Tyler Thompson

2/4/23

tjt5498@psu.edu

**Lab 7**

**Objective**

Use sonar and alarm to allow the robot to continue moving forward until an object was detected. Determine if the object is only on right or left side so the robot can do a quick maneuver to get out of the way. If an object was detected on both sides then to robot backs up and moves out of the way and continues on the path. An alarm is signaled every time an object is detected forward.

**Results**

|  |  |
| --- | --- |
| A picture containing floor, indoor, wooden, wood  Description automatically generated | Here is the robot successfully set up with the sonar and alarm installed properly. Changes were needed in order to allow for the sonar to be fitted into place included wires needing to be remapped. The alarm also functions are normal to work with the sonar device. |

|  |  |
| --- | --- |
| Robot Test Videos | Results |
| Robot moves forward and detects an object on both sides.  <https://share.icloud.com/photos/02bs8ygQCdJ8oH3GM4ievfxpw> | The robot continues forward until it reaches the object. It is about 10 inches away from the object when it is detected. The alarm is the sounded and the robot stops. The sonar scans both sides and realizes that it can not maneuver either direction easily without hitting the object. The robot then backs up for 1 second then moves to right and continues the path. |
| Robot moves forward and detects an object on the right side.  <https://share.icloud.com/photos/034DdLKrB32Og5WuhUorWsxHA> | The robot continues forward and detects an object in front. The alarm goes off and the robot stops. The object is about 10 inches away. It then scans the sonar that realizes it can move to the left since the object is mostly on the right side. The robot then moves slightly to the left and continues forward on the path. Robot comes within 7 inches of the object when moving away. |
| Robot moves forward and detects an object on the left side.  <https://share.icloud.com/photos/017kyuZWkEMzYmhFtLMdK2rdA> | The robot continues forward and detects an object in front. The alarm than goes off and the robot stops. The object is about 10 inches away. It then scans the sonar and realizes that the object is mostly on the left side. It then moves slightly to the right and continues forward on the path. Robot comes within 7 inches of the object when moving away. |

**System Block Diagram**

Buzzer Alarm

Control Software

Arduino

Sonar Device

Motor Controller

H Bridge

Right Motor Control Front

Left Motor Control Back

Right Motor Control Front

Right Motor Control Back

**Pseudocode**

// Moves the robot forward until something is detected.

void checkForObjects() {

  head\_Move(90); // Sets sonar position to straight.

  delay(200);

  while (watch() > 20) { // Goes forward if nothing is in front of the robot.

    Go Forward // Move robot forward.

  }

  stop // Something was detected in front.

  delay(1000);

  turn on buzzer // The alarm signals that it detected an object.

  obstacle\_On\_Right = 0; // Used to keep track of object.

  obstacle\_On\_Left = 0; // Used to keep track of object.

  Head Move(130); // Sonar to left side.

  delay(200);

  if (watch() < 25) { // If something is in range set a value.

    obstacle\_On\_Left = 1;

  }

  Head\_Move(50); // Sonar to right side.

  delay(200);

  if (watch() < 25) { // If something is in range set a value.

    obstacle\_On\_Right = 1 // Set the value.

  }

  if (obstacle\_On\_Right == 1 and obstacle\_On\_Left == 0) { // Checks to see if an object was only on right.

    Go Left // Moves robot left.

    delay(300);

  }

  else if (obstacle\_On\_Right == 0 an obstacle\_On\_Left == 1) { // Checks to see if an object was only on left.

    Go Right // Moves robot right.

    delay(300);

  }

  else { // Object was detected on both sides.

    Go Back // Moves the robot back.

    delay(1000);

    Go Right // Turns the robot to the right to get out of way of object.

    delay(250);

  }

}

**Code**

// Moves the robot forward until something is detected.

void checkForObjects() {

  head.write(90); // Sets sonar position to straight.

  delay(200);

  while (watch() > 20) { // Goes forward if nothing is in front of the robot.

    go\_Advance();

  }

  stop\_Stop(); // Something was detected in front.

  delay(1000);

  alarm(); // The alarm signals that it detected an object.

  int obstacle\_On\_Right = 0;

  int obstacle\_On\_Left = 0;

  head.write(130); // Sonar to left side.

  delay(200);

  if (watch() < 25) { // If something is in range set a value.

    obstacle\_On\_Left = 1;

  }

  head.write(50); // Sonar to right side.

  delay(200);

  if (watch() < 25) { // If something is in range set a value.

    obstacle\_On\_Right = 1;

  }

  if (obstacle\_On\_Right == 1 && obstacle\_On\_Left == 0) { // Checks to see if an object was only on right.

    go\_Left(); // Moves robot left.

    delay(300);

  }

  else if (obstacle\_On\_Right == 0 && obstacle\_On\_Left == 1) { // Checks to see if an object was only on left.

    go\_Right(); // Moves robot right.

    delay(300);

  }

  else { // Object was detected on both sides.

    go\_Back(); // Moves the robot back.

    delay(1000);

    go\_Right(); // Turns the robot to the right to get out of way of object.

    delay(250);

  }

}

**Conclusion**

The robot sonar program was a complete success. After a few trails and adjusting some code the result could be duplicated many times. The code in all was not to complicated and was a slight modification of the original with some not needed functions replaced. The robot was able to maintain a safe distance away from the object while continuing with the path. This really shows the power of the sonar with the robot and the alarm is a nice added component that can come in handy when signaling an event that might have occurred. Overall the sonar on the robot will be really important to update and maintain for later projects down the line.