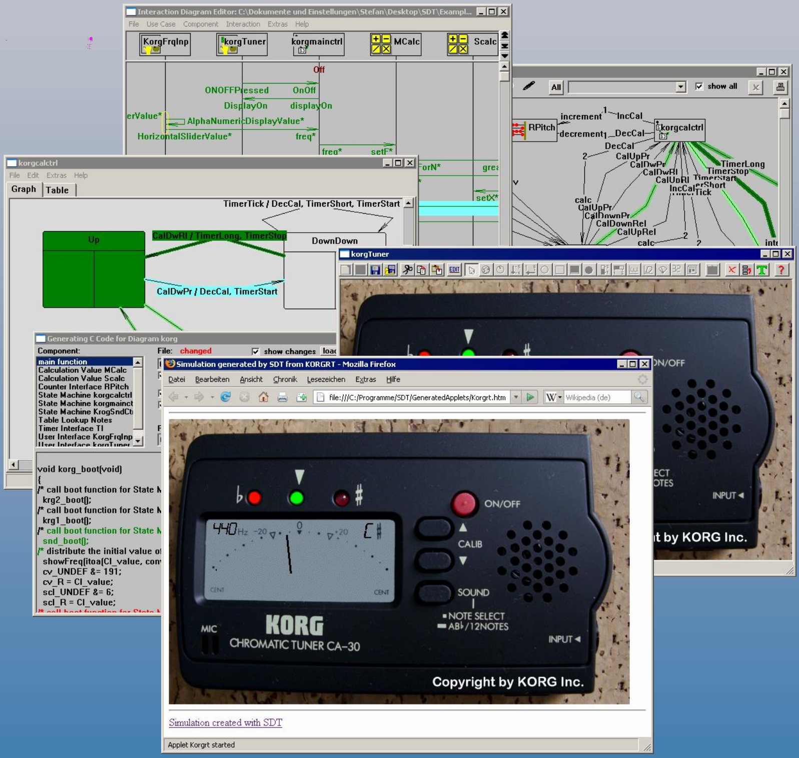
System Design Toolkit

Java Applet Generator



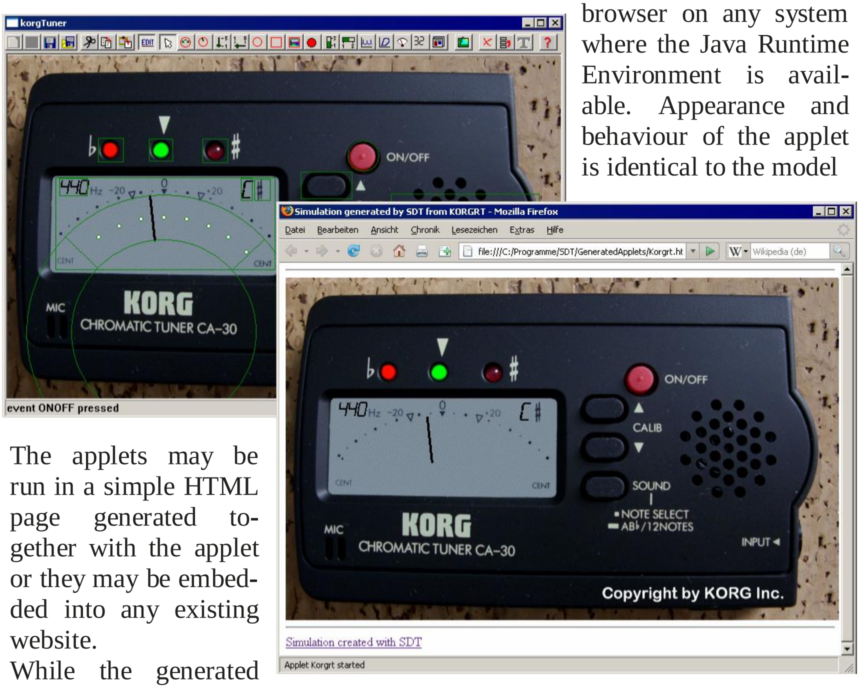
```
void korg_boot(void)
{
/* call boot function for State M
  krg2_boot();
/* call boot function for State M
  krg1_boot();
/* call boot function for State M
  snd_boot();
/* distribute the initial value of
  showFreq(itoa(CI_value, con
  cv_UNDEF &= 191;
  cv_R = CI_value;
  scl_UNDEF &= 6;
  scl_R = CI_value;
```

Addentum to User's Manual

Java Applet Generator

Java Applet Generator

The Java Applet generator is a code generation utility, which uses models generated with the System Design Toolkit to generate portable simulations. These portable simulations are generated as Java applets, which run under any browser on any system where the Java Runtime Environment is available. Appearance and behaviour of the applet is identical to the model


korgTuner
EDIT
440 Hz -20 0 +20 C♯
CENT
MIC
KORG
CHROMATIC TUNER CA-30
ON/OFF
Simulation generated by SDT from KORGRT - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
file:///C:/Programme/SDT/GeneratedApplets/Korgrt.ht
Wikipedia (de)
CALIB
SOUND
NOTE SELECT
ABb/12NOTES
INPUT
Copyright by KORG Inc.
Simulation created with SDT

event ONOFF pressed

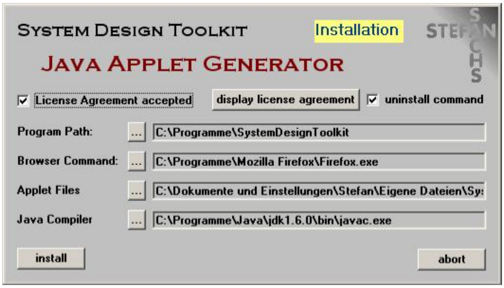
Applet Korgrt started

The applets may be run in a simple HTML page generated together with the applet or they may be embedded into any existing website.

While the generated code for a target system does not include the user interface nor the simulation of input data, Java applets contain all components of the model.

Installation

The Java Applet Generator add-on comes with its own installation program, which starts automatically, when the installation CD is inserted. The license agreement has to be acceped to enable the installation. A deinstallation icon is added to the start menu., when its checkbox is selected. The installer searchs for the standard program directory; when a dif-



☑ License Agreement accepted

☑ uninstall command

Program Path: C:\Programme\SystemDesignToolkit

Browser Command: C:\Programme\Mozilla Firefox\Firefox.exe

Applet Files C:\Dokumente und Einstellungen\Stefan\Eigene Dateien\Sy:

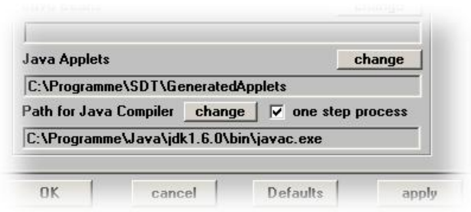
Java Compiler C:\Programme\Java\jdk1.6.0\bin\javac.exe ferent directory is used, it has to be selected manually. The same holds for the browser path, the default directory for the generated applet files and the program path for the Java compiler. When the install button is pressed, the program files are copied and the default paths are added to the program defaults. With the next start of the toolkit, the language selection of the extras menu will offer the choice of Java Applet.

Manual Installation and Deinstallation

The Java Applet Generator may be installed manually by copying the files JavaAppl.sll and JavaCode.sll to the SDK program directory and by adding the names of these two files on separate lines to the file SDT.BND. To deinstall the Java Applet Generator, both files mentioned above have to be removed from the program directory and their naes have to be removed from the file SDT.BND

Default Settings

The defaults set , when installing may be revised in the options dialogue under the ‘JAVA’ tab. Here the ‘one step process’ option may be changed.



Java Applets

C:\Programme\SDT\GeneratedApplets

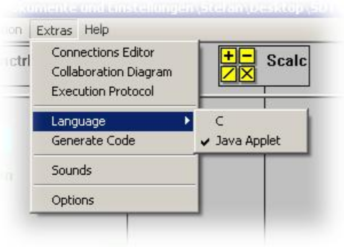
Path for Java Compiler

☑ one step process

C:\Programme\Java\jdk1.6.0\bin\javac.exe

Code Generation

When ‘one step process’ is selected, the code generation window is not opened, when the code generation command is selected from the extras menu. Instead, the Java applet source code, the HTML page and the auxiliary files are written directly to the default directory for applet files, the bitmaps and audio files are converted and the compiler is started. Finally, the HTML page is opened in the standard browser.



When the Code Generator Window is used (‘one step process’ unchecked), the code can be reviewed under ‘main function’. There are no options available for Java applets.

Java Applet File

The complete code is written to one file as a subclass of Applet. The generated applet is then translated to a .class file by invoking the command-line version of the Java compiler, javac, as delivered as a part of the JDK.

HTML Page

A rudimentary HTML page is generated together with the applet. It shows the applet and a note about the generator with a link to the toolkit homepage

Auxiliary Files

In addition to the Java applet itself (.class file), the simulation requires bitmaps showing the simulated device and its modified parts on screen, bitmaps containing the font glyphs of the alphanumeric readouts and files containing sound support. Bitmaps and Sounds are written as .bmp and .wav files to the target directory and there converted into the Java supported formats .png and .au using the freeware utilities bmp2png and convert. The bitmap and wave files are then removed from the applet directory.
When publishing an applet on the web, all files, except the Java source file (and the HTML file, when the applet is embedded in an existing page), have to be copied to the embedding documents base directory.

Fonts

Fonts are transformed from vector fonts to pixel based fonts. This is already done in the UI Simulator, to make the runtime versions independent froom the base of installed fonts of a specific machine.
These pixel based fonts deliver the glyphs bitmap for the Java version of the fonts. Therefore the generated java applets do not depend at all on the installed fonts base of the client running the applet.

Handling of Timers

All timers of the model apart from the RT data source are served by a Java timer.
The interval of this timer is set to the greatest common divisor of all timer intervals used in the model. This master timer then serves each of the model timers including:

- User defined timers (The interval is set in the configuration dialogue of each component)
- The actual set of flashing UI elements (The intervals for On and OFF state are defined in the options dialogue under the execution tab.

- The actual set of Realtime Curves (The interval is set in the configuration dialogue of any curve)
- All hotspots reacting to a long-pressed or long-released event. (The threshold intervals is defined in the options dialogue under the execution tab.

When the interval settings of one or more of the relevant components requires a master interval of less than the minimum of 50 ms, a prompter explains, that the minimum interval has been set and indicates the component requiring a lower setting.

Input data from a Realtime Data Source is generated in the main loop of the run function; it is independent from the timer processing function and has its own timing, which is defined in the configuration dialogue of the source component.

Stefan Sachs
Dr. Ing.
Beratender Ingenieur

Ringreiterweg 20
23558 Lübeck
Germany

Tel.	+49 (451) 8993444
Fax	+49 (451) 8993445
E-Mail:	ssachs@acm.org
web:	www.ssachs.de