


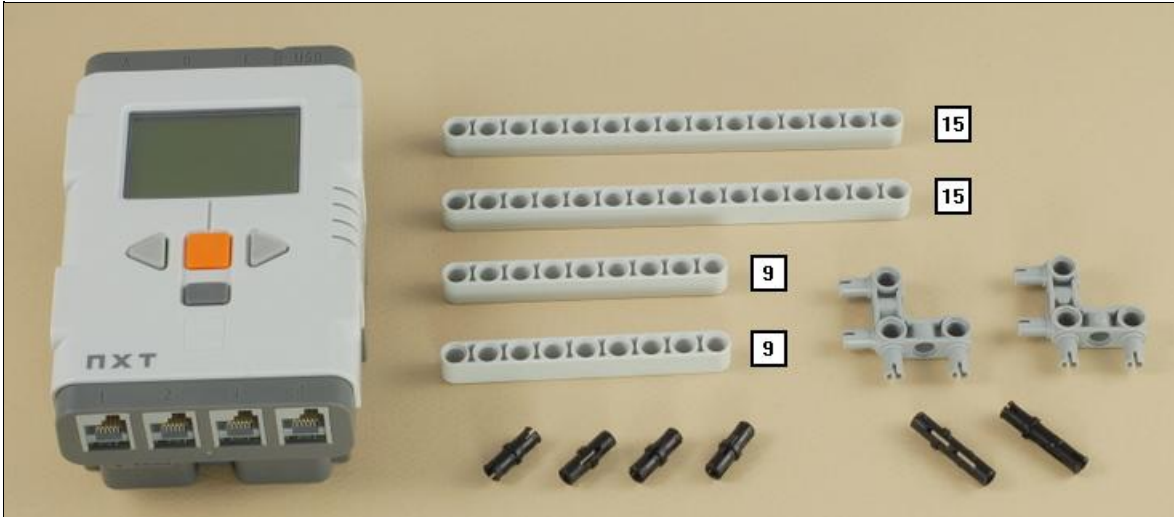


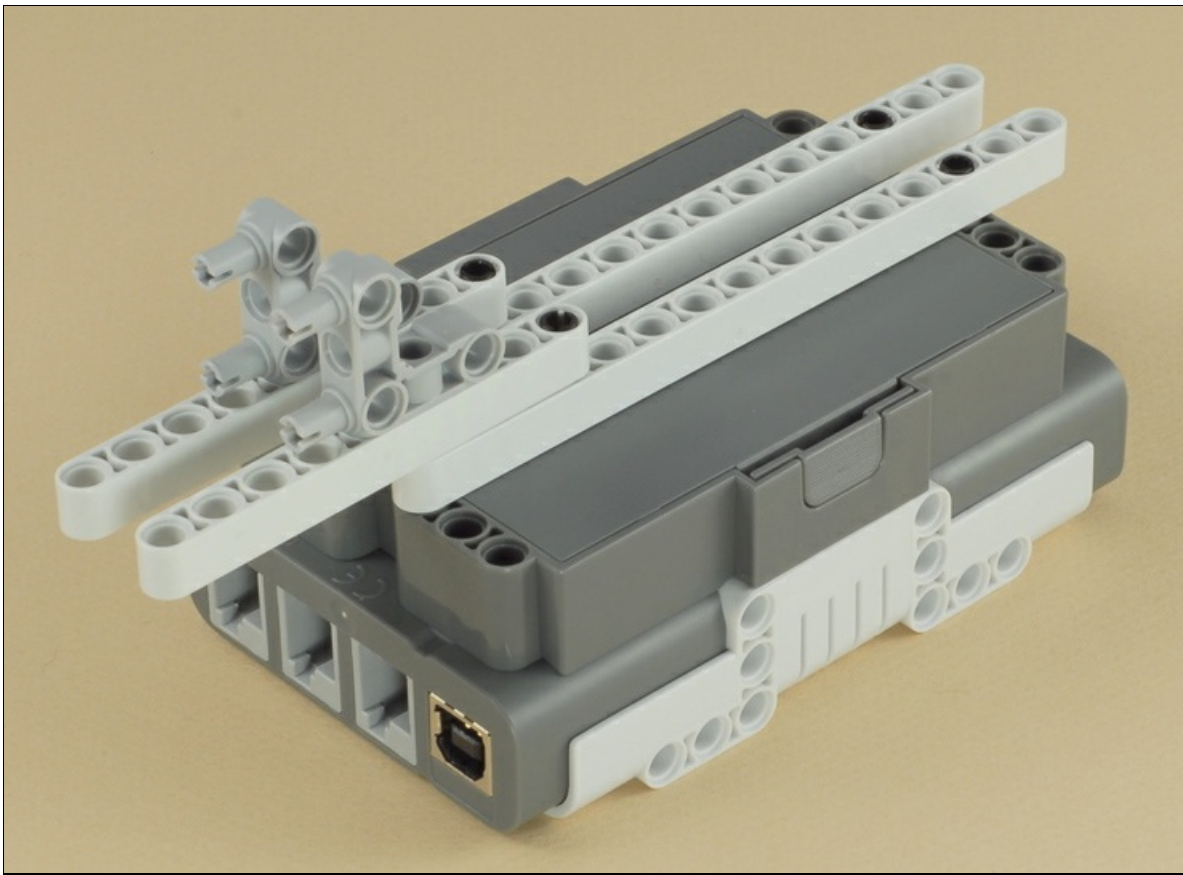
Castor Bot

[Building:](#)   [Program:](#) (none)Designed for **NXT 1.0** (8527, or 9797 + 9695/9648)

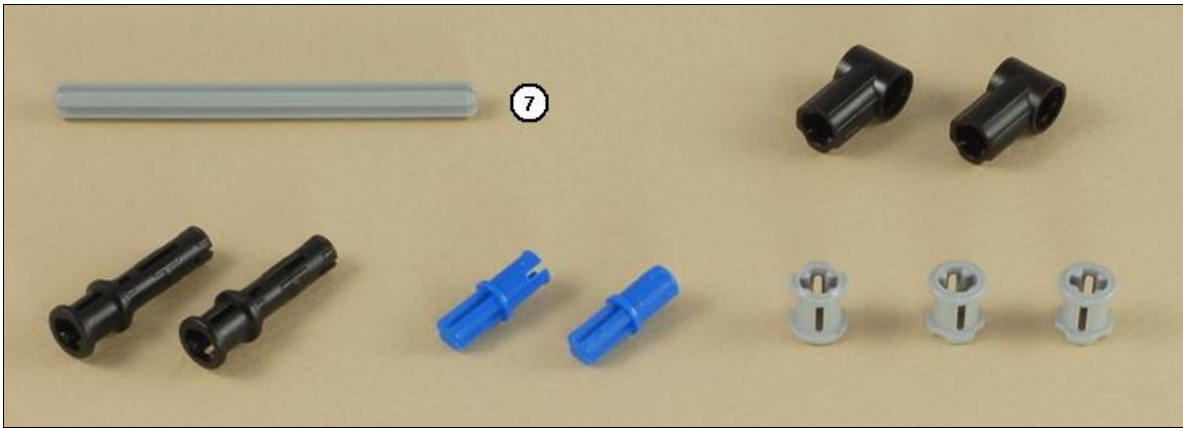
Building Instructions

1

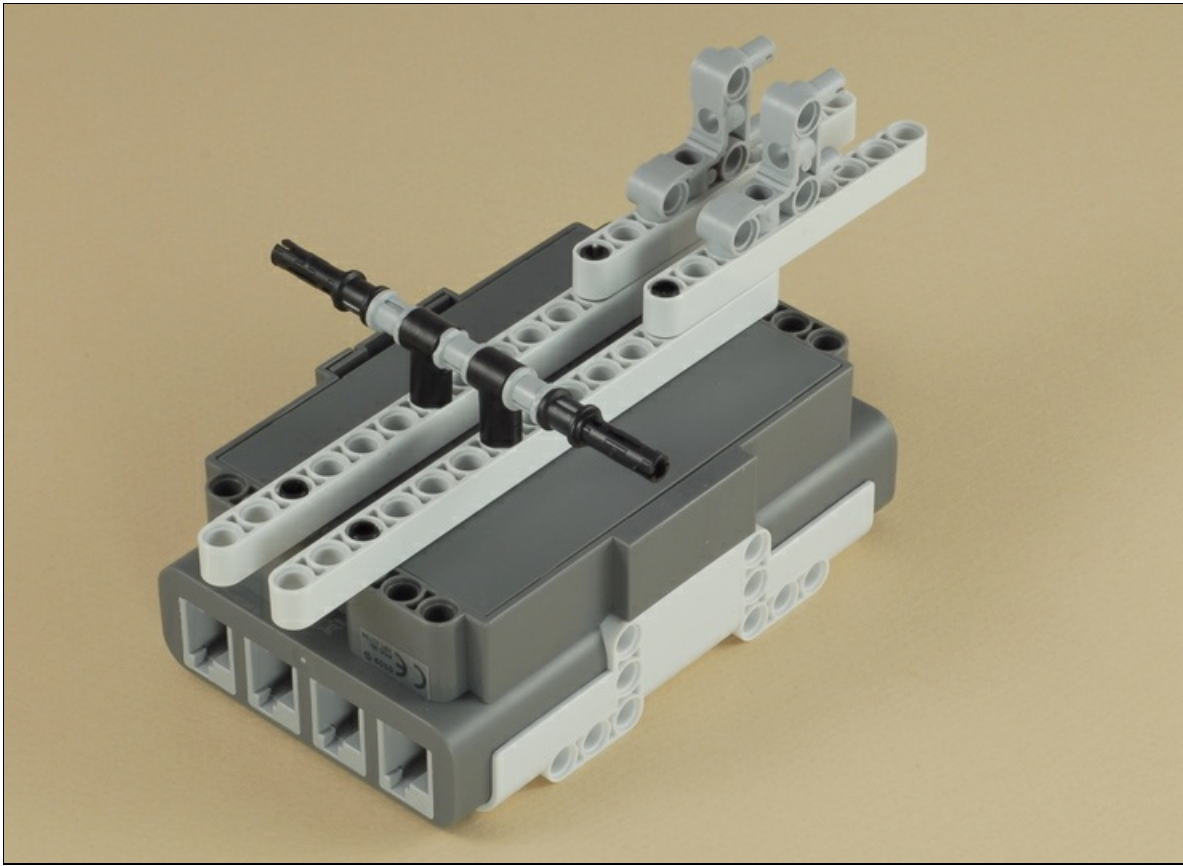




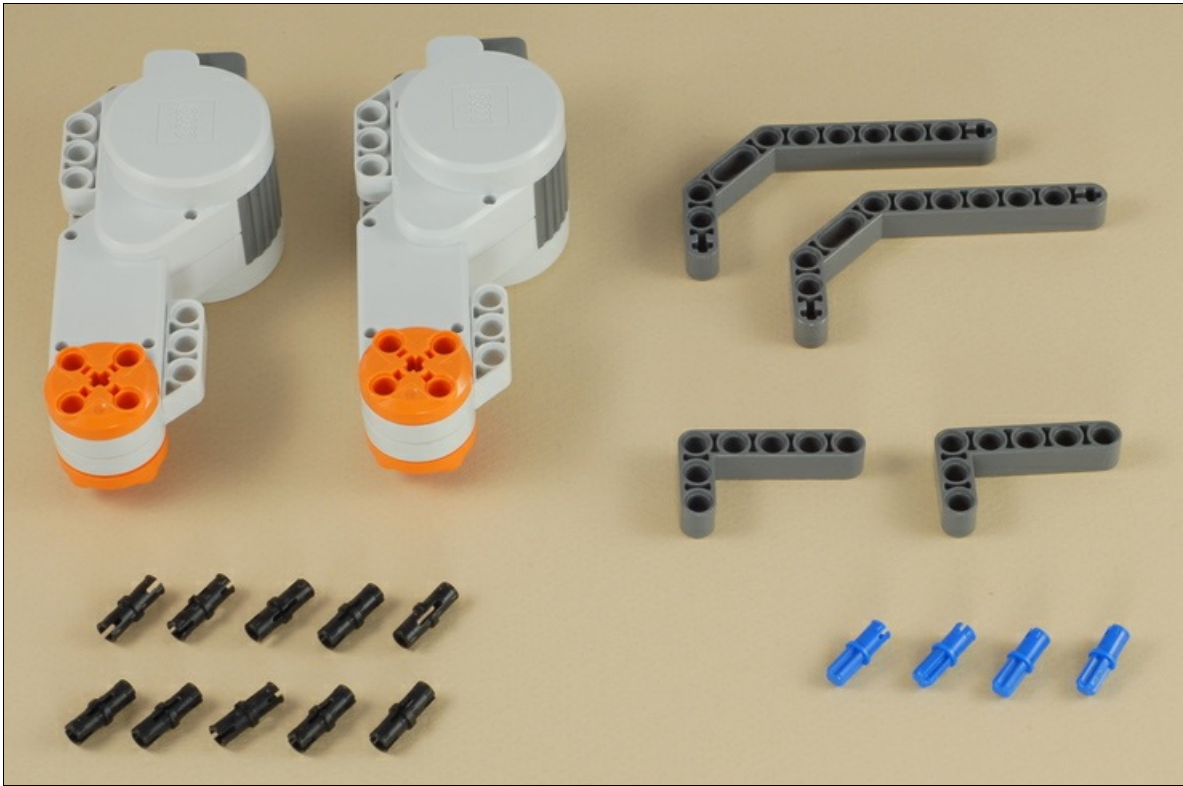
2

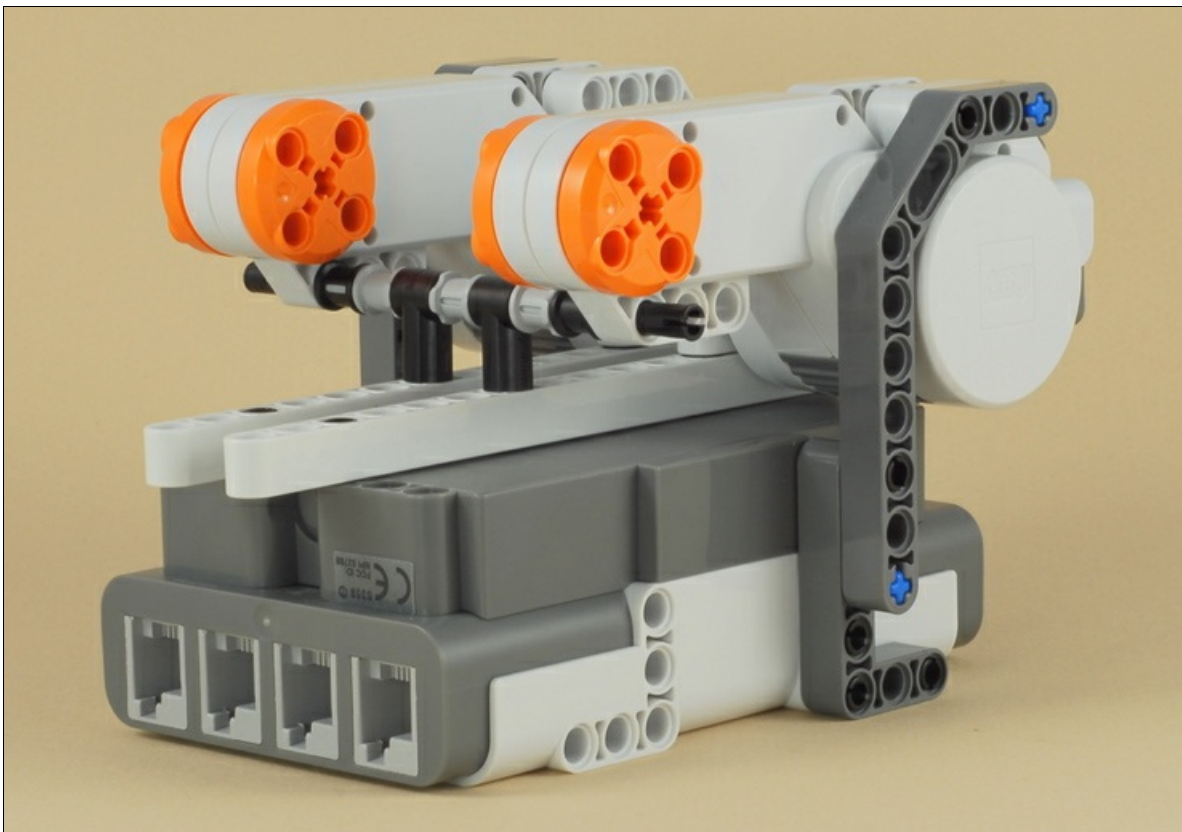
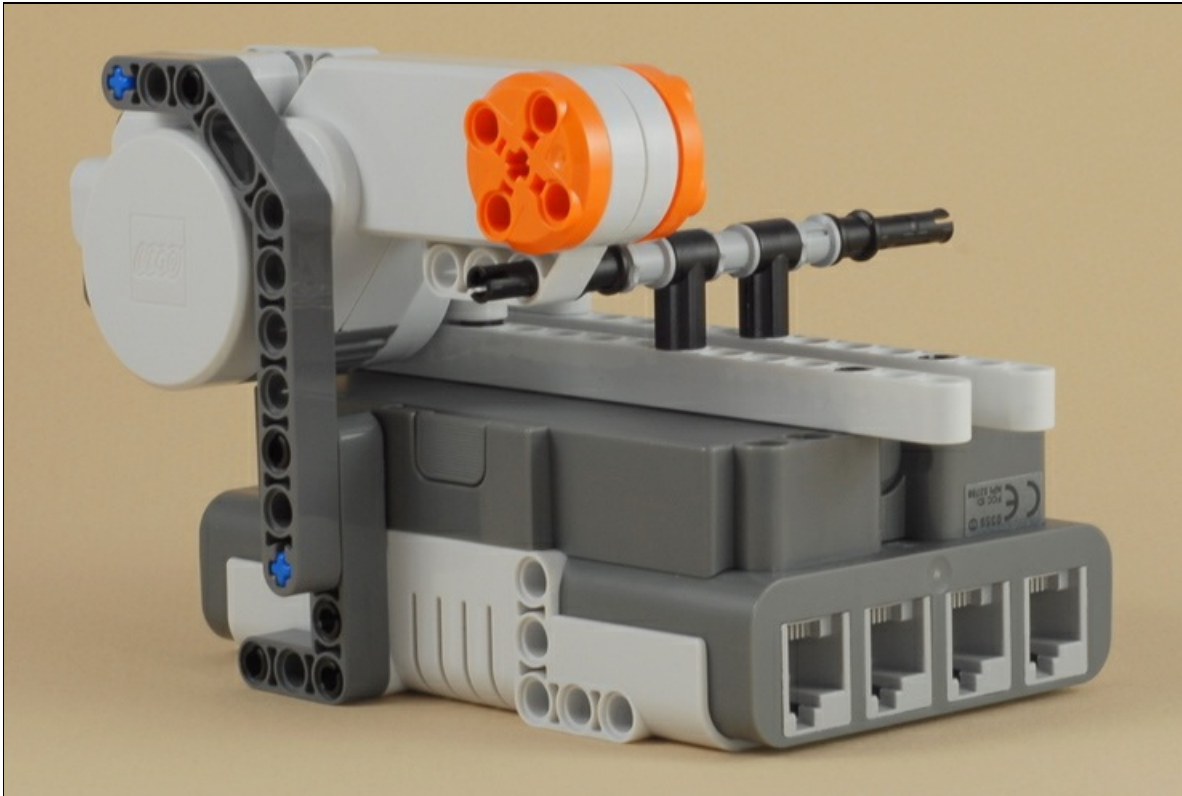




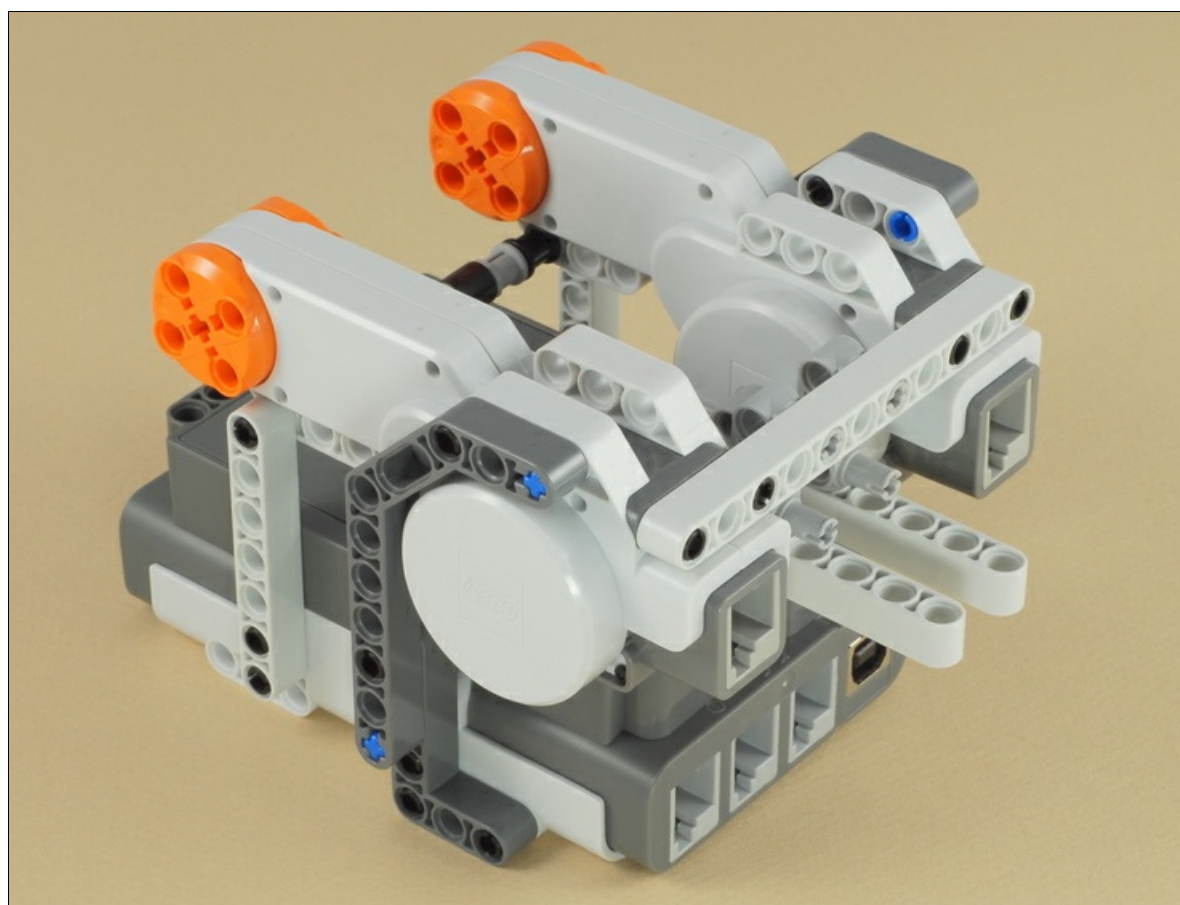
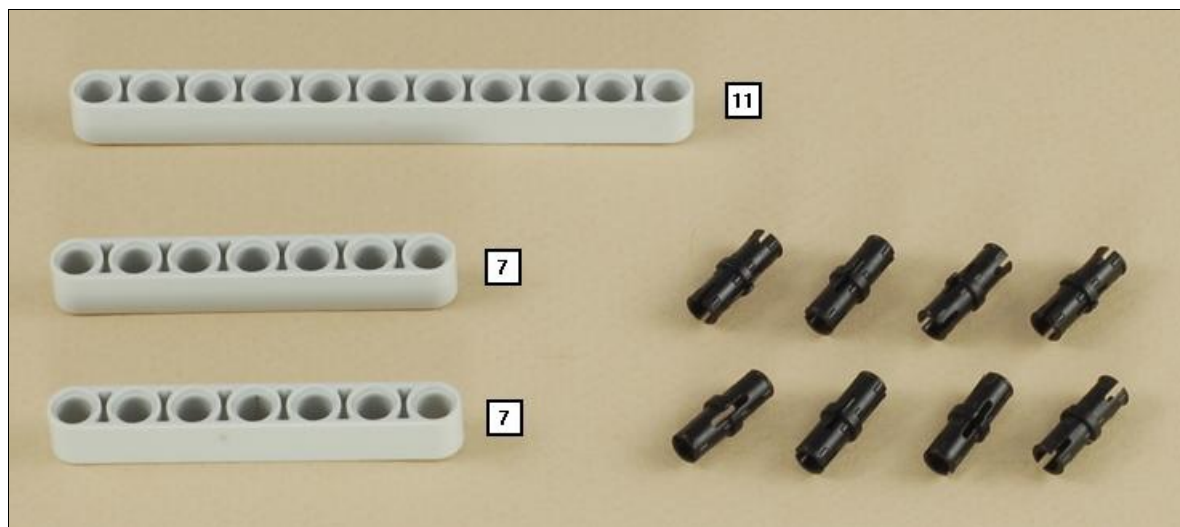


3

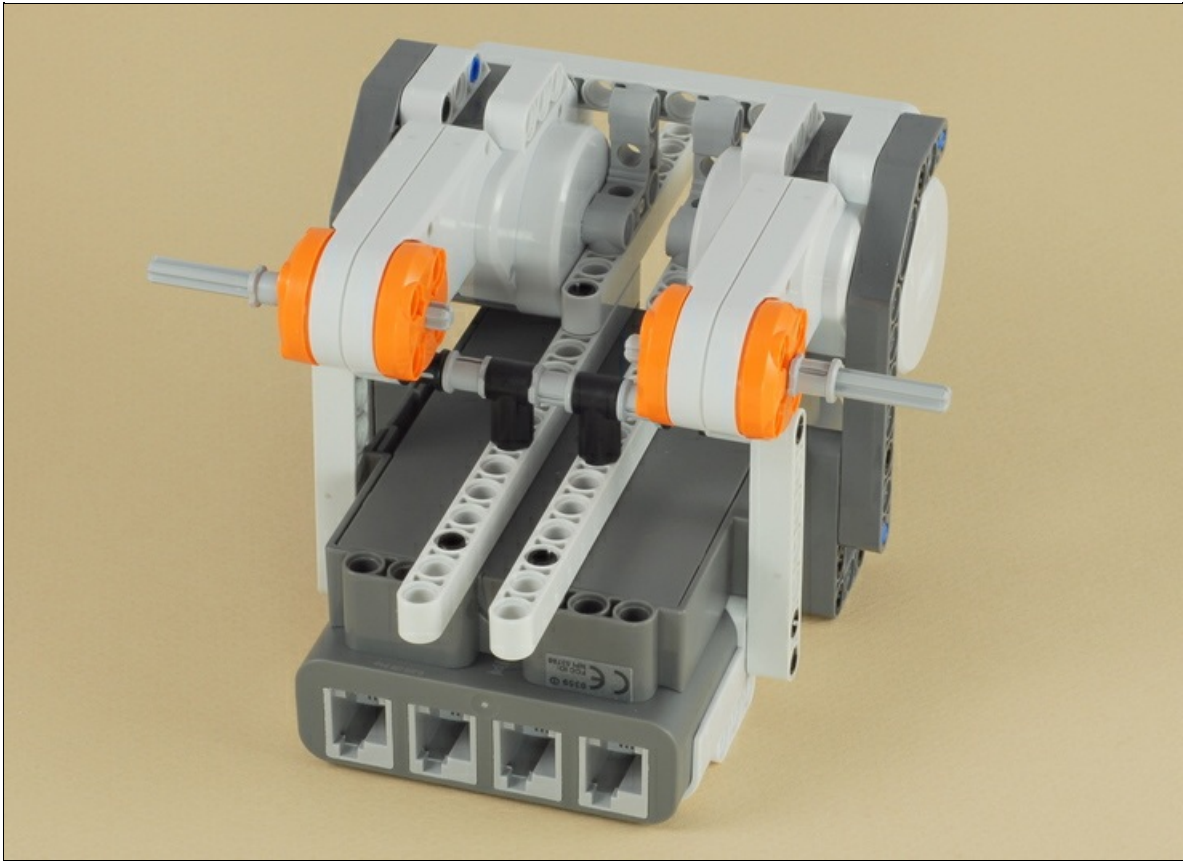
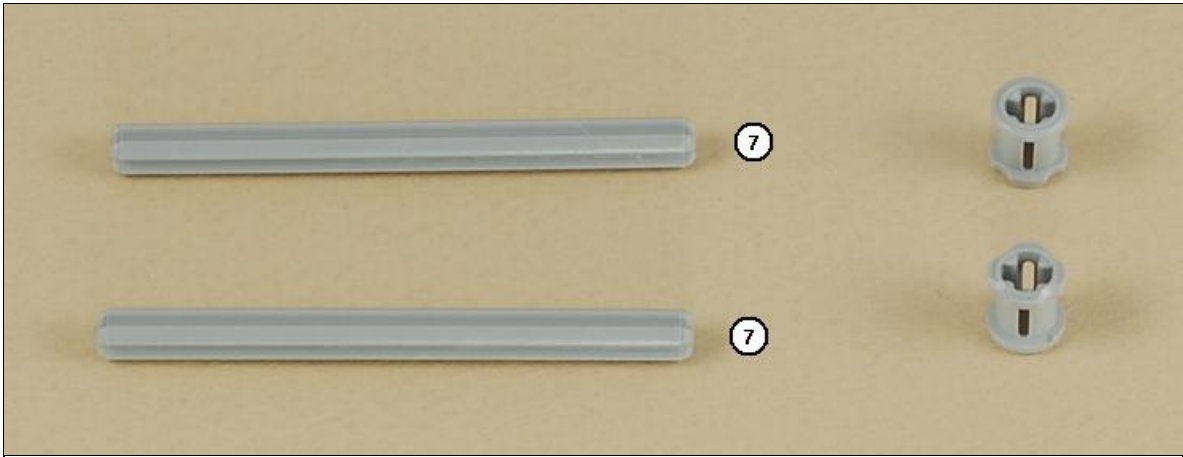




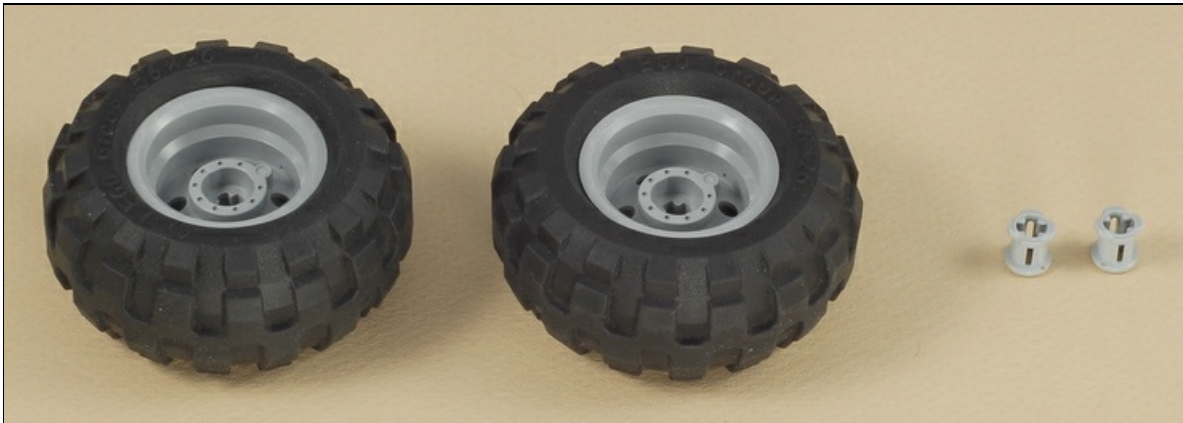
4

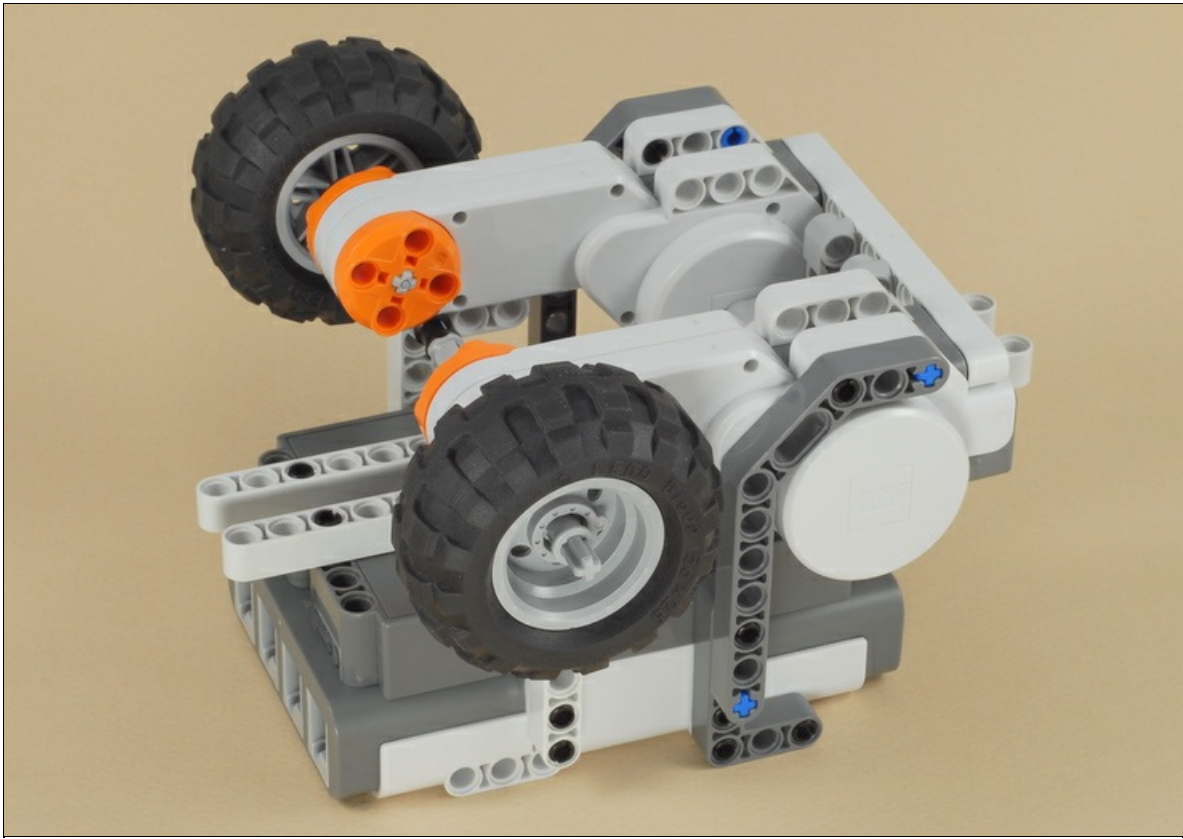


5



6





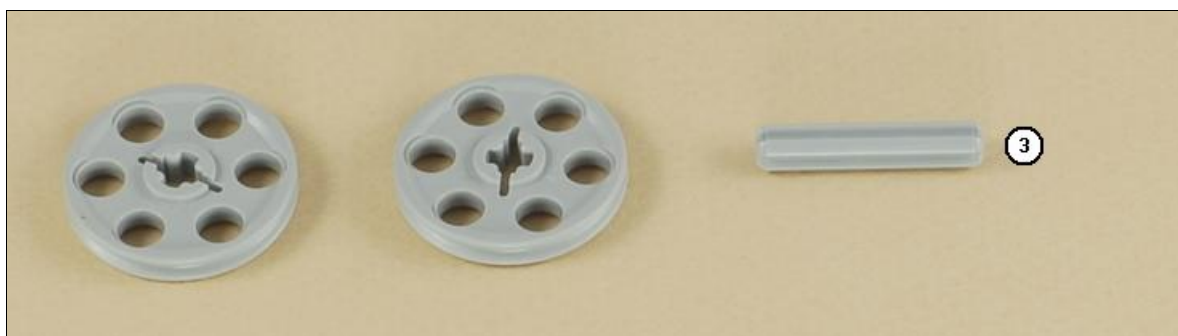
7



8

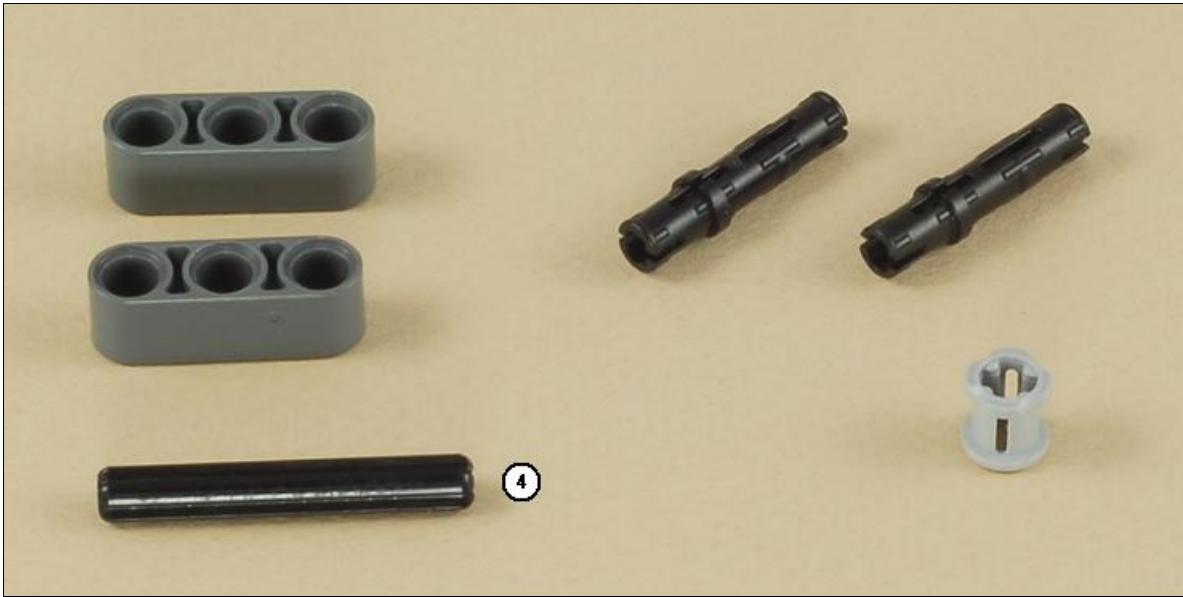


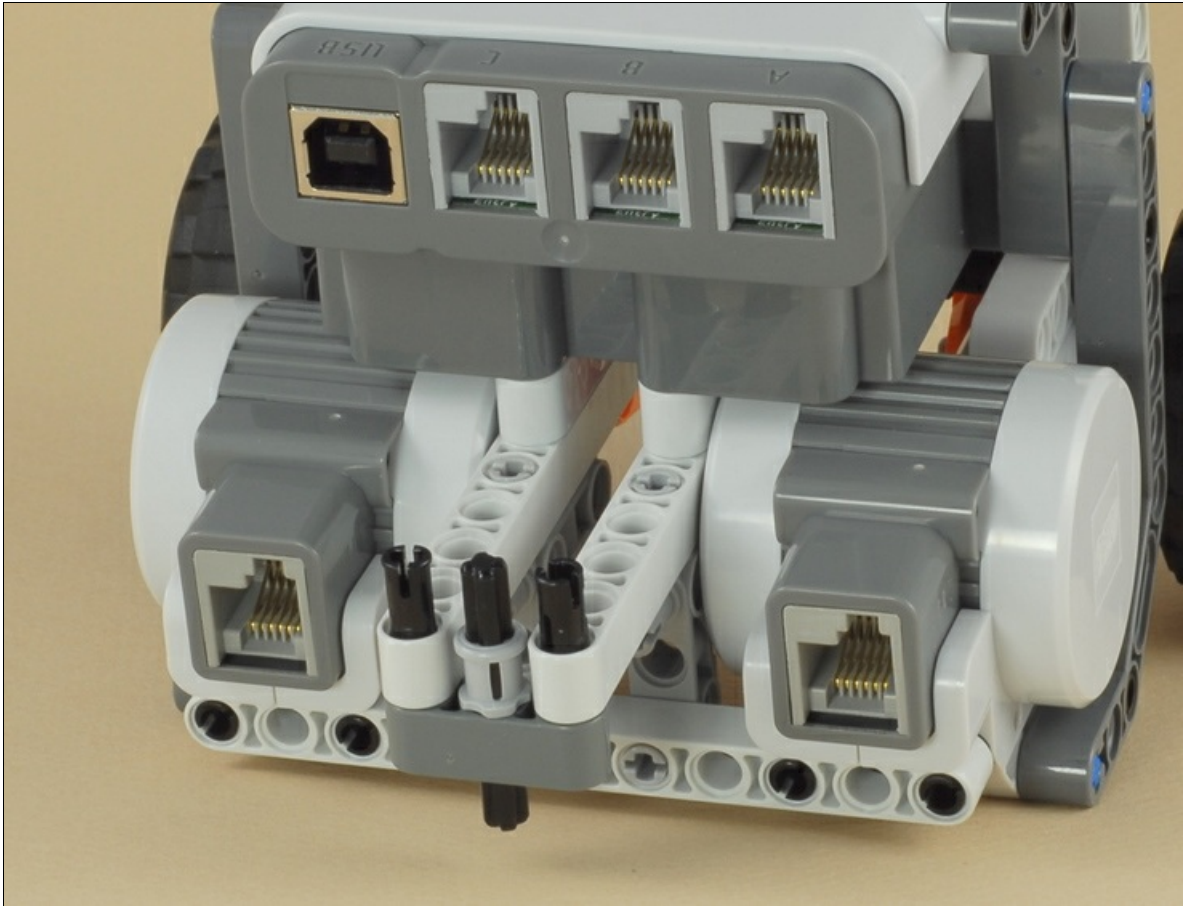
9

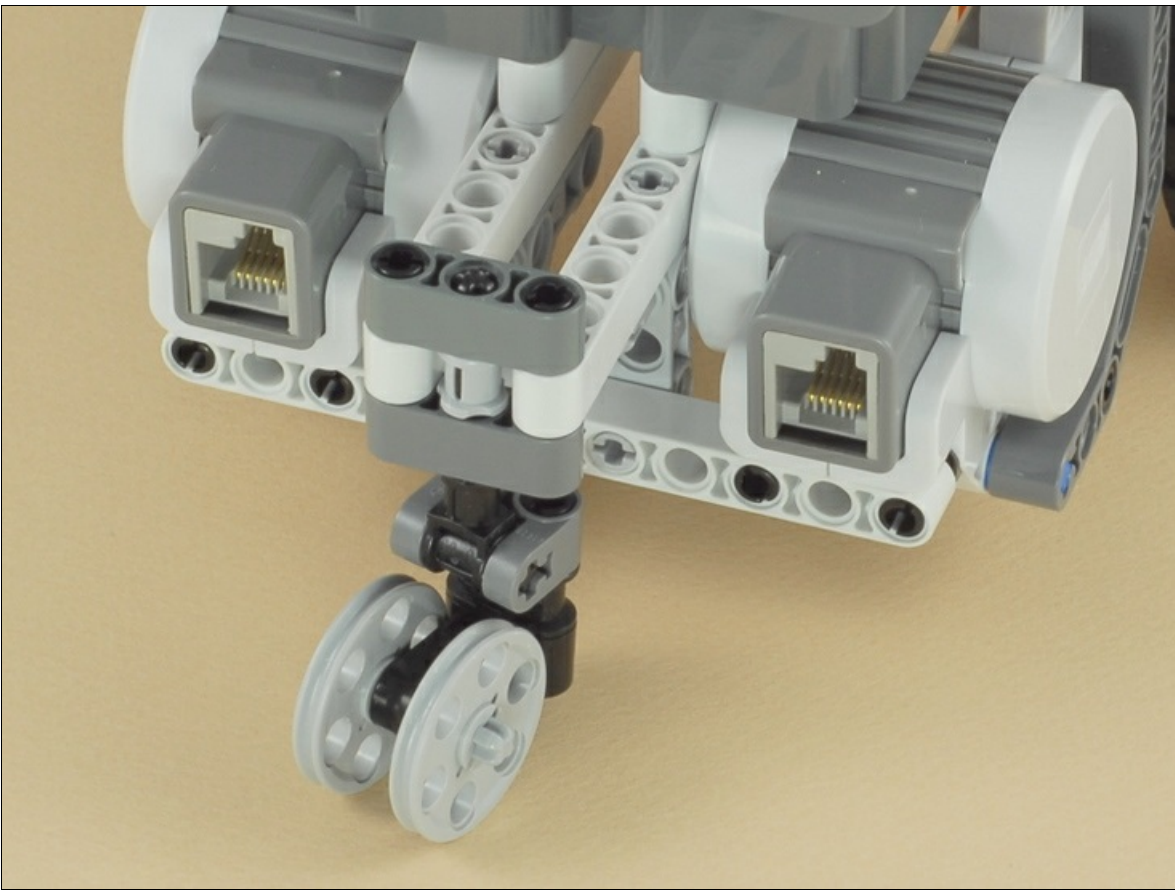




10



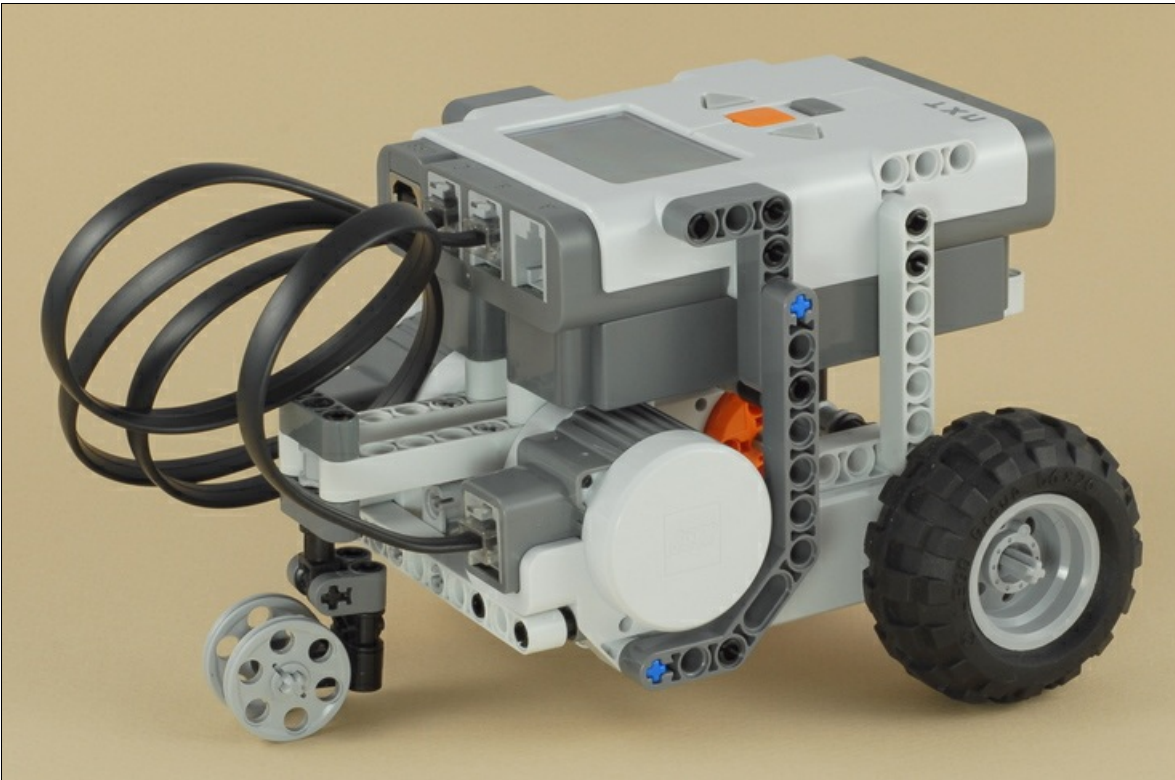




11

Use two medium length wires to connect the two drive motors to ports **B** and **C** on the NXT.

Important: Keep the left wire on the left and the right wire on the right (do not cross the wires).



Building Tip: Weight Balance

When building a robot with a castor wheel, it is important to consider the weight balance of the robot. For good turning, you want to have most of the robot's weight over the drive wheels, if possible. But there must also be enough weight over the castor to keep the robot stable and avoid tipping over. If too much weight is over the castor, the robot may struggle to turn, might get caught up and stall, or the drive wheels might start slipping. However, if too much weight is over the drive wheels, the robot may pop a wheelie if driven with a lot of power with the drive wheels in the back (see the [Dragster](#)).

As determined by the experiment below with a small scale, this Castor Bot robot has a total weight of 600 grams, with 388 grams over the drive wheels. This works out to 65% ($388/600$) of the weight over the drive wheels.



The Castor Bot's total weight is 600 grams



388 grams (65%) is over the drive wheels

Castor Bot Programming


The Castor Bot is a basic two-motor drive robot with sturdy construction that turns easily. You can use it as a starting point for your own projects, and program it however you want. To get started, you could try the "NXT Program" feature of the NXT brick to select some simple movements using the buttons on the NXT brick.

Challenges

- Write some simple programs to make the Castor Bot move, turn, etc. Use "Move" blocks with the motors set to C and B. As built, the castor is on the back of the robot, so motor C is on the left and motor B is on the right, but you can just as easily drive it "backwards" if you want.
- Make some attachments to add to the Castor Bot with sensors, etc. For example, you could add a sound sensor to make the robot alternate between going and stopping whenever it hears a sound.



Explorer

[Building:](#) [Program:](#) Designed for **NXT 1.0** (8527, or 9797 + 9695/9648)

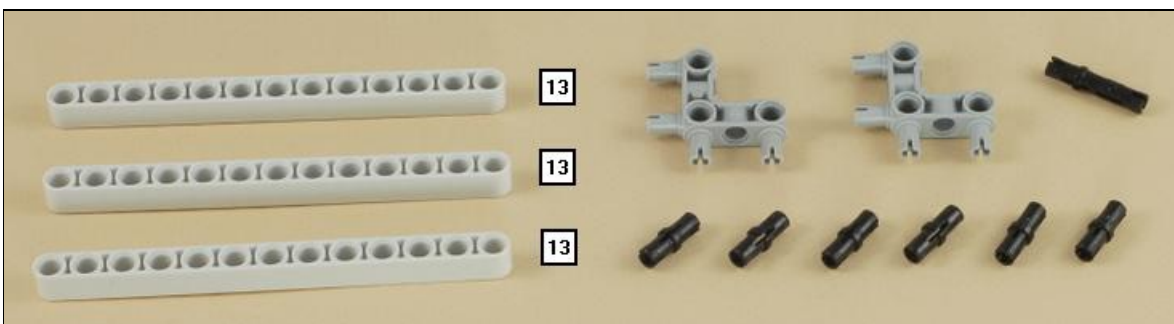
Building Instructions

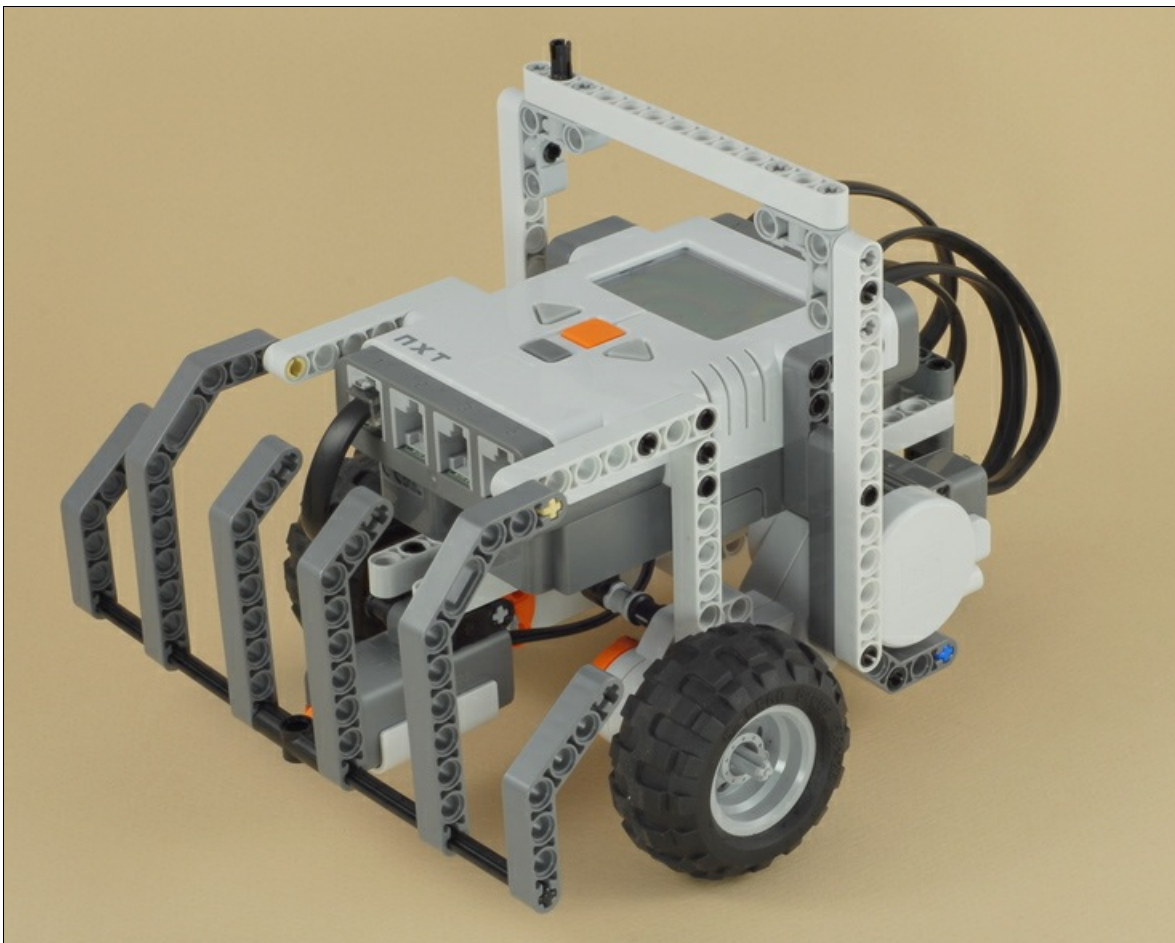
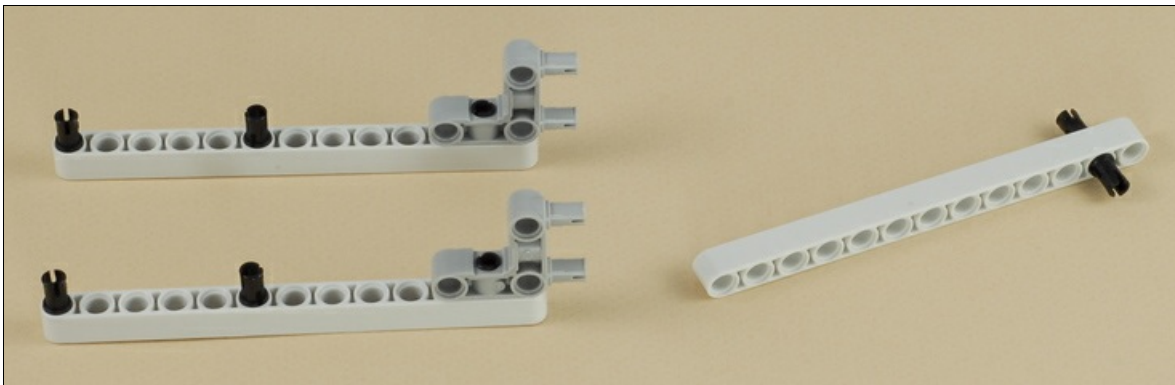
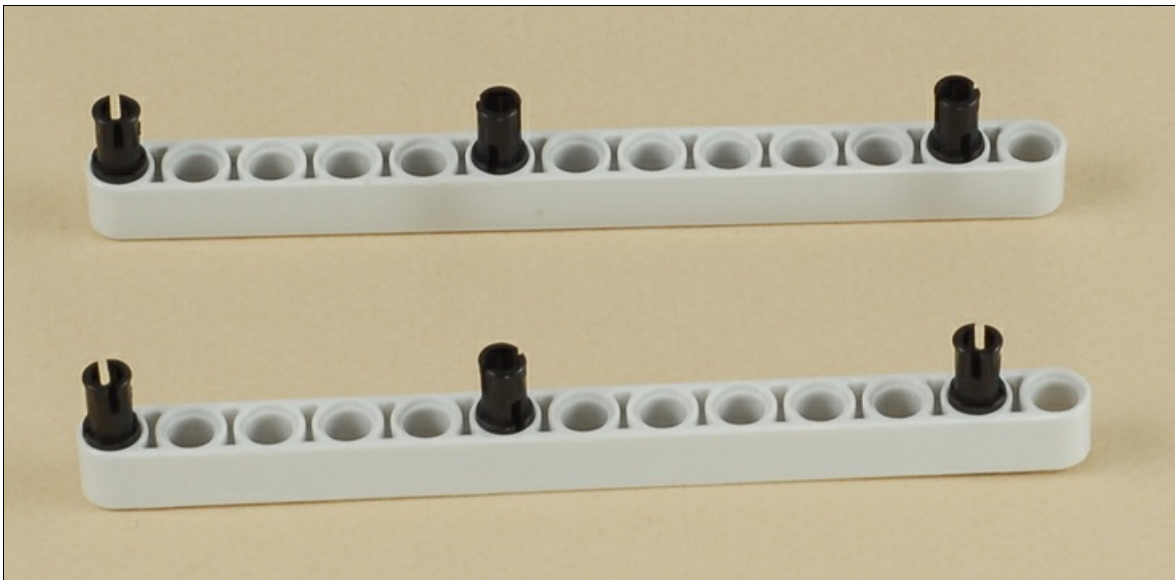
1-13

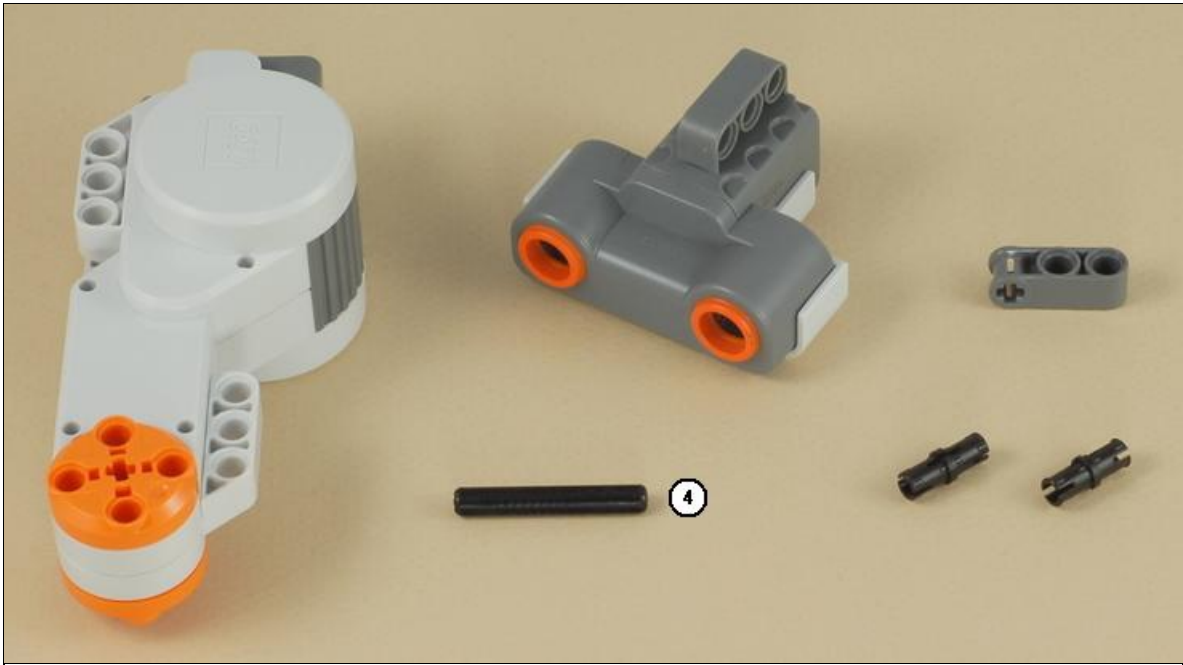
Start by building the [Bumper Car](#). Click the picture for building instructions.

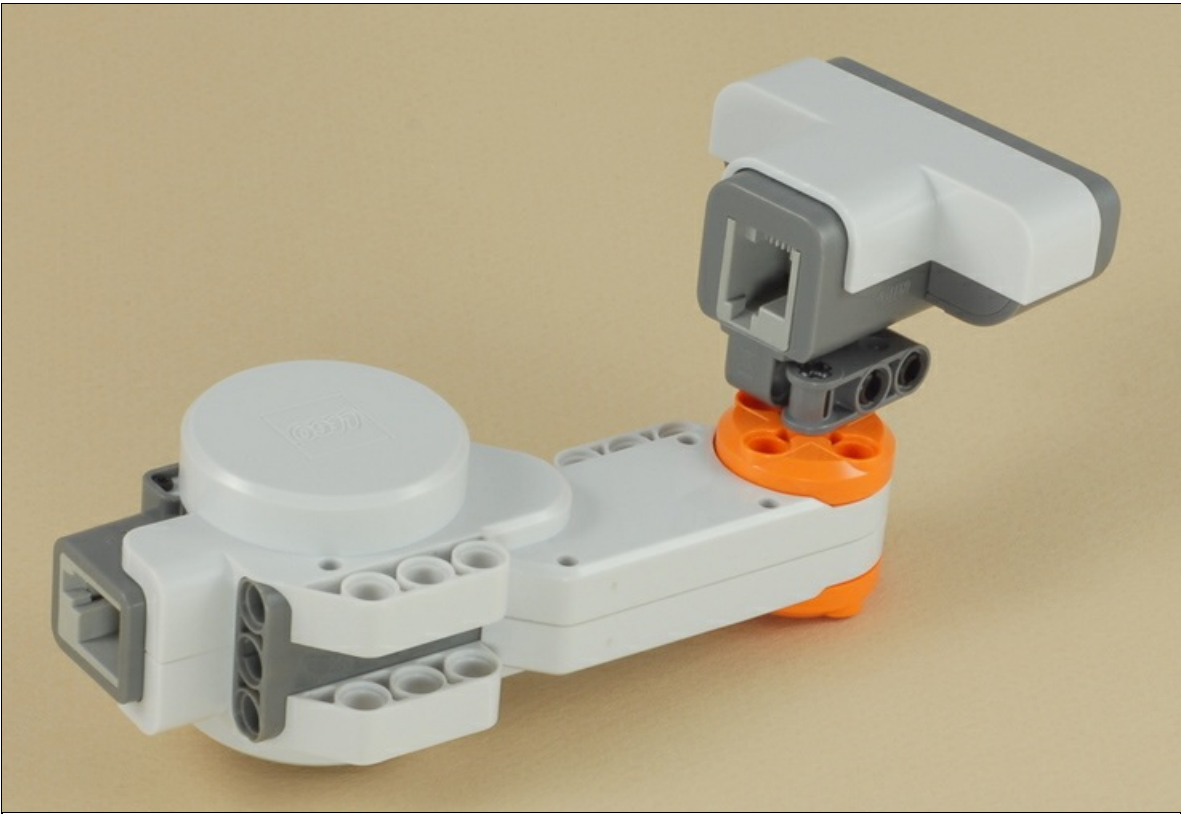
[Building Instructions](#)

14

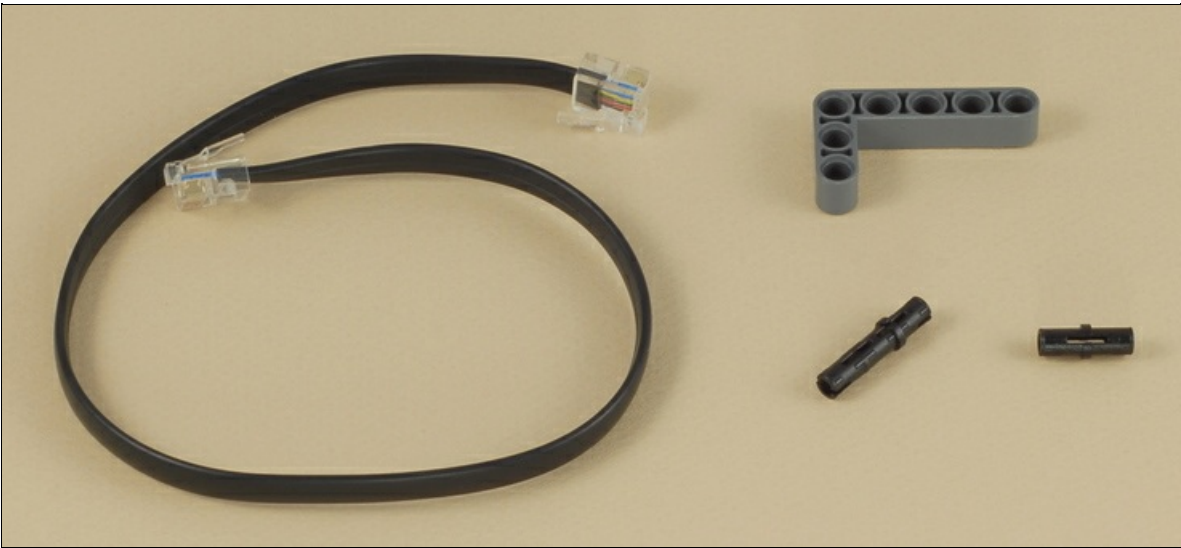


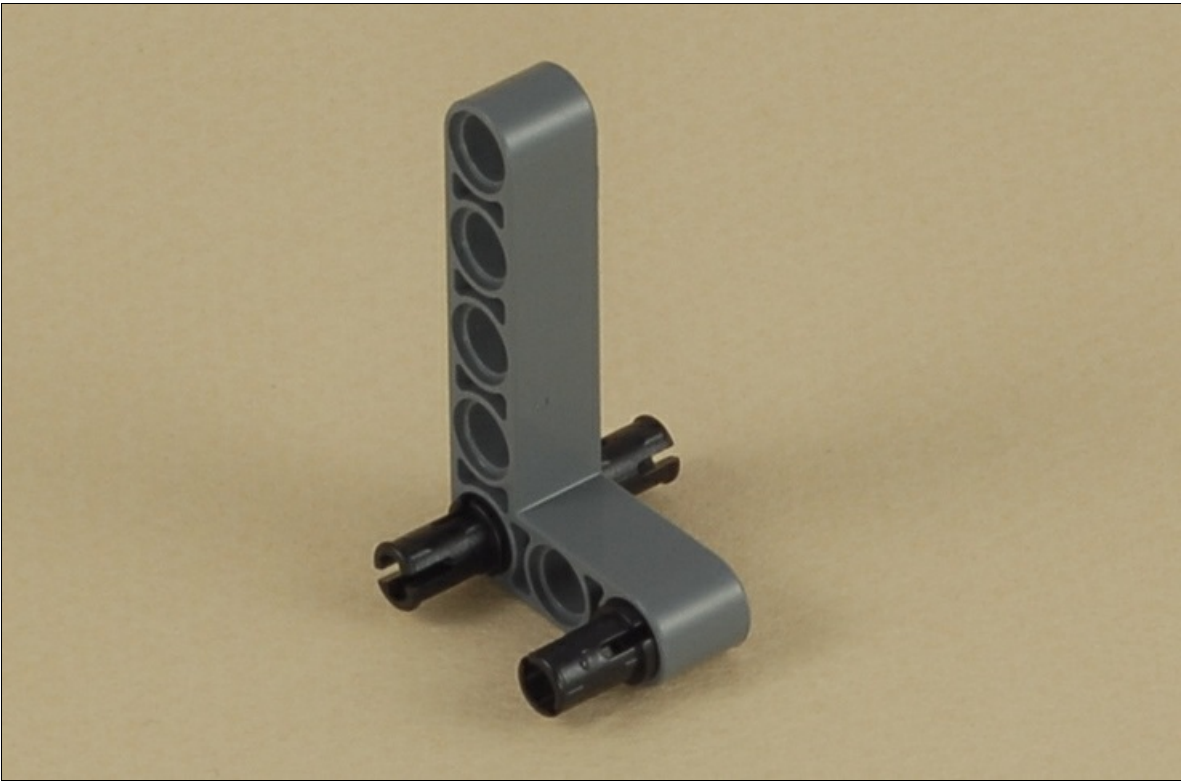






16



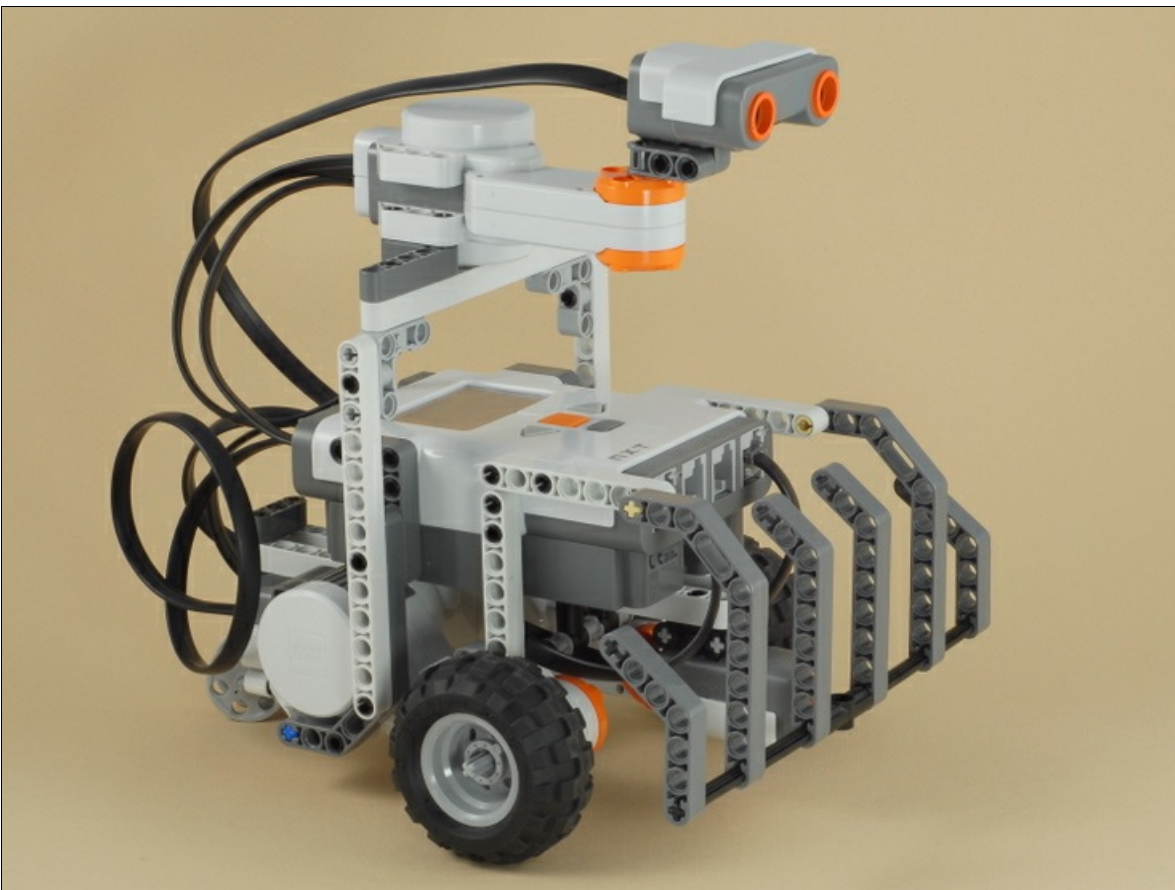


Connect the wire from the "neck" motor to port **A** on the NXT.



17

Use the longest wire to connect the ultrasonic sensor to port **4** on the NXT. Route the wire under the robot, and make sure that that it stays out of the way of the bumper and that there is enough slack so that the head can pivot all the way to the left and right.



Explorer Programming

Use the program [Explorer](#) for the Explorer. This program makes the robot repeat the following steps:

1. Go straight until either the ultrasonic sensor sees something close or the bumper is hit.
2. If the bumper was hit, make a sound and back up a little.
3. Turn the head to look right and then left, and use the ultrasonic sensor to determine which direction appears to have more space, then turn in that direction.

Using the Explorer

Make sure the "eyes" of the ultrasonic sensor are pointed straight ahead before running the [Explorer](#) program, so that robot is not confused about which way is which.

Challenges

- Set your Explorer loose and see if you can predict which way it will turn each time it approaches an obstacle, and try to guess where it will end up. Will it get stuck anywhere?
- For an advanced programming challenge, try modifying the [Explorer](#) program to change what happens when the Explorer "sees" an obstacle or bumps into something. Can you come up with a better navigation strategy that is more interesting or less likely to get stuck?



Get nxtprograms.com on CD!
[Click here for info](#)

[Home](#) [Projects](#) [Help](#) [Contacts](#)