Draco

Mark Garrett and Clayton Hoover

The design of the robot portion of the robot was very simple. It had sensors on it, but that was only because I could not convince Clayton to take them off. They were only there for ascetics. Apparently he liked how it made it “look like a crab”. Since we were recycling the hardware from the first project, I don’t have as strong an understanding of the hardware as I would like. And I don’t have access to it. From what I can tell he attached two motors to the bottom of the brick, and attached the ball and socket to the side of the motors with a port to keep it balanced. There were some other pieces to hold it all together. The design resembled the robot from the instructions at this website. <https://le-www-live-s.legocdn.com/sc/media/lessons/mindstorms-ev3/building-instructions/ev3-rem-color-sensor-down-driving-base-d30ed30610c3d6647d56e17bc64cf6e2.pdf> but with the sensors configured differently.

//controller pseudocode

Setup

While(not esc button)

Read accelerometer data

Byte[] o= new byte[1];

if(x value if above deadzone and z is slightly smaller)

byte=4; //Send backward byte pattern

if(x value if below -deadzone and z is slightly smaller)

byte= 3; //forward byte pattern

if(y value if below -deadzone and z is slightly smaller)

byte= 2; //Send left byte pattern

if(y value if above deadzone and z is slightly smaller)

byte= 1; //Send right byte pattern

else

byte= 0; //Send stay byte pattern

update the display

output.write(o[0]) //like in EV3Connect.java

Wait a tenth of a second to keep from having the protocol from running too quickly

End of while loop

//car pseudocode

Open connection

While(not esc button)

Wait for a byte

if n[0]==3 //forward

spin motors both forward with max speed using Synchronization

else if n[0]==1 //right

spin motor B backward and motor C forward with1/8th the max speed using Synchronization

else if n[0]==2 //left

spin motor B forward and motor C backward with 1/8th the max speed using Synchronization

else if n[0]==4 //backwards

spin motors both backward with max 1/4th the max speed using Synchronization

else //stay

set the speed to be half the max speed on both motors

wait 5/100ths of a second

stop both motors at the same time

end of while loop

The only problem with our hardware was lack of speed. The way we would have fixed this would be to take advantage of gear ratios. As I understand our best strategy would be to use the 40 tooth gear on a motor to turn an 8 tooth gear connected to a wheel or axel. In terms soft. We should have change the Bluetooth protocol to send an array of 8 bytes instead of one, and send a byte representation of the x and y axis on the accelerometer converted to bytes or float(0) and float(0) if the z axis was not below its threshold. On the side of the car it should had two variables for left and right speed, with left x-y+C and right x+y+c where \*=either + or x and c=some fraction of max speed. This would have made it easier to control.