4A-Dog.vi
Fiamma:Users:andy:Professional:UW Design:Art 483, Winter 2007:Robot Exercises:4A-Dog.vi
Last modified on 12/13/06 at 8:27 PM
Printed on 12/13/06 at 8:51 PM

The Dog [A]

Port 1: touch sensor
Port 2: light sensor
Port 3: sound sensor
Port 4: distance sensor

Port A: motor
Port B: lamp
Port C: motor

This robot crudely models the behavior of a dog. Being a canine, it is a highly sentient, intelligent, and perceptive creature, of course. Our dog is a hunter and tracker.

This dog's mission is to find something ahead of it in the bush. Since we don't have smell sensors for this robot, instead we'll have him follow a visual trail on the ground–a black line on a light background.

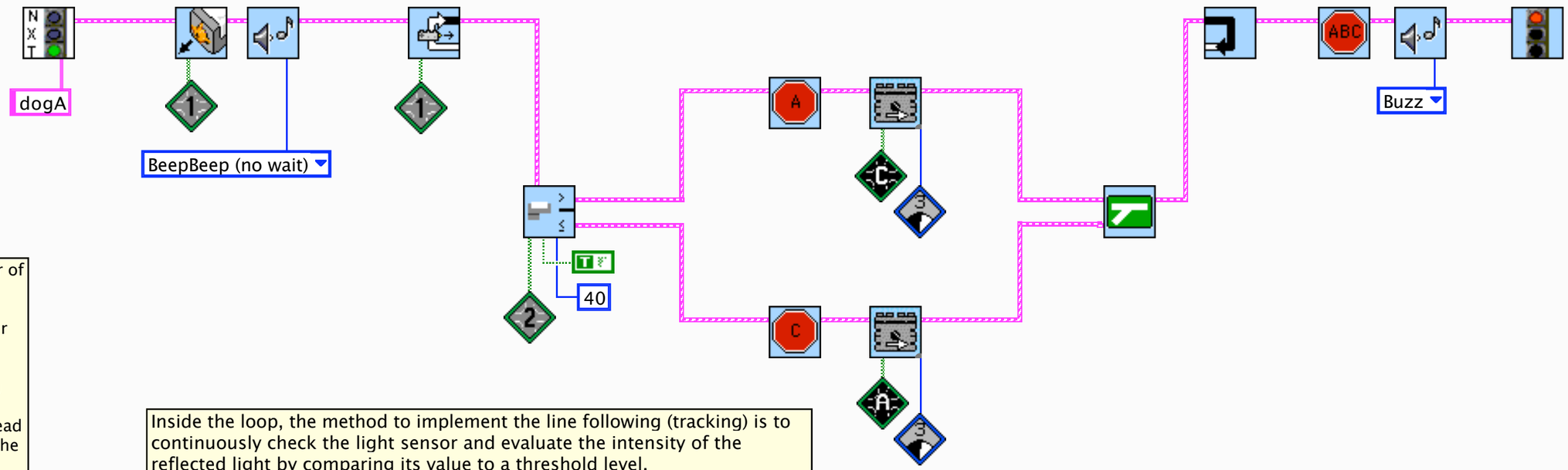
The robot does this by using a light sensor pointed at the ground to evaluate reflected light and determine its intensity, and then steering itself to stay on the black line.

In addition to this activity, the robot is simultaneously doing another: stopping if it makes contact with anything.

This program is a simple modification of the Moth C program. Inside the main loop, the logic for the light sensor is modified to implement the tracking behavior.

To start the dog tracking, wait until the bumper is touched, then beep, and start the main loop.

This loop will repeat as long as the touch sensor is released (that is, not pressed). As soon as the sensor is touched, the loop exits. This means that the segment inside the loop will execute continuously, and as fast as the microprocessor can go, until the robot hits something.



Inside the loop, the method to implement the line following (tracking) is to continuously check the light sensor and evaluate the intensity of the reflected light by comparing its value to a threshold level.

If the sensor detects a value above the threshold (here we set it to 40% brightness, but this can be changed depending the specific environment), then it is assumed to be white and the robot is off the track. If it is 40% or less, then it is assumed to be black and on the track.

If the robot is on the track, it turns a little to the right and continues forward. If it is off the track, it instead turns a little to the left and proceeds forward also. Since it is doing this continuously, in this way it inches forward (at a given speed, also modifiable according to taste), constantly adjusting its course to the left or right to stay on the black line.

This scheme seems like magic, but it actually works. There is a flaw in the logic which you will undoubtedly discover if you give the robot a complicated track to follow, but it works especially well for a circular track. Improving the scheme to follow a more general track is, as math professors love to say, left as an exercise for the student.

Designing Behaviors
Fundamentals of Interface Design
UW Division of Design
ART 483, Winter 2007
Davidson/Roesler