Ridgesoft IntelliBrain-Bot

Matthew Coyne, Juan Ferreiro, and Matthew Schehl

Janusz Zalewski, Ph.D.

CEN 4935 Senior Software Engineering Project

Florida Gulf Coast University

Ft Myers, Florida

April 25th, 2012
Table of Contents

1. Introduction
2. Hardware
3. Software
4. Program Development Process
5. Case Studies
   Ultrasonic Range Sensor
     Lesson 1   Tractor Beam
     Lesson 2   Acquire Object
   Infrared Range Sensor
     Lesson 3   Line Follower
     Lesson 4   Line Follower with Remote Control
   Line Sensor
     Lesson 5   Wanderer
     Lesson 6   Interact
6. Conclusion and Future Work
7. References

Appendix 1.1 Lesson Plan - Lesson 1: Tractor Beam
Appendix 1.2 Code - Lesson 1: Tractor Beam
Appendix 2.1 Lesson Plan - Lesson 2: Acquire Object
Appendix 2.2 Code - Lesson 2: Acquire Object
Appendix 3.1  Lesson Plan - Lesson 3: Line Follower

Appendix 3.2  Code - Lesson 3: Line Follower

Appendix 4.1  Lesson Plan - Lesson 4: Line Follower with Remote

Appendix 4.2  Code - Lesson 4: Line Follower with Remote

Appendix 5.1  Lesson Plan - Lesson 5: Wanderer

Appendix 5.2  Code - Lesson 5: Wanderer

Appendix 6.1  Lesson Plan - Lesson 6: Interact

Appendix 6.2  Code - Lesson 6: Interact
1. **Introduction**

The Ridgesoft IntelliBrain-Bot educational robot is a small easily programmable robot designed with the original intention of teaching students beginning programming and engineering concepts (Figure 1). The focus of this project is to both explore those concepts and make them accessible to less experienced students and teachers. A textbook titled “Exploring Robotics with the IntelliBrain-Bot” [1] by RidgeSoft is used as a starting point and several small projects extend out from the concept in it.

![Figure 1](image)

**Figure: 1**
2. Hardware

Hardware required for this project includes the IntelliBrain-Bot Deluxe and a Windows, Mac OSX, or Linux computer capable of running the RidgeSoft RoboJDE software and outputting via a RS232 serial connection. A converter such as USB to Serial may be used. The IntelliBrain-Bot Deluxe is a fully featured kit for testing the various abilities of the robot. The robot normally comes assembled but in case it is not, an assembly guide [4] is available on the RidgeSoft website.

The hardware included in the package contains:

- IntelliBrain 2 robotics controller: A robotics controller featuring a 16x2 character liquid crystal display, two push buttons, a thumbwheel, a buzzer, six LEDs, and an infrared universal remote control receiver.
- IntelliBrain-Bot robot chassis: The modular nature of the controller and sensors allow this robot to be mounted onto any frame that the user wishes, but this chassis is designed for it.
- Two wheel encoder sensors: infrared sensors pointed at the wheels designed to report current wheel positions and movement.
- Two line sensors: infrared sensors designed to detect the presence of a path on the ground (visible at the bottom center of Figure 2).
- Two infrared range sensors: infrared sensors designed to report the current position of nearby objects (visible on both sides of Figure 2).
- Ultrasonic range sensor: ultrasonic sensor designed to report how far away objects are from the front of the robot (visible at the top of Figure 2).
3. **Software Installation and Operation**

   The robot is programmed in a version of Java using the RoboJDE. This is a free development environment available from the RidgeSoft website allowing for the editing, building, loading, and debugging of robot projects. Use the following procedure to install the RoboJDE software:

   1. Install the RoboJDE Java-enabled Robotics Software Development Environment from the files provided.
      
      **Note:** You can also download the RoboJDE from [http://www.ridgesoft.com/robojde/download2.htm](http://www.ridgesoft.com/robojde/download2.htm)

   2. Run the installation file (RoboJDESetup_2_0_0.exe).

   3. A welcome screen will advise you to close all of your programs before continue (Figure 3). Click next after doing so.

   ![Welcome screen](image)

   **Figure 3:** Welcome screen

   4. Accept the license agreement (Figure 4) and click next.
5. Select the folder where you would like RoboJDE to be installed, then click Next (Figure 5). Recommend it location “c:\Program Files (x86)\RoboJDE”.

6. Select the start menu folder in which you would like Setup to create the program’s shortcuts, then click Next (Figure 6) by default is RoboJDE.
7. Select file associations and then click Next (Figure 7).

8. When the installation process finishes click Finish to exit the setup (Figure 8).
9. Go to Start menu->All program->RoboJDE and click RoboJDE. At this point the Robotics Software Development Environment is ready (Figure 9).

Figure 8: Completing Installation process screen.

Figure 9: RoboJDE ready to be use.
4. **Program Development Process**

Use the following procedure to load a program on the robot.

1. Connect the robot controller to a serial port on your computer (Figure 10). In case your computer does not have a serial port, use a USB to serial adapter cable (Figure 11).

   ![Intellibrain-Bot connected to the computer using a serial cable and a USB to serial adapter](image1)

   **Figure 10:** Intellibrain-Bot connected to the computer using a serial cable and a USB to serial adapter

   ![USB to serial adapter](image2)

   **Figure 11:** USB to serial adapter

2. Run the RoboJDE GUI from the start menu.

3. Use the Tools->Settings menu item to bring up the Setting dialog box.

4. If the compiler field is empty, browse to and select the Java compiler you would like the RoboJDE GUI to use.

5. Select the controller type you are using.
6. Select the port to which you attached the robot controller (Figure 12).

7. Select the baud rate your computer should use to communicate with the robot controller. By default is 115200 baud (Figure 12).

![Figure 12: Setting Screen](image)

8. Click Ok. If another program is using the port you selected, you will receive an error message (Figure 13).

![Figure 13: Error display when Intellibrain-Bot is not connected on the port selected](image)

9. Now we will create a project. Click File and then New Project.

10. Click the Browse button (Figure 14) and select a folder to store your project.
11. Enter a lesson name in the Main class field and then click the OK button.

When using the lesson code, main class names are as follows:

   Lesson 1: TractorBeam
   Lesson 2: AcquireObject
   Lesson 3: IntelliBrainBotLineFollower
   Lesson 4: IntelliBrainBotLineFollower
   Lesson 5: wanderer
   Lesson 6: Interact

10. Double-click the source code file which is provided in the lessons. The file should open in text editor (such as notepad).

The source code files for the lessons are as follows:

   - Appendix_1.1_Lesson1_Source_Code_Tractor_Beam.txt
• Appendix_2.2_Lesson2_Source_Code_Acquire_Object.txt

• Appendix_3.2_Lesson3_Source_Code_Line_Follower.txt

• Appendix_4.2_Lesson4_Source_Code_Line_Follower_with_Remote.txt

• Appendix_5.2_Lesson5_Source_Code_Wanderer.txt

• Appendix_6.2_Lesson6_Source_Code_Interact.txt

11. Copy and paste the code from the file to the RoboJDE software (Figure 15).

12. Turn on your robotic controller, the green led will turn on and the LCD will display “Load” (Figure 14).

![Figure 14: Green led is on meaning the Intellibrain-Bot is on and the LCD displays “Load” meaning that is ready to load the program.](image-url)
At this point you can make any change to the code.

Click the build and load button (Figure 16) and when the program is completely loaded (Figure 17) the LCD screen will display “Ready” (Figure 18).
Press the button start on the Robot (Figure 19).

The robot runs the loaded program.
5. **Case Studies**

This section illustrates the programming operation of the robot for various types of sensor and actuators such as the ultrasonic sensor, line sensors and infrared sensors, wheel encoders and wheel motors.

**Ultrasonic Sensor**

The ultrasonic transmitter emits a sound wave that bounces off an object and is received by the ultrasonic sensor. The precise time between the sound wave emission or “ping” and the detection is recorded. Using the speed of sound and time traveled, the distance to the object is calculated. The following lessons demonstrate how the ultrasonic sensor can be used to determine the movement of the robot. Lesson one is designed to follow an object and lesson two demonstrates manual control of the robot's movement and engaging the Tractor Beam by remote.

**Lesson 1: Tractor Beam**

The robot will follow an object by moving toward it when the distance to the object is less than 20 inches but greater than 6 inches. The robot will move away from an object that is less than 5 inches away.

The details of the lesson are outlined in the lesson plans (Appendix 1.1).

The final source code is provided (Appendix 1.2).

A diagram (Figure 27) shows how the program flows.
Lesson 2: Acquire Object

The robot will be programmed to allow manual control with a common TV infrared remote. When the channel and volume buttons are pressed, the robot will move right, left, forward or backward. When the mute button is pressed the Tractor Beam is engaged. When the mute button is pressed again, the Tractor Beam is then disengaged and manual control is restored.

The details of the lesson are outlined in the lesson plans (Appendix 2.1).
The final source code is provided (Appendix 2.2).

A diagram (Figure 28) shows how the program flows.

*Figure: 28*

**Line Sensor and remote control**

The line sensors enable the robot to follow a line on the floor. What is needed is either a line following poster or a one inch wide stripe of a non-reflective black electrical tape on a white surface to complete the line following demonstration of this project.

**Lesson 3: Line Sensor**
This state diagram (Figure 22) shows the layout of the code. This algorithm is able to take in count the current state and previous state read by the line sensor. This way the robot can react to its current situation with six possible responses: Stop, go slightly right, go hard right, go straight, go slightly left, and go hard left.

Figure 22

The details of the lesson are outlined in the lesson plans (Appendix 3.1).

The final source code is provided (Appendix 3.2).

A basic overview of the lesson is provided below:

1. Load the program (Appendix IV) on the robot using the procedure in Section 4: Program Development Process.

2. Put down your robot on top a one inch line surround for a white surface (Figure 22), make sure both line sensors are pointing to this line, when they do the LCD will display “Centered” (Figure 23).
Figure 22: Line sensor on top of the one inch black tape and white paper.

Figure 23: Line sensors are centered

3 Press Start. The robot will start moving following the line

Lesson 4: Line Sensor with Remote Control

The details of the lesson are outlined in the lesson plans (Appendix 4.1).

The final source code is provided (Appendix 4.2).

A diagram (Figure 29) shows how the program flows.
Figure 29

Infrared Sensors
The front mounted infrared sensors transmit a beam of infrared light out from both front corners of the Intellibrain-Bot and capture reflections. This sensor can give the robot the ability to sense and avoid objects. The reflected infrared detection also senses the distance between a detected object and the robot. Lessons with this sensor will require enough space for the robot to drive around, and a wall or objects for it to avoid.

**Lesson 5: Wanderer**

The Wanderer program instructs the Intellibrain-Bot to drive forward until it senses an object with its infrared sensors. If it does sense such an object, it will then turn 45 degrees away from it. Once it no longer senses the object, it will continue driving forward.

This lesson will help students understand this simple program. This is done through reading, review, alteration, and observation. The student will read the program and make small changes, then observe what effect the changes had.

The details of the lesson are outlined in the lesson plans (Appendix 6.1).

The final source code is provided (Appendix 6.2).

A diagram (Figure 25) shows how the program flows.
Lesson 6: Interact

The Interact program allows the user to drive the Intellibrain-Bot around with a Sony television or universal remote control. Continuing the use of the infrared sensor, it can detect if it is being driven towards an object, and will start beeping at the user when it is close to an object.
This lesson extends from the Wanderer lesson. The student will complete similar exercises of exploration through reading, understanding, and testing hands-on changes in a more complicated program.

The details of the lesson are outlined in the lesson plans (Appendix 6.1).

The final source code is provided (Appendix 6.2).

A diagram (Figure 24) shows how the program flows.
Program Starts

Infrared sensors check for reflections within desired range

Did a sensor find an object?

Yes

Robot’s speaker beeps

Remote control sensor checks if the robot is being sent instructions

No

Did the robot detect a remote directional instruction?

No

Yes

Robot drives in specified direction

Figure: 24
Conclusion and Future Work

The lessons included in this report covered the basics of programming the robot to respond to data acquired from the line, infrared and sonic sensors. We also implemented remote control functionality demonstrating how the robot can be controlled manually as well as autonomously. By examining pieces of the code and changing certain variables, a better understanding of programming the robot was achieved.

Future work may include programming the speaker and LEDs of the robot to indicate the different activities the robot is performing. Lessons also could be developed that allow a student to create their own program. Code snippets that perform certain functions could be provided and the student would have to choose which ones to implement and organize them properly.
7 References


[3] VCP Drivers (for USB to serial adapter),  
http://www.ftdichip.com/Drivers/VCP.htm

[4] Robot assembling guide:  