Line Follower Project

Irving Becerril

December 14, 2021

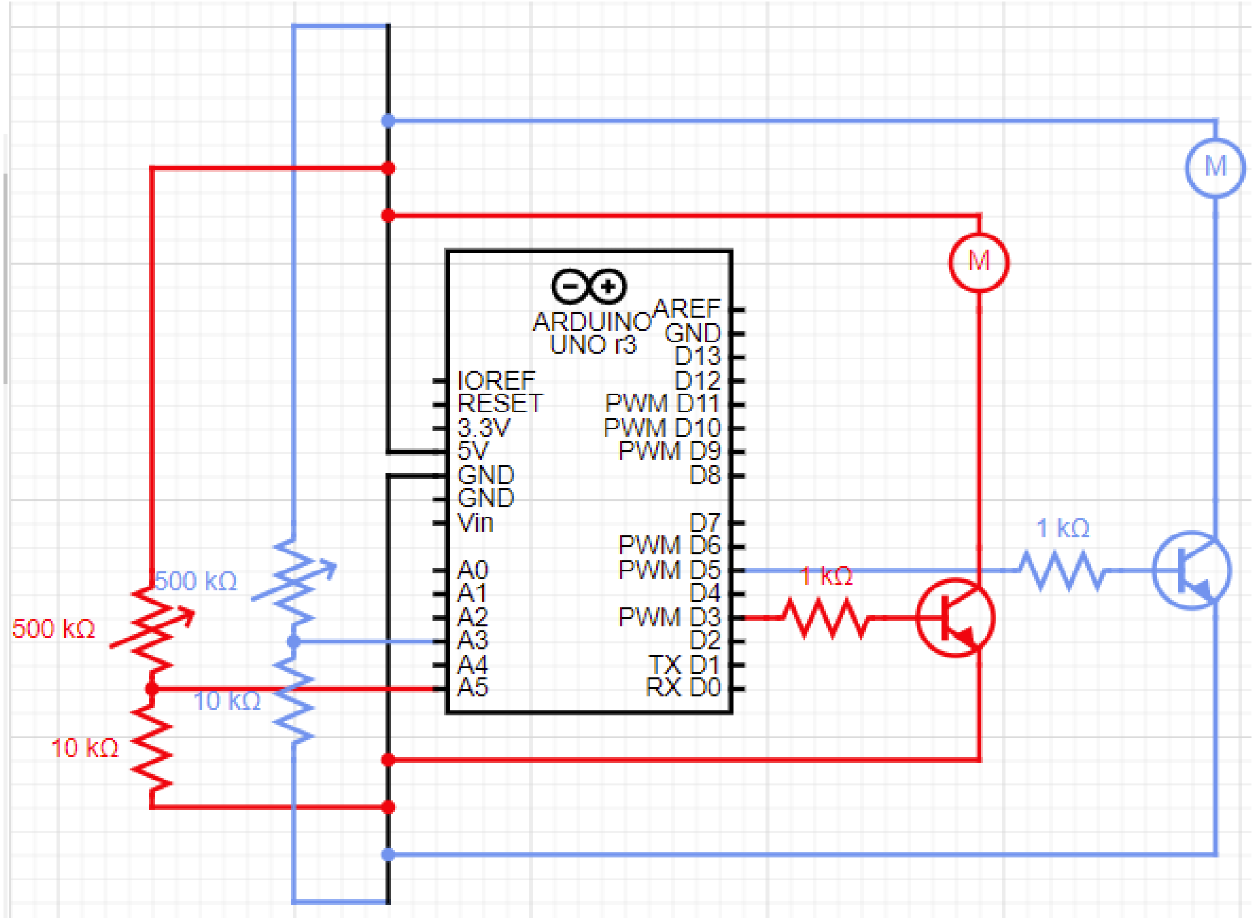
Partner: Nicholas Nebel Proffessor Braunstein

SECTION 1

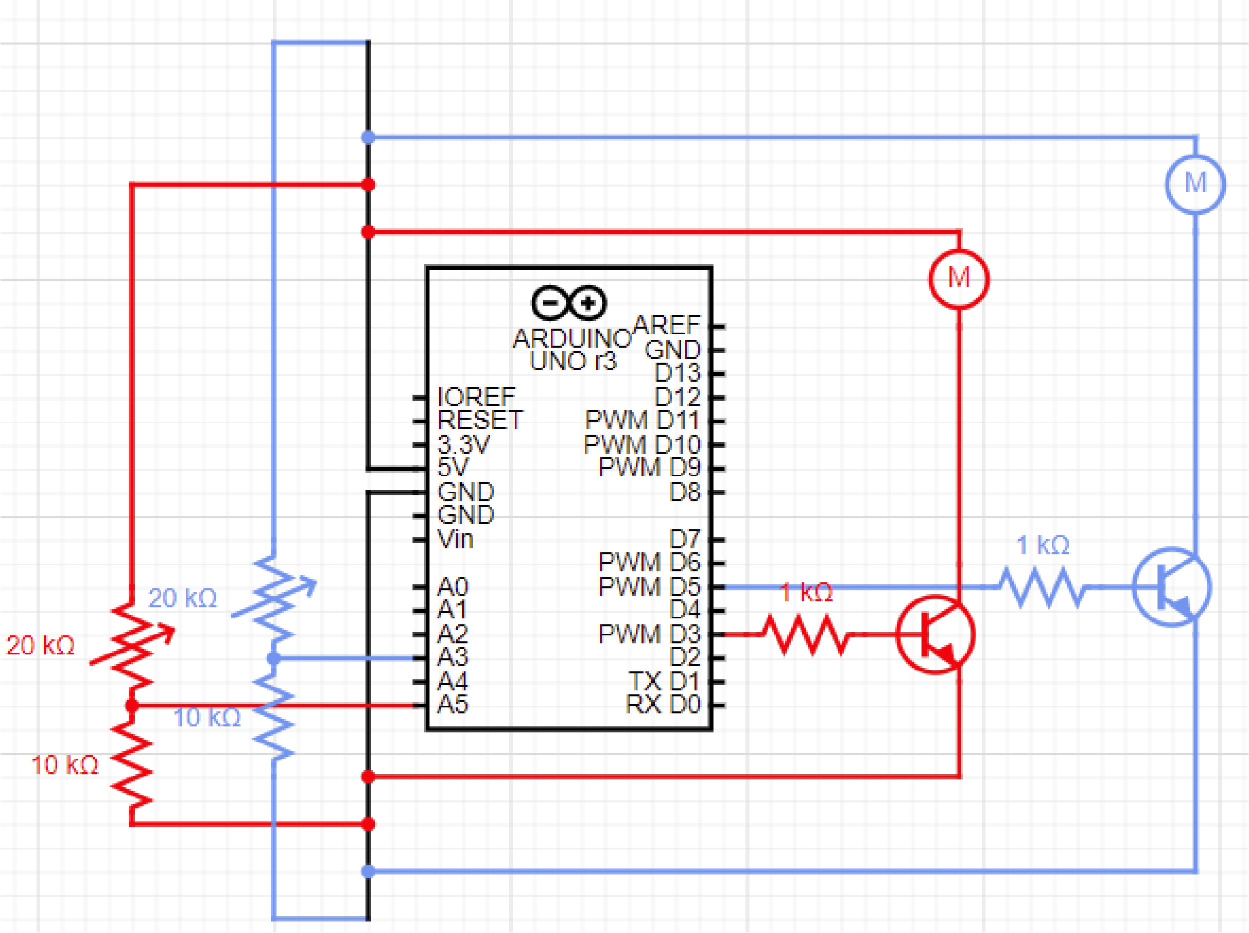
For this final project, we were tasked with building a robot that would follow a black line. We used a chasis which acted as the main body, two motors for each wheel, a breadboard for the cicuitry, and an arduino in which code that we made was put into. Our final product was a success completeing both the regular track and the extra credit track which was a loop. There were of course a few problems when building the robot such as one of the wheels not turning, the robot not being able to read the light levels and it going to fast to stay on the track. We implemented code into the arduino which would give it directions to follow the black line. The black line reflects less light than the table it was on so we made it when wasn't bright enough it would stop and when it was, it would go. We perfomed several tests in order to find the right light levels so it would either stop or go. We also had to find the right amount of time to let each part of the loop in the code to run so we had to go through several trials on that one as well. After all was done, the robot was running perfectly. The light levels were just right so it would stay on track, which it did, it wasn't going too fast as it would've otherwise gone off track, which it didn't, and both wheels were finally running again.

SECTION 2

This schematic shows when the light is dim on both photocells and the line follower isn't running.



This schematic shows when the light is bright on both photocells and the line follower is running.



SECTION 3

The circuits above were the ones used for our line-folling robot, one just shows when the motors aren't running while the other one shows when they are. That circuit itself is made up of two smaller ones which have different jobs. The one located on the right-hand side dictates whether or not the motors are turned off or on. The transistors specifically are the ones which will or won't allow the current to flow. The circuits on the left-hand side reads the light levels and also determines if the motors run or not. The photocell, which helps make up the voltage divider circuit, has a resistance which can alter based on the light levels. When it's bright, its around 500kΩ and when its dim it's around 20kΩ. The 10kΩ resistor, along side the photecell, helps make up the voltage divider circuit which goes into one of the analog ports which reads it as Vout which is the light level that is needed to either have that motors on or off.

SECTION 4

```
Given variable analogPin which equals 5

Given variable analogPin2 which equals 3

Given variable pin which equals 3

Given variable pin2 which equals 5

Given variable raw which equals 0

Given variable raw2 which equals 0

Given variable Vin which equals 5

Given variable Vout which equals 0

Given variable Vout2 which equals 0

Given variable threshold which equals 3.85

Given variable threshold2 which equals 3.0

Given variable buffer which equals 0


Start of initial function

{

        Start exchanging data with computer at 9600 bits per second

        Set pin as an output

        Set pin2 as an output

}


Start loop function

{

 raw is equal to whatever analogPin reads

 if raw is read

{

        buffer is equal to raw times Vin

        Vout is equal to buffer divided by 1024.0

        buffer is equal to Vin divided by Vout and then is subtracted by 1

        Print “Vout: “ to output

        Print whatever Vout is equal to to the output

}

raw2 is equal to whatever analogPin2 reads

 if raw2 is read

{

        buffer is equal to raw2 times Vin
        Vout2 is equal to buffer divided by 1024.0
        buffer is equal to Vin divided by Vout2 and then is subtracted by 1
        Print “Vout2: “ to output
        Print whatever Vout is equal to to the output
}
}

 if Vout is greater than then threshold2
{
        a high value is given to digital port pin 3
        keep it running for 50 milliseconds
        a low value is given to digital pin port 3
        keep it running for 100 milliseconds
}
 Otherwise
{
        a low value is given to digital port pin 3
}

if Vout2 is greater than then threshold
{
        a high value is given to digital port pin 5
        keep it running for 50 milliseconds
        a low value is given to digital pin port 5
        keep it running for 100 milliseconds
}
 Otherwise
{
        a low value is given to digital port pin 5
}
}
```

SECTION 5

```
int analogPin= 5;
int analogPin2= 3;
int pin=3;
int pin2=5;
int raw= 0;
int raw2=0;
int Vin= 5;
float Vout= 0;
float Vout2=0;
float threshold=3.85;
float threshold2=3.0;
float buffer= 0;

// setup function
void setup()
{
 Serial.begin(9600);
 pinMode(pin,OUTPUT);
 pinMode(pin2,OUTPUT);
}

void loop()        // infinite loop function
{
 raw= analogRead(analogPin); //read the analog input
 if(raw)
 {
 buffer= raw * Vin;
 Vout= (buffer)/1024.0; //determine the voltage value
 buffer= (Vin/Vout) -1;
 Serial.print("Vout: "); //print voltage value
 Serial.println(Vout);
 }

{
 raw2= analogRead(analogPin2); //read the analog input
 if(raw2)
 {
 buffer= raw2 * Vin;
 Vout2= (buffer)/1024.0; //determine the voltage value
 buffer= (Vin/Vout2) -1;
 Serial.print("Vout2: "); //print voltage value
 Serial.println(Vout2);
 }

 }
 if(Vout>threshold2)
 {
  digitalWrite(pin,HIGH);
  delay(50);
  digitalWrite(pin,LOW);
  delay(100);

 }
 else
 {
  digitalWrite(pin,LOW);
 }

 if(Vout2>threshold)
 {
  digitalWrite(pin2,HIGH);
  delay(50);
  digitalWrite(pin2,LOW);
  delay(100);
 }
 else
 {
  digitalWrite(pin2,LOW);
 }
}
```

SECTION 6

All variables we are going to use are going to be stated and put in the beginning

This next part of the code only runs once and starts the sharing of data with the computer and the arduino.

It also sets our pins as our outputs.

This part of the code is where the infinite loop function starts and is written. It starts to read the along input and sets it to raw which then leads to our if loop function. The if loop function basically finds the voltage value and prints it to the output. This code is repeated again for the second motor.

Here we have another if loop function. It says that when our value for Vout is greater than whatever our value for threshold is, then it sends a high value to our pins for a certain amount of time and the a low value for a certain amount of time. If not then a low value is sent to the pins. Basically if Vout is greater, our motors will run on and off in a loop, and if not, then they won't run at all and this code is repeated again for the second motor.

SECTION 7

Though we had figured out how to solve the problems we had with the robot, there was a lot of testing that still needed to be done. First off, the speed of the robot. The way we originally had it was it would run all in one go, but the robot ran to fast to actually read the black line in time. We had then decided to use the 3.3V port, but that wasn't enough to turn one of the wheels so we changed it back to the 5V port. I had then seen that some of the robots the others were making were sort of shuffling along, stoping and going every few milliseconds. I had realized that if we stopped it for long enough it would manage to read the track and stay on it. The only problem then was finding the right delay to put them on. We had found that putting too high of a delay when the motor was on would make it go off track and it would also tilt since we had a portable charger on it so we kept that delay lower. The delay when it was off couldn't be too low or else the same problems would occur again due to it's momentum so we tested until we found the right amount of time so that wouldn't be a problem by increasing its delay. Secondly the light levels. We needed to find what light level would be best to used as the threshold in order for the robot to actually follow the black line. We had the photocell at a fixed height off of the table so that threshold wouldn't change. Once that was done we measured the light level the black line was giving off for each photocell, the light levels the table top was giving off, and then the light levels at the edge of the black line where it met the table top. We found that using the average between the light levels the table top gave off and the edge of the black line gave off was a good fit for the robot. We did afterwards have to change the threshold for the extra-credit because it was made with marker instead of black electric tape so it gave off a different level but we repeated the same process as before and the new threshold worked well on the straight paths but on the turns it would sometimes not stay on track and just keep going straight. To stop that we decided to increase the threshold. This would make it so that in order for either one of the wheels to stop turning, the light level doesn't need to be as low as before. It would stop at a brighter light level than before, sort of making it more sensitive to the light. This fixed the problem once we found the right light level to use after several tries on the turns. For the the regular track, our robot ran perfectly, it stayed on track and adjusted itself when the track changed directions. The only thing I would say about it was that it ran pretty slow. That was fixed once we did the extra credit and lowered the delay for when it was off to 100 so that it would run the loop faster thus making the robot run faster. There were also some instances in which the robot would still go off track but that was when either my partner or I were blocking the light source and causing a shadow on the track. We also had to move the track under a light source so it would hit the track evenly and make it easier for the robot to read the track because originally it was in a corner so one photocell would always be on some sort of shadow. I would say that if we wanted to keep improving the robot I would test for when there were shadows on the track and when the light source is coming from a different direction. We could solve these problems by lowering its threshold. Since both tracks are a good bit darker than the shadows caused by the light, lowering the threshold would make it so that it would only stop either wheel when it's darker, so like when either one of the photocells is directly in the middle of the track.