Welcome to E-XFL.COM

Understanding Embedded - Microcontroller, Microprocessor, FPGA Modules

Embedded - Microcontroller, Microprocessor, and FPGA Modules are fundamental components in modern electronic systems, offering a wide range of functionalities and capabilities. Microcontrollers are compact integrated circuits designed to execute specific control tasks within an embedded system. They typically include a processor, memory, and input/output peripherals on a single chip. Microprocessors, on the other hand, are more powerful processing units used in complex computing tasks, often requiring external memory and peripherals. FPGAs (Field Programmable Gate Arrays) are highly flexible devices that can be configured by the user to perform specific logic functions, making them invaluable in applications requiring customization and adaptability.

Applications of Embedded - Microcontroller,

Details	
Product Status	Obsolete
Module/Board Type	MPU Core
Core Processor	SX48AC
Co-Processor	-
Speed	-
Flash Size	32KB EEPROM
RAM Size	32KB
Connector Type	24-DIP
Size / Dimension	1.2" x 0.6" (30mm x 15mm)
Operating Temperature	0°C ~ 70°C
Purchase URL	https://www.e-xfl.com/product-detail/parallax/js1-ic

-
-
-
-
- `System.out.println()` `Terminal.getChar()`

Virtual Peripherals

Background VPs

-
-
-
-
-

Foreground VPs

-
-
-
-
-

```
  static int dsTemp(int command){
    CPU.writePin(enablePin,true);
    CPU.shiftOut(dataPin,clockPin,8,CPU.SHIFT_LSB,command);
    data = ((CPU.shiftIn(dataPin,clockPin,9,CPU.POST_CLOCK_LSB)>>7));
    CPU.writePin(enablePin,false);

    sign = data >> 8;
    if ( sign == 1 ) {
      return -data;
    }
    else
      return data;
  }                                                   // end dsTemp

  // The main routine calls the dsInit method to initialize the
DS1620,
  // then it gets the temperature value from the dsTemp method and
displays it.

  public static void main(){

    dsInit(setup);
    while (true){
      System.out.print(HOME);
      System.out.println ("Celsius temperature: ");
      System.out.println(dsTemp(READ_TEMP)/2);       // Divide by 2
for deg-C
      CPU.delay(5000);
    }                                                 // end while
  }                                                   // end main
}                                                     // end class
declaration
```

Communicating with Other Computers

Uart

Uart

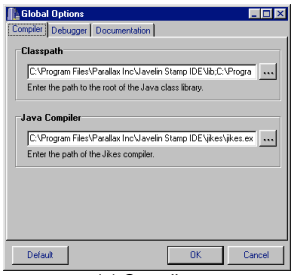
Uart

You can find more information about the ***Uart*** *class in Chapter 9.*

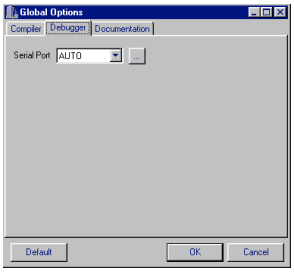
Starting the IDE

Setting Global Options

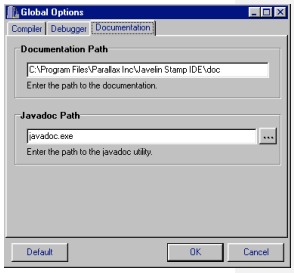
Figure 5.1 Global Options for IDE



(a) Compiler



(b) Debugger



(c) Documentation

Starting a Project

```
import stamp.core.*;

/**
 * Put a one line description of your class here.
 * <p>
 * This comment should contain a description of the class.
What it
```

```
 * is for, what it does, how it use it.
 *
 * You should rename the class and then save it in a file with
 * exactly the same name as the class.
 *
 * @version 1.0 Date
 * @author Your Name Here
 */
public class MyClass {

  // Put variables here.
  static int myVar;

  public static void main() {
    // Your code goes here.
  }

}
```

MyClass
myVar

MyFirstClass
MyFirstClass.java

for *(count)*

```
for ( int i = 0; i <; i++ ) {

}
```

if
if

Cut	Remove the selection to the clipboard	CONTROL+X
Copy	Copy the selection to the clipboard	CONTROL+C
Paste	Paste the clipboard contents to the document	CONTROL+V
Select All	Select all text	CONTROL+A
Find and ReplaceÉ	Find or find and replace text	CONTROL+F
Find Again	Repeat last find operation	F3

Toolbars and Menubars

Class Path Considerations

```
c:\myclasses\lib1         \lib1
```

try

try

Program Listing 6.8 - Exceptions Ex1

```
public class Ex1 {
  public static void main() {
    int x=0;
    int y=20;
    int z;
    try {
      z=y/x;
    }                                                   // end try
    catch (Exception e) {
      System.out.println("Divide by zero");
    }                                                   // end catch
  }                                                     // end main
}                                                       // end class
declaration
```

sdill
Comment [22]:

catch **try** **try**

Program Listing 6.9 - Exceptions Ex2

```
public class Ex2 {
  static int docomp(int a, int b) {
    return a/b;
  }                                                     // end docomp
  public static void main() {
```

sdill
Comment [23]:

```
    if (b==0) throw new ScaleError();
      return a/b;
  }

  public static void main() {
    int x=0;
    int y=20;
    int z;
    try {
      z=docalc(y,x);
    }
    catch (ScaleError e) {
      System.out.println("Scale Error");
    }
    catch (Exception e) {
      System.out.println("Unknown exception");
    }
  }
}
```

catch
Exception
Exception
ScaleError **catch**
try **catch**

main **ScaleError**

Packages and CLASSPATH

sdill
Comment [25]:

```
C:\Classes
C:\Classes
```

Cache **stamp.util**

```
stamp.util.Cache  = new stamp.util.Cache();
```

Cache.class

C:\Classes

C:\Classes\javelin stamp\util\Cache.class

javelin stamp\util

stamp.util.Cache

java.lang

import

Cache stamp.util.Cache

```
import stamp.util.Cache;
```

stamp.util

```
import stamp.util.*;
```

import

import

```
import System.out.println;
println("Hello World");
```

System java.lang

out

println import

Summary

Online Resources

http://java.sun.com
Java's home on the Web. Free downloads of the JDK, tutorials, news, and more.

http://www.norvig.com/java-iaq.html
Java Infrequently Asked Question (IAQ) list.

http://mindprod.com/gotchas.html
Java gotchas

http://www.afu.com/javafaq.htm
Java programmer's FAQ

http://www.mindspring.com/~chroma/docwiz/
Adds java doc comments to your code

http://uranus.it.swin.edu.au/~jn/java/style.htm
Automatically format your Java code

Javelin Stamp Keyword Reference

abstract

abstract **abstract**

abstract

abstract

```
abstract class AbaseClass {
  abstract void someMethod();
}
```

```
y = 5/256*x;


y = (5*x)/256;  // parenthesis not necessary, but added for
clarity



y = (50*x)/256;

                              /          %


System.out.println("Volts = " + y/10 + "." + y%10);
```

```
y = 10 + 33 / 17 % 3 * 100;  // answer is 110
```

<<, >>, >>>

```
                                      <<       >>   >>>

                                                                100>>4
                100/16



                                                              10*x  as
(x<<3)+(x<<1)

   <<                                                        >>


            >>>
```

NullPointerException
null

_____ RuntimeException

_____
static RuntimeException throwIt()

Object

Object
Object

```
_______

StringBuffer sb = new StringBuffer("ATDT");
sb.append(getTelephoneNumber());       //getTelephoneNumber
returns String
dial(sb.toString());                   //dial expects a String
argument
```

System

```
   System         static


________ Object

_____
static PrintStream out    PrintStream



_______

System.out.println("Parallax");
```

Throwable

```
Throwable

________ Object

___________________Error Exception

_______
String getMessage ()
```

The *java.io* Package

```
   java.io               PrintStream
```

DAC dac dac
dac installVP

```
DAC dac = new DAC(CPU.pin2);
dac.update(125);
CPU.removeVP(dac);

// Later in the program after pwm was installed and removed...
CPU.installVP(dac);
dac.update(125);
```

message

```
static void message(char [] data, int length)
```

char
String.toCharArray String
System.out.println

```
char [] characters = {'a', 'b'};
CPU.message(characters,2);
String s = "CDE";
CPU.message(s.toCharArray(),3);
```

nap

```
static void nap(int period)
```

The Javelin Stamp can only call nap as a foreground process with no background processes running. Before calling this method, you must use ***VirtualPeripheral.removeItem*** *to uninstall any VPs that are installed. See example below.*

period
nap
nap
period

setInput, 174
shiftIn, 174
short, 89, 125
static, 33, 125
StringBuffer, 42, 100, 104, 154, 192
Strings, 104–5, 152, 192
StringBuffer. *See StringBuffer*

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
Calculations, 34
char, 89
Declaration, 89-91
final, 32, 90-91
int, 89
short, 89
static, 80

Virtual Peripherals
Timer, 187
UART(s), 64, 188void, 99, 130
volatile, 136

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