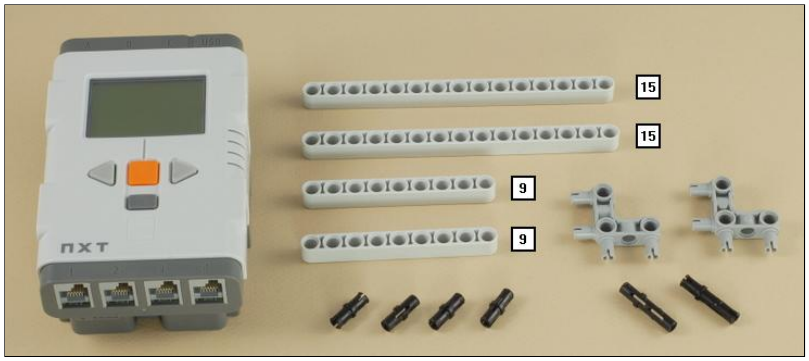
Castor Bot

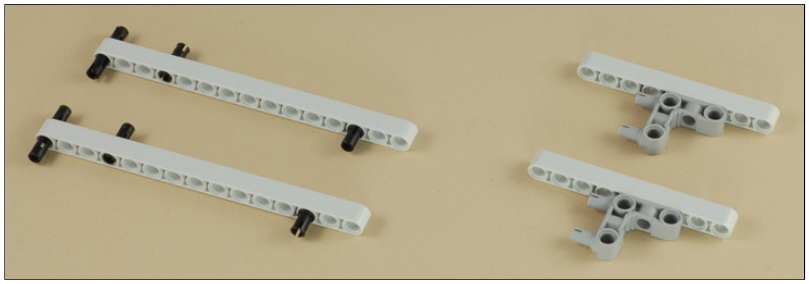
Building:
Program: (none)

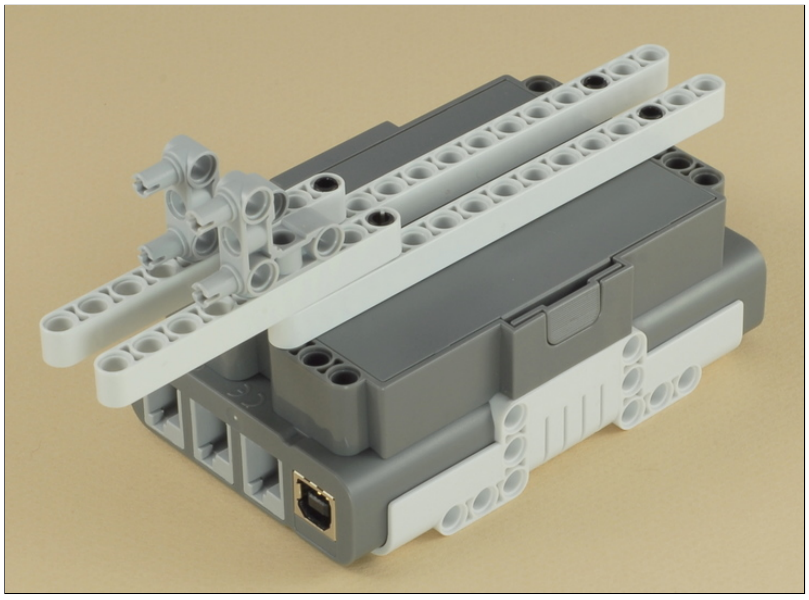
Designed for **NXT 1.0** (8527, or 9797 + 9695/9648)

Building Instructions

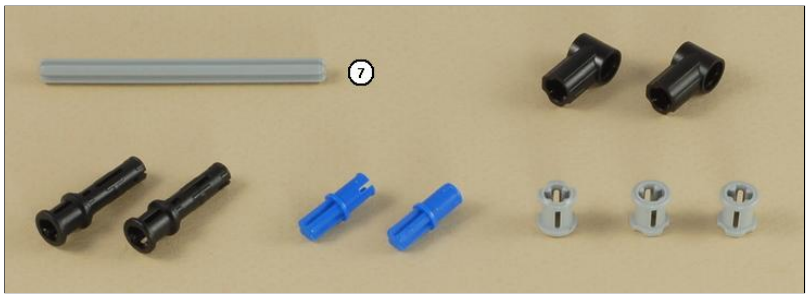
1

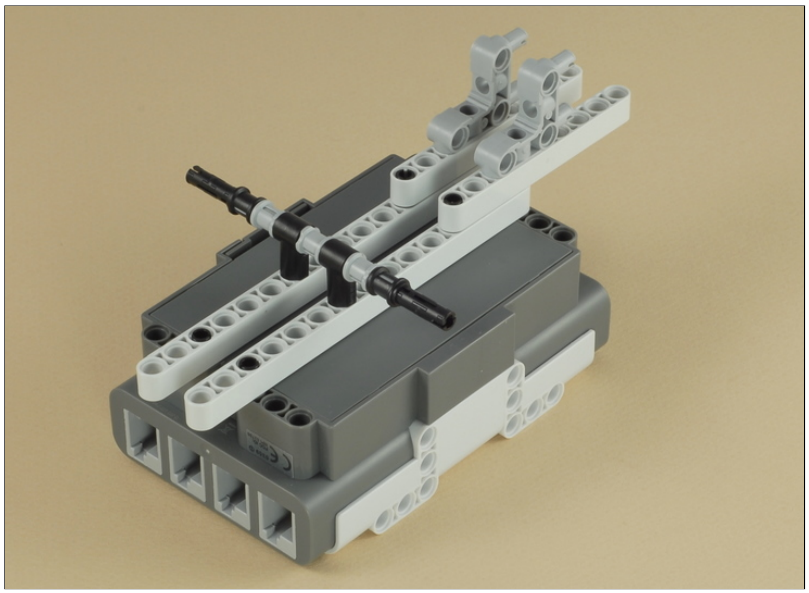




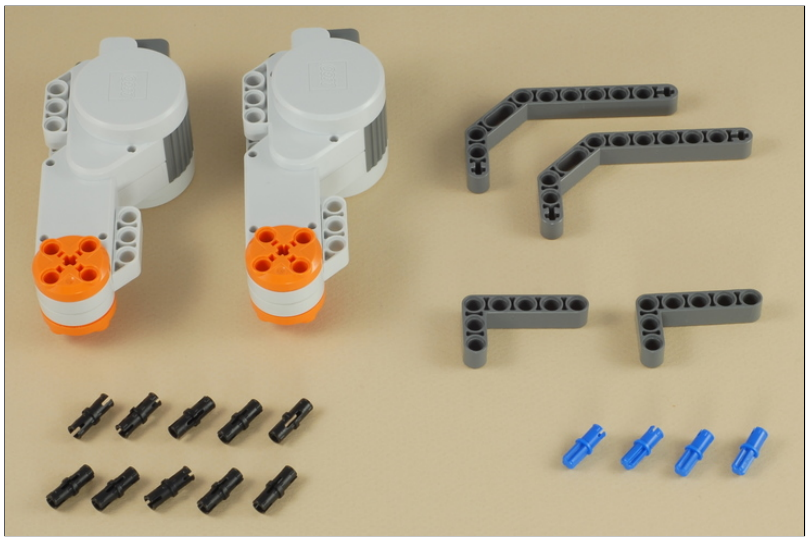


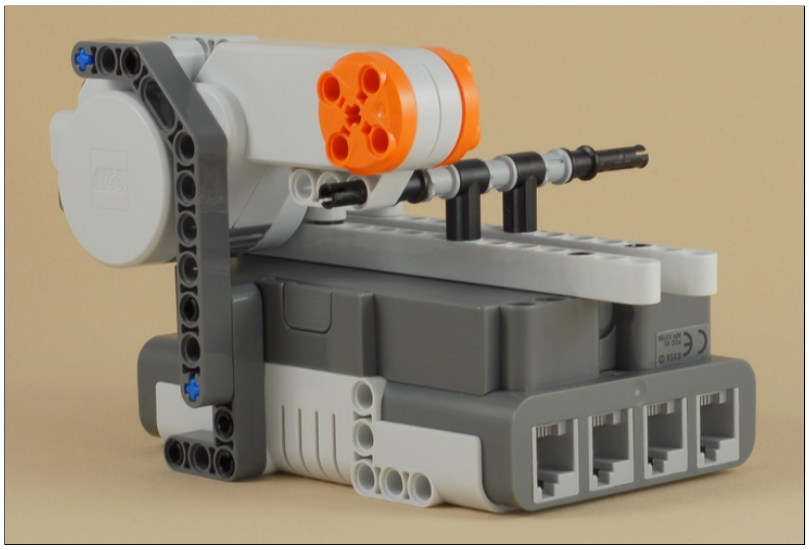
2

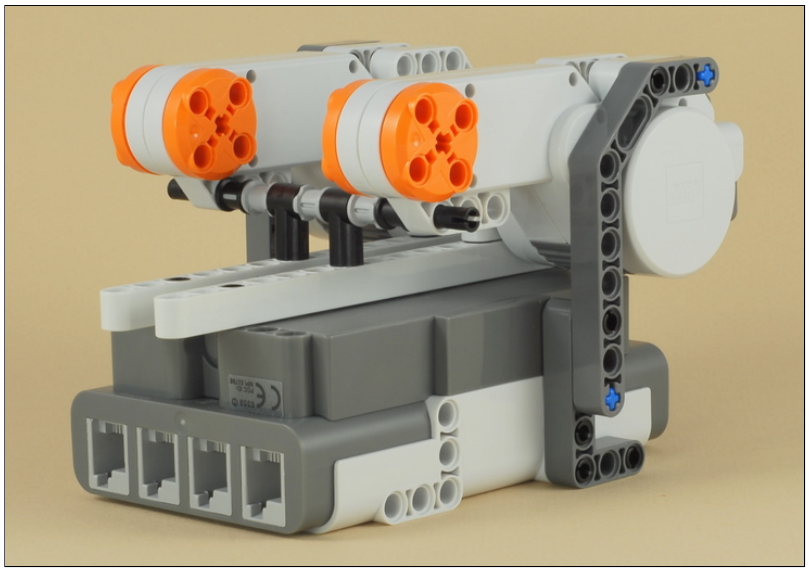




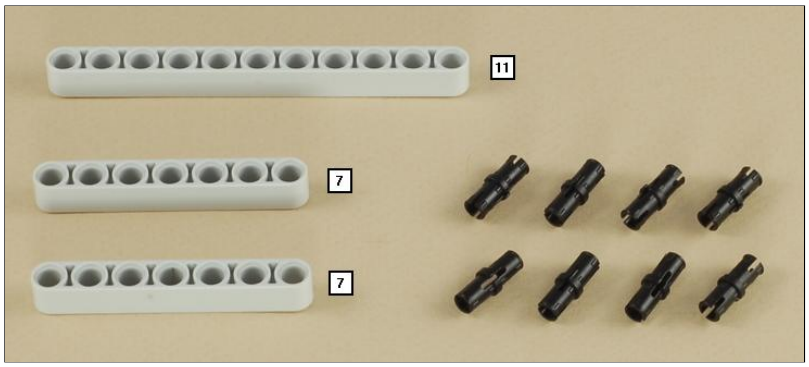
3

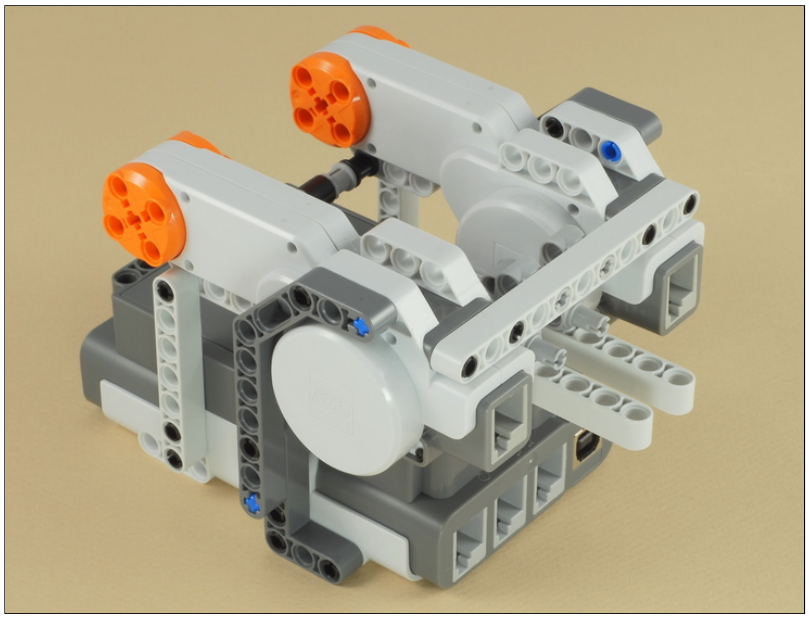




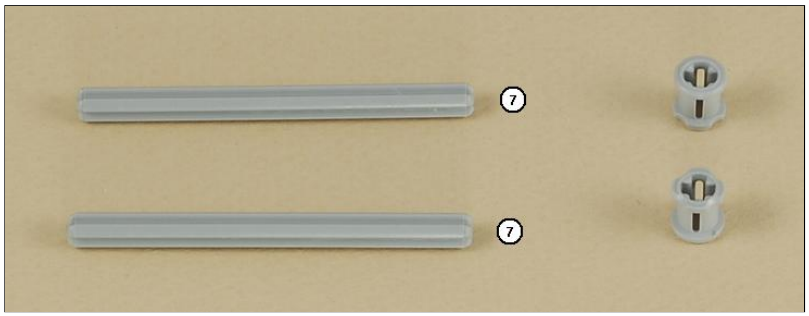


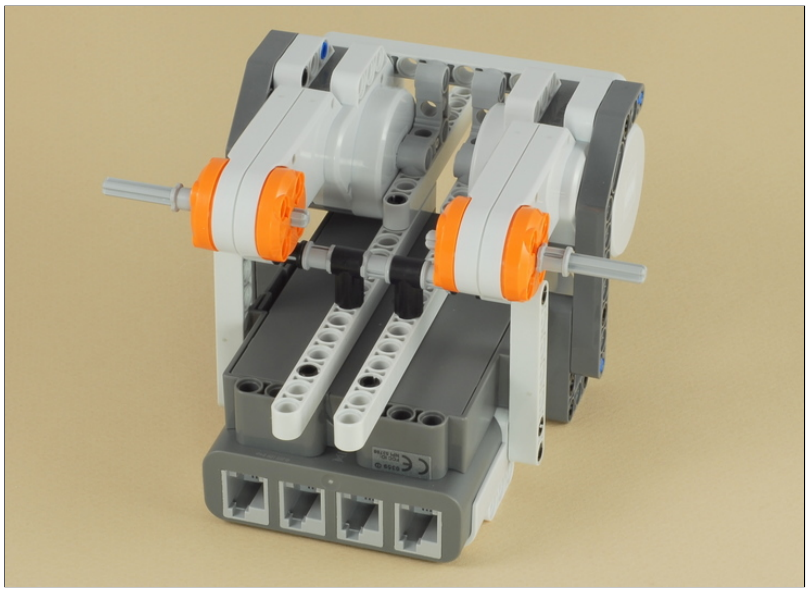
4



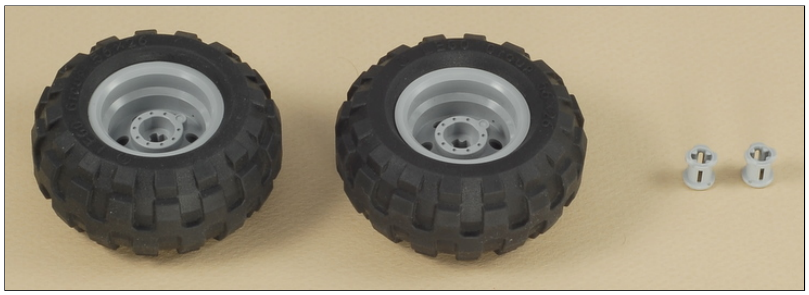


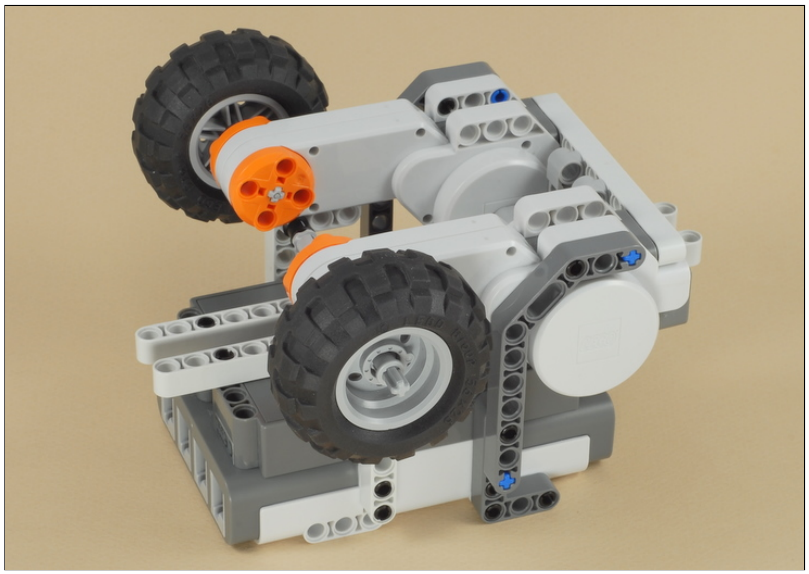
5





6

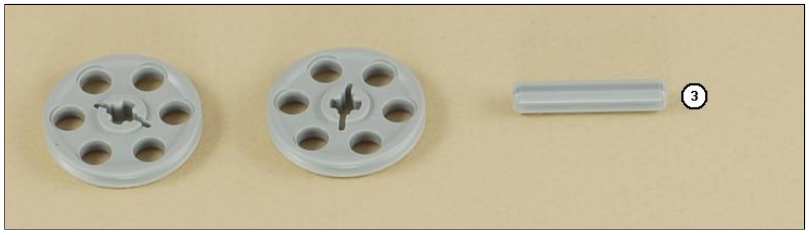


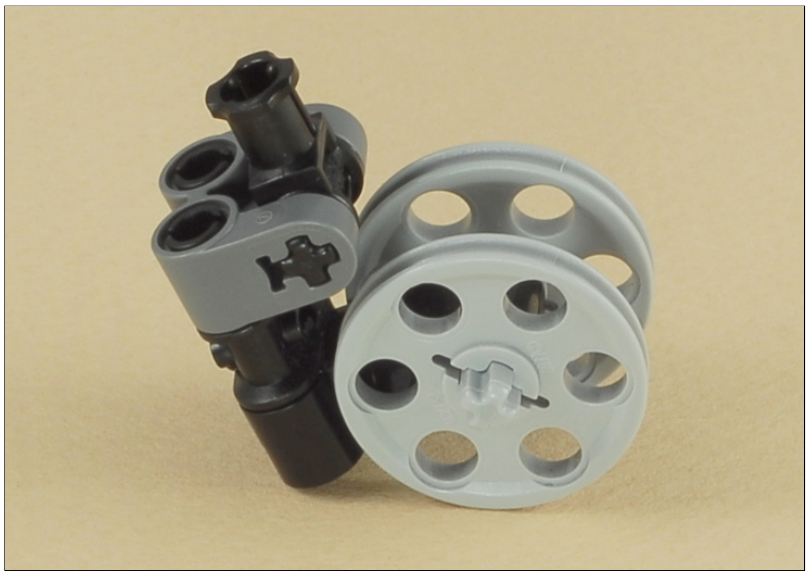


7

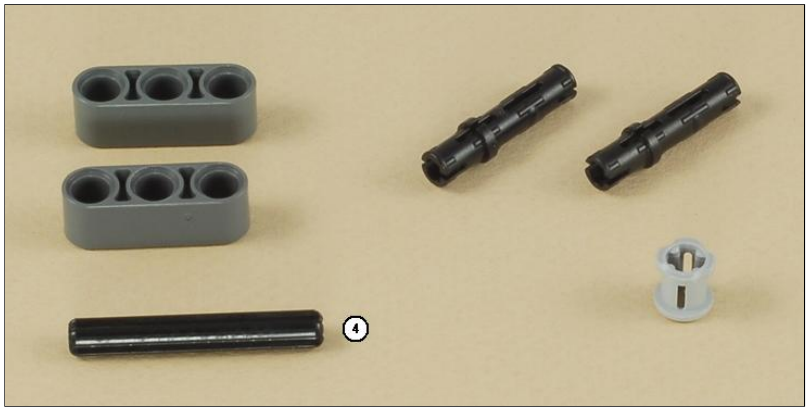
8

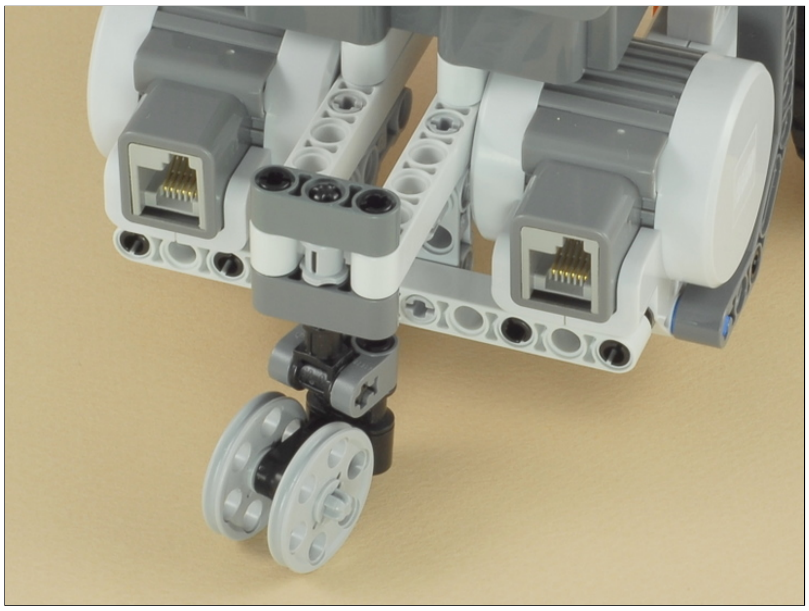
9





10

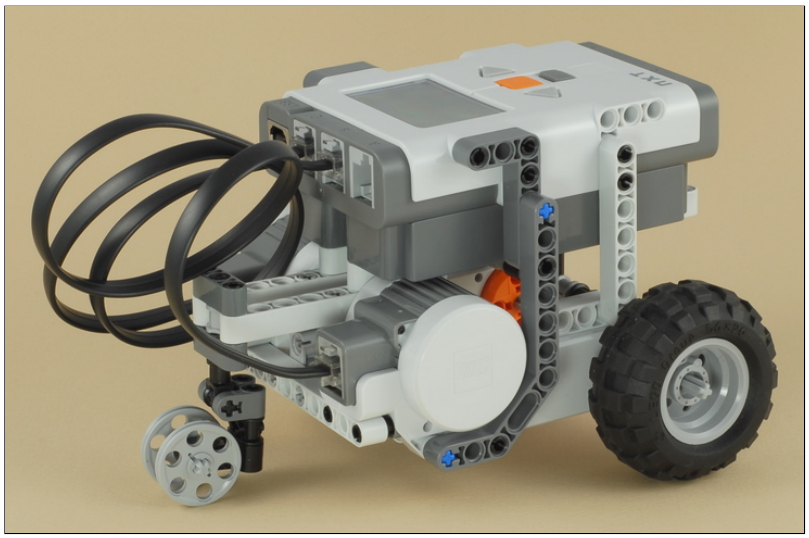




11

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.

Mini Sumo Bot

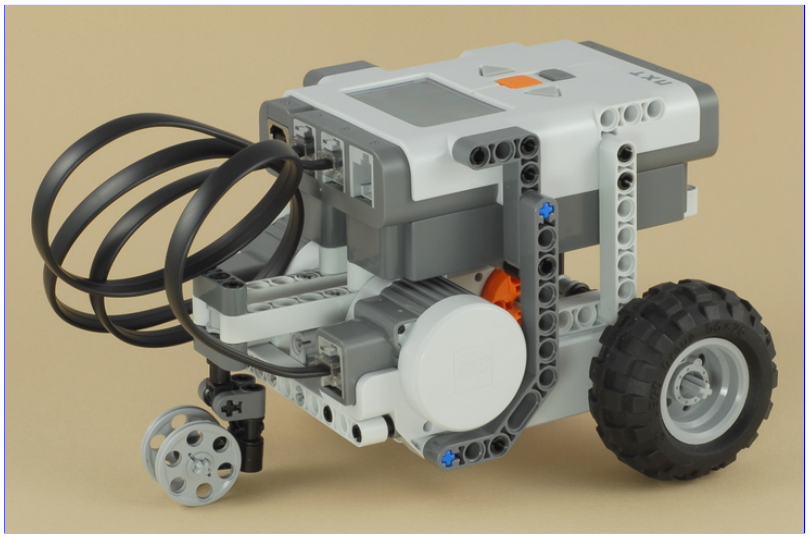
Building:
Program:

Designed for **NXT 1.0** (8527, or 9797 + 9695/9648)

Building Instructions

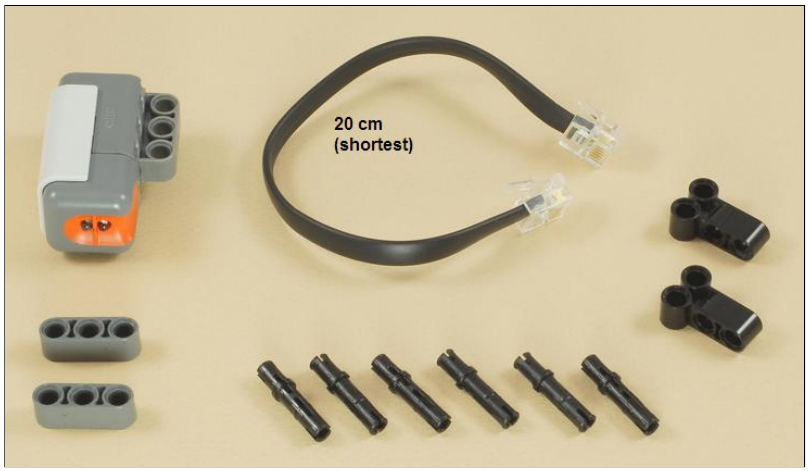
1-11

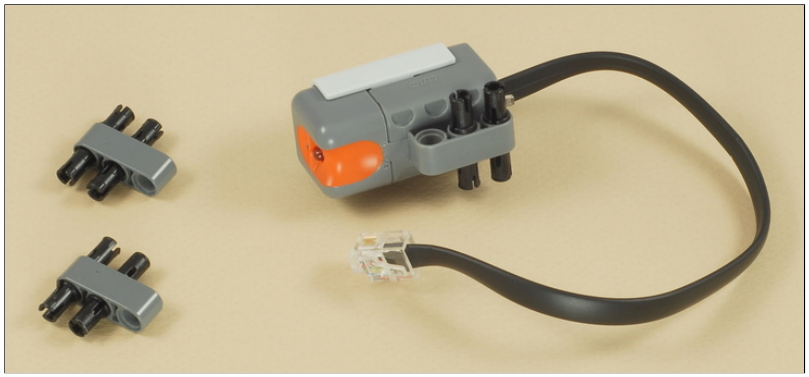
Start by building the Castor Bot. Click the picture for building instructions.

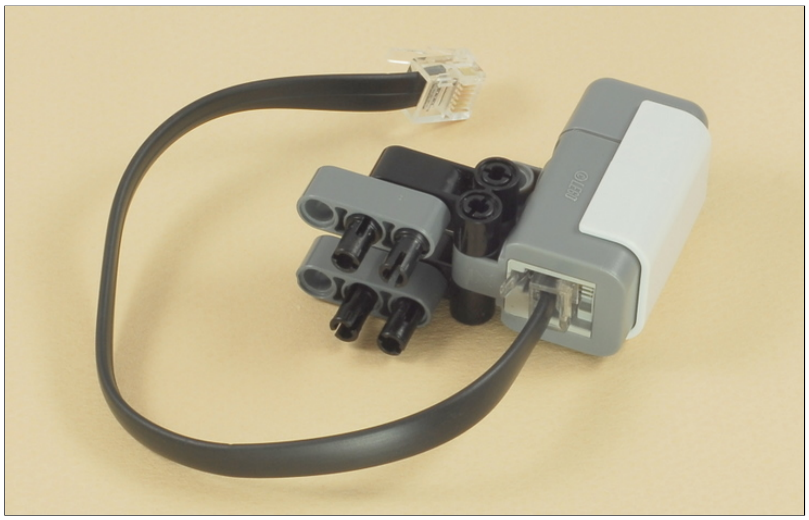


Building Instructions

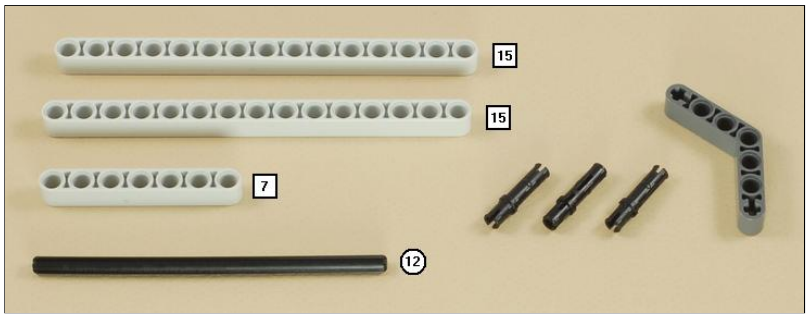
12

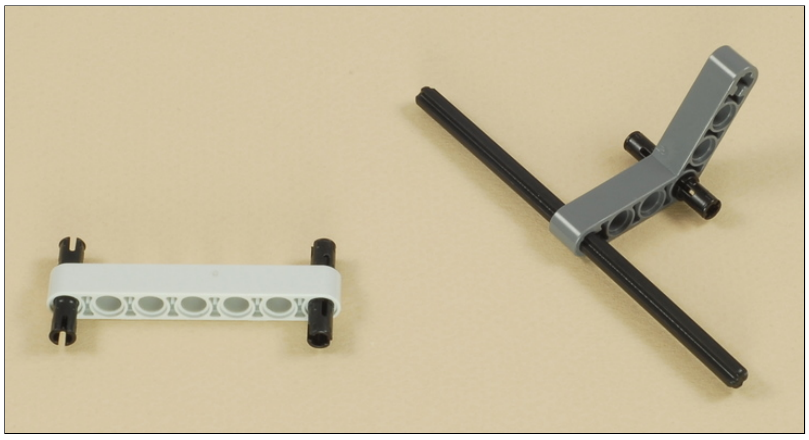


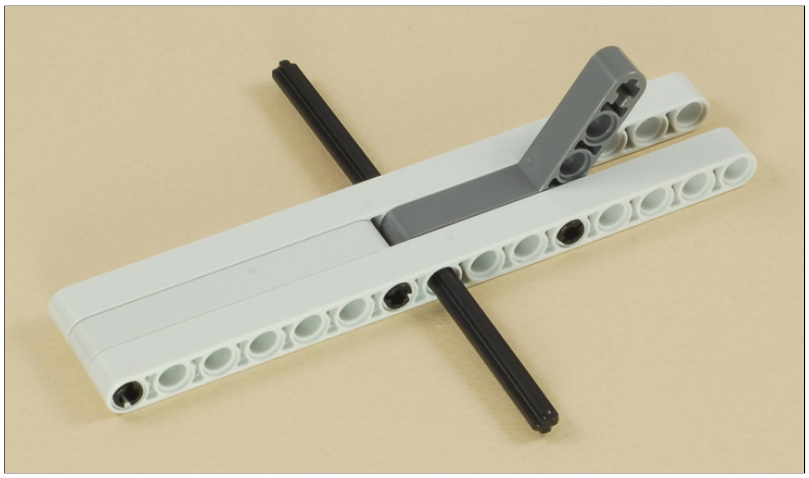




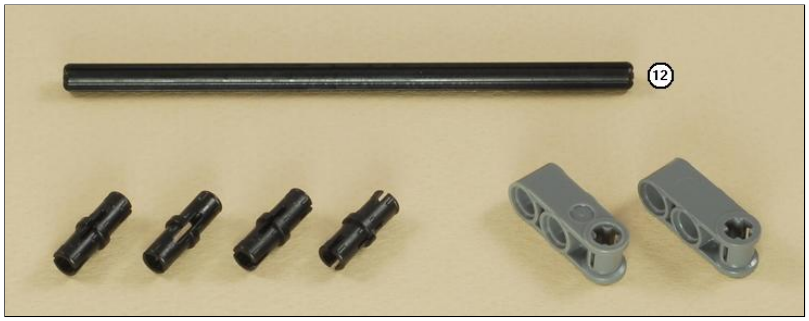
13

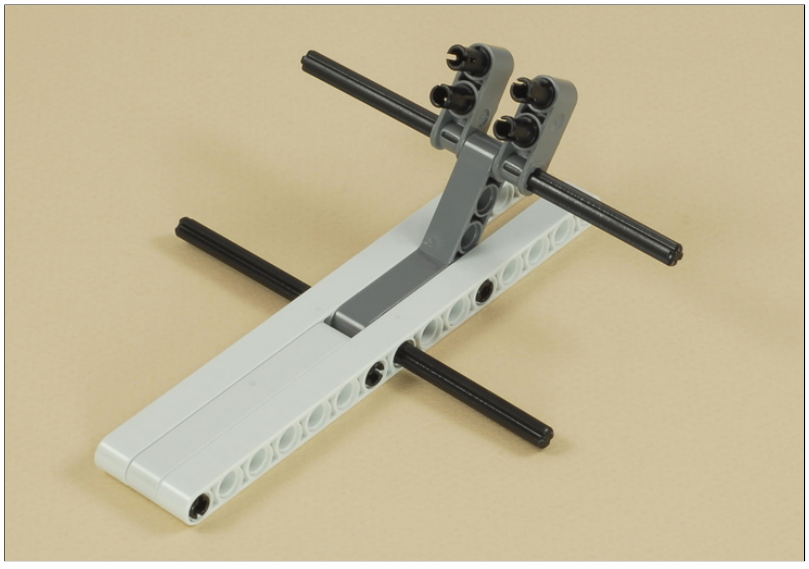




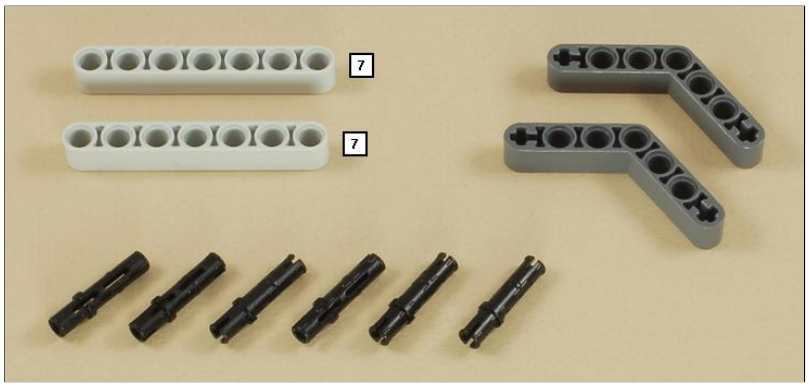


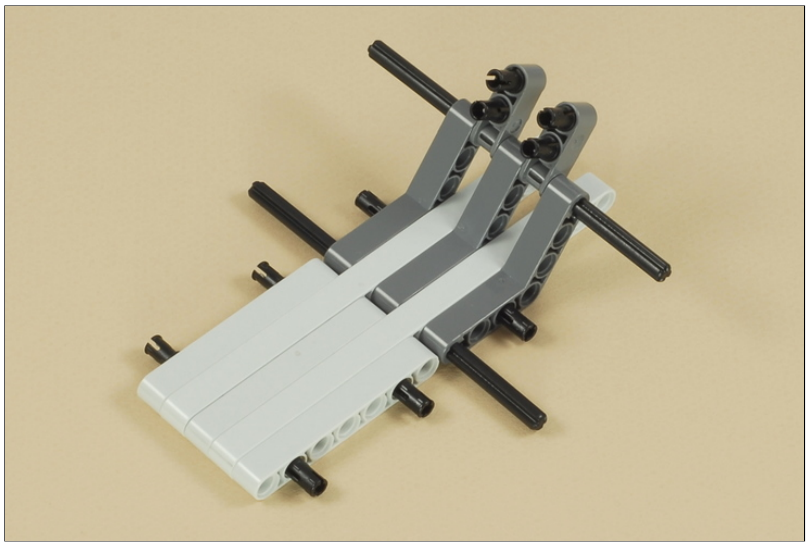
14



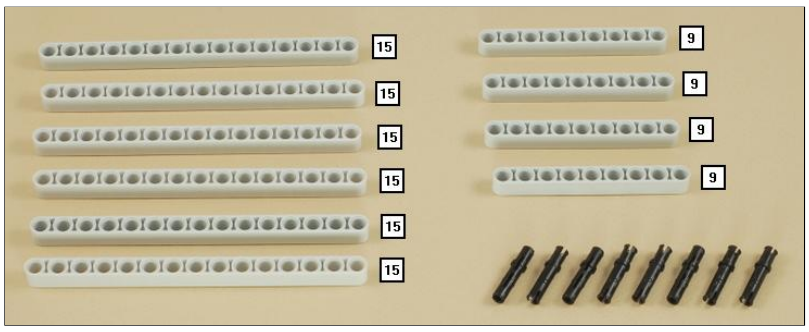


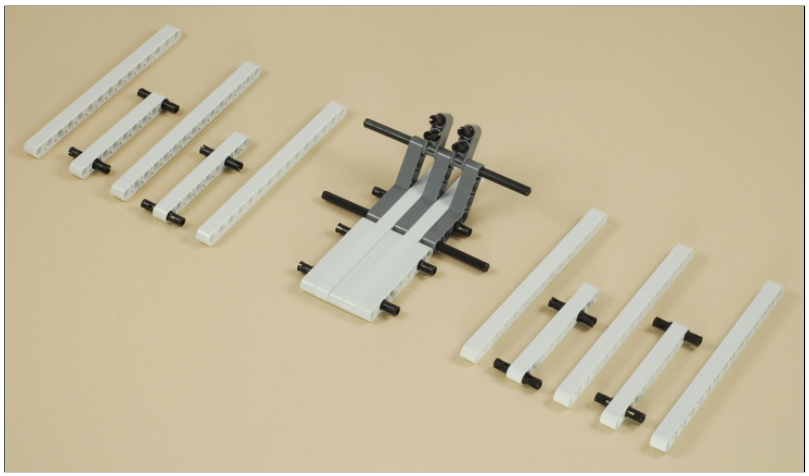
15

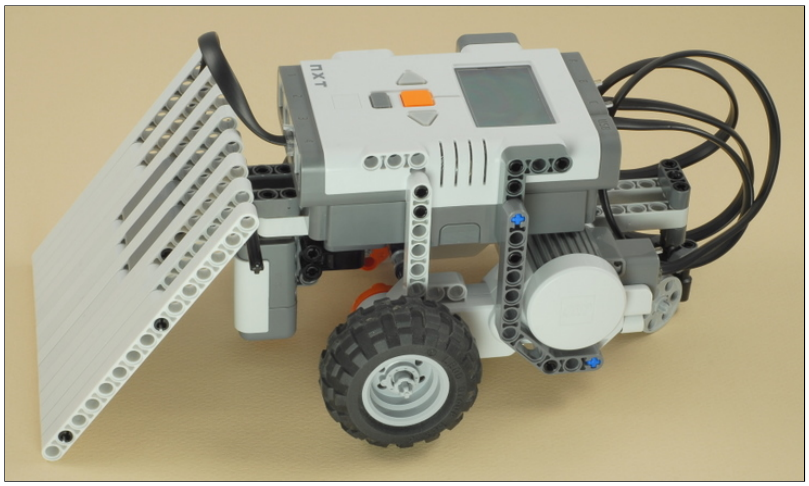




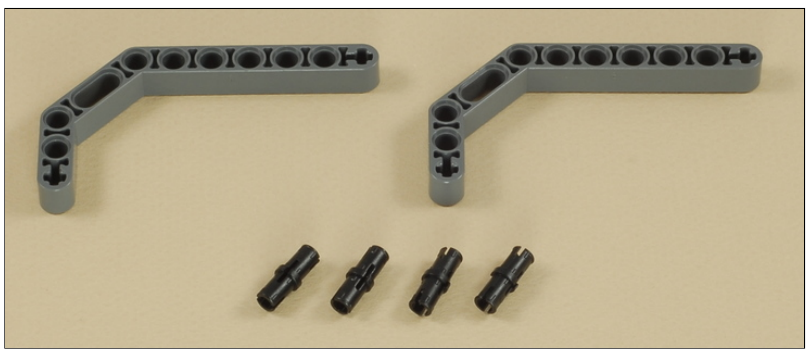
16

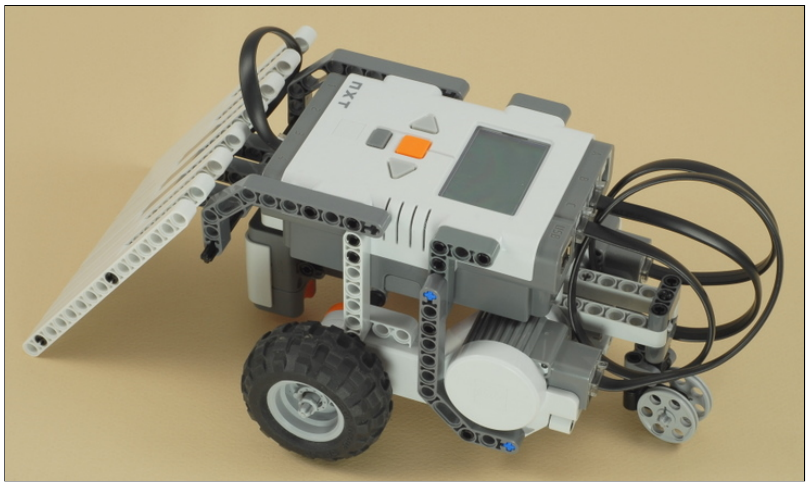






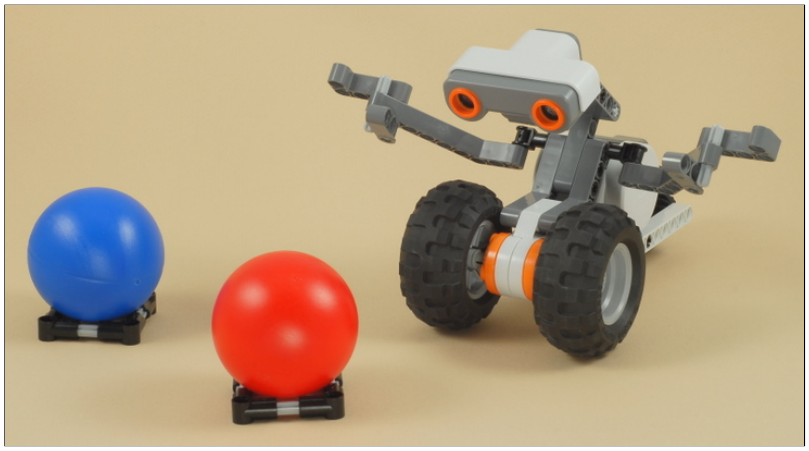
17





18

Design your own dummy robots or other objects for your Sumo bot to try to push outside the ring.



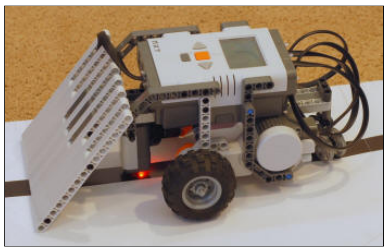
Mini Sumo Bot Programming

Before using your Sumo bot, you must calibrate the light sensor to the colors on the actual surface that you will be using (with the sensor attached to the robot as it will be used). The program Calibrate_L3 can be used to calibrate a light sensor attached to port 3. The NXT will remember the sensor calibration between program runs, so you can run the Calibrate_L3 program once, then run your Sumo program as many times as you want after that, as long as the lighting conditions don't change. This program can also be used to calibrate a light sensor on port 3 any time you need it for other robots.

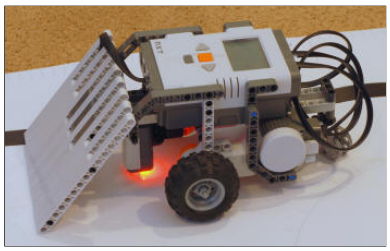
The Mini_Sumo program is a very simple driving strategy for a Sumo robot. It simply makes the robot go straight forward until the light sensor sees something dark on the surface (might be the ring border), then it backs up a little, turns right to head back into the ring, then repeats (going straight and looking for the ring border again). This will make the robot wander blindly around the ring, pushing whatever is in front of it, and hopefully whatever it finds will get pushed outside before the robot accidentally drives outside the ring on its own.

Using the Mini Sumo Bot

- Be sure to calibrate your light sensor before starting a series of runs, or whenever the lighting conditions or robot design changes. The Calibrate_L3 program will lead you through the steps.



Step 1 of the light sensor calibration wants the sensor to be over the black border line.



Step 2 of the light sensor calibration wants the sensor to be over the plain white surface.

- Put your Sumo bot inside the ring (make sure the light sensor is over the white surface inside the ring), place any dummy robots or other objects inside the ring, then run the Mini_Sumo program and see if your robot can push all the objects outside the ring.
- If you are battling head-to-head against another Sumo bot, then the robots should start facing sideways or back to back (not straight at each other), so that the match doesn't just become a simple straight ahead pushing match.

Making your own Sumo Ring

The test pad that comes with the NXT set is not a plain white surface with a simple border. There are other markings inside the ring, and the border is partially interrupted by the "Start" area (What do you think this will do to your robot's strategy?) Making your own ring will produce more consistent results and avoid damaging your test pad. If you want to make your own ring, I suggest:

- For a surface you could use a light colored hard floor, or a flat board-like surface that is light colored, painted white, or covered with white paper.
- For the borders, black electrical tape works well and comes off easily if you are on a floor. Although a traditional robot Sumo ring is round, making a round border with tape is difficult, so I would suggest an octagon (like a Stop sign) shape .
- Don't make the ring too large or the robots will have trouble finding each other.
- For off-season FIRST LEGO League teams, turning the competition mat over and taping out an octagon area with electrical tape works great.

Challenges

- Using the robot and program given here, and a small dummy robot such as that shown above, it should be easy for the NXT robot to push the dummy robot outside of the ring, because the dummy robot is much lighter and not powered. However, if you try several runs starting in different positions, you may get a round where the dummy wins because your Sumo bot accidentally drove outside the ring on its own, perhaps due to contact with the dummy robot, or perhaps totally on its own. Can you figure out why and improve the robot or the program to try to prevent this?
- There are many ways to design a Sumo robot, and if you go head-to-head with another robot, then your success will also depend a lot on the design and strategy of the other robot. If you have a friend with an NXT set (or two sets of your own), try some real head-to-head matches. After seeing the results of a few runs, both sides should try improving their robots or programs. You can go on this way as long as you want in an "arms race" of increasingly better and smarter robots.
- Using a light sensor to detect the ring border is a typical starting point, but more sophisticated Sumo robots might try using the touch sensor and/or the ultrasonic sensor to try to detect the other robot. Try adding other sensors and writing a new program that uses them. For examples of how to use the other sensors, take a look at the Bumper Car and Ball Hunter projects. Programming a robot that uses multiple sensors at the same time gets much trickier, though. For an example of using two sensors at the same time, see the Explorer project.

Warning: If both robots are using the ultrasonic sensor, they might occasionally interfere and confuse each other.