

SunSPOT Hello World Tutorial

Tom Huber (huber@gac.edu), Gustavus Adolphus College Physics Department
http://physics.gac.edu/~huber/classes/phy235/SunSpot_HelloWorld.doc
January 7, 2008

Make a New Project

- In NetBeans, open an existing project: a good one to start with is HelloWorld which can be found on the Physics network in the Robotics folder
- Right Click on the Project Name
- Select Copy Project
- Give the project a new name (MyHelloWorld)
- Give the project a new location (such as Z:\MyData\Robotics)

Build and Deploy the Project

- In NetBeans, right click on the project
- Select Clean Project (which deletes all of the output files)
- Select Build Project – This compiles the project into source code into executable code that the SunSPOT can run
- Select Deploy Project
 - In the NetBeans Output Window, there are a couple of possibilities on what will show up
 - **Download operation completed successfully**
Exiting
jar-deploy:
BUILD SUCCESSFUL (total time: 4 seconds)
 - Congratulations! The program downloaded successfully
 - **Waiting for target to synchronise...**
(please reset SPOT if you don't get a prompt)
[waiting for reset]
 - Press the RESET button on your SPOT, and it should finish loading the program
 - **Error: COM3 is in use**
The SPOT client will now exit
followed by several lines with blue underlined text
BUILD FAILED (total time: 2 seconds)
 - This means that the SPOT is running a program at the same time that you are trying to deploy a program.
 - Press the RESET button and try again
 - If this does not work, make sure that you press the RESET button roughly ½ second before you press Deploy.

- If this also does not work, open the Task Manager window (Control-Alt-Delete), and look on the Processes tab for applications named Java. One instance of this job will take about 115kB of memory – this is the NetBeans process, and you don't want to kill it. If there is another instance of Java that takes less than about 10kB, kill it and try again.

Run the Project

- In NetBeans, right click on the project, and select Run Project
- In the Output Window it should print
Hello World

Structure of the Program

- In NetBeans, right click on the + sign,
- Then right click on + for src (which means Source code), and GAC
- At the bottom level, there should be link labeled HelloWorld.java
This is called a Java class – this is the program that the SunSPOT is running
Double click on this, and it should come up with a window on the right side of the screen. This is the source code for the program that is running on the SPOT

- The program starts with a section of grey text (see lines 1-5 in printout below)

```

/*
    Hello World Application
And ends with
*/

```

In Java, as well as C and C++, any text that comes between a /* and */ is called a comment, and is completely ignored by the compiler

- The next section of the program includes lines such as
package GAC;
import com.sun.spot.util.*;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

This section of code tells the compiler that there are additional libraries that need to be loaded to run your program. For example, the EDemoBoard library is the library that tells the program how to handle devices on your SPOT such as the LED's and switches on the front panel. You should leave all of this alone – your program will need some or all of these imported packages (and we will add others later), so you should not need to change these at all

- Scrolling down a little farther, you will notice the text
public class HelloWorld extends MIDlet {
 protected void startApp() throws MIDletStateChangeException {
This is the start of the real program that your SPOT is running. Again, you should not need to change anything in the section before the startApp declaration.

```
1  /*
2   Hello World Application for SunSPOT
3   Tom Huber (huber@gac.edu), Gustavus Adolphus College Physics Department
4   January 7, 2008
5  */
6
7  package GAC; // This is the package name: this code is in the directory src\GAC\
8
9  // The lines below import Java packages - You may need to add to this list for other packages
10 import com.sun.spot.util.*;
11 import javax.microedition.midlet.MIDlet;
12 import javax.microedition.midlet.MIDletStateChangeException;
13 import com.sun.squawk.*;
14 import com.sun.spot.peripheral.*;
15 import com.sun.spot.sensorboard.EDemoBoard;
16 import com.sun.spot.sensorboard.peripheral.*;
17 import java.util.*;
18 import java.io.*;
19 import java.lang.*;
20 /*
21  The line below defines a Java Class that contains your application
22 */
23 public class HelloWorld extends MIDlet {
24
25  // ----- This is the start of the program ----- //
26  protected void startApp() throws MIDletStateChangeException {
27      new BootloaderListener().start(); // monitor the USB (if connected) and recognize c
28
29      System.out.println("Hello World"); // Print Hello World to the display
30
31      ExitSpot(0); // This exits the program
32  }
33  // ----- This is the end of the program ----- //
34
35
36
37  protected void pauseApp() {
38      // This will never be called by the Squawk VM
39  }
40
41  protected void destroyApp(boolean arg0) throws MIDletStateChangeException {
42      // Only called if startApp throws any exception other than MIDletStateChangeException
43  }
44
45  protected void ExitSpot(int code) {
46      // This subroutine will exit a program and restart the Virtual Machine
47      // Tom Huber, Gustavus Adolphus College
48      // January 7, 2008
49
50      VM.stopVM(code);
51  }
52
53  protected void PowerDownSpot(int code) {
54      // This subroutine will turn off power to the SPOT
55      // Tom Huber, Gustavus Adolphus College
56      // January 7, 2008
57
58      final int SPI_CONFIG = (ISpiMaster.CSR_NCPHA | ISpiMaster.CSR_BITS_8 | ISpiMaster.CSR
59      ISpiMaster spi = Spot.getInstance().getSPI();
60      spi.sendReceive8(new SpiPcs(PeripheralChipSelect.SPI_PCS_POWER_CONTROLLER, SPI_CONFIG
61  }
62 }
```

- Finally you will get to the line


```
System.out.println("Hello world");
```

 This is the actual command that causes the program to print the words on your computer screen!
- At the end of the program there is the statement


```
ExitSpot(0);
```

 Followed by a series of definitions of other procedures. Again, you should not need to modify anything after this point in the program.
- **There are several important notes about the structure of Java programs:**
 - **Comments start with // or enclosed in /* */ pairs**
 - **All statements end with a semicolon**
 - **You must be careful about upper and lower case!**
 - **Curly braces { } are used to differentiate blocks of code**
 - **Using indentation for formatting is helpful**

Modifying the Hello World Program

- Change the System.out.println line to have it say `Hello, Your Name Here`
 - It is important that you make sure that the text that you want to display remains inside the quotation marks.
 - Build your program again and deploy it as discussed above. When you run it, it should print out your new text message. Cool...
- Add another System.out.println statement to have it print some additional text.
 - Make sure that you have the proper upper and lower case
 - Make sure that your text is enclosed in quotation marks
 - Make sure that you end the statement with a semicolon
 - Build, deploy and run your program. Hopefully it works correctly.
- We are now going to add a loop to repeatedly execute a section of code. Add the following text to your program

```
int i; // This defines a new variable i
for (i=0;i<5;i++) { // Do the following code 5 times
    System.out.println("The square of " + i + " is " + i*i);
}
```

- Make sure that you are careful about the syntax for each line
- Every quote must have a matching quote
- While indentation of the statements in the for loop are not required, it makes it much more readable
- Your program will run with i going from 0 to 4. To change it to go from 0 to 5, you could use `for (i=0;i<=5;i++) {`

***Congratulations, you have made your first Java program
for the SunSPOT!***