

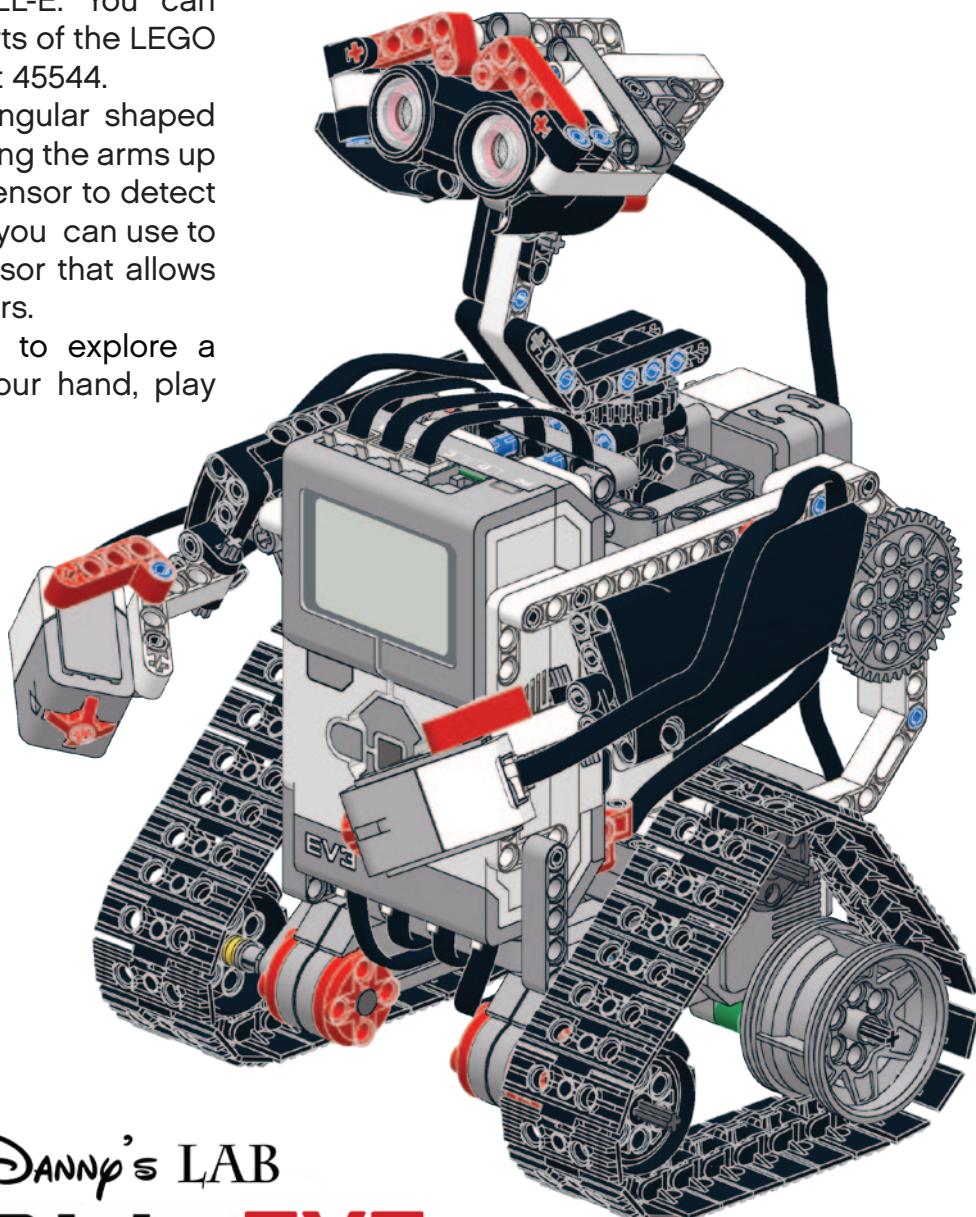


# WALL•EV3

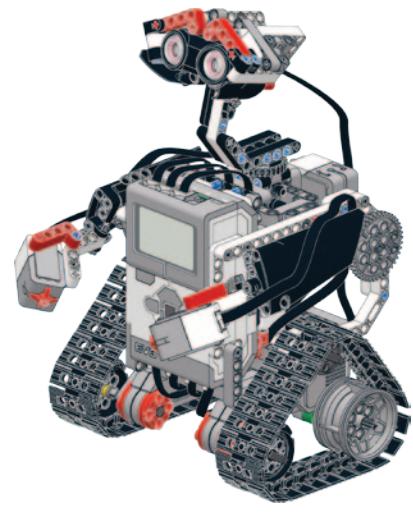
WALL•EV3 is my LEGO MINDSTORMS EV3 rendition of Disney-Pixar's WALL-E. You can build this robot using just the parts of the LEGO MINDSTORMS Education EV3 set 45544.

WALL•EV3 moves on two triangular shaped tracks, can turn the head and swing the arms up and down. It has an Ultrasonic Sensor to detect objects, two Touch Sensors that you can use to interact with him, and Gyro Sensor that allows him to perform precise maneuvers.

You can program WALL•EV3 to explore a room, avoid obstacles, follow your hand, play games, and much more!

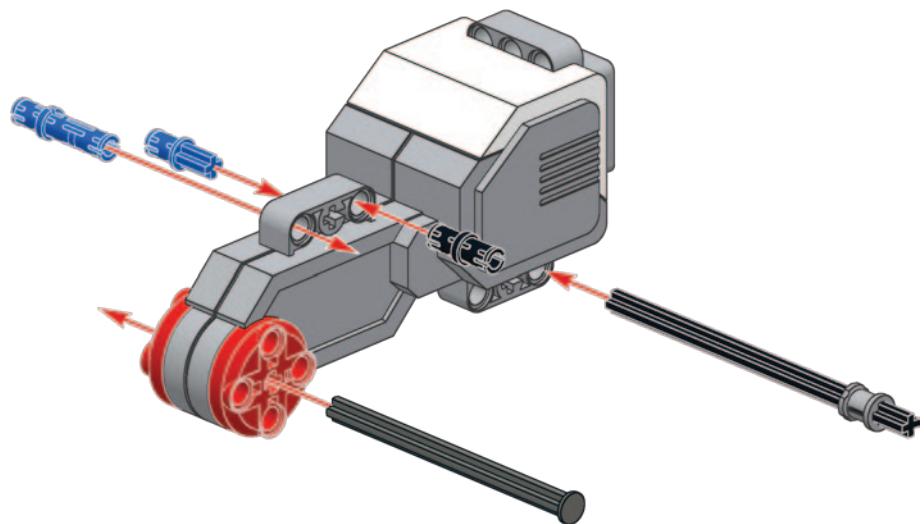
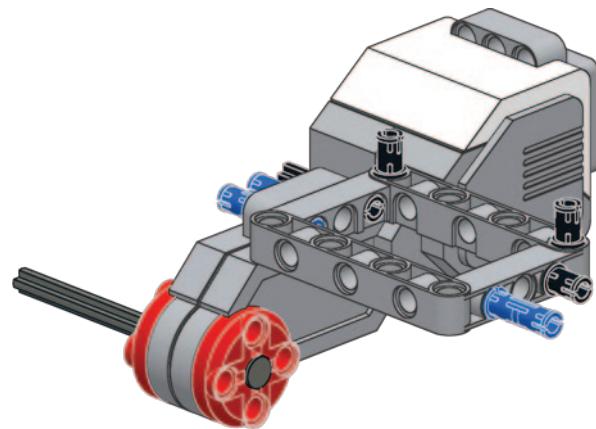
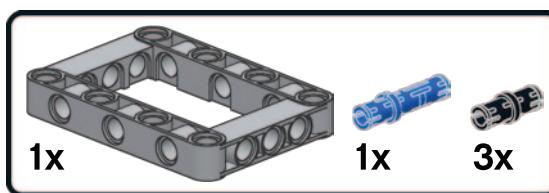


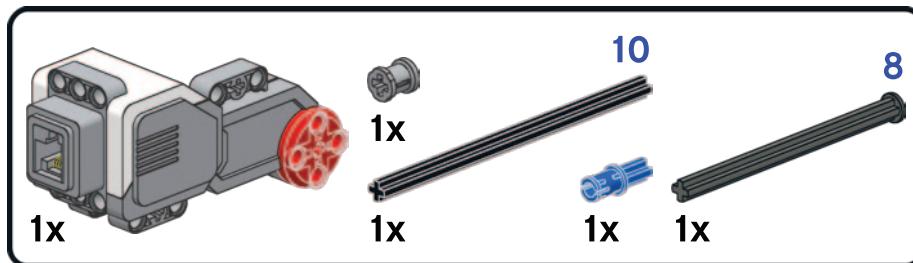
DANNY'S LAB  
**WALL•EV3**



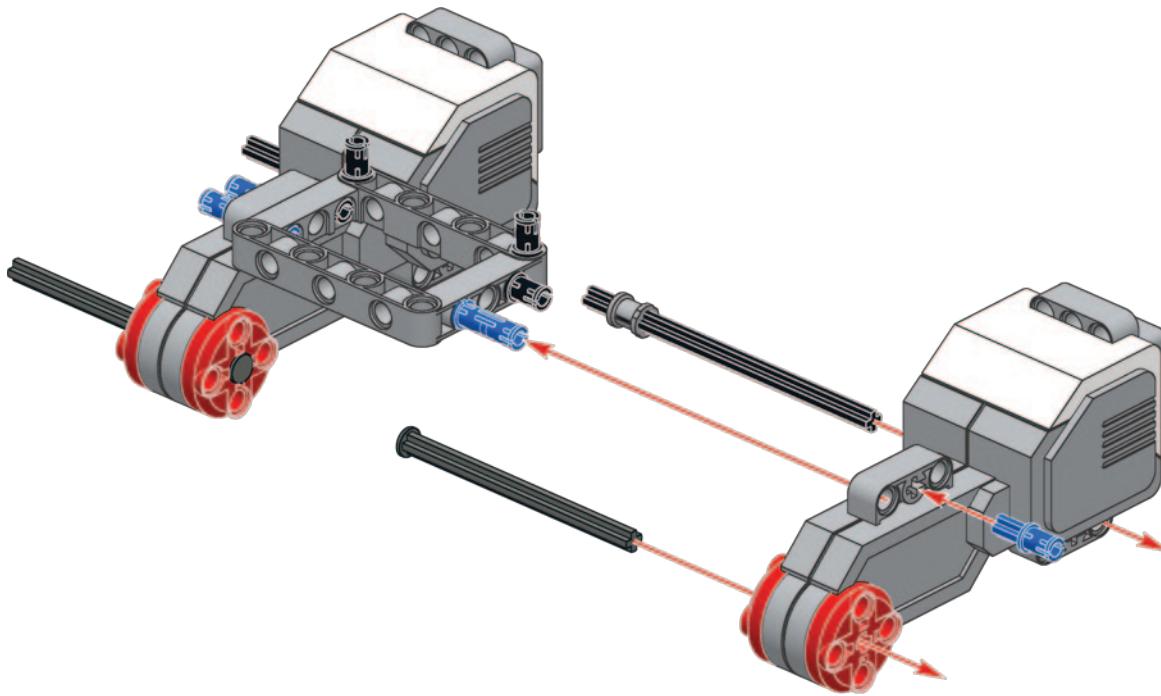
# Building WALL•EV3

## Main assembly

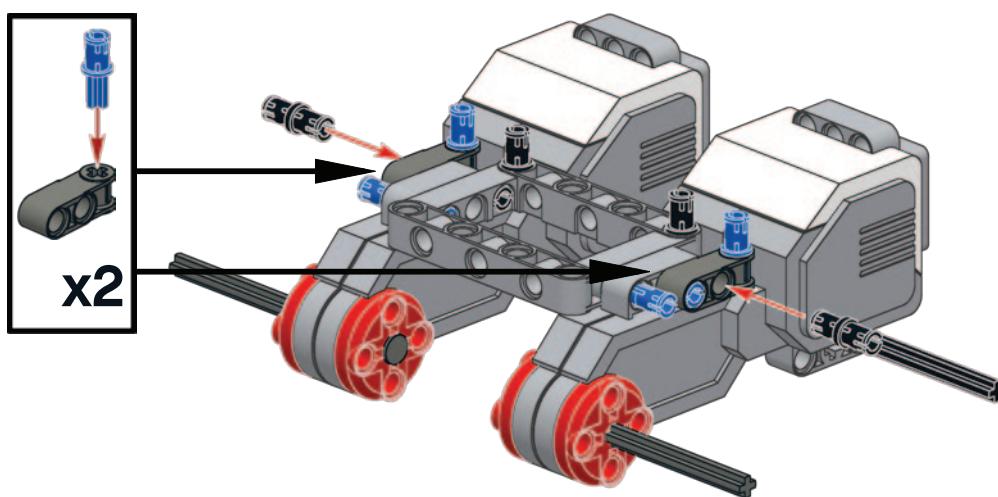
**1****2**

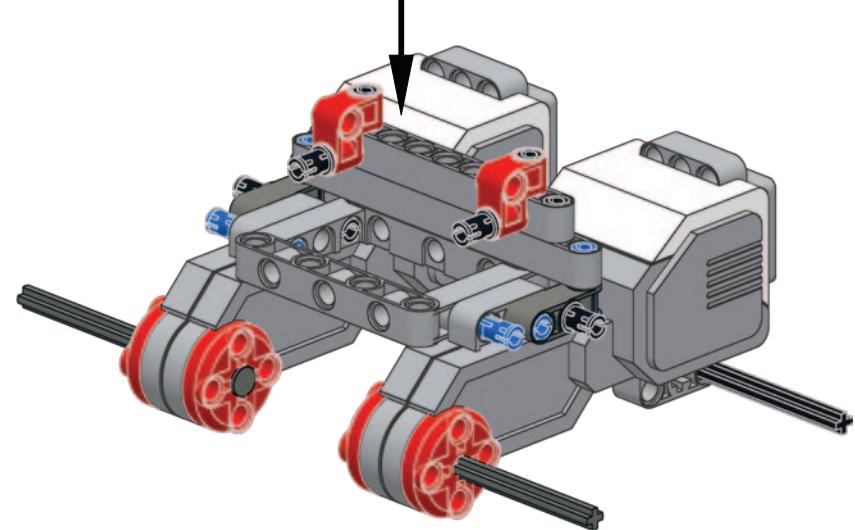
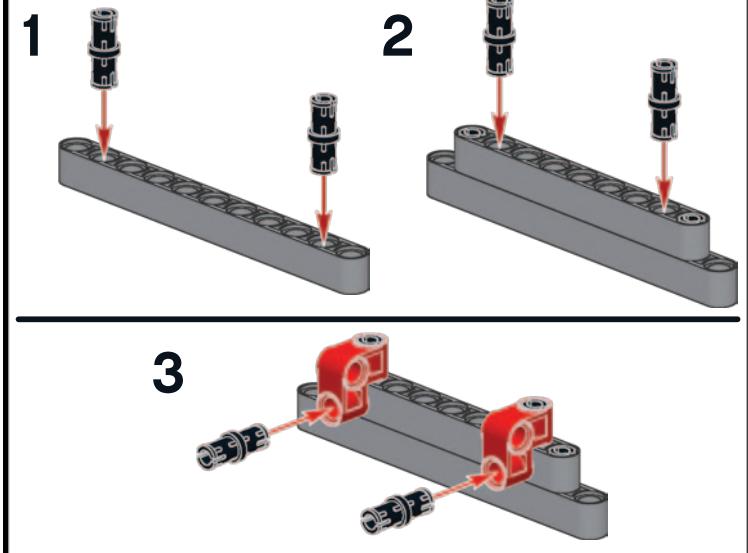
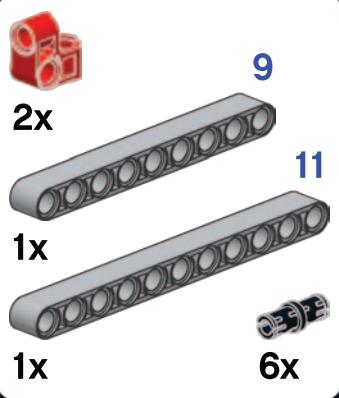
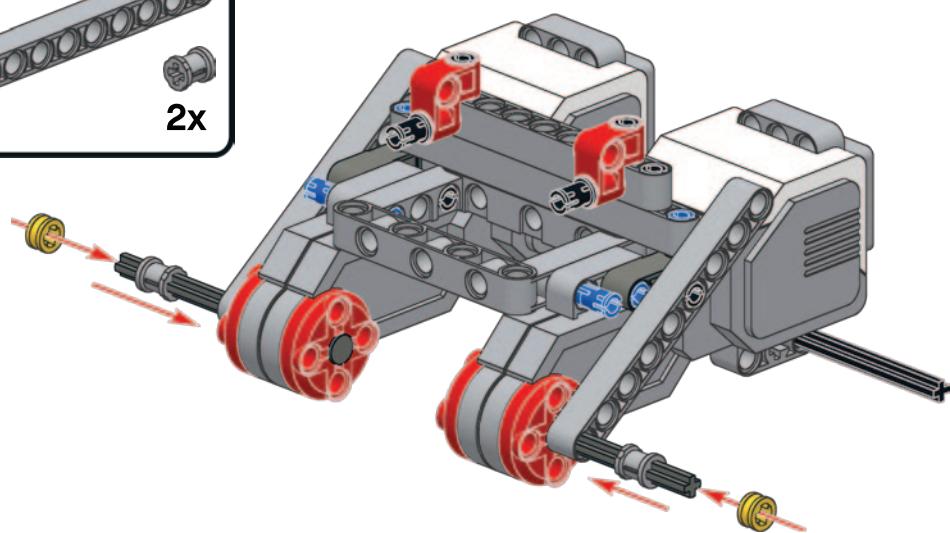
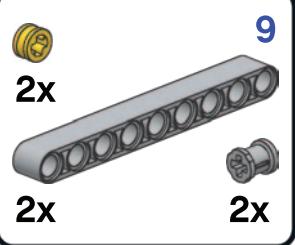


**3**



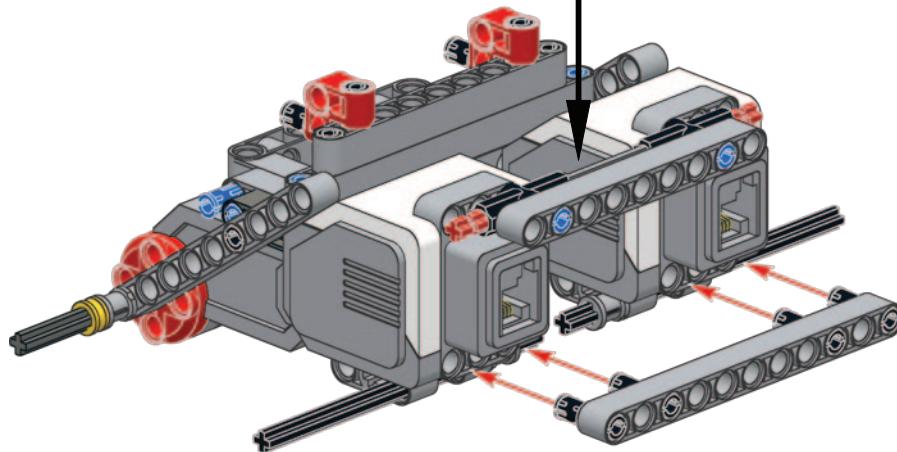
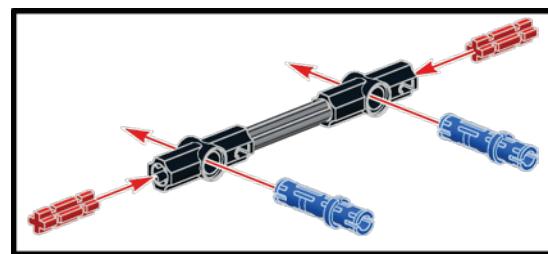
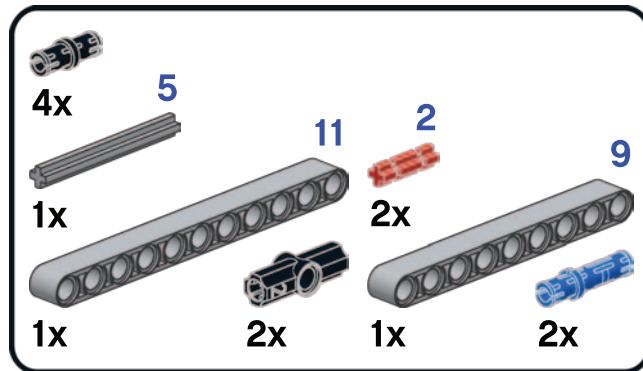
**4**



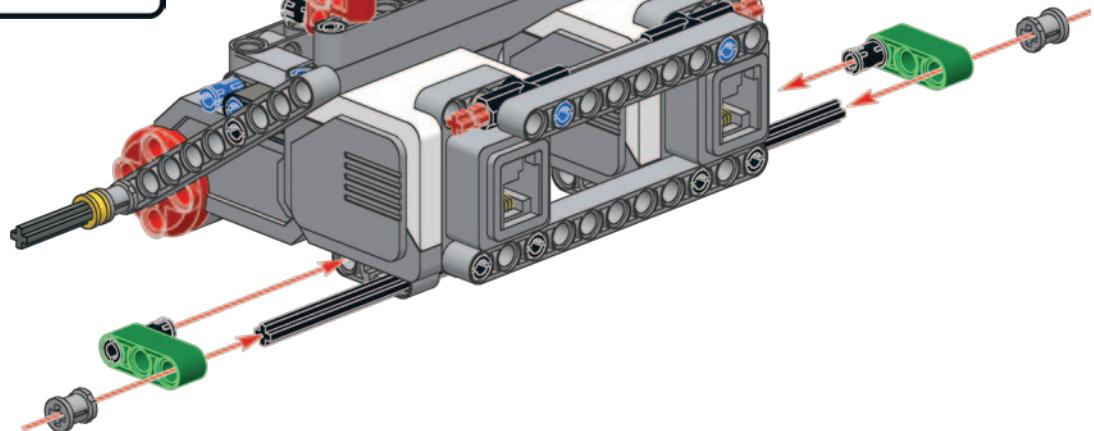
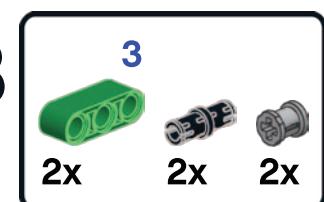
**5****6**

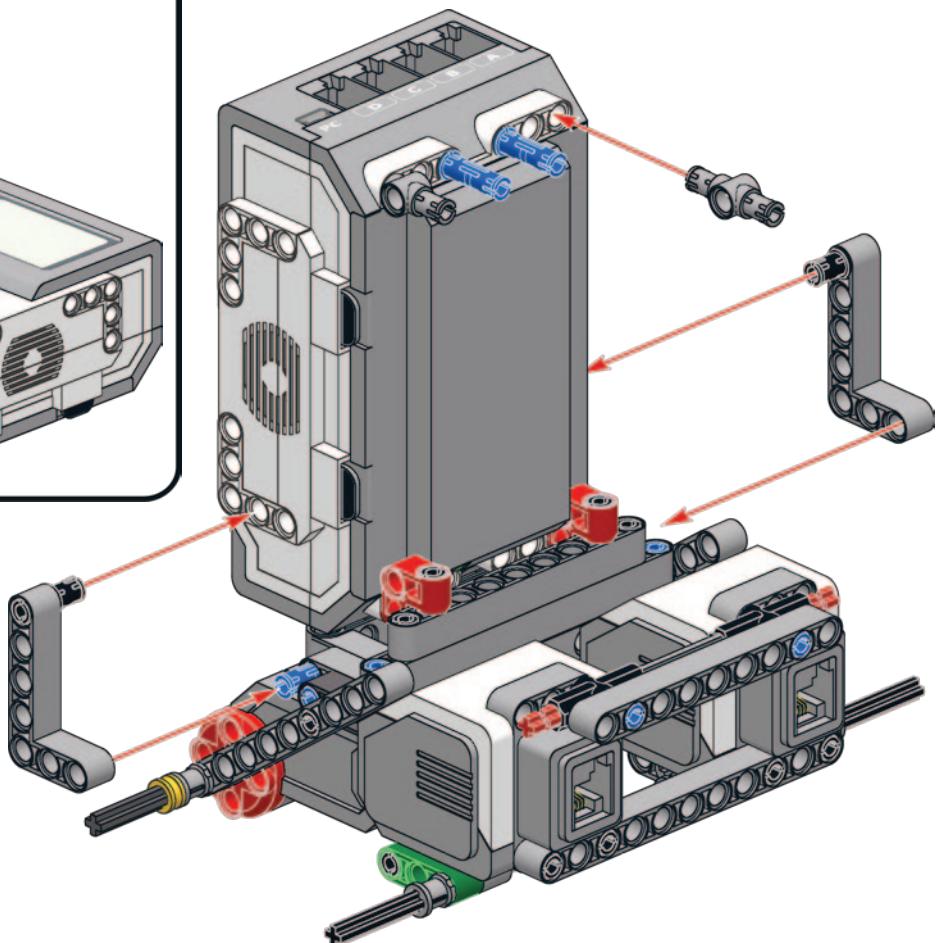
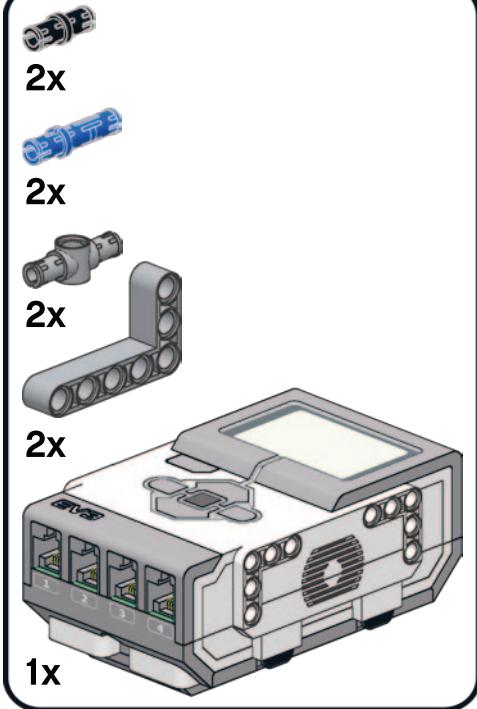
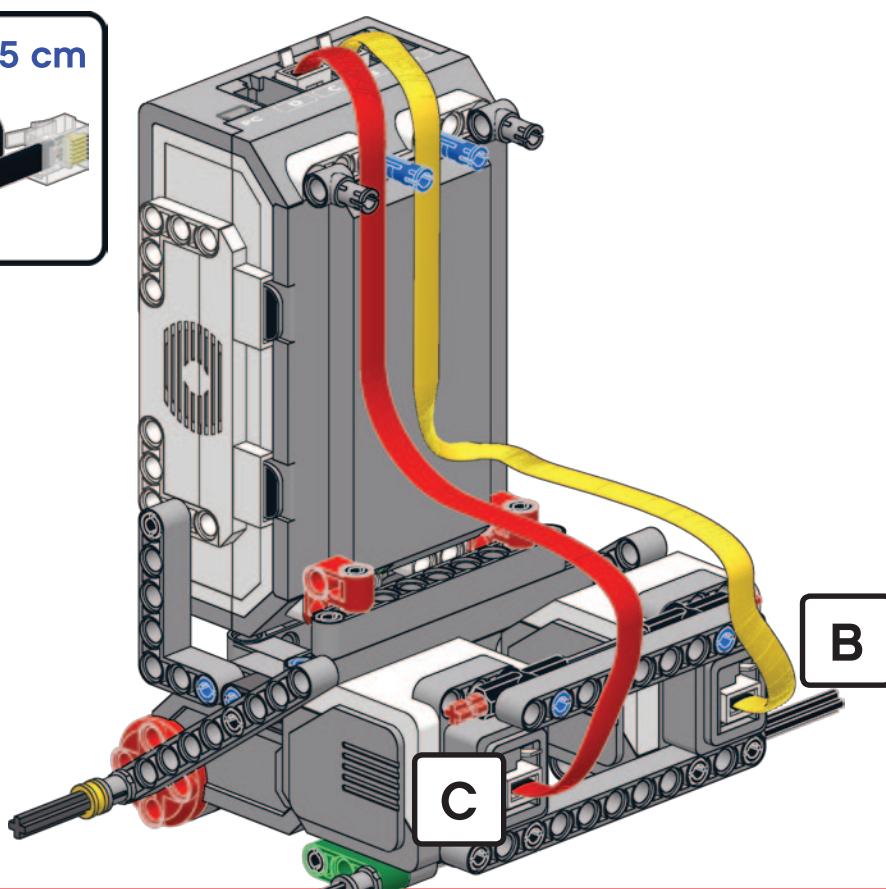
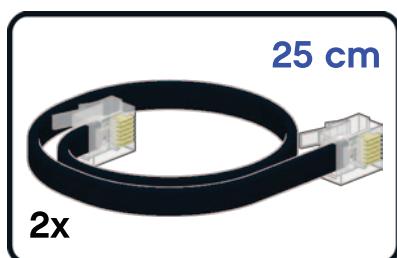


7



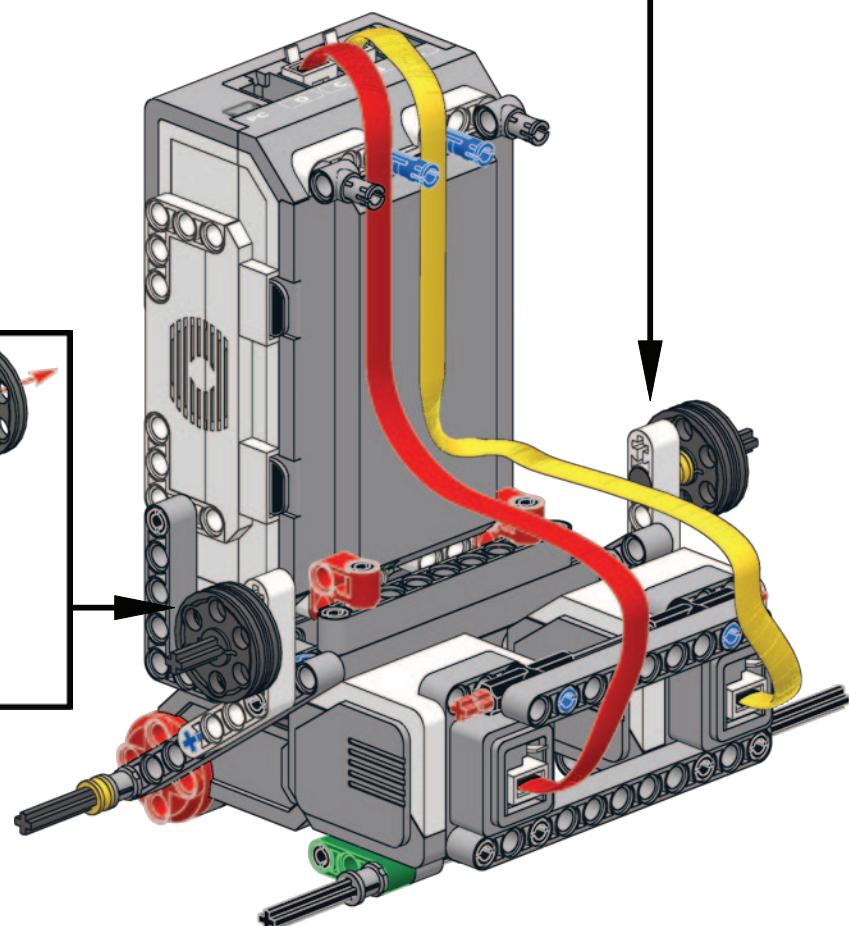
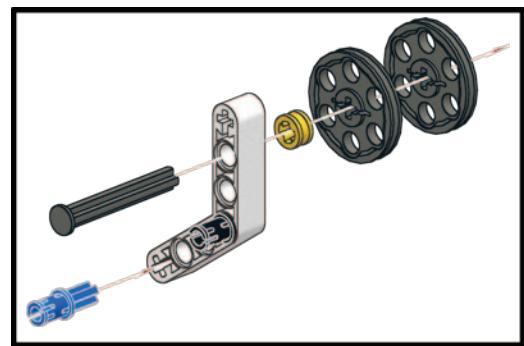
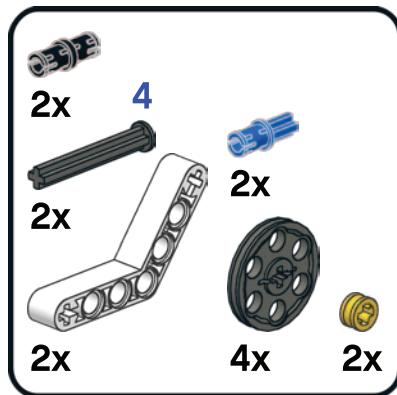
8



**9****10**

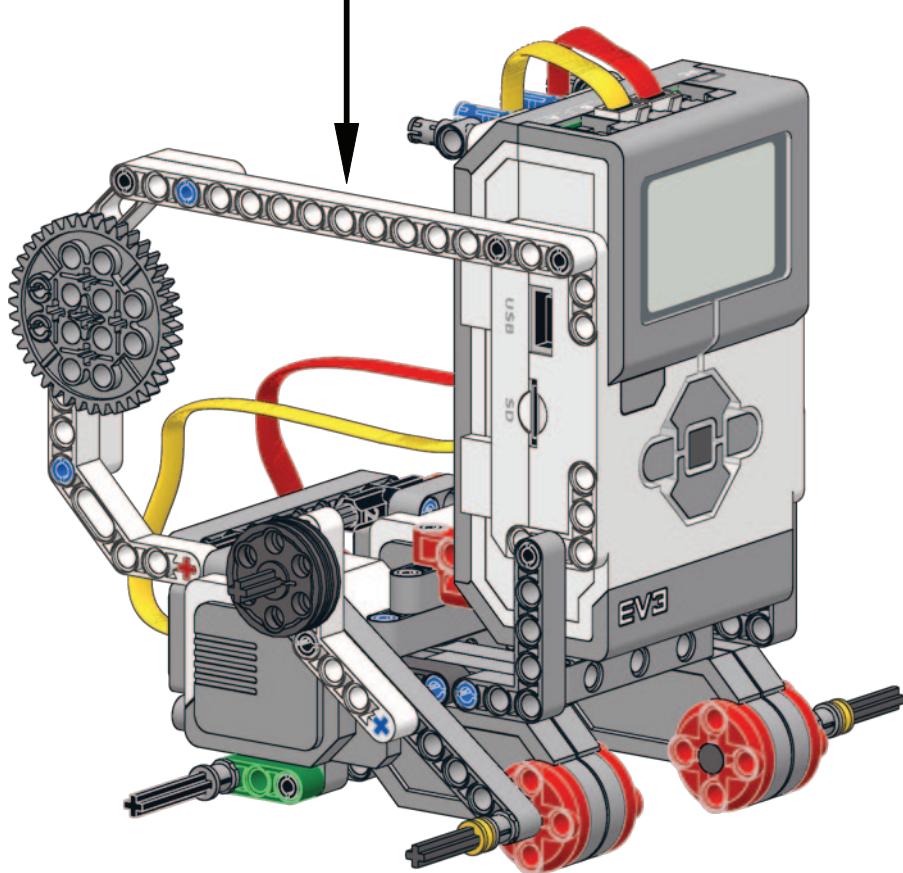
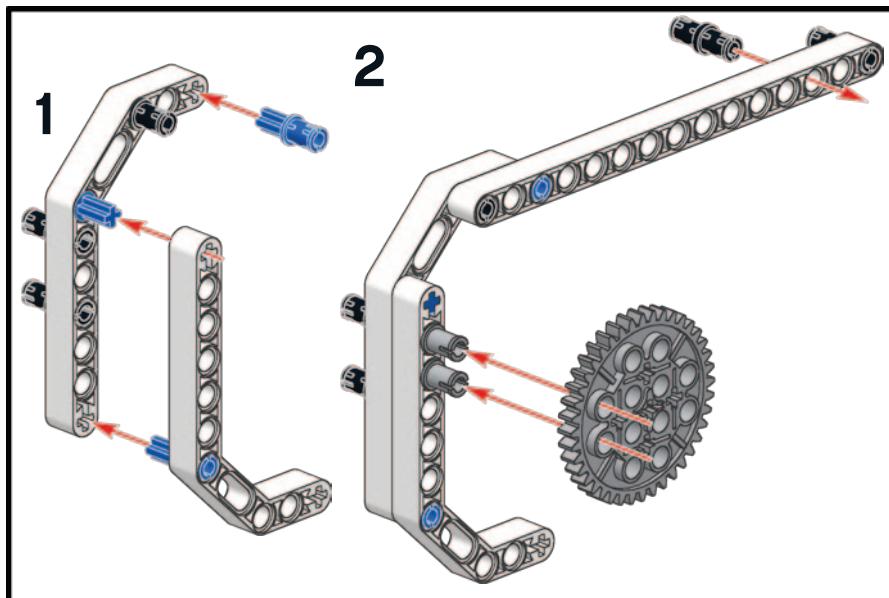
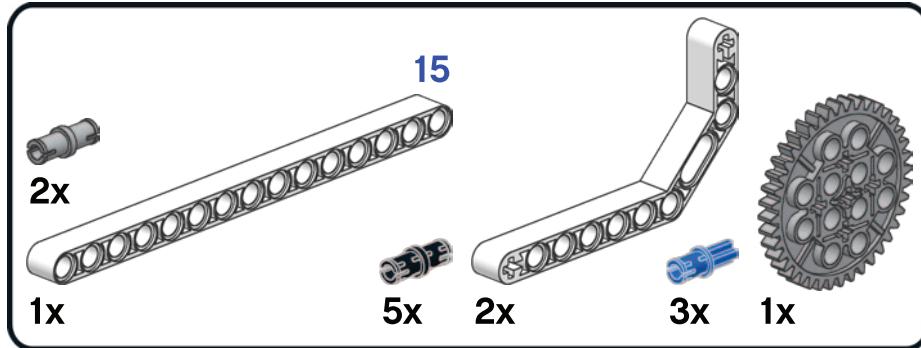


11



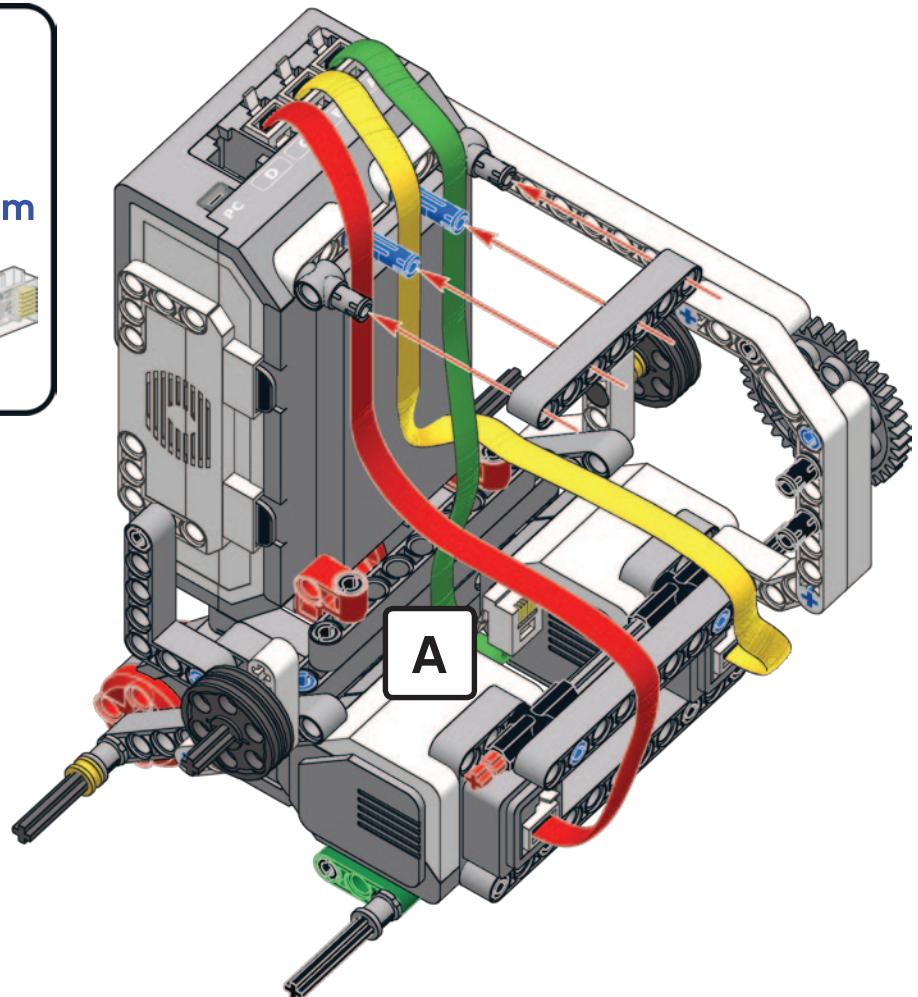


12

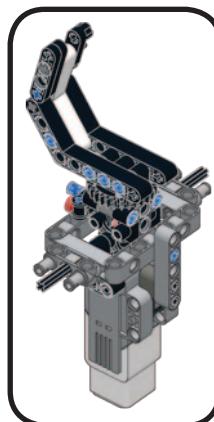




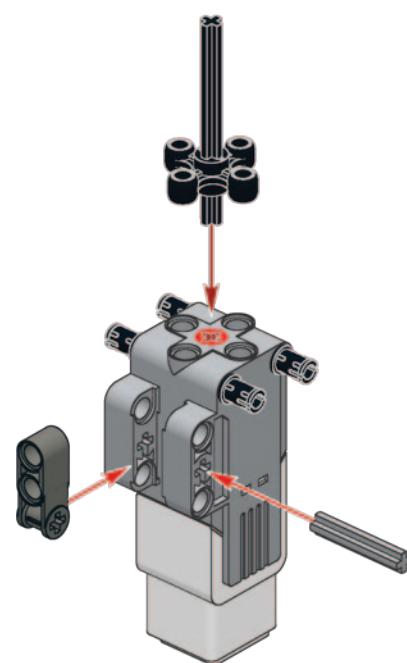
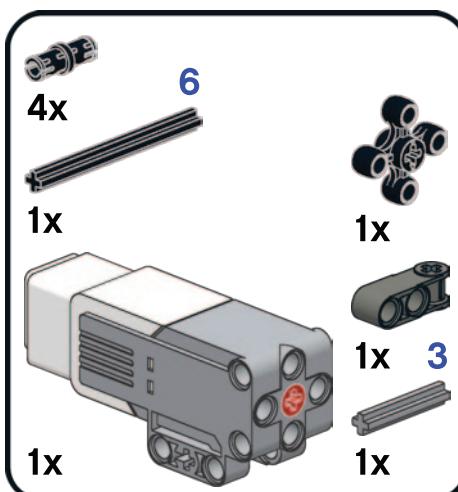
13

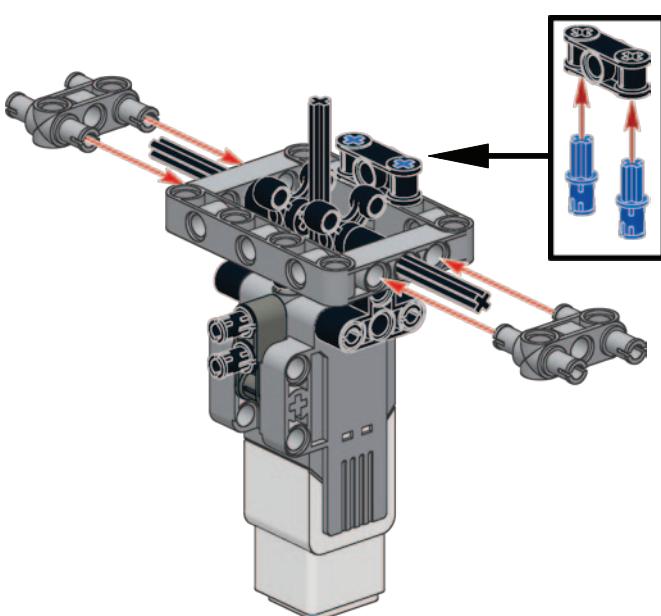
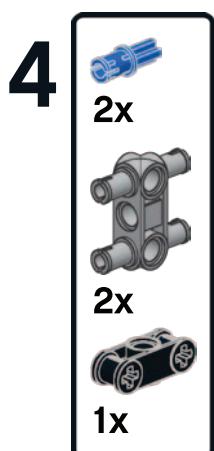
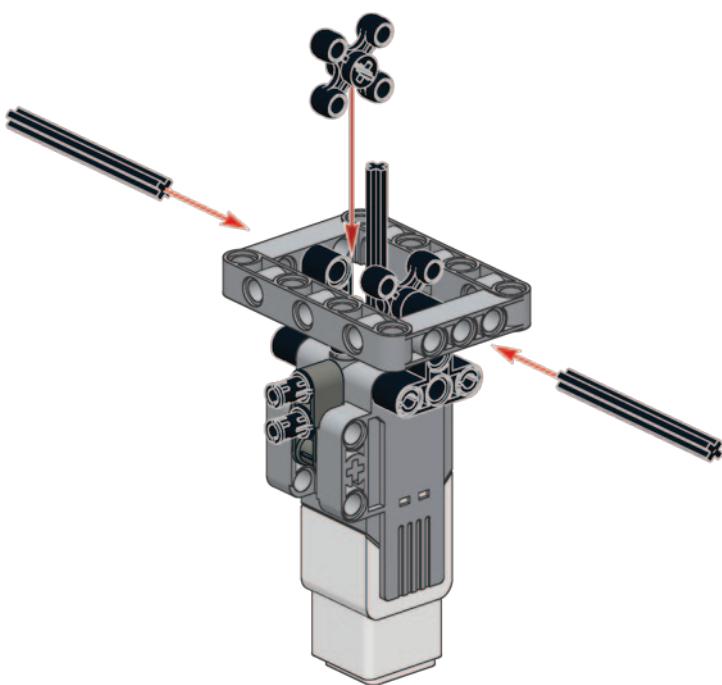
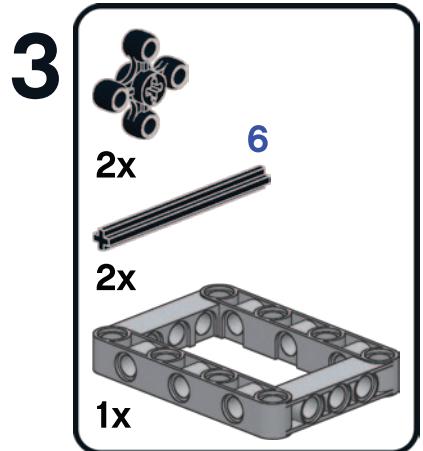
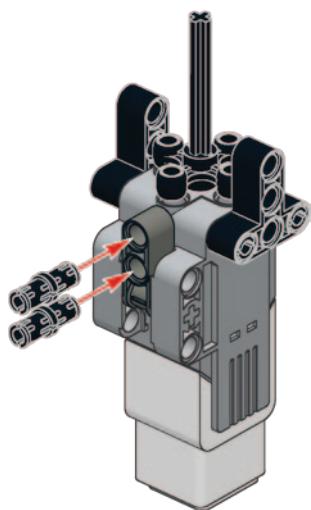
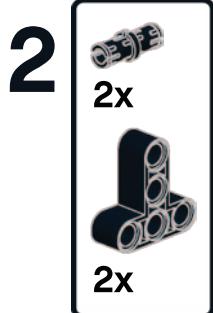


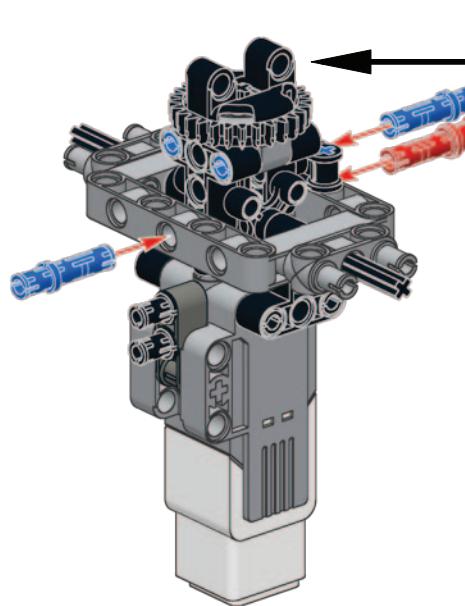
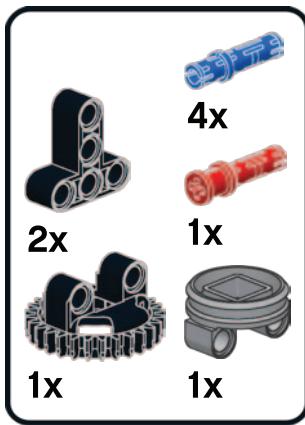
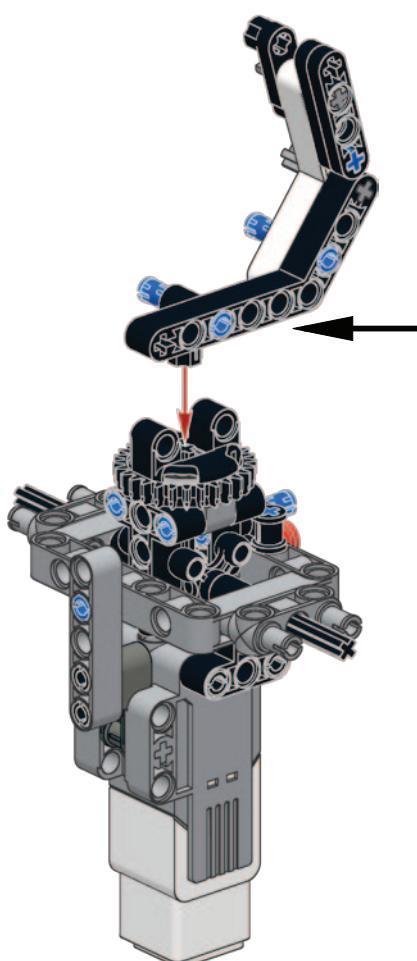
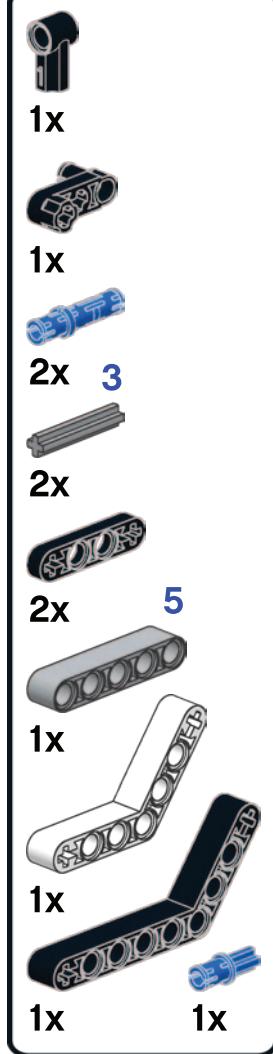
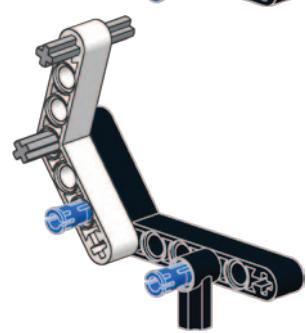
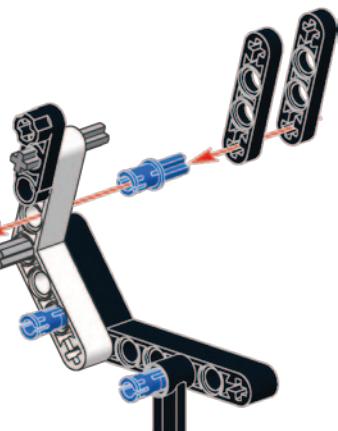
## Motor subassembly



1

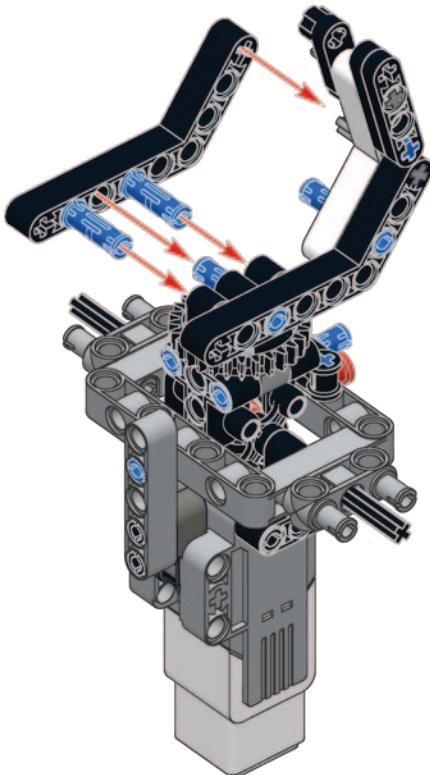
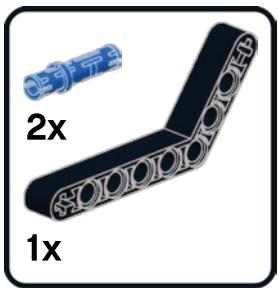




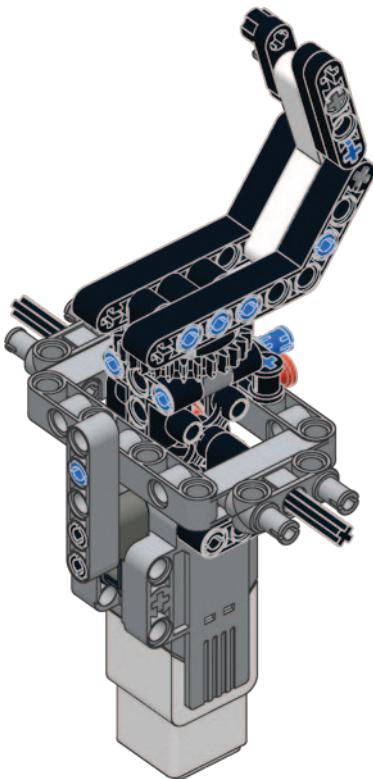
**5****1****2****6****1****2****3**



7

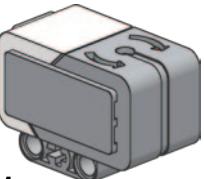


8

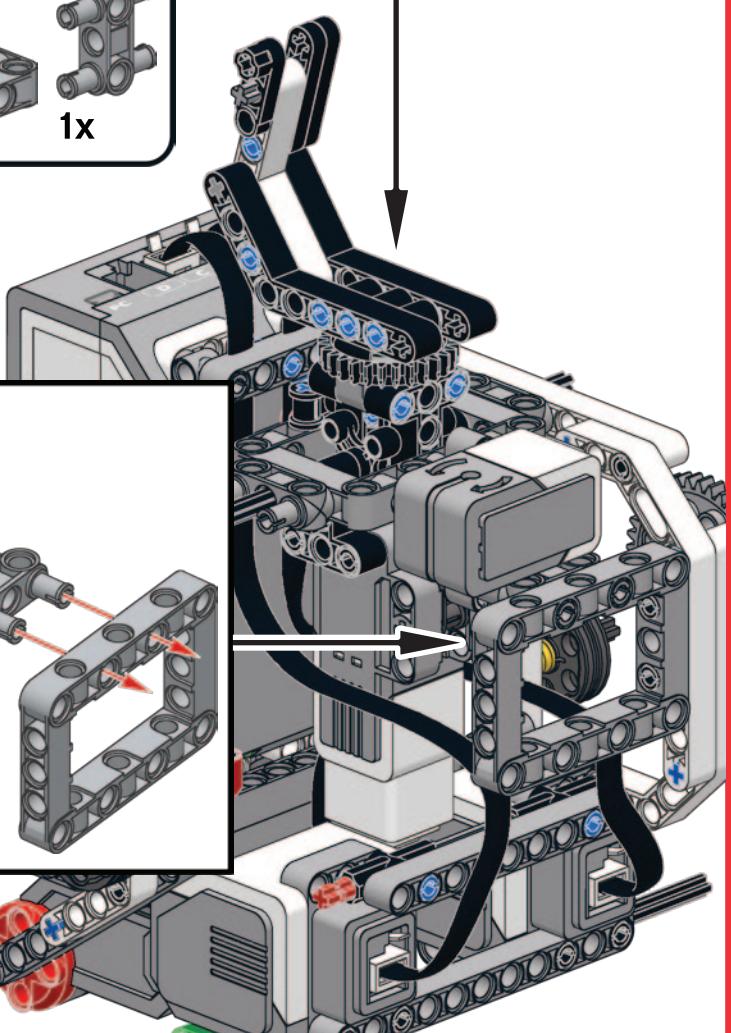
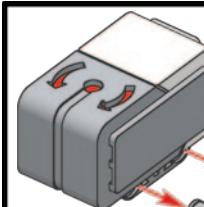


1x  
1x  
1x

1x

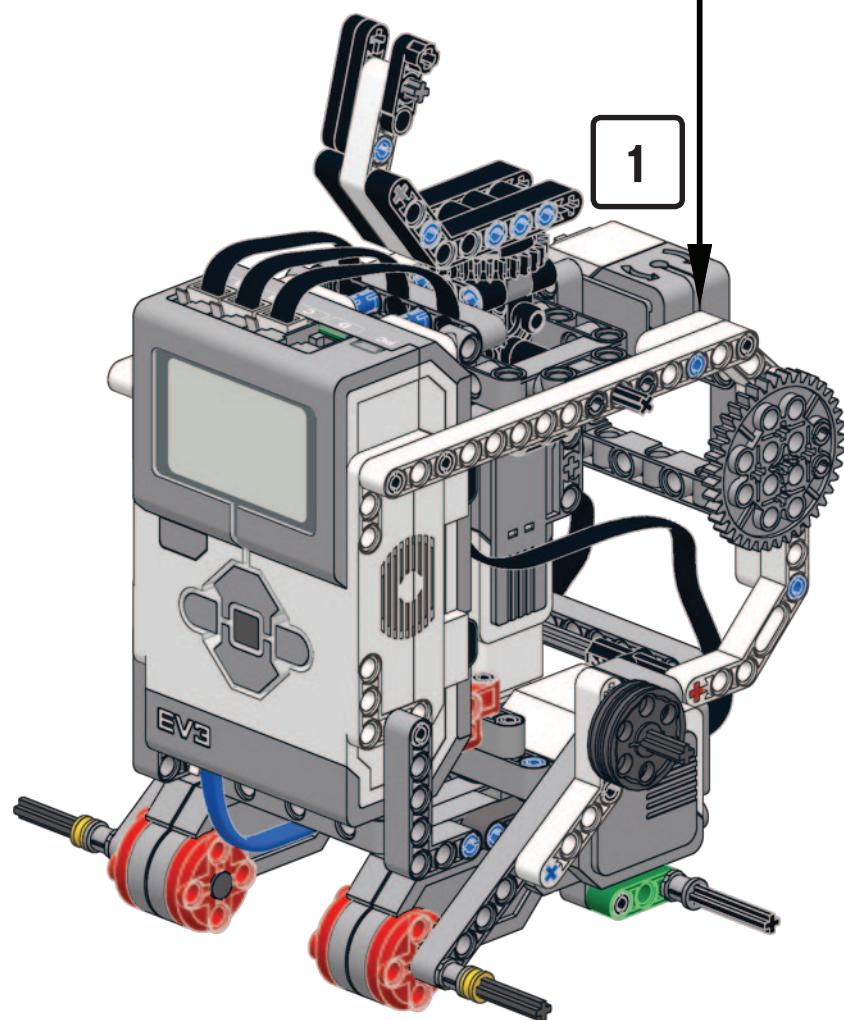
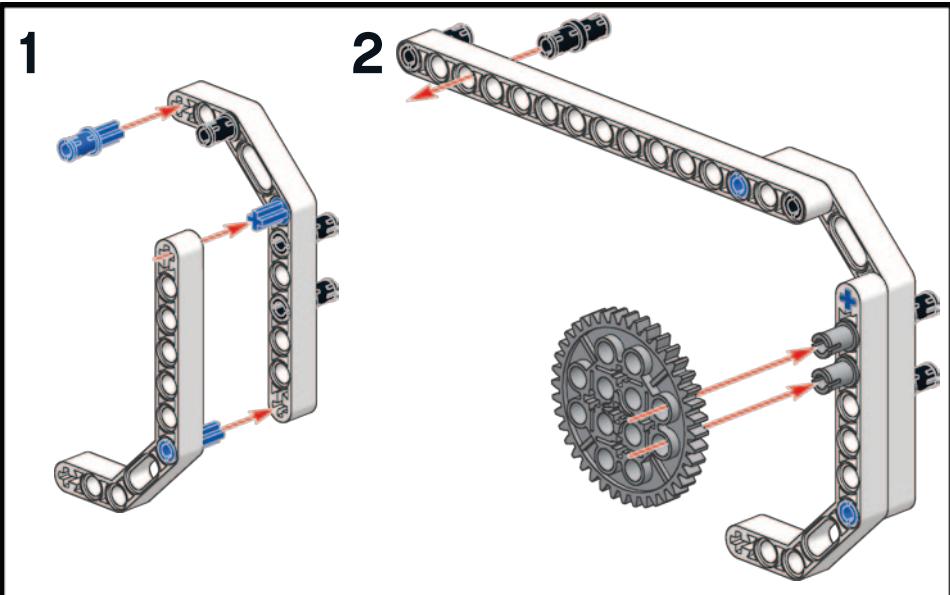
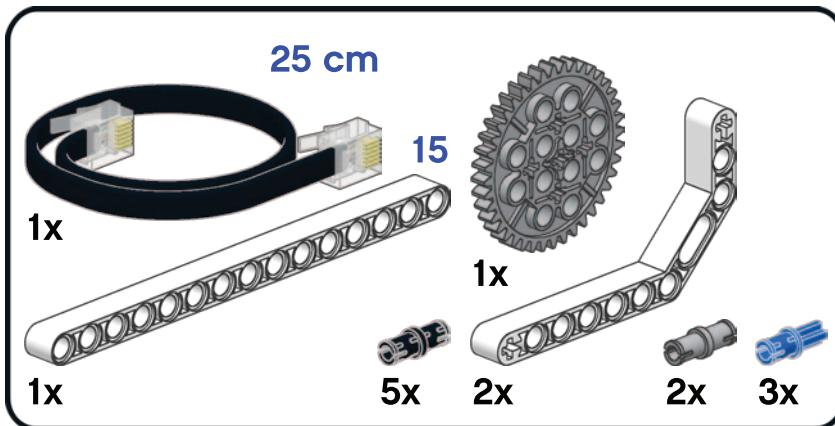


14



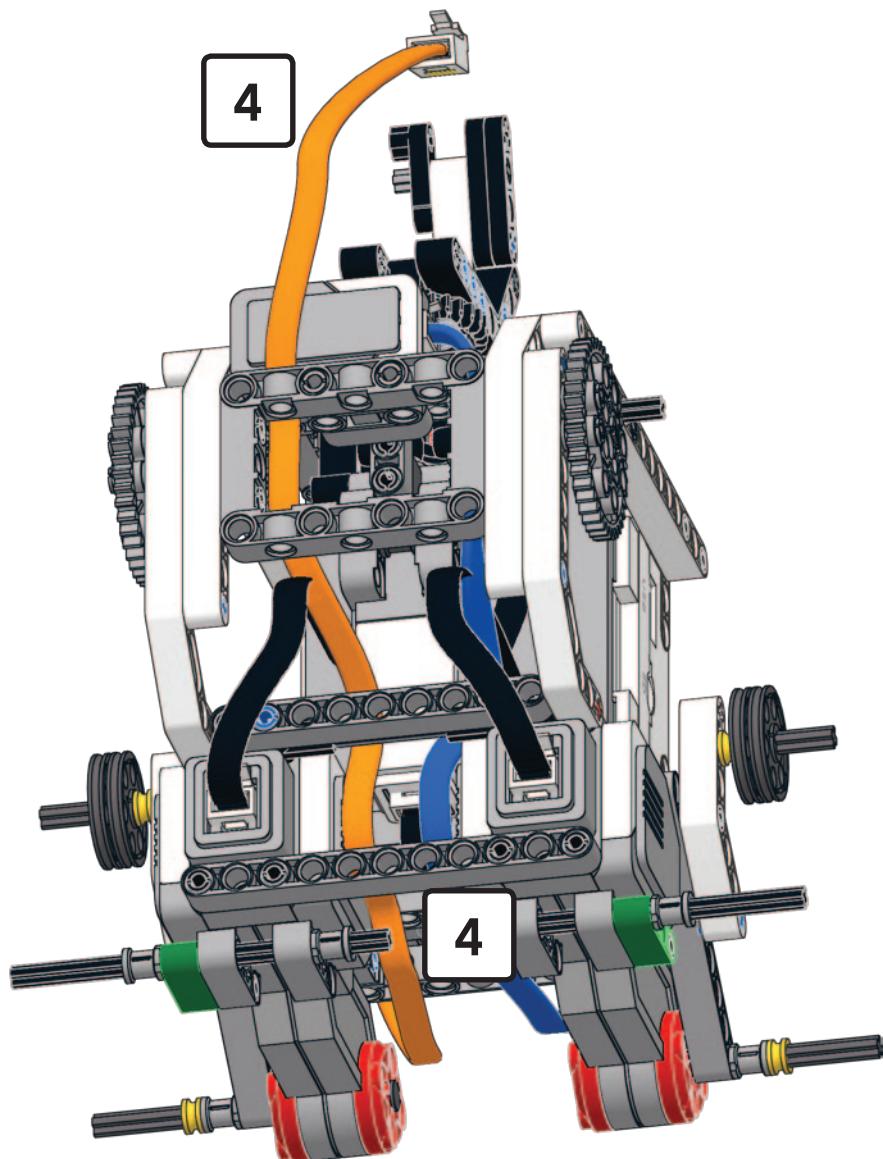


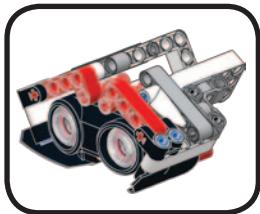
15



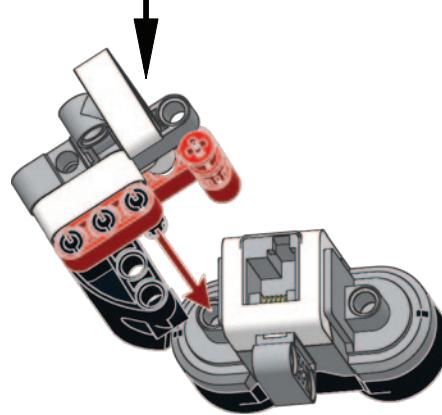
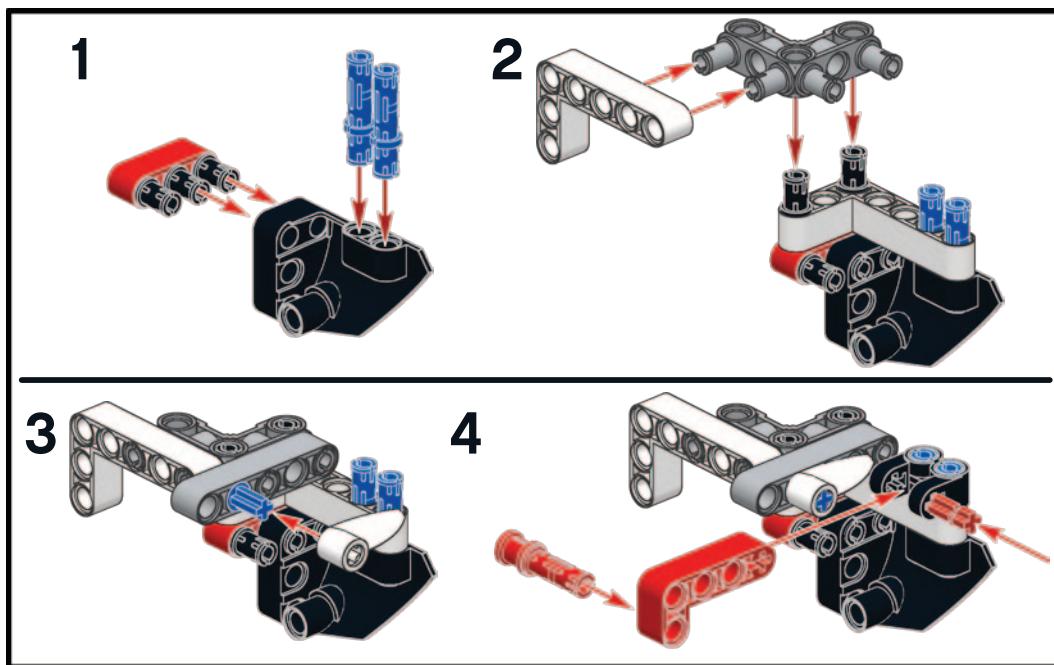
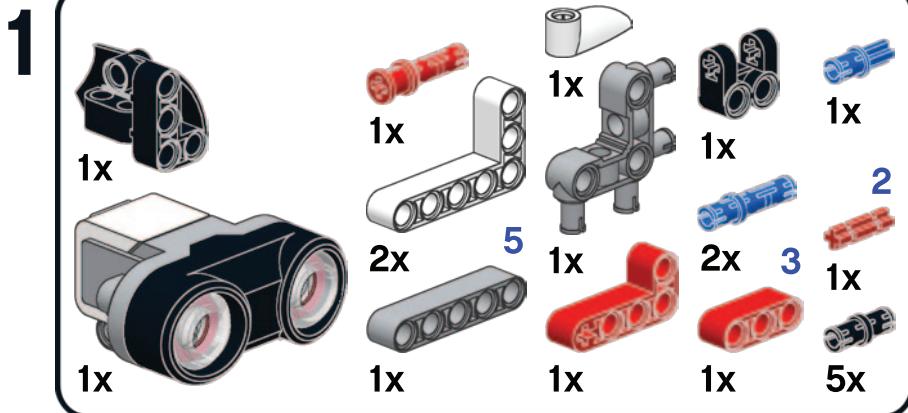


**16**



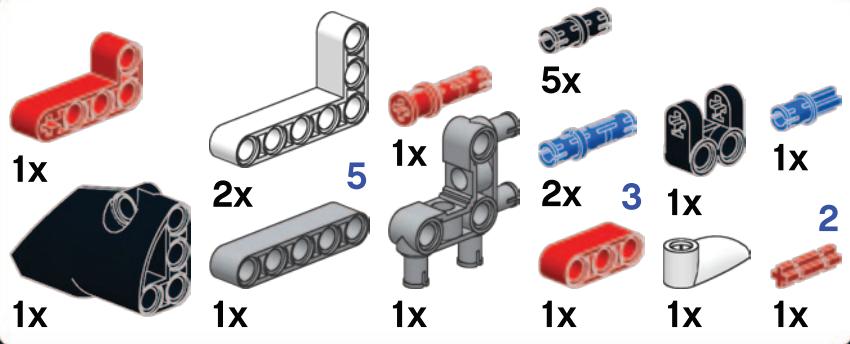


# Head subassembly

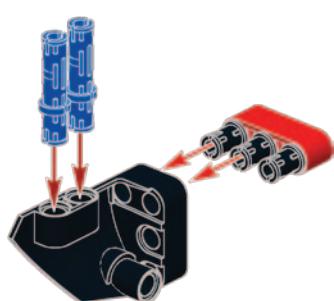




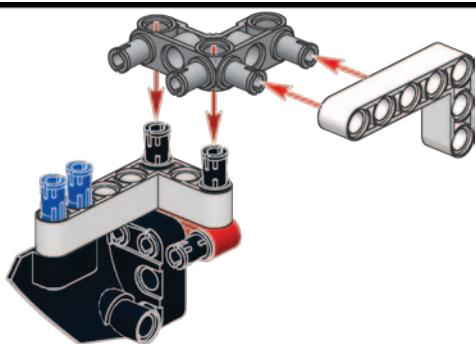
2



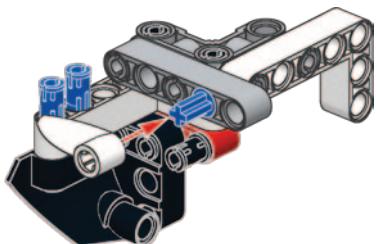
1



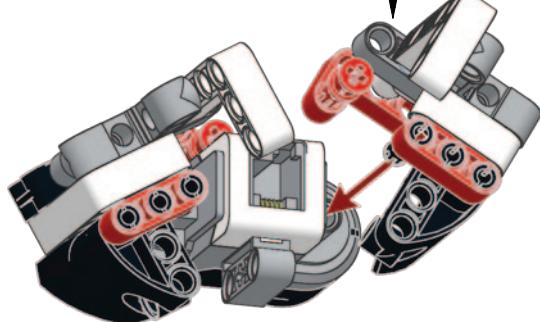
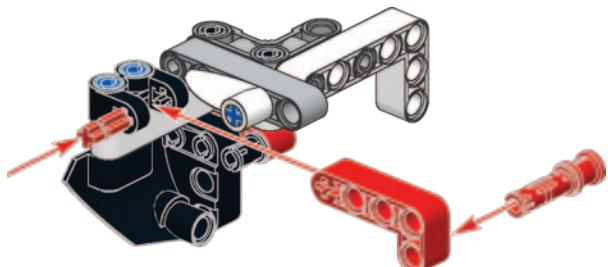
2



3

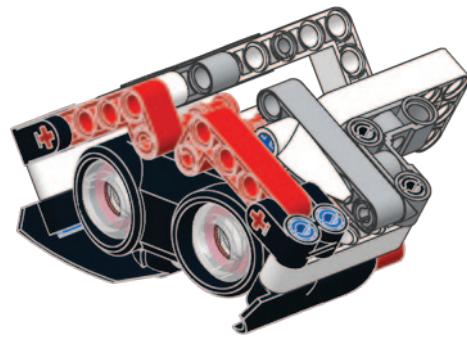


4

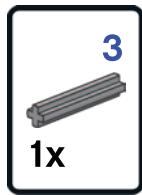




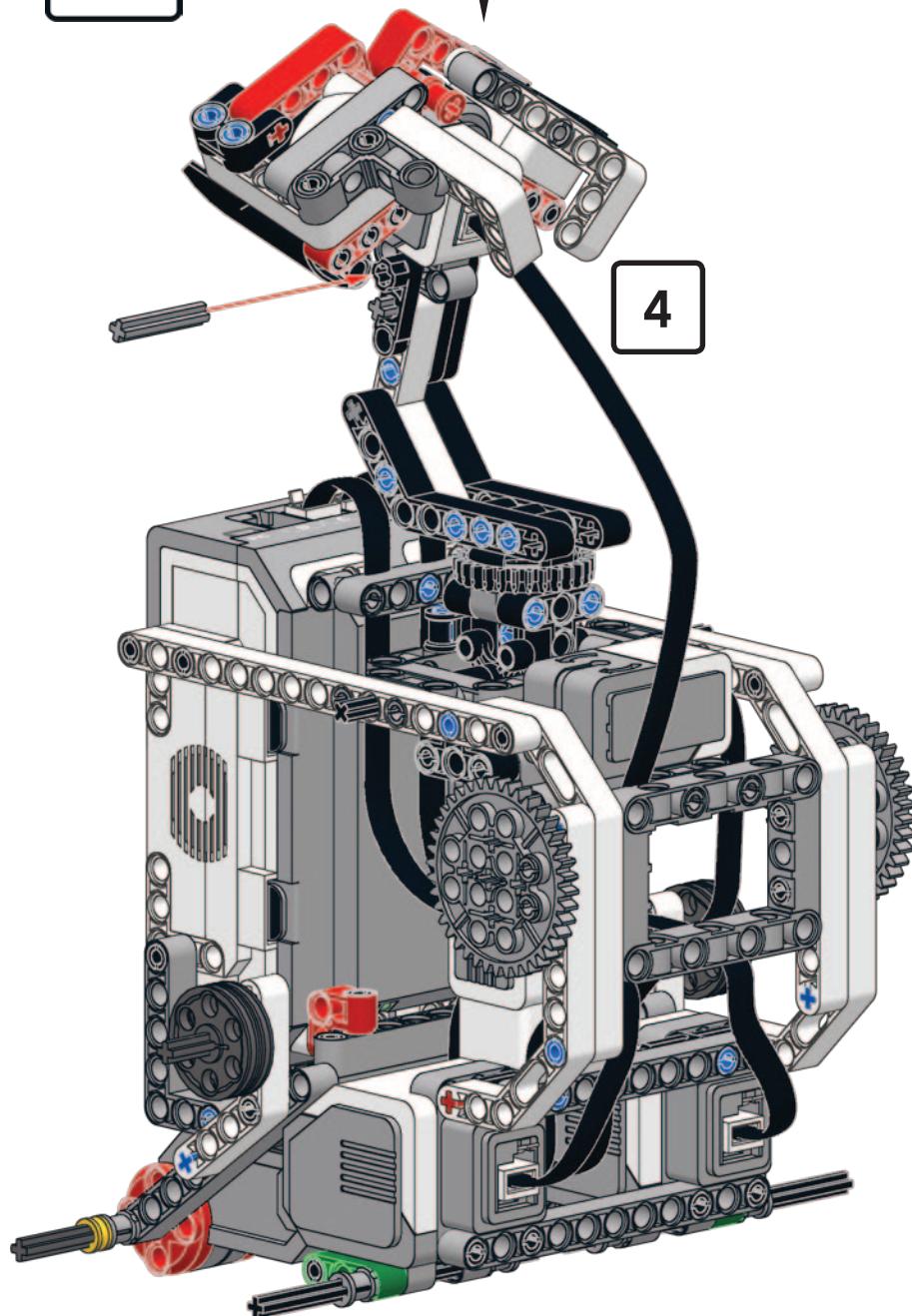
3



17



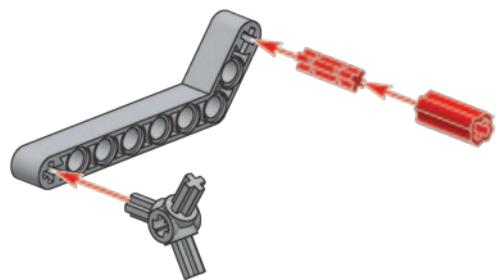
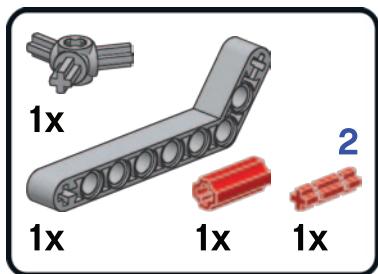
4



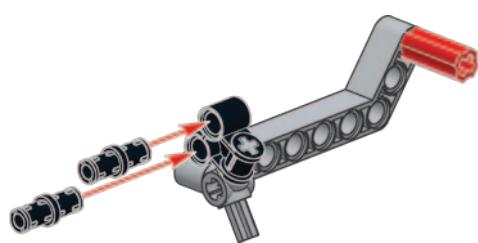


# Right arm subassembly

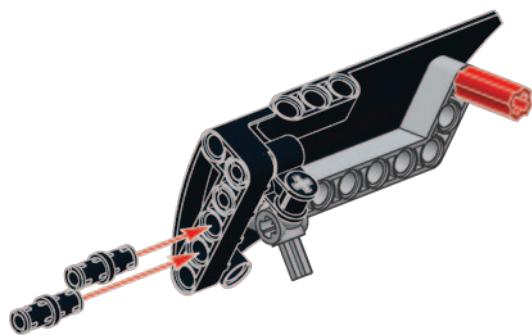
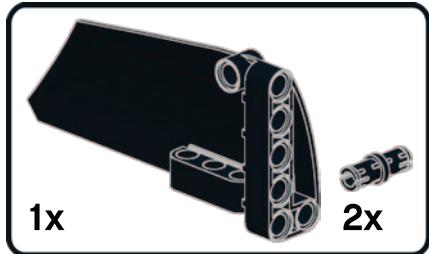
1



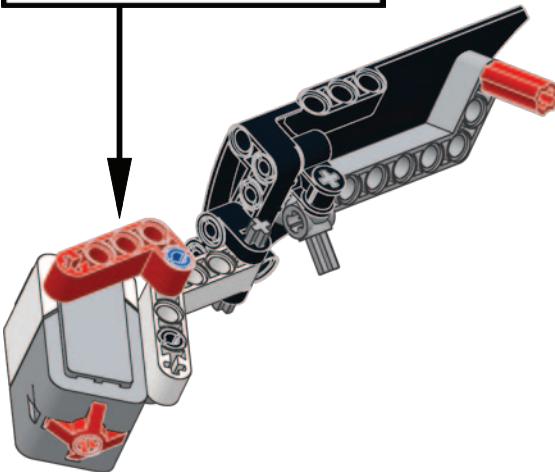
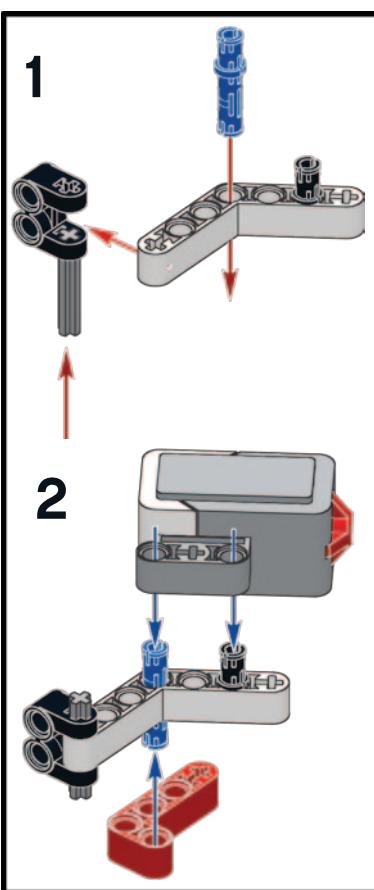
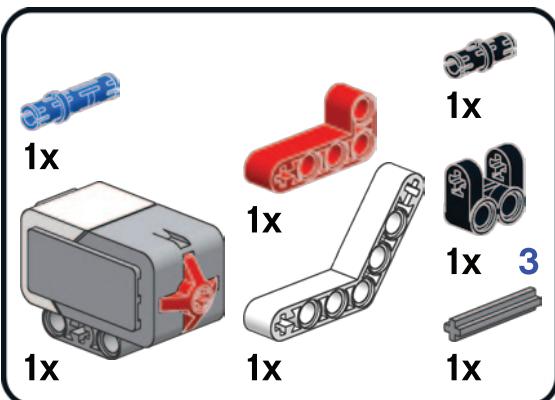
2



3



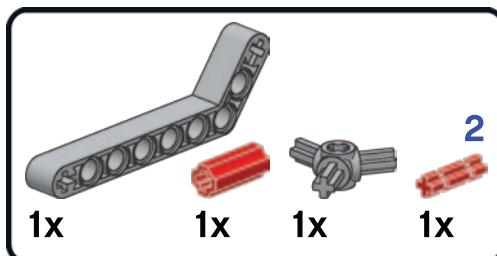
4



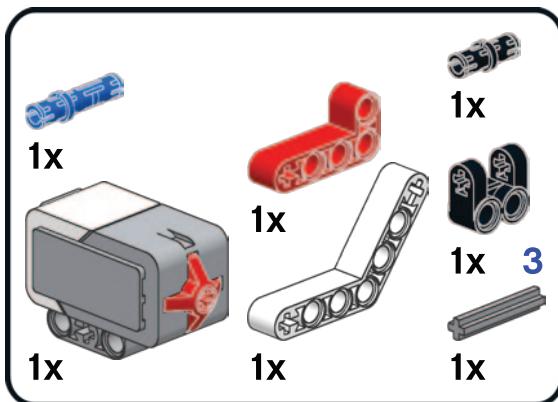


# Left arm subassembly

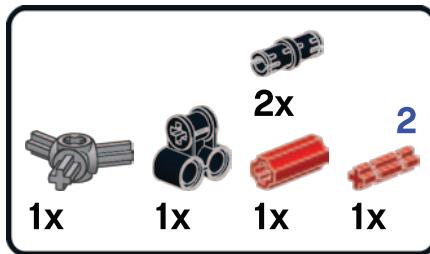
1



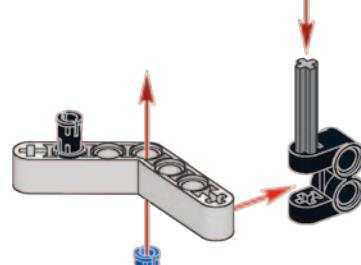
4



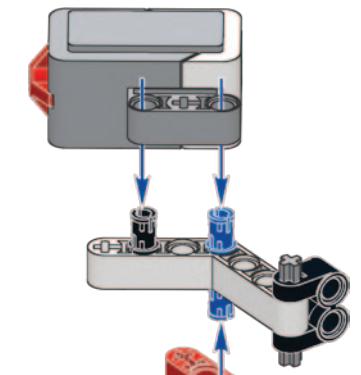
2



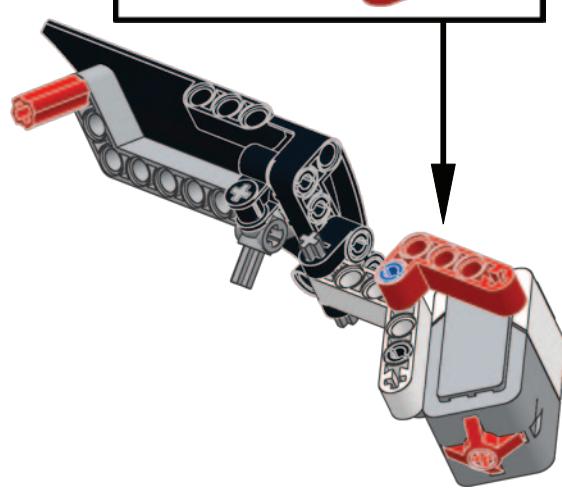
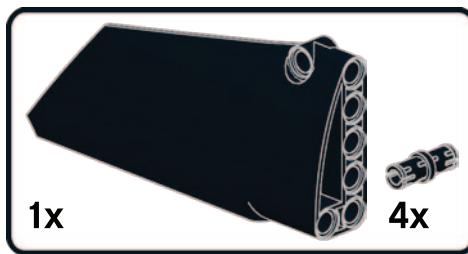
1



2

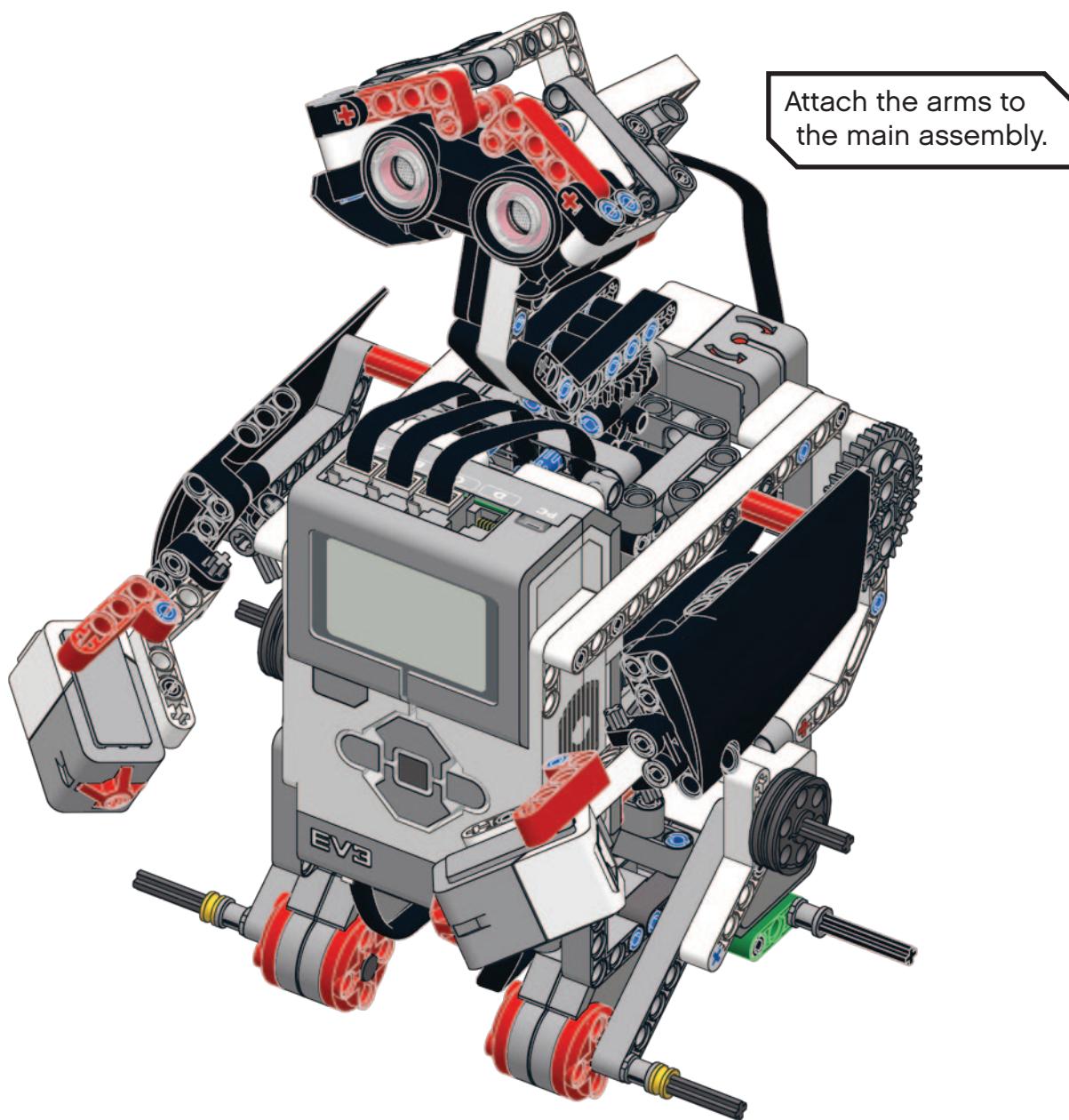


3





18

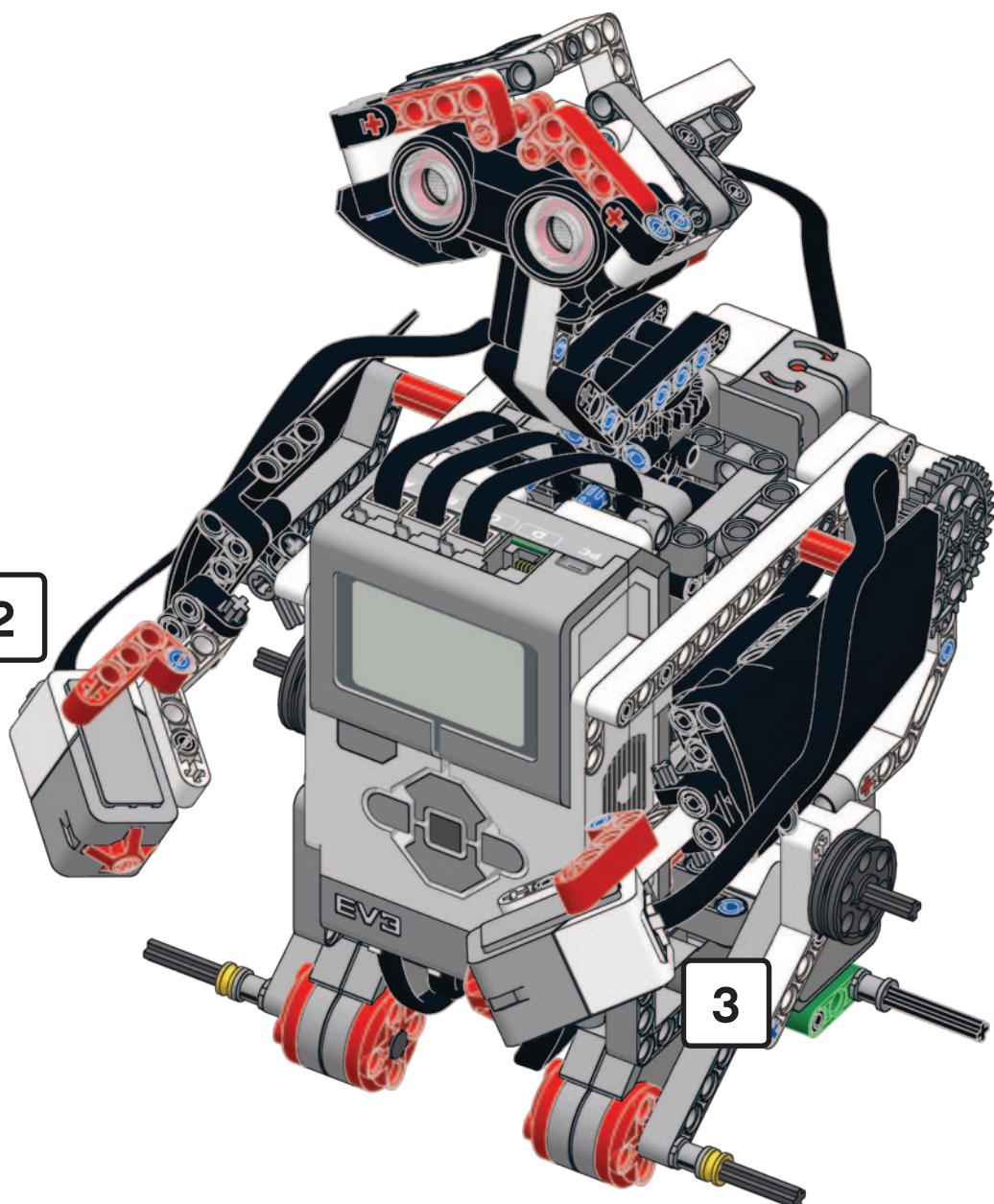




19



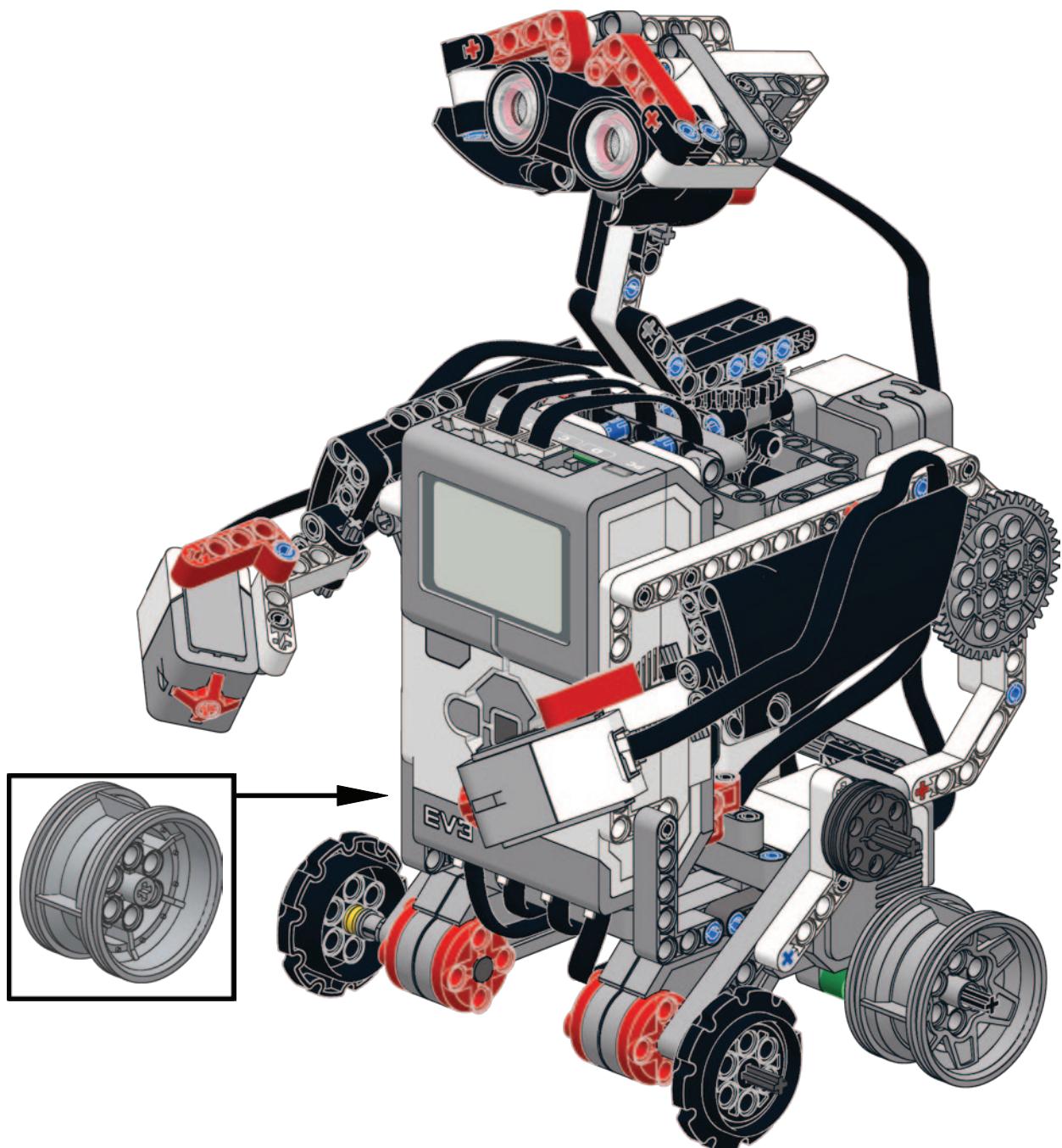
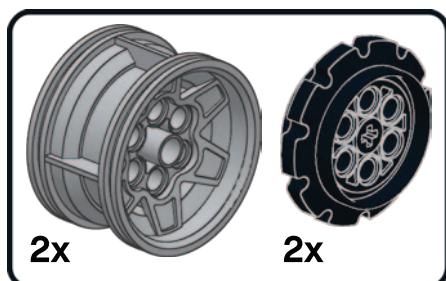
2x



If you do not need the touch sensors, you can replace them with the other two red 2x4 angular beams, to make the fingers.



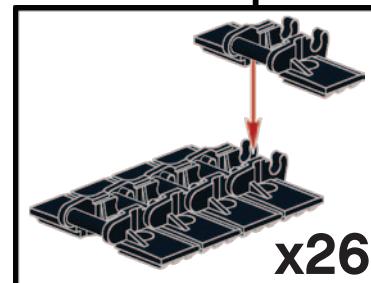
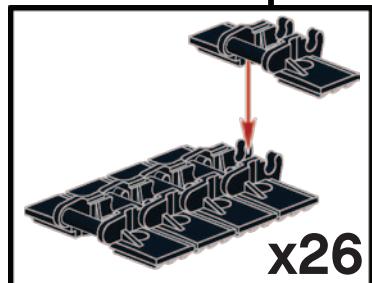
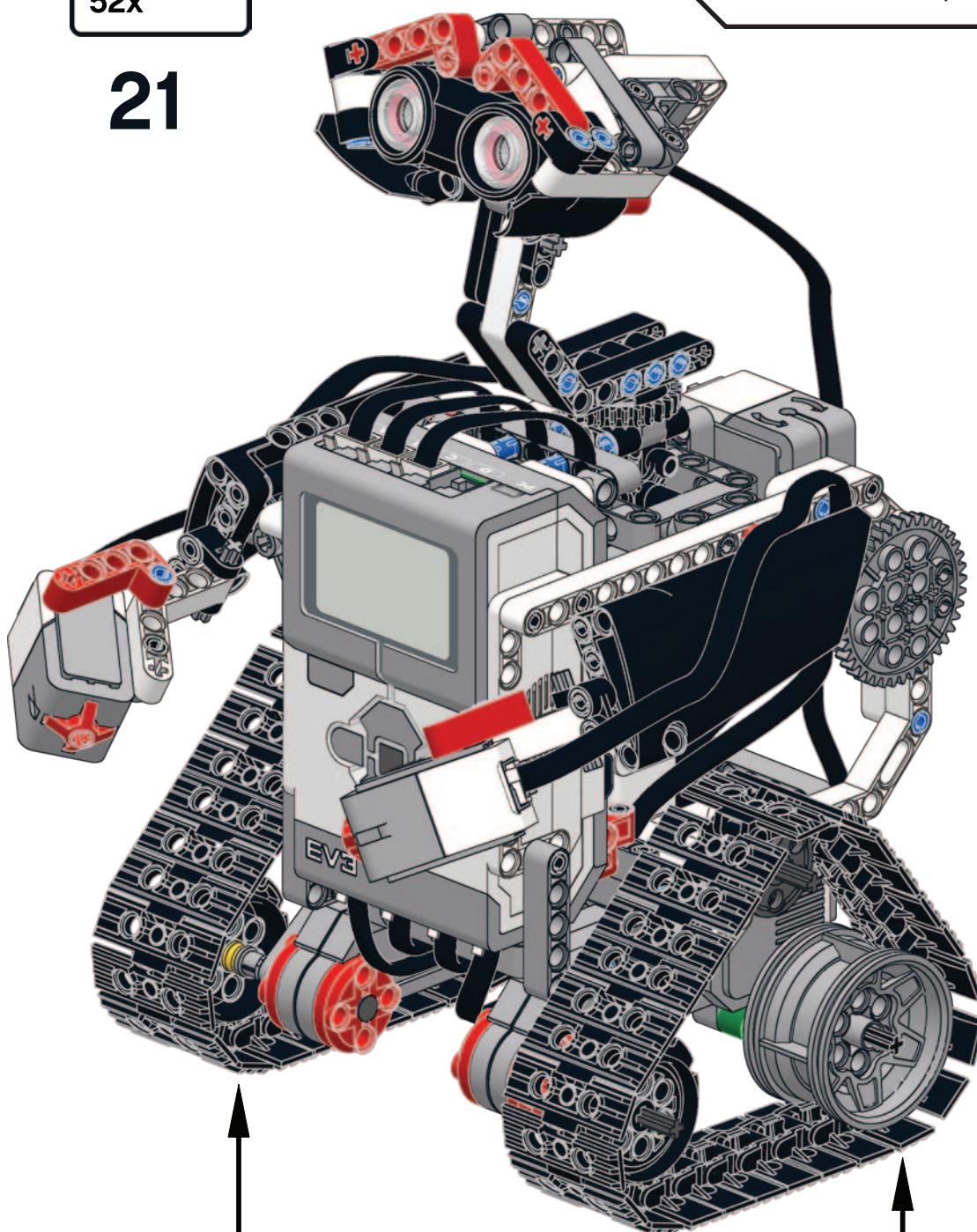
20





21

WALL•EV3 is complete!





# Programming WALL•EV3

## Precise Navigator

Robots that move on wheels and treads usually don't drive straight. Bumps on the ground, uneven floors, and wheel slippage can affect the robot trajectory.

The *Explore* program you are going to make (Figure 3) allows WALL•EV3 to travel perfectly straight and turn by a precise amount of degrees, by using the continuous feedback from the Gyro Sensor, which measures rotating motion around its axis (indicated by the red dot carved on its case).

Since the *Explore* program includes some My Blocks, you will start making the needed My Blocks first.

**NOTE** If you need to review how to make My Blocks, check out [The LEGO MINDSTORMS EV3 Laboratory book](#).

### My Block #1: RST

The *RST* My Block (Figure 1) resets the head position to the front and the arms to the middle. To do that, we use an Unregulated Motor block to move the motor **A** without maintaining automatically its power under load.

Then, a Wait block in **Motor Rotation - Compare - Current Power** mode waits until the actual speed of the motor decreases below the

threshold. This happens when the mechanism reaches a mechanical stop, and the motor stalls because it cannot move further. At that point, the head is moved back to the middle using a Medium Motor block. The motor encoder is reset, so that later its value will be 0 when the head is straight.

### My Block #2: ERR

The *ERR* My Block (Figure 2) makes WALL•EV3 play its typical worried sound effect, and shake the head and arms, to show complaint. This My Block will be used to make react to an obstacle and to your mistakes when playing a game.

### Making the Explore program

Now that you have created the needed My Blocks, you can make the *Explore* program (Figure 3). The program initializes the head and the arms, then resets the Gyro Sensor (a). In this phase, the robot has to be still on a flat surface parallel to the ground, so that that Gyro can be properly initialized.

Then, in a loop, the robot travels straight until it encounters an obstacle, turns by 90 degrees and starts travelling forward again.

To make the robot drive perfectly straight, even if its treads slip, and it encounters bumps

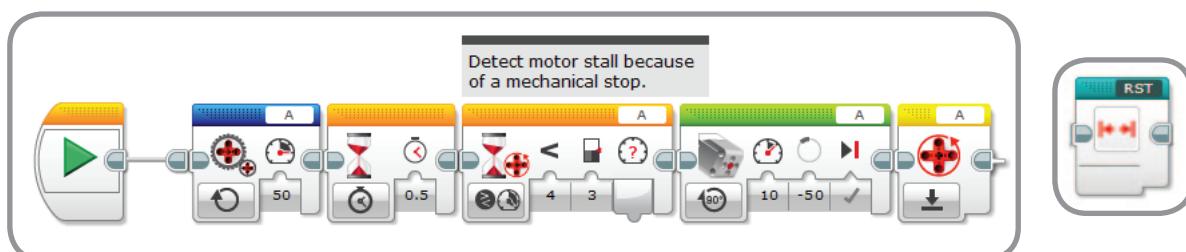


Figure 1. The RST My Block

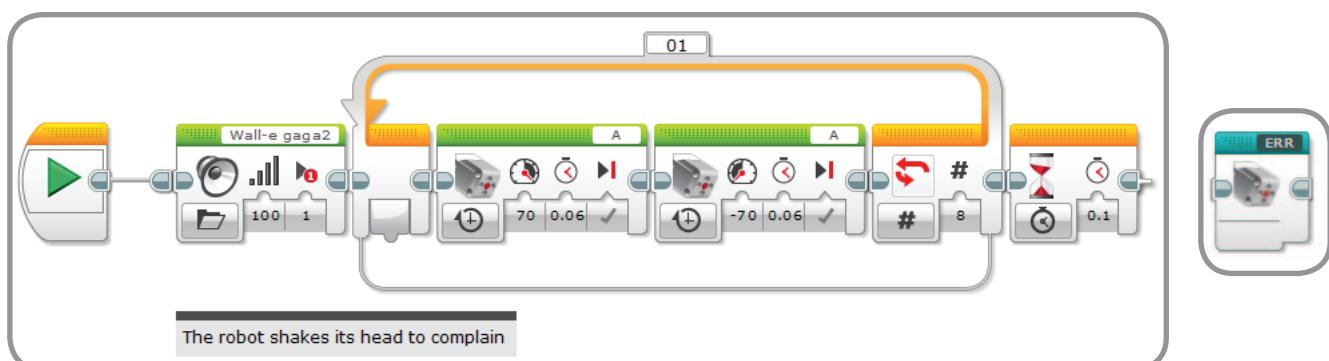


Figure 2. The ERR My Block



or holes in the ground, the program uses a proportional feedback controller.

This controller (b) is implemented by a simple formula that computes a steering command for the Move Steering block by multiplying the reading from the Gyro by a number (the controller gain).

**NOTE** For more details about a feedback controller, check out [The LEGO MINDSTORMS EV3 Laboratory book](#).

So, if the robot is drifting to one side, the Gyro will detect this change in heading, the steering command will change, correcting the driving direction. Simple yet very effective.

To spin in place by a precise amount of degrees, the Move Steering My Block is set to spin in place (Steering parameter set to 100). You use a Wait block in **Gyro Sensor - Change - Angle** mode to wait for the Gyro reading to decrease by 91 degrees. The extra degree is needed to compensate for small system delays when continuously polling the sensor. Then, the Gyro is reset again (d), so that the robot will continue driving straight in the new direction.

## Follow My Hand!

The *HandFollow* program (Figure 6) makes WALL-EV3 follow your hand. The robot turns the head to the right, to the front and to the left in a loop, and when it sees your hand, it moves towards it. Since this program includes the *LookAndFollow* My Block that in turn uses the *TurnHead* My Block, you will make this first.

## My Block #3: TurnHead

The My Block (Figure 4) makes the Medium Motor shaft rotate to an absolute position determined by the last time its rotation count was reset (regardless of its actual position). The head is aligned and the motor rotation count is reset in the *RST* My

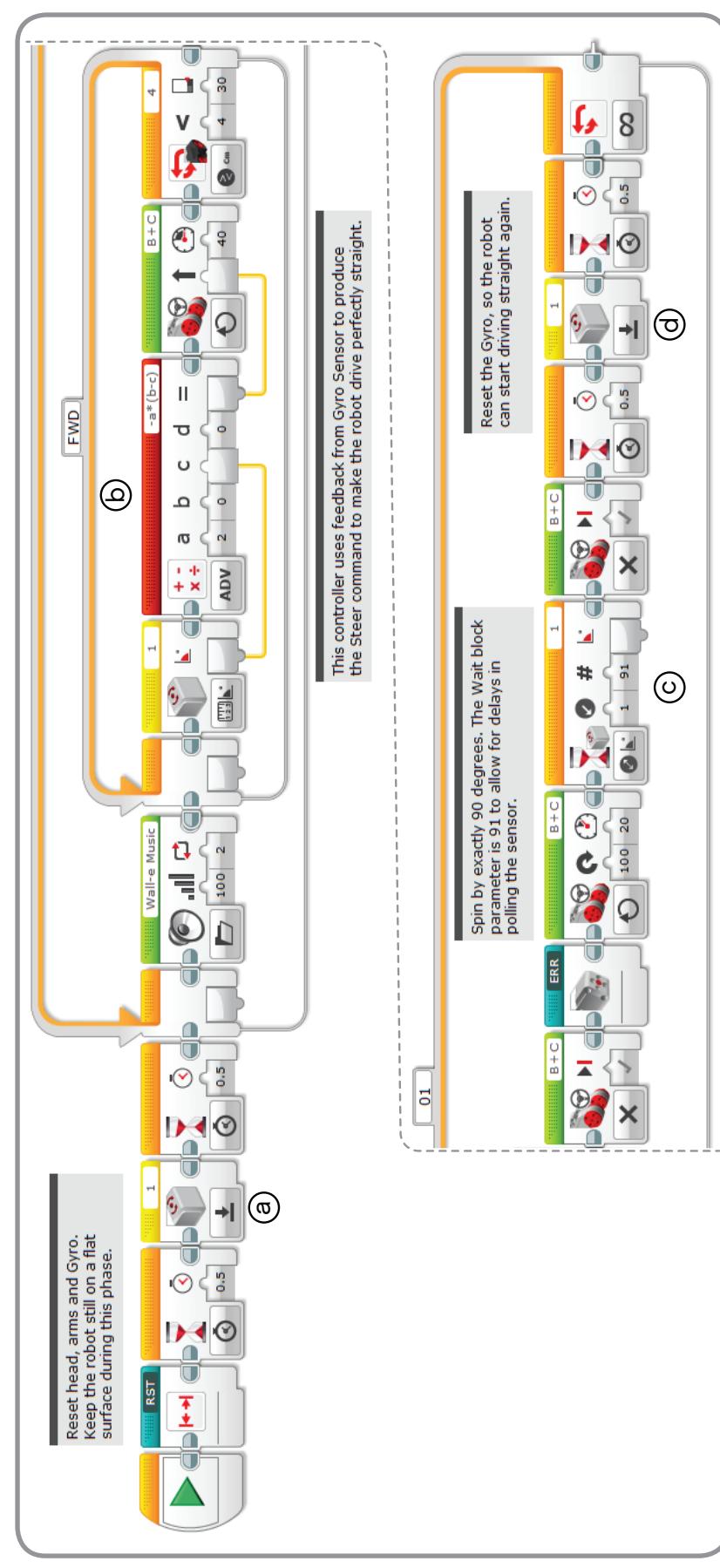


Figure 3. The Explore Program

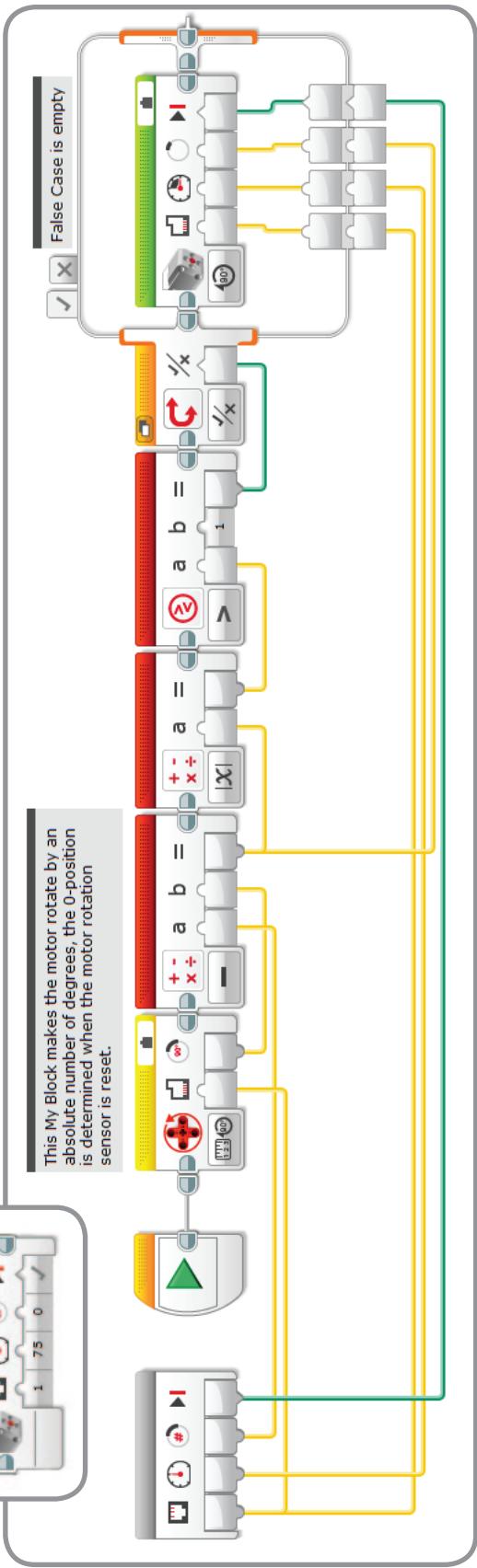


Figure 4. The TurnHead My Block

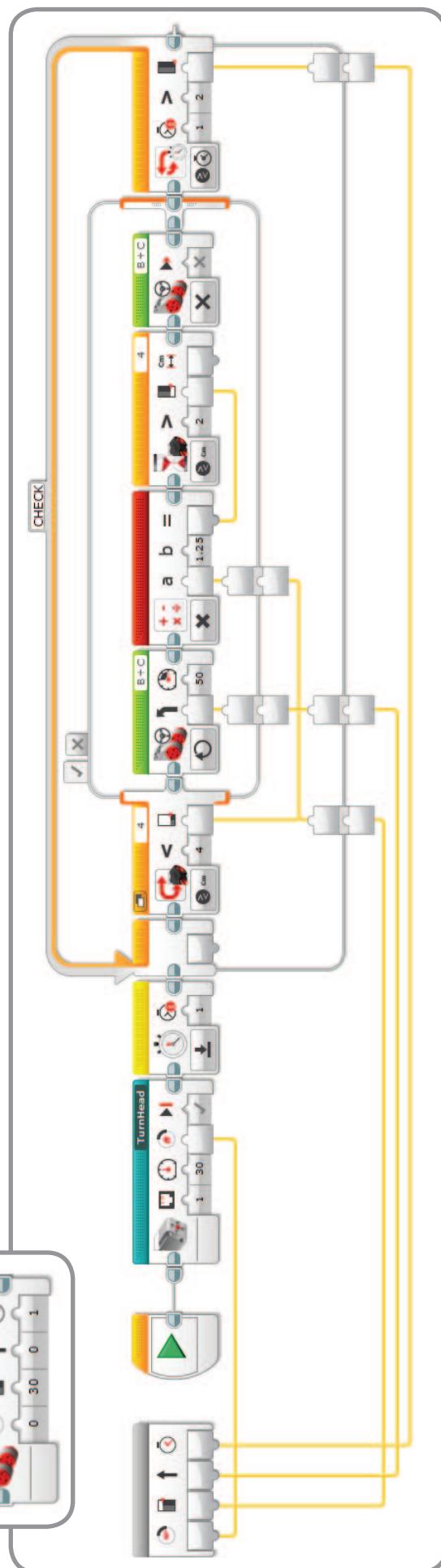


Figure 5. The LookAndFollow My Block



Block at the beginning of the program.

Build this My Block using Figure 4 as reference, and configure its inputs as indicated in Table 1.

*Table 1. TurnHead My Block parameters configuration*

Name	Type	Data Type	Default Value	Style	Limits
Port	Input	Number	1	Text Input	
Power	Input	Number	75	Horiz. Slider	-100 100
Angle	Input	Number	0	Horiz. Slider	-50 50
Brake at End	Input	Logic	True		

# My Block #4 : LookAndFollow

The *LookAndFollow* My Block (Figure 5) includes a sequence of blocks to make the robot turn the head and check for a certain amount of time whether the Ultrasonic Sensor detects your hand. If within this time, the hand is seen, the robot drives towards it. The amount of steering is passed to the My Block as an Input parameter. To check for obstacles within a certain interval of time, we reset Timer 1, and check the Ultrasonic Sensor in a loop that ends when the Timer 1 exceeds the Pause threshold. Build the My Block using Figure 5 as a reference, and configure its inputs as indicated in Table 2.

*Table 2. LookAndFollow My Block parameters*

Name	Type	Data Type	Default Value	Style	Limits
Head Angle	Input	Number	0	Horiz. Slider	-40 40
Distance	Input	Number	30	Vert. Slider	5 255
Steering	Input	Number	0	Horiz. Slider	-100 100
Pause	Input	Number	1	Text Input	

# Making the HandFollow Program

Now that you have created the needed My Blocks, you can make the *HandFollow* program (Figure 6).

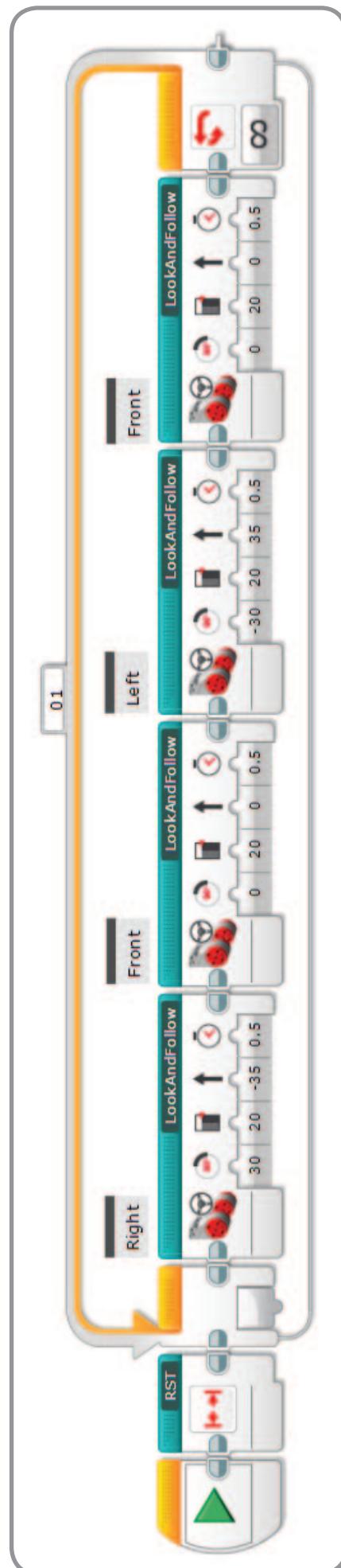


Figure 6. The HandFollow program



# Number Guessing Game

WALL•EV3 can play a simple guessing game. For each turn he thinks of a random number between 2 and 8, then it waits for you to guess whether the next number will be bigger or smaller than the current one.

You can express your choice by pressing the left hand button to say it's smaller, the right hand button to say it's bigger. If you guessed it right, WALL-EV3 will cheer and add a point to your score, it will shake his head and complain otherwise. For each mistake you make, the variable *Life* will be decreased by one. When it reaches 0, the game will end, and your final score will appear on the EV3 Brick screen.

Before making the Game program using Figure 11 and Figure 12 as references, you will make the needed My Blocks first.

# My Block #5: ShowLife

This My Block shows on the screen a progress bar according to the *Life Input* parameter value. Here we use the image resources included in the EV3 Software, *Bar 0*, *Bar 1*, *Bar 2*, *Bar 3*, *Bar 4*, selected using a Switch Block in **Flat View**, **Numeric mode**.

Build the ShowLife My Block using Figure 7 as reference and configure the the Input parameter as indicated in Table 3

Table 3. ShowLife My Block parameters

Name	Type	Data Type	Default Value	Style	Limits
Life	Input	Number	0	Horiz. Slider	0 4

## My Block #6: Random

The Random My Block (Figure 9) generates the two different random numbers, the first between 2 and 8, the second between 1 and 9. To make sure the numbers are different, the second one is drawn until it is different from the first one that had been drawn before the loop.

Make the Random My Block using Figure 9 as reference and configure the two numeric outputs by naming them *A* and *B*.

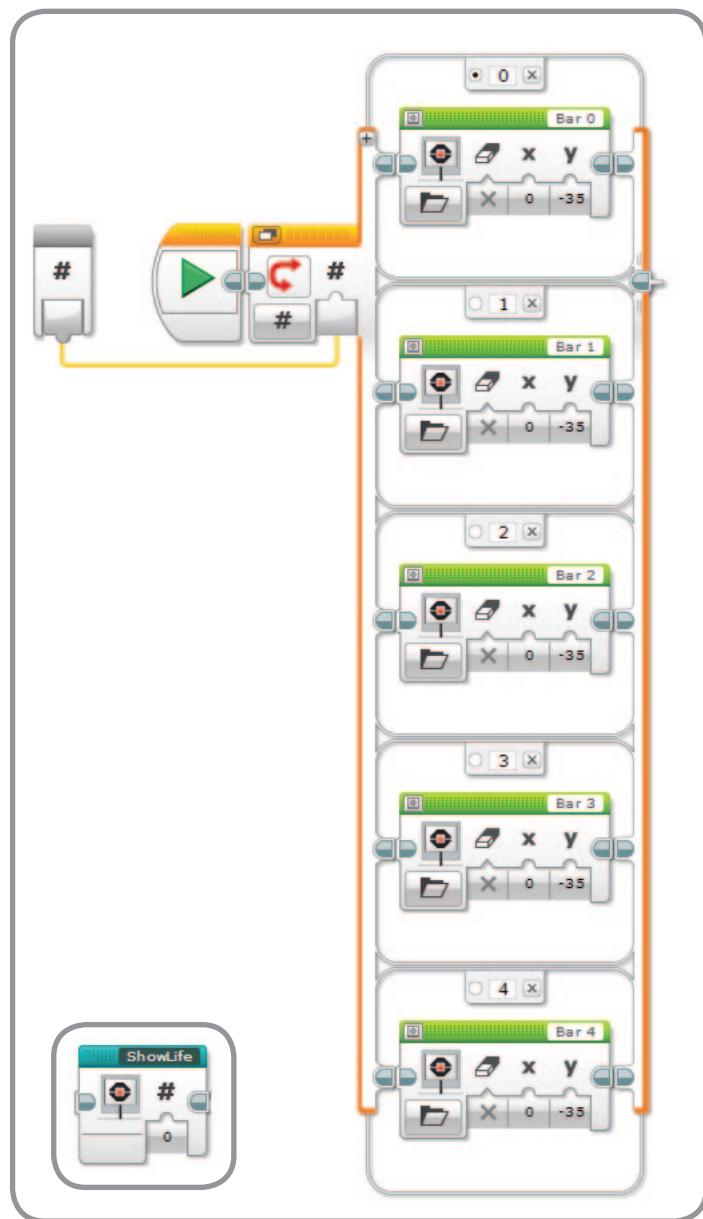


Figure 7. The ShowLife My Block

# My Block #7: Flash

The *Flash My Block* makes the EV3 Brick Status Lights flash as many times as specified by the input parameter. Make this My Block using Configure the Input parameter as indicated in Table 4.

*Table 4. Flash My Block parameters*

Name	Type	Data Type	Default Value	Style	Limits
Times	Input	Number	0	Text Input	



## My Block #8: Cheer

The Cheer My Block (Figure 10) makes the robot dance and whistle joyfully. Make this My Block using Figure 10 as reference, and configure the Input parameter as indicated in Table 5.

Table 5. Cheer My Block parameters

Name	Type	Data Type	Default Value	Style	Limits
Times	Input	Number	1	Text Input	

## Conclusion

I hope you liked building, programming and playing with WALL•EV3, my LEGO MINDSTORMS EV3 version of WALL-E, as much as I enjoyed designing it. To deepen some topics and get more programming knowledge, read *The LEGO MINDSTORMS EV3 Laboratory book*, and visit <http://EV3L.com>.

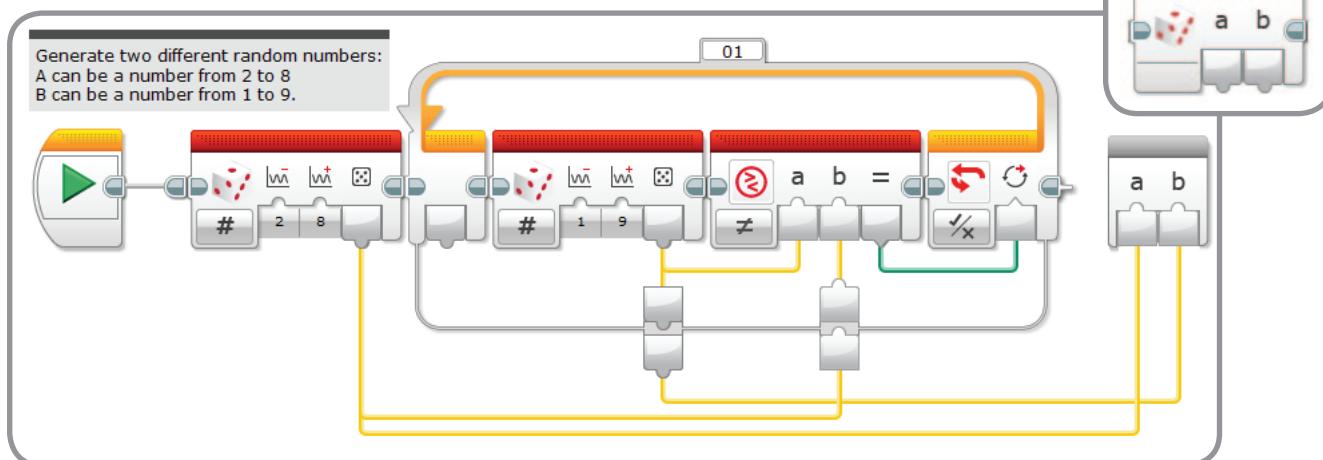


Figure 8. The Random My Block

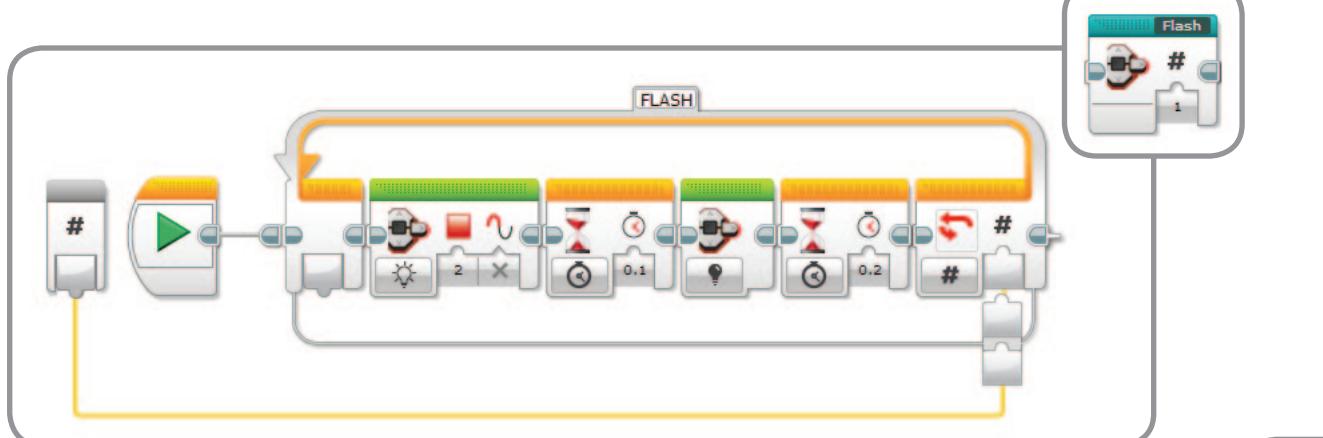


Figure 9. The Flash My Block

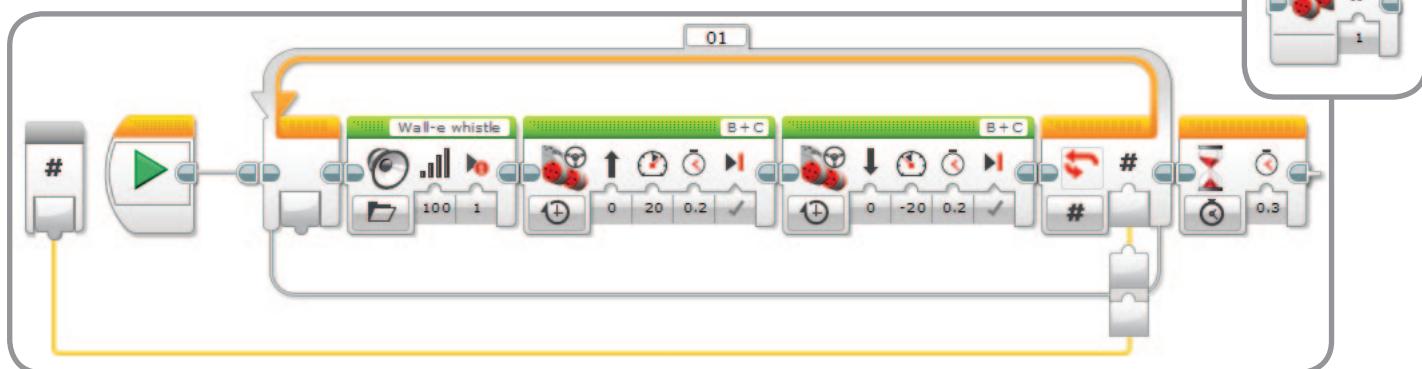


Figure 10. The Cheer My Block

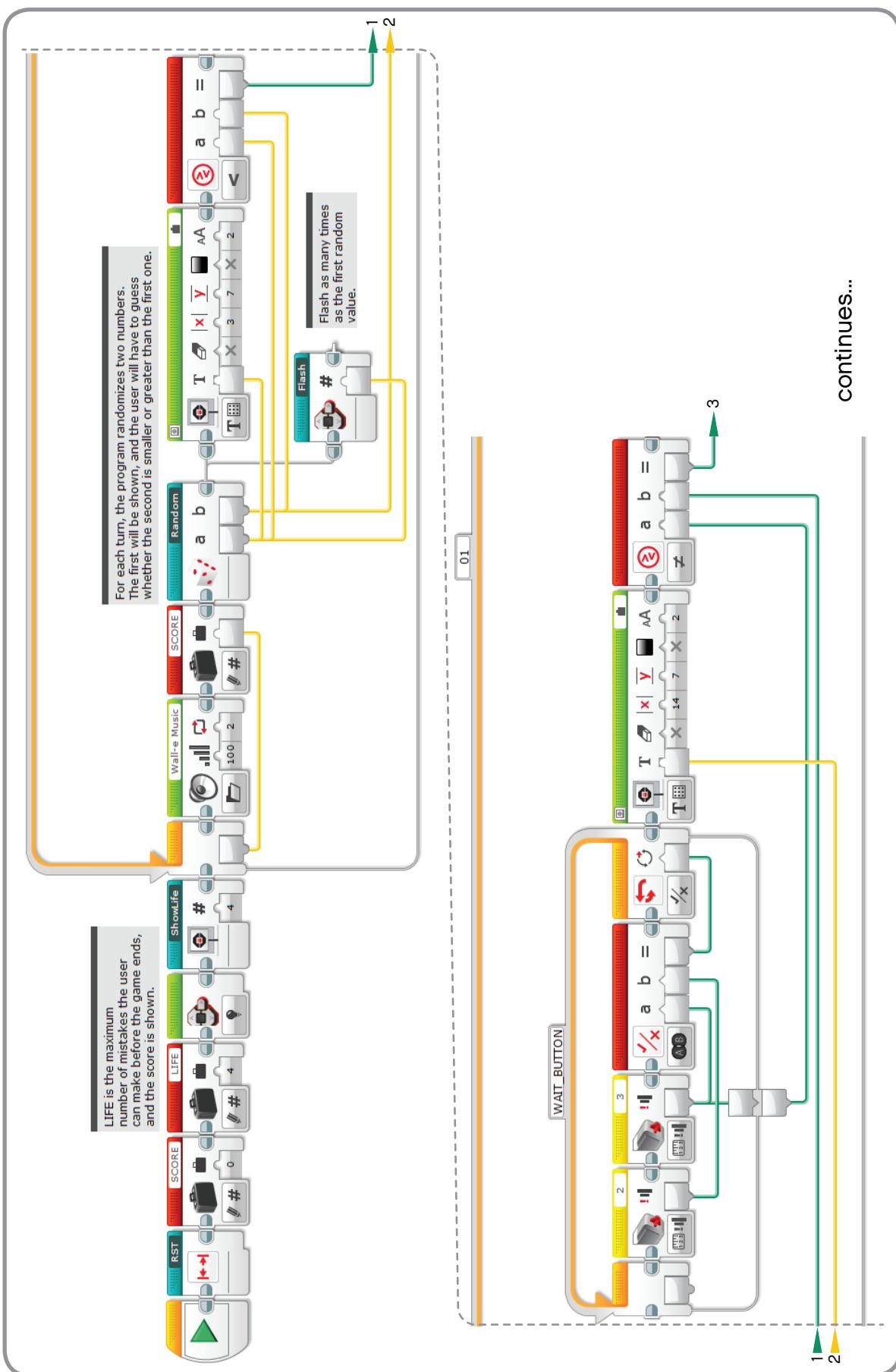


Figure 11. The Game program (continues).

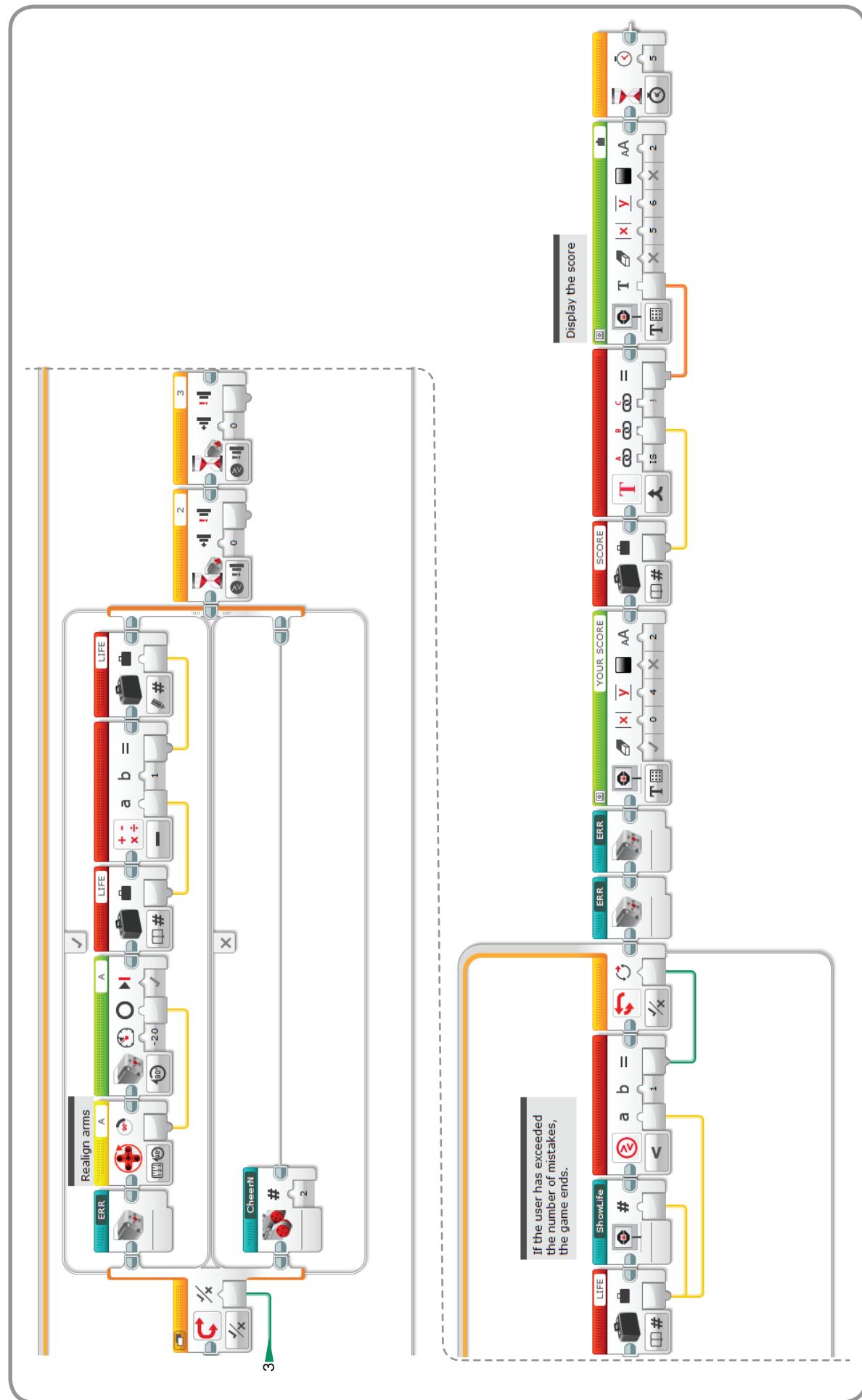


Figure 12. The Game program (continued).