Thinking in Terms of Behaviors

Behaviors

Robot programmers break down every action a robot makes into simple tasks called behaviors. Robot programs are built one behavior at a time. A behavior is anything your robot does: turning on a single motor is a behavior, moving forward is a behavior, tracking a line is a behavior, navigating a maze is a behavior. There are three types of behaviors: basic behaviors, simple behaviors, and complex behaviors. New programmers will find it much easier to make their robots "behave" when they learn to think in these terms.

Basic Behaviors

Basic behaviors are the most fundamental decisions a programmer makes for his/her robot. Examples would include turning on a single motor, or checking a single sensor port. Basic behaviors are combined to form simple behaviors.

Simple Behaviors

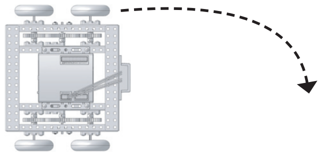
Simple behaviors are small, combinations of basic behaviors that allow your robot to perform a simple, yet significant task, like moving forward for a certain amount of time. These are perhaps the most useful behaviors to think about, because they are big enough that you can describe useful actions with them, but small enough that you can program them easily by combining basic behaviors.

Complex Behaviors

These are behaviors at the highest levels, such as navigating an entire maze. Though they may seem complicated, one nice property of complex behaviors is that they are always composed of smaller behaviors. This means that if you observe a complex behavior, you can always break it down into smaller and smaller behaviors until you eventually reach something you recognize.

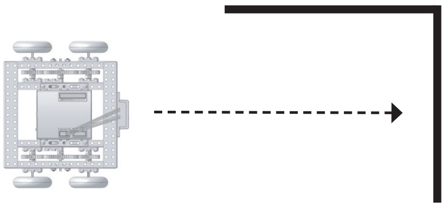
The Basic Behavior...

Turn on left motor in forward direction



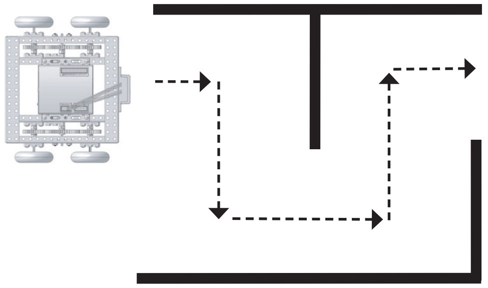
...is used in the Simple Behavior

Move forward until Bumper Switch Sensor is pushed



...which is used in the Complex Behavior

maze navigation



HELPFUL HINTS:

1. Begin by breaking the programming challenge into simple behaviors.
2. Divide the simple behaviors into basic behaviors; these will be the building blocks you will use to build the robot program.
3. Test each simple behavior as you build it.
4. Add a wait state, sound, or some other signal between simple behaviors. This will allow you to verify that the program worked to that point.
5. Test. Test. Test. The key to building a reliable program is to test it under a variety of conditions.