

Using the Calibrate commands on the Norland Research Calculator robot.

First, we have to understand how the robot gets its motion commands. We send an electronic pulse to each servomotor. These pulses go from 1 to 2 milliseconds. A pulse of around 1.5 causes the servomotor not to move. A pulse of 1 millisecond causes the wheel of the robot to turn full speed clockwise and a pulse of 2 milliseconds causes it to turn full speed counterclockwise. As we moved these pulses closer to 1.5 milliseconds, the speed slows. This is not linear. To make our software simpler, we only use the values of 0 to 255 to control the speed. Zero is full speed clockwise and 255 is full speed counterclockwise. To have the robot move forward, we must turn the right motor clockwise and the left motor counterclockwise.

So we send the right motor a zero and the left motor a 255 and the robot goes forward. Send the right motor a 255 and the left motor a zero and the robot goes backwards.

Now, due to outside influences, we do not always get a robot that goes perfectly straight. This can be due to battery levels, friction, manufacturing tolerances, air currents, surface textures and grain, just to name a few. Our robot is called open loop, as we have no feedback of the wheel speed. More expensive and complicated robot kits are called closed loop. They have encoders on each wheel that talk with the processor to control the speed. Then, the only variable is the environment.

One simple cure is to increase the diameter of the slow wheel with a rubber band or two! Or, build our maze with short narrow passageways. But, we are going to take that a step further. We have the ability to change the speed of each wheel! The following program lets you do just that. Have this program in your calculator, robot attached, robot on, and run the program. Press the button under the GO(zoom). Note which way the robot veers and press the bumper to stop the robot. Now, to slow down the faster wheel, press the button under the DN on that side and notice the value changing for that side. Due to the design of the electronics, you may have to change this 70-100 units to see much speed change. (Like we said, this value is not linear) Press go and see the speed change. Press the bumper and repeat until the robot goes straight. As the values get to 127(+/- 15), that wheel should come close to stopping. Keep going and the wheel reverses. Once you fine-tune this, you now have your calibration values to get your robot to run straight. Just plug these calibration commands in at the beginning of your program. (At least until something changes, but you can run the program again)

```
AxesOff
ClrHome
0->B
255->C
Lbl A
ClrHome
Disp " CALIBRATE"
```

```
Disp " UP DN BUTTONS"  
Disp "CHANGE FWD SPEED"  
Disp "PRESS GO TO TEST"  
Disp "BUMPER STOPS BOT  
Disp "PRESS ON TO END"  
Output(8,1,"UP DN GO UP DN")  
Output(7,1,C)  
Output(7,14,B)  
While 1  
getKey->K  
While K=0  
getKey->K  
End
```

```
If K=11:Goto G  
If K=12:Goto H  
If K=13:Goto J  
If K=14:Goto E  
If K=15:Goto F  
End  
Goto A
```

```
Lbl E  
B-1->B  
Output(7,14,B)  
Goto A
```

```
Lbl F  
B+1->B  
Output(7,14,B)  
Goto A
```

```
Lbl G  
C+1->C  
Output(7,1,C)  
Goto A
```

```
Lbl H  
C-1->C  
Output(7,1,C)  
Goto A
```

```
Lbl J  
Send({421,C})  
Get(R)  
Send({412,B})  
Get(R)  
Send({222,200})  
Get(R)  
Goto A
```