Visual Aids in Programming
Mark Cieslikowski, Roak Ely, Mekai Ely
Lincoln BCBs Community Team 07-0047
Linda Reynolds: teckteacher@yahoo.com

Visual Aids in Programming

1. Introduction

Our team is a first year team with inexperienced members. Some team members have participated in FIRST LEGO League [1] and have used Robolab [2]programming so they know the basics of how to program a robot. But IC [3] is quite different from Robolab which uses icons linked together by wires to form program chains. It is very visual whereas IC is all text. To make it easier to learn IC we decided to use some visual aids.

2. The Flow Chart

The first aid is the flow chart. Flow charts have been around for many years. A flow chart is a pictorial tool that can be used to organize steps of a process using pictures or symbols. [4]

2.1 Computer Software for Flow Charts

At first we tried drawing out our flow chart on paper, but it was a mess. So we looked for a computer based program that would let us create the flow chart and print it out. We tried using Pages, the MAC based word processor, but it required a lot of work on our part. There are no graphics designed for flow charts in Pages. Then we tried Kidspiration. It did have graphics that looked like the ones for use in flow charts. But the program is very limited in how you can put things on the page and doesn't work well for making a flow chart for an IC program.

Finally we searched on the internet for a program that would create flow charts which were downloadable. We found OmniGraffle.[5] (http://www.omnigroup.com/applications/omnigraffle/) It has a great section just for the creation of flow charts with all of the graphics and lines included. It also saves the flow chart as a pdf file so we can include it in our Botball Online Documentation Project. You need the pro version to edit more than 20 items on a page. It is only $79 and well worth it for all kinds of outlining and brainstorming projects.

2.2 Our Flow Chart

Our team decided that we should keep the programming as simple as possible because we are inexperienced. In fact, none of us has done any programming in IC before this year. We decided it would be better to use simple programing to do simple movements. By using a flow chart we are able to see what we need to do in the written program. We created the flow chart and wrote

out comments for each section so we could see what the robot was doing at each part of the program. The comments are from out pseudo code that we wrote for the Botball Online Documentation Project. See Figure #1 below. This is the flow chart for our 1st robot for the regional Botball tournament.

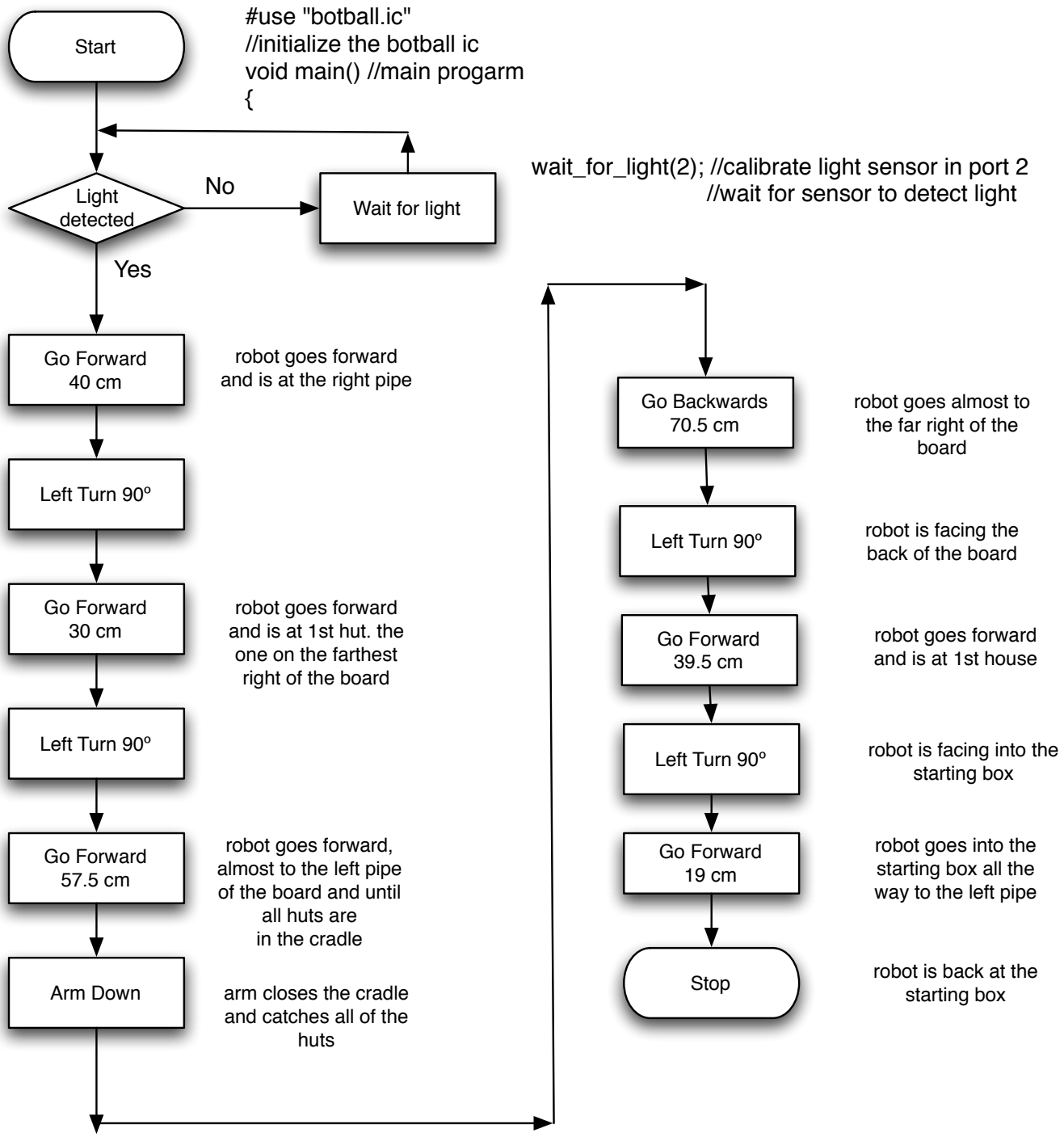


Figure #1 : Flow Chart for Robot 1 with pseudo code
Tasks: gather all 3 huts in a row, place lava shields inside huts and then return all huts to the starting box.

3. Adding Code to Flow Chart

To use the flow chart to write our program we decided to put each section of code on the flow chart matching it up with the part of the flow chart that the code would accomplish. We created a page that showed both the flow chart and the IC code. See Figures #2 and #3 below.

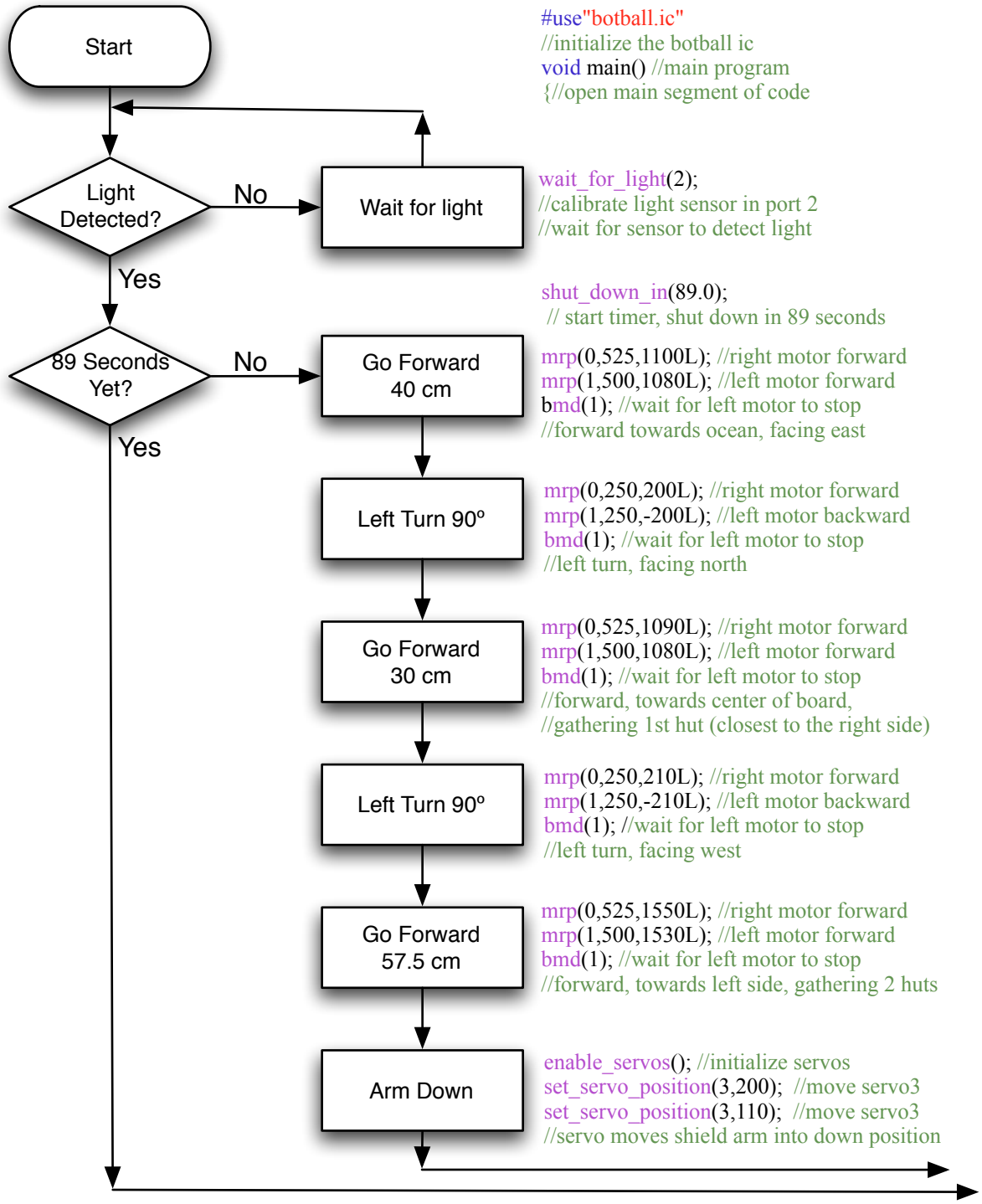


Figure #2
Flow Chart with Matching IC Code (1st half of the program)

We took our flowchart of the program and inserted IC code for each step of the flowchart steps.
We ran the robot1_ticks_test to find the right numbers and then wrote the final code from this flowchart.

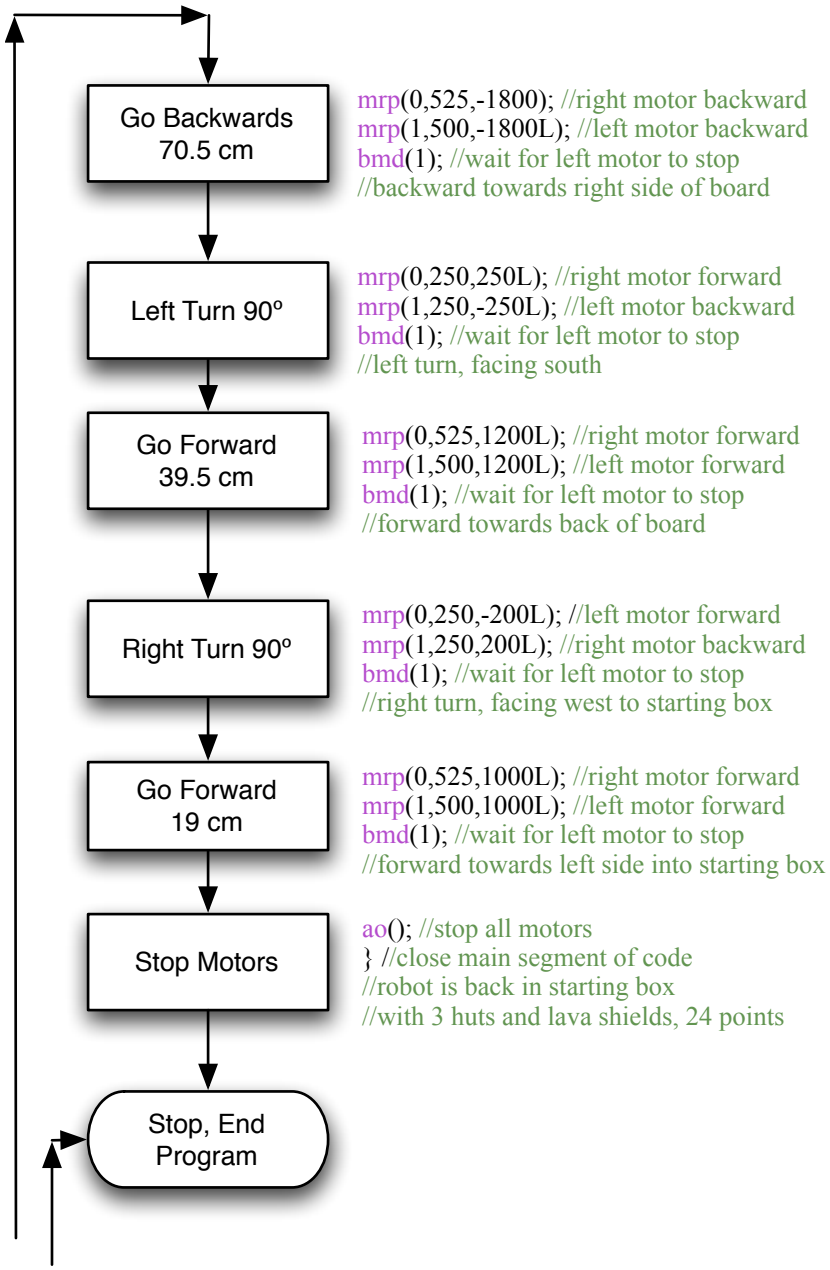
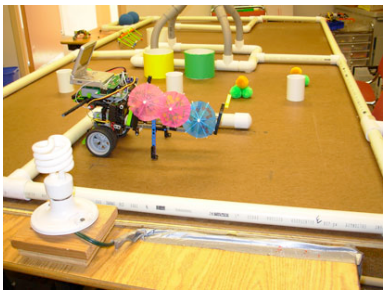


Figure #3
Flow Chart with Matching IC Code for Robot 1(2nd half of the program)

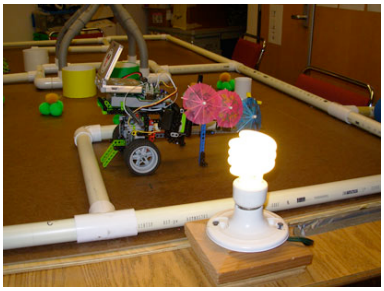
4. Visually Enhancing the IC Code with Photographs

The last step in our process was to run the code on our robot an see what each section of code did. This helped us to make the needed changes to make the program work properly. To help the programmers we took pictures of what the robot should be doing during each section of the code. then we put them along side the code so we had a visual guide of the entire program.
See Figures #4 and #5 below.

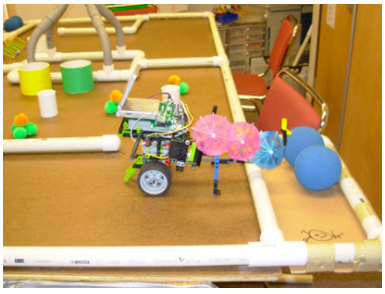
1.
```
#use"botball.ic"
//initialize the botball ic
void main() //main program
{//open main segment of code
```



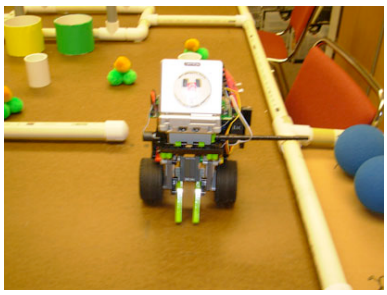
2.
```
wait_for_light(2);
 //calibrate light sensor in port 2
 //wait for sensor to detect light
shut_down_in(89.0);
// start timer, shut down in 89 seconds
```



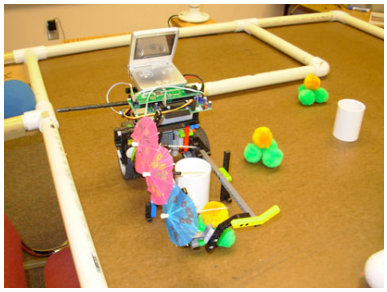
3.
```
mrp(0,525,1100L); //right motor forward
mrp(1,500,1080L); //left motor forward
bmd(1); //wait for left motor to stop
//forward towards ocean, facing east
```



4.
```
mrp(0,250,200L); //right motor forward
mrp(1,250,-200L); //left motor backward
bmd(1); //wait for left motor to stop
//left turn, facing north
```



5.
```
mrp(0,525,1090L); //right motor forward
mrp(1,500,1080L); //left motor forward
bmd(1); //wait for left motor to stop
//forward, towards center of board,
//gathering 1st hut (closest to the right side)
```



6.
```
mrp(0,250,210L); //right motor forward
mrp(1,250,-210L); //left motor backward
bmd(1); //wait for left motor to stop
```

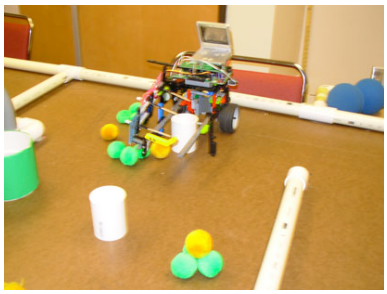
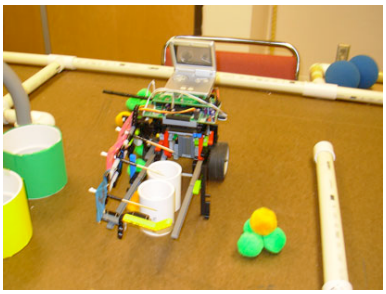
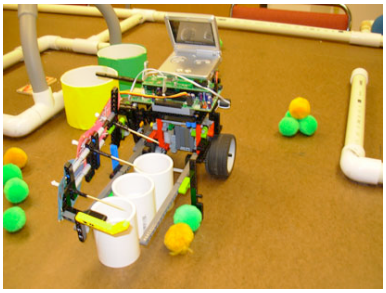


Figure #4
First 6 Steps of the Code with Photos Showing the Robot's Position

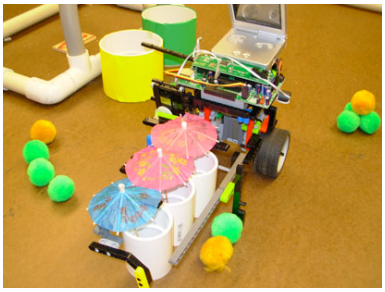
7. mrp(0,525,1550L); //right motor forward
mrp(1,500,1530L); //left motor forward
bmd(1); //wait for left motor to stop
//forward, towards left side, gathering 2 huts



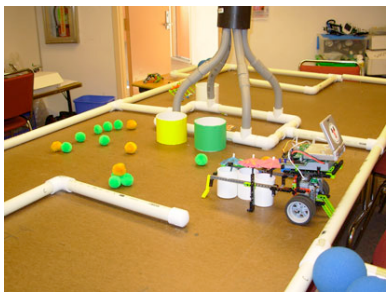
8. enable_servos(); //initialize servos
set_servo_position(3,200); //move servo3
set_servo_position(3,110); //move servo3
//servo moves shield arm into down position



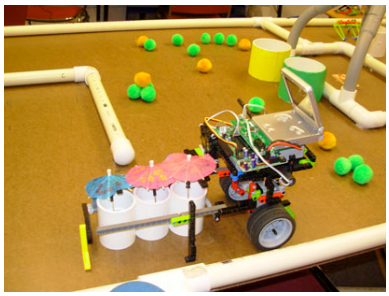
9. enable_servos(); //initialize servos
set_servo_position(3,200); //move servo3
set_servo_position(3,110); //move servo3
//servo moves shield arm into down position



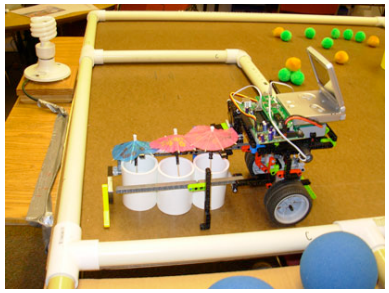
10. mrp(0,525,-1800); //right motor backward
mrp(1,500,-1800L); //left motor backward
bmd(1); //wait for left motor to stop
//backward towards right side of board



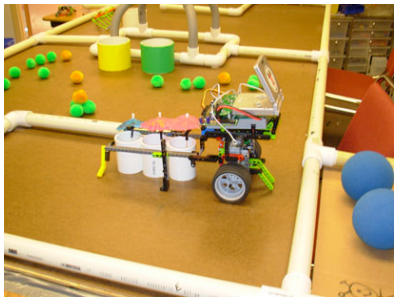
11. mrp(0,250,250L); //right motor forward
mrp(1,250,-250L); //left motor backward
bmd(1); //wait for left motor to stop
//left turn, facing south



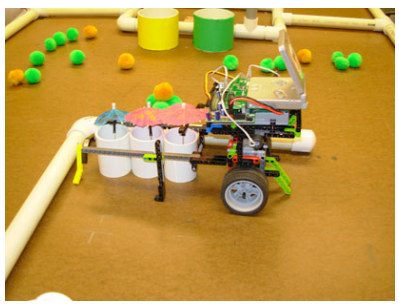
12. mrp(0,525,1200L); //right motor forward
mrp(1,500,1200L); //left motor forward
bmd(1); //wait for left motor to stop
//forward towards back of board



13. mrp(0,250,-200L); //left motor forward
mrp(1,250,200L); //right motor backward
bmd(1); //wait for left motor to stop
//right turn, facing west to starting box



14. mrp(0,525,1000L); //right motor forward
mrp(1,500,1000L); //left motor forward
bmd(1); //wait for left motor to stop
/ /forward towards left side into starting box



ao(); //stop all motors
} //close main segment of code
//robot is back in starting box
//with 3 huts and lava shields, 24 points

Figure #5 Steps 7-14 of the Code with Photos Showing the Robot's Position

References:

[1] FIRST LEGO League http://www.firstlegoleague.org

[2] Robolab http://www.lego.com/dacta/robolab

[3] IC7.0.7 Copyright © 2007 KISS Institute for Practical Robotics
http://www.kipr.org/
http://www.botball.org

[4] http://en.wikipedia.org/wiki/Flow_chart

[5] OmniGraffle 4.2 Beta 1
Powerful diagramming & charting for Mac OS X.
T H E O M N I G R O U P
2707 NE Blakeley St · Seattle · Washington · USA · 98105-3118
1 800 315-OMNI · +1 206 523-4152
fax +1 206 523-5896