



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

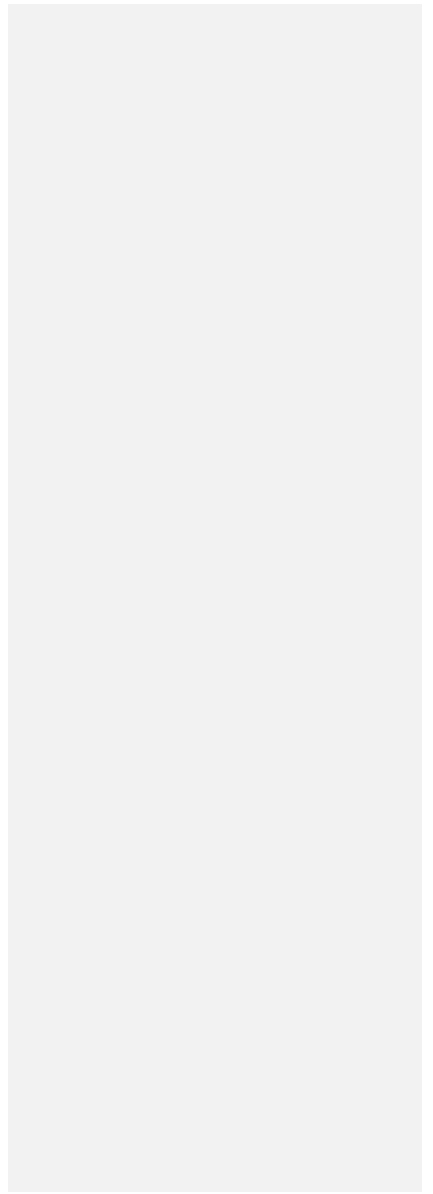
Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



# **Javelin Stamp Manual**

**Version 1.1**



## **WARRANTY**

Parallax Inc. warrants its products against defects in materials and workmanship for a period of 90 days from receipt of product. If you discover a defect, Parallax Inc. will, at its option, repair or replace the merchandise, or refund the purchase price. Before returning the product to Parallax, call for a Return Merchandise Authorization (RMA) number. Write the RMA number on the outside of the box used to return the merchandise to Parallax. Please enclose the following along with the returned merchandise: your name, telephone number, shipping address, and a description of the problem. Parallax will return your product or its replacement using the same shipping method used to ship the product to Parallax.

## **14-DAY MONEY BACK GUARANTEE**

If, within 14 days of having received your product, you find that it does not suit your needs, you may return it for a full refund. Parallax Inc. will refund the purchase price of the product, excluding shipping/handling costs. This guarantee is void if the product has been altered or damaged. See the Warranty section above for instructions on returning a product to Parallax.

## **COPYRIGHTS AND TRADEMARKS**

This documentation is copyright 2005 by Parallax Inc. By downloading or obtaining a printed copy of this documentation or software you agree that it is to be used exclusively with Parallax products. Any other uses are not permitted and may represent a violation of Parallax copyrights, legally punishable according to Federal copyright or intellectual property laws. Any duplication of this documentation for commercial uses is expressly prohibited by Parallax Inc.

BASIC Stamp, Stamps in Class, Board of Education, Boe-Bot SumoBot, SX-Key and Toddler are registered trademarks of Parallax, Inc. If you decide to use registered trademarks of Parallax Inc. on your web page or in printed material, you must state that "(registered trademark) is a registered trademark of Parallax Inc." upon the first appearance of the trademark name in each printed document or web page. HomeWork Board, Parallax, and the Parallax logo, are trademarks of Parallax Inc. If you decide to use trademarks of Parallax Inc. on your web page or in printed material, you must state that "(trademark) is a trademark of Parallax Inc.", upon the first appearance of the trademark name in each printed document or web page. Other brand and product names are trademarks or registered trademarks of their respective holders.

**ISBN #####**

## DISCLAIMER OF LIABILITY

Parallax Inc. is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, or any costs of recovering, reprogramming, or reproducing any data stored in or used with Parallax products. Parallax Inc. is also not responsible for any personal damage, including that to life and health, resulting from use of any of our products. You take full responsibility for your BASIC Stamp application, no matter how life-threatening it may be.

## INTERNET DISCUSSION LISTS

We maintain active web-based discussion forums for people interested in Parallax products. These lists are accessible from [www.parallax.com](http://www.parallax.com) via the Support → Discussion Forums menu. These are the forums that we operate from our web site:

- BASIC Stamps – This list is widely utilized by engineers, hobbyists and students who share their BASIC Stamp projects and ask questions.
- Stamps in Class<sup>®</sup> – Created for educators and students, subscribers discuss the use of the Stamps in Class educational program in their courses. The list provides an opportunity for both students and educators to ask questions and get answers.
- Parallax Educators – Exclusively for educators and those who contribute to the development of Stamps in Class. Parallax created this group to obtain feedback on our curricula and to provide a forum for educators to develop and obtain Teacher's Guides.
- Translators – The purpose of this list is to provide a conduit between Parallax and those who translate our documentation to languages other than English. Parallax provides editable Word documents to our translating partners and attempts to time the translations to coordinate with our publications.
- Robotics – Designed exclusively for Parallax robots, this forum is intended to be an open dialogue for a robotics enthusiasts. Topics include assembly, source code, expansion, and manual updates. The Boe-Bot<sup>®</sup>, Toddler<sup>®</sup>, SumoBot<sup>®</sup>, HexCrawler and QuadCrawler robots are discussed here.
- SX Microcontrollers and SX-Key – Discussion of programming the SX microcontroller with Parallax assembly language SX-Key<sup>®</sup> tools and 3rd party BASIC and C compilers.
- Javelin Stamp – Discussion of application and design using the Javelin Stamp, a Parallax module that is programmed using a subset of Sun Microsystems' Java<sup>®</sup> programming language.

## ERRATA

While great effort is made to assure the accuracy of our texts, errors may still exist. If you find an error, please let us know by sending an email to [editor@parallax.com](mailto:editor@parallax.com). We continually strive to improve all of our educational materials and documentation, and frequently revise our texts. Occasionally, an errata sheet with a list of known errors and corrections for a given text will be posted to our web site, [www.parallax.com](http://www.parallax.com). Please check the individual product page's free downloads for an errata file.



## Table of Contents

<b>PREFACE</b> .....	<b>VII</b>
MANUAL ORGANIZATION .....	VII
JAVA PROGRAMMERS – READ THIS .....	VIII
BASIC STAMP ENTHUSIASTS – READ THIS .....	IX
MANUAL CONVENTIONS .....	IX
RESOURCES AND TECHNICAL SUPPORT .....	X
FREE DOWNLOADS FROM WWW.JAVELINSTAMP.COM.....	XI
ACKNOWLEDGEMENTS .....	XI
<b>1: INTRODUCTION</b> .....	<b>1</b>
THE JAVELIN STAMP AND ITS FEATURES .....	1
PROGRAMMING LANGUAGE - JAVA™ FOR THE JAVELIN STAMP .....	2
JAVELIN STAMP INTEGRATED DEVELOPMENT ENVIRONMENT .....	2
VIRTUAL PERIPHERALS .....	3
<i>Background VPs</i> .....	3
<i>Foreground VPs</i> .....	3
HOW THE JAVELIN STAMP WORKS .....	4
JAVELIN STAMP HARDWARE .....	5
EQUIPMENT AND SYSTEM REQUIREMENTS .....	6
USEFUL HARDWARE .....	7
<b>2: JAVELIN QUICK START</b> .....	<b>11</b>
HARDWARE SETUP.....	11
INSTALLING THE JAVELIN STAMP IDE .....	16
RUNNING THE JAVELIN STAMP IDE AND LOADING A TEST PROGRAM .....	19
DEBUGGING ENVIRONMENT .....	22
ONLINE HELP .....	26
I/O EXAMPLE .....	27
DID THAT WORK? – TROUBLE SHOOTING .....	29
WHERE TO NEXT?.....	32
<b>3: BEGINNERS GUIDE TO EMBEDDED JAVA PROGRAMMING</b> .....	<b>35</b>
THE CLASS WRAPPER AND MAIN METHOD .....	35
DECLARING CONSTANTS, VARIABLES, AND ARRAYS .....	36
PERFORMING CALCULATIONS .....	39
MAKING DECISIONS .....	40
REPETITIVE OPERATIONS .....	43
DISPLAYING MESSAGES FROM THE JAVELIN .....	47
SENDING MESSAGES TO THE JAVELIN.....	49

## Table of Contents

CREATING A METHOD .....	51
CREATING AND USING A LIBRARY CLASS.....	55
<b>4: APPLICATION EXAMPLES – CIRCUITS AND PROGRAMS.....</b>	<b>59</b>
CIRCUITS AND EXAMPLE CODE.....	59
ABOUT SOLDERLESS BREADBOARDS .....	59
PUSHBUTTON AND LED REVISITED .....	61
DIGITAL TO ANALOG CONVERSION .....	63
ANALOG TO DIGITAL CONVERSION .....	64
MEASURING RESISTIVE AND CAPACITIVE ELEMENTS.....	64
CONTROLLING A SERVO WITH A BACKGROUND PWM OBJECT .....	66
COMMUNICATING WITH PERIPHERAL ICs .....	69
COMMUNICATING WITH OTHER COMPUTERS.....	74
COMMUNICATING WITH PERIPHERAL DEVICES.....	78
<b>5: USING THE JAVELIN STAMP IDE.....</b>	<b>81</b>
STARTING THE IDE .....	81
SETTING GLOBAL OPTIONS.....	81
STARTING A PROJECT.....	82
BUILDING YOUR PROGRAM.....	86
DEALING WITH ERRORS .....	87
USING THE DEBUGGER TO LOOK INSIDE THE JAVELIN.....	89
AN EXAMPLE DEBUGGING SESSION.....	92
EDITING TEXT .....	94
TOOLBARS AND MENUBARS .....	95
CLASS PATH CONSIDERATIONS.....	95
WORKING WITH PACKAGES .....	96
WORKING WITH PROJECTS .....	97
<b>6: JAVELIN STAMP PROGRAMMERS REFERENCE.....</b>	<b>99</b>
JAVA DIFFERENCES .....	99
GETTING STARTED.....	99
VARIABLES, TYPES, AND CONSTANTS .....	101
<i>Constants</i> .....	103
<i>Number Bases</i> .....	104
<i>Expressions</i> .....	104
<i>Special Operators</i> .....	106
<i>Comments</i> .....	108
<i>Control Flow</i> .....	108
<i>Classes and Objects</i> .....	111

## Table of Contents

<i>Methods and Parameters</i> .....	113
<i>Where are the Pointers?</i> .....	115
<i>Arrays</i> .....	118
<i>Strings</i> .....	119
<i>Extending Classes</i> .....	120
<i>Basic Type Classes</i> .....	124
<i>Numeric Conversions</i> .....	125
<i>Statics</i> .....	125
<i>Abstraction</i> .....	126
<i>Exceptions</i> .....	126
<i>Packages and CLASSPATH</i> .....	129
ONLINE RESOURCES .....	131
JAVELIN STAMP KEYWORD REFERENCE .....	131
<i>abstract</i> .....	131
<i>boolean</i> .....	132
<i>break</i> .....	132
<i>byte</i> .....	132
<i>case</i> .....	133
<i>catch</i> .....	133
<i>char</i> .....	133
<i>class</i> .....	133
<i>continue</i> .....	133
<i>default</i> .....	134
<i>do</i> .....	134
<i>else</i> .....	134
<i>extends</i> .....	134
<i>final</i> .....	135
<i>finally</i> .....	135
<i>for</i> .....	135
<i>if</i> .....	137
<i>import</i> .....	138
<i>int</i> .....	139
<i>new</i> .....	140
<i>null</i> .....	140
<i>package</i> .....	140
<i>private, protected, public</i> .....	141
<i>return</i> .....	143
<i>short</i> .....	143
<i>static</i> .....	143
<i>super</i> .....	145



## Table of Contents

<i>switch</i> .....	145
<i>this</i> .....	146
<i>throw, throws</i> .....	147
<i>try</i> .....	147
<i>void</i> .....	149
<i>while</i> .....	149
JAVELIN STAMP OPERATOR REFERENCE.....	150
<i>[]</i> .....	150
<i>++</i> , <i>--</i> .....	150
<i>(type)</i> .....	150
<i>+</i> , <i>-</i> , <i>*</i> , <i>/</i> , <i>%</i> , <i>()</i> .....	151
<i>&lt;&lt;</i> , <i>&gt;&gt;</i> , <i>&gt;&gt;&gt;</i> .....	152
<i>&lt;</i> , <i>&gt;</i> , <i>&lt;=</i> , <i>&gt;=</i> , <i>==</i> , <i>!=</i> .....	153
<i>&amp;</i> , <i> </i> , <i>^</i> .....	153
<i>&amp;&amp;</i> , <i>  </i> .....	154
<i>~</i> , <i>!</i> .....	154
<i>?:</i> .....	154
<i>instanceof</i> .....	155
UNUSED KEYWORDS.....	156
<i>Unsupported Reserved Words</i> .....	156
<b>7: WORKING WITH OBJECTS</b> .....	<b>157</b>
WHAT'S AN OBJECT?.....	157
<i>Encapsulation</i> .....	159
<i>Polymorphism</i> .....	160
<i>Class Relationships</i> .....	161
<i>An Object Oriented Example</i> .....	161
<i>Decoupling the Code</i> .....	164
<i>Virtual Peripherals</i> .....	167
<i>A Timer Example</i> .....	168
<i>Object-Oriented Opportunity</i> .....	170
<b>8: OBJECT REFERENCE</b> .....	<b>173</b>
THE <i>JAVA.LANG</i> PACKAGE.....	173
<i>Boolean</i> .....	173
<i>Error</i> .....	174
<i>Exception</i> .....	174
<i>IndexOutOfBoundsException</i> .....	175
<i>Math</i> .....	175
<i>NullPointerException</i> .....	175

## Table of Contents

<i>Object</i> .....	176
<i>OutOfMemoryError</i> .....	177
<i>RuntimeException</i> .....	177
<i>String</i> .....	177
<i>StringBuffer</i> .....	180
<i>System</i> .....	181
<i>Throwable</i> .....	181
THE <i>JAVA.IO</i> PACKAGE .....	181
THE <i>JAVA.UUTIL</i> PACKAGE .....	182
<i>Random</i> .....	182
THE <i>STAMP.UUTIL</i> PACKAGE .....	182
<i>Expect</i> .....	183
<i>List</i> .....	183
<i>LinkedList</i> .....	183
LINKEDLISTITEM .....	184
<b>9: JAVELIN STAMP HARDWARE REFERENCE .....</b>	<b>185</b>
ADC .....	185
BUTTON .....	186
CPU .....	190
<i>carry</i> .....	191
<i>count</i> .....	191
<i>delay</i> .....	192
<i>installVP</i> .....	192
<i>message</i> .....	193
<i>nap</i> .....	193
<i>pulseIn</i> .....	194
<i>pulseOut</i> .....	196
<i>rcTime</i> .....	196
<i>readPin</i> .....	199
<i>readPort</i> .....	200
<i>removeVP</i> .....	201
<i>setInput</i> .....	202
<i>shiftIn</i> .....	202
<i>shiftOut</i> .....	206
<i>writePin</i> .....	210
<i>writePort</i> .....	210
DAC .....	211
EEPROM .....	212
MEMORY .....	215

## Table of Contents

---

PWM .....	215
TERMINAL .....	216
TIMER .....	219
UART .....	220
<b>10: TECHNICAL DETAILS .....</b>	<b>225</b>
SUMMARY OF JAVA DIFFERENCES .....	225
<i>Single Thread</i> .....	225
<i>No Garbage Collection</i> .....	225
<i>Subset of Primitive Data Types</i> .....	226
<i>Subset of Java Libraries</i> .....	227
<i>Strings are ASCII</i> .....	229
<i>No Interfaces</i> .....	229
<i>One Dimensional Arrays</i> .....	229
UNDERSTANDING THE JAVELIN STAMP'S MEMORY MANAGEMENT .....	230
MEMORY AND VARIABLE TYPES .....	232
<b>INDEX .....</b>	<b>233</b>

### Manual Organization

This manual was written under the assumption that the reader's level of experience could be anywhere between beginner and advanced embedded Java™ aficionado. We recommend that you start from the beginning and work your way through this manual sequentially, especially if you are new to both circuits and Java. Make sure to try all the examples and understand how they work before moving on to the next. For those of you who do not fall at either end of the spectrum, below is a condensed table of contents with comments regarding the intended audience and uses of each chapter.

#### **Preface**

General information - discusses Javelin Stamp's features, this manual's format and conventions, resources and acknowledgements.

#### **1: Introduction**

General information - about the Javelin, its uses, equipment it can be used with, specifications, software, etc.

#### **2: Javelin Quick Start**

Recommended for all – includes step by step instructions for software installation, hardware setup, trouble shooting, a couple of example programs, an example circuit, and a software tour.

#### **3: Beginners Guide to Embedded Java™ Programming**

Recommended for Java newcomers and BASIC Stamp users - if you've never programmed in Java before, read this, and try the examples!

#### **4: Application Examples – Circuits and Programs**

Recommended for embedded newcomers and BASIC Stamp users – provides good examples for BASIC Stamp users to make the transition to Java based hardware design, and helps those new to circuit based programming projects get their feet wet.

#### **5: Using the Javelin Stamp IDE**

Recommended for all – the Javelin Stamp IDE is a powerful tool with many useful features.

## Preface

---

### 6: Javelin Stamp Programmers Reference

If you are a Java programmer, pay close attention to the differences between Java for the Javelin and Java on your PC. For beginners, this is a good way to learn programming in Java.

### 7: Working with Objects

Recommended if you are still learning Java – by this point, if you were new to Java at the beginning of this manual, you are now well into the learning curve.

### 8: Object Reference

Recommended for all – whether you are an experienced Java programmer or you just finished Chapter 7, this chapter explains the Java library classes available for use with the Javelin.

### 9: Javelin Stamp Hardware Reference

Recommended for all – explains all the hardware related library classes and methods. If it has to do with a VP, a peripheral or an external circuit, the information is here.

### 10: Technical Details

Appendix material.

## Java Programmers – READ THIS

The Javelin Stamp is a small yet powerful controller that makes use of a subset of Java 1.2. The Javelin Stamp has firmware enhancements (called Virtual Peripherals or VPs) that emulate, or virtualize, hardware devices such as UARTs, timers, A/D converters, D/A converters, and more. These VP's have been painstakingly optimized, and they take the form of native methods that make it easy to interface with just about any circuit or peripheral device. Many of these firmware features are similar to those that lead the BASIC Stamp's popularity, and others have long been on BASIC Stamp users' wish lists.

The flip side of the Virtual Peripheral firmware features is that they have been incorporated into the Javelin Stamp at the expense of Java purity. You will find the experience of developing applications with the Javelin Stamp uniquely different from developing applications on a PC. To get to the rewards of a rapid prototype of your product design or project with minimal stumbling, we recommend above all that you try the many programming and circuit examples in this text. Before getting started on the examples, take a few minutes

skill

**Comment [1]:** 10: Internet Programming Advanced example – connecting to UDP servers. This section needs to be added back.

to review the reading list below. It will acquaint you with the scope of Javelin Stamp projects and help you avoid some of the programming pitfalls you might otherwise encounter.

Suggested reading for Java Programmers:

Section	Page
The Javelin Stamp and Its Features	1
Programming Language - Java™ for the Javelin Stamp	2
Summary of Java Differences	225
Javelin Stamp Integrated Development Environment	2 3
Virtual Peripherals	
Background VPs	3
Foreground VPs	3
How the Javelin Stamp Works	4

### **BASIC Stamp Enthusiasts – READ THIS**

As with the Java Programmers who were addressed in the previous section, programming the Javelin Stamp is also likely to be very different from what you, the BASIC Stamp Enthusiast, are expecting. This manual has LOTS of example programs and circuits to help you transition from PBASIC to the Java subset used to program the Javelin Stamp. Especially if you are unfamiliar with Java, we strongly recommend that you work through the examples in this text sequentially. The majority of this manual's organization was established with you in mind, so, if you have not already done so, please take a look at the Manual Organization section at the beginning of this preface. If you are like the rest of us at Parallax, you probably can't wait to get started, so have fun with Chapter 2: Javelin Quick Start.

### **Manual Conventions**

Below is a list of typographical conventions used in this manual:

**Monospaced** is used for:

- Words that are part of the language syntax when they are part of a sentence.
- Fragments of programs. The code snippet below is an excerpt from a program, but it cannot be run on its own. It has to appear in either a complete program or a complete class file, both of which are discussed next:

## Preface

---

```
System.out.println("Not a complete program.");
```

A gray box is used for:

- Complete programs that can be entered into the Javelin Stamp IDE and executed on a Javelin Stamp, for example:

```
import examples.manual_v1_0.*;
public class CompleteProgram{
    public static void main() {
        CompleteClassFile example = new CompleteClassFile();
        System.out.println("Now, it's in a complete program.");
        example.displaySameMessageAgain();
    }
}
```

- Complete class files that can be instantiated by other programs. Here is an example:

```
package examples.manual_v1_0;
public class CompleteClassFile {
    public static void displaySameMessageAgain() {
        System.out.println("Now, it's in a complete class file");
    }
}
```

### Resources and Technical Support

The inside cover of this manual has three sections pertaining to resources:

- Internet Access
- Internet Javelin Stamp Discussion List
- Contacting Parallax

Follow the Tech Support link at [www.javelinstamp.com](http://www.javelinstamp.com) for the latest in tech support contact info, discussion group links, manual errata, answers to frequently asked questions, and more!

---

### **Free Downloads from [www.javelinstamp.com](http://www.javelinstamp.com)**

You can always get the latest revisions and updates of the following from [www.javelinstamp.com](http://www.javelinstamp.com):

- Javelin Stamp Manual
- Javelin Stamp IDE
- Application Notes
- Library Files

### **Acknowledgements**

Chris Waters and Celsius Research provided the Javelin Stamp firmware and reference design. This manual was developed using information and research provided by Al Williams Consulting. Each and every employee at Parallax has made some contribution to the Javelin Stamp project, so as always, thanks to the entire Parallax staff.





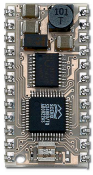


Figure 1.1 Javelin (top view)

## The Javelin Stamp and Its Features

The Javelin Stamp is a single board computer that's designed to function as an easy-to-use programmable brain for electronic products and projects. As shown in Figure 1.1, it's about the size and shape of a commemorative postage stamp. It is programmed using software on a PC and a subset of Sun Microsystems Java® programming language. After the program is downloaded to the Javelin, it can run the program without any further help from the PC. The Javelin can be programmed and re-programmed up to one million times.

We hope you enjoy working with your new Javelin Stamp as much as we have while preparing this manual. The Javelin Stamp is somewhat of a departure from Parallax's BASIC Stamps. Most notably, the Javelin is programmed using a subset of the Java programming language. Some of the other features that set the Javelin apart from BASIC Stamps are:

- The instruction codes for the Javelin are fetched and executed from a parallel SRAM instead of a serial EEPROM.
- The Javelin has 32k of RAM/program memory with a flat architecture. No more program banks, and no more tight squeezes with variable space.
- The Javelin has built in Virtual Peripherals (VPs) that take care of serial communication, pulse width modulation and tracking time in the background.
- Serial communication is buffered as a background process. When writing programs, all you have to do is periodically check the buffer.
- The Javelin Stamp Integrated Development Environment (Javelin Stamp IDE) software is a significant departure from a simple Editor and messages window combination. When used with the Javelin connected to a PC by a serial cable, this software can be used as a highly integrated in-circuit debugging system that allows you to run code, set breakpoints and view variable values, memory usage, I/O pin states and more. There is also no need for emulators; the Javelin can be placed directly into the circuit and debugged there.
- Delta-sigma A/D conversion.
- D/A conversion is accomplished in the background as a continuous pulse train delivered by an I/O pin. The pulse width modulation VP can

Aristides 4/28/05 11:35 AM

Formatted Table

skill

**Comment [2]:** The Javelin Stamp takes Parallax's BASIC stamps into a new direction. The Javelin Stamp thrusts Parallax's BASIC stamps into a new direction. Parallax's BASIC Stamps take a new direction with the Javelin. The Javelin Stamp is an exciting new microcontroller.

## 1: Introduction

also be used for generating pulse trains, frequencies, and D/A conversions in the background while your foreground code is free to perform other tasks

Those of you who appreciate the simplicity and ease of use of the BASIC Stamps need not worry; the Javelin Stamp has many features that BASIC Stamp users have come to depend on in their projects and designs. Here is a list of features built into the Javelin with BASIC Stamp users in mind:

- Synchronous serial communication (shiftIn/shiftOut)
- The ability to both send and measure discrete pulses (pulseIn/pulseOut)
- Frequency counting (count)
- Simple and intuitive methods for reading from and writing to I/O pins
- Measurement of RC charge and discharge times (rcTime)

BASIC Stamps have been used for everything from lessons in basic computer programming and electronics, all the way up to aerospace subsystem designs. We expect to see the Javelin used in a similar manner. However, by making use of the Javelin's new features, it can be used to tackle some more demanding designs that used to require larger processors.

### Programming Language - Java™ for the Javelin Stamp

The Javelin's programming language supports many of the Java languages most useful features:

- Object Orientation - Inheritance, method overloading, polymorphism and static initializers.
- Exceptions - Try-catch-finally blocks and the ability to catch exceptions with a super-class.
- Strings – Programmed using many familiar Java commands.
- Custom Library Support - For many popular peripherals such as LCDs, temperature, AD, communication ICs, and common Internet protocols such as ARP, UDP, and PPP.

<b>Java Differences</b>	There are some differences between writing applications for your PC using Java 1.2 and the subset of Java used by the Javelin. Experienced Java programmers should consult the Summary of Java Differences section in Chapter 10.
-------------------------	---

### Javelin Stamp Integrated Development Environment

sdill

**Comment [3]:** Check with Jim on this

## 1: Introduction

Javelin Stamp Integrated Development Environment (Javelin Stamp IDE) offers the features that you would commonly expect from a source-level debugger:

- Multiple breakpoints
- Stack backtrace
- Inspection of all variables and objects, both static and dynamically allocated
- Single-step, run, stop, reset
- Built-in bi-directional serial message terminal for `System.out.println()` and `Terminal.getChar()` type debugging

The Javelin Stamp IDE is introduced in Chapter 2, and then discussed in more detail in Chapter 5. This IDE makes real-time debugging so easy that a PC emulator is completely unnecessary. It is just as easy to develop and debug on the Javelin module itself.

### Virtual Peripherals

The Javelin Stamp firmware supports a variety of Virtual Peripherals (VPs). The VPs are separated into two separate categories, foreground and background. The background processes allow you to create UARTs, pulse trains, and a timer. Once created, background VP objects run independently from the program. Since time-sensitive tasks are taken care of by the VPs in the background, designs that used to be difficult become easy. For example, serial communication does not stop just because the Javelin is measuring the duration of an incoming pulse. The programmer simply needs to periodically check the serial buffer in the foreground code. Below is a list of background and foreground VPs.

#### Background VPs

- UART (Full duplex, HW flow control, buffered)
- PWM
- 32-bit Timer
- 1-bit DAC
- Delta/Sigma ADC

#### Foreground VPs

- Pulse count
- Pulse width measurement
- Pulse generation
- RC Timer
- SPI master

## 1: Introduction

---

*These Virtual Peripherals are built into the Javelin Stamp's firmware. Although you can write library classes that make use of these VPs, the VPs themselves cannot be modified or rewritten.*

### How the Javelin Stamp Works

The Javelin Stamp's hardware architecture is shown in Figure 1.2. Programming and debugging is done via communication with the serial port. The COM circuit takes care of the voltage conversions necessary for a TTL device to talk with an RS232 port. The Java interpreter processes all serial port/COM circuit information. Whether it's byte codes, debugging data or serial messages, the interpreter processes the data and decides what to do with it.

When a program is downloaded, the interpreter buffers the program bytecodes and writes them to the EEPROM. Upon reset (or a power interruption), all the Javelin Stamp's I/O pins are set to input. The interpreter copies the bytecodes to the SRAM, then starts fetching bytecodes from the SRAM and executing them. The bytecode instructions can be executed very rapidly because all data is transmitted along parallel data busses instead of synchronous serial lines. A typical fetch and execute cycle involves a couple of read/write cycles. During a read/write cycle, the interpreter loads some of the 15 bit address information into an address latch and writes the other portion directly to the SRAM. When the SRAM address is set, then the data is read or written by the interpreter as needed.

The Javelin's internal voltage regulation is done using a switching regulator. The switching regulator runs cooler and is significantly more efficient than a linear regulator. It accepts voltages between 6 and 24 V, and makes 5 V available for the Javelin Stamp with a total current budget of 150 mA. The passive components including the input and output capacitors, switching diode and inductor are on the top side, and the switching IC is on the bottom side of the board next to the EEPROM. The switching IC monitors the output voltage and adjusts the switching duty cycle to the passive components to maintain a constant 5 V output.

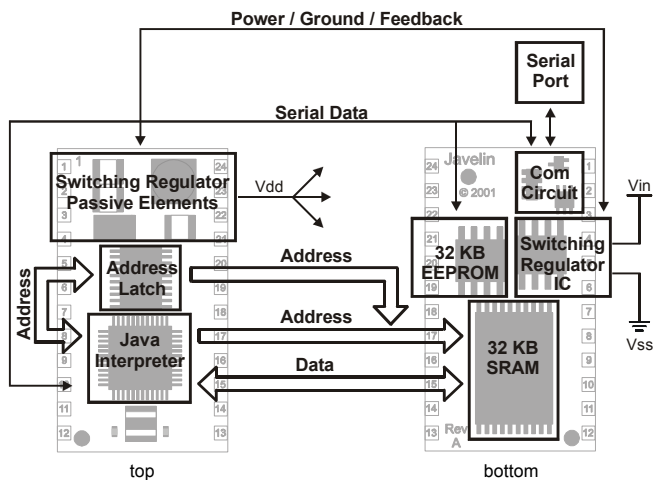


Figure 1.2 Javelin Block Diagram

### Javelin Stamp Hardware

Table 1.1 shows the Javelin Stamp’s specifications. Note that the onboard voltage regulator can accept between 6 and 24 V<sub>DC</sub> and output up to 150 mA of current. Since the Javelin consumes approximately 60 mA, you have 90 mA available for other uses. Keep in mind that if you are utilizing the full 60 mA of total I/O pin source/sink that only 30 mA is left over for powering peripheral devices using the Javelin’s Vdd pin. On the other hand, if all the I/O pins are being used for input, 90 mA can be used drawn from the Javelin’s voltage regulator output (Vdd) for peripherals. If in doubt, use an external 5 V regulator for your peripherals.

Table 1.1: Javelin Hardware Specifications

Attribute	Value
Module Footprint	24-pin DIP module
Package Measurements (LxWxH)	1.2"x0.6"x0.4" (3.0x1.5x1.0 cm)
Operating Environment	0° - 70° C (32° - 158° F)
Microcontroller	Ubicom SX48AC

## 1: Introduction

RAM	32 kilobytes
EEPROM	32 kilobytes
Number of I/O pins	16
Voltage Supply	6 – 24 VDC (unregulated) - or - 5 VDC (regulated)
Voltage regulator current output	$0 < I_{out} < 180 \text{ mA}$
Current Consumption	60 mA / 13 mA nap
Sink/Source Current per I/O	30 mA / 30 mA
Sink/Source Current per module	60 mA / 60 mA per 8 I/O pins
Sink/Source Current per Bank Pins (0 – 7) and (8 - 15)	30 mA / 30 mA
Windows Editor/Debugger	Javelin Stamp IDE

### Equipment and System Requirements

To run the IDE and program the Javelin, you will need an IBM PC or compatible computer with the following:

- Windows 95, 98, ME, 2000, or XP.
- A CDROM or Internet connection.
- An available 9-pin serial port
  - Or – A USB port with an approved USB to serial adaptor. See [www.javelinstamp.com](http://www.javelinstamp.com) for information on products that have been tested and approved.
  - Or – A 25-pin serial port with a 25 to 9-pin adaptor.

The Javelin Stamp Starter Kit is discussed in detail in the following section: *Useful Hardware*. If you do not have a Javelin Stamp Starter kit, you will need to acquire at least the following.

- Recommended DC Power Supply: 7.5 VDC, 1000 mA 2.1 mm, center-positive  
Acceptable battery/DC Power Supply values range between 6 and 24 VDC. Minimum output current rating depends on voltage. A 6 V supply can have an output current rating as low as 100 mA while higher voltage supplies may need higher output current ratings.
- Serial programming cable  
Be sure to use a straight-through serial cable or adaptor. Do not try to use a null modem cable or adaptor for downloading programs to the Javelin.
- Carrier board or serial cable and power supply connections

## ***1: Introduction***

---

Parallax makes a variety of carrier boards for BASIC Stamps. The Javelin Stamp can be powered and programmed using any of these carrier boards. You can also make your own connections for supply voltage and serial cables. See the Hardware Setup section in Chapter 2.

### **Useful Hardware**

The Javelin Stamp Starter kit is a great way to get started, especially if this is your first adventure into Javelin based projects. Projects featured in Chapters 2, 4, and 9 make use of the carrier board and parts in this starter kit. The Javelin Stamp Demo Board is the carrier board included in the kit, and its features are shown in **Error! Reference source not found.** and listed below.



## 1: Introduction

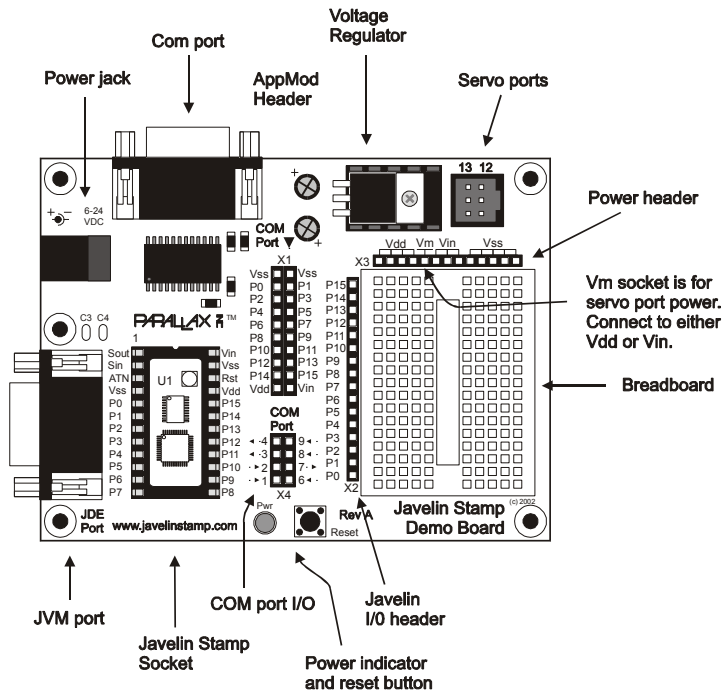


Figure 1.3 Javelin Stamp Demo Board Features

The Javelin Stamp Demo board (Error! Reference source not found.) has the following features:

- Socket for the Javelin Stamp (Labeled U1).
- JIDE port for debugging, messages, and downloading programs from the PC into the Javelin Stamp.
- A power jack that can accept input voltage ranging from 6 to 24 V<sub>DC</sub>.

## 1: Introduction

- A COM port that can be used to connect the Javelin Stamp to other computers. Alternately, you can attach a null modem adaptor to this COM port and then connect the Javelin to peripherals such as serial GPS units, mice, etc.
- Linear voltage regulator for prototype circuits.
- Small breadboard area for building, testing and prototyping circuits.
- A power header (supplied by the linear voltage regulator). This header can be used to supply circuits with power.
- A Javelin I/O header to connect your Javelin Stamp I/O pins to your circuit.
- COM Port I/O header. You can use jumper wires to connect Javelin Stamp I/O pins to the COM port I/O header. Then you can write code to communicate with another serial device such as a computer or peripheral that's connected to the COM port.
- LED power indicator (labeled PWR).
- Reset pushbutton. Press and release to restart the program from its beginning.
- A servo port for connecting and controlling servo motors.

As mentioned earlier, the circuit examples in this manual feature parts you can find in the Javelin Stamp Starter Kit. The parts are listed in Table 1.2. Table 1.3 lists parts that are also recommended but not included in the kit.

Table 1.2: Javelin Stamp Starter Kit

Quantity	Part Number	Part Description
1	550-00019	Javelin Stamp Demo Board Rev A
1	JS1-IC	Javelin Stamp Module Rev B
1	27957	Javelin Stamp Manual
1	800-00003	Serial Cable
1	800-00002	DB9 Null Modem Adapter Male to Male
1	604-00002	DS1620 Digital Thermometer
1	350-00009	Photoresistor
1	900-00001	Piezo Speaker
1	602-00009	74HC595 Output Shift Register
1	602-00010	74HC165 Input Shift Register
3	400-00002	Tact Switch (Pushbutton)
2	350-00006	LED - Red - T1 3/4
8	350-00001	LED - Green - T 3/4
1	150-02210	RED - 220 - 1/4 W - 5%
8	150-04710	RES - 470 - 1/4 W - 5%
1	150-01020	RES - 1 k - 1/4 W - 5%
3	150-01030	RES - 10 k - 1/4 W - 5%
2	150-02230	RES - 22 k - 1/4 W - 5%

skill

Comment [4]: Make sure to add these examples.

skill

Comment [5]: Verify Current Rev

skill

Comment [6]: Verify Current Rev