```
int leftReceiver = A6;
int middleReceiver= A5;
int rightReceiver = A4;

int leftMotor = 6;
int rightMotor = 5;

int red = 14;
int green = 15;
int blue = 16;

int leftUpperBound = 500;
int middleUpperBound = 300;
int rightUpperBound = 400;

bool stopper = true;
int check = 0;
int checkTime = 750;

int maxSpeed = 140;
int minSpeed = 60;

void setup()
{
  Serial.begin(9600);

  //motors
  pinMode(leftMotor,OUTPUT);
  pinMode(rightMotor,OUTPUT);

  //receivers
  pinMode(leftReceiver,INPUT);
  pinMode(middleReceiver,INPUT);
  pinMode(rightReceiver,INPUT);

  //LED
  pinMode(red,OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(blue,OUTPUT);
}

bool withinBounds(char direction, int val)
{
  int upperBound;
```

```
  switch (direction)
  {
    case 'm':
      upperBound = rightUpperBound;
      break;
    case 'l':
      upperBound = leftUpperBound;
      break;
    case 'r':
      upperBound = leftUpperBound;
      break;
  }

  if (val <= upperBound)
    return true;
  return false;
}

void loop()
{
  int leftVal = analogRead(leftReceiver);
  int midVal = analogRead(middleReceiver);
  int rightVal = analogRead(rightReceiver);

  // If unable to detect a path, enter an infinite loop of LED flashes

  if (!withinBounds('l', leftVal) && !withinBounds('m', midVal) && !withinBounds('r', rightVal))
  {

    // Perform a check if a path was still available if the car exceeds the boundaries of the path

    while (check < checkTime)
    {
      analogWrite(leftMotor,0);
      analogWrite(rightMotor,maxSpeed);
      check++;
    }

    for (check = 0; check < checkTime*2; check++)
    {
      analogWrite(leftMotor,maxSpeed);
      analogWrite(rightMotor,0);
```

```
    leftVal = analogRead(leftReceiver);
    midVal = analogRead(middleReceiver);
    rightVal = analogRead(rightReceiver);

    if (withinBounds('l', leftVal) || withinBounds('m', midVal) || withinBounds('r', rightVal))
    {
      stopper = false;
      break;
    }
  }

  if (stopper)      // If unable to check a path, flash the LEDs 5 times per second
  {
    analogWrite(leftMotor,0);
    analogWrite(rightMotor,0);
    for (;;)
    {
      digitalWrite(green, HIGH);
      digitalWrite(blue, HIGH);
      digitalWrite(red, HIGH);
      delay(100);
      digitalWrite(green, LOW);
      digitalWrite(blue, LOW);
      digitalWrite(red, LOW);
      delay(100);
    }
  }
  stopper = true;
}

// Navigate the path based on a precedence in directions

if (withinBounds('l', leftVal) && withinBounds('m', midVal) && withinBounds('r', rightVal))

// If all directions are detected, move forward and light the green LED

{
  digitalWrite(blue, LOW);
  digitalWrite(red, LOW);
  digitalWrite(green, HIGH);
  analogWrite(leftMotor,maxSpeed);
  analogWrite(rightMotor,maxSpeed);
}
else if (withinBounds('m', midVal))
```

```
  // If only the middle path is detected, move forward and light the green LED

  {
   digitalWrite(blue, LOW);
   digitalWrite(red, LOW);
   digitalWrite(green, HIGH);
   analogWrite(leftMotor,maxSpeed);
   analogWrite(rightMotor,maxSpeed);
  }
  else if (withinBounds('l', leftVal) || withinBounds('r', rightVal))

  // Check if either left or right paths were detected

  {
   if (withinBounds('l', leftVal))

   // Check for a left path and light blue LED

   {
    digitalWrite(green, LOW);
    digitalWrite(red, LOW);
    digitalWrite(blue, HIGH);
    analogWrite(leftMotor,minSpeed);
    analogWrite(rightMotor,maxSpeed);
   }
   else if (withinBounds('r', rightVal))

   // Check for a right path and light the red LED

   {
    digitalWrite(green, LOW);
    digitalWrite(blue, LOW);
    digitalWrite(red, HIGH);
    analogWrite(leftMotor,maxSpeed);
    analogWrite(rightMotor,minSpeed);
   }
  }
}
```