Sparki Bluetooth Controller

SJ Davis Robotics

v1.0.0
June 2014

Summary

The Sparki Bluetooth Controller is an Android application that uses Bluetooth to control and read data from the ArcBotics Sparki robot platform (http://arcbotics.com/products/sparki/start) A companion application written specifically for the Android controller must be loaded and run on the Sparki robot in order for the system (controller and robot) to function.

The controller app aims to demonstrate the full functionality of the Sparki robot and exercises nearly all build-in functions including the control functions (stepper motors/wheels, LCD, LED, Servo, Gripper, and Buzzer) and sensor functions (Accelerometer, Magnetometer, Ultrasonic Range Finder, and Light Sensors). You can make Sparki move in any direction and command him to pick up objects with his gripper. He can even report distance measurements from objects that are in his forward or left/right field of vision with his ultrasonic range finder and servo. All measurements from his various sensors can be displayed on the controller screen in real time.

The controller app also demonstrates how to control Sparki with voice commands. A list of commands that Sparki will responds to are listed in Appendix A.

Background/Purpose

My son, who is now 7 years old, has always loved robots! As early as three years old, after seeing the world famous Honda robot, Asimo (and others), on YouTube, he asked me if we could build a robot together. Being an engineer myself, of course I said, “yes!”. Starting a bit smaller than Asimo, my son and I have been exploring and programming robots to fuel his (and my) love of technology and robots. As such, the Sparki Bluetooth Controller was designed and developed as a father/son project to explore the capabilities of the ArcBotics Sparki robot platform and remote Bluetooth control.

The Android app was developed using the MIT App Inventor 2 environment (http://appinventor.mit.edu/explore) where kids, beginning programmers, and hobbyists can quickly create working applications — with very little or no programming experience. The full source code for the Android app is available upon request and is licensed under the MIT Open Source License Agreement.

Keep in mind, this application is for educational purposes only. The app is intended to simply demonstrate remote Bluetooth communication and control of the Sparki robot. We hope that other kids and adults will enjoy exploring and learning about the Sparki platform and use our application and code to spark new ideas and learning about technology!

Basic Requirements

- 7" Android Tablet (tested on Nexus 7 with Andriod 4.4.2)
- ArcBotics Sparki Robot http://arcbotics.com/products/sparki)
- Bluetooth Module for Sparki http://www.arcbotics.mybigcommerce.com/bluetooth-module)
- PC or Mac (to install the SparkiDuino software for uploading the companion control program to Sparki)
- The companion SparkiDuino code that runs on Sparki (SJD_BT_Control.ino) available here: https://github.com/sivads/SJD_BT_Control (select Download ZIP on the righthand side of the repository window — more details below).
- You will also need patience and a love of robots!

Setting Up Sparki:

There is plenty of information and support on the ArcBotics web site (http://arcbotics.com) to get started with Sparki. You must have a PC or Mac to install the SparkiDuino software for uploading control code to Sparki.

Sparki Bluetooth Module

If the Bluetooth module was not included with your Sparki robot when purchased, you can purchase it separately from ArcBotics or wherever Sparki is sold. Please follow the instructions on the ArcBotics site to setup and test the Bluetooth module:

http://arcbotics.com/products/sparki/parts/bluetooth-module

Uploading Code to Sparki

You must be familiar with how to upload code to Sparki in order to upload the companion application for the Bluetooth Controller. Please refer to the instructions and examples on the ArcBotics site:

http://arcbotics.com/lessons/how-to-upload-sparki-code/

We suggest that you test your Sparki robot and the code upload process using the examples provided by ArcBotics before attempting to upload the Bluetooth control code.

Uploading the BT Control Code

The Sparki Bluetooth Control code file, SJD_BT_Control.ino, is available here: https://github.com/sivads/SJD_BT_Control. If you are not familiar with GitHub, that's ok. Just download the ZIP file by clicking the Download ZIP button on the right-hand side of the screen. A file named SJD_BT_Control-master.zip will be downloaded to your computer. After the file downloads, unzip the file and place the SJD_BT_Control.ino file somewhere on your hard disk where it is easily remembered — this is the file you will be uploading to Sparki.

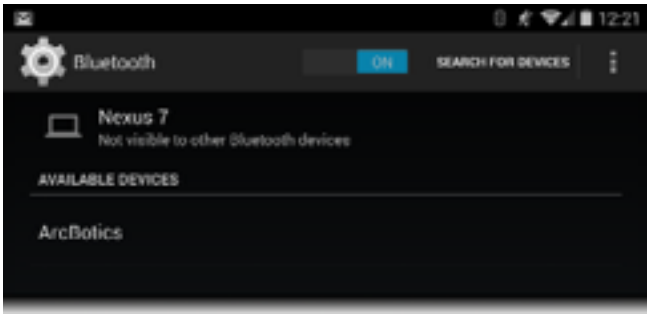
The basic steps for uploading the SJD_BT_Controll.ino file to Sparki is as follows:

1. Assuming you have installed the SparkiDuino program and it is working properly (see above), start the SparkiDuino program.
2. Connect your Sparki robot using the provided USB cable to your computer running SparkiDuino. The robot will be powered through the USB cable, so no need to turn on the robot.
3. In the SparkiDuino program, select File > Open… and open the SJD_BT_Control.ino file located wherever you placed it on your hard drive above (navigate to the file, select it, and press Open).
4. The code will be loaded into a second window.
5. Press and hold the tiny RESET button on the Sparki robot. The RESET button is located in front of the power switch.
6. Select File > Upload or press the second (from the left) round blue icon with the arrow at the top left of the window.
7. Wait for the word "Uploading…" to appear in the blue status bar near the bottom of the window then release the RESET button on Sparki. The red LED next to the green power LED should flash.
8. After the program is uploaded, the SparkiDuino program reports "Done Uploading." in the status bar. Sparki will now start executing the program. You should hear two beeps (lower beep then higher beep) to alert you that the program loaded properly and Sparki is running. You should also see "BT Control" at the top of Sparki's LCD screen.
9. You can now unplug Sparki from your computer. This will turn off Sparki and stop the program.
10. Turn on Sparki's power switch and he should restart the BT Control program. Listen for the two beeps to signal that the program started and initialized ok.
11. If the program fails to upload, before disconnecting Sparki, go back to step 5 and try to upload the program again. If this does not work, then your SparkiDuino program is probably not set up properly. Refer to the ArcBotics site to troubleshoot.

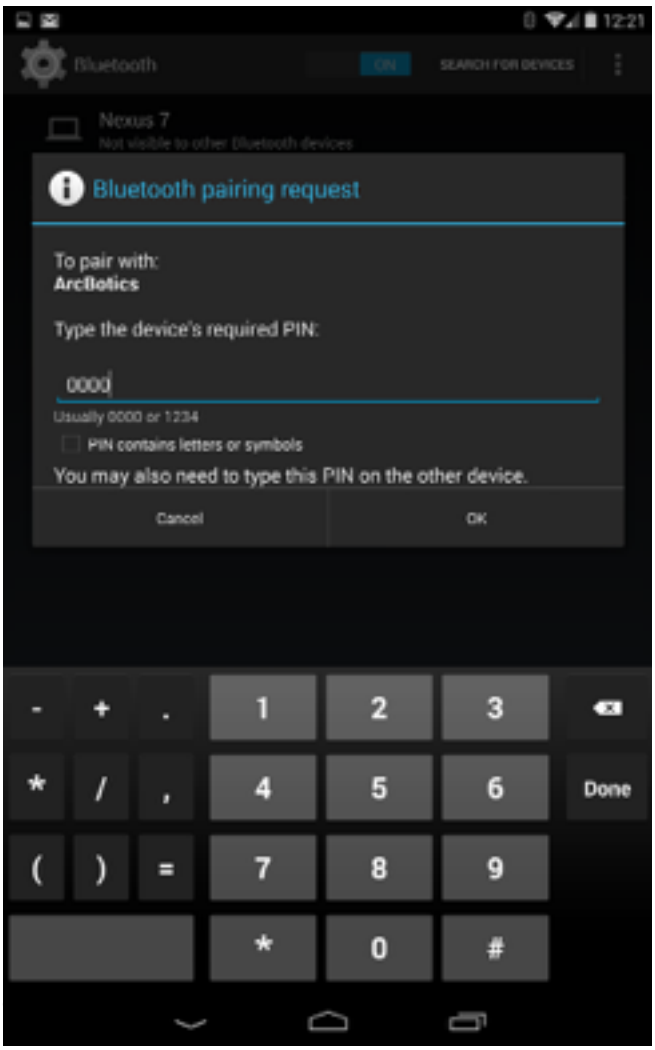
Pairing the Android Tablet with the Sparki Robot via Bluetooth

1. Turn on your Android Tablet.
2. Turn on the Sparki robot.

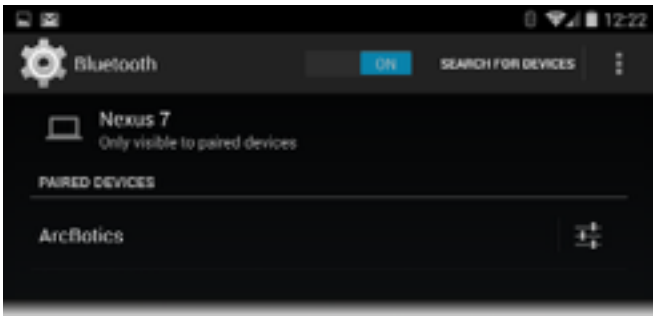
3. Go to Home Menu > Settings. Or, you can also get to the settings quickly by swiping down from the upper right corner of the action bar (top-most bar with icons).
4. Select Bluetooth. Make sure Bluetooth is ON.
5. Turn on your Sparki robot with installed Bluetooth module. The Bluetooth module should have a rapidly flashing red LED if installed properly.
6. Select “Search for Devices”. “ArcBotics” should appear under the list of Available Devices.



7. Select ArcBotics from the list. The Bluetooth pairing request dialog should appear. The default PIN is 0000 (as specified on the ArcBotics site here: http://arcbotics.com/products/bluetooth-2-1-module). Type in the PIN and press OK.



8. The tablet should now be paired with the Sparki robot!



Installing and Starting the Android Sparki Bluetooth Controller

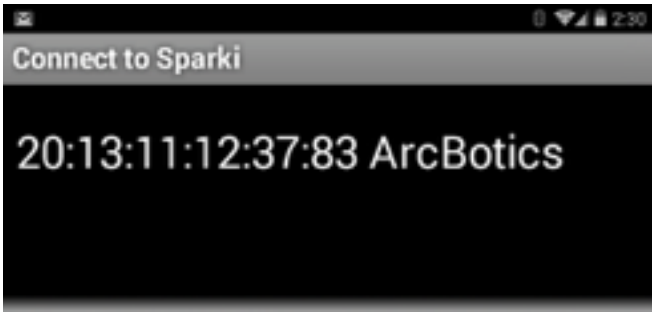
The Sparki Bluetooth Controller app was designed and tested on a 7-inch tablet (specifically the Nexus 7) in order to have plenty of display surface for an all-in-one controller. If you attempt to run the app on a smaller screen, it may not fit and/or may not run properly.

You can obtain and install the app from the Google Play Store (Sparki Bluetooth Controller)

After the app is installed, click the Sparki Bluetooth Controller icon to start the app — and away we go!

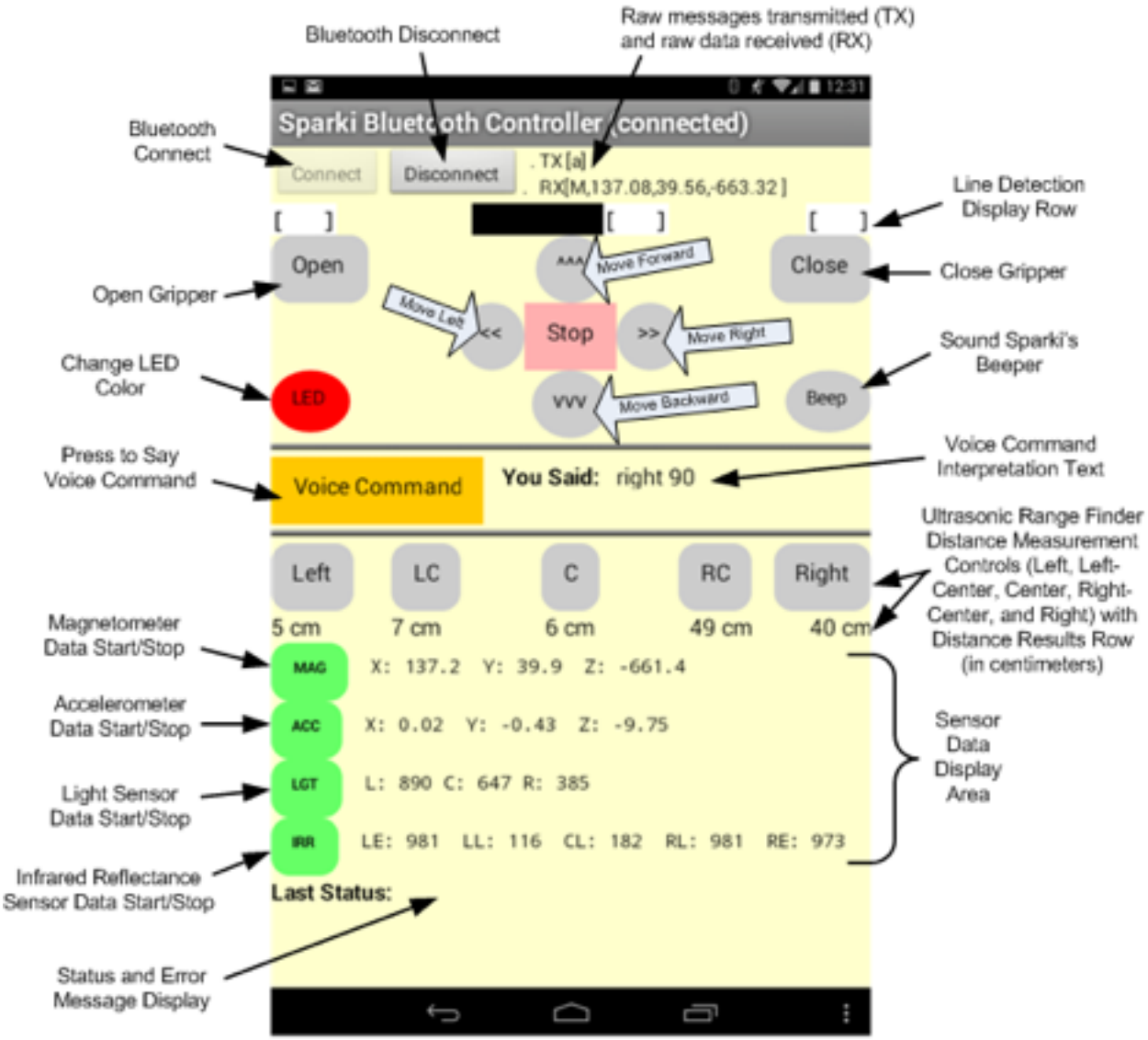
Basic Operations

When the app first starts, only the Connect button is functional. Make sure you have loaded the SJD_BT_Control.ino code to your Sparki robot (as above) and make sure the robot power switch is ON. You will hear two beeps (low then high) when Sparki had in fished initializing and is ready to connect. The red LED on Sparki's Bluetooth module should be flashing. Press the Connect button (upper left) on the tablet controller app. This will bring up a list of Bluetooth devices by address (the 12 alphanumeric characters) and name, similar to the picture below:



Select the ArcBotics choice from the list. When the Bluetooth connection is established, the list should disappear and you should see the main controller screen again. All the controller buttons should now be functional (except the Connect button).

The following diagram shows a summary of the controller functions:



Basic Movement Controls

The four buttons centered around the red Stop button control Sparki's movement. By pressing and holding one of the four buttons (Forward, Backward, Left, and Right), Sparki will move in that direction. Note that Sparki will automatically stop when you release the button. The Stop button is only for cases when you give voice command, like "forward" and you need to Stop Sparki (or he will continue to move forward). This is discussed further below.

Gripper Controls

You can open and close Sparki's front Gripper by pressing and holding the Open or Close buttons. A word of caution: Sparki has no feedback on his gripper to detect a fully open or fully closed position. Therefore, stop opening the gripper when the gripper moves to a fully opened position and likewise, stop closing it when it reaches a fully closed position. This avoids unnecessary stress on Sparki's gripper motor and drain on the battery.

LED Control

The LED button changes Sparki's RGB LED from red, to green, to blue, and then off again in sequence. Each time the button is pressed, the LED changes color. The button color will match the LED color.

Beeper

You can make Sparki's buzzer (beeper) beep once for every press of the Beep button. Beep, beep!

Distance Measurement

You can command Sparki to measure distances to objects using his Ultrasonic Range Finder (his eyes). For example, pressing the Left button (below the Voice Command section), Sparki will look to his left (90-degrees left) and send back the distance (in centimeters) to the nearest object in his field of view. This distance is displayed under the corresponding control (the Left button in this example. The other buttons are LC (left-center or 45-degrees left), C (center or straight ahead), RC (right-center or 45-degrees right), and Right (90-degrees right).

If he sends back a -1, this means that the sensor did not detect an object. This often occurs if a hard object is placed at a 45-degree angle to his sensors where the ultrasonic ping will get reflected away from the return sensor. You can read more about how the Ultrasonic Range Finder works on the ArcBotics site.

Magnetometer

Pressing the MAG button (button turns green) will instruct Sparki to start reading his magnetometer sensor and send the data back to the controller for display. Pressing it again (button turn back to normal color) instructs Sparki to stop reading and sending data. The 3-axis X, Y, and Z) magnetometer senses the magnetic field that Sparki is experiencing — which is often very erratic. We found this sensor somewhat unreliable for detecting magnetic direction because of the magnetic interference from Sparki. However, it is fun to see the sensor data displayed in real time!

Accelerometer

Pressing the ACC button starts and stops the flow of data from Sparki's Accelerometer sensor. Sparki has a 3-axis (X, Y, and Z) accelerometer from which data is displayed on the controller. Noticably, when Sparki goes up a hill, the Y value will become more negative. As he goes down a hill, the Y value becomes positive. The X value shows tilt and the Z value is handy for detecting when Sparki has been lifted.

Light Sensors

Sparki has three light sensors on the front of his circuit board: Right, Center, and Left. Pressing the LGT button starts and stops the flow of data read from these sensors. The more light that is detected, the higher the number. Shine a flashlight on Sparki and see what happens!

Infrared Reflectance Sensors

Sparki has five infrared reflectance sensors on the bottom of his chassis near the front edge. The three sensors near the center (LL = Left Line, CL = Center Line, and RL = Right Line) are typically used for black line detection and following. The two sensors on the outer edges (LE = Left Edge, and RE = Right Edge) are often used for edge detection - so Sparki does not fall off the edge of a table, for example. The IRR button starts and stops the flow of data from all five sensors. The sensor values are displayed horizontally to the right of the IRR button. Along with the numeric sensor data, a graphic display of the black/white surface detection is displayed above the movement controls. This shows how Sparki can detect black or white surfaces for line following and position cues.

Voice Command

Press the Voice Command button, wait for the prompt, then speak a voice command (see Appendix A).

Data Transmitted and Received

The raw data commands that are sent to Sparki (TX) and the raw data that is received from Sparki (RX) is displayed in the upper right area of the controller display. You will notice that

each time a command is sent to Sparki, the dot next to the TX label will advance one position to the left. This simple animation gives a visual indicator that a command is being sent. For example, if the same command is sent multiple times, without the animation, you would not be able to tell that it was multiple commands. The data received has a dot that advances to the right. Same principal.

Also, the commands that are sent to Sparki appear on Sparki's LCD as Command and (optional) Value.

Last Status Message

A status message will be displayed from time to time to report errors or other conditions (for example, disconnected status). Occasionally, data transmission errors are detected/trapped and a message explaining what was detected will be displayed. If, for example, the data coming back from Sparki is malformed for some reason, the RX data parser will catch the error and report the details of the malformation. This was kept in the final program to show a practical, working example of data format validation and error trapping. Although errors should not occur very often, don't be alarmed if you see one occasionally. The program is handling the errors and simply reporting what was detected and handled.

Appendix A: Voice Commands

By pressing the Voice Command button on the Sparki Bluetooth Controller app, you can speak simple commands and watch Sparki react to those commands. The Voice Command function uses the built-in voice recognition engine in Android. Like with all voice recognition engines, speak at a normal pace and speak clearly. The text displayed next to the "You Said:" label on the controller will print what the voice recognition engine interprets. The controller was programmed to recognize very specific commands that will control Sparki. The following is a list of those commands:

Voice Command	Sparki's Response
"Forward"	Moves Sparki Forward until the red Stop button is pressed or the "Stop" voice command is issued and interpreted correctly (best to press the Stop button!)
"Forward {number of steps}", for example, "Forward 20"	Moves Sparki Forward the number of incremental steps specified. Steps are limited to 50 (i.e., if you say "Forward 1000", Sparki will only move forward 50 steps).
"Left" (or "Port")	Moves Sparki Left until the red Stop button is pressed or the "Stop" voice command is issued and interpreted correctly (best to press the Stop button!)
"Left (number of degrees}", for example, "Left 90"	Moves Sparki Left the number of degrees specified. Degrees are limited to 1800 or 5 complete 360-degree turns.
"Right" (or "Starboard")	Moves Sparki Right until the red Stop button is pressed or the "Stop" voice command is issued and interpreted correctly (best to press the Stop button!)
"Right {number of degrees}", for example, "Right 360"	Moves Sparki Right the number of degrees specified. Degrees are limited to 1800 or 5 complete 360-degree turns.
"Back" (also "Backward" or "Reverse")	Moves Sparki Right until the red Stop button is pressed or the "Stop" voice command is issued and interpreted correctly (best to press the Stop button!)
"Back {number of steps}", for example, "Back 30" or "Backward 30"	Moves Sparki Backward the number of incremental steps specified. Steps are limited to 50 (i.e., if you say "Back 1000", Sparki will only move backward 50 steps).

Voice Command	Sparki's Response
"Stop" (or "Halt")	Stops Sparki if he is moving. Note that the The Stop command (or Stop button) does not work if Sparki is executing a command with number of steps or number of degrees. For example, if you issue a "Forward 50", Sparki will move forward 50 steps without stopping. Use caution when specifying large step values forward or backward (although they are limited to 50 as described above).
"Beep (or "Horn")	Sounds Sparki's beeper (or buzzer as referred to by ArcBotics).
"Hi Sparki" (or "Hello Sparki")	Sparki will sound a hello greeting beep.
"Play Song" (or "Play Shave and a Haircut")	Sparki will play the familiar "Shave and a Haircut" (http://en.wikipedia.org/wiki/Shave_and_a_Haircut)

If you are interested in modifying the voice recognition commands, please refer to the SpeechRecognizer1.AfterGettingText block in the App Inventor code. Also, the step limit can be modified by changing the value the LIMIT_STEPS constant in the Sparki code in the SJD_BT_Control.ino file. Likewise, the turn limit (number of total degrees) can be modified by changing the LIMIT_TURNS constant.

Appendix B: Communications Protocol

Two way Bluetooth communication is used between the tablet and Sparki. Commands are sent from the tablet controller to Sparki and Sparki reacts and/or responds with data read from his various sensors. The commands are simple one character commands followed by an optional numeric command value. For example, the command "F" makes Sparki move forward and the command "f,10" makes Sparki move forward 10 steps.

There are also commands for Sparki to read a sensor continuously and report the data read back to the controller. Since the data transmitted back to the controller needs to be read and processed before more data is transmitted (asynchronously), the controller will send an ACK (command "a") back to Sparki when the data has been processed and the controller is ready to receive data again. If multiple sensors are being read and multiple data reports are being transmitted to the controller, a round-robin approach is used to send the data from the various sensors in a sequential and controlled manner.

A typical response from the light sensors looks like this:

T,605.00,578.00,584.00

where “T” in the first position indicates from what sensor the data was read (the data type). There are three light sensors, so the next three numbers correspond to those three sensors: Left read 605.00, Center read 578.00, and Right read 584.00.

Sparki can read and send data back from his Light Sensors (T), his Magnetometer (M), his Accelerometer (G), his Ultrasonic Range Finder (R), and his Infrared Reflectance Sensors (I). All of these sensors can be actively reporting data back to the controller at the same time using a round-robin (sequential) mode.

The complete list of commands that Sparki responds to can be found in the switch statement in the SJD_BT_Control.ino code and are summarized here:

Transmitted Command	Sparki’s Action
1	Range Left - measures the distance to object on left (90-degrees left)
2	Range Left of Center (LC) - measures the distance to object left of center (45-degrees left)
3	Range Center (C) -measures the distance to object that is directly in front of Sparki
4	Range Right of Center (RC) -measures the distance to object right of center (45-degrees right)
5	Range Right - measures the distance to object on right (90-degrees right)
a	Acknowledge data received. Equivalent to ACK
B	Move backward (until stop command is received)

Transmitted Command	Sparki's Action
b,{n}	Step Backward with number of steps (second parameter is number of steps). For example, b,20 moves Sparki backward 20 steps
C	Close the gripper (until stop gripper is received)
F	Move forward (until stop command is received)
f,{n}	Step Forward with number of steps (second parameter is number of steps). For example, f,30 moves Sparki forward 30 steps.
G	Starts Accelerometer data streaming
g	Stops Accelerometer data streaming
H	Sparki says hello!
I	Starts Infrared Reflectance sensor data streaming
i	Stops Infrared Reflectance sensor data streaming
L	Move left (until stop command is received)
l,{n}	Turn Left with Angle (second parameter is the angle in degrees)
M	Starts magnetic data streaming
m	Stops magnetic data streaming
O	Open the gripper (until stop gripper is received)
Q	Beep
q	Play the 'Shave and a Haircut' melody
R	Move right (until stop command is received)
r,{n}	Turn Right with Angle (second parameter is the angle in degrees)
S	Stop
T	Starts light sensor data streaming
t	Stops light sensor data streaming
w	Turn RGB LED Off
X	Stop gripper
x	Turn RGB LED Red
y	Turn RGB LED Green
z	Turn RGB LED Blue

{n} = an integer value.

The messages that are received from Sparki that contain the sensor data are as follows:

RX Data Message (data sent from Sparki to Controller	Sensor Data Type
G,{left},{center},{right}	Accelerometer Data
I,{left-edge},{left-line},{center-line},{right-line},{right-edge}	Infrared Reflectance Sensor Data
M,{X},{Y},{Z}	Magnetometer Data
R,{range in centimeters}	Range Measurement (from Ultrasonic Range Finder)
T,{left},{center},{right}	Light Sensor Data

Appendix C: Help and Support

We want you to have fun learning and exploring robotics! You will find that the ArcBotics Sparki forum (http://forum.arcbotics.com) is helpful for getting answers to question related to the Sparki robot platform. Likewise, if you have questions about the MIT App Inventor 2 environment, the community is a great place to ask questions and get answers (https://groups.google.com/forum/#!forum/mitappinventortest).

For help with the Sparki Bluetooth Controller app or the SJD_BT_Control.ino code, feel free to e-mail us at support@sjdavisrobotics.com and we'll answer your questions.

The MIT App Inventor 2 source code (.aia file) is available for the controller app by sending a request to the above support e-mail address along with the "request code." The request code is a four digit code that Sparki will tell you if you ask him! Just ask him, "Code Please" (using the Voice Command button on the controller). The controller will then speak the code. If you cannot get the code, send us an e-mail and we'll probably give it to you anyway!

The source code is free under the MIT Open Source License.

Please consider making a donation to our educational fund that will go toward further projects like this and other educational endeavors (i.e., my son's college fund!)

E-mail/PayPal Donations: donations@sjdavisrobotics.com

BitCoin Donations: 1Hp3htmCAyiYNg52XXhoVtjB7VdkqaQDMq

THANK YOU!

Help Us Help You!

If you are thinking about posting a less than stellar review of our application on Google Play, please take a minute and e-mail us — give us a fair chance to address your issue or concern.

We will do our best!