# **Ping Bot Notes**

Straight-line motion is controlled by PID, implemented by Brett Beauregard’s PIDLibrary. Rotation measured using encoder steps. Arduino provides 5V power to robot, and is powered by an external 9V battery. The TX payload is a four-value integer array equal to: (1) n1, the encoder count from start to the edge of the first detected wall, represented by step one in Figure 1; (2) e1, the “eastern” distance of the robot to the wall as it travels parallel and “north” to the wall, represented by Figure 1 step 2; (3) e2, the “eastern” distance from the same line of latitude as the Treasure Bot to that of the treasure wall, represented by Figure 1 CP3 to midway between steps four and five; and (4) n2, the “northerly” distance from the easterly line of travel to the treasure, represented by Figure 1 point five to the treasure location.

# **Ping Bot Synopsis**

|  |
| --- |
|  |
| **Figure 1.** Ping Bot path and ping viewing direction |

|  |  |  |
| --- | --- | --- |
| Step | Code Line | Ping Bot Action (see Fig 1) |
| 1 | 69 - 105 | * Orient encoders. * Orient PING))) to the right. * Drive forward until leading edge of wall is found. When the wall is found, measure the x-coordinate distance to the wall. Store x-coordinate distance to wall and encoder steps as variables for the transmission array. * Move forward 18 inches to CP1. * Rotate 90 degrees to align with bridge. |
| 2 | 106 - 116 | * Orient PING))) to the left. * Drive e1 distance east. |
|  | 110 - 116 | * Drive 24 inches east to CP3. |
| 4 | 118 - 125 | * Move forward * Scan for leading edge of treasure wall. |
| 5 | 127 - 138 | * Measure distance to treasure wall. * Move 24 inches to eliminate interference with Treasure Bot. |
| 6 | 140 - 144 | * Transmit distance array to Treasure Bot. |

Table1.Ping Bot synopsis

**Treasure Bot Notes**

All straight-line movement is controlled using PID control through the Arduino.

All rotations are based on selected wheel speeds and timing. This gives the programmer the ability to fine tune turn angles based on delays.

Arduino is powered through a nine volt battery. All servos and encoders run on five volts.

The treasure bot determines its route based on an array sent from the Ping bot containing four integer values (N1, N2, E1, and E2). See figure 1 for these four integer values.

**Treasure Bot Synopsis**

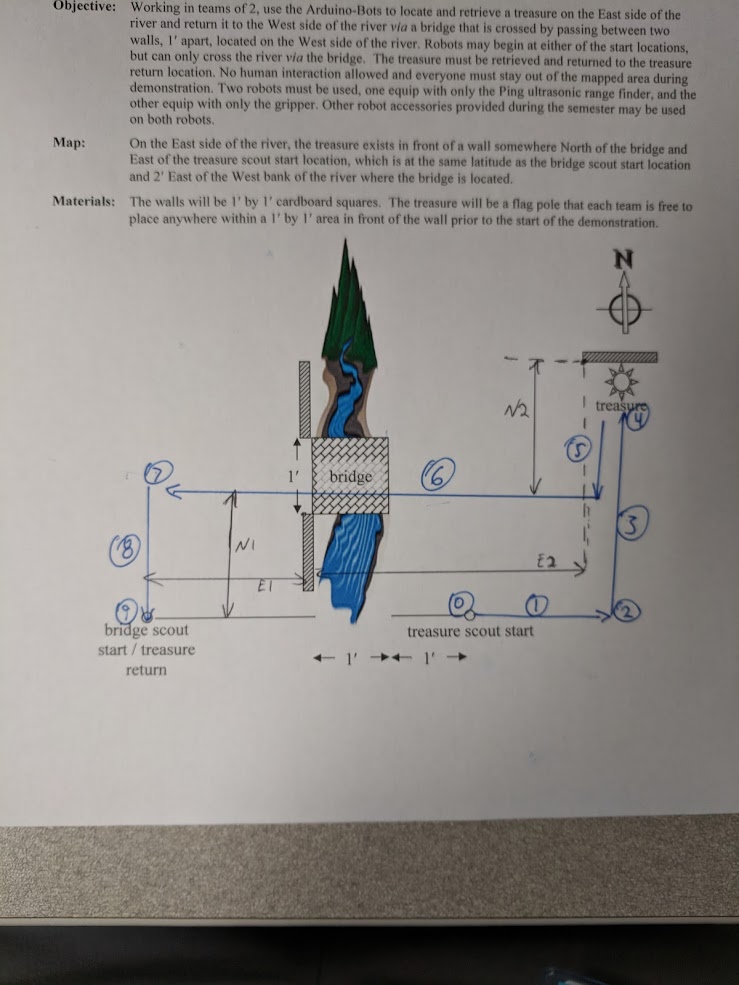
****

Figure 2. Treasure bot route and transmitted dimensions

|  |  |  |
| --- | --- | --- |
| Step | Code Line | Robot Response |
| 0 | 63-77 | Listen for array to be sent from Ping bot. Start with gripper open. |
| 1 | 85-89 | Start facing due east. Travel due east until “E2 – 2 feet” |
| 2 | 90-92 | Turn 90 degrees left |
| 3 | 94-99 | Travel due north until “N1 + N2” has been covered. Stop. |
| 4 | 101-106 | Close gripper and turn 180 degrees left |
| 5 | 108-112 | Travel due south until N2 has been covered. Turn 90 degrees right |
| 6 | 119-123 | Travel due west until “E2 + E1” has been covered |
| 7 | 124-126 | Turn 90 degrees left |
| 8 | 128-132 | Travel due south until “N1” has been covered |
| 9 | 134-138 | Open gripper |

Table 2. Treasure bot summary and corresponding code lines.

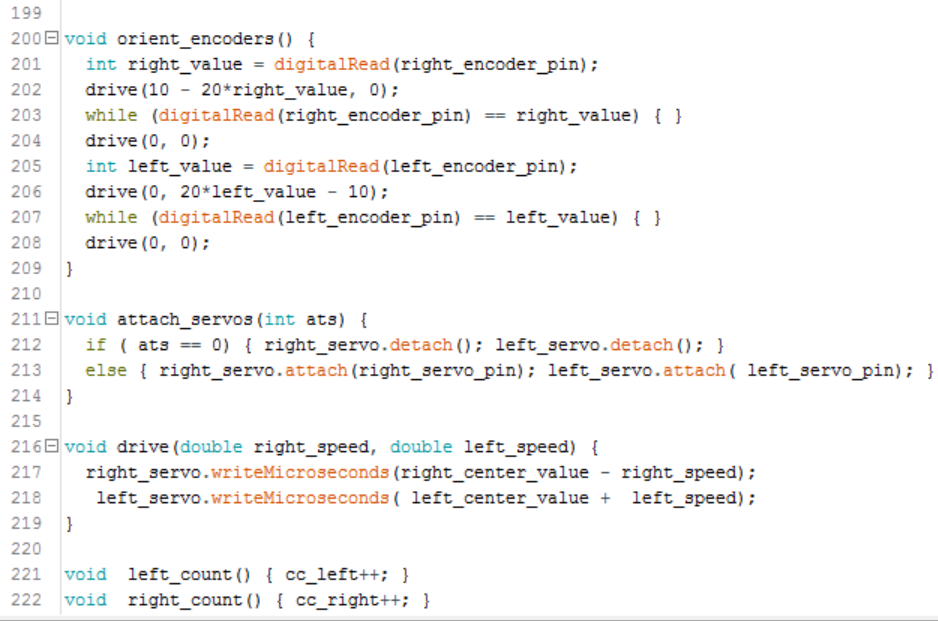
**Appendix**

Ping Bot Code









Treasure Scout Code



