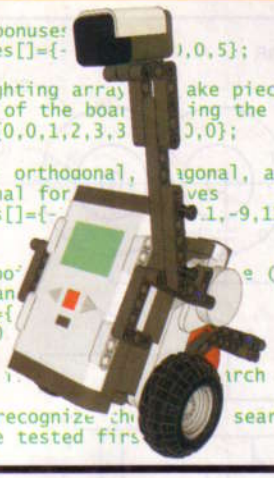
```
// castling bonuses
B8 castleRates[]={-
                  ),0,5};

//center weighting array
//the center of the boa
B8 center[]={0,0,1,2,3,3
                  ),0,0};

//directions: orthogonal,
from orthogonal for
B8 directions[]={-
                  1,-9,11,9,10,-10,1,-
1};

//direction po
bishop rook an
B8 dirFrom[]={
B8 dirTo[]={0

//Good moves
this array
//so we can recognize
sure they are tested firs
```



A bi-monthly column for kids!

LESSONS FROM THE LABORATORY

The Runt and the Compass

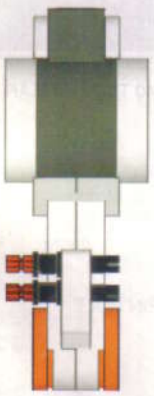
by James Isom

Spring is a time of renewal – flowers flowering, birds singing, new robot chassis glistening in the sun ... well, maybe for some of us. So, in this light, I present to you without further flourish a fresh robot design for spring – Runt. I developed Runt in my never-ending quest for a chassis that is sturdy and quick to build. It obviously takes some of its strengths from JennToo (see Lesson's from the Lab – October '06), but unlike JennToo, it can be built from a single LEGO Education NXT Base Set which I know many of you have, know, and love.

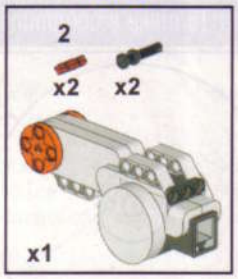
Towering above Runt is the new compass sensor from HiTechnic (**www.hitechnic.com**). This little guy senses direction by measuring fluctuations in the Earth's magnetic field. After we build Runt, I'll show you how to program a popular robot trick using the new Compass Block for the NXT-G software.

BUILDING RUNT

STEP 1:

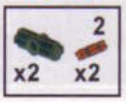


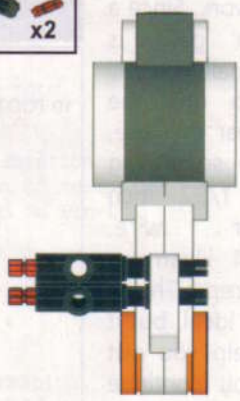
Parts:



STEP 2:

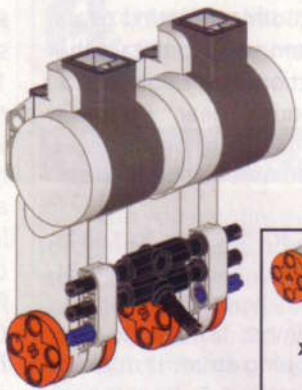
Parts:



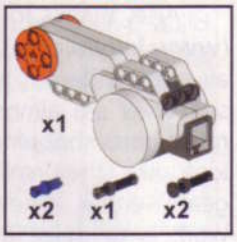


STEP 3:

Tilt the bottom connector as shown in the picture. It will hold the long pin that keeps the ball from falling out.



Parts:

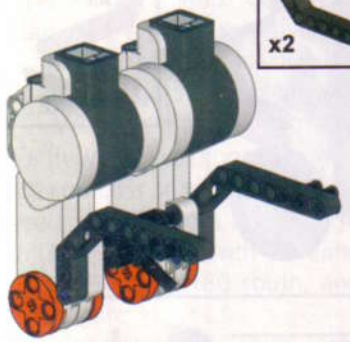


STEP 4:

Parts:

x4

x2

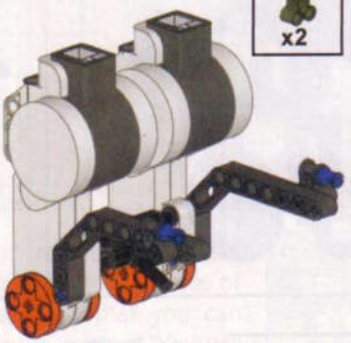


STEP 5:

Parts:

x2

x2

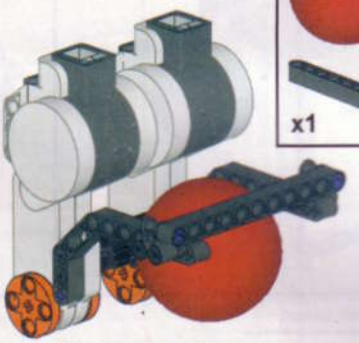


STEP 6:

Parts:

x1

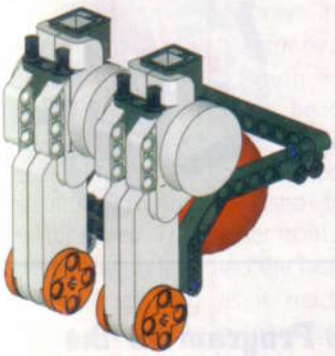
x1



STEP 7:

Parts:

x4



STEP 8:

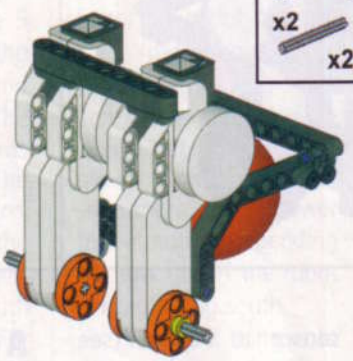
Parts:

5

x1

x2

x2

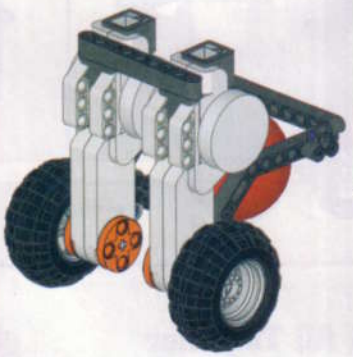


STEP 9:

Parts:

x2

x2



STEP 10:

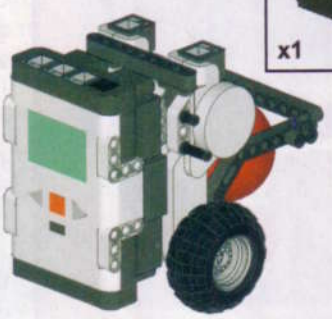
Parts:

x2 x2

STEP 11:

Parts:

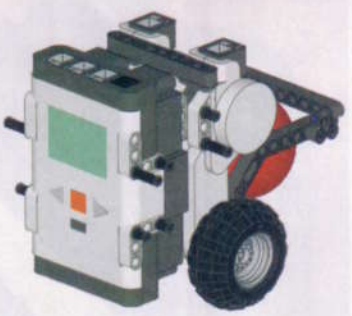
x1



STEP 12:

Parts:

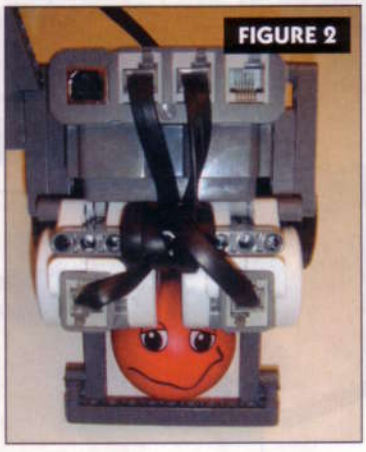
x2 x6



Wiring it All Together

Face the front of the NXT (ball in the back). Wire the left motor to Port B and the right motor to Port C. Connect the compass sensor to Port 2 (see Figure 1).

Use the nine-hole beam connecting the two motors together to wrap your extra wire around (see Figure 2).



A Little Program for the HiTechnic Compass Sensor

This little south-pointing trick has been around awhile but I hadn't seen a version that used the new HiTechnic Compass Sensor Block, so I thought I would give it a whirl here. The program is a simple demonstration of how the compass sensor works. When the program starts, the robot will turn to face whichever direction it's programmed to face. The farther away the robot is from the desired direction, the faster it will turn to get back. Special thanks to Brian Davis for helping me work out a nice, efficient way to do this.

To begin, download and install the Compass Sensor Block. If you don't know how to do this, visit the

resources section at **LEGOedwest.com** for a quick tutorial.

Once installed, drag the following programming blocks from the Complete Palette into a loop: Compass Sensor, Math (x2), and Move (see Figure 3).

Select the Compass Block to bring up the configuration panel.

We want the robot to turn to the south when the program runs. Setting the compass sensor action to Absolute Reading will return a heading from 0 to 359 degrees, with 0 representing north, 90 east, 180 south, and so on (see Figure 4).

Connect the wire from the Absolute Heading plug into the B plug of the first Math block. Run a second wire from the YES/NO Logic plug to the Motor Block's Direction plug. This will control which direction the robot turns when trying to return to south.

The Math Block shown in Figure 5 gives us the margin of error by taking the desired direction "A" (south = 180) and subtracting the current heading. The number the block spits out represents how far from south the robot is. The resulting number is used to control motor speed. The farther from south, the higher the number and the faster it goes, gradually slowing as it reaches south where it comes to a complete stop. For example, 180 (the desired heading) - 180 (the actual heading) = 0 (power).

Connect a wire from the Result plug of this block to the A plug of the second Math block.

The second Math Block shown in Figure 6 isn't really necessary but I thought it was a fun way to tune the robot's reaction time so it's been left in. Try dividing by 2 or more to see what happens to the robot's reaction time.

Connect a wire from this block's Result plug into the Power plug of the Motor block.

Finally, set up the Motor Block to the configuration in Figure 7. Then download your program and run it. Try putting your robot on a piece of paper so that you can turn it in place. You will find that no matter which way you try to turn, the robot always returns to face south. Furthermore, the faster you move the paper, the faster the robot turns.

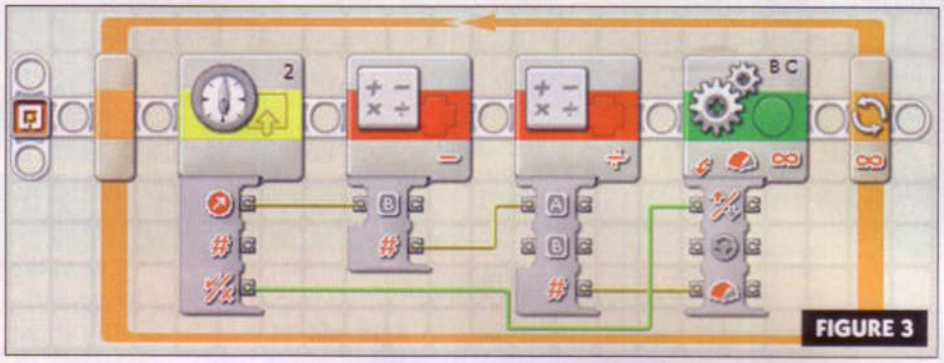


FIGURE 3

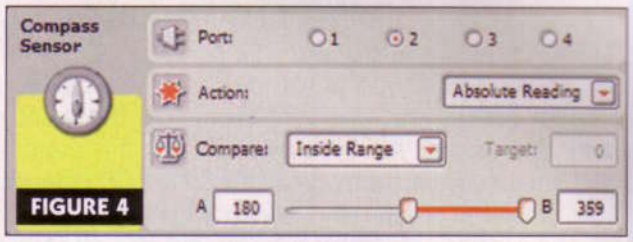


FIGURE 4

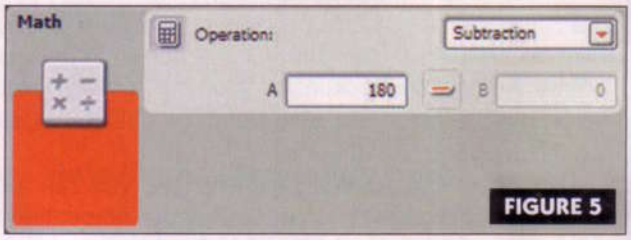


FIGURE 5

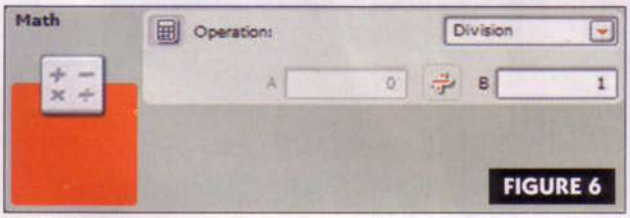


FIGURE 6

What's going on? Two things really. The first we've already gone over, establishing a margin of error in order to send more or less power to the motors, depending on how far off the robot is from facing south.

The next thing is to watch the Logic plug. The Logic plug sends out a binary signal (0 or 1, off or on, true or false), depending on whether or not it is in the range of 180-359 that tells the robot which direction to turn to get back facing south the quickest. If the robot is off by a small amount, it doesn't have to turn all the way around; it knows which way is the quickest back to south.

Like most things made with LEGO, the Runt chassis is a work in progress. By the time this article runs, the design will probably have changed. For updates to the design, building instructions for other sensor attachments, and an additional program by Brian Davis that uses the Relative Heading of the Compass Sensor, visit **www.LEGOedwest.com**. SV

FIGURE 7