Project: WEAVER

Victoria Johnson

Department of Computer Science

Montclair State University

CSIT-497

Robotics Independent Study

April 24, 2015

Table of Contents

[Introduction 3](#_Toc417509660)

[About Arduino System 3](#_Toc417509661)

[Project 3](#_Toc417509662)

[Task1: Robot Assembly 3](#_Toc417509663)

[Task2: Light Searching 4](#_Toc417509664)

[Task3: Route Following 5](#_Toc417509665)

[Conclusion 6](#_Toc417509666)

[Source Code 6](#_Toc417509667)

[Updated Table 14](#_Toc417509668)

Introduction:

Weaver is a light seeking rover robot that has also been implemented to follow a route that has been hard coded into his system. Named after two of the cutest robots in the history of animation, Weaver looks like WALL-E with his orange wheel treads and his blue Arduino board that matches EVA’s eyes. Of course, since Weaver is a boy robot the combined names of WALL-E and EVA, Weava, become the masculine Weaver.

This small rover robot runs on an Arduino Uno board which connects to a dual motor system that rotates the five wheel track that allows Weaver to easily more left and right. This also helped Weaver to move in a circle. Weaver runs on the Arduino software with its programs written in Java. He runs on four AA batteries and has an analog ambient light sensor to read in the light values that are in the area.

Weaver has been assembled to complete three tasks; to be a full functional rover robot that moves around the room; locate a target within the room; and follow a route. In this report you will read about how I built Weaver to accomplish these tasks and the challenges I faced to make Weaver the robot he is today.

About Arduino System:

Arduino is a programming language that is used to program microcontrollers. For this project I used the Arduino Software IDE which allows its programs to be written in Java. These programs were then uploaded onto the Arduino Uno board that came in the kit. The board itself is larger than the standard Arduino Uno boards so that the wheels can be mounted to the board. The length of the board also allows for a small bread board in the front of the board, as well as a mount for a sensor. Other than then that the board is just like another Arduino board.

Project:

## Task1: Robot Assembly

The first task I that I had to complete was assembling Weaver together. However this was kind of a challenge since most of the instructions were in Chinese. Thankfully years of Lego building helped me use my intuition in how to put the rover together. The most challenging part of this assembly was the dual motor system, because the instructions were very confusing. However even after I finished putting the gear box together I was still having problems with the dual motors, specifically with the left wheel, which for some unknown reason would not rotate. This left Weaver to move in a circle instead of simply moving forward.

To fix this problem I rebuilt the gear box at least six more times in hopes that the left wheel axel would move when the motor was on. The first thought I had on why it was not working was that one of the gears had fallen out of place. So I took it at apart and rebuilt it but the left wheel still did not turn. The next thought was that the axel itself was not staying in the proper position and that was why the wheel would not turn. Of course after I rebuilt it again the wheel did not budge. I tried greasing the wheels; greasing the gears; redoing the wiring to the dual motor gear box, and still the left wheel would not turn. Finally I came to the conclusion that the axel was not gripping the wheel well enough and so I bought myself some crazy glue and coated the axel with the glue. I shoved the axel into the wheel and waited for it to dry. Thankfully when I turned Weaver on the wheel moved and Weaver was able to move forward. It was extremely exciting to actually see my little rover move.

The implementation of Weaver was much easier to handle than the assembly of the little rover. The first action that Weaver was programmed to do was too simply move at the motors top speed, which can be seen in *Full Speed*, under Source Code. Here the code *Full Speed* has Weaver’s motor move in a straight line at its maximum speed. In the setup function the for loop initiates that the code will be using the four blue pine whole that are connected to the motor. In the loop function of the code is where the information for the motors to move are pushed to through the pins.

After mastering the art of moving, Weaver was programmed to move forward, backward, left and right with the press of the “w”, “s”, “a”, “d” keys on a laptop that was transmitted to Weaver through a HyperTerminal application. From the code *Keyboard Control*, by using a switch loop Weaver changes directions each time a new key command is pressed.

## Task2: Light Searching

The next task that Weaver had to complete was target location and acquisition, which first required a bit of research. After researching what each sensor, that came in the box did, I tested the various sensors to determine which one would be the best for accomplishing this task. From this I decided that Weaver’s goal would be to find the brightest light source in the room. However before I even began programming how Weaver would move to search for the light, I had to figure out how the light sensor would recognize how Weaver would stop when the light source was found.

From there I wrote a simple code for the light sensor to print out the light values it was reading in, in the Serial Window, which can be seen in *Sensor* in the Source Code. In this code, once a *LightValue* is found that is less than the *RoomValue* then the temperature and light values are not printed on the screen. Of course if the *LightValue* is greater than or equal to the *RoomValue* than the values are printed in the Serial window. After figuring out who the sensor would work I was able to move onto how Weaver would move and search for the light.

My first thought was that Weaver would move in a square movement that would get larger until he found the light; in which at this point he would stop. However the problem with the square search was that Weaver would not stop once he found the light. This was because he usually found the light while he was still in the if statement to complete a square and would not recheck the *LightValue* as he completed the square. Then after some tweaking of the square code I came to believe that having Weaver move in circles would be a better solution.

The reason I did not choose for him to move in a spiral in the beginning was because I did not think that the motors could handle Weaver rotating at long lengths. Unfortunately my reasoning was not too far off, since the motors have a hard time moving in circles for long periods of times. Honestly there were a few times when I thought I had killed the motors after testing the *Find the Light* code for a good five minutes or so. At this point I am not sure how I would mix the motors so that they could move in a spiral like manner easily and without seeming to die out. However the motors worked a lot longer than I believed they would, which made the spiral search method a success.

The reason the spiral method worked so well was because each time Weaver moved a little the *LightValue* would be recheck as the loop function was called again and again. Where are in when Weaver had to move in a square there was no to recheck the *LightValue* to complete a square pattern. The spiral method also did not relay on a for loop to move in a circle but instead depended on a radius that was grow as he looked for the light. Once the light was found the radius would be reset and Weaver would stop.

## Task3: Route Following

The last task that Weaver had to be developed for, was route following or maze. For this task I wanted to use the Ultrasonic Sensor to have Weaver move through a maze of objects that had up walls and for him to avoid the objects and get out of the maze. Unfortunately I had to change my goal for this task due to many problems that occurred. The first problem that I faced was that I needed to go buy an Ultrasonic Sensor. Without this sensor Weaver would not be able to avoid objects and follow a route without crashing into walls. Though the solution to the problem was a simply one; go buy a Ultrasonic Sensor at RadioShack, with the bizarre winter weather we had this semester and my own health issues it became a bigger set back. Eventually I finally made it to RadioShack and bought the sensor, but that was not the end of my problems.

After getting the Ultrasonic sensor I had issues connecting the sensor. For some reason the wire I had to connect the Sensor to Weaver was not soldering together. My dad suggested that the wires would not solder to the sensor because the wires I was using were made of copper and I was trying to solder them onto tin connectors. He said this would not work because the metals had two different heating points which would explain why the solder would ball up every time I tried to connect the two. So I had to come up with a different idea as to how Weaver would follow a route. The solution was a simply one, I simply would hard wire Weaver a path and he would execute it.

Though hard coding the route for Weaver to follow sounds like an easy task, it took a while to figure out how I wanted Weaver to move. This was hard because once I figured out what the path I wanted him to follow looked like; I had to use geometric conversions to convert my path into Arduino code.

This code was also very different from the other programs I wrote for Weaver, because the code did not have to use a loop since the path only needed to be executed once. As seen in the source code, *Path Following* the route that Weaver follows is in the septup() function so that he only follows the route encoded once. From this task I have learned from this program that even though the loop() function is unnecessary for this particular task, the complier will not compile the code without the loop() function written into the program. So even though it is an empty function it must be written in.

Conclusion:

From this project I have learned more about the Arduino system. For one thing I learned how to use the Serial Monitor, which was very important in learning about how each sensor worked, that came in the box. This was also the first time I had ever worked with motors and it was quite a learning experience learning how to make Weaver move to the right or move in a circle. Previously my experience with Arduinos was creating a light display which was fun but could never compare to being able to build a fully functional robot, which is an achievement that I will always cherish.

I also learned about the anatomy of the little rover, and honestly I can probably build a gear box in less than five minutes after the numerous times I had to rebuild this one. Though rebuilding the gear box was frustrating and tedious, it was worth it in the long run. It was worth it because I now have a better understanding of the physics behind the gears. For the wheels to rotate the motors need to move at a certain speed so that it will have enough torque to move them. However this can be strenuous on the motors and so the gears help create a great torque so that the motors do not have to do as much work to get the wheels to rotate at the desired speed. This was something I needed in my physics class but from this project I was able to understand the concept better.

Though I had faced some challenges during this project, I was able to overcome them and make a rover robot that was fully functional. This functionality consists of Weaver being able to move around the room; being able to locate the brightest light source in the area; as well as following a route. Weaver has been a great project to work on and has honestly inspired me to pursue a career in robotics.

Source Code:

Arduino Code for “Weaver” the rover:

1. Test if Weaver Works:

*Full Speed:*

int E1 = 6; //M1 Speed Control

int E2 = 5; //M2 Speed Control

int M1 = 8; //M1 Direction Control

int M2 = 7; //M2 Direction Control

void setup()

{

int i;

for(i=5;i<=8;i++)

pinMode(i, OUTPUT);

Serial.begin(9600);

}

void loop()

{

int leftspeed = 255; //255 is maximum speed

int rightspeed = 255;

analogWrite (E1,255);

digitalWrite(M1,LOW);

analogWrite (E2,255);

digitalWrite(M2,LOW);

delay(100);

}

2. Test Weaver Moving skills:

*Keyboard Control:*

int E1 = 6; //M1 Speed Control

int E2 = 5; //M2 Speed Control

int M1 = 8; //M1 Direction Control

int M2 = 7; //M2 Direction Control

void setup()

{

int i;

for(i=5;i<=8;i++)

pinMode(i, OUTPUT);

Serial.begin(9600);

}

void loop()

{

int leftspeed = 255; //255 is maximum speed

int rightspeed = 255;

analogWrite (E1,255);

digitalWrite(M1,LOW);

analogWrite (E2,255);

digitalWrite(M2,LOW);

delay(100);

}

3. Testing the sensor

*Sensors*

/\* To read the onboard light and temperature sensors, run the code below, then open

the serial window at 9600 baud. Ensure the two jumpers are in place\*/

int LightValue = 0;

int TemperatureValue = 0;

int RoomValue = 0;

void setup() {

RoomValue = analogRead(A0);

Serial.begin(9600);

}

void loop() {

LightValue = analogRead(A0);

TemperatureValue = analogRead(A1);

if (LightValue >= RoomValue)

{

Serial.print("Light: ");

Serial.print(LightValue);

Serial.print(" Temperature: ");

Serial.println(TemperatureValue);

}

delay(100);

}

4. Testing Moving in a square

*Moving in Square*

int E1 = 6; //M1 Speed Control

int E2 = 5; //M2 Speed Control

int M1 = 8; //M1 Direction Control

int M2 = 7; //M2 Direction Control

void setup()

{

int i;

for(i=5;i<=8;i++)

pinMode(i, OUTPUT);

Serial.begin(9600);

}

void loop()

{

int leftspeed = 255; //255 is maximum speed

int rightspeed = 255;

for(int i = 500; i<= 100000; i\*2)

{

forward(255,255);

delay(i);

stop();

delay(500);

right(255,-255);

delay(1600);

forward(255,255);

delay(i);

stop();

delay(500);

right(255,-255);

delay(1600);

}

}

void stop(void) //Stop

{

digitalWrite(E1,LOW);

digitalWrite(E2,LOW);

}

void forward(char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void reverse (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

void left (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void right (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

5. Searching for the Light

*Find the Light*

/\* Weaver will use the onboard light sensor to find the greatest light source in the room then stop.\*/

int LightValue = 0;

int RoomValue = 0;

int radius = 0;

int E1 = 6; //M1 Speed Control

int E2 = 5; //M2 Speed Control

int M1 = 8; //M1 Direction Control

int M2 = 7; //M2 Direction Control

void setup(){

int i;

for (i = 5; i<=8; i++)

{

pinMode(i, OUTPUT);

}

Serial.begin(9600);

RoomValue = analogRead(A0);

}

void loop(){

//int leftspeed = 255;

//int rightspeed = 255;

LightValue = analogRead(A0);

if((LightValue + 10) < RoomValue)

{

stop();

radius = 0;

}

else{

forward(255,255);

delay(radius);

right(255,-255);

delay(100);

radius++;

}

}

void stop(void) //Stop

{

digitalWrite(E1,LOW);

digitalWrite(E2,LOW);

}

void forward(char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void reverse (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

void left (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void right (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

6. Following the Route

*Path Following*

int E1 = 6; //M1 Speed Control

int E2 = 5; //M2 Speed Control

int M1 = 8; //M1 Direction Control

int M2 = 7; //M2 Direction Control

void setup()

{

int i;

for(i=5;i<=8;i++)

pinMode(i, OUTPUT);

Serial.begin(9600);

//Route

int leftspeed = 180; //255 is maximum speed

int rightspeed = 180;

forward(leftspeed,rightspeed);

delay(1600);

right(leftspeed,rightspeed);

delay(1000);

forward(leftspeed,rightspeed);

delay(1600);

left(leftspeed,rightspeed);

delay(1000);

forward(leftspeed,rightspeed);

delay(3200);

reverse(leftspeed,rightspeed);

delay(1600);

left(leftspeed,rightspeed);

delay(1000);

forward(leftspeed,rightspeed);

delay(2000);

stop();

}

void loop()

{

}

void stop(void) //Stop

{

digitalWrite(E1,LOW);

digitalWrite(E2,LOW);

}

void forward(char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void reverse (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

void left (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,HIGH);

analogWrite (E2,b);

digitalWrite(M2,LOW);

}

void right (char a,char b)

{

analogWrite (E1,a);

digitalWrite(M1,LOW);

analogWrite (E2,b);

digitalWrite(M2,HIGH);

}

|  |  |
| --- | --- |
| Updated Table | |
| Initialized Document | 25 February 2015 |
| Finding the Light | 31 March 2015 |
| Route Following | 22 April 2015 |
|  |  |