



Microbit Robotics Beginner Level 2

Lesson 4

Sunflower

Presented by Advanced Superlogic Team

Today's Topic

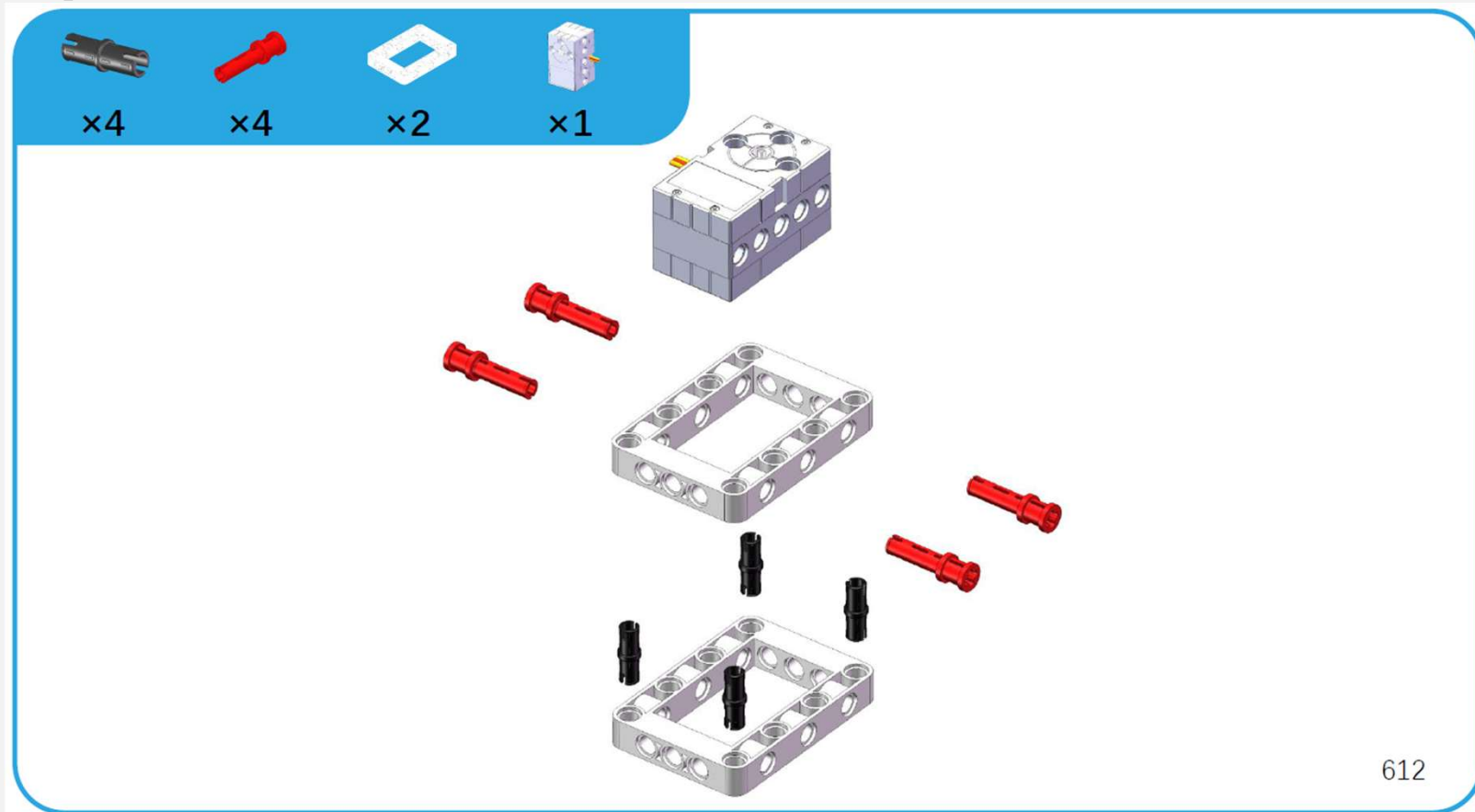
- 1. Build Sunflower with World of Modules**
- 2. Button programming**
- 3. Functions and Array/Lists programming**
- 4. Servo programming**

Learning Outcome

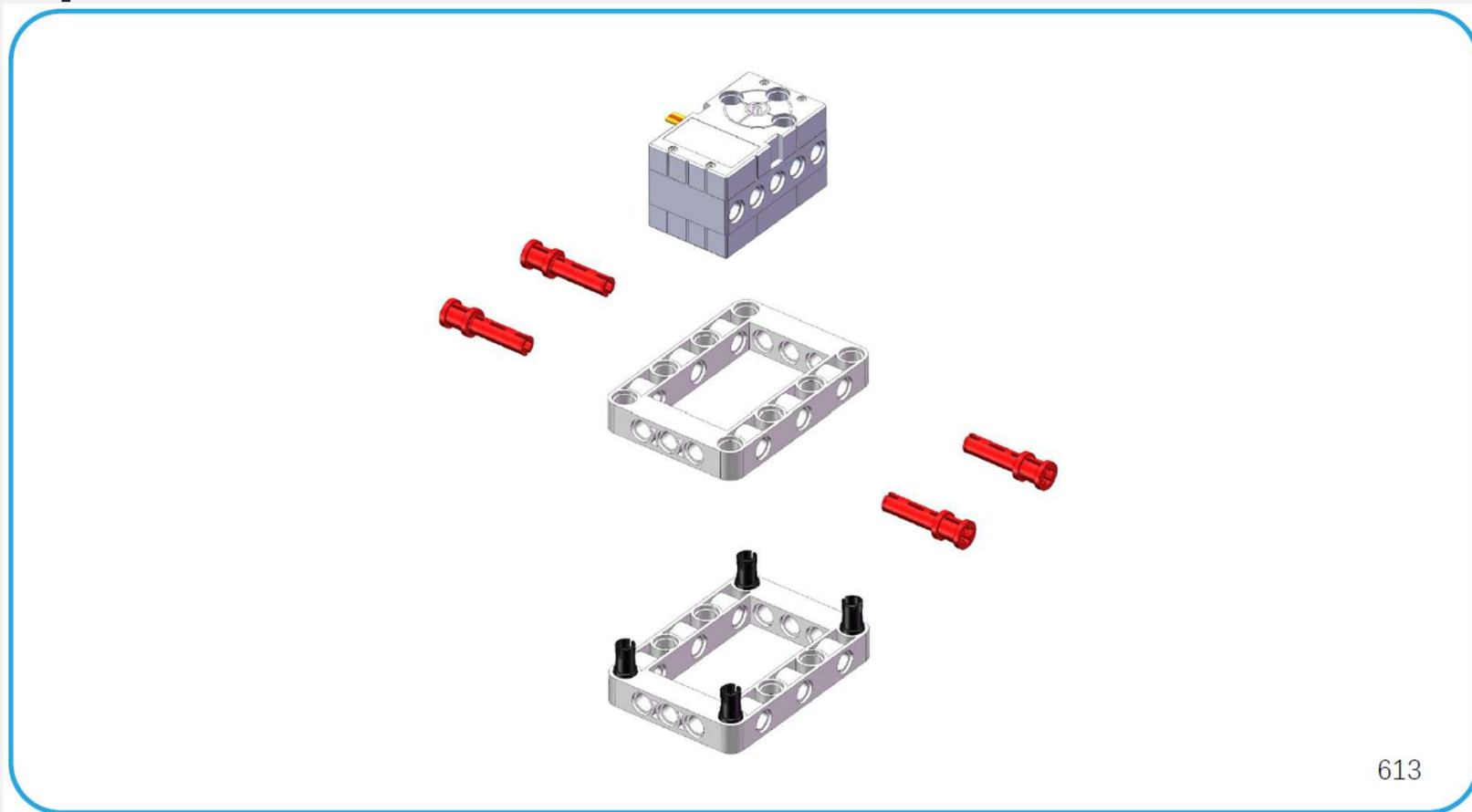
- 1. Able to build Sunflower with instruction manual**
- 2. Understand Button and Rocker programming**
- 3. Understand Servo programming work**
- 4. Understand arrays, variables and functions used**
- 5. Able to program Sunflower**

Lets build Sunflower

Step 1

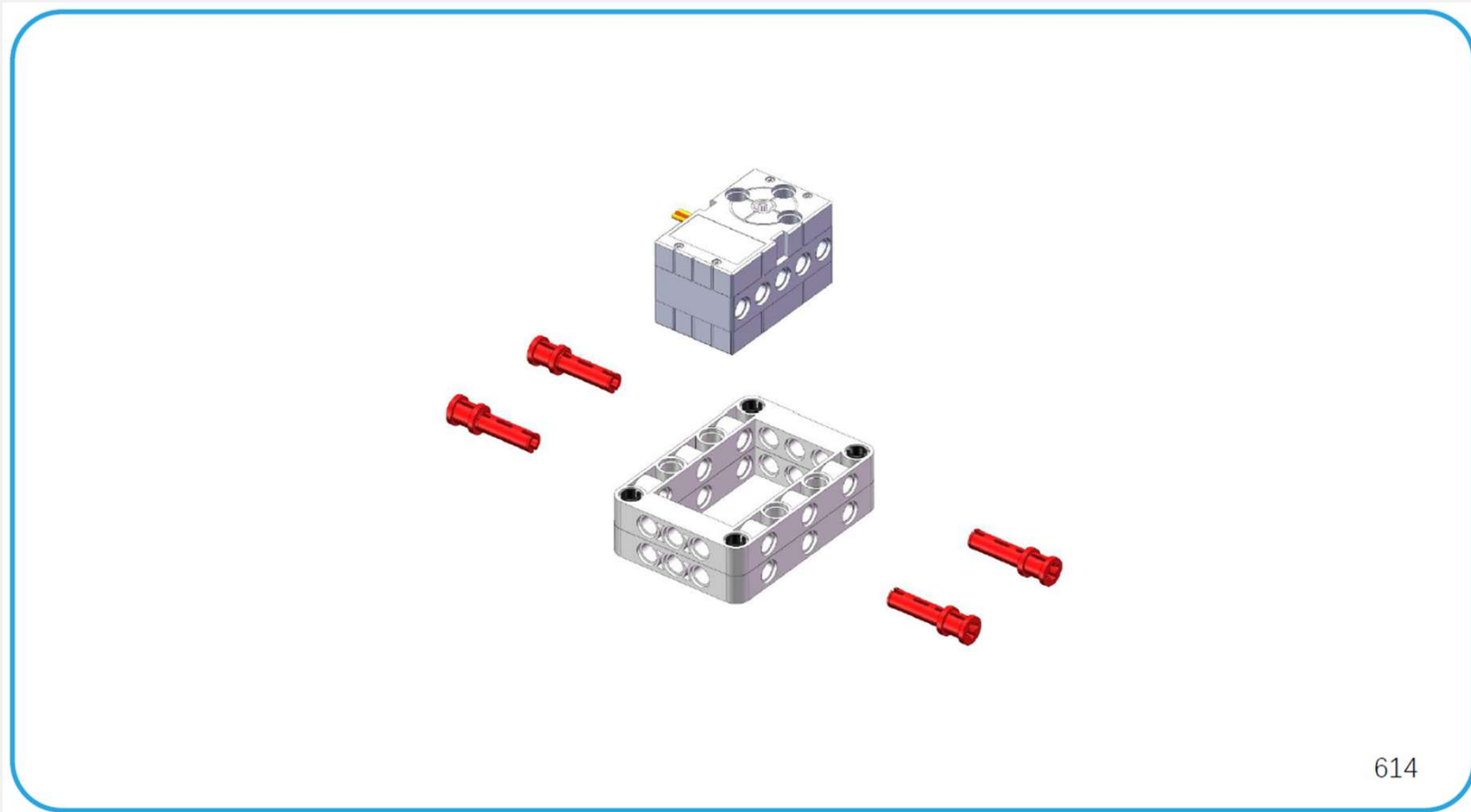


Step 2



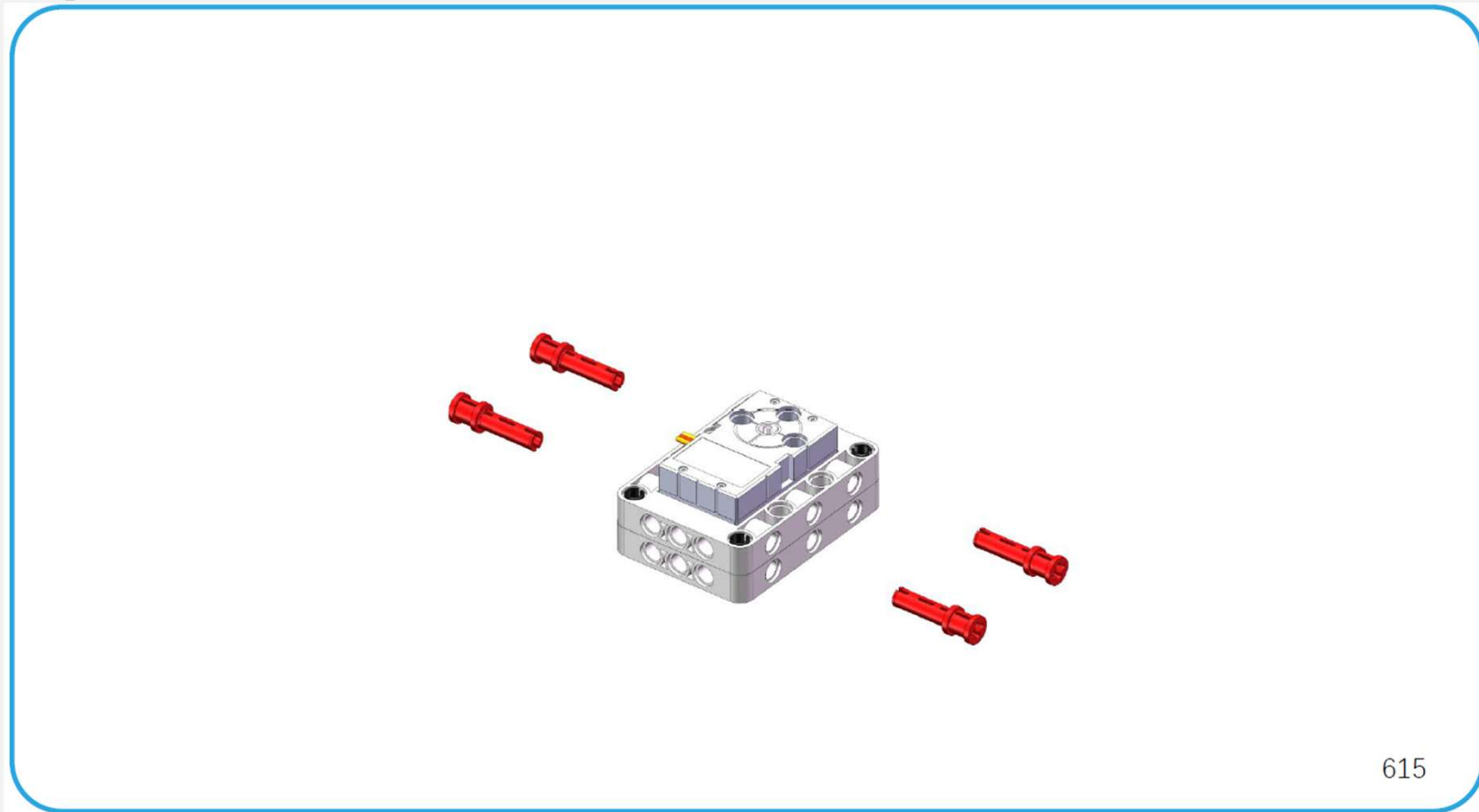
613

Step 3



614

Step 4



615

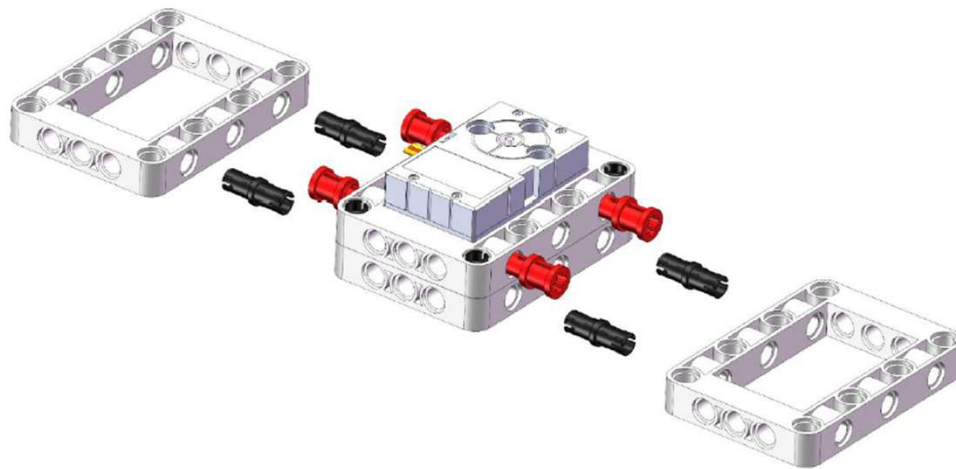
Step 6



x4

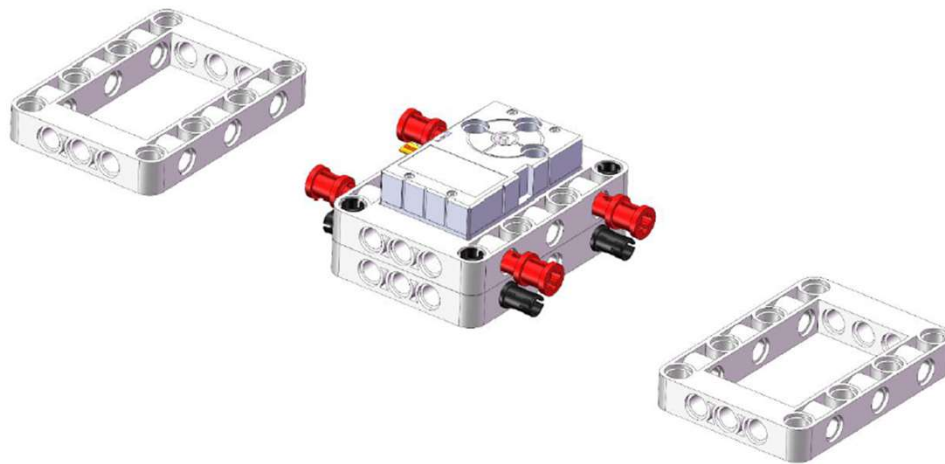


x2



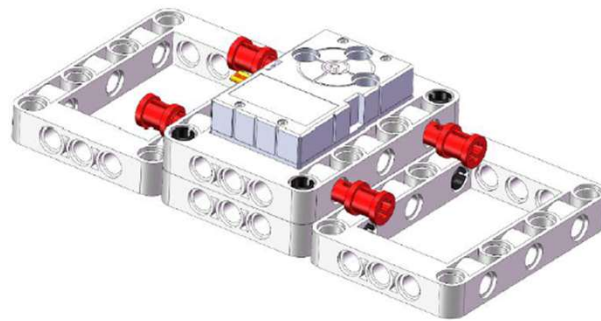
617

Step 7



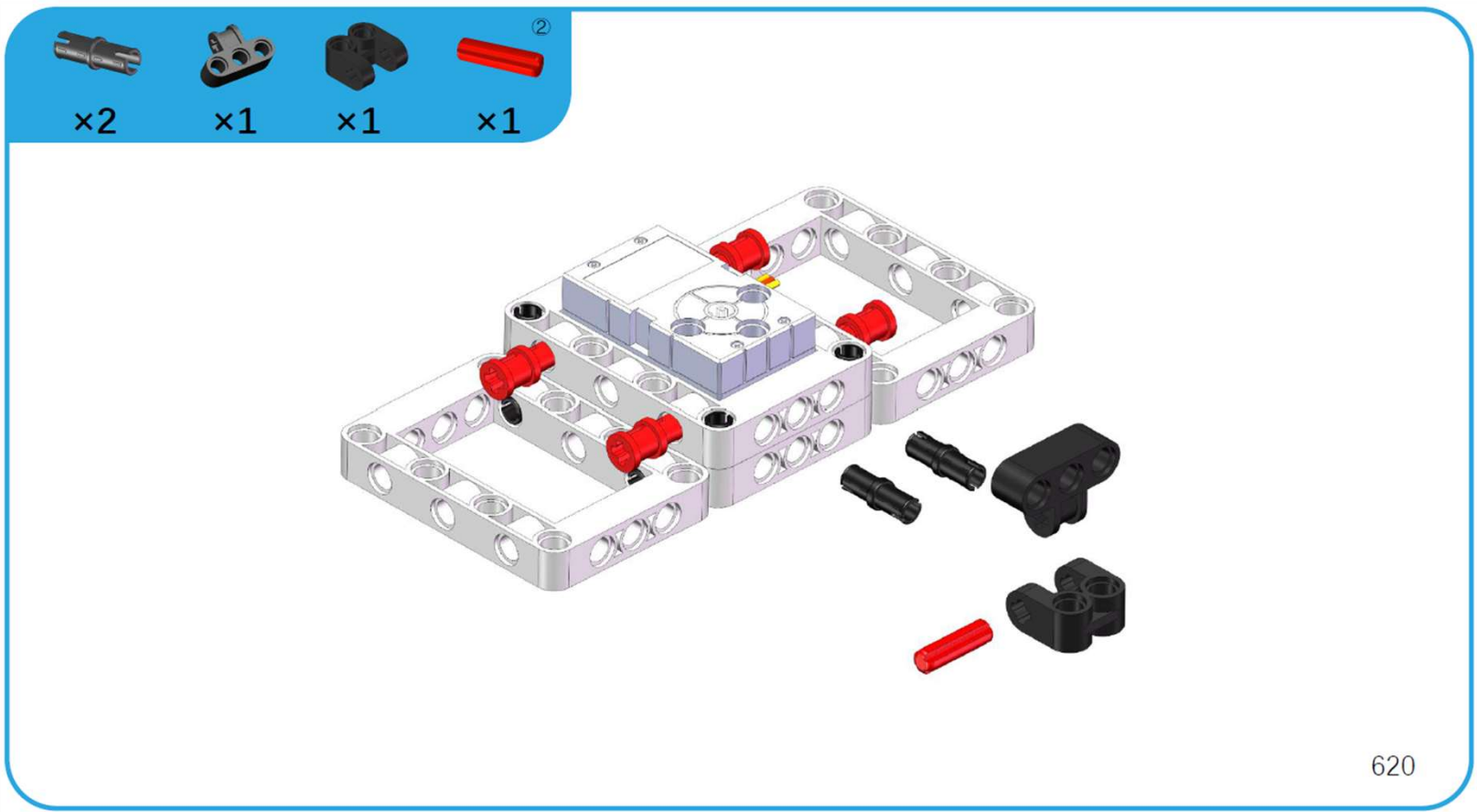
618

Step 8

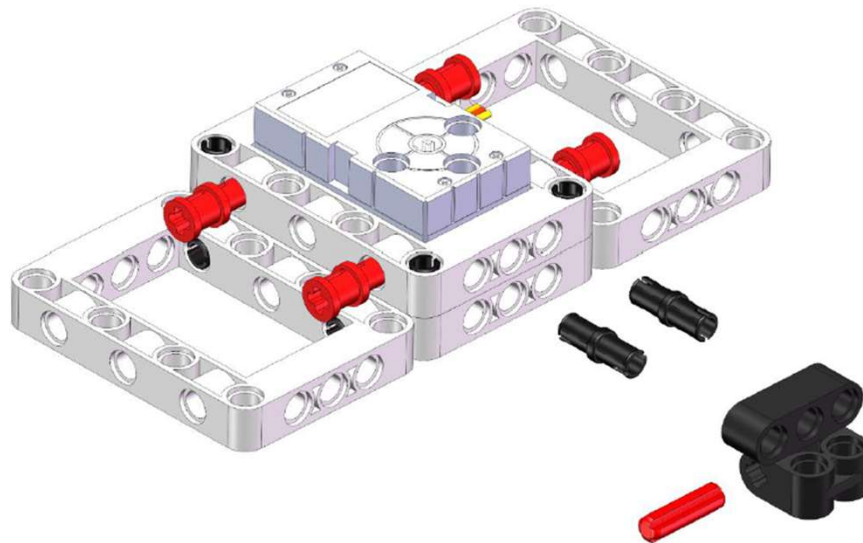


619

Step 9

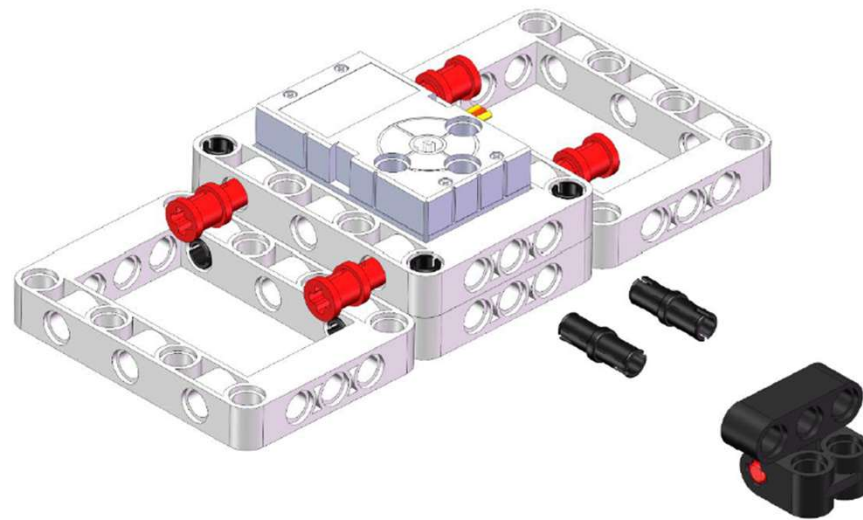


Step 10



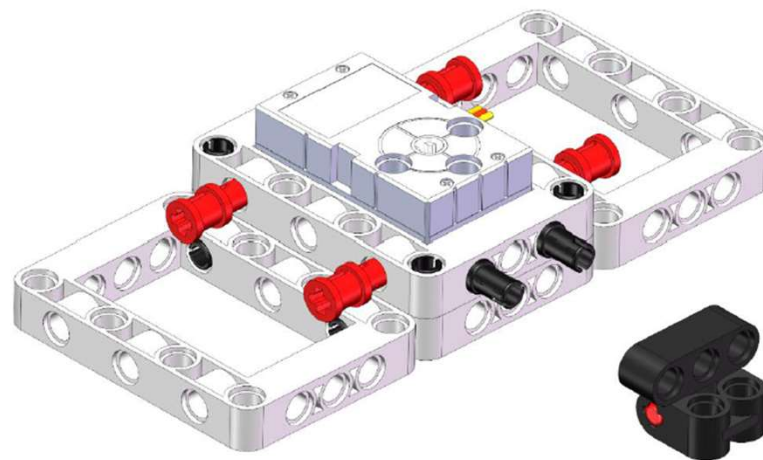
621

Step 11



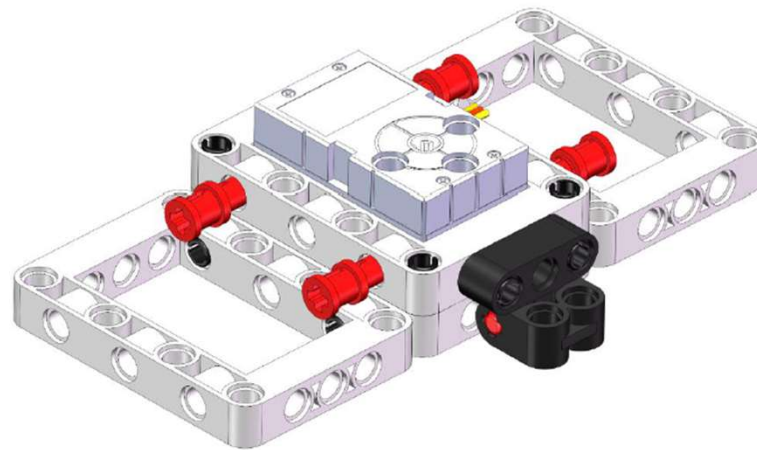
622

Step 12



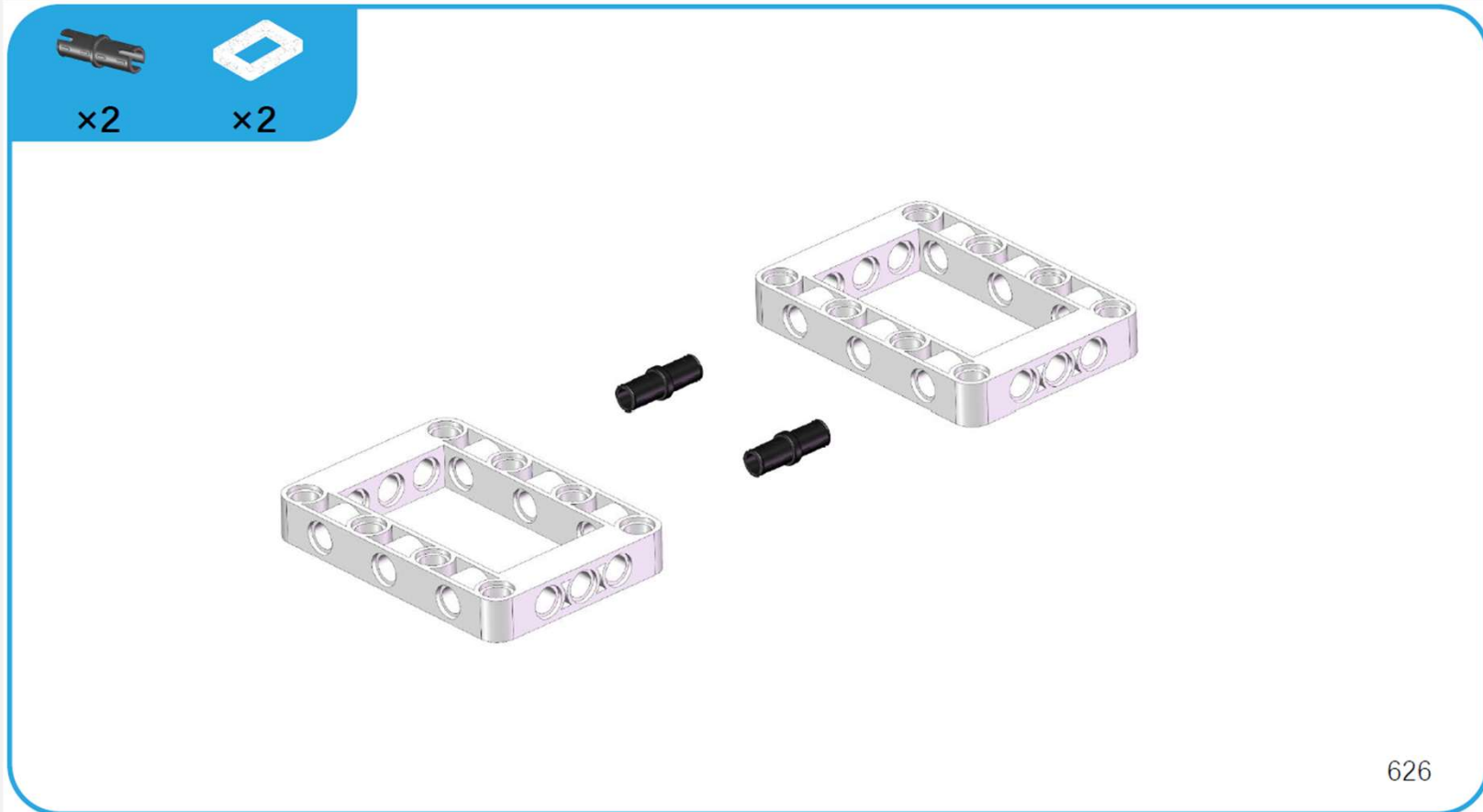
623

Step 13



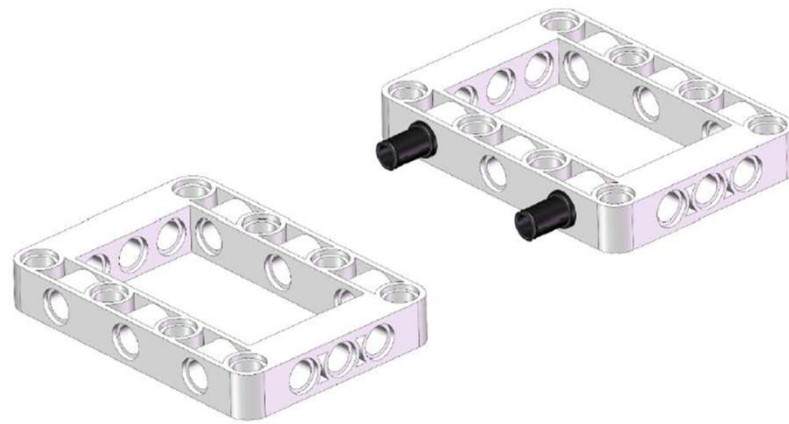
625

Step 14



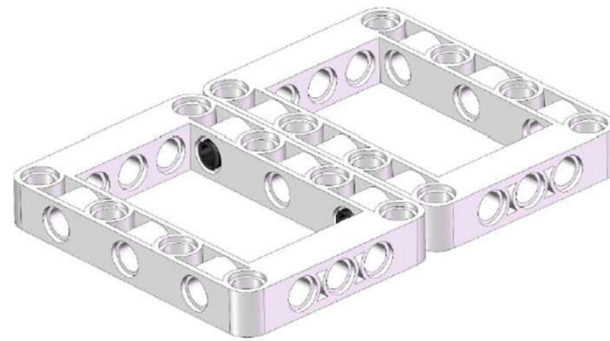
626

Step 15



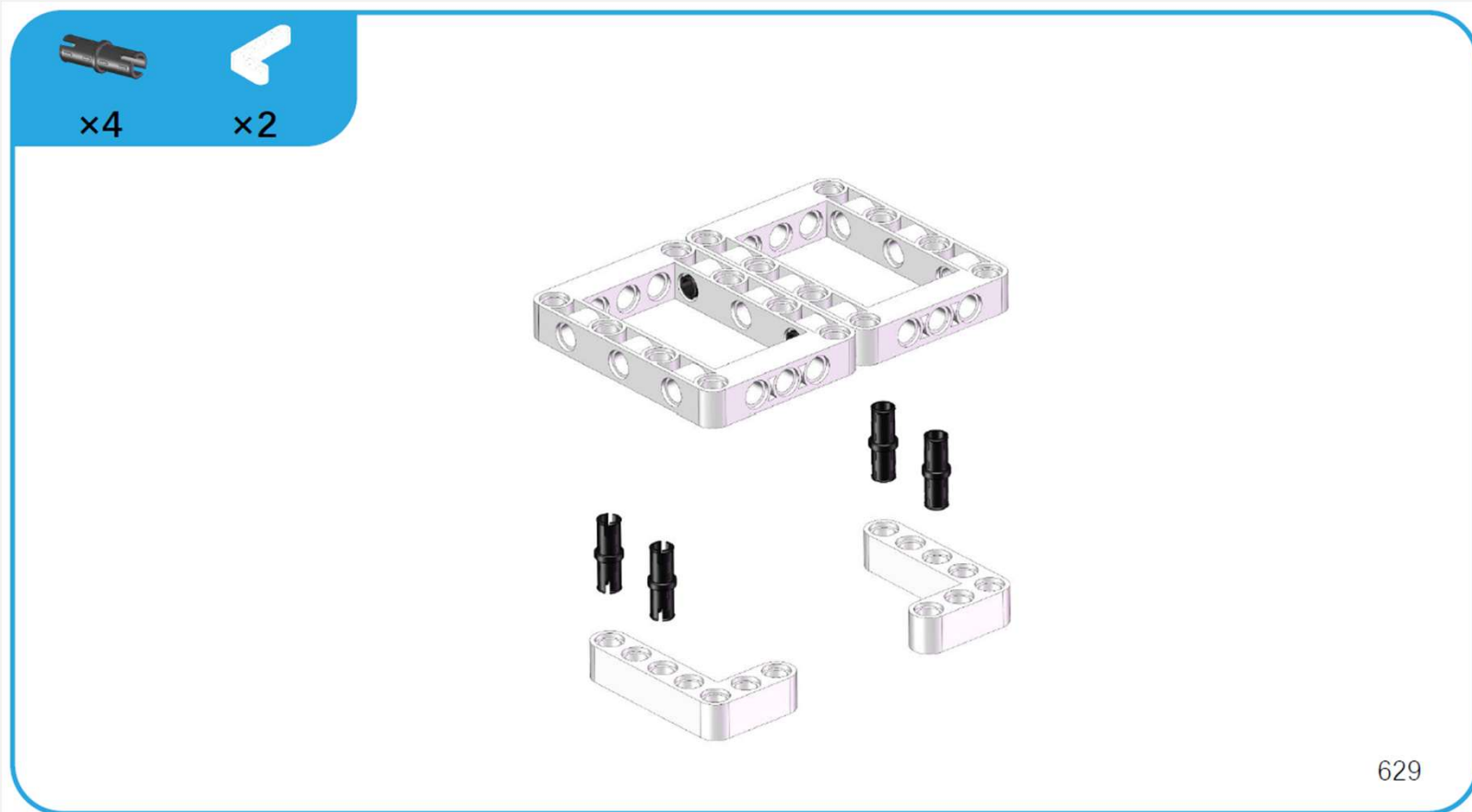
627

Step 16

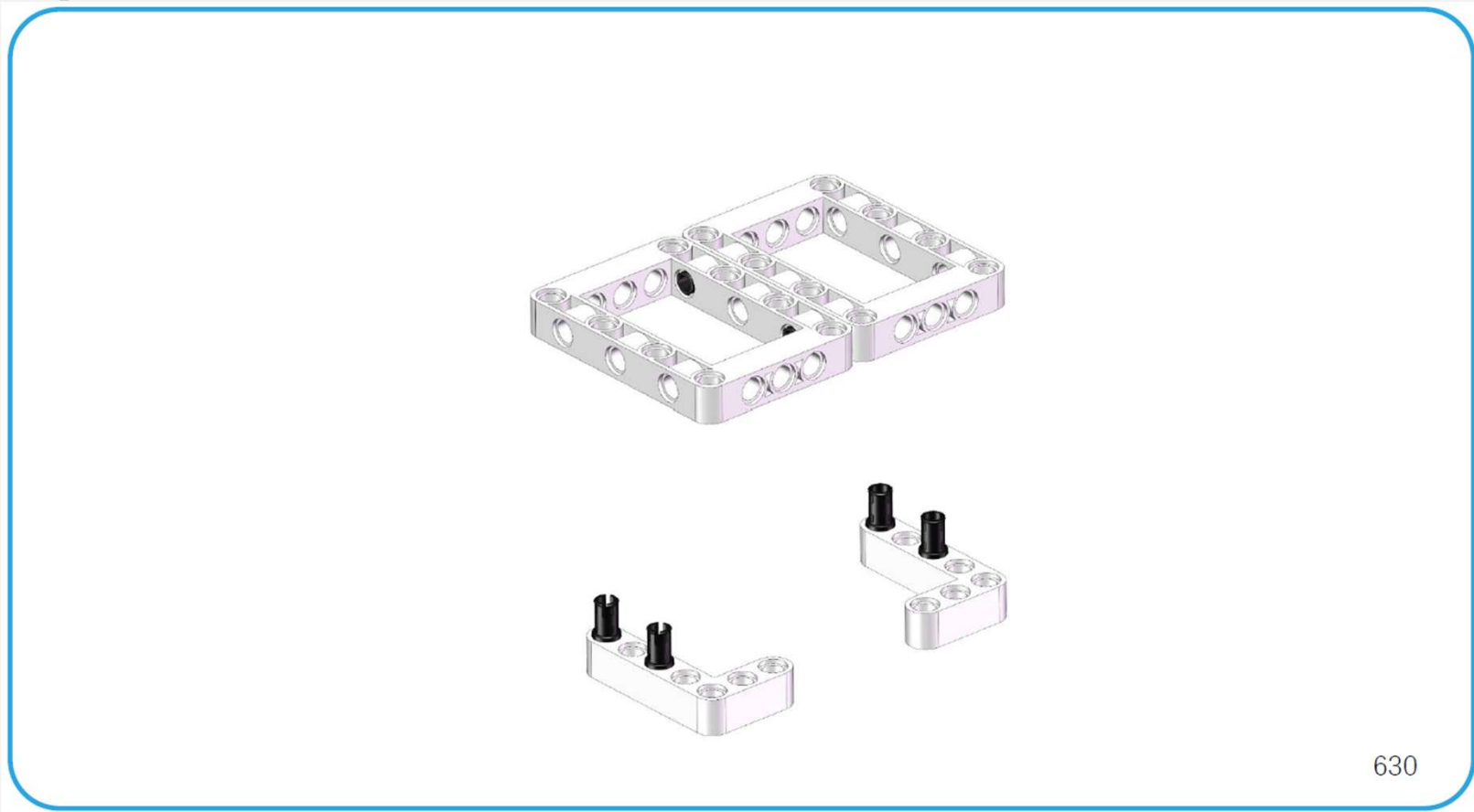


628

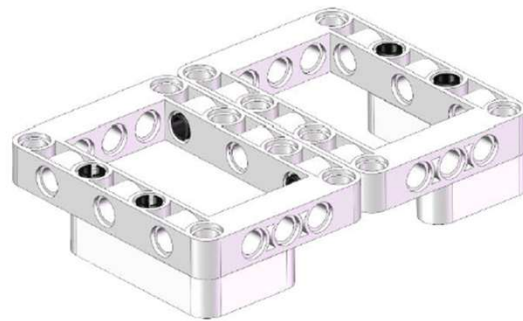
Step 17



Step 18



Step 19

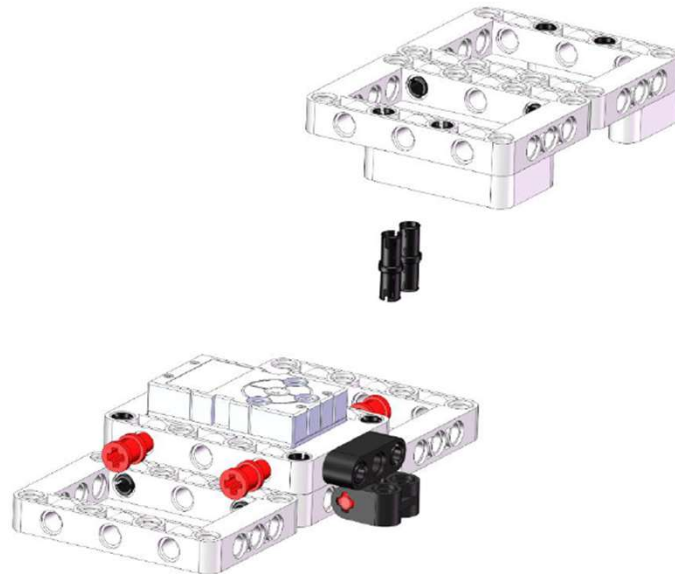


631

Step 20

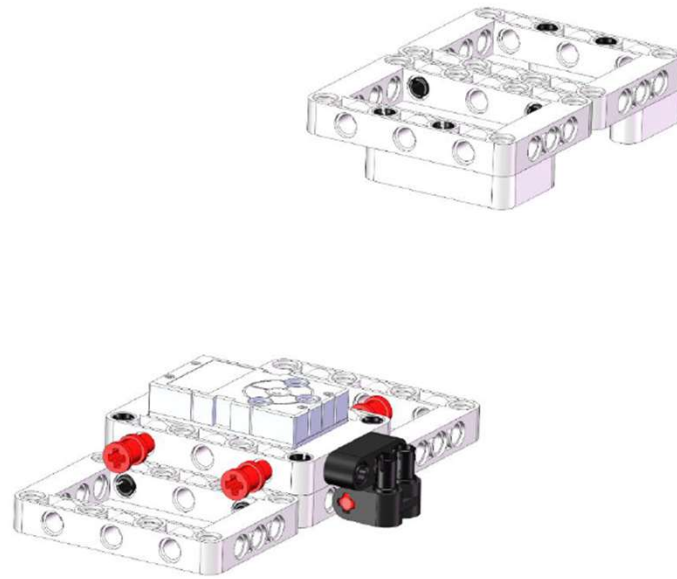


x2



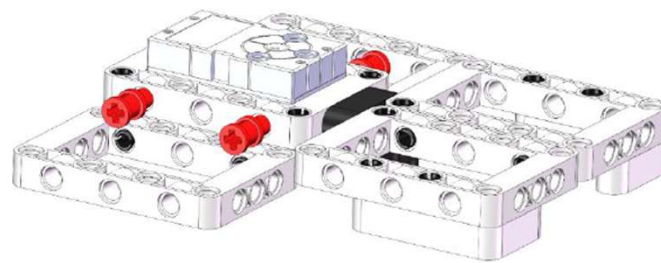
632

Step 21



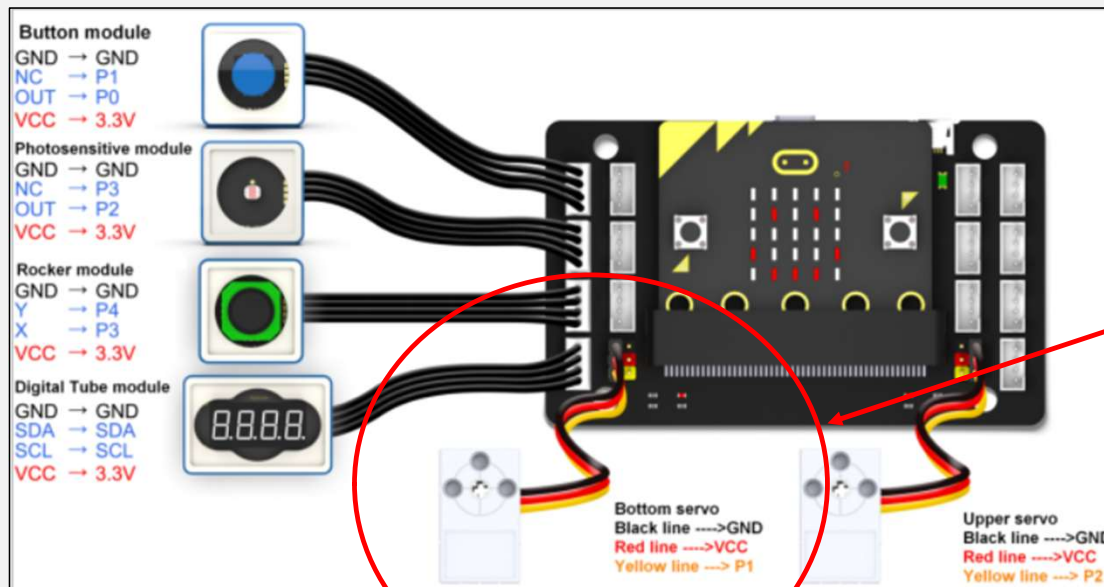
633

Step 22



634

Step 22 – Calibrate your Servo P1



Connect your servo to P1 on your expansion board, this servo is the bottom servo for the sunflower model.

Before continuing the next building step, we need to calibrate the servo and make sure it's in the correct position.

Step 22 – Calibrate your Servo P1



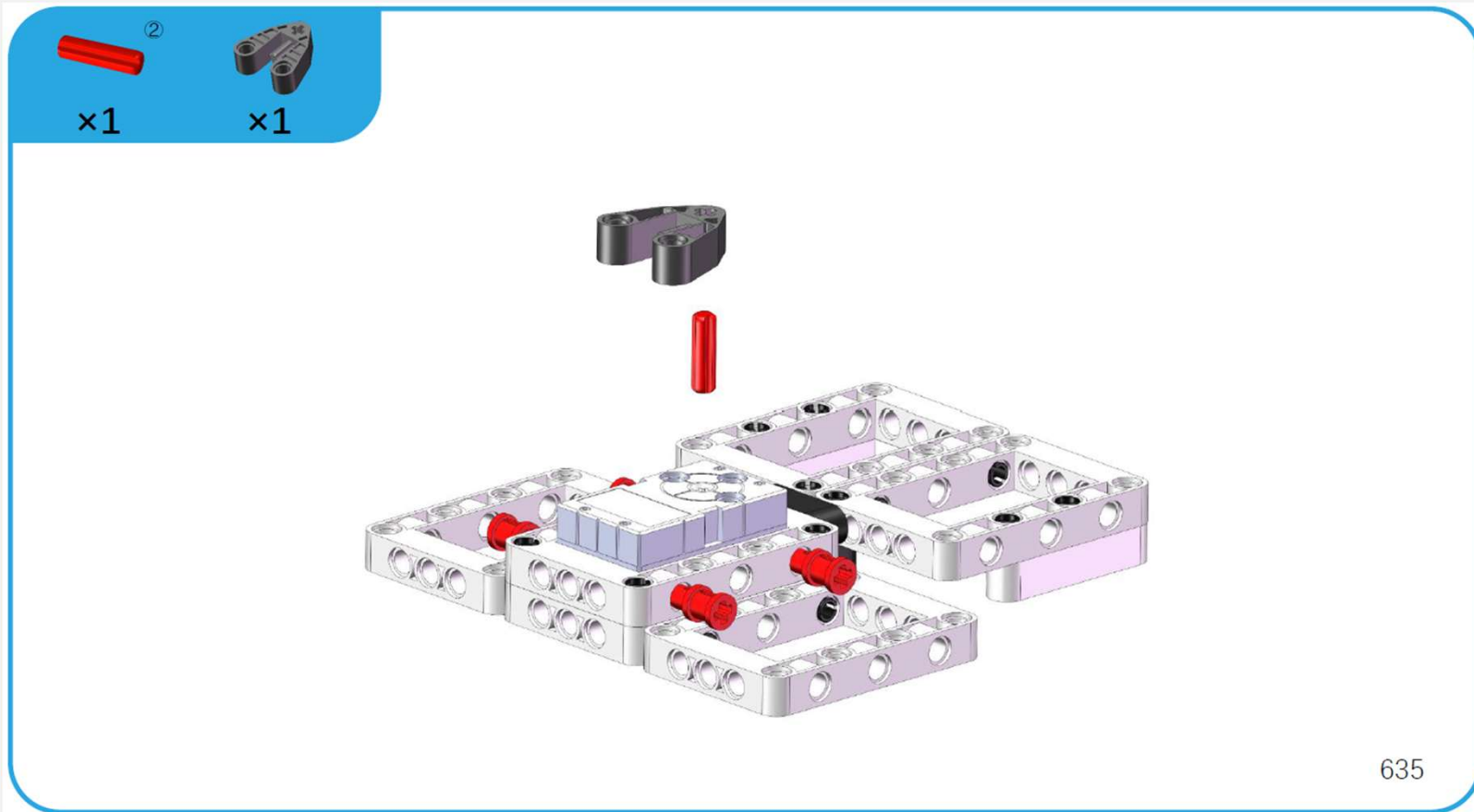
Then go to makecode and set the servo (360) pin P1 value to 180.

Remember to add the extension:

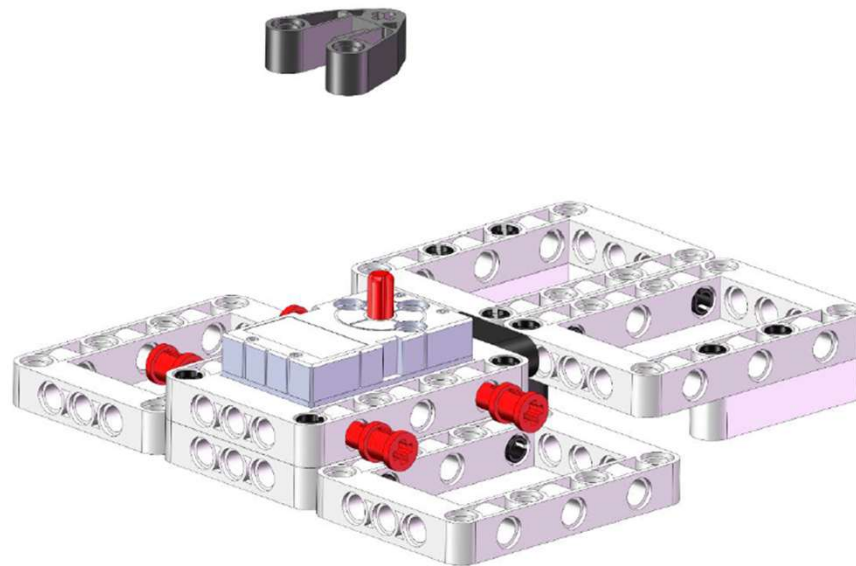
[Yahboomtechnology/module-world](https://www.yahboomtechnology.com/module-world/)

Then download and make your servo go to position 180.

Step 23

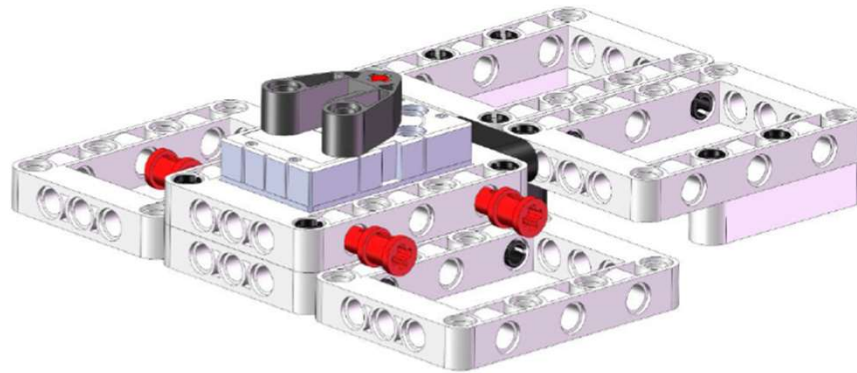


Step 24



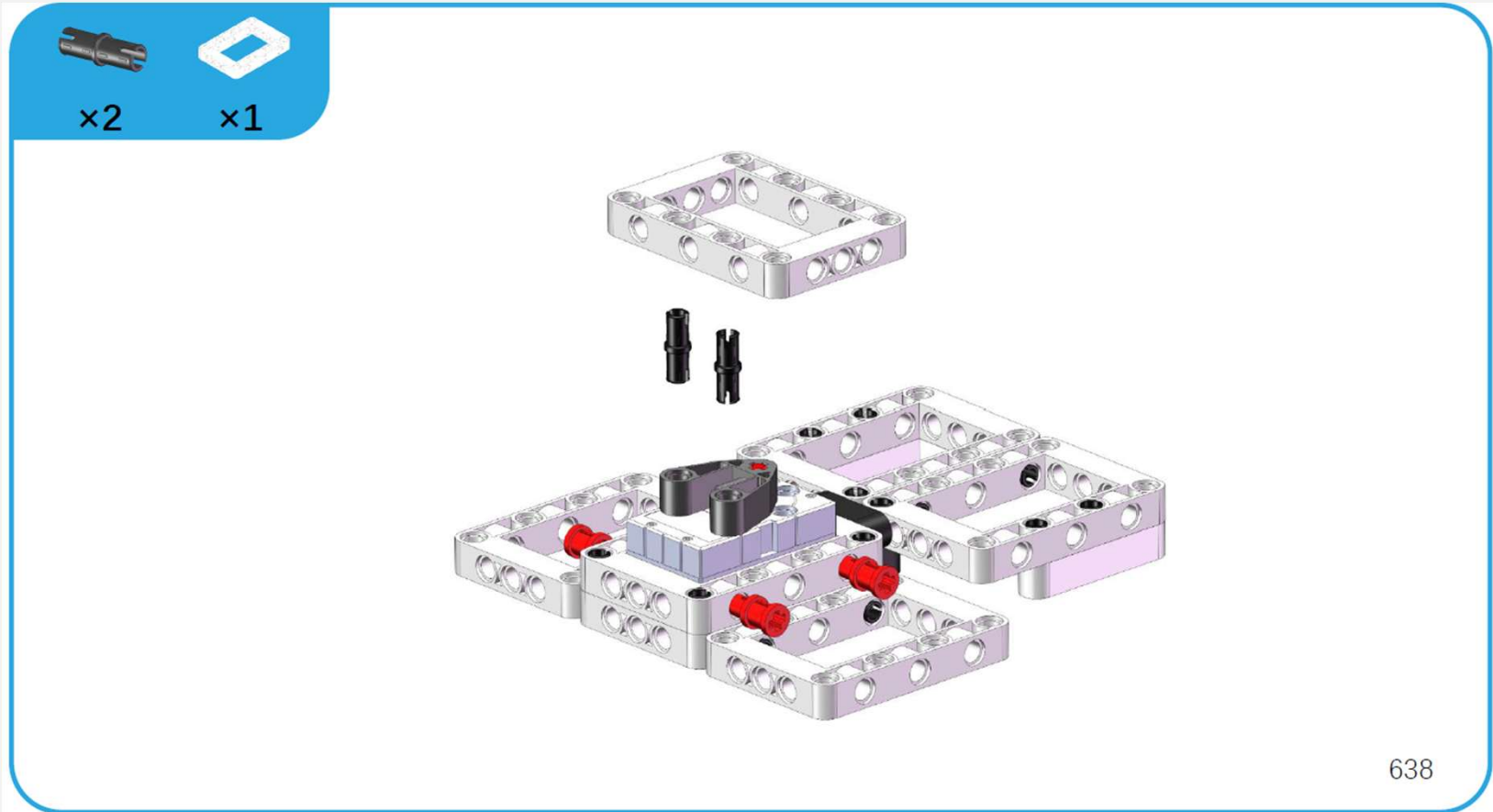
636

Step 25



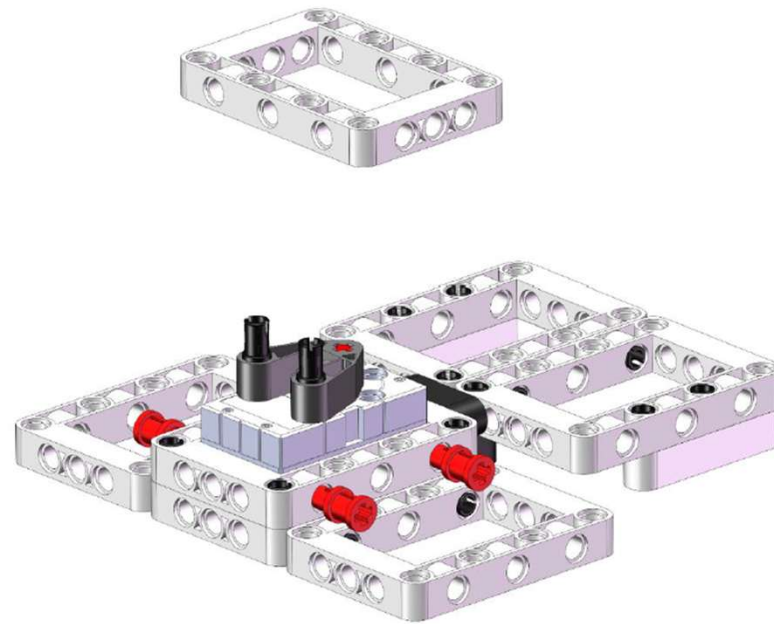
637

Step 26



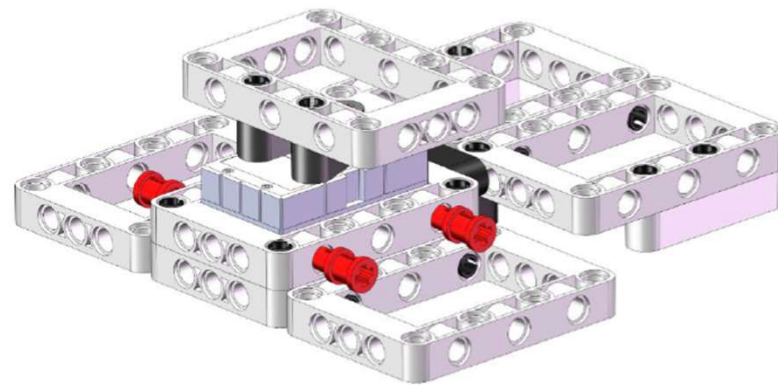
638

Step 27



639

Step 28



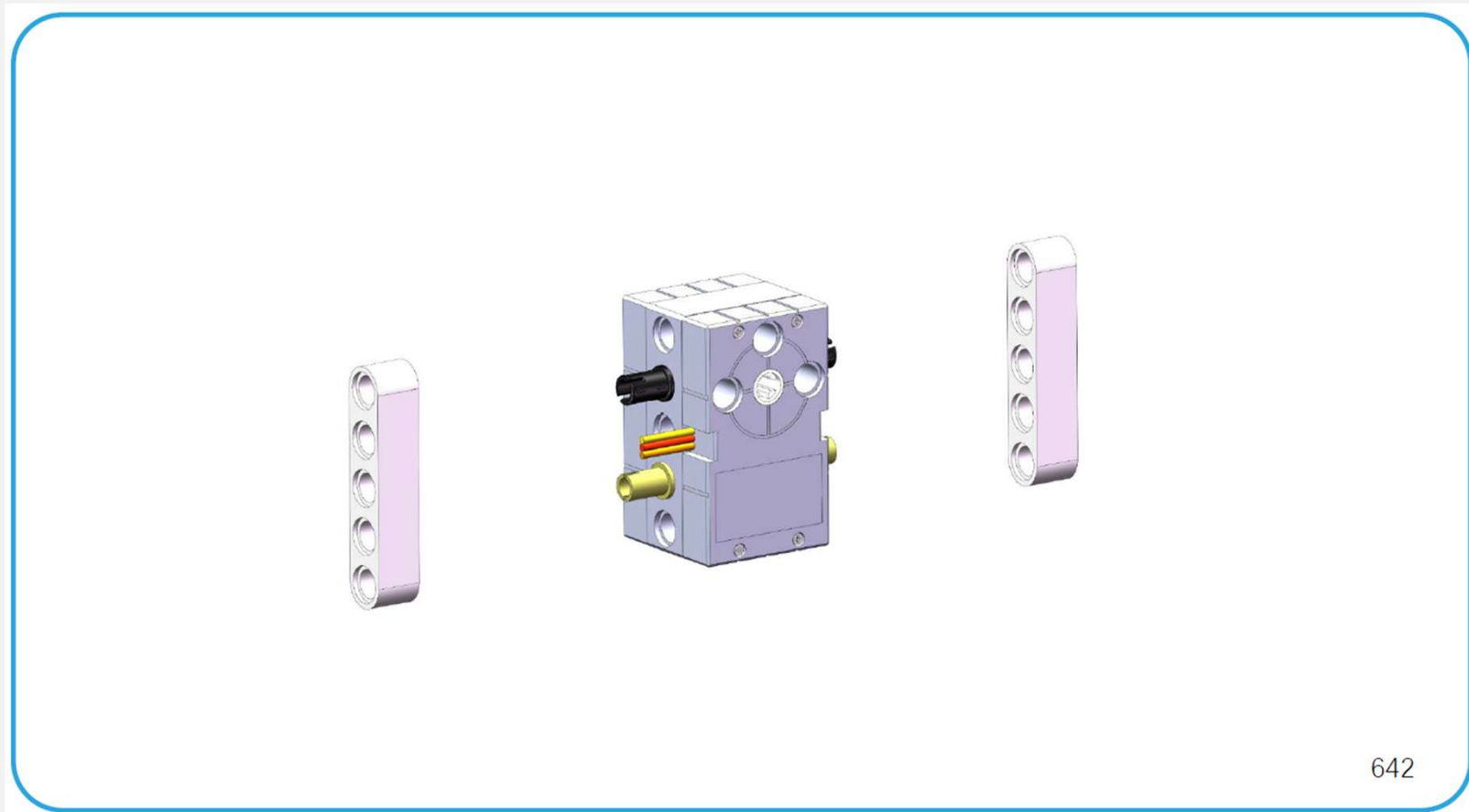
640

Step 29

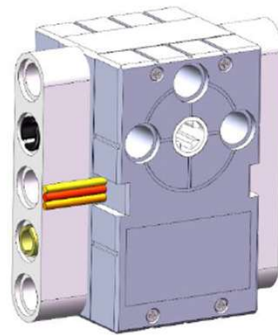


641

Step 30

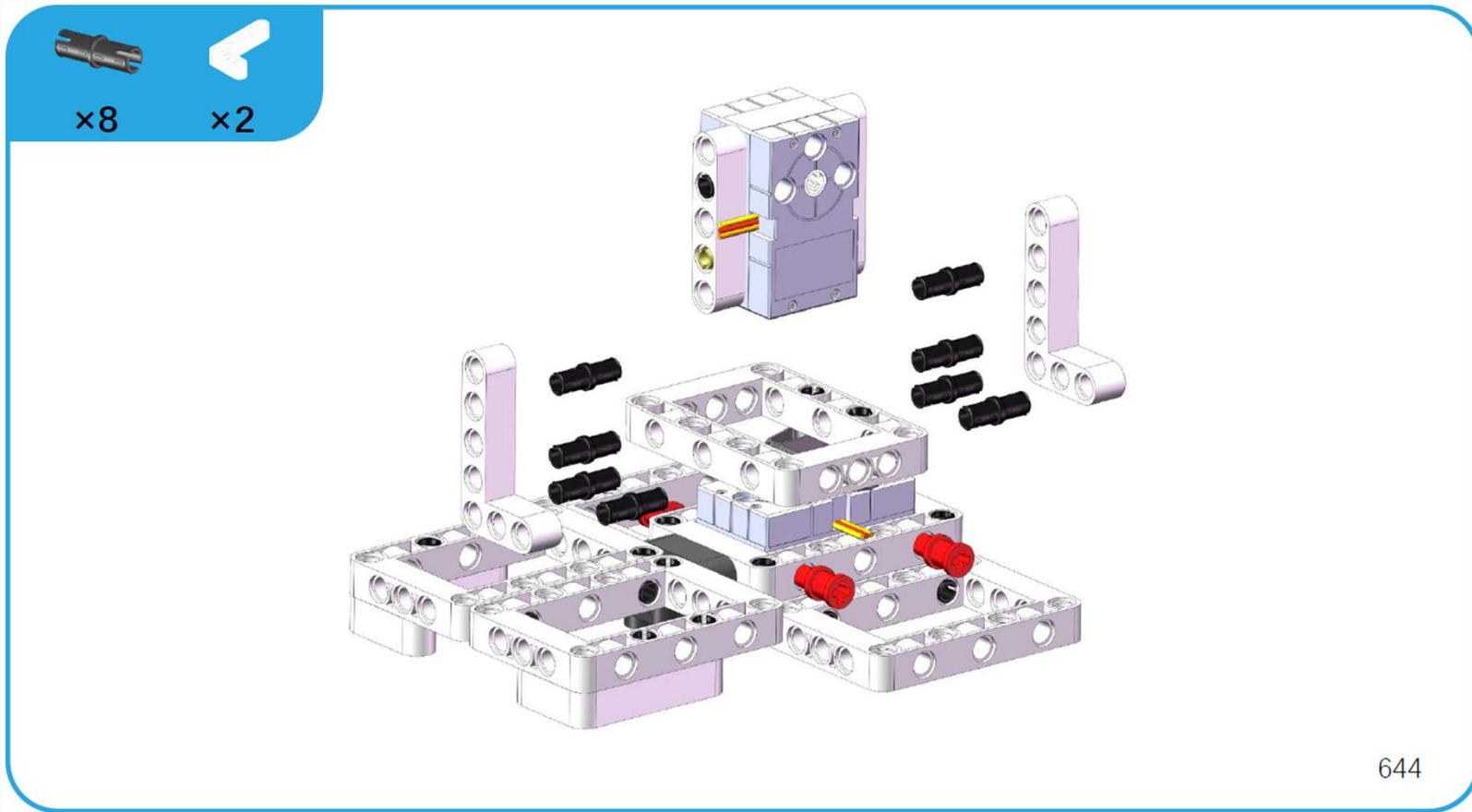


Step 31

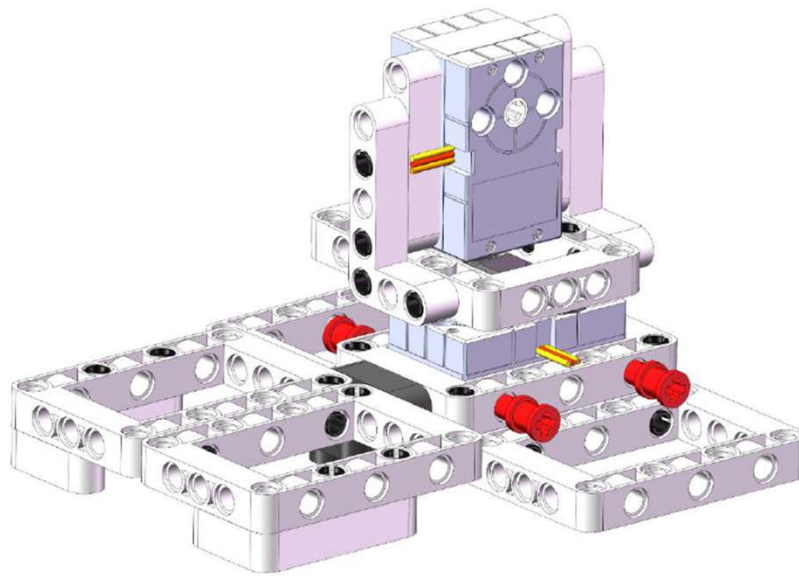


643

Step 32

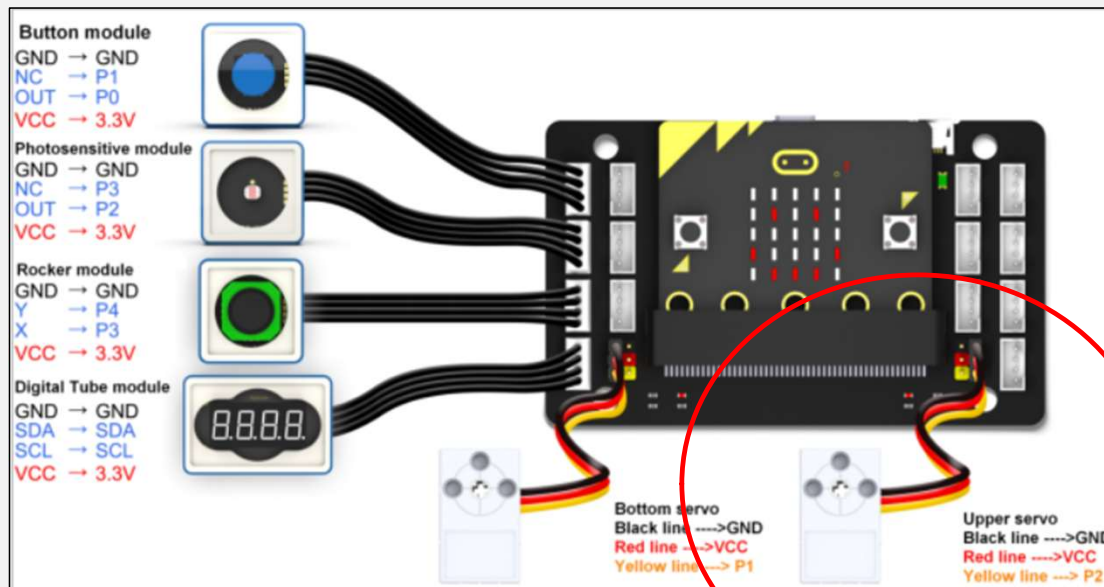


Step 33



647

Step 33 – Calibrate your Servo P2 (Upper servo)



Connect your servo to P2 on your expansion board, this servo is the upper servo which will hold the photosensitive module.

Before continuing the next building step, we need to calibrate the servo and make sure it's in the correct position.

Step 33 – Calibrate your Servo P2 (Upper Servo)



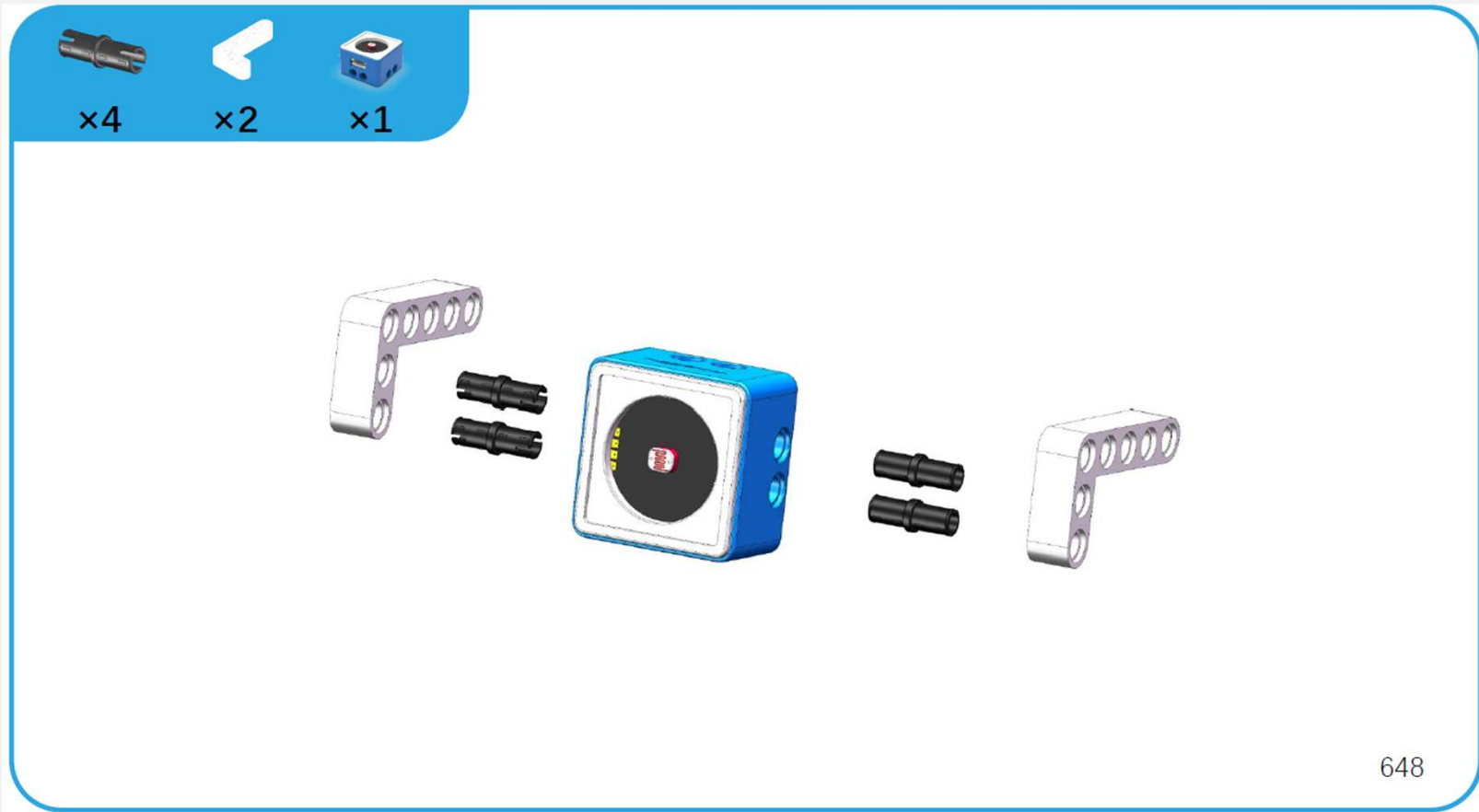
Then go to makecode and set the servo (360) pin P2 value to 230.

Remember to add the extension:

[Yahboomtechnology/module-world](https://www.scratch.mit.edu/resources/view/yahboomtechnology-module-world/10000)

Then download and make your servo go to position 230.

Step 34



648

Step 35



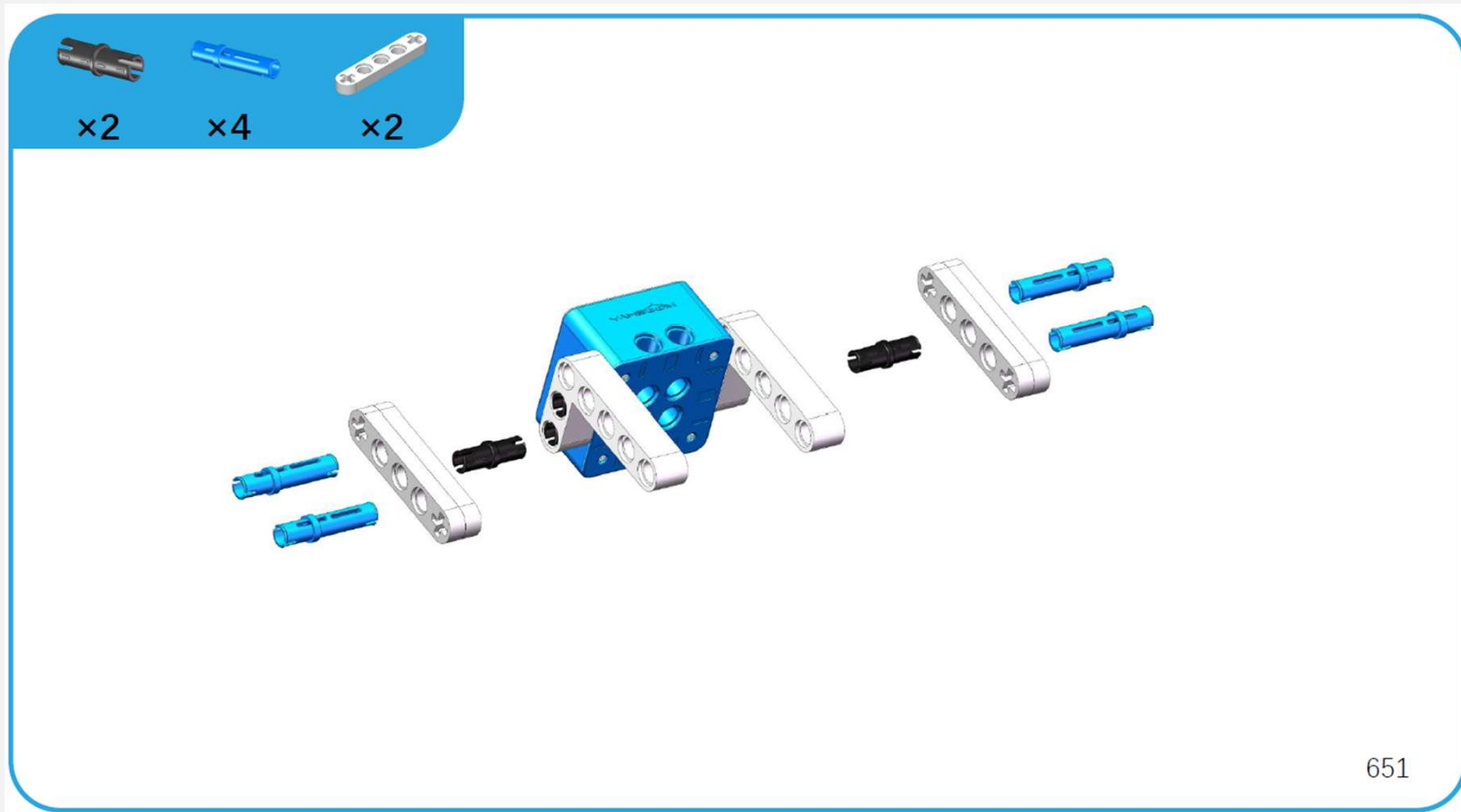
649

Step 36



650

Step 37

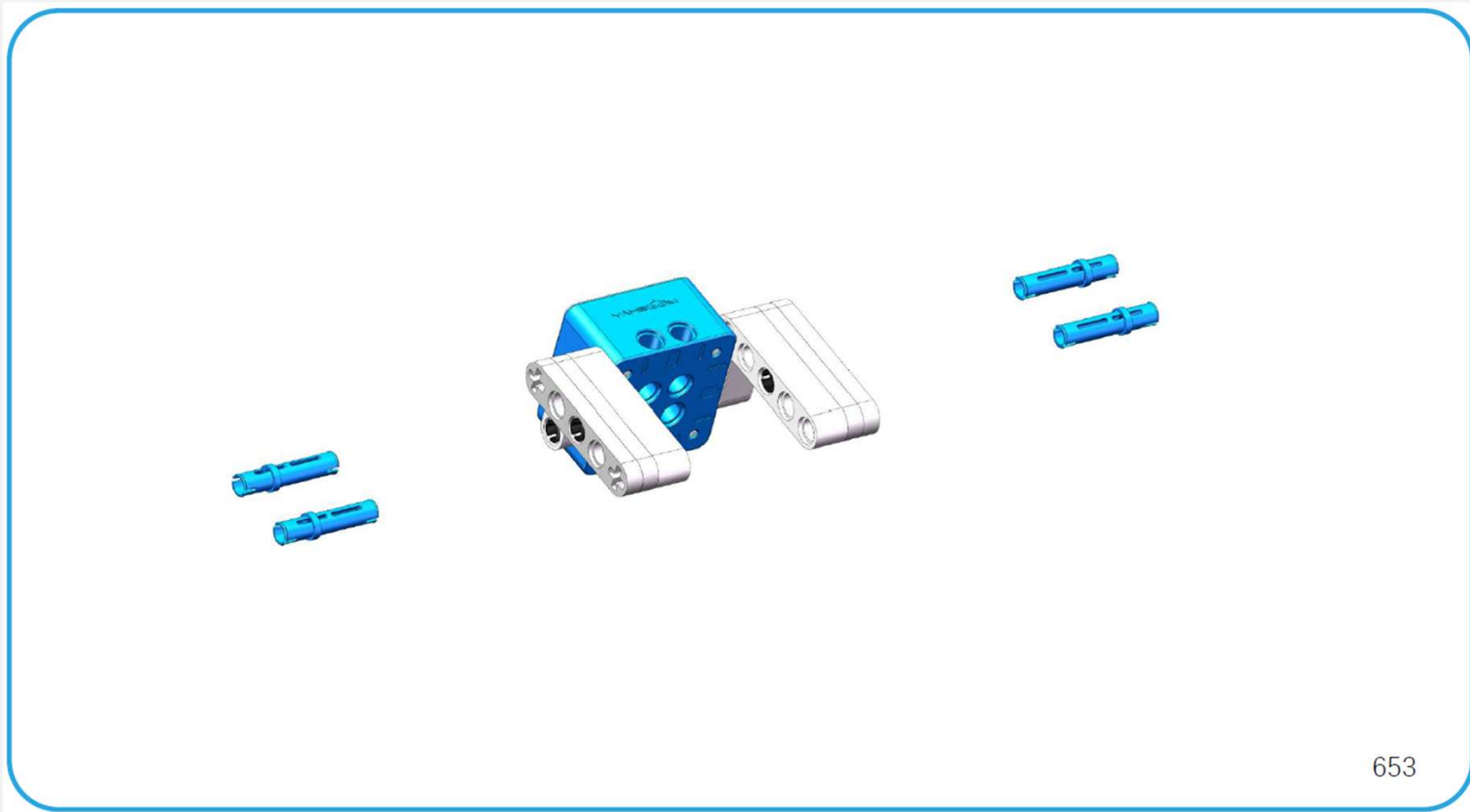


Step 38

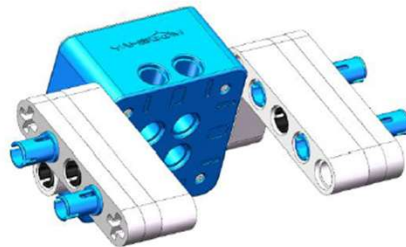


652

Step 39

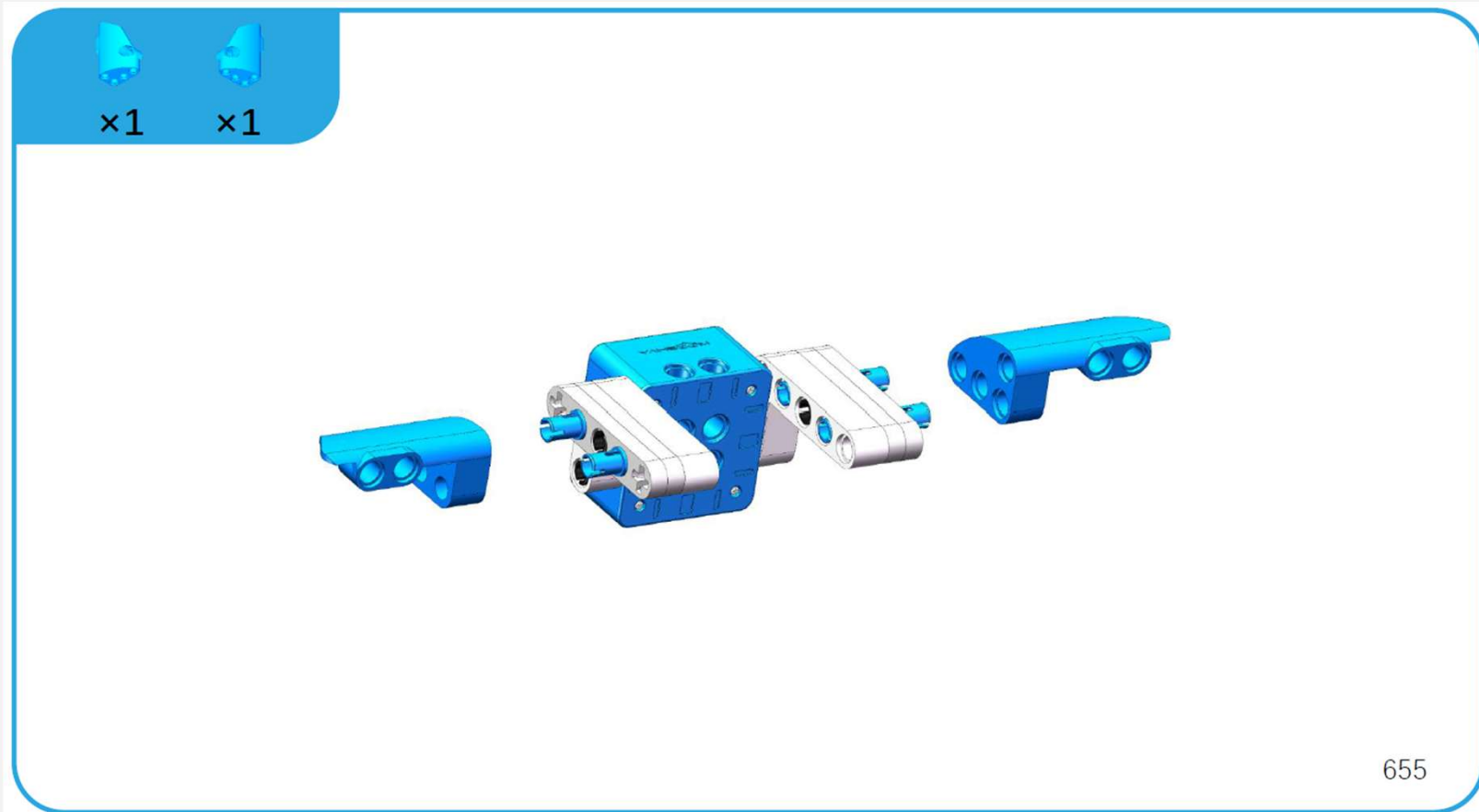


Step 40



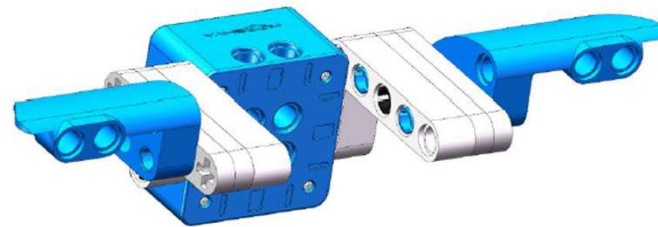
654

Step 41



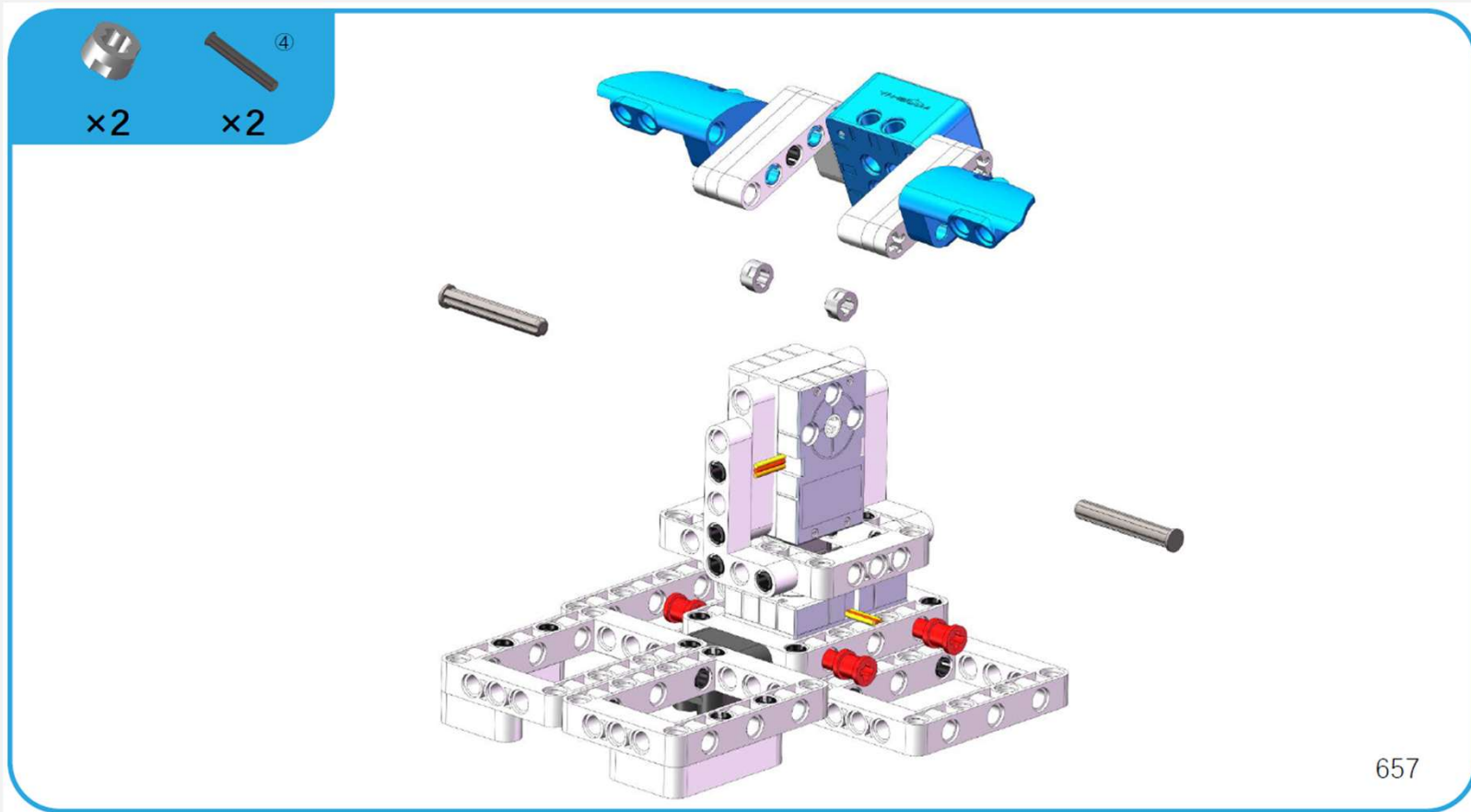
655

Step 42



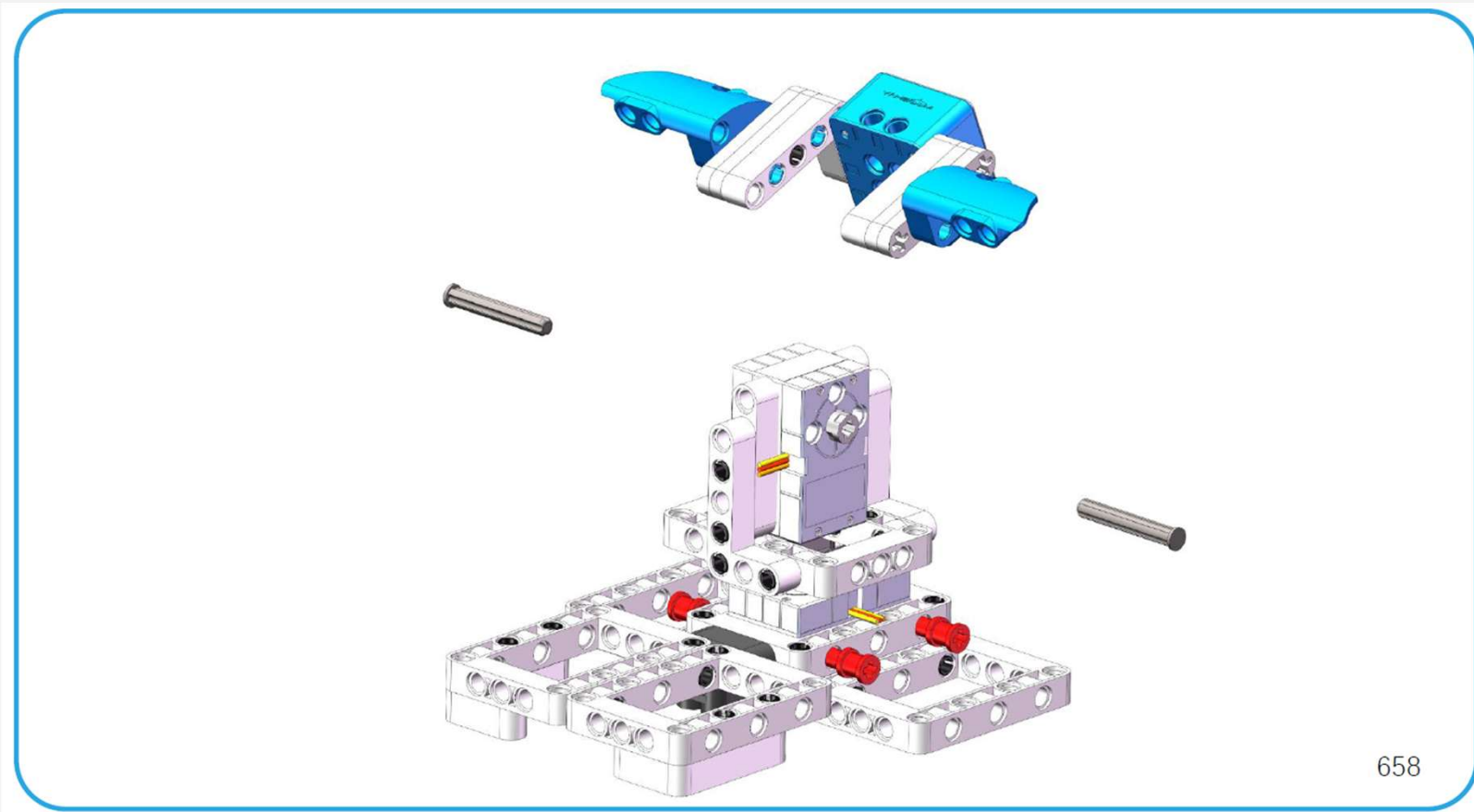
656

Step 43

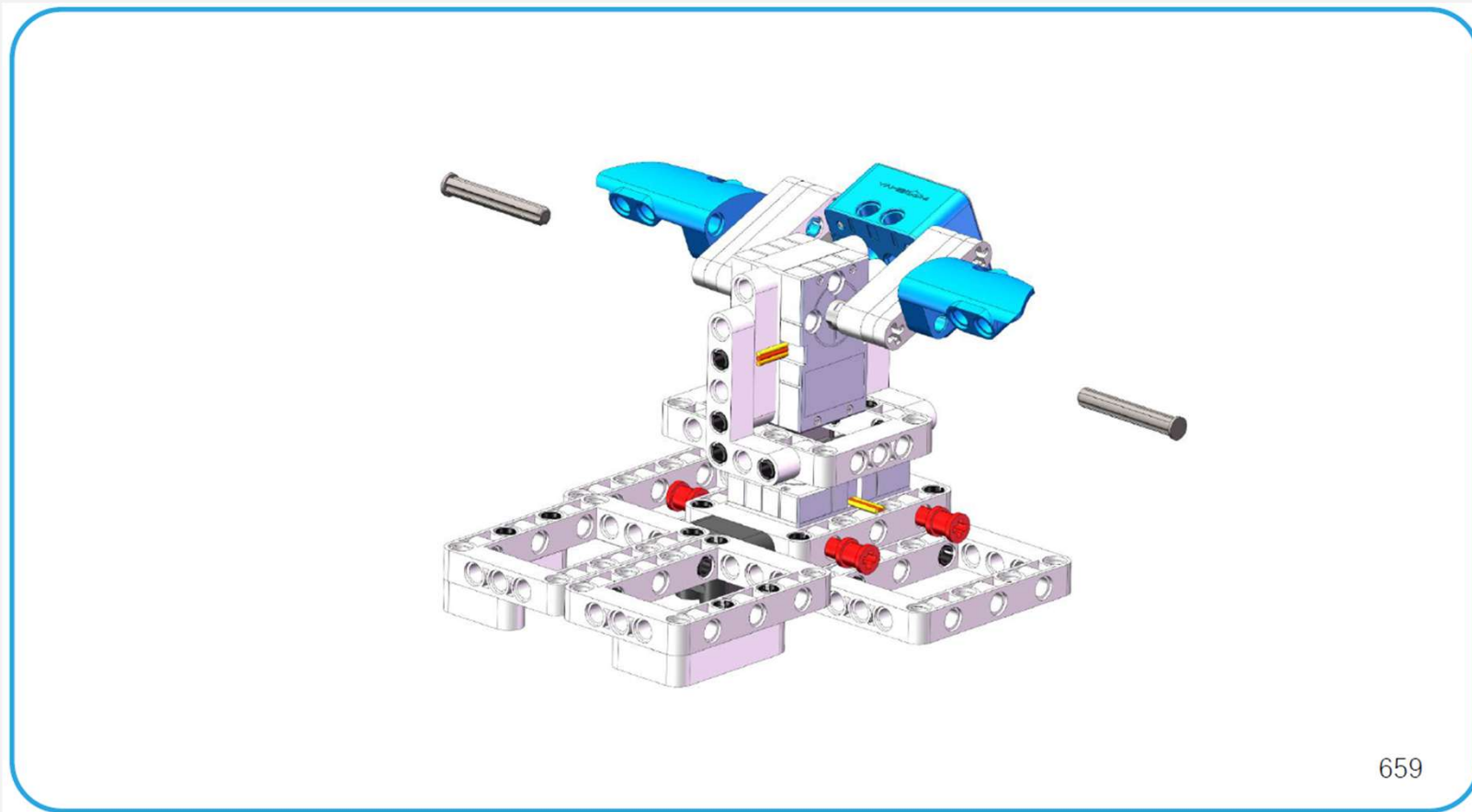


657

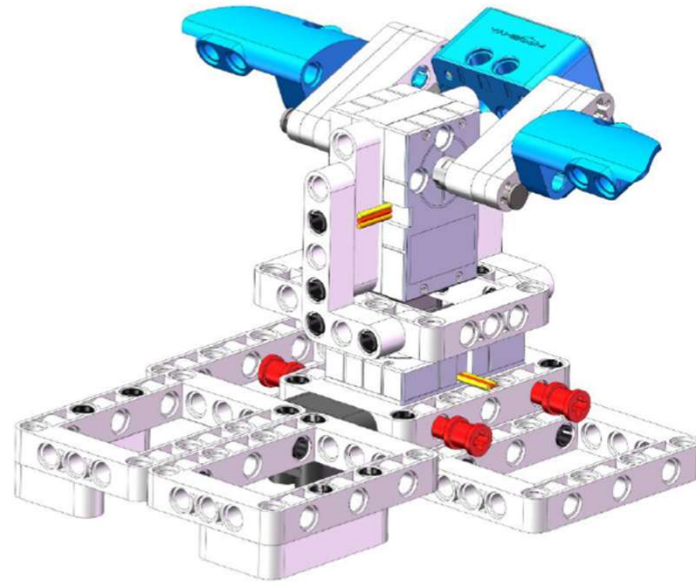
Step 44



Step 45



Step 46



660

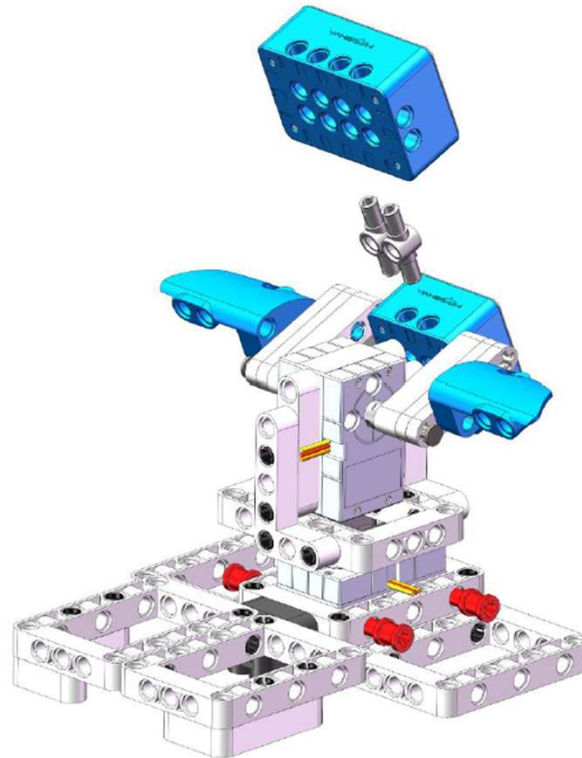
Step 47



x2

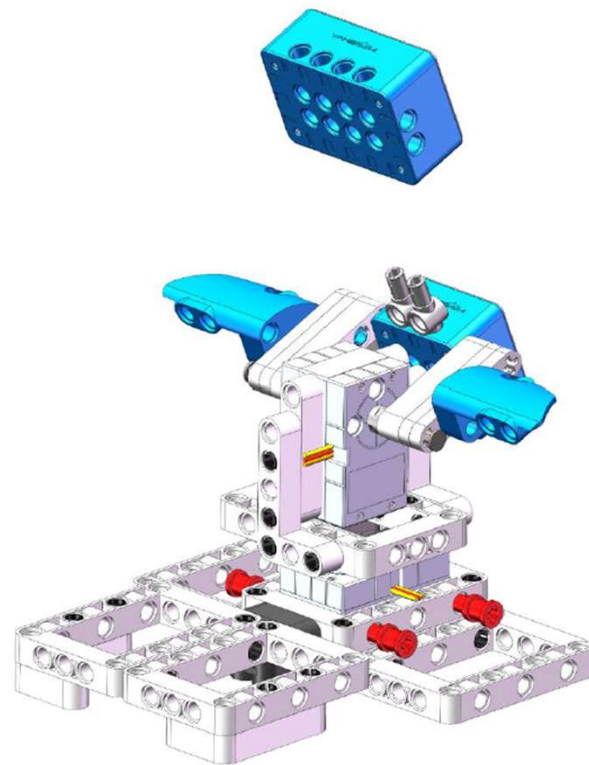


x1



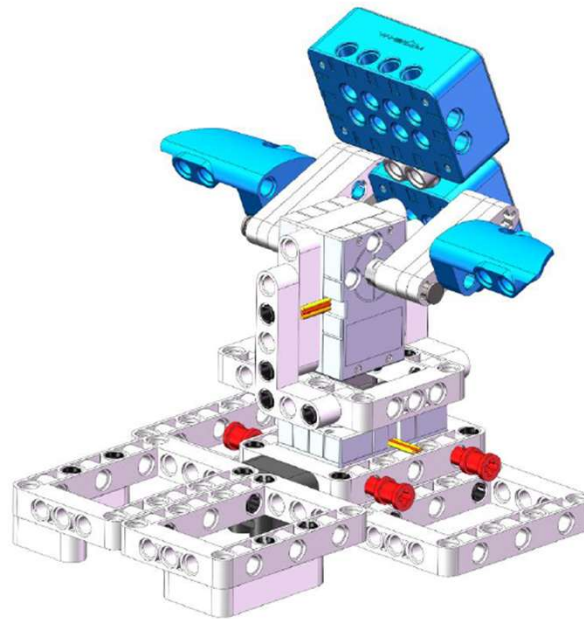
661

Step 48



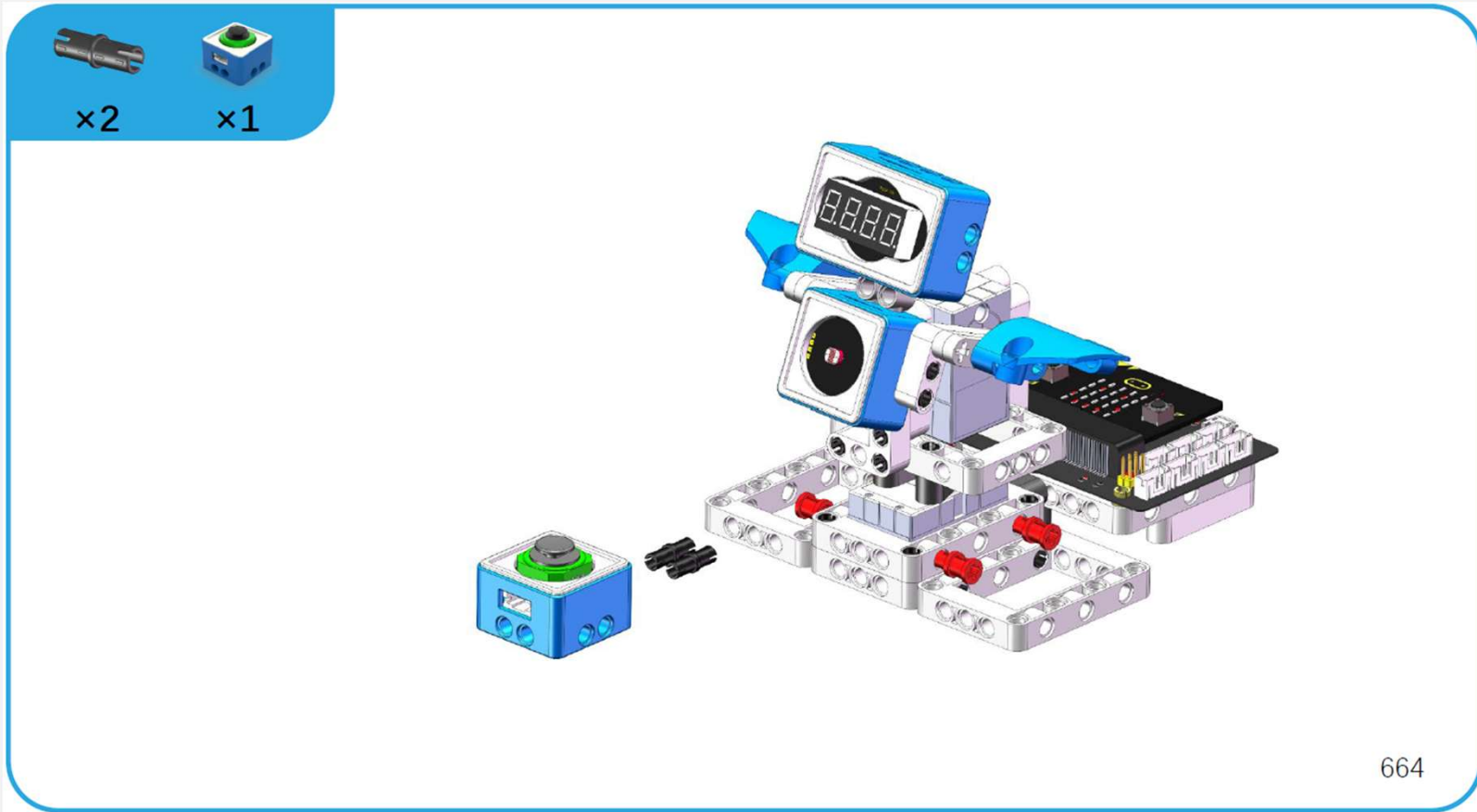
662

Step 49



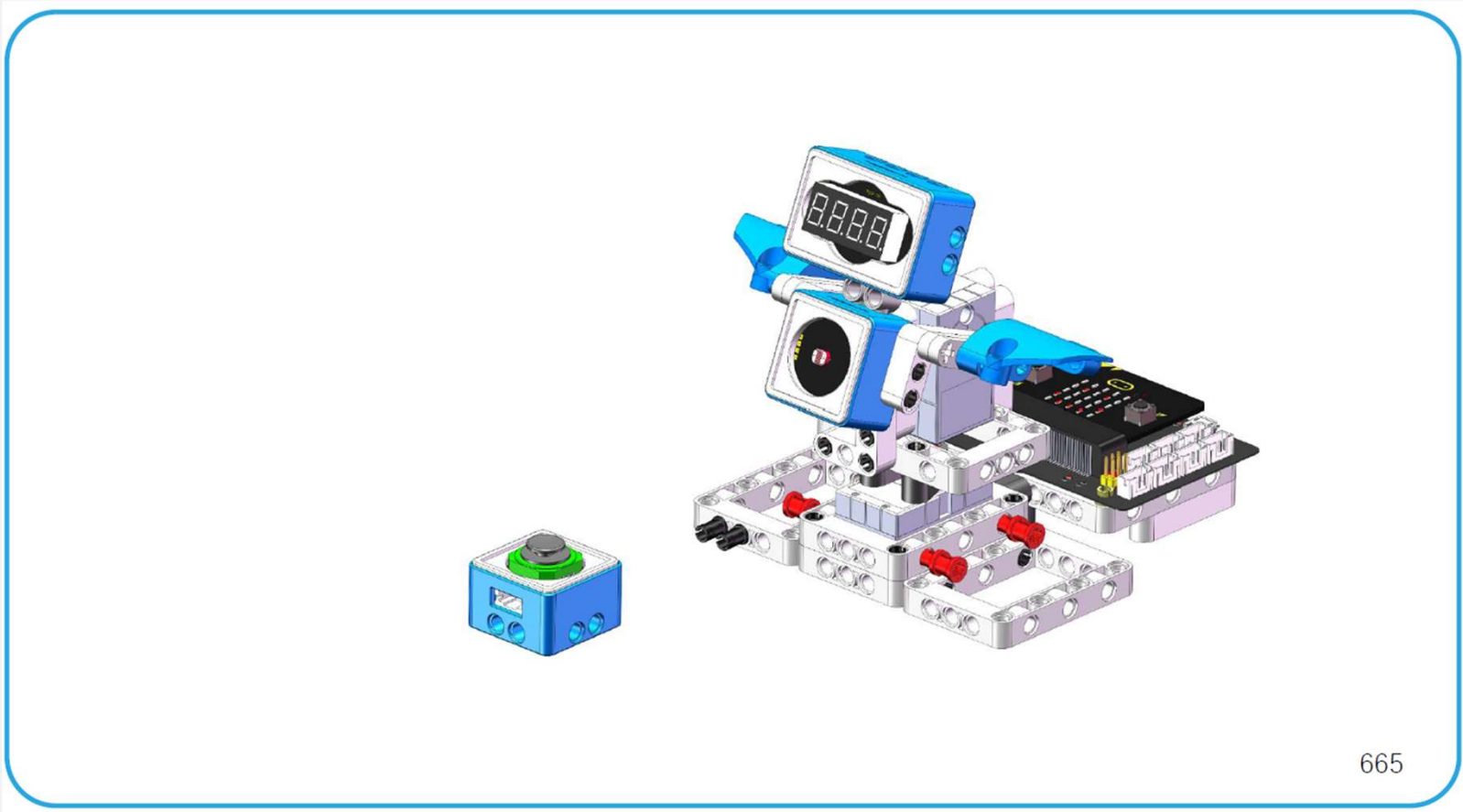
663

Step 50



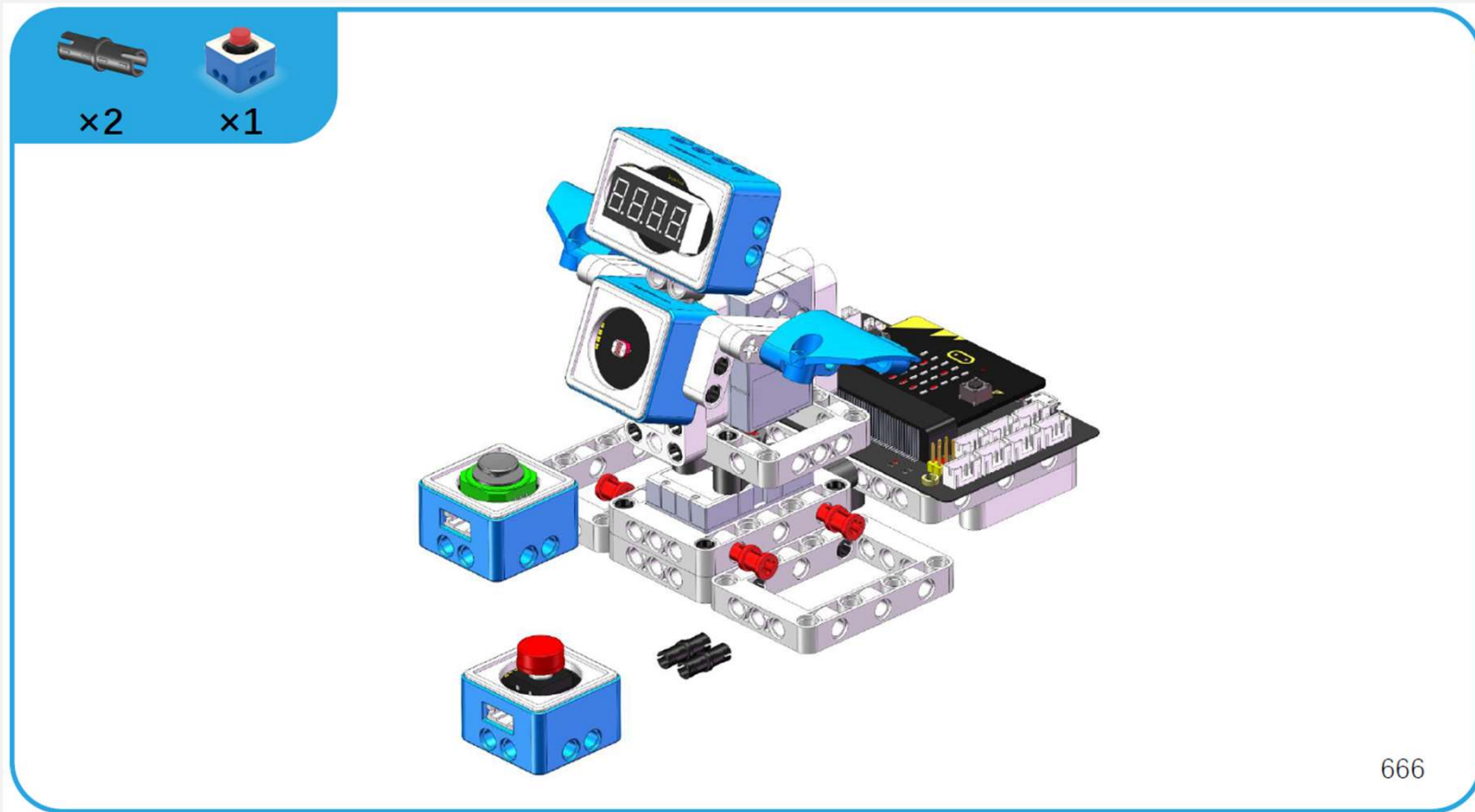
664

Step 51



665

Step 52

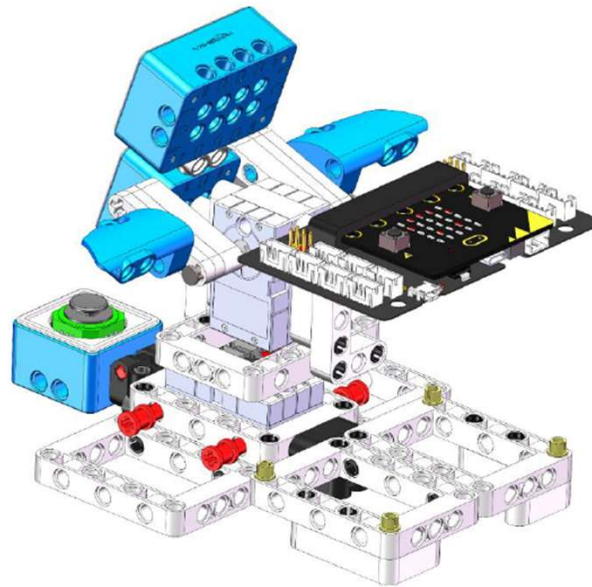


Step 53



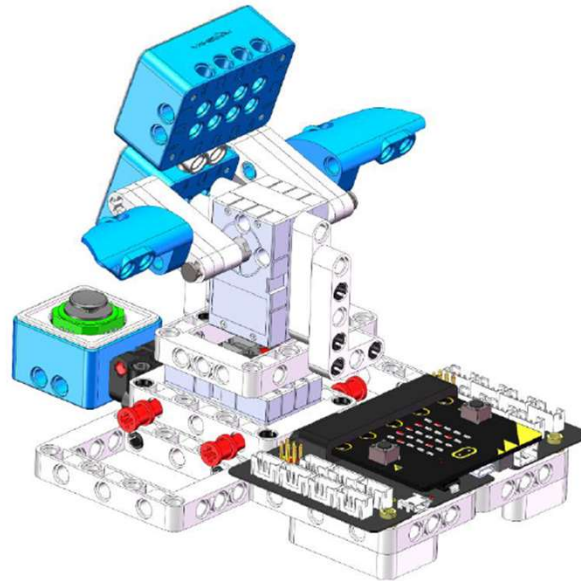
668

Step 54



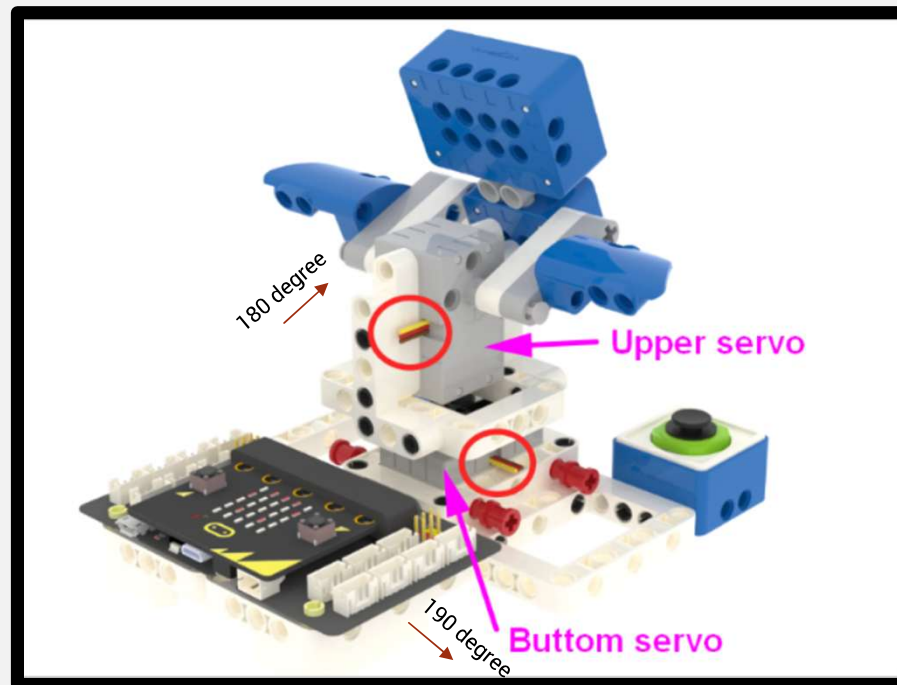
676

Step 55



677

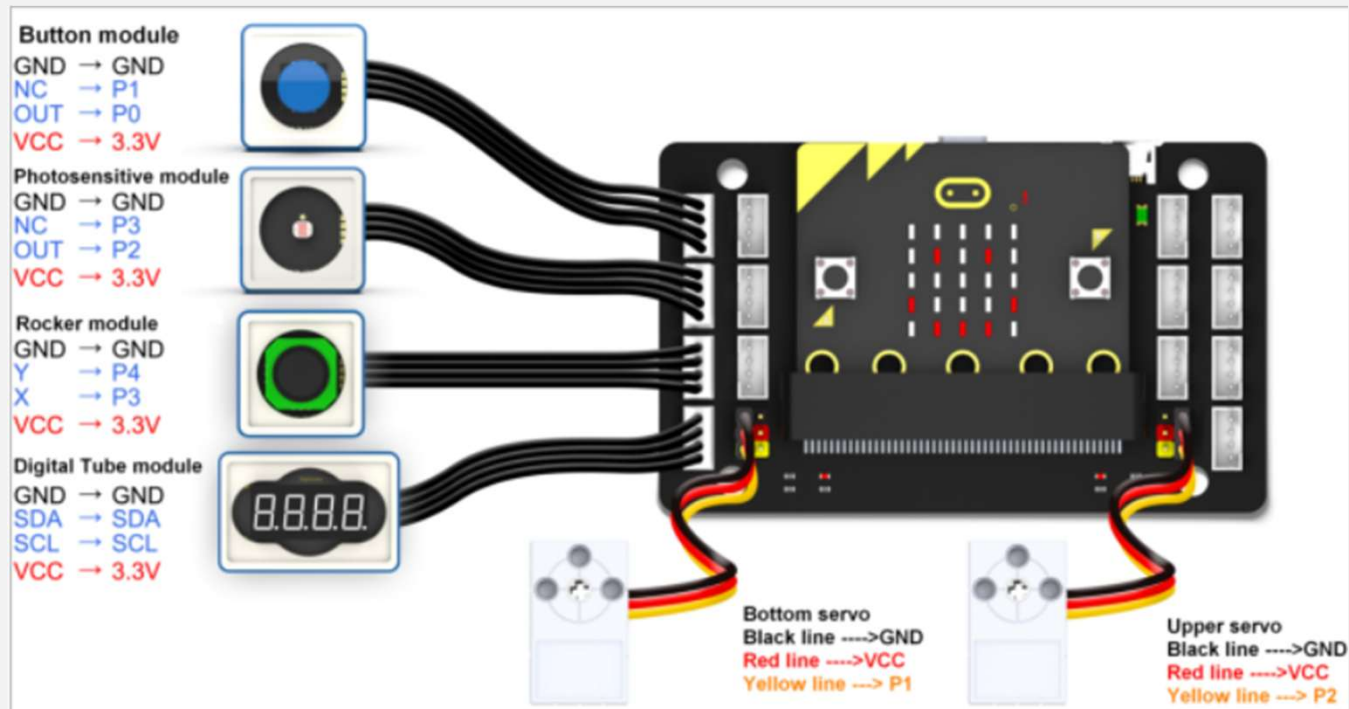
Please make sure that the wiper shaft is installed **correctly**, as shown below.



30 Points

Wire Connection

Connect the modules



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

10 Points

MakeCode Programming

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

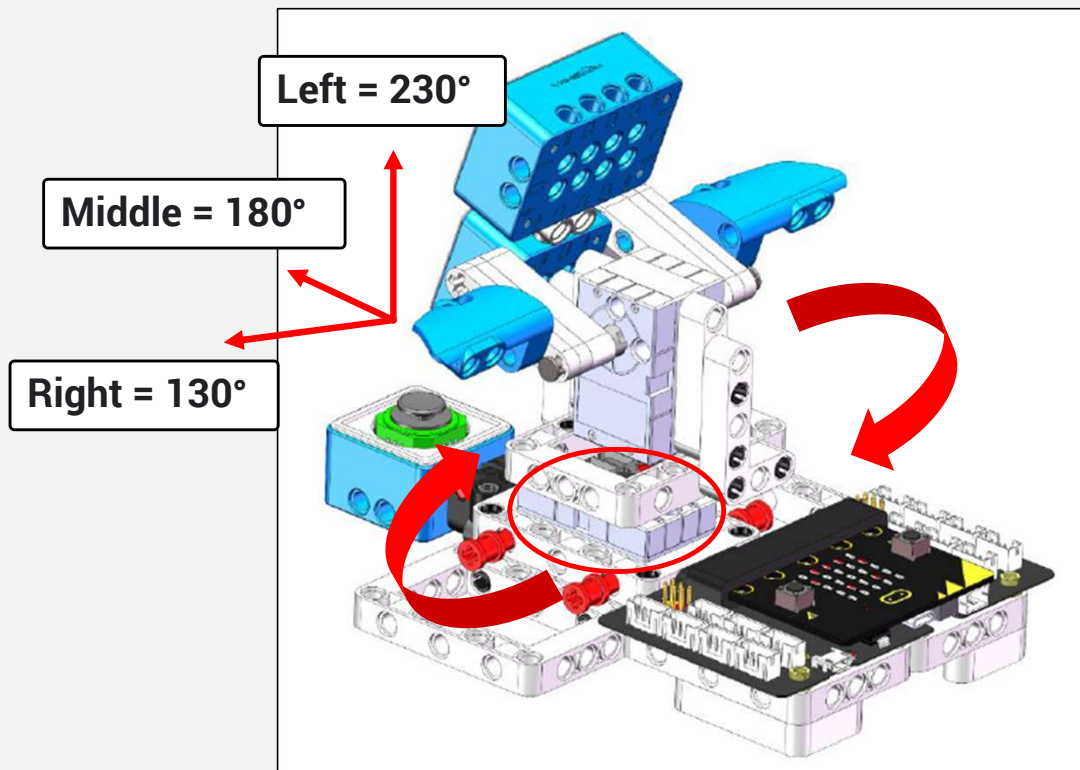
And

<https://github.com/YahboomTechnology/tm1650>

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

and search [YahboomTechnology/tm1650](https://github.com/YahboomTechnology/tm1650) in the extension block

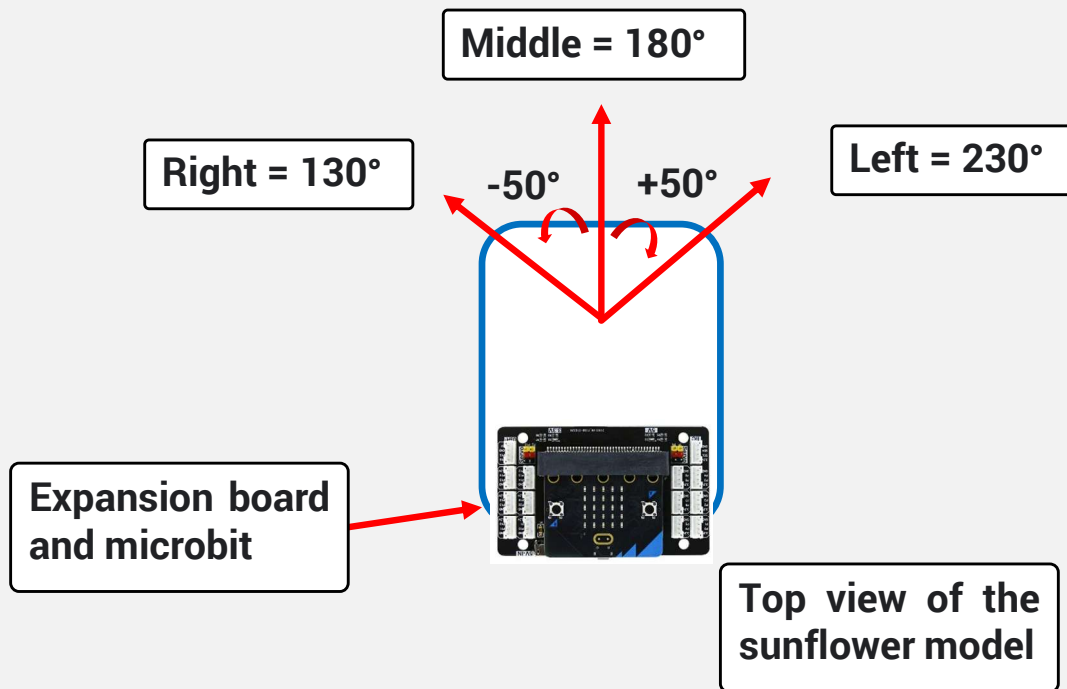
Before Coding – Understand Bottom Servo’s position



Before you start your coding, first we need to understand what are the positions we want to set for our bottom servo.

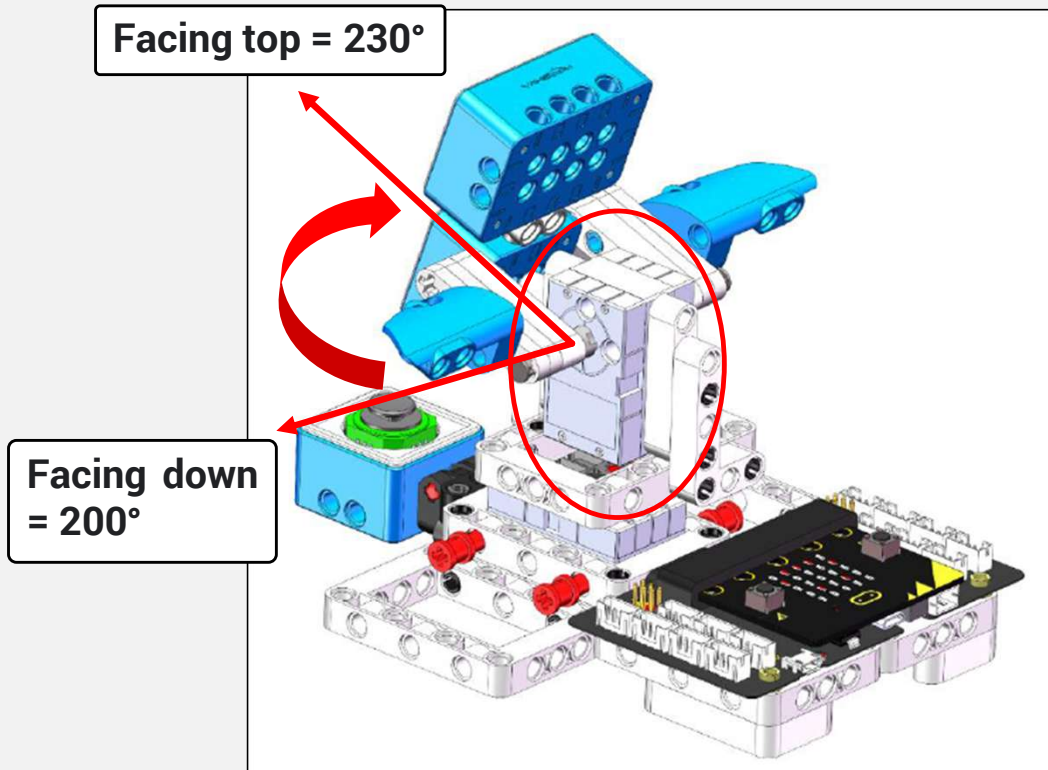
As shown at the left, the middle is to point towards opposite of our microbit, which is when servo position is 180°. Where Right is with 130°, and left with 230°

Before Coding – Top View



This is the top view of the sunflower model, we set the model's position at min = 130° (right) and max = 230° (left), where starting position set at the middle = 180°.

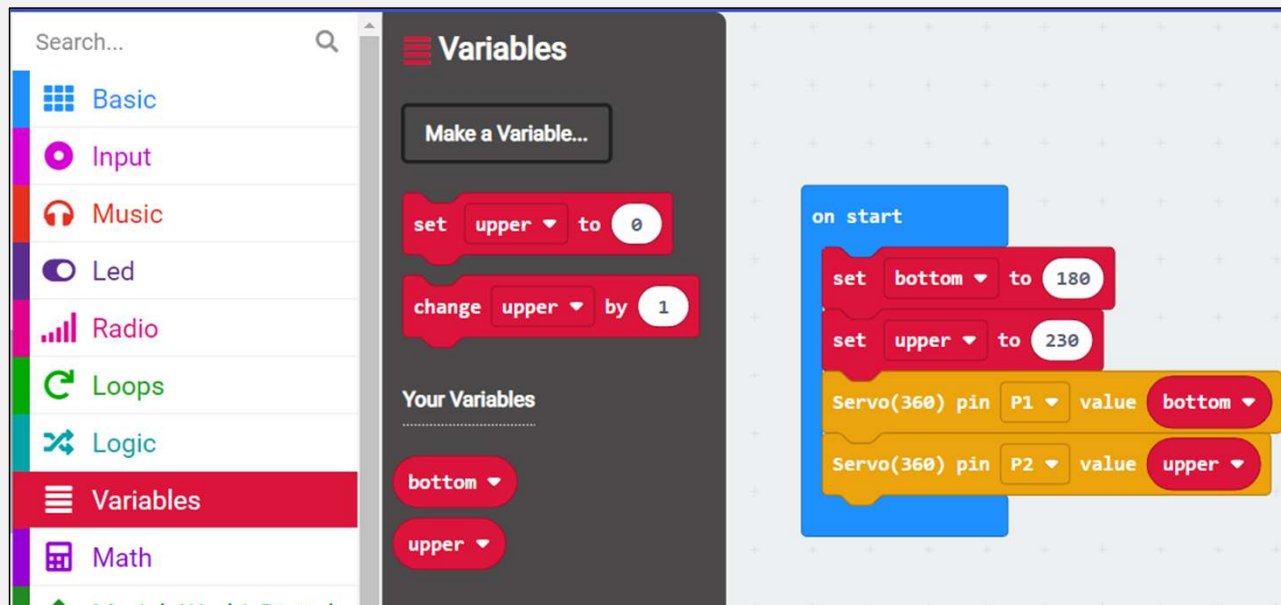
Upper Servo – Turning up and down



For upper servo, it is to turn the sunflower facing upward or downward.

So, we will set the upper limit to 230° (Facing top), and lower limit to 200° (Facing down).

Coding – Variables



We will start with setting the starting position for our sunflower model, let's make variables for:

1. bottom
2. upper

Then set the bottom to 180 and upper to 230, then link the variable to your servos on start.

Coding – On Start

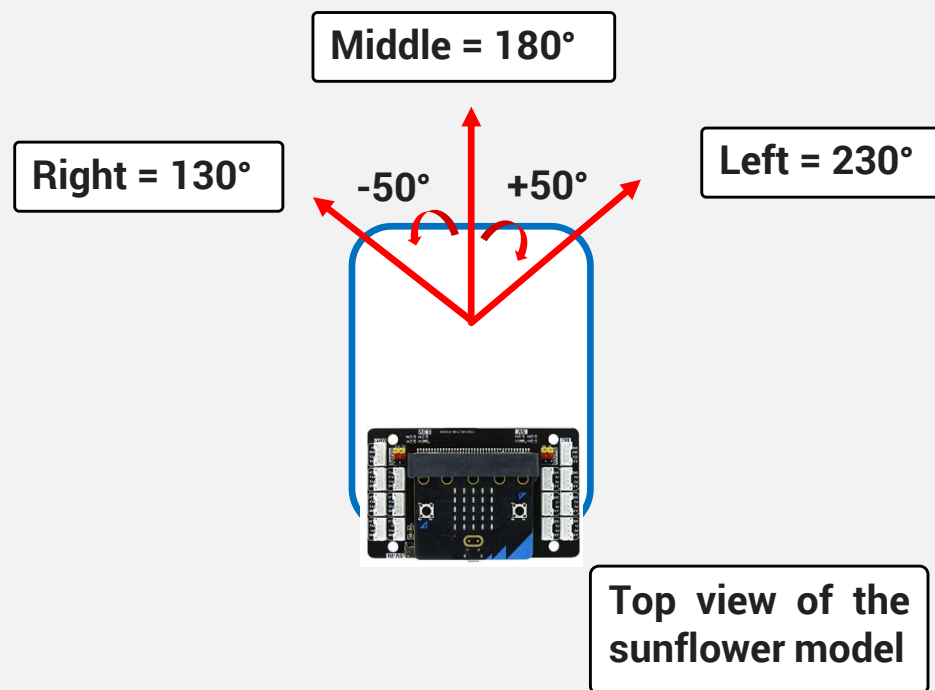
```

on start
  turn on display
  set bottom to 180
  set upper to 230
  Servo(360) pin P1 value bottom
  Servo(360) pin P2 value upper
  pause (ms) 500
  
```

Let's turn on the digital display at start (this coding block will require [yahboomtechnology/tm1650 extension](https://www.yahboomtechnology.com/tm1650-extension/)).

And let's wait for 0.5 second before entering the forever loop.

Purpose – Tracking the brightest position

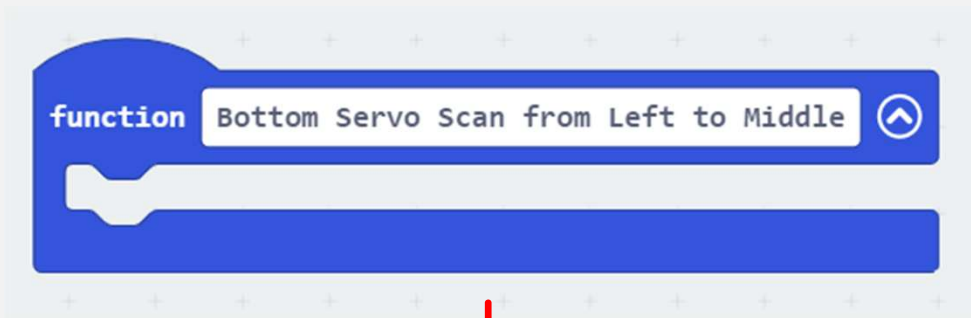


Now the purpose of this sunflower is it can track the brightest position out of the 3 positions (Right, Middle and Left), so when we press the button, the sunflower will scan from left to right.

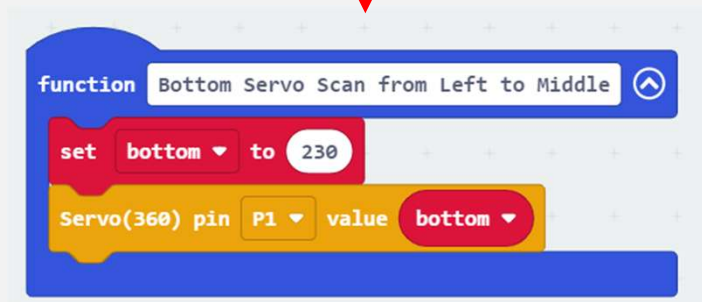
The photosensitive module on top will start collecting the value of brightness when the sunflower is turning.

After scanning, the bottom servo will choose the highest value and turn to that position.

Create Function



To make the coding simple and clean, I will create a function and call it "Bottom Servo Scan from Left to Middle"



Then make the bottom servo go to the left position first (which is 230°)

Bottom Servo Scanning – Left to Middle

```

function Bottom Servo Scan from Left to Middle
  set bottom to 230
  Servo(360) pin P1 value bottom
  pause (ms) 200
  while bottom > 180
  do
    change bottom by -1
    Servo(360) pin P1 value bottom
    pause (ms) 10
  
```

Before our servo start to move to middle, I put a pause for 0.2 second of time.

To make the servo slowly go to the middle from the left, we add in the while loop.

We keep reducing the position of the servo until 180°, so the bottom servo will move from 230° to 180° slowly.

The lower value of the pause time set in the while loop, the faster the servo will move from left to middle.

Bottom Servo Scanning – Middle to Right

```

function Bottom Servo Scan from Middle to Right
  set bottom to 180
  Servo(360) pin P1 value bottom
  pause (ms) 200
  while bottom > 130
  do
    change bottom by -1
    Servo(360) pin P1 value bottom
    pause (ms) 10
  
```

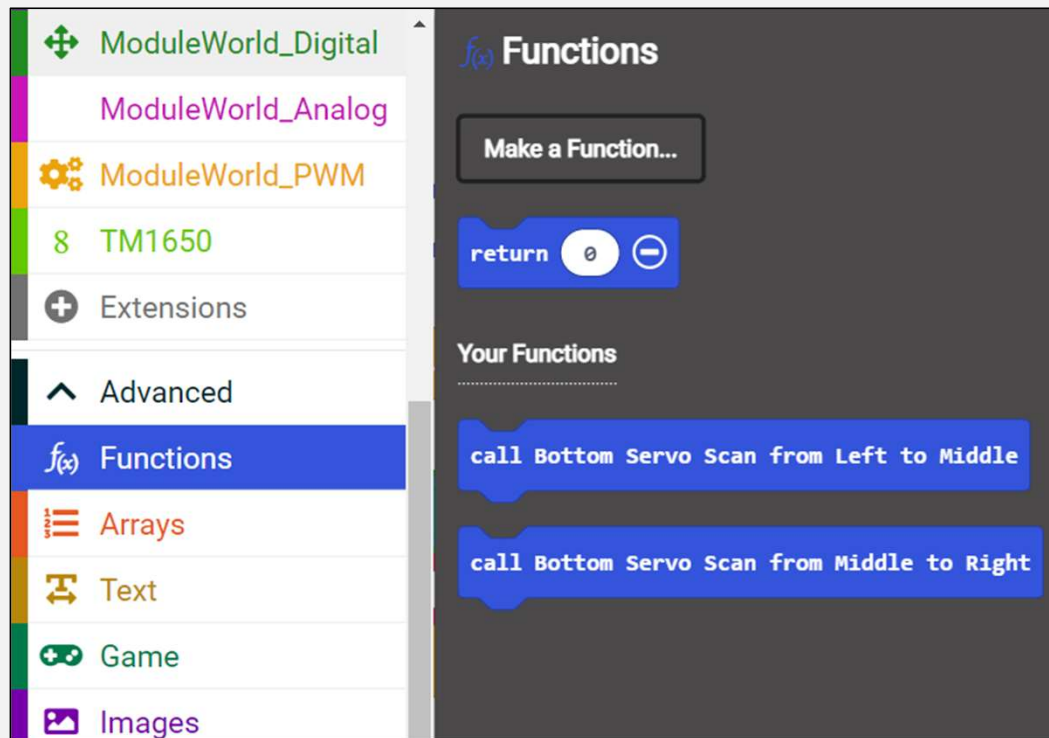
Then we need to set a function for Middle to right also.

Let's duplicate the function and change the function name to:
 "Bottom Servo Scan from Middle to Right"

Since it is now moving from 180° to 130°, so we need to change the value inside.

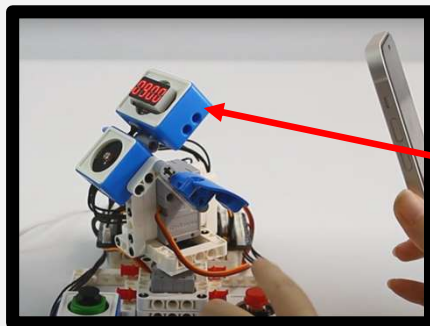
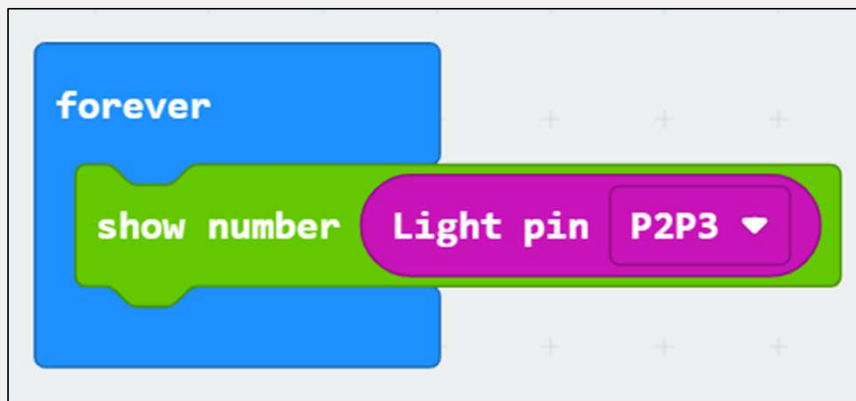
The change bottom by -1 is to keep reducing the position by 1 in each loop.

Bottom Servo Scanning – Middle to Right



Now you should have 2 functions block in your Functions category.

Checking Photosensitive Value



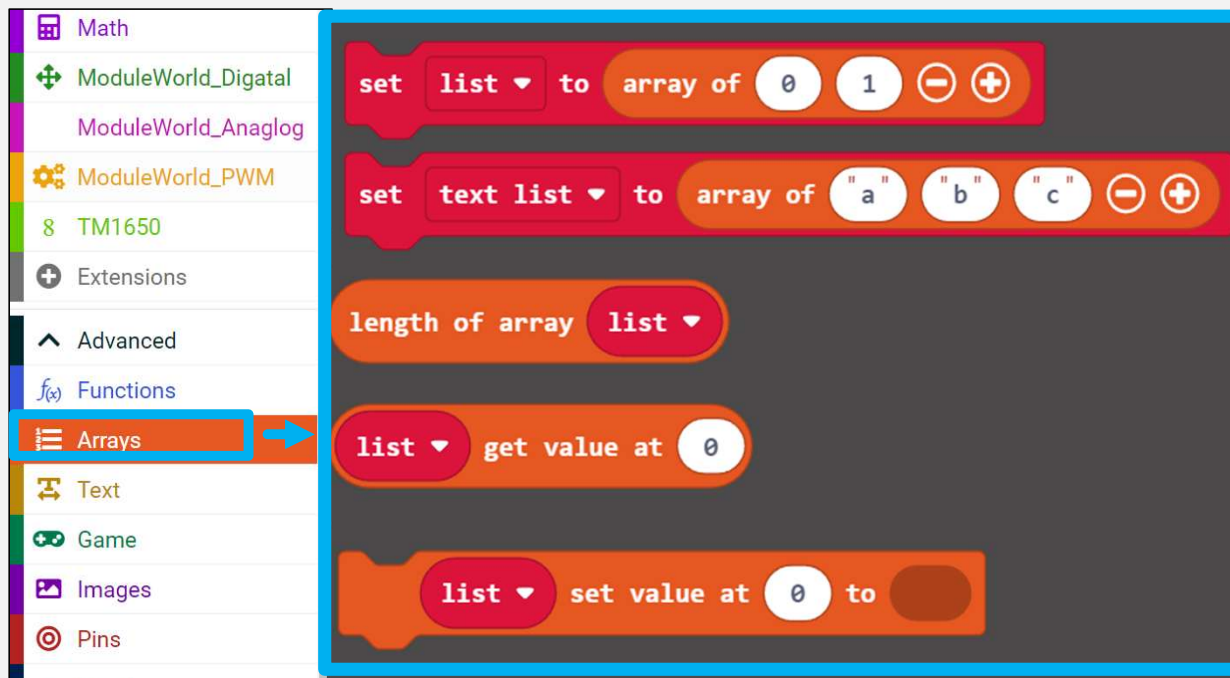
Now it's time to check the photosensitive value in each of the 3 positions.

We need to display the value in our digital tube.

Drag a show number from your tm1650 coding block into your forever block and put the light pin P2P3 inside the "show number".

After downloaded your program, you should see the value being displayed on the digital tube.

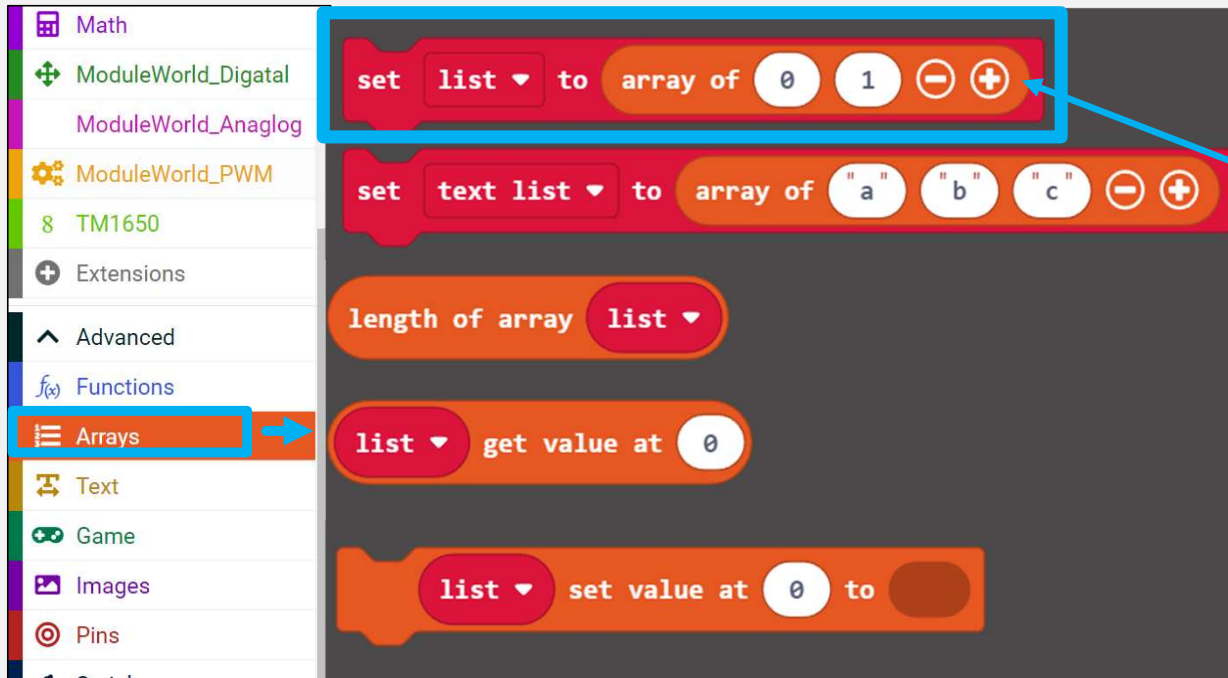
Coding – Arrays



Since we want to collect the photosensitive value from 3 different positions, and compare them to find out the highest value, then our servo will move to that position.

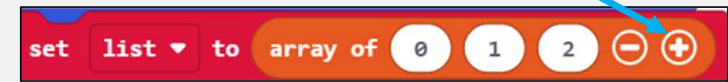
Hence, we need to record down these 3 value when the bottom servo is doing scanning from left to right.

Coding – Arrays



The image shows a Scratch code editor with a sidebar on the left containing categories like Math, ModuleWorld_Digatal, ModuleWorld_Analog, ModuleWorld_PWM, TM1650, Extensions, Advanced, Functions, Arrays, Text, Game, Images, and Pins. The 'Arrays' category is highlighted with a blue arrow. The main workspace contains several code blocks: a 'set list to array of' block with values 0, 1, and a '+' button; a 'set text list to array of' block with values 'a', 'b', 'c', and '+' buttons; a 'length of array list' block; a 'list get value at 0' block; and a 'list set value at 0 to' block.

To take down the values in number into list for comparison later, we need this and we want array of 3 values, means we need click the “+” button to add another value.



A Scratch code block: 'set list to array of' with values 0, 1, 2, and '+' buttons.

Coding – Add new function



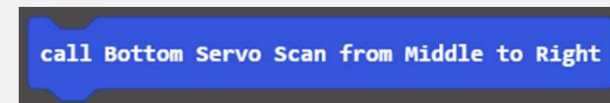
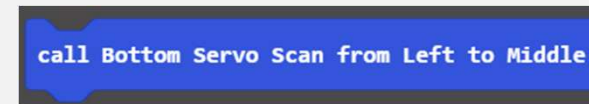
To make our single script short, we can make a new function called “ Scan and record photosensitive value”

Whatever coding that make the sunflower to scan and check the photosensitive values we can put inside.

Pseudocode

Steps to take down the values in 3 positions:

1. We want to set a memory list for taking down 3 different values
2. Make the bottom servo go to left position first (230°)
3. Take in the photosensitive value and put it to first item of the list.
4. Make the bottom servo go from Left to Middle (180°)
5. Take in the photosensitive value and put it to second item of the list.
6. Make the bottom servo go from Middle to Right (130°)
7. Take in the photosensitive value and put it to third item of the list.



Pseudocode

Steps to take down the values in 3 positions:

1. We want to set a memory list for taking down 3 different values
2. Make the bottom servo go to left position first (230°)
3. Take in the photosensitive value and put it to first item of the list.
4. Make the bottom servo go from Left to Middle (180°)
5. Take in the photosensitive value and put it to second item of the list.
6. Make the bottom servo go from Middle to Right (130°)
7. Take in the photosensitive value and put it to third item of the list.



```

1 set list to array of 0 1 2
2 Servo(360) pin P1 value 230
3 list set value at 0 to Light pin P2P3
4 call Bottom Servo Scan from Left to Middle
5 list set value at 1 to Light pin P2P3
6 call Bottom Servo Scan from Middle to Right
7 list set value at 2 to Light pin P2P3
  
```

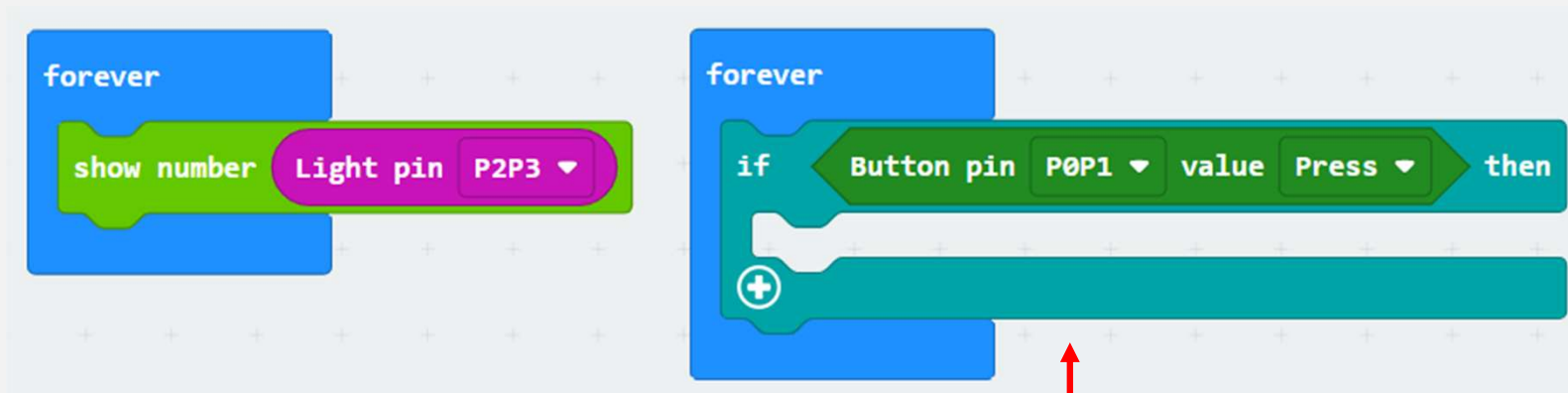
Scan and record down value

```

function Scan and record photosensitive value
  set list to array of 0 1 2
  Servo(360) pin P1 value 230
  pause (ms) 200
  list set value at 0 to Light pin P2P3
  call Bottom Servo Scan from Left to Middle
  pause (ms) 200
  list set value at 1 to Light pin P2P3
  call Bottom Servo Scan from Middle to Right
  pause (ms) 200
  list set value at 2 to Light pin P2P3
  
```

Let's make the sunflower to stay on each position steadily for 0.2 second before taking down the value. Then your "Scan and record photosensitive value" function should look like this.

Start Checking the light with button pressed



```

forever
  show number
  Light pin P2P3
forever
  if Button pin P0P1 value Press then
  
```

Let's drag out another forever loop and add the if statement for when button pin P0P1 is pressed.

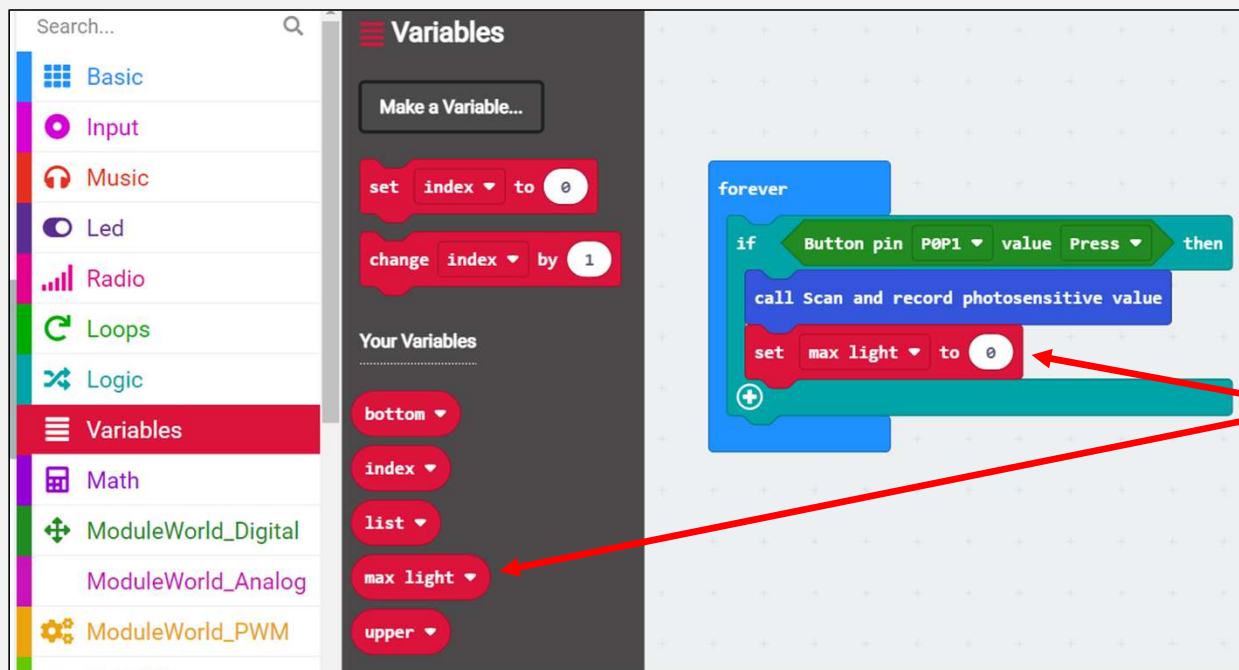
Start Checking the light with button pressed

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
  
```

This will only scan and record the values.
 But we still need to compare which is the highest and which position should the bottom servo turn to.

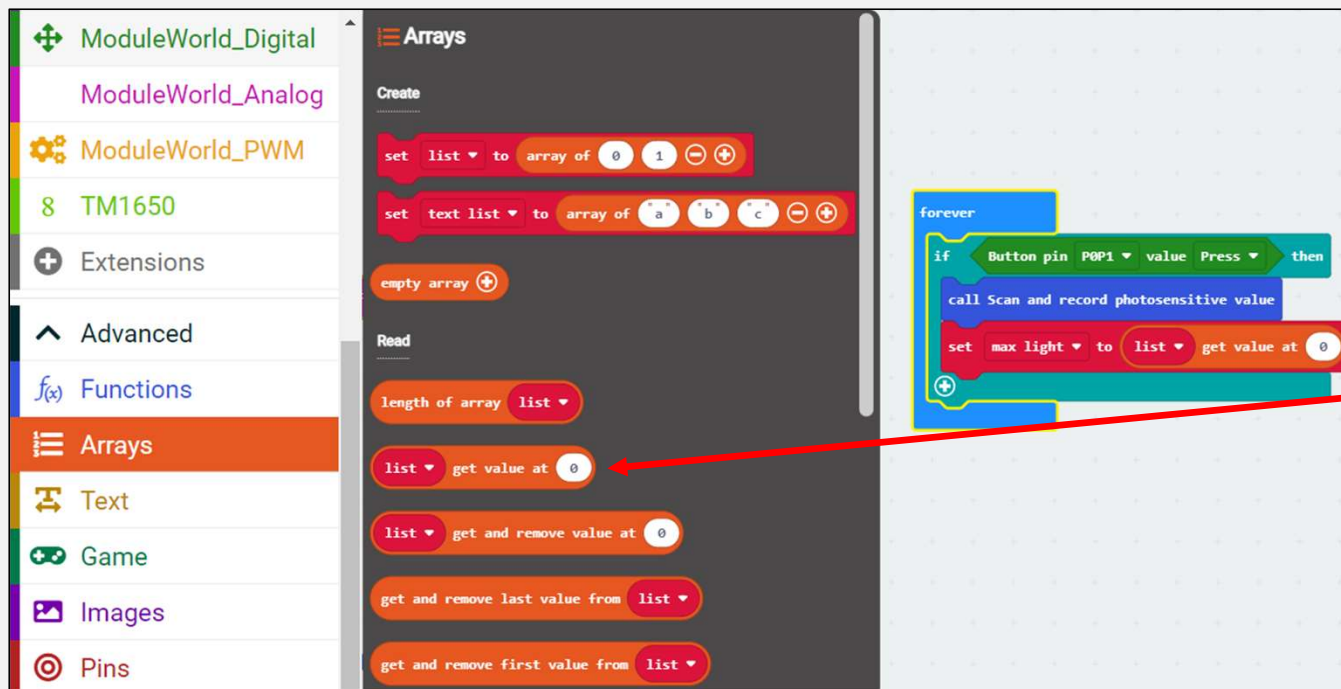
Start Checking the light with button pressed



The screenshot shows the Microbit IDE interface. On the left, the 'Variables' panel is open, displaying a list of variables: 'bottom', 'index', 'list', 'max light', and 'upper'. The 'max light' variable is highlighted. In the center, a code block is visible, consisting of a 'forever' loop. Inside the loop, there is an 'if' statement: 'if Button pin P0P1 value Press then'. The 'then' block contains two steps: 'call Scan and record photosensitive value' and 'set max light to 0'. A red arrow points from the 'set max light to 0' block in the code to the 'max light' variable in the panel. Another red arrow points from the 'max light' variable in the panel to the text box on the right.

We need to create a new variable called “max light”, which will temporarily take in the value from 1 of the array (photosensitive values), and we will keep compare each value until found the highest value of light we got, then we will store in “max light”.

Start Checking the light with button pressed



The screenshot shows the Microbit IDE interface. On the left, the 'Arrays' block is selected in the library. The main workspace contains the following code:

```

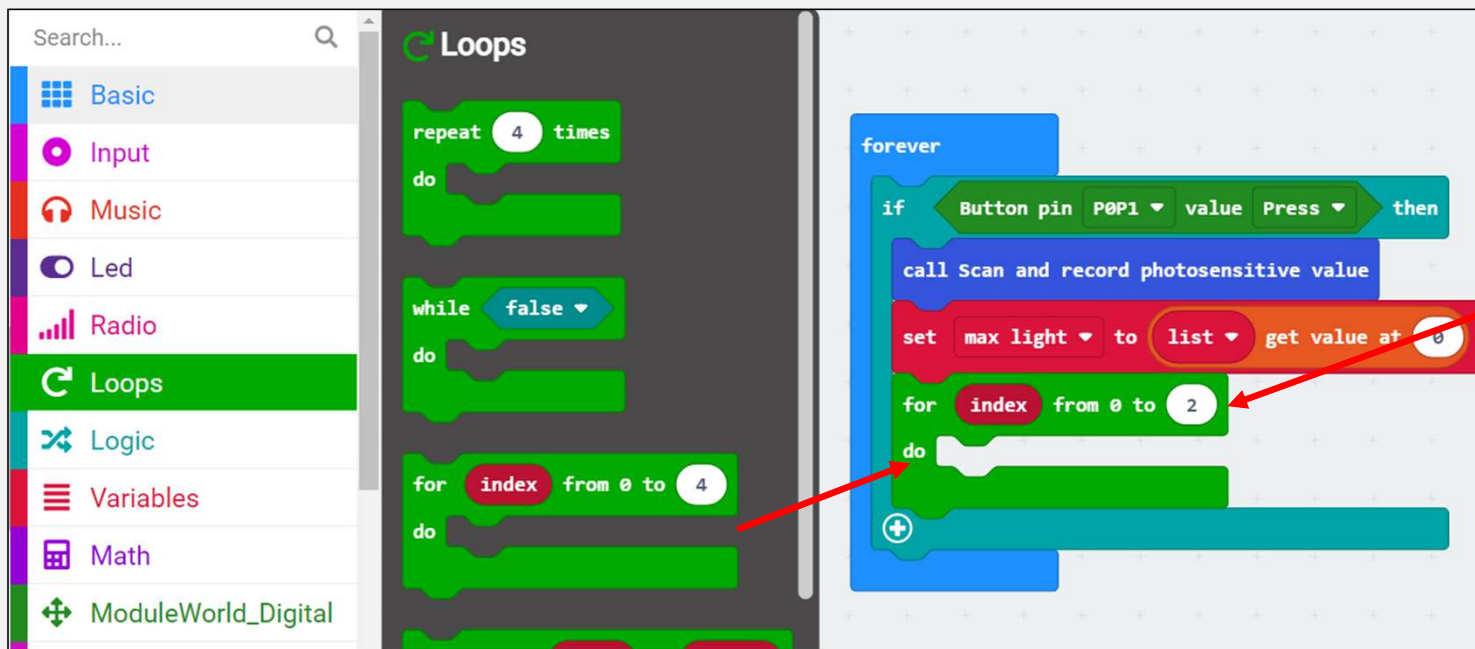
set list to array of 0 1
set text list to array of a b c
empty array

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
  
```

A red arrow points from the 'set max light to list get value at 0' block in the code to a text box on the right.

Now we want to assume the highest value of photosensitive comes from the first position, which is the value recorded when sunflower is at left position.

Start Checking the light with button pressed



The screenshot shows the Microbit IDE interface. On the left is a sidebar with a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops (highlighted), Logic, Variables, Math, and ModuleWorld_Digital. The main workspace is divided into two panes. The left pane shows a 'Loops' category with blocks for 'repeat 4 times', 'while false', and 'for index from 0 to 4'. The right pane shows a 'forever' loop block containing an 'if Button pin P0P1 value Press then' block, a 'call Scan and record photosensitive value' block, a 'set max light to list get value at 0' block, and a 'for index from 0 to 2' block. A red arrow points from the 'for' loop block in the code to a text box on the right.

Now we want to do comparison for 3 values from the list. So, we will do a for loop with index from 0 to 2.

Start Checking the light with button pressed

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if 0 > 0 then
          +
        +
      +
    +
  +

```

Each loop, we will take out the value to compare with our max light, if the value in that loop (we use index to track which value we are looking at) is more than max light, then we will replace the value stored in max light with the new value.

Start Checking the light with button pressed

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if list get value at index > max light then
          +
          +
  +
  +
  
```

So, if the loop is running for first round, which the index is 0, it will take the first value from the list to compare with the max light.

Then second loop will use the second value to compare with max light.

Third loop will use the third value to compare with the max light.

Start Checking the light with button pressed

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if list get value at index > max light then
          set max light to list get value at index
  
```

When doing the comparison in each loop, if the value being compared is higher than the max light, the max light will change its value to a new value (which is the higher value).

Example of highest value comparison

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if list get value at index > max light then
          set max light to list get value at index
  
```



In this case, the max light will be set to 756 before comparison starts.

Checking starts:

1st Loop -> 1st position (756) > max light (756) --- no change

2nd Loop -> 2nd position (826) > max light (756) --- max light = 826

3rd Loop -> 3rd position (652) > max light (826) --- no change

So, after the loop done running, the max light will become 826.

Set bottom servo position

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if list get value at index > max light then
          set max light to list get value at index
        end if
      end do
    end for
  end if
  if max light = 0 then
    Servo(360) pin P1 value 230
  else if max light = 1 then
    Servo(360) pin P1 value 180
  else if max light = 2 then
    Servo(360) pin P1 value 130
  end if
end forever
  
```

```

if max light = 0 then
  Servo(360) pin P1 value 230
else if max light = 1 then
  Servo(360) pin P1 value 180
else if max light = 2 then
  Servo(360) pin P1 value 130
end if
  
```

Since our max light has the value got from our photosensitive module, so it will be the values of the list, not the index from the list. We need to get the value from the list, not the index of the list.

Now we want our sunflower to run to the position which has the most light (highest photosensitive value), so we can refer our max light for the position.

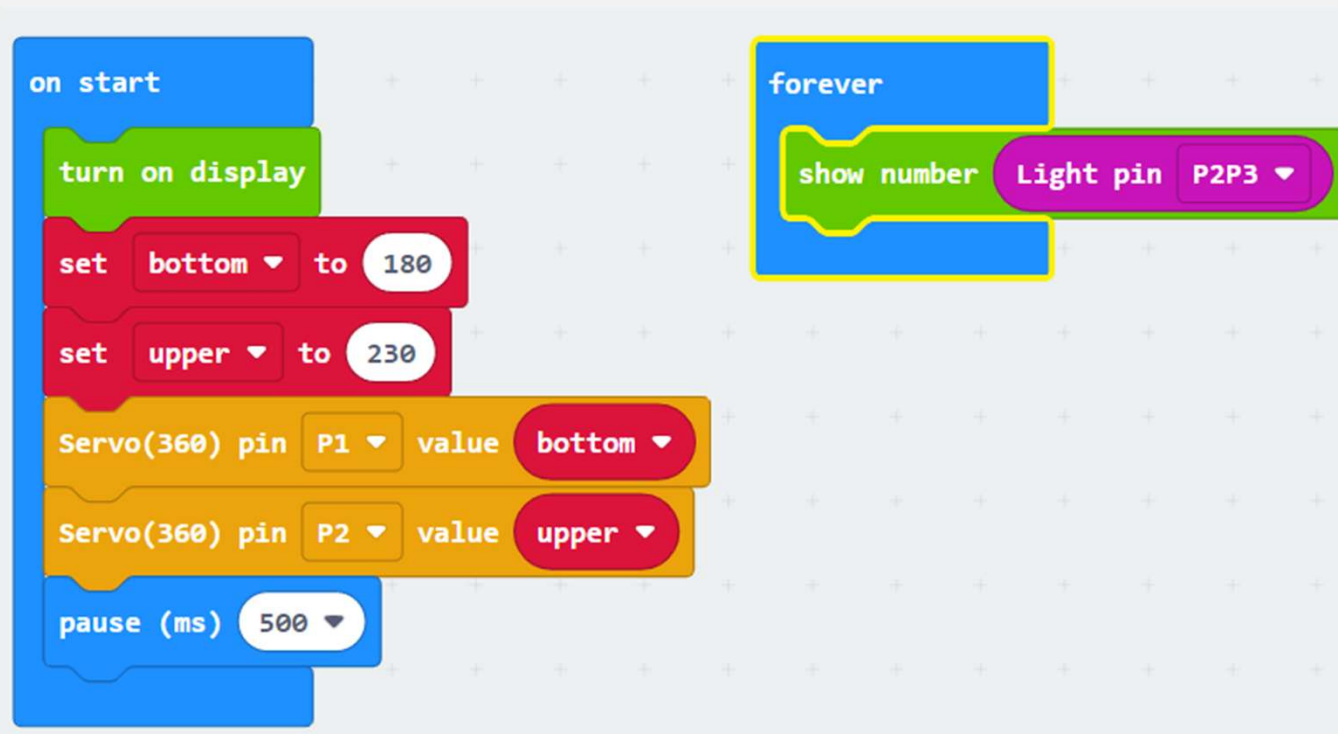
Set bottom servo position

```
forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if list get value at index > max light then
          set max light to list get value at index
        if max light = list get value at 0 then
          Servo(360) pin P1 value 230
        else if max light = list get value at 1 then
          Servo(360) pin P1 value 180
        else if max light = list get value at 2 then
          Servo(360) pin P1 value 130
```

```
if max light = list get value at 0 then
  Servo(360) pin P1 value 230
else if max light = list get value at 1 then
  Servo(360) pin P1 value 180
else if max light = list get value at 2 then
  Servo(360) pin P1 value 130
```

Change the value of 0, 1, 2 to get the value from list (0, 1, 2).

Coding Overview – Page 1

A screenshot of a Scratch code editor showing two scripts. The first script, labeled "on start", contains five blocks: "turn on display", "set bottom to 180", "set upper to 230", "Servo(360) pin P1 value bottom", and "pause (ms) 500". The second script, labeled "forever", contains one block: "show number Light pin P2P3".

```
on start
  turn on display
  set bottom to 180
  set upper to 230
  Servo(360) pin P1 value bottom
  Servo(360) pin P2 value upper
  pause (ms) 500

forever
  show number Light pin P2P3
```

Coding Overview – Page 2

```

function Bottom Servo Scan from Left to Middle
  set bottom to 230
  Servo(360) pin P1 value bottom
  pause (ms) 200
  while bottom > 180
  do
    change bottom by -1
    Servo(360) pin P1 value bottom
    pause (ms) 10

function Bottom Servo Scan from Middle to Right
  set bottom to 180
  Servo(360) pin P1 value bottom
  pause (ms) 200
  while bottom > 130
  do
    change bottom by -1
    Servo(360) pin P1 value bottom
    pause (ms) 10
  
```

Coding Overview – Page 3

```

function Scan and record photosensitive value
  set list to array of 0 1 2
  Servo(360) pin P1 value 230
  pause (ms) 200
  list set value at 0 to Light pin P2P3
  call Bottom Servo Scan from Left to Middle
  pause (ms) 200
  list set value at 1 to Light pin P2P3
  call Bottom Servo Scan from Middle to Right
  pause (ms) 200
  list set value at 2 to Light pin P2P3
  
```

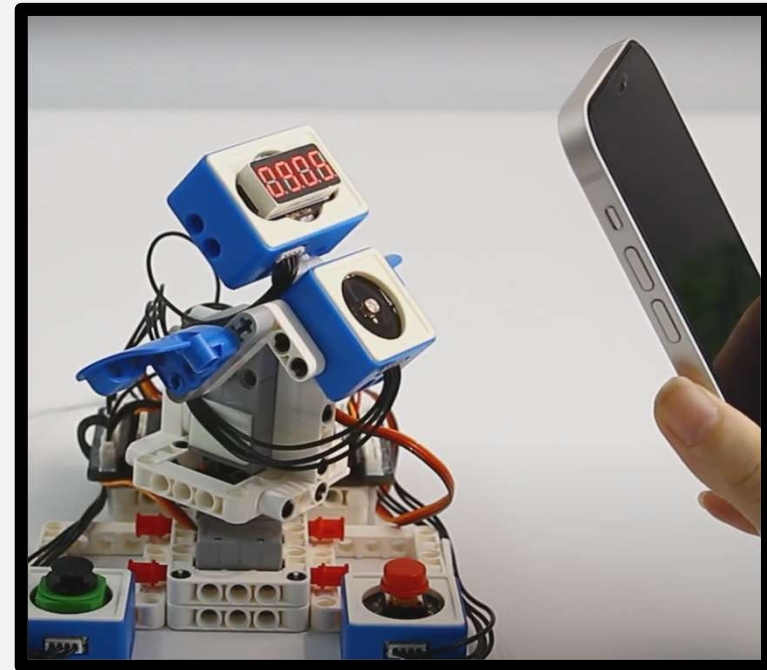
Coding Overview – Page 4

```

forever
  if Button pin P0P1 value Press then
    call Scan and record photosensitive value
    set max light to list get value at 0
    for index from 0 to 2
      do
        if list get value at index > max light then
          set max light to list get value at index
        +
      end
    end
    if max light = list get value at 0 then
      Servo(360) pin P1 value 230
    else if max light = list get value at 1 then -
      Servo(360) pin P1 value 180
    else if max light = list get value at 2 then -
      Servo(360) pin P1 value 130
    +
  end
  +
end
  
```

Phenomenon

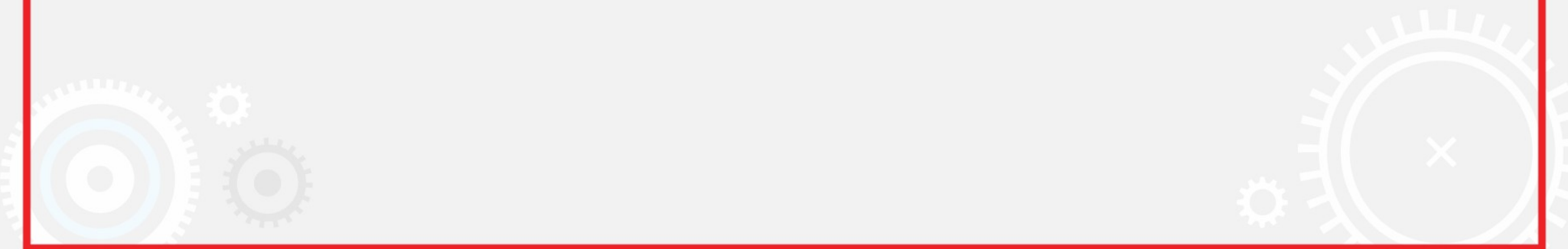
After the program is downloaded successfully.
When we **press the button**, The sunflower will
move **left to right**, and automatically **track** the
brightest light.



30 Points

CHALLENGE

for : Lesson 4



L4 – Challenge 1 - Tips

```

forever
  if Rocker pin P3P4 value Left then
    change bottom by 1
    Servo(360) pin P1 value bottom
    pause (ms) 10
  else if Rocker pin P3P4 value Right then
    change bottom by -1
    Servo(360) pin P1 value bottom
    pause (ms) 10
  
```

To make your rocker controller able to manipulate the position of the bottom servo, we can set the code like on the left.

But for safety purpose, we don't want to overturn our servo, so we need to set min and max position for our bottom servo.
(see next slide)

L4 – Challenge 1 - Tips

```

on start
  turn on display
  set bottom min to 130
  set bottom max to 230
  set bottom to 180
  set upper to 230
  Servo(360) pin P1 value bottom
  Servo(360) pin P2 value upper
  pause (ms) 500
  
```

Make new variables called "bottom min" and "bottom max" accordingly.

Then set bottom min to 130 and bottom max to 230 on start

L4 – Challenge 1 - Tips

```

forever
  if Rocker pin P3P4 value Left then
    if bottom < bottom max then
      change bottom by 1
      Servo(360) pin P1 value bottom
      pause (ms) 10
    else if Rocker pin P3P4 value Right then
      if bottom > bottom min then
        change bottom by -1
        Servo(360) pin P1 value bottom
        pause (ms) 10
  
```

To make sure it won't turn out of the range, we will use the if statement to check if the position is out of range or not.

If it's out of range, it won't turn further, it can only go to the opposite positions.

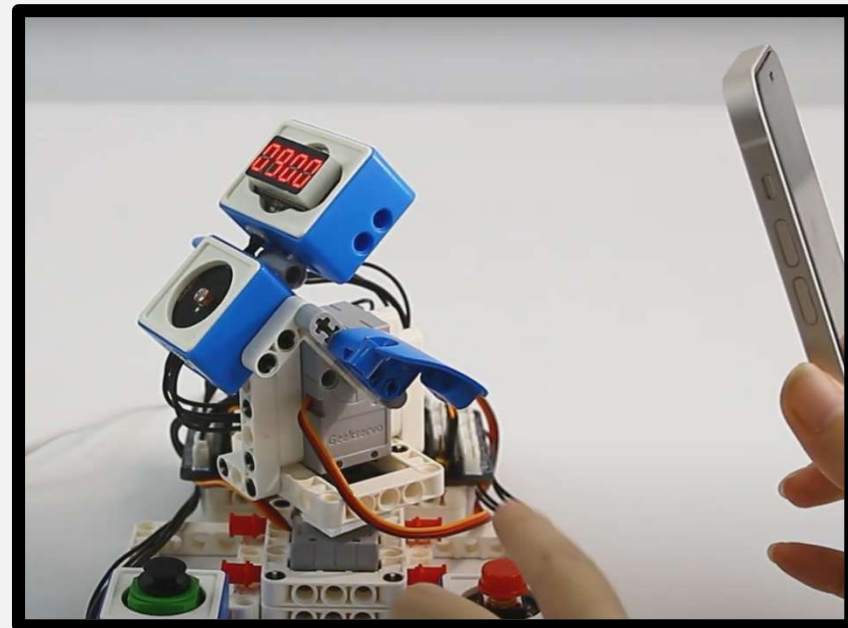
If you want to set upper servo limit, you should set from min 200 to max 230.

L4 – Challenge 1

After the program is downloaded successfully.

Use the joystick to control the photosensitive module up, down, left, and right.

30 Points



*example only

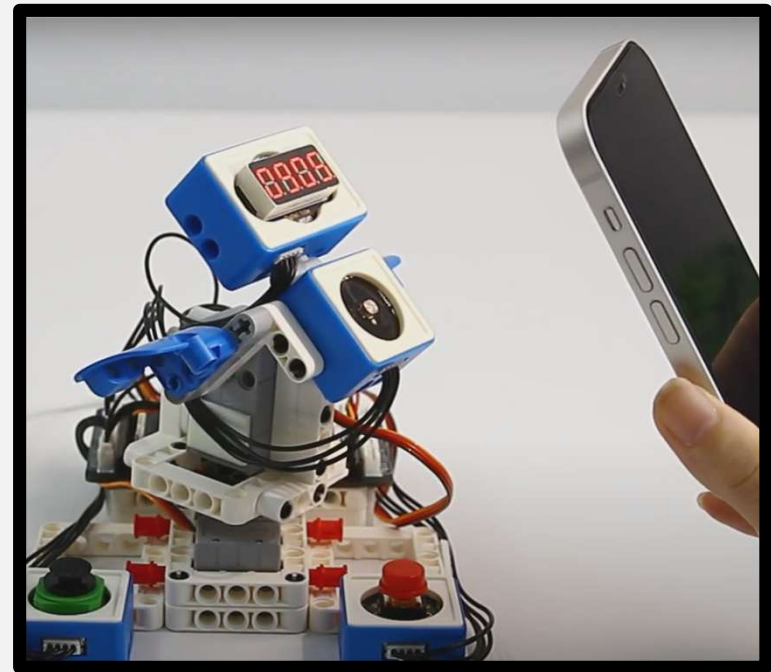
L4 – Challenge 2

Explain the mechanism of this sunflower robot.

Briefly explain how your coding works in this sunflower robot.

What is the application of this sunflower robot in real-life?

30 Points



L4 – Challenge 3

Make the sunflower scan light up and down. Then it will track which is the **brightest**.

30 Points

