Javelin Stamp Errata v1.3

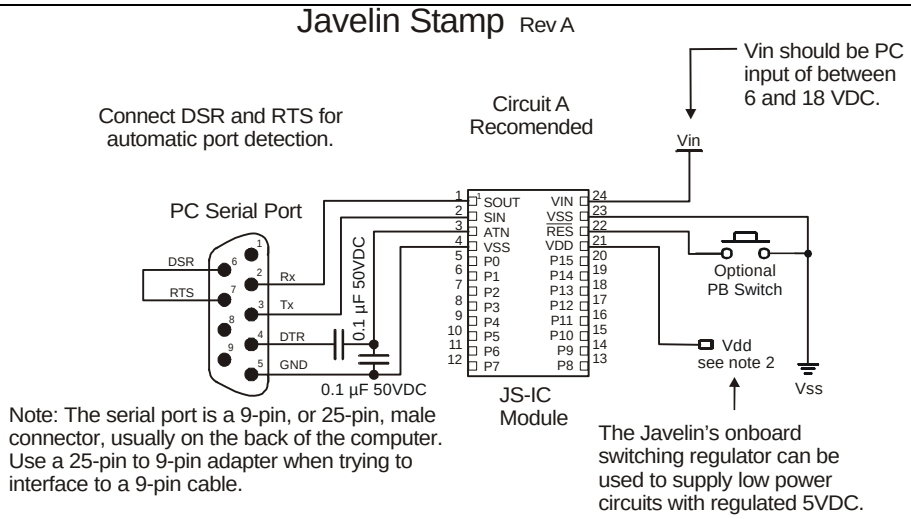
For the Javelin Stamp Manual Version 1.0

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

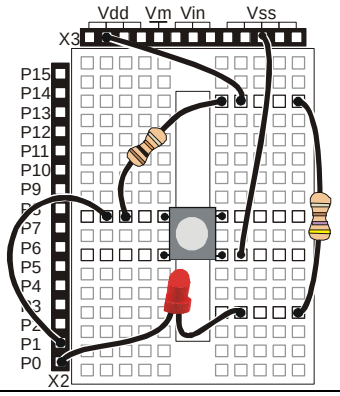
The following are known printing mistakes in the Javelin Stamp Manual v1.0, please be aware of them.

Page viii · Table of Program Listings	
Problem	Table of Program Listings is incomplete.
Explaination	Program Listing 9.7 - Timer Example is missing.
Solution	Add "Program Listing 9.7 – Timer Example" to the table.

Page 14 · Figure 2.3	
Problem	Schematic incomplete.
Explaination	Two capacitors are missing, there should be a capacitor between DTR and ATN, and a capacitor between ATN and VSS(GND). Both capacitors are 0.1µF @50 VDC.
Solution	

Page 15 · Bottom Paragraph	
Problem	Text Clarification
Explaination	The sentence: "If it does not appear, run the welcome application from the CD's root directory."
Solution	Should read: "If it does not appear, run the *Welcome application (*`Welcome.exe`*)* from the CD's root directory."

Downloaded from Elcodis.com electronic components distributor

Page 26 · Figure 2.13(b)	
Problem	None, circuit will work as pictured.
Explaination	The breadboard in Figure 2.13(b) does not match the schematic in Figure 2.13(a).
Solution	The LED and 470 Ω resistor from Figure 2.13(b) were swapped to match the circuit shown in Figure 2.13(a). 

Page 35 · Program Listing 2.6 – Math Example	
Problem	Program variables do not match actual variables in code.
Explaination	The variables, **`temporary`** and **`results`**, each have been referenced twice with capitalization as **`Temporary`** and **`Results`**.
Solution	Change the following lines: **`Temporary = temporary/10;`** **`Result = temporary*scale;`** **`Temporary /= 10;`** **`Result = (14*2+3)/10*scale;`** To this: **`temporary = temporary/10;`** **`result = temporary*scale;`** **`temporary /= 10;`** **`result = (14*2+3)/10*scale;`**

Downloaded from Elcodis.com electronic components distributor

Page 47 · Program Listing 3.12 – Method Example	
Problem	Program variable does not match actual variable in code.
Explaination	The variable, **`sigmaT`**, was referenced with capitalization as **`SigmaT`**.
Solution	Change the following line: **`SigmaT = avg(a,b,c,d,e) + 100/x;`** To This: **`sigmaT = avg(a,b,c,d,e) + 100/x;`**

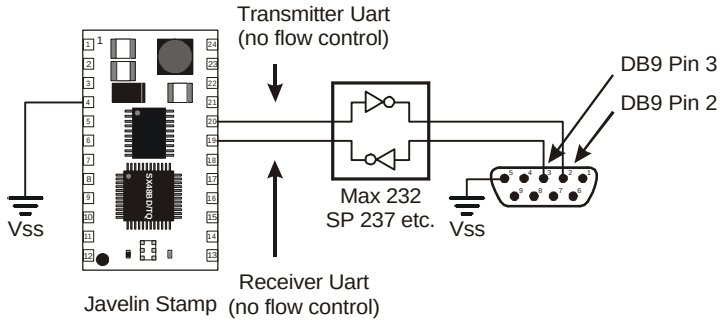
Page 52 · Bulleted list, 2nd item	
Problem	Power supply is not included with the Javelin Stamp Starter Kit.
Explaination	The 2nd line which reads "If you are using the Javelin Stamp Starter Kit, which comes with a 1000 mA supply, connect Vm to Vdd."
Solution	Should read, "If you are using a wall mounted power supply (1000 mA recommended), connect Vm to Vdd."

Page 91 · Table 6.2: Escape Sequences	
Problem	Misrepresented value of an escape sequence.
Explaination	The row: \u0013 | Clear Screen
Solution	Should be: \u0010 | Clear Screen

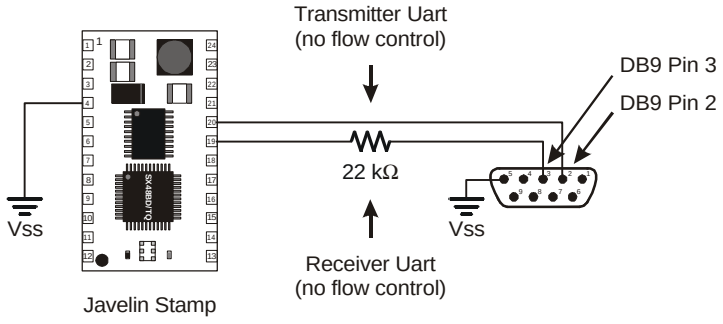
Page 184 · PWM text	
Problem	Misrepresented input value for PWM.
Explaination	The sentence "(or any two equal numbers from 1 to 255)".
Solution	Should read: (or any two equal integers). The **`PWM`** will accept integer values from 0 to 65535. The Javelin's **`int`** field can hold values from –32768 to 32767. To enter **`PWM`** values above 32767 use the following map:

PWM value:	0	1	2	…	32767	32768	32769	32770	…	65533	65534	65535
Integer value:	0	1	2	…	32767	-32768	-32767	-32765	…	-3	-2	-1

Downloaded from Elcodis.com electronic components distributor

Page 188 · Uart "`final static boolean invert`"	
Problem	Labeled incorrectly/Elaboration
Explaination	The title "**`final static boolean invert`**"
Solution	Should read: **`final static boolean dontInvert`** *The explaination for dontInvert has been expanded. The new explaination is as follows:* **`final static boolean dontInvert`** – selects non-inverted mode. Non-inverted mode allows you to connect the Javelin Stamp to an RS232 device. To do this you will need to boost the Javelin's TTL signal (0/5 V) to ±12 V as required by the RS232 specifications. This can be accomplished by using a MAX232 or the SP237 Uart transceivers. Either of these transceivers will invert the TTL signal as it boosts them to ±12 V, which is why we use the non-inverted mode. The Javelin Stamp Demo board has an SP237 that you can use by connecting the I/O pins of the Javelin Stamp to the 8-socket COM header (X4) on the Demo Board (See Chapter 4, figure 4.8b). The figure below shows you this as a 2-wire connection, without flow control. **Figure** Uart Not-Inverted  **Important** The pins on the male/female DB-9 connectors are different. They are mirror images of each other; care is needed when making these connections that you are connecting to the appropriate pin.

Downloaded from Elcodis.com electronic components distributor

Page 188 · Uart "`final static boolean dontInvert`"	
Problem	Labeled incorrectly/Elaboration
Explaination	The title "**`final static boolean dontInvert`**"
Solution	Should read: **`final static boolean invert`** *The explaination for dontInvert has been expanded. The new explaination is as follows:* **`final static boolean invert`** – selects inverted mode. Inverted mode allows you to connect to a computer's RS232 port without using a MAX232 or an SP237 RS232 Uart transceiver. This can be accomplished by using a 22 kΩ resistor to connect the Javelin Stamp I/O pin (that you are using as the receiver) to pin #3 on a 9-pin serial port (DB-9). This will allow you to receive data. To send data to a PC, connect a Javelin Stamp I/O pin to pin #2 of the serial port on your computer. This method will create a voltage that is not to the Uart specifications; some receivers do not accept this nonstandard voltage. If this is your situation you will need to use a Uart transceiver in non-inverted mode. The figure below shows you this as a 2-wire connection, without flow control. **Figure** Uart Inverted 

Downloaded from Elcodis.com electronic components distributor

Downloaded from Elcodis.com electronic components distributor

Corrected Index

-- (Decrement), 92, 94, 131
- (Subtraction), 92, 94, 132
! (Boolean Invert), 92, 134
!= (Not Equal to), 92, 133
% (Modulus), 92, 132
& (Bitwise AND), 92, 95, 134
&& (Logical AND), 92, 95
( ) (Parentheses), 93, 132
*** (Multiplication), 93, 94, 132**
***/ (Multi-line Remark, closing), 95**
/ (Division), 94, 132
/* (Multi-line Remark, opening), 95
/ (JavaDoc Remark), 95**
// (Remark), 95
; (Semicolon), 87
?: (Conditional), 92, 135
[ ] (Square Brackets), 93, 130-131
^ (Bitwise XOR), 92, 134
{ } (Curly Braces), 88–89
| (Bitwise OR), 92, 134
|| (Logical OR), 92
~ (Bitwise Invert), 92, 134
+ (Addition), 92, 93, 94, 132
++ (Increment), 92, 94, 131
< (Less Than), 92, 133
<< (Left-Shift), 92, 133
<= (Less Than Equal to), 92, 133
== (Equal to), 92
> (Greater Than), 92, 133
>= (Greater Than Equal to), 92, 133
>> (Right-Shift, Signed Extension), 92, 132
>>> (Right-Shift), 92, 132

- A -

abstract, 114
Abstraction, 109
ADC, 159-1260
Analog to Digital, 159-160
Arrays, 103–5, 193
ASCII, 41

- B -

Base
Hexadecimal, 91
Octal, 91
boolean, 89, 115, 149
break, 40, 96, 115
Button, 160
byte, 89, 115

- C -

Cache, 113
Calculations, 34
carry, 164
case, 97, 116
case sensitive, 32
cast, 107, 115-116, 131
catch, 111, 116, 129-130, 150
char, 41, 89, 116
Checked Exceptions, 111
class
definition, 31
Library, 48
Classes, 97–100, 116
Basic Type, 107
Clone, 105
Constructors, 107
DS1620, 59
Equals, 105
Extending. *See extends*
HashCode, 105
import, 113
Integer, 109
Member, 98
Relationships, 140
toString, 105
Virtual Peripherals. *See Virtual Peripherals*
wrapper. *See wrapper*
CLASSPATH, 86, 112
Clock, 174
COM Ports, 30

Downloaded from Elcodis.com electronic components distributor

const, 136
constant, 32
Constants. ***See final***
construct, 99
continue, 40, 96, 116
count, 164
Counter, 169-170
CPU, 163
Message, 42–44

- D -
DAC, 59, 181
Debug, 78–83, 185
delay, 164, 193
Digital to Analog, 181
do, 37, 115, 116
double, 136
DS1620, 8, 59-64

- E -
Editor, 83
EEPROM, 182
else, 95, 116
Encapsulation, 138
Errors, 76–78, 110, 150
Escape Sequences, 91
Exception Handling, 110, 150
Exceptions, 146
Expressions, 91–95
extends, 105–108, 117

- F -
final, 32, 106, 117
finally, 117, 129
float, 136, 193
for, 39–40, 88, 95, 115, 117

- G -
Garbage Collection, 100, 191
getMessage, 150
Global Options, 71–72
goto, 136

- H -
Hardware, 11–14
EEPROM, 183
Hexadecimal, 91

- I -
I/O Pins, 163, 164, 167-169, 170, 174-175
IDE, 2, 15–17, 71-86
Call Stack, 80
CLASSPATH, 84
Compile, 76
Debug, 27–28, 76
Step Into, 81
Editor, 83
Installation, 15–17
Link, 76
Memory Usage, 80
Menubars, 83
Packages, 84–85
Program, 76
Projects, 85-86
Resume Debug, 76
Starting a Project, 72–76
Toolbars, 83
if, 35–37, 88–89, 95, 119
implements, 136
import, 113, 120
Inheritance, 105
Inputs, 156, 160-163
installVP, 164
instanceof, 135
int, 89, 121, 192
interface, 136

- J -
Java Differences, 87, 193-197
break, 96

Downloaded from Elcodis.com electronic components distributor

Java Differences *(continued)*
const, 136
continue, 96
double, 136
Floating Point, 193
for, 95
Garbage Collection, 191
goto, 136
if, 95
implements, 136
int, 192
interface, 136
long, 136
Loops, 192
native, 136
objects, 192
static, 191
StringBuffer, 191, 192
Strings, 104, 192
synchronized, 136
Threads, 191
transient, 136
Unicode, 193
volatile, 136
while, 97
Javelin Stamp
Architecture, 3–4
Demo Board, 7
Hardware, 1, 4, 5
Heavy Loads, 57
I/O pins, 12–14
Power Supply, 13–14
Starter Kit, 5

- K -

Keywords, 114–130

- L -

Libraries, 193
Library Class, 48
Lists, 157-158
long, 136
Loops, 37–40
break, 40
continue, 40
do, 37
for, 39–40, 88, 95
while, 37–40, 88

- M -

Math, 151, 164
message, 42–44, 166
Methods, 99
Constructors, 99
equals, 99
Returning a Value, 99
void, 99

- N -

nap, 166, 172, 173
native, 136
new, 100, 121
null, 121

- O -

Object Oriented, 140, 147
Objects, 98-100, 137-138, 151
Arrays, 103–5
casting. *See cast*
keywords. *See the key word*
new, 100
Pointers, 101-102
Strings, 104–5
substring, 105
this, 102
Timer, 146
UART(s), 137, 146
Octal, 91
Online Resources, 114

Downloaded from Elcodis.com electronic components distributor

Operators
Basic Java Operators, 92
Order of Operations, 93
Order of Operations, 93
override, 106

- P -
Packages, 84–85, 112, 122-123
CLASSPATH. *See CLASSPATH*
Pointers, 101–2
Polymorphism, 106, 139
print, 41–42, 155-156
println, 41–42, 155-156
PrintStream, 155-156
private, 106, 123
protected, 106, 123
public, 106, 123
pulseIn, 167
pulseOut, 168
PWM, 58, 146, 184-185

- R -
Random, 156
RC circuit, 170
RC timing, 170
rcTime, 169-171
readPin, 171-173
readPort, 172
removeVP, 173
return, 99, 124

- S -
Serial Port, 30
setInput, 174
shiftIn, 174
short, 89, 125
static, 33, 125
StringBuffer, 42, 100, 104, 154, 192
Strings, 104–5, 152, 192
StringBuffer. *See StringBuffer*
Strings *(continued)*
substring, 105
super, 106-108, 126
switch, 97, 115, 127
synchronized, 136
System.out
print, 41–42, 154-156
println, 41–42, 154-156

- T -
Template, 72–76
Terminal, 185
this, 127
Threads, 191
throw(s), 111, 128, 150
throwIt, 152
Timer, 146, 187
toString, 109
transient, 136
try, 110-111, 128-130, 150
Type, 131
Garbage Collection, 191
int, 192

- U -
UART(s), 64, 146, 188
Unicode, 202
URL's, 114

- V -
Variables, 32–34
boolean, 89
byte, 89
Calculations, 34
char, 89
Declaration, 89-91
final, 32, 90-91
int, 89
short, 89
static, 80

Downloaded from Elcodis.com electronic components distributor

Variables *(continued)*
static final, 91
Virtual Peripherals, xiv, 3, 145, 163, 164, 173, 184
ADC, 159
Background, 3
DAC, 59
Foreground, 3
PWM, 58, 146, 184-185
Virtual Peripherals
Timer, 187
UART(s), 64, 188void, 99, 130
volatile, 136

- W -
while, 37–40, 88, 97, 115, 130
wrapper, 109

Downloaded from Elcodis.com electronic components distributor