Final Code for the robot:

#include <SendOnlySoftwareSerial.h>

#include <ArduinoInit.h>

//Changes the direction the robot turns when both bumpers are trigerred

char turn = 'l';

//Contains the origional side of the robot

char sideOrig;

//Changes the direction that the robot goes when both bumpers are trigerred

int number = 0;

void setup( )

{

configArduino ();

//Turns on floor sensor light

outputHigh(4);

pause(500);

outputLow(12);

//Interrupts for when the bumpers are trigerred

attachInterrupt(0, bumper, LOW);

attachInterrupt(1, bumper, LOW);

//Sets the origional side

char side;

if (sideDark())

side = 'b';

else

side = 'w';

sideOrig = side;

//Finds navigation light and proceeds to the other side

FirstSide();

//Checks if the robot is still on the first side

if (sideDark())

side = 'b';

else

side = 'w';

if (side == sideOrig)

{

FirstSide();

}

//Finds the two target lights

SecondSide();

}

void FirstSide()

{

//Sets the side that the robot is on

char side;

if (sideDark())

side = 'b';

else

side = 'w';

//Turns on the orange LED to debug code and make sure that the robot is always detecting the right side

if (sideOrig == 'b')

{

outputHigh(11);

}

//The recorded light value that others are compared against

unsigned int first;

//The next value that is checked to see if it is larger(darker) than the first value

unsigned int second;

//Sets true to enter the loop

bool test = true;

//Sets the value to 0 so that if only the left is checked, then the right side is checked

int num1 = 0;

while (test)

{

num1++;

//Sets the current value to the first

first = readADC(4);

//Rotates the robot to the left by about 30 degrees

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

//Pause is long enough to get an accurate reading by the robot

pause(600);

//Sets the second value after the robot turns

second = readADC(4);

//Tests if the second value is darker than the first one while also making sure that the difference is significant enough

if (second>first && ((second-first)>100 || (second-first)<-100))

{

//Turns the robot again and repeats checking the next values to make sure that they are darker than first so that it is certain that the robot is facing the navigation light

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','A',100);

motors('2','B',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

//Turns the robot back to the highest light sensor reading

pause(500);

test = false;

motors('1','B',100);

motors('2','A',100);

pause(375);

motors('B','O',100);

pause(500);

}

}

}

}

}

}

}

}

}

}

}

//Repeats the steps above, but while turning right if only the first position on the right side was checked

if (num1 == 1)

{

bool test2 = true;

while(test2)

{

first = readADC(4);

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

second = readADC(4);

if (second>first && ((second-first)>100 || (second-first)<-100))

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

motors('1','B',100);

motors('2','A',100);

pause(35);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

test2 = false;

motors('1','A',100);

motors('2','B',100);

pause(375);

motors('B','O',100);

pause(500);

}

}

}

}

}

}

}

}

}

}

}

}

//The robot continues forward and checks the side that it is on until the side color is that of the opposite side

//If the robot keeps moving forward for more than 85 times then the robot searches for the navigation light again

int x = 0;

while (side == sideOrig && x<85)

{

x++;

motors('1', 'A', 50);

motors('2', 'A', 70);

pause(200);

if (sideDark())

side = 'b';

else

side = 'w';

}

//If the robot keeps moving forward for more than 85 times then the robot searches for the navigation light again

if (x==85)

{

motors('B','O',100);

pause(100);

FirstSide();

return;

}

//Exits the subroutine for the first side once the robot is on the other side

motors('B','O',100);

pause(100);

return;

}

//Checks if the robot is on the dark side and returns true if it is and false if it is not

bool sideDark()

{

unsigned int light = readADC(0);

if (light <35000)

{

pause(50);

return false;

}

return true;

}

void loop( )

{

return;

}

//Activated when either bumper is trigerred

void bumper()

{

//Turns the motors off and waits to make sure that the bumper was nto accidentilly trugerred without bumping into anything

motors('B','O',100);

pause(500);

//Checks if neither bumper was trigerred

if (readInput(2) == 1 && readInput(3) == 1)

{

return;

}

//Turns left about 30 degrees when the right bumper is hit

if (readInput(3)==0 && readInput(2) != 0)

{

turn = 'l';

//Moves the robot back and turns the robot to the left

motors('B', 'B', 100);

pause(400);

motors('B','O',100);

pause(100);

motors('1','A',100);

motors('2','B',100);

pause(70);

motors('B','O',100);

pause(100);

}

else

{

//Turns right about 30 degrees when the left bumper is hit

if (readInput(3)!=0 && readInput(2) == 0)

{

turn = 'r';

//Moves the robot back and turns the robot to the right

motors('B', 'B', 100);

pause(400);

motors('B','O',100);

pause(100);

motors('1','B',100);

motors('2','A',100);

pause(47);

motors('B','O',100);

pause(100);

}

else

{

//Turns right when both bumpers are trigerred and the robot turned right last

if (readInput(2)==0 && readInput(3) == 0 && turn == 'r')

{

//Changes the direction the robot turns when the center is hit, but only if it is the first time any bumper was hit

if (number == 0)

{

turn = 'l';

}

else

{

turn = 'r';

}

number++;

//Moves the robot back and turns the robot to the right

motors('B', 'B', 100);

pause(400);

motors('B','O',100);

pause(100);

motors('1','B',100);

motors('2','A',100);

pause(47);

motors('B','O',100);

pause(100);

}

else

{

if (readInput(2)==0 && readInput(3) == 0 && turn == 'l')

{

//Changes the direction the robot turns when the center is hit, but only if it is the first time any bumper was hit

if (number == 0)

{

turn = 'r';

}

else

{

turn = 'l';

}

number++;

//Moves the robot back and turns the robot to the left

motors('B', 'B', 100);

pause(400);

motors('B','O',100);

pause(100);

motors('1','A',100);

motors('2','B',100);

pause(70);

motors('B','O',100);

pause(100);

}

}

}

}

//Turns motors off and then returns to the previous subroutine

motors('B','O',100);

pause(100);

return;

}

void SecondSide()

{

//Moves the robot forward so that the robot is beyond the diamond in the center of the arena

motors('B', 'A', 100);

pause(800);

motors('B', 'O', 100);

pause(200);

//Looks twince ot the left and right and keeps turning if the value is lower

//The loop is limited to 500 so that if there is a problem it will go back the first side

int n = 0;

bool lookForSecond = true;

while (n<500)

{

//Turns the robot to the side that is greater

if(readADC(1)<(readADC(3)-200))

{

//Checks that left side and keeps turning until the next two are greater than teh previous value

unsigned int first;

unsigned int second;

bool test = true;

bool testLeft = false;

int num1 = 0;

while (test)

{

num1++;

first = readADC(2);

motors('1','A',100);

motors('2','B',100);

pause(75);

motors('B','O',100);

pause(600);

second = readADC(2);

if (second>first && ((second-first)>75 || (second-first)<-75) && first<30000)

{

pause(500);

motors('1','A',100);

motors('2','B',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

motors('1','A',100);

motors('2','B',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

test = false;

motors('1','B',100);

motors('2','A',100);

pause(150);

motors('B','O',100);

pause(400);

}

}

}

}

//Checks that right side and keeps turning until the next two are greater than teh previous value

if (num1 == 1)

{

unsigned int first;

unsigned int second;

bool test = true;

bool testLeft = false;

bool test2 = true;

while(test2)

{

first = readADC(2);

motors('1','B',100);

motors('2','A',100);

pause(75);

motors('B','O',100);

pause(600);

second = readADC(2);

if (second>first && ((second-first)>75 || (second-first)<-75) && first<30000)

{

pause(500);

motors('1','B',100);

motors('2','A',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

motors('1','B',100);

motors('2','A',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

test2 = false;

motors('1','A',100);

motors('2','B',100);

pause(225);

motors('B','O',100);

pause(400);

}

}

}

}

}

}

else

{

//Turns the robot to the right if it is brighter there

unsigned int first;

unsigned int second;

bool test = true;

bool testLeft = false;

int num1 = 0;

bool test2 = true;

while(test2)

{

num1++;

first = readADC(2);

motors('1','B',100);

motors('2','A',100);

pause(75);

motors('B','O',100);

pause(600);

second = readADC(2);

if (second>first && ((second-first)>75 || (second-first)<-75) && first<30000)

{

pause(500);

motors('1','B',100);

motors('2','A',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

motors('1','B',100);

motors('2','A',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

test2 = false;

motors('1','A',100);

motors('2','B',100);

pause(225);

motors('B','O',100);

pause(400);

}

}

}

}

if (num1 == 1)

{

unsigned int first;

unsigned int second;

bool test = true;

bool testLeft = false;

while (test)

{

first = readADC(2);

motors('1','A',100);

motors('2','B',100);

pause(75);

motors('B','O',100);

pause(600);

second = readADC(2);

if (second>first && ((second-first)>75 || (second-first)<-75) && first<30000)

{

pause(500);

motors('1','A',100);

motors('2','B',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

motors('1','A',100);

motors('2','B',100);

pause(75);

motors('B','O',100);

pause(600);

if (readADC(2)>first)

{

pause(500);

test = false;

motors('1','B',100);

motors('2','A',100);

pause(150);

motors('B','O',100);

pause(400);

}

}

}

}

}

}

motors('B', 'A', 100);

pause(600);

motors('B', 'O', 100);

//Checks if the robot is back on its origional side and tries to turn 180 degrees and go straight and if that does not work, the the robot goeas back to FirstSide()

char current;

if (sideDark())

{

current = 'b';

}

else

{

current = 'w';

}

if (current == sideOrig)

{

motors('1','B',100);

motors('2','A',100);

pause(365);

motors('B','O',100);

pause(200);

motors('B','A',100);

pause(900);

motors('B','O',100);

pause(200);

if (sideDark())

{

current = 'b';

}

else

{

current = 'w';

}

if (current == sideOrig)

{

//Goes back to the subroutine FirstSide() if the robot is back to the origional soie

FirstSide();

}

}

}

return;

}