

Reference

Random Numbers

Sometimes a behavior will call for a robot to use a random number in one of its measurements. This may seem strange, but randomness can actually be helpful to a robot in avoiding patterns of movement that would otherwise get it “stuck”.

Using Random Numbers

Random numbers is pretty straightforward. Wherever you want the random number to appear, simply add the code `random(maxNumber)`. Each time the line is run, a random (whole) number between 0 and the number you entered will fill in the spot where the `random()` command is.

```
task main()
{
  bMotorReflected[port2]=1;
  motor[port3]=127;
  motor[port2]=127;
  wait1Msec(random(5000));
}
```

Wait for a random time

The number of milliseconds that the `wait1Msec` command will wait for will be a random number between 0 and 5000.

This program runs the robot forward for a random amount of time up to 5 seconds.

Using Other Numbers

If you need something other than whole numbers between zero and something, you may need to be a little creative...

```
4000 + random(1000)
```

Minimum value (as shown: 4000-5000)

Adding the random value “on top of” a base number lets you get random numbers between a minimum (the base number) and a maximum (base+maximum random) value.

```
random(100)/100
```

Percent (as shown: 0-100% in 1% increments)

Dividing your random value by its own maximum value normalizes the value so that it always falls between 0 and 1.

Seeds

Computers can't be truly random. Instead, they try to use a hard-to-predict series of numbers based off a “seed” value. Under certain circumstances, you may want to set the seed manually.

```
srand(123);
wait1Msec(random(5000));
```

Set random seed

The `srand` command sets the random number seed for this robot. Run with the same seed, “random” numbers will always be generated in the same sequence.