Lola’s Programming Chain as of 4/9/2014

Main : output.txt (Contains output

robotWindow:

Up:

Down:

Left:

Right:

Forward:

Backward:

EForward:

EBackward:

ELeft:(Overloaded)

:ELeft (Robot)- Turns 90 degrees left.

:ELeft(Robot, Time) – Turns left for number of ticks specified.

ERight:

:ERight(Robot)- Turns 90 degrees Right.

:ERight(Robot, Time)- Turns Right for number of ticks specified.

waterTests: -Tests both Turbidity[0] and Salinity[1], returns double array of values.

salinityTest: -Tests Salinity, returns double Salinity.

turbidityTest: -Tests Turbidity, returns double Turbidity.

FullRun: OBSELETE, deleted from Github 4/3/14

setPing: (Overloaded)

:setPing(Robot, whichPing, Lola)-Detects Ping X[0] or Y[1] based on int whichPing input, gives the Ping detected to Lola, and returns Ping.

:setPing(Robot, whichPing)- Detects X[0] or Y[1] based on int whichPing input, and returns Ping.  
:setPing(Robot, Lola)- Detects both X and Y and passes them to Lola.

:setPingATBRIDGE(Robot, Lola)- Detects X and passes it to Lola, returns array.

:setPing(Robot)- Detects X and Y and returns them in an array.

testDoublePing: (Overloaded)

:testDoublePing(Robot, Lola) – checks X and Y and sends to Lola, also prints and returns coordinates.

:testDoublePing(Robot) – does the same but doesn’t send to Lola.

Retrieve:(Robot, Times)- Rams forward then backs up, based on int Times.

coverOpen: - Opens cover with servo 1.

coverClose: - Closes cover with servo 1.

bridgeRun: (Overloaded) Contains Ping.

:bridgeRun(Robot, Time)- Runs forward at high speed for specified time.

:bridgeRun(Robot)-Uses default values to keep Lola straight as bridge is crossed. Similar to checkForward in Ping.

bridgeFind: (Robot,Lola) –Contains Ping and Movement. Tests three positions for bridge using the EPLIST. If found it will stop Lola and return out of the function.

getPosition: (Robot, int[] expectedPosition, Lola) – Contains setPing. detects PingX and PingY using setPing, compares it to expectedPosition, and returns Coordinates[changes needed X,Y, PingX,Y].

correctPosition: (Robot, int[] Changes, Lola) – Contains Eleft, right, forward, and backward. Runs motors to correct position based on changes as specified in int[] Changes. Also checks orientation to decide most optimal way to get to new position and sets the position it went to in the EPLIST as 0.

testPosition: (Robot, int[] expectedPosition) – Driver function for getPosition and correctPosition. Checks to see if within 5 of position, if not reruns.

UTurn: (Robot, int Direction) - Executes a 180 degree turn, with direction based on int Direction. 0 for Left, 1 for Right.

testBridge: (Robot) – Runs along in front of bridge searching with line sensor, then once found executed bridgeRun. OBSELETE

Bump: -Detects if bump sensor is pressed.

IR: -Detects from IR Proximity Sensor and does math to convert to useful value.//OBSELETE

ColorTest: -Returns output from line sensor.

ballArm :(Robot, Angle) -Moves ‘Arm’ servo to specified angle.

getEPosition: - Returns the two motors’ positions in an int array[Left,Right].

Bump: -Detects if bump sensor is pressed.

Final:( Robot, int First,Second,Third,Fourth)- Driver program for full robot run. Accepts four ints for the four dispenser values. Calls most functions.//OBSELETE

Movement: Constructor class for Movement, which generally contains movement related functions, including Forward, Backward, Left, Right, Emovement, Uturn, open/close cover, bridgeRun, ballArm, Retrieve, createExpectedPosition, correctPosition, and the Ping Object. Also contains basic conversions between cm and ticks. NOTE: Not necessary, just convenient for coding to have functionality grouped.

Ping: Constructor class for Ping, which generally contains Ping sensor related functions, including setPing, testDoublePing, testPosition, createExpectedPosition, checkForward and all its variants, as well as basic conversions between ticks and cm. NOTE: Not necessary, just convenient for coding to have functionality grouped.

LolaObjectMichael: Constructor class for LolaObjectMichael, usually referred to as Lola. Contains Movement, attributes X and Y Coordinates as well as Coordinates array, as well as current number of balls for Salinity and Turbidity, as well as encoder values start and end, as well as Lola’s size and movementState, as well as conversions between cm and ticks, as well as the EPLIST. Also contains getters and setters for all attributes. NOTE: Setters interact with each other. i.e. setX sets the X attribute, but also sets Coordinates[0], as they are the same attribute, just called different ways.

NormalObjectMichael: Constructor class for NormalObjectMichael, which is a basic obstacle like a dispenser. Contains attributes XLength and YLength as well as Lengths array, along with getters and setters for all attributes. Used to check for if the Robot will hit an obstacle, as seen by ping sensors.//NOT USED

BoundaryObjectMichael: Constructor class for BoundaryObjectMichael, which is the field. Contains attributes XLimit and YLimit as well as Limits array, along with getters and setters for all attributes. Used to check that the Robot will stay within the object, aka the limits of the course, and not get stuck on a wall.//NOT USED

checkBounds: Compares Lola’s position to Boundary Object. //NOT USED

createExpectedPosition:(Robot,Lola) – Picks the first two values in EPLIST and returns them in an array. Also cycles out all 0’s, allowing easy deletion by changing a value to 0.

Automonous\_Driver: (Robot,Lola)—Imports Ping, Movement and bridgeFind. Executes full autonomous run, calling all functions, defining EPList and running all driver math.

The GO button.