

SunSPOT Tutorial and Exercises

Set up:

1. Click on Applications in the menu at the top of the screen.
2. Select Programming > NetBeans
3. NetBeans will open. There will be several projects in the left frame. We will be using these projects throughout the Exercises.

Part 1: The Bouncing Ball Demo

This demonstration program will familiarize you with the process of building a project and deploying it to the SunSPOT. It will also show you how the accelerometer and LEDs work.

1. Right-click on BounceDemo-OnSPOT and select "Set Main Project".
2. Right-click on BounceDemo-OnSPOT and select "Clean and Build Project".
3. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful (total time: 1 second)". If you do not see "Build Successful" notify a teacher.
4. Make sure your SPOT is connected to the USB cable.
5. Right-click on BounceDemo-OnSPOT and select "Deploy to Sun SPOT".
6. At the bottom of the screen, you will see a bunch of text. It may say "(please reset SPOT on port COM6)". This means that the program wants you to reset your SPOT. To reset, press the small button on the bottom of the SPOT. If you have done this correctly, you will see more text at the bottom of the screen that will end with the following:

```
|=====| 30%  
|=====| 35%  
|=====| 40%  
|=====| 45%  
|=====| 50%  
|=====| 55%  
|=====| 60%  
|=====| 65%  
|=====| 70%  
|=====| 75%  
|=====| 80%  
|=====| 85%  
|=====| 90%  
|=====| 95%  
|=====| 100%
```

```
Exiting  
jar-deploy:  
BUILD SUCCESSFUL (total time: 42 seconds)
```

If you do not see this message, notify a teacher.

7. Unplug your SPOT from the cable.
8. Reset your SPOT again.
9. The LEDs at the top of your SPOT should light up. At each end you will see red LEDs and there will be a 'moving' LED in the middle.
10. Move your spot from left to right and watch the ball bounce.
11. Remove the 'SunRoof' by pushing gently on the bump at the top of the SunRoof with the tip of your fingernail.
12. Find a partner and explore what happens when you click the switches labeled SW1 and SW2 just under the LEDs.

Part 2: The Air Text Example

In this exercise, you will first run a program that lights up the LEDs such that they spell words when you shake the SPOT back and forth. Then, you will then change the color of the words and the text displayed. Finally, you will use the temperature sensor on the SPOTs and display the current temperature rather than a text message.

1. Right-click AirTextExample and select "Set Main Project"
2. Right-click on AirTextExample and select "Clean and Build Project".
3. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful (total time: 1 second)". If you do not see "Build Successful" notify a teacher.
4. Make sure your SPOT is connected to the USB cable.
5. Right-click on AirTextExample and select "Deploy to Sun SPOT".
6. If you see a message asking you to reset, press the button on the bottom of the SPOT.
7. If you do not see a message ending with "Build Successful", notify a teacher.
8. Unplug your SPOT from the cable.
9. Reset your SPOT again.
10. Hold your SPOT sideways such that the reset button is on the right and at the bottom. The LEDs should be vertical and at the left side.
11. Gently shake your SPOT side to side. You should see a message.

Open the Java code for this example using the following procedure:

12. Double-click AirTextExample
13. Double-click src
14. Double-click org.sunspotworld.demo
15. Double-click AirTextDemo.java
16. On the right of the screen you will see some Java code.

Exercise 1:

Scroll down until you see `/** Part 2 Exercise 1**/`

You will notice several lines of code. These lines set the color and text of the message you just saw.

(1) Explore what happens when you change the color of the text. To do this, you will change the parameters passed into the `setColor` method. The `setColor` method is invoked anywhere you see `disp.setColor(...)`. You'll notice that there are three comma-separated numbers between the parentheses. The first number specifies the amount of RED, the second number specifies the amount of GREEN, and the last number specifies the amount of BLUE. Each number must be between 0 and 255. The values (255, 0, 0) means that the color will be completely RED. If you change the first line to (255, 0, 255), you should see "HI!" in purple.

Change the values in any way you choose, but make sure to use numbers between 0 and 255. Once you have made a change select File->Save. Then, right-click on AirTextExample and select "Clean and Build Project". Make sure the text at the bottom ends with "Build Successful". Next, attach your SPOT and select "Deploy to Sun SPOT". As before, reset your SPOT if necessary and make sure you see the "Build Successful" message. Unplug your SPOT, reset it, and see what happens!

(2) Explore what happens when you change the message displayed by the SPOT. To do this, you will change the parameters passed into the swingThis method. The swingThis method is invoked anywhere you see disp.swingThis(...). You'll notice that there are two comma-separated values between the parentheses. The first is a String that specifies the text to display. The second is a number specifying the number of swings to use to display the message. If you change "HI!" to "SEP", the SPOT will display SEP instead of HI!. Note that the quotation marks MUST appear on either side of your string.

Change the values in any way you choose. However, we recommend that you keep the words to a maximum of 6 characters. Longer words may not appear correctly, or may appear backwards. Once you have made a change select File->Save. Then, right-click on AirTextExample and select "Clean and Build Project". Make sure the text at the bottom ends with "Build Successful". Next, attach your SPOT and select "Deploy to Sun SPOT". As before, reset your SPOT if necessary and make sure you see the "Build Successful" message. Unplug your SPOT, reset it, and see what happens!

Exercise 2:

You will now change the AirTextExample so that it displays the temperature rather than a random message. First, delete all of the code between the comment `/** Part 2 Exercise 1**/` and the comment `/**End Part 2 Exercise 1**/`.

Next, delete the `/**` just before `//Code for Part 2 Exercise 2:` and the `**/` just after `//Step 2: display the temperature stored in the temp variable using the swingThis method.`

Locate the statement `// add your code here.` Just below that statement, you will add two lines of code. The first line of code will use the setColor method to set the color of the message you will display. Remember, the code should look something like this:
`disp.setColor(255, 0, 0);`

The second line of code will use the swingThis method to set the text of the message you will display. In this case, you want to display the value stored in the variable temp. Remember, the code should look something like this: `disp.swingThis(temp, 5);` In the previous examples we looked at, we specified the string we wanted to display between quotation marks. In this case, we want to specify the value stored in a variable. To do this, we do not need to include the quotation marks, just the name of the variable.

Once you have made a change select File->Save. Then, right-click on AirTextExample and select "Clean and Build Project". Make sure the text at the bottom ends with "Build Successful". Next, attach your SPOT and select "Deploy to Sun SPOT". As before, reset your SPOT if necessary and make sure you see the "Build Successful" message. Unplug your SPOT, reset it, and see what happens

Part 3: Radio Strength

This project will demonstrate how the SPOTs communicate. You will run a program that sends messages between the SPOTs. When your SPOT is receiving messages, the left-most LED will be green. When it is not receiving messages, the left-most LED will be red.

1. Right-click RadioStrength and select "Set Main Project"
2. Right-click on RadioStrength and select "Clean and Build Project".
3. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful (total time: 1 second)". If you do not see "Build Successful" notify a teacher.
4. Make sure your SPOT is connected to the USB cable.
5. Right-click on RadioStrength and select "Deploy to Sun SPOT".
6. If you see a message asking you to reset, press the button on the bottom of the SPOT.
7. If you do not see a message ending with "Build Successful", notify a teacher.
8. Unplug your SPOT from the cable.
9. Reset your SPOT again.
10. Find a partner and, together, find an area where there are no other students around.
11. If you hold your SPOTs close together, you will see a green LED on the left side indicating that your SPOT is communicating correctly with the other SPOT. You will also see several blue LEDs that specify the strength of the communication signal between your SPOTs. As you walk apart, you will notice that the number of blue LEDs gets smaller, and eventually the LED on the left will turn red. This means that your SPOT is too far away from the other SPOT and is no longer receiving messages. How far apart can your SPOTs be and still communicate?

Part 4: LEDs and the Light Sensor

This project demonstrates the LEDs and the light sensor. It will also give you the opportunity to write some Java code. Feel free to experiment! Play around with the code and see what happens. As always, make sure to ask us if you have any questions!

1. Right-click LightAndLEDEExample and select "Set Main Project"
2. Right-click on LightAndLEDEExample and select "Clean and Build Project".
3. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful (total time: 1 second)". If you do not see "Build Successful" notify a teacher.
4. Make sure your SPOT is connected to the USB cable.
5. Right-click on LightAndLEDEExample and select "Deploy to Sun SPOT".
If you see a message asking you to reset, press the button on the bottom of the SPOT.
6. If you do not see a message ending with "Build Successful", notify a teacher.
7. Unplug your SPOT from the cable.
8. Reset your SPOT again.
9. You should see the LEDs light up one at a time.

Open the Java code for this example using the following procedure:

10. Double-click LightAndLEDEExample
11. Double-click src
12. Double-click org.sunspotworld.demo
13. Double-click LightAndLEDEExample.java

On the right side of the screen you will see some Java code.

Try the following exercises. Remember, after you make any change, you must "Clean and Build Project", attach your SPOT, "Deploy to Sun SPOT", unplug, and reset your SPOT.

Notify a teacher if you have ANY questions or wish to discuss your solutions! We're here to help!

- (1) Change the LEDs to be RED instead of BLUE.
- (2) Turn on all of the LEDs at the same time. Hint: the line of code `Utils.sleep(500);` changes the amount of time that the SPOT waits before turning on the next LED.
- (3) Find the where it says `/*Part 4 Exercise 1*/` in the code. Write an if statement, similar to the example given in the code, that sets the color of the LEDs based on the level of the light observed by the SPOT. Try removing the Sun Roof to see if this changes the color of the lights.

Part 5: Social Apps

Exercise 1 - Twitter

In this exercise, our SunSPOTS will send updates to Twitter every 10 minutes. The updates will include information about the temperature of the room and how light/dark the room is. We'll start these running at the end of the day to track how the room changes overnight.

This exercise is based on the [CenceMe](#) application for the iPhone. The application uses the iPhone's sensors to determine what you're doing. For example, it can tell if you're at a party, talking with someone, at the gym, etc, and then posts this message to Facebook.

1. For this exercise, you need a Twitter account. Go to twitter.com and create a new account. Make sure to take note of your login and password, you'll need these later!!
2. Right-click on TwitterDemo and select "Set Main Project"
3. Right-click on TwitterDemo and select "Clean and Build Project".
4. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful (total time: 1 second)". If you do not see "Build Successful" notify a teacher.
5. Double-click TwitterDemo
6. Double-click src
7. Double-click org.sunspotworld.demo
8. Double-click TwitterDemo.java
9. Find in the code where it says `/** Part 5 Exercise 1 - add your username and password */`. You'll see a line of code like the following:

```
authStr = Base64.base64Encode("[username]:[password]");
```

10. Delete the [username] part and replace with your username from twitter.
11. Do the same for the [password].

12. Find in the code where it says `/** Part 5 Exercise 1 - add the temperature */`. After this comment, add code that will read the current temperature (if you don't remember how to do this, please see Part 2, Exercise 2. The project in NetBeans is called AirTextExample. If you need help with this step, please ask a TA!). Once you have the temperature, use it to create a message for the SunSPOT to send to twitter. The tweet could be something like

```
tweet = "The current temperature is " + temp + ". ";
```

or you can get fancy and use an "if statement" to select how the message should read

```
if(temp > 80.0)
    tweet = "It's hot in here! ";
else
    tweet = "It's cold in here! ";
```

Be creative!

13. After you have the temperature added to the message, let's add some information about how light/dark it is in the room. Find in the code where it says `/** Part 5 Exercise 2 - add light information */`.

14. Uncomment the code below this line (aka, remove the `//`). This code gets a number value for the amount of light sensed by the SunSPOT.

15. Use the light value to add more info to your message. You can add something like the following:

```
tweet += "The light sensor reads: " + light;
```

or use an if statement like before!

NOTE: Notice the `+="` sign in the code above. This takes what you already have in the tweet and adds more to it. If you don't add the `+`, your temperature message will get overwritten!

ALSO NOTE: the max length of a twitter tweet is 140 characters!! Make sure your message is less than 140 characters.

16. Make sure your SPOT is connected to the USB cable.

17. Right-click on TwitterDemo and select "**Build +** Deploy to Sun SPOT".

If you see a message asking you to reset, press the button on the bottom of the SPOT.

18. If you do not see a message ending with "Build Successful", notify a teacher. Don't reset the SPOT yet!

19. Connect the BaseStation to the USB cable.

20. Open a terminal window and type: `cd /usr/java/SunSPOT/sdk`

21. Now type: `ant -Dport=/dev/ttyACM0 socket-proxy-gui`

22. A window should open. In this window, select I/O and then click "Start"

23. Now reset your SunSPOT. This will start the application. You should see one of the lights on the sensor blink a couple times. This means your message is being sent to Twitter!

24. Go to `twitter.com/yourusername` to verify.

Exercise 2 - Find a Friend

Now let's see how we can use the sensors to socialize! We'll come up with 8 multiple choice questions as a group that tell something about you. For example:

1. What is your favorite type of music?
 - A. Rock
 - B. Indie
 - C. Classical
 - D. Hip-hop

...

You're going to add the answers to the questions in the code. The SunSPOTs will then be used to see how similarly you answered the questions. Let's first come up with some questions.

1. Come up with 8 questions as a group.
2. Right-click on SocialApp and select "Set Main Project"
3. Right-click on SocialApp and select "Clean and Build Project".
4. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful (total time: 1 second)". If you do not see "Build Successful" notify a teacher.

5. Double-click SocialApp
6. Double-click src
7. Double-click org.sunspotworld.demo
8. Double-click SocialApp.java

9. Find the part in the code that says `/** Part 5 Exercise 2 - Add answers */`
10. Uncomment the code following it (remove `/*` and `*/`). There will be some errors, that's OK. This is where you will place the answers to the questions.
11. To answer the questions, place lower case letters after the equal signs in the code. `answer[0]` corresponds to the first question. `answer[1]` corresponds to the second question, etc. The letters need to have a single quote on each side. Here is an example:

```
answer[0] = 'a';
```

...

12. When you're done answering all the questions, make sure your SPOT is connected to the USB cable.
13. Right-click on SocialApp and select "**Build + Deploy to Sun SPOT**". If you see a message asking you to reset, press the button on the bottom of the SPOT.
14. If you do not see a message ending with "Build Successful", notify a teacher.

15. Unplug your SPOT from the cable.

10. Find a partner and, together, find an area where there are no other students around.

11. Reset the SPOT.

11. If you hold your SPOTs close together, the lights should light up on the sensor. The more lights there are, the more similarly you answered the questions. If all 8 lights are RED, then you answered no questions in common. :(If all 8 lights are NOT red and on, then you answered every single question the same!! NOTE: you might need to reset the sensor if nothing is happening.

Part 6: Advanced Application

One of the grad students in the CS program at USF created a really cool application for the wireless sensors. She programmed the sensor to interact with a computer as a mouse!
Let's see her application in action.

1. Double-click MouseControl-onSPOT
2. Double-click src
3. Double-click org.sunspotworld.cs686
4. Double-click MouseControl.java

5. Find the line in the code that reads

```
private static final String baseStationaddr = "0014.4F01.0000.3474";
```

6. Change the the last 8 digits to match the 8 digits on your base station. These numbers are found on the base station's fin.

7. Attach a SunSPOT to the USB.
8. Right-click on the MouseControl-onSPOT again and select "Build + Deploy to Sun SPOT". If you see a message asking you to reset, press the button on the bottom of the SPOT.
9. If you do not see a message ending with "Build Successful", notify a teacher.

10. Double-click MouseControl-onDesktop
11. Double-click src
12. Double-click org.sunspotworld.cs686
13. Double-click MouseControlHostApplication.java

14. Find the line in the code that reads

```
private static final String baseStationaddr = "0014.4F01.0000.3474";
```

15. Change the the last 8 digits to match the 8 digits on your base station. These numbers are found on the base station's fin.

16. Right-click on MouseControl-onDesktop and select "Set Main Project"
17. Right-click on MouseControl-onDesktop and select "Clean and Build".
18. At the bottom of the screen you will see a bunch of text. It should end with "Build Successful...". If you do not see "Build Successful" notify a teacher.

19. Hook up the base station to the USB.
20. Right-click on MouseControl-onDesktop again and select "Run".

21. Reset your SPOT that has the MouseControl-onSPOT application loaded.
22. Move the SunSPOT around, see what happens to the Cursor. Remove the cover of the SunSPOT and press the buttons, see what happens.