

I. **Design Brief**

To complete the competition, we split our design into two categories: the lift and the drive train. Our lift mechanism had changed over the days from a scissor lift to a simple rotating arm then back to a scissor lift and finally back to a modified rotating arm. The drive train was kept simple from the beginning. It comprised of three wheels, two of which were powered by motors and then another swivel wheel that remained without power. The actual robot design changed from an REM to a modified REM that would include a place for a light sensor. The final design was an original creation that was simpler and allowed the NXT brain to stand up vertical. The main reason that we modified the design is because we wanted the lift motor to attach to the top of the NXT brain. The design still included the swivel wheel from the REM design and the inclusion of this wheel allowed for stability of the robot. The message holder was a part of the lift system that seemed to pose a challenge because, with our design, the holder could not be too heavy or else the motor would not be able to lift the long arm. At first, we tried to create a basket, but soon realized that this would be way too heavy and cumbersome. The stability of our robot was also important. Our final holder mechanism involved the use of rubber bands to clamp down on the messages. This design was light, efficient, and wieldy. Now, for the driving aspect, we were first determined to keep the design simple by sticking to dead reckoning. Our first few attempts seemed to work as the robot made it out the door without any problem. The real issue arose when we were going down the hall way. When our robot was told to drive in a straight line, it always seemed to sway towards one direction (usually right). With the robots direction in question, we would not be able to assure precision. Since our robot could make it out of the door without any

problems and this seemed to be the fastest method for reaching the hall, we decided to combine dead reckoning with following the line via light sensor. So basically, our plan involved using dead reckoning to go forward and then turn left and drive forward outside of the door. Then, we set the robot to follow the left side of the line using the light sensor until it reached its first destination (Mrs. McGinnis's office). For the first few runs, this plan worked well, but the varying battery power had drastic effects on the robots ability to complete the dead reckoning. Also, the starting position seemed to play a crucial role in the robots success. A small angle change in the placement of the robot resulted in a huge difference of the robots position after the dead reckoning which then affected the line following tremendously. In the end, we stuck to using the line following throughout the entire track which reduced our dependency on the starting position and made our robot more precise overall. Our final robot design seemed to be the simplest design of all of our attempts. It included an original drive train with a light sensor mounted and a touch sensor to act as a button acknowledging receipt of the message. The robot followed the line the entire time except for when it turned from either destination. The lift mechanism was a simple arm that rotated on a motor to deliver the message. The motor turned 69° to bring the arm up to a safe and high point where the recipient would be able to reach and obtain the message. The holder, as previously described, consisted of a rubber band operated clamp that would grasp the message.

II. Sources

The first part of the program uses the light sensor to judge whether the line is light or dark. The input is the light that is reflected from the ground. The light sensor interprets the light reflectance into a frequency and then assigns it a value from 0 – 100. This value

is then sent to the NXT brain. The threshold is set at 35 so the NXT judges whether this value is greater or less than 35. If it is greater than 35, this would signify that the surface is light which means that the robot is off of the line. The out put to this scenario would be for the robot to turn right (the left motor is activated at a power greater than that of the right motor). If the light sensor reading is less than 35, this would signify that the surface is dark which means that the robot is on the line. The output to this scenario would be for the robot to turn left (the right motor is activated at a power greater than that of the left motor). The switch function is then placed in a loop that continues to occur until the light sensor picks up a value greater than 57 which corresponds to the threshold of a blank white sheet of paper where the robot is supposed to stop and the white tiles on the ground. The end of the loop has the same inputs as the previous light sensor program, but the threshold is set at 57 and the output is different as it involves ending the loop. The next part of the program pertains to the message delivery. This part includes lifting the arm, repeating a sound “good morning”, waiting for the button to be pressed, lowering the arm, and turning left. The program itself serves as the input for the arm lift as the 69° rotation is specified in the motor block. The output is, quite simply, the robot’s arm rotates 69° which elevates the message in hand. The program also serves as the input for playing the sound file “good morning”. The sound repeatedly plays (the output) until the button connected to the touch sensor. The input for the lowering of the arm is the pressing of the touch sensor. Once the touch sensor is pressed, a signal is sent to the NXT brain which then interprets this signal. In the program, the signal allows the robot to commence with the program as the touch sensor serves as the parameter for a wait block. After the arm has lowered, the robot turns to its left. The input for this action exists in the

program as the right wheel is set to move forward while the left wheel moves backward. The parameter for this turn is 300 degrees for both wheels in their individual directions. The output for the turn is, quite simply, the turn itself which results in the robot facing the opposite direction outside of the doorway of destination 1. The next input also comes from the program as the robot gradually turns left for 1.5 rotations. The program's next function involves going from destination 1 to destination 2 (Mrs. Banks' office). The line following program is used to accomplish this task. The threshold for this line tracking portion is still set at 35. The main difference between the first line tracking code and the second line tracking code is the output. On its way back, the robot is following the right side of the line (the same side it was following on the way to destination 1, but in the opposite direction). So, the path changes as if the NXT brain interprets the line to be dark, the robot turns right. If the NXT brain interprets the line to be light, the robot turns left. The same loop function exists around this line following code so the light sensor reading for the loop to stop is still set at any value greater than 57. Once the robot reaches its destination 2 and interprets a light reading greater than 57 (the white paper outside destination 2), the loop function ends. The next command set by the program is for the robot to stop. Therefore, the input exists within the program and the output involves the robot stopping. The next part of the program retains exactly the same code as the code at destination 1. This includes the raising of the arm, the playing of the sound file, the waiting for a response from the touch sensor, the lowering of the arm, and the 300 degree rotation of the wheel for the robot to turn. The only difference is that the robot turns right instead of left. Therefore, the input for this turn still exists in the program, but the path involves rotating the left motor forward and the right motor backwards. The robot then

turns right for 1.5 rotations, again specified in the program. After this, the robot begins its third line tracking code which is exactly the same as the first line tracking code. The loop function is also unchanged. Once the light sensor reports a value greater than 57, the robot stop and should be inside the starting box. This completes the program for the mail design bot competition.

III. Picture

- A.** NXT Brick, or NXT Brain. Houses and runs the given program. Also gives power to the motors and sensors plugged into it.
- B.** Touch Sensor. Sensor that responds to touch. Reacts when the orange piece on the front center of the sensor is pressed in, or when pressure is taken off. The touch sensor on our robot is supposed to be pressed when the note is taken. Once the sensor is pressed the program continues to travel to the next destination. The touch sensor is held onto the robot with a 5 holed Lego brick and 2 small black pegs.
- C.** Small Wheel. The purpose of the wheel is not for movement, but merely to extend the reach of the touch sensor. The wheel also helps make the touch sensor more noticeable and thus easier to recognize. Although it is a wheel, it has no movement what so ever.
- D.** Cord. Used to connect the sensors or motors to the NXT brick. The cord allows for information to be sent to or received by the sensor and the NXT brick. The cord also serves to power to motors or sensors. The cords come in varied lengths, but since all of the motors or sensors are close enough to the brick, we didn't use particular cord lengths, just which ever we took first.

- E. Wheel Motors.** These two motors are on either side of the NXT brick. They control the two front wheels which are used for movement and steering. The motors are plugged into Ports B and C. The orange circles touching the wheels spin counter clockwise or clockwise, depending on the program. Using these in combination can move the robot forward, backwards or to turn.
- F. Wheels.** The two big wheels are responsible for moving the robot forward and backwards and such. The wheels themselves can move forwards or backwards in accordance with the motor it is attached to. The wheels can navigate by moving forwards or backwards with a certain speed. If both wheels are moving uniformly (same direction and same speed), then the robot will move forwards or backwards. If the wheels are moving in different directions, or at different speeds, then a turning motion will be created.
- G. Light Sensor.** Emits a red light and uses the reflection of the light to measure a certain threshold of light. Plugs into the bottom of the NXT brick and uses program blocks in a given program to function. We used the light sensor to track the black duct tape line along the floor to get from destination to destination.
- H. Lift.** The lift is our lifting device, used to raise the given message a minimum of 6 inches from its starting point. The lift is made up of the following-
1. A motor, plugged into Port A. The motor rotates 70 degrees clockwise to raise the left, then the reverse to drop it.
 2. 4 Grey Joint Pieces (See L) Two of the pieces keep the lift from being over extended, the other two hold the black bars in place that hold the original two joint pieces in place

3. 4 medium/long black rods. Two of the rods hold the joint pieces in place. The other two are kept parallel to each other, one at the very top point of the lift, the other in the very front. These keep the lift together better.
 4. 2 yellow spacers. These keep the side of the lift from getting too close, or folding in on themselves.
 5. 1 medium/short black rod. This rod goes between two of the wing shaped pieces. It is used as an anchor between the lift and the wing piece.
 6. 4 Wing shaped Lego pieces. These serve as a clamp to hold the message in place.
 7. 4 Fifteen Hole Lego Pieces. These serve to build the lift and give it its height. They are combined in two pairs using 2 small black pegs.
 8. 4 Angled Lego pieces. Two of these pieces connect the long Lego pieces to the motor. The other two form the “basket” for which the message sits in.
 9. 2 Long black pegs. Used as the anchor point between the “basket” and the “clamp”
 10. Red Rubber Band. Connects to two anchor points, creating a stronger clamp on the basket that holds down the message.
- I.** Angled Lego Piece. Connects the motors to the brick, these are the only pieces we have keeping the motors intact with the brick. On the angled end towards the wheels, we have 2 black rods that stretch across the entire robot and are used to hold the light sensor in place.
- J.** Long Black Lego Rods. The rods are used to hold the light sensor in place. They are also there to keep the robot from buckling in on itself.

- K. Lift Motor.** This motor, plugged into Port A is responsible for the raising and lowering of the lift. The motor is attached, by using two of the small black pegs, to the back of the NXT brick, as that was the easiest place to fit it. The motor rotates clockwise 70 degrees, until the lift is level, or parallel to the ground. Once the touch sensor is hit, the lift rotates counter-clockwise 70 degrees and the lift is back at its starting position, which as you can see, is lying limply over the NXT brick.
- L. Grey Joint Pieces (1 Cross Hole, 2 Circular Holes)** – The purpose of these two pieces are to make sure that the lift does not over extend. The power of the lift motor was set rather high, so these two pieces were added to stop the lift incase the motor can't stop it from flinging to fast. They worked beautifully. The two pieces were held up by two medium length black rods.
- M. Back Wheel.** The wheel on the very back of the robot is identical to the one used for the Taskbot and REM bots. The wheel is added to the back of the robot to keep it held up. The wheel is held in place so that it will swivel and not hinder the robots ability to turn. The wheel is constructed from 2 of the 4 holed angled Lego pieces, 3 short black rods, 1 medium black rod, 2 yellow spacers and a grey spacer. The wheel is held in place by the grey spacer and a small 5 holed Lego piece that is on the underside of the motors (Cannot see in picture). The swivel motor is able to spin around in a 360 degree path, as well as the wheel may spin forward or backwards. Seeing as this wheel is in no way connected to a motor, it has no effect on the propulsion of the robot. However, it will have an effect on the direction. If the robot is moving perfectly straight, but the back wheel is disturbed and forced to turn sideways, then the robot will turn slightly until the back wheel can swivel back into uniformity.

IV. Debrief

Our robot was certainly successful to an extent in that it completed the competition with few problems despite its oversimplified design. The robot was able to reach both Mrs. McGinnis's office and Mrs. Bank's office, but not without a few resets from a point. The main reason for the robot's ability to complete the competition would have to be that its design was simple with only one light sensor and one touch sensor. There was little room for error in the programming as each action was clearly specified and the most extreme cases were taken into consideration. Our main problem, however, was the excessiveness of resets which came about as a result of too sharp turns. With just one light sensor, it was difficult to properly navigate the line every time. Certain situations such as a misreading of the line due to dust or a turn that was too sharp due to sharp curves in the track made the robot continue to turn too far over the line. If the robot's turn was too sharp sometimes, the light sensor would pass over the line too quickly and could not force the robot to turn in the opposite direction fast enough. If this happened, the robot would pass over the line and continue turning in circles until it found the line again. This charade would continue to happen because even as the robot found the line, its next turn would again be too sharp and the robot would again turn in circles. To fix this problem, we decided just to reset the robot on the line as, during the practice rounds, it only happened once per run. During the actual competition, however, the robot made the same mistake several times so we had to use several resets which took away many points. One main reason that made the robot make too sharp of a turn involved the reverse and turnaround from destinations 1 and 2. If the robot did not hit the line at a safe angle, it would make the mistake of going over the line and turning in circles. To fix this problem, I would

attach another light sensor, but not use the normal programming: keeping the light sensors off of the line creating a sort of rocking motion. Instead, I would still use the basic program that we used (having one light sensor go back and forth across the line), but the additional light sensor would serve as a backup for when the robot happened to stray over the line. I think by adding this extra light sensor to the program, the mistakes could be minimized and the amount of resets would lessen. Other than this addition, I think that the simplicity of our design worked very well as there were no other problems during the competition for our robot. The key aspects that made our robot successful were the simplicity of design in the lift mechanism as all that was required was the rotation of a motor which would, in turn, elevate the message. The drive train functioned somewhat properly except for the sporadic tendency of the robot to not change direction fast enough. Lastly, the human interface worked properly, again, for its simplicity in design as all that was required was for the button to be pressed. Overall, the simplicity, speed, and functionality of our robot made it successful for this competition.

During this competition, I learned that first and foremost, simplicity in design works the best in building anything. With our robot, we started off complicated with a scissor lift, REM design, and dead reckoning modified robot, but in the end, our design was simplified down tremendously. Another thing I learned was the light sensor's susceptibility to its setting. Factors such as ambient light, dust on the track, and changed conditions all affected the robot's functionality. With the programming, I learned how to use switch blocks to create a line tracking code that is made effective for this competition by being encompassed in a loop that is set to end at a high light reading. In terms of the drive train, I learned that dead reckoning is not a good method in making a robot follow a

predetermined course because there are so many factors that could affect the robots motion. Although the robot is expected to drive straight, it often goes against expectations and veers towards a certain direction. With the line tracking program, our robot's path is almost guaranteed. In a broader sense, I learned the basics of LEGO robotics programming with the NXT and the structural designs that worked well.

Apart from learning from our specific robot, I gained a lot of knowledge from watching other people's robots. One robot that caught my interest because of its success was group #.. It is their robot that gave me the idea that the use of two light sensors is actually better than one. During the competition the robot had a tendency to veer to one particular direction so the second light sensor was seldom used. Only on very sharp turns did the other light sensor have a job. Although this effect was unintentional, it gave me the idea that this design could actually work well. One light sensor would serve as a back up while the other had the real job of navigating the robot. One thing I would do to fix the effectiveness of their robot would be to put a long block on their touch sensor. Placing this block would increase the area for the touch sensor and would allow the robot to make fewer mistakes when stopping at the pile of books. Another change I would make would be to redesign the lift system as it was complicated and sometimes failed to deliver. I would probably use our lift design instead. Overall, this robot seemed to be very effective and had the greatest run out of all the robots. I think that this robot was successful because of its speed, unintentional reliance on just one light sensor and having the second as a back up, and ability to make the sharp turns easily.

Another robot that was interesting was group #. This robot worked decently well although it was kind of slow. I think that the robot would still be able to function properly

with a faster speed. The problem with the robot arose when it reached Mrs. McGinnis's office and had to restart the program. The issue, I believe, is the fact that this robot had to rely on time to get it to its first destination. The time constraint made a huge difference because the robot could not just be restarted at a point, but rather had to run from the very beginning when the program set off. To change this, I would use a different constraint such as a light sensor reading or touch sensor. In essence, the main lesson to carry away from this robot is the fact that time constraints are often incapable of being trusted and, during the worst case scenario when the program shuts off, time creates a huge issue.

The Mail Bot Design competition was interesting and taught me many things about robotics. The challenge was difficult, fun, entertaining, and, in general, surpassed by expectations.