

# EAS 102 Final Design Project Spring 2011

May 3, 2011

Section L

Team 8

James Janssen

Jonathan Kosir

Tess Barnett

Yan He

## Summary:

In this paper, we are going to introduce the entire process of Team 8's EAS 102 final design project, including a brief introduction of the goal of this project, designs of the robot, the code for our solution, all the testing procedure, the result of every test, and our conclusion. This project includes one of programming languages we have learned this semester--NXC language.

We had three different preliminary designs before deciding on our fourth and final design. Most of the time in the design of our robot was spent on the wheels that made the robot move along the track. Our final design involved two wheels and a stabilizer that prevented the robot from falling off the track on turns. We also had a feature added that made the robot a clap on/clap off robot. We could stop the robot at any time on the track and have it start again. Our design also allowed for the robot to run in both directions on the track without having to make any adjustments to the physical design.

During testing, many of the problems we ran into involved keeping the robot on the track during turns and weight distribution. By adding the stabilizer we eliminated pretty much any chance of the robot falling off of the track. As for the weight distribution problem, we added waits in the code that allowed for the robot to stop shaking when the arm moved. This did not change the problem of the weight not being evenly distributed but it was a clever way of still making the robot do the task successfully.

Our results for this project were quite successful. The problems we encountered, big or small, we fixed and made the robot even better than it had previously been. We worked really well together and distributed the work evenly so no one felt overwhelmed. We took different approaches to solving the issues that arose while testing and tried to create a well-designed and unique robot.

We learned two important things during this project. The first one is how to approach solving problems we encountered. Sometimes it was a design flaw and sometimes it was a programming flaw. Either way we worked together to fix it. This leads us to the second thing we learned, teamwork. We learned how to work together and bounce ideas off of one another to complete the tasks of the assignment.

## Introduction:

For this task we were supposed to design a device that would hang from a track. It then had to follow the track and identify different colors of blocks and interact with them depending on the color. On the floor is black tape that indicates the location of the blocks. The robot must then identify if the block is red or green with a beep or other sound. If the block is green, then the robot will ignore it and move on. If the block is red then the robot will beep and knock down the block with an arm before it moves on. There is a physical bumper at the end of the track (fingers or hands as a bumper is also allowed) , and the robot is supposed to correctly respond to the bumper by stopping and playing some notes.

In this project, our robot could not touch the ground, and should keep from falling off the track. The track was formed with several supports and a pipe. The height of the supports was 8 inches and the diameter of the pipe was 0.5 inches (+/-0.1 inches). The dimension of the blocks was 0.75 inches thick, 6-8 inches high and 2-4 inches wide. The length of the track was 3-10 feet. We could set the blocks depending on how our robot functions. The farther the distance from the center line to the block, the higher we earn points. Also, our final score depended on our robot's running time, the total cost of the robot, and the robot design based on quality, stability, use of parts and looks. In the final demonstration, we would earn points depending on how our robot behaves. The robot should beep at the correct blocks to earn 5 points and then knock them down to earn another 10 points and avoided knocking down the incorrect blocks. If the robot knocked down those incorrect blocks, there would be negative attempted points for each blocks.

We were provided with a Lego brainstorm kit to build our robot. In this kit, we could use several kinds of sensors including touch sensors, light sensors, color sensors and ultrasonic sensors, which made a significant part and function of our robot. Our robot would start to run, detect the black tape, identify different blocks and stop after hitting the bumper with the help of sensors. Three Interactive Servo Motors were available for using. They were in charge of the moving system of the entire robot and the movement of the arm. The core of our robot was the NXT brick which contains all programs we write and decides every action of our robot. And it was surrounded by all

the physical designs including the moving system (with the Almighty Stabilizer), the touch sensor, the light sensor and the arm system. A lot of little parts like pins and bars were also included in the kit.

Design:

#### Idea 1

For part of the assignment we had to add a touch sensor to the robot so it would stop when it hit a bumper at the end of the track and play a tone. We decided to try and eliminate the touch sensor by having the robot recognize when the robot ran into the bumper. The idea was that there could be code written that would be able to tell when the motor running the wheels was slowed down. We were unable to find a part of code that could complete this task. That was because both speed and power are considered the same thing in the motor. There is no way of discerning between the two. The motor was just not sophisticated enough to be able to measure when the speed changed. Sadly, we had to scrap this idea and keep the touch sensor.

#### Design 1



This first idea involved using four wheels to add stability and keep the robot from shaking as it moved down the track. But we found that as it tried to make turns, the stationary wheels would come off of the track (sometimes even the tires came off the wheels) and the robot failed.

## Design 2



We decided to remove the second set of wheels and just have one set. This was our main design for quite some time until we started testing more and more. When we added the arm the weight was no longer evenly distributed. The robot would swing violently back and forth whenever the arm moved. We determined that we needed some sort of second support that would provide more stability.

## Design 3



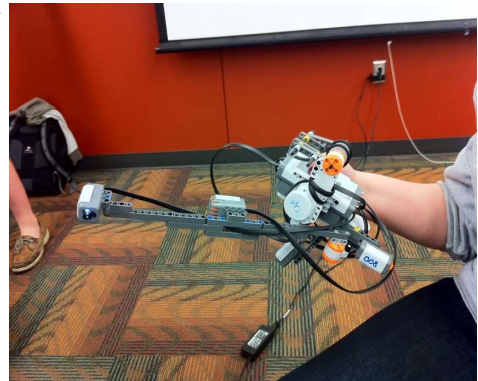
We added a simple stabilizer and a rubber band for more friction on the track. This did not work to our advantage. The stabilizer really did not add any stability. It only helped the robot stay on the track during turns but the robot still did a lot of shaking when the arm moved. The wheels would also slide on the rod holding them and the robot then didn't have enough traction to move on the track.

## Final Design



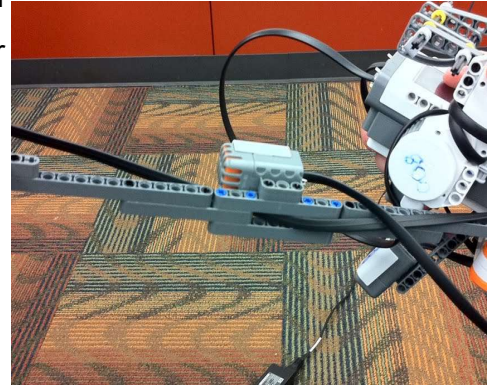
This ended up being our final design. We part we added for stability we like to call it “The Almighty Stabilizer.” It allows the robot to make tight turns and still remain on the track. We also added pieces on either side of the wheels so they wouldn’t move back and forth and stay in the same position while traveling on the track. We positioned the parts of the stabilizer on the outside of the wheels so when the robot traveled on a curve, the bent track could be easily maneuvered between the two supports.

The arm is approximately twelve inches long and has a color sensor attached to the end of it so it can read the color of the blocks. It is attached to a motor that is attached to the bottom of the Lego brick. It swings out when the light sensor on the bottom of the robot sees the black tape. It then detects the color and knocks down the red blocks and leaves the green ones alone. This design of our arm was the original design and we did not have to change it at all.



The light sensor hangs from the bottom of the robot and is used to detect the black tape on the group. It is able to move back and forth on its axis so that when the robot tilts, it still stays at a 90-degree angle with the floor. The pieces it is attached to allows us to move it up and down depending on how close to the ground we want the sensor.

We added a sound sensor to the robot so we could decide when the robot stops and goes. The sound sensor gives the robot a clap on/clap off quality. Whenever the robot is in motion, you can clap your hands once and it will stop until you clap your hands again. It was originally put on the back of the robot but as we tested it we realized that it was picking up the sound from the motor moving the wheels. To remedy this, we moved it out to the arm. This also made it more convenient for us to turn it on or off.



#### Code:

```
#define NOISETHRESHOLD 70
#define THRESHOLD 45
byte speed = 120;
// Method which checks block color
// If Red Knocks Block over and tones
// If any other color moves arm back
void checkColor()
{
// Set color sensor and assigns variable
    SetSensorColorFull(S1);
//
    OnFwdReg(OUT_A, 0, 0);
    RotateMotor(OUT_C, 40, -60);
    Wait(1500);

    byte valColor = Sensor(S1);
// If red knocks down and plays tone
    if(valColor == 5)
    {
```

```

        PlayToneEx(262,400,3,FALSE);
        RotateMotor(OUT_C, 50, -60);
        RotateMotor(OUT_C, 40, 120);
    }
// If not red moves arm back to start position
    else
    {
        RotateMotor(OUT_C, 40, 60);
        Wait(1000);
    }
}
// Method that when youch sensor is pushed in it stops
// moves back from the block slightly and
// plays tone
// It also changes speed to opposite direction
// So it goes around the track in oppisite direction
void touchSensor()
{
    // Sets touch sensor and assigns variable
    // If sensor is touched
    // *Stops Robot
    // *Changes speed to oppisite direction
    // *Plays tone
    // *Moves away from block slightly
    SetSensorTouch(S2);
    byte valTouch = Sensor(S2);
    if(valTouch == 1)
    {
        OnFwd(OUT_A, 0);
        PlayTone(494,800);
        Wait(480);
        PlayTone(554,800);
    }
}

```



```

        Wait(480);
        PlayTone(440,800);
        Wait(480);
        PlayTone(220,800);
        Wait(480);
        PlayTone(330,800);
        Wait(960);
        OnFwd(OUT_A, -speed);
        Wait(500);
        speed = speed * -1;
    }
}

// Preforms both methos and completes the rest of
// the task the needs to be performed
task main()
{
    // Sets and assigns Sound sensor
    SetSensorSound(S3);

    // Sets and assigns Light sensor
    SetSensorLight(S4);

    // Assigns count variable
    byte count = 0;
    //Loops program
    while(true)
    {
        byte valSound = Sensor(S3);

        touchSensor();
        if(valSound > NOISETHRESHOLD)
        {

```

```

    count += 1;
    Wait(500);
}
if(count%2 != 0)
{
    OnFwdReg(OUT_A, speed, 0 );
    byte valLight = Sensor(S4);
    if(valLight < THRESHOLD)
    {
        checkColor();
        OnFwdReg(OUT_A, speed, 0 );
        Wait(1000);
    }

}
else
{
    OnFwdReg(OUT_A, 0, 0);
}

}
}

```

### Testing:

The testing of our final robot had many different parts that needed to be evaluated. The first test we did was to see if the robot could stay on the track when it curved. Our final stabilizer was originally too tight and pushed the robot off when it went over the turns. We discovered that by putting the sides of the stabilizer just past the wheels, the robot could easily stay on the track. The robot moves smoothly across the track and does not shake as much when the arm moves to knock down the block.

Once the robot was finished one of our big problems was that the weight of the robot was not evenly distributed when the arm was fully extended. This caused the light sensor to move back and forth and it would read the black tape over and over and not continue on the track. To fix this problem, we programmed the robot to move forward a little bit before the sensor took another reading of the ground. This allowed the robot to get far enough past the tape that it wouldn't be stuck in a loop anymore.

During some of our earlier tests we noticed that when the arm moved outward to see the block, it didn't wait long enough to actually get a reading. It thought it didn't see anything and then would move on. It was okay for it to skip the green blocks because we didn't want it to knock them down, but we needed to knock down the red blocks. We added a wait in the programming after the arm was extended. This allowed the robot to stabilize a little bit before the color sensor tried to read the block.

For our extra credit idea we decided to add a sound sensor that would make the robot turn on and off when we clapped our hands. As we tested it, we ran into many different issues. The threshold for the sound was always either too high or too low. In order for the robot to hear us clap, the threshold had to be lower. But if it was too low, the sensor would pick up background noise and shut off. The sensor was originally located on the bottom of the robot and would pick up noise from the motor that ran the wheels. This caused the robot to stop in the middle of the track. We moved the sensor out to the arm and we no longer had to worry about it picking up the sound from the motor. Before our design had also required us to practically lie down on the floor and shout into the sensor to get the robot to stop and go. The new location of the sensor makes the robot much more user friendly.

## Results/Discussion:

The outcome of this project seemed to overall be very successful. Our robot works very well and can complete the tasks very well. We went through many different design ideas to make sure that the final robot was able to properly complete the task. We added extra sensors and code that made the robot very versatile when it travels on the track.

When it comes to the design of the robot, there are some things that work and some things that don't. Having only one point of contact on the track when there is an

arm that swings outward almost 12 inches is not a successful design. At least two well distributed points of contact are needed to keep the robot stable while traveling on the track and completing the task. Weight distribution is also very important when the robot has such a long and heavy arm. when the distribution is not correct, the robot shakes too much and fails when trying to complete a task. There are two different ways of fixing the problem. You can either counteract the weight by adding more pieces or you can change the code for the robot that allows the robot to extend the arm and stabilize itself before doing anything else. We chose the latter of those two because we were trying to be cost effective.

It is very important to know exactly how each of the parts of the robot work. When dealing with a light sensor, you have to take into consideration how sensitive it is. If you do not have the correct threshold, then the sensor could see more things than it is supposed to. The best thing to do is keep testing and find the perfect threshold that makes the sensor work at its best. Sound sensors are very sensitive and they have to have to be at the right threshold or they will pick up background noise and not work properly. making the sound sensor have a lower threshold doesn't work very well for our situation because the room would have to be completely silent for the robot to not be interfered with. With a room of about 40 people, complete silence is not always possible.

When it came to teamwork for this project we did very well. We divided up the tasks evenly and worked well together. Those of us who worked on the design of the robot discussed what needed to be changed in the code to make the robot work properly. This same communication was used by the programmers to express what needed to be changed in the design. We split up the work so two could program while the other two could build. This same philosophy was used when it came to testing and writing up the report. The only thing that could be changed would be following the timeline a little bit better. We cut a couple of our deadlines a little bit close. This was not due to any lack of communication or poor work. This was mostly due to laziness and procrastination. But this is the case for many groups.

Fixing the problems, we ran into was difficult but once we got started, we were able to fix the majority of the issues with the robot. Tweaking parts of code or moving

pieces around was a big part of solving the majority of the problems we encountered. The bigger problems we had sometimes needed us to start completely fresh and take a whole new approach.

#### Conclusion:

In this project, one important thing we have learned is the way to solve problems- considering t problems in different ways. When the robot went wrong, it might be a designing problem or a programming problem. We had to consider the problem in different ways and in different situations. Sometimes we fixed the code or change the robot design. In the designing part, we should build a robot which can move on the track stably, keep the balance from the beginning to the end, knock down all red blocks, ignore all green blocks and successfully stop in front of the bumper. And we had to try our best to keep the total cost of the robot as low as possible. We changed our design a lot of times. With so many requirements and limits, our final design could work well on the track and finish all the mandatory jobs. The robot design and code program were not individual part. They are always stuck together. Sometimes the robot design needed to change along with the new code or the code changed because the new robot design.

Another crucial thing we have learned is the teamwork. In our team, all four people concentrated on accomplishing this project with 100% effort. We met at least two times a week. We had really good communication with each other. We talked about our own opinions and found out the best one. When the problem was not very easy solve, someone of us got frustrated and the rest of the teammates always encouraged him/her to keep the faith, then concentrated on solving the problem. We worked as a real team.

Thanks to this final project for teaching us the correct way to solve problems and the necessary teamwork. We really enjoy this project.

The table below shows the cost of the robot as a whole. The part of the table at the bottom is the cost of the parts that it took for us to have the extra credit clap on/ clap off aspect of the robot.

| Piece | Cost | Number | Total cost |
|-------|------|--------|------------|
|-------|------|--------|------------|

|                        |        |    |         |
|------------------------|--------|----|---------|
| Small connector pieces | \$0.10 | 26 | \$2.60  |
| Pieces 1" or less      | \$0.25 | 29 | \$7.25  |
| Pieces > 1" long       | \$0.50 | 23 | \$11.50 |
| Wheels                 | \$0.50 | 2  | \$1.00  |
| Tracks                 | \$1.50 | 0  | \$0     |
| Motors                 | \$3.00 | 2  | \$6.00  |
| Sensors                | \$3.00 | 3  | \$9.00  |
| Gears                  | \$0.25 | 0  | \$0     |
| Sheet of paper         | \$1.00 | 0  | \$0     |
| Plastic balls          | \$0.50 | 0  | \$0     |
| Misc Molded Pieces     | \$0.75 | 0  | \$0     |
| The entire robot       |        |    | \$37.55 |
| Extra Credit Parts     |        |    |         |
| Small Connector Pieces | \$0.10 | 2  | \$0.20  |
| Sensor                 | \$3.00 | 1  | \$3.00  |
| Total E.C. Parts       |        |    | \$3.20  |