Controlling the ER1

The example is given in testrobot.m

The ER1 is controlled using the RCMplugin. Make sure that you have: RCM.dll, ER1SerialInterface.dll, and ptypes20.dll in your working directory. There are a group of .m files for wrapping up the functions available in RCM. Be sure to put these in your working directory or in a directory on your path.

Run the following commands to create an RCM object, initialize the system, start the wheels, and start the odometry:

```
CreateRCM(); % Creates the RCM object... must be called first
RCMInit('COM7'); % Initializes the RCM on COM7
WheelInit(); % Initializes and turns on the wheels
OdometryInit(20); % Initializes the odometry with a 20 Hz frequency
```

COM7 may need to be changed depending on the driver setup of your particular computer. Either find the "USB Serial Port" device under "Ports" in Windows' Hardware Device Manager or ask us. Once those commands are issued, velocity movement commands can be sent using:

```
MoveL(v);
MoveR(v);
```

Where v is the velocity command (in m/s) to the individual wheel, either left or right. To get the current odometry or velocity of the robot, use:

```
p = Position();
v = Velocity();
```

Position returns a three component vector: [x y θ], referenced with respect to the initial position of the robot when the odometry was initialized. The reference frame of the robot is defined with +x pointed out the front of the robot, +y pointed out the left side of the robot, and +θ corresponding to counter-clockwise rotation. Velocity returns a two component vector: [v ω] of the forward and turning velocities as defined above.

When done using the ER1 hardware, stop the RCM program and free the memory by calling the following functions:

```
WheelKillPower(); % Stops the robot and turns off the wheels
DestroyRCM(); % Frees the RCM object
```

Using the IR sensor # 1 (Old sensor)

The example is given in readIRsensors.m

The output of the sensor is a triple [ir1 ir2 ir3] where each number denotes the reading from one of the IR sensors. The range of each reading is [0 255], and higher readings indicate proximity to an obstacle.

Using the new IR sensor (A better IR sensor) : TBA

Obtaining Camera Images

To obtain images and video we use the Image acquisition toolbox in Matlab.

Furthermore, you can pull images from the USB Webcams using the VFW software in MATLAB. With VFW.dll in your working directory, call the following command to get image data:

pic = vfw('grab')

This command calls VFW, grabs the current image, and stores it in the variable pic. The image is stored as a 3-D array, where the first two indices correspond with image row and column and the third controls the color layer: R, G, or B.