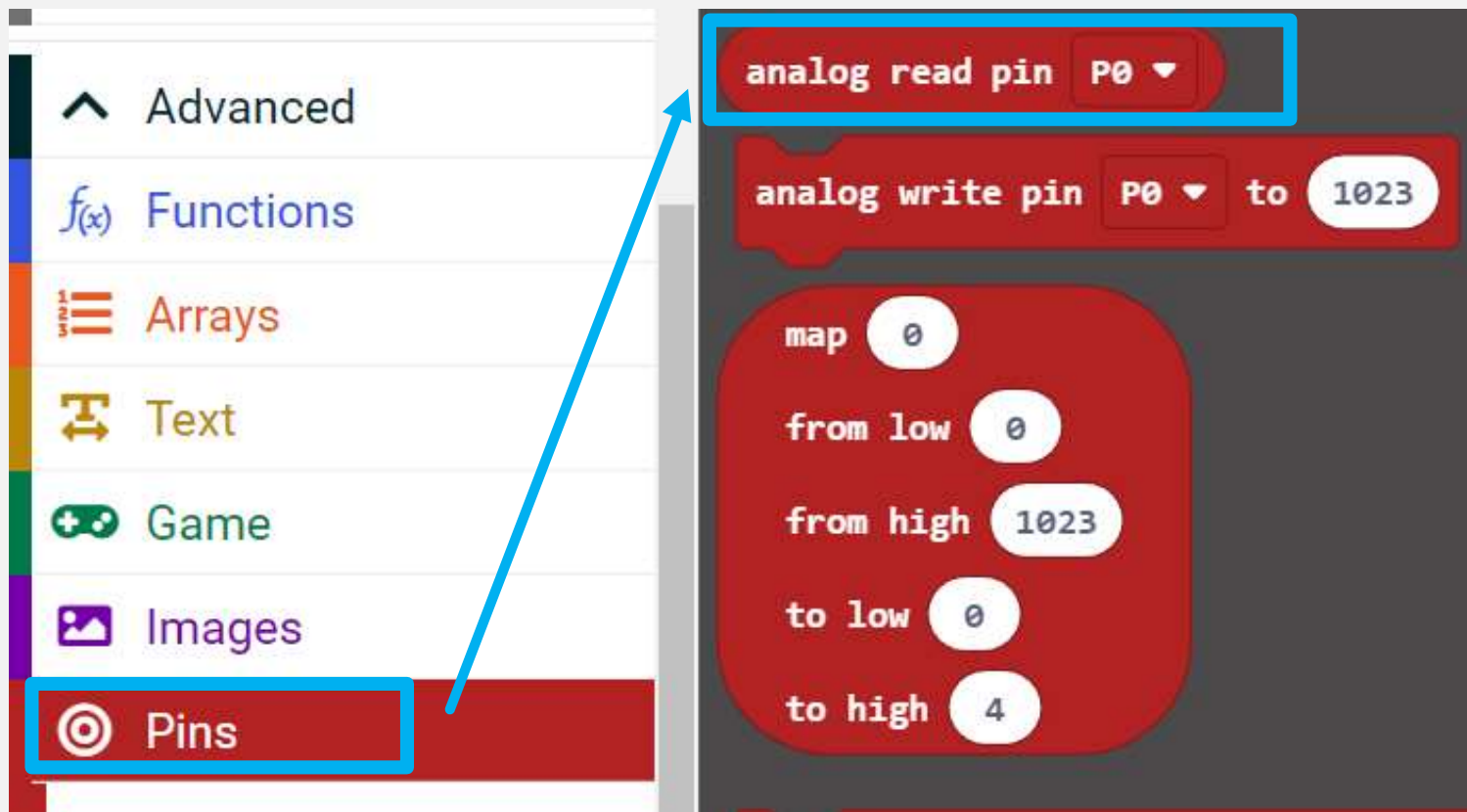
Functions Palette:

- ModuleWorld_Digatal
- ModuleWorld_Analog
- ModuleWorld_PWM
- TM1650
- Extensions
- Advanced
- f(x) Functions**
- Arrays
- Text

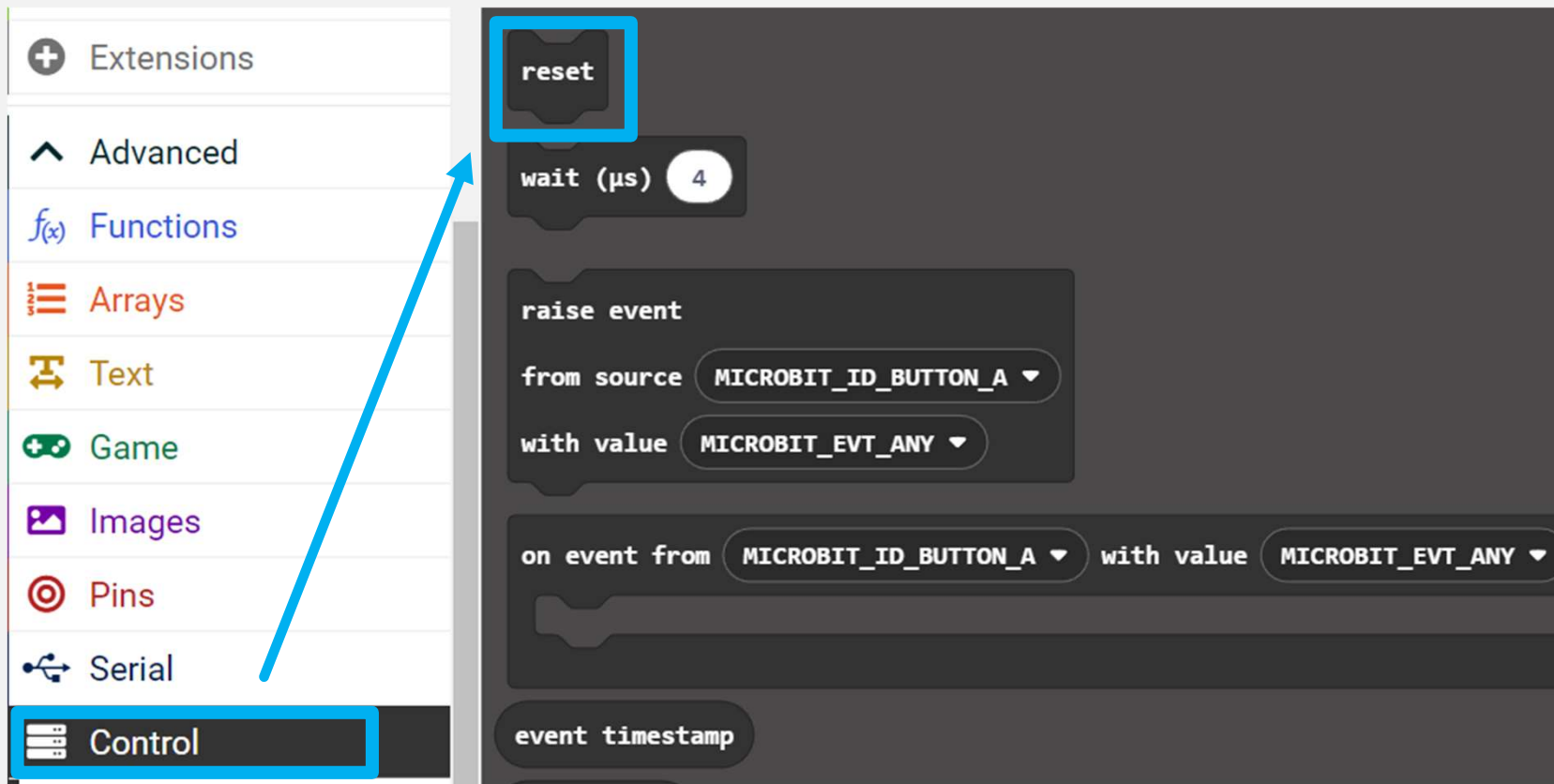
Workspace:

- Make a Function...
- return 0
- Your Functions
- call doJoystick_control**
- call doButton_control

Coding – Pins

A screenshot of the Microbit coding environment. On the left, a vertical menu lists various categories: Advanced, Functions, Arrays, Text, Game, Images, and Pins. The Pins category is highlighted with a blue box. A blue arrow points from this box to the top of a code block on the right. The code block contains several blocks: an "analog read pin" block with "P0" selected, an "analog write pin" block with "P0" selected and a value of "1023", and a "map" block with "0" in the "map" field, "0" in the "from low" field, "1023" in the "from high" field, "0" in the "to low" field, and "4" in the "to high" field.

Coding – Control



The screenshot displays the Microbit coding interface. On the left sidebar, the 'Control' extension is selected and highlighted with a blue box. A blue arrow points from the 'Control' extension in the sidebar to the 'reset' block in the main workspace. The 'reset' block is also highlighted with a blue box. Below the 'reset' block, there is a 'wait (µs)' block with the value '4'. Further down, there is a 'raise event' block with 'from source' set to 'MICROBIT_ID_BUTTON_A' and 'with value' set to 'MICROBIT_EVT_ANY'. Below that is an 'on event from' block with 'MICROBIT_ID_BUTTON_A' for the source and 'MICROBIT_EVT_ANY' for the value. At the bottom, there is an 'event timestamp' block.

Coding - Overview

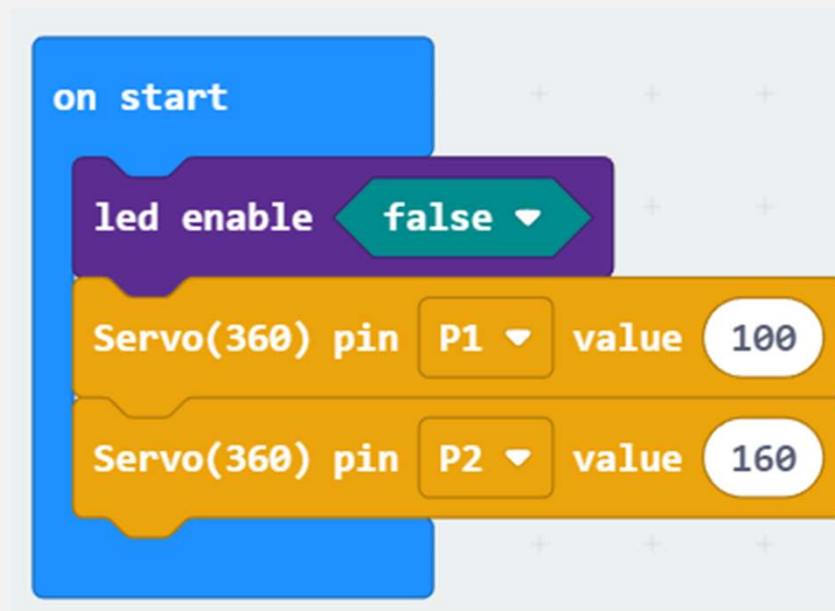


```
on start
  led enable false
  Servo(360) pin P1 value 100
  Servo(360) pin P2 value 160
```

```
forever
  call doJoystick_control
```

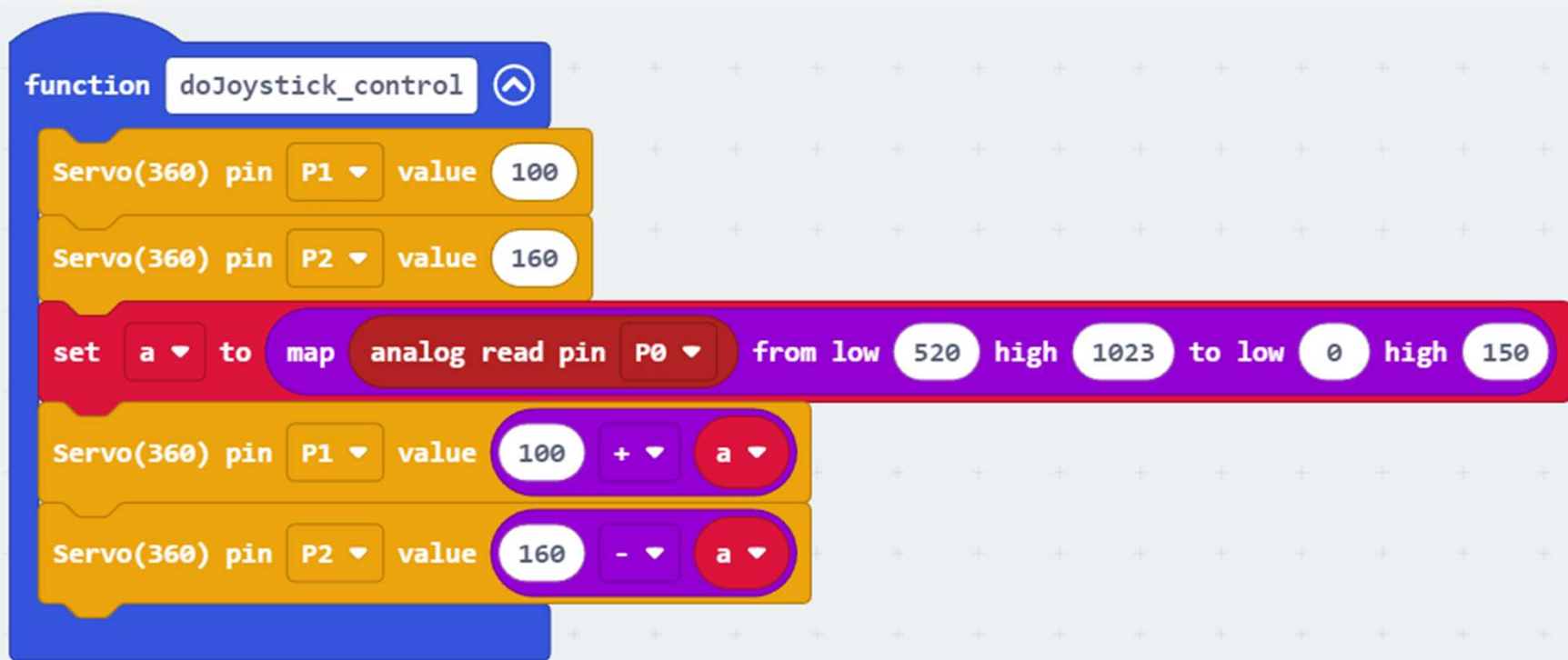
```
function doJoystick_control
  Servo(360) pin P1 value 100
  Servo(360) pin P2 value 160
  set a to map analog read pin P0 from low 520 high 1023 to low 0 high 150
  Servo(360) pin P1 value 100 + a
  Servo(360) pin P2 value 160 - a
```

Coding – On start

A screenshot of a Scratch-style code editor showing an 'on start' event block. The event block is blue and contains three stacked blocks: a purple 'led enable' block with a dropdown menu set to 'false', an orange 'Servo(360)' block with 'pin' set to 'P1' and 'value' set to '100', and another orange 'Servo(360)' block with 'pin' set to 'P2' and 'value' set to '160'. The code blocks are connected by a blue vertical bar on the left side of the 'on start' block.

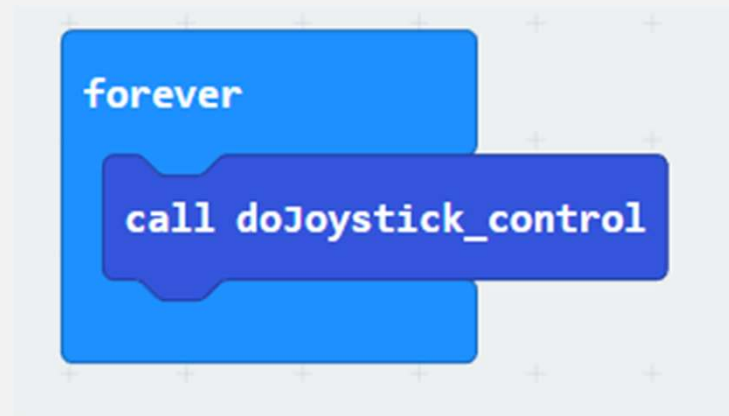
```
on start
  led enable false
  Servo(360) pin P1 value 100
  Servo(360) pin P2 value 160
```

Coding – rocker function, doJoystick_control

A Scratch code block for a function named 'doJoystick_control'. The function starts with a blue 'function' block. It contains five stacked blocks: two yellow 'Servo(360) pin' blocks for P1 and P2 with values 100 and 160; a purple 'set a to map' block for pin P0 with low 520, high 1023, to low 0, high 150; and two yellow 'Servo(360) pin' blocks for P1 and P2 with values 100 and 160, each followed by a purple 'math' block containing a '+' or '-' sign and the variable 'a'.

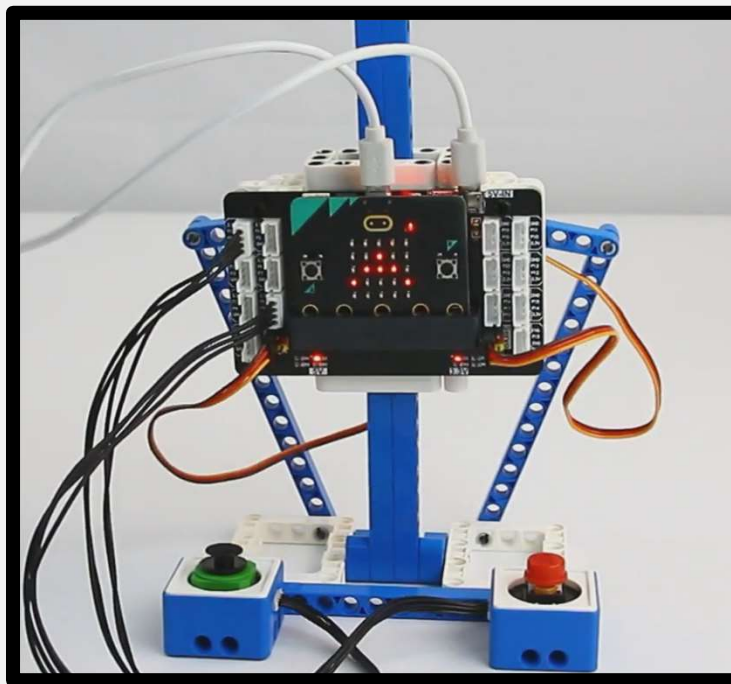
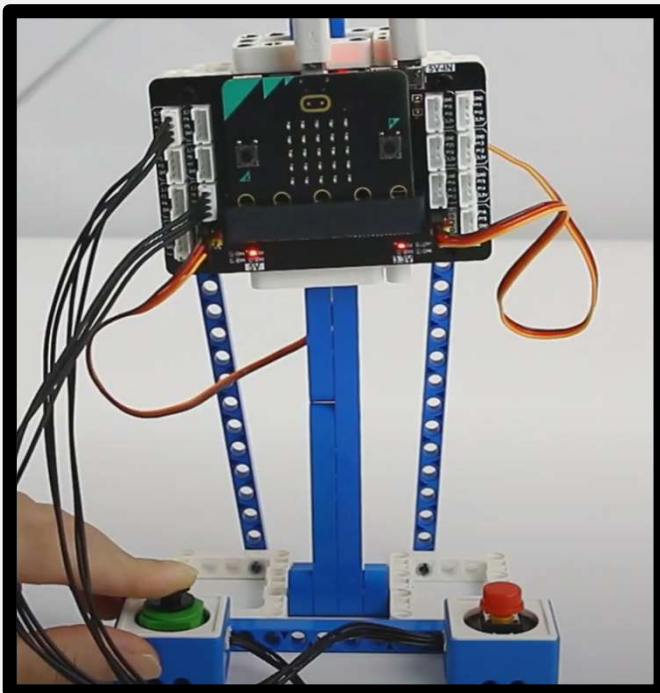
```
function doJoystick_control  
  Servo(360) pin P1 value 100  
  Servo(360) pin P2 value 160  
  set a to map analog read pin P0 from low 520 high 1023 to low 0 high 150  
  Servo(360) pin P1 value 100 + a  
  Servo(360) pin P2 value 160 - a
```

Coding – forever



Phenomenon

When we push the **rocker** forward, the lifting platform will **rise**,
When the rocker return to the **middle(no state)**, and the lifting platform will **fall** to the bottom.



30 Points

CHALLENGE

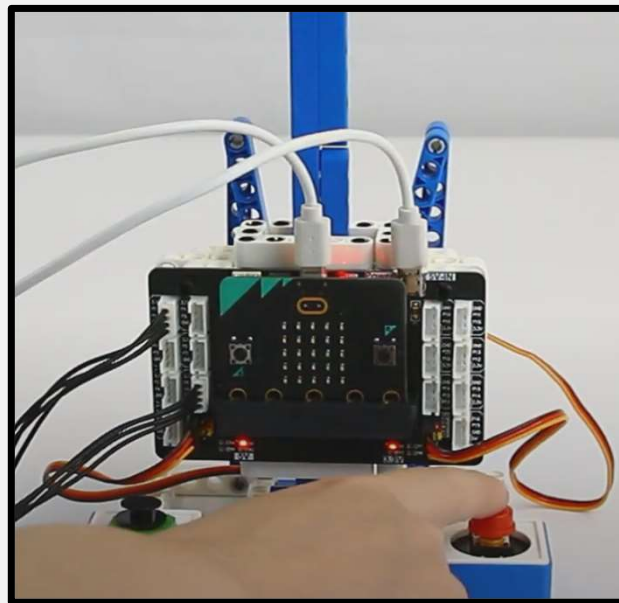
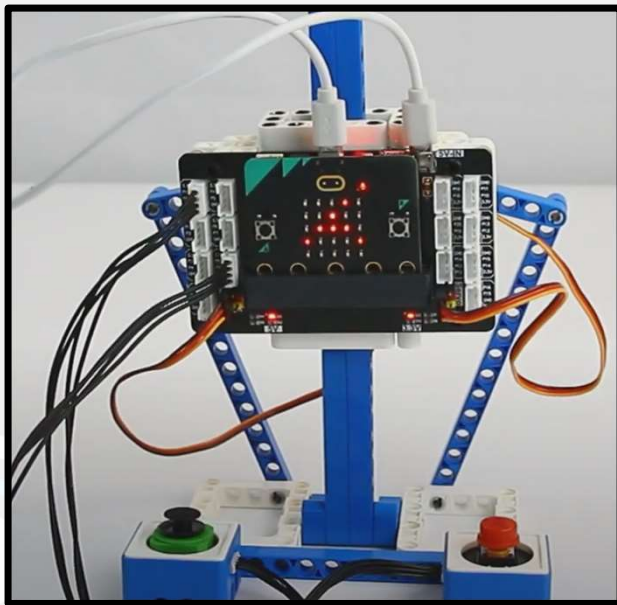
for : Lesson 2

L2 – Challenge 1

Using **button** module,

The lifting platform will **rise** when we **press** the button for the **first** time.

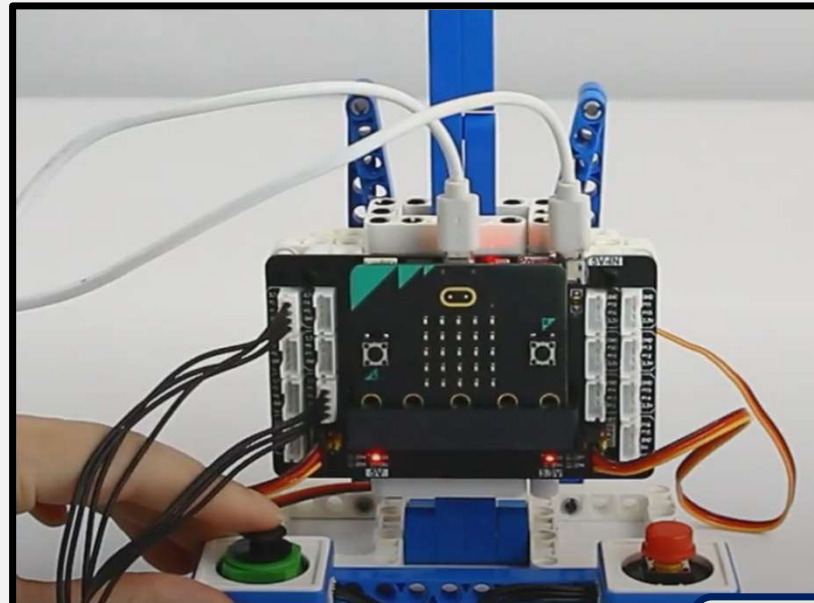
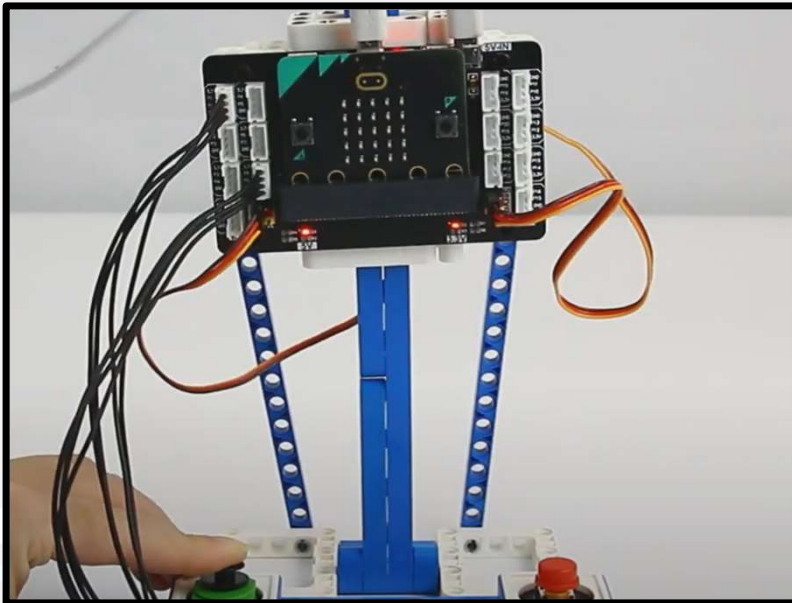
The lifting platform will **fall** to the **bottom** when we **press** the button again.



30 Points

L2 – Challenge 2

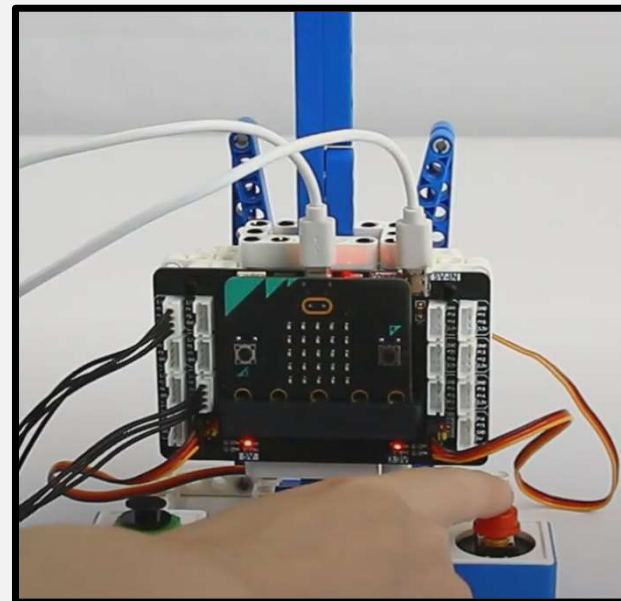
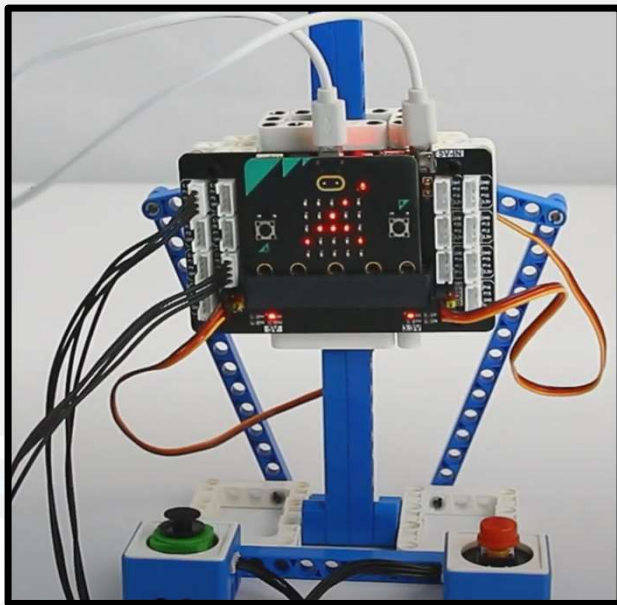
Can you make your **rocker** module **control** the platform **up and down** instead of just letting it fall?



30 Points

L2 – Challenge 3

Make the buzzer play a **power up** sound when the lifting platform **ris**es,
 When the lifting platform will **fall** to the bottom, the buzzer will play **power down** sound.



30 Points

L2 – Mission

Based on your lifting platform.

When the button module is pressed, the platform will rise into a certain **height**.

The **more** you press the button, the **higher** the platform rises.

Example:

1st time: The platform will rise a little

2nd time: The platform will rise more than 1st time

3rd time: The platform will rise more than 2nd time

4th time: The platform will to its maximum height

5th time: The platform will return to original position

50 Points