Name: Shaan Yadav
NetID: ay140
Honor Code: *I have adhered to the Duke Community Standard in completing this assignment.*

1. Code for Navigation and Sensing

```
//Pins for QTI connections on board
#define leftQTI 51
#define middleQTI 53
#define rightQTI 52

#include <Servo.h>                          // Include servo library

char val = 0; // variable to store the data from the serial port
int len = 12;
Servo servoLeft;                            // Declare left servo signal
Servo servoRight;

// Define pins for built-in RGB LED
#define redpin 45
#define greenpin 46
#define bluepin 44

#define redLED 2
#define greenLED 3
#define blueLED 4
#define yellowLED 5

int hashCount = 0;
int reds[5] = {0, 0, 255, 255, 100};
int greens[5] = {255, 0, 0, 255, 255};
int blues[5] = {255, 255, 255, 0, 0};

void setup() {
 Serial.begin(9600); //start the serial monitor so we can view the output
 Serial1.begin(9600); // connect to the serial port for the RFID reader
 Serial2.begin(9600); // initialize Xbee Tx/Rx

 servoLeft.attach(12);                      // Attach left signal to P13
 servoRight.attach(11);                      // Attach left signal to P12

 servoLeft.writeMicroseconds(1500);         // 1.5 ms stay still sig, pin 13
```

```
 servoRight.writeMicroseconds(1500);        // 1.5 ms stay still sig, pin 12

 pinMode(redpin, OUTPUT);
 pinMode(greenpin, OUTPUT);
 pinMode(bluepin, OUTPUT);

 // start with light off
 analogWrite(redpin, 255);
 analogWrite(greenpin, 255);
 analogWrite(bluepin, 255);

 pinMode(redLED, OUTPUT);
 pinMode(greenLED, OUTPUT);
 pinMode(blueLED, OUTPUT);
 pinMode(yellowLED, OUTPUT);
}


void loop() {
 int lQTI = rcTime(leftQTI);
 int mQTI = rcTime(middleQTI);
 int rQTI = rcTime(rightQTI);

 int state = 4*(lQTI < 200) + 2*(mQTI < 200) + (rQTI < 150);
 // Serial.println(state);

 switch(state) {
   // not on line
   case 7:
     servoRight.writeMicroseconds(1450);
     servoLeft.writeMicroseconds(1550);
     break;
   // right sensor --> turn right
   case 6:
     servoRight.writeMicroseconds(1550);
     servoLeft.writeMicroseconds(1550);
     delay(30);
     break;
   // middle sensor --> go forward
   case 5:
     servoRight.writeMicroseconds(1450);
     servoLeft.writeMicroseconds(1550);
```

```
    break;
  // middle + right sensor --> turn right, slight
  case 4:
    servoRight.writeMicroseconds(1550);
    servoLeft.writeMicroseconds(1550);
    delay(40);
    break;
  // left sensor --> turn left
  case 3:
    servoRight.writeMicroseconds(1450);
    servoLeft.writeMicroseconds(1450);
    delay(25);
    break;
  // left + middle sensor --> turn left, slight
  case 1:
    servoRight.writeMicroseconds(1450);
    servoLeft.writeMicroseconds(1450);
    delay(25);
    break;
  // at HASHMARK --> stop, forward
  case 0:
    servoRight.writeMicroseconds(1500);
    servoLeft.writeMicroseconds(1500);

    rfidScan();

    delay(2000);

    hashCount+=1;

    // turn light off
    analogWrite(redpin, 255);
    analogWrite(greenpin, 255);
    analogWrite(bluepin, 255);

    servoRight.writeMicroseconds(1300);
    servoLeft.writeMicroseconds(1700); // right 13 is forward, left 17 is forward

    delay(500);

    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, LOW);
```

```
    digitalWrite(blueLED, LOW);
    digitalWrite(yellowLED, LOW);

    break;
  // everything else
  default:
    break;
 }
 delay(50);
}

//Defines funtion 'rcTime' to read value from QTI sensor
// From Ch. 6 Activity 2 of Robotics with the BOE Shield for Arduino
long rcTime(int pin)
{
 pinMode(pin, OUTPUT);     // Sets pin as OUTPUT
 digitalWrite(pin, HIGH); // Pin HIGH
 delay(1);                 // Waits for 1 millisecond
 pinMode(pin, INPUT);      // Sets pin as INPUT
 digitalWrite(pin, LOW);   // Pin LOW
 long time = micros();     // Tracks starting time
 while(digitalRead(pin)); // Loops while voltage is high
 time = micros() - time;   // Calculate decay time
 return time;              // Return decay time
}


//code to use RFID Scanner
void rfidScan() {
 char rfidData[len+1] = {};
 int get_more = 1;
 int timeoutInt = 0;
 int i = 0;
  while(get_more == 1 && timeoutInt < 200){
   if(Serial1.available() > 0) {
   val = Serial1.read();
      // Handle unprintable characters
    switch(val) {
      case 0x2: break;               // start of transmission - do not save
      case 0x3: get_more = 0; break; // end of transmission - done with code
      case 0xA: break;               // line feed - do not save
      case 0xD: break;               // carriage return - do not save
```

```
      default:  rfidData[i]=val; i+=1;  break; // actual character
    }
  }
  timeoutInt += 1;
  Serial.println(timeoutInt);
 }
 Serial.println(rfidData);
}
```

2. Code for Navigation, Sensing and Transmission

```
//Pins for QTI connections on board
#define leftQTI 51
#define middleQTI 53
#define rightQTI 52

#include <Servo.h>                          // Include servo library

char val = 0; // variable to store the data from the serial port
int len = 12;
Servo servoLeft;                            // Declare left servo signal
Servo servoRight;

// Define pins for built-in RGB LED
#define redpin 45
#define greenpin 46
#define bluepin 44

#define redLED 2
#define greenLED 3
#define blueLED 4
#define yellowLED 5

int hashCount = 0;
int reds[5] = {0, 0, 255, 255, 100};
int greens[5] = {255, 0, 0, 255, 255};
int blues[5] = {255, 255, 255, 0, 0};

void setup() {
 Serial.begin(9600); //start the serial monitor so we can view the output
 Serial1.begin(9600); // connect to the serial port for the RFID reader
 Serial2.begin(9600); // initialize Xbee Tx/Rx

 servoLeft.attach(12);                      // Attach left signal to P13
 servoRight.attach(11);                      // Attach left signal to P12

 servoLeft.writeMicroseconds(1500);         // 1.5 ms stay still sig, pin 13
 servoRight.writeMicroseconds(1500);        // 1.5 ms stay still sig, pin 12

 pinMode(redpin, OUTPUT);
 pinMode(greenpin, OUTPUT);
```

```
 pinMode(bluepin, OUTPUT);

 // start with light off
 analogWrite(redpin, 255);
 analogWrite(greenpin, 255);
 analogWrite(bluepin, 255);

 pinMode(redLED, OUTPUT);
 pinMode(greenLED, OUTPUT);
 pinMode(blueLED, OUTPUT);
 pinMode(yellowLED, OUTPUT);
}



void loop() {
 int lQTI = rcTime(leftQTI);
 int mQTI = rcTime(middleQTI);
 int rQTI = rcTime(rightQTI);

 int state = 4*(lQTI < 200) + 2*(mQTI < 200) + (rQTI < 150);
 // Serial.println(state);

 switch(state) {
   // not on line
   case 7:
     servoRight.writeMicroseconds(1450);
     servoLeft.writeMicroseconds(1550);
     break;
   // right sensor --> turn right
   case 6:
     servoRight.writeMicroseconds(1550);
     servoLeft.writeMicroseconds(1550);
     delay(30);
     break;
   // middle sensor --> go forward
   case 5:
     servoRight.writeMicroseconds(1450);
     servoLeft.writeMicroseconds(1550);
     break;
   // middle + right sensor --> turn right, slight
   case 4:
     servoRight.writeMicroseconds(1550);
```

```
    servoLeft.writeMicroseconds(1550);
    delay(40);
    break;
  // left sensor --> turn left
  case 3:
    servoRight.writeMicroseconds(1450);
    servoLeft.writeMicroseconds(1450);
    delay(25);
    break;
  // left + middle sensor --> turn left, slight
  case 1:
    servoRight.writeMicroseconds(1450);
    servoLeft.writeMicroseconds(1450);
    delay(25);
    break;
  // at HASHMARK --> stop, forward
  case 0:
    servoRight.writeMicroseconds(1500);
    servoLeft.writeMicroseconds(1500);

    // turn on light
    // analogWrite(redpin, hashmarkColours[hashCount][0]);
    // analogWrite(greenpin, hashmarkColours[hashCount][1]);
    // analogWrite(bluepin, hashmarkColours[hashCount][2]);

    // analogWrite(redpin, reds[hashCount % 5]);
    // analogWrite(greenpin, greens[hashCount % 5]);
    // analogWrite(bluepin, blues[hashCount % 5]);

    // digitalWrite(redLED, HIGH); //transmit
    // digitalWrite(greenLED, HIGH); //recieve
    // digitalWrite(blueLED, HIGH); // sense
    // digitalWrite(yellowLED, HIGH); // object found

    rfidScan();

    delay(2000);

    hashCount+=1;

    // turn light off
    analogWrite(redpin, 255);
```

```
    analogWrite(greenpin, 255);
    analogWrite(bluepin, 255);

    servoRight.writeMicroseconds(1300);
    servoLeft.writeMicroseconds(1700); // right 13 is forward, left 17 is forward

    delay(500);

    digitalWrite(redLED, LOW);
    digitalWrite(greenLED, LOW);
    digitalWrite(blueLED, LOW);
    digitalWrite(yellowLED, LOW);

    break;
  // everything else
  default:
    break;
 }
 delay(50);
}

//Defines funtion 'rcTime' to read value from QTI sensor
// From Ch. 6 Activity 2 of Robotics with the BOE Shield for Arduino
long rcTime(int pin)
{
 pinMode(pin, OUTPUT);     // Sets pin as OUTPUT
 digitalWrite(pin, HIGH); // Pin HIGH
 delay(1);                 // Waits for 1 millisecond
 pinMode(pin, INPUT);      // Sets pin as INPUT
 digitalWrite(pin, LOW);  // Pin LOW
 long time = micros();     // Tracks starting time
 while(digitalRead(pin)); // Loops while voltage is high
 time = micros() - time;  // Calculate decay time
 return time;              // Return decay time
}


//code to use RFID Scanner
void rfidScan() {
 char rfidData[len+1] = {};
 int get_more = 1;
 int timeoutInt = 0;
```

```
 int i = 0;
  while(get_more == 1 && timeoutInt < 200){
   if(Serial1.available() > 0) {
   val = Serial1.read();
      // Handle unprintable characters
     switch(val) {
       case 0x2: break;               // start of transmission - do not save
       case 0x3: get_more = 0; break; // end of transmission - done with code
       case 0xA: break;               // line feed - do not save
       case 0xD: break;               // carriage return - do not save
       default:  rfidData[i]=val; i+=1;  break; // actual character
     }
   }
   timeoutInt += 1;
   Serial.println(timeoutInt);
 }
 Serial.println(rfidData);

 if (timeoutInt < 200) {
   char outgoing = rfidData[9]; // Read character
   if (outgoing == 'D') {
     xbeeTransmit(74+hashCount+1); //hashcount is the index of the hashmark bot is at
indexed starting 0.
   }
   xbeeTransmit(outgoing);
 }
}

void xbeeTransmit(char charToSend) {
 digitalWrite(redLED, HIGH); //transmit
 Serial.print(charToSend);
 Serial2.print(charToSend); // Send to XBee
}
```

3. Reflection Paragraph

Really enjoyed this lab. Struggled a bit at first with figuring out the format to transmit signals in but drawing out a plan on the whiteboard to use integer and remainder division to interpret data made everything clear. The RFID scanning part was fairly straightforward. To improve we should probably discuss what the goal for the lab is and lay out concrete steps to get there rather than just meander to the checkpoints. The manual was very clear with what it wanted us to do.