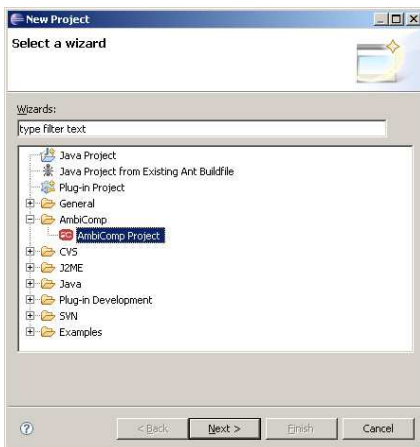


AmbiComp Eclipse Plug-In



AmbiComp is a research project running from 2006 to 2009 that is funded by the German Ministry of Education and Research (Bundesministerium für Bildung und Forschung). The topic of the research is to simplify the development of software for embedded systems in the area of "Ambient Intelligence". Especially small and medium-sized enterprises shall be enabled to quickly develop and offer appropriate devices, including the necessary software, and to ensure interoperability with products of other vendors.

Within the AmbiComp project, a software development platform based on the freely available tool "Eclipse" will be implemented. Using this software, it will be easy to develop, test, and distribute even ambitious Ambient Intelligence applications.



Features

- Easy installation, update, and de-installation using the Eclipse Update Framework.
- Project generation with an AmbiComp-Wizard
- Comfortable Java programming with the support of Eclipse
 - Code completion
 - Code refactoring
 - Debugging
- Mapping of the complete tool chain in Eclipse
 - Programming in Java using well-designed APIs
 - Transcoding in AmbiComp-BLOB files
 - Transfer of applications and firmware updates to the target device
- Specific AmbiComp views with drag&drop support for
 - information about the connected/emulated devices
 - configuration of emulators and "Otto" (see figure below)
 - Presentation of the complete data exchange
 - Transfer of applications
- Automatic logging into a file and output to an AmbiComp Console
- AmbiComp preferences and project properties for user-specific presets
- Eclipse help pages / cheat sheets with tutorials and example code
- Operating systems: Linux and Windows, open source

Development of AmbiComp Applications with the AmbiComp Eclipse Plugin

The Eclipse plugin allows the user to easily develop, compile, transcode, and deploy applications to a real or emulated AICU (Ambient Intelligence Control Unit).

Furthermore, an Emulator is provided, which allows emulating an AICU on a Linux server. Via a USB-to-Backplane Bus converter, real AICUs can also be connected and controlled.

Hardware and Software Prerequisites

If the developer wants to develop not only with the support of the emulator, but also using real hardware, at least one AICU is needed. This AICU is composed of one or more SMs (sandwich modules, see also the companion data sheets), which communicate with the developer's computer via the software component "Otto" using a socket connection. In addition, a converter is needed to translate between the USB interface and the Backplane Bus used as an AICU-internal communication bus.

In terms of software, the developer only needs the freely available tool "Eclipse" (version 3.4 is recommended), as well as the Java SDK (version JDK 5 or 6), which is also freely available.

Within Eclipse, the AmbiComp plug-in has to be installed (which is described in detail at <http://www.ambicomp.org>).

Programming in Java

On each SM, an ACVM (AmbiComp Virtual Machine) is running, which is an adapted Java Virtual Machine that interprets and runs the transcoded code.

Thus, the application can be written in the comfortable programming language Java. Two variants of the AmbiComp API are provided:

- bundled with a small library (mini.jar) for modules with small memory, and
- bundled with a comprehensive, comfortable library (cldc.jar) that conforms mainly to the CLDC API.

Depending on the module connecting the functionalities of analogous and digital ports, Ethernet or Bluetooth can be used.

In order to develop an AmbiComp application, the developer creates a new AmbiComp project with the help of the AmbiComp Project Wizard, which automatically links all necessary libraries.

Afterwards, when programming in Java, the developer is supported by Eclipse's well-known functions such as code completion, easy creation of packages and classes, refactoring, etc.

Before the code can be run by the ACVM on an SM, the compiled byte code has to be transcoded into the small and efficient AmbiComp-specific format. This is performed via a context-sensitive menu in the Java project. The result is the creation of so-called BLOB (Binary Large Object) Files.

These BLOB files now have to be transferred to the appropriate device. This process is also supported by the AmbiComp plug-in via a graphical user interface.

Further Features

AICU List View

In an AmbiComp-specific view, all information about an AICU is displayed that is relevant for the developer, such as SMs contained, hardware and firmware versions, available memory, etc.

The AICU List View also serves as a source for the direct access to an AICU or even an SM. This access is realized via drag&drop functionality.

AICU Communication View

If an AICU/SM is dragged from the AICU List View and dropped into the AICU Communication View, all data is displayed that is exchanged between the computer used for the development and the AICU/SM.

Additionally, single commands can be executed or complete BLOB files can be transferred.

Furthermore, debug messages of the programmed applications are displayed, which allows the developer to easily trace the program.

Emulator View

The Emulator View offers a lot of possibilities to interact with the emulator and with emulated AICUs. Emulated devices offer the same possibilities as real ones. This allows configuring AICUs, transferring BLOB files, and starting emulated devices.

Otto Configuration View

In this view, the connection parameters of the software component "Otto" running on a Linux server or Windows computer can be adjusted. The Eclipse plugin can communicate with several "Ottos" on different servers and with the emulated and real AICUs connected to them.

Error Handling / AmbiComp Console

In the AmbiComp Console, all important messages related to the AmbiComp plug-in are shown. At the same time, the same data are written into a log file that the developer can send to the AmbiComp staff if needed (see below).

Firmware Update

AmbiComp is a research project that continues to evolve. This means that new versions of the ACVM will be published regularly, together with improvements of the API and the correction of errors, if applicable.

In order to enable the developer to take advantage of such new developments, the Eclipse plugin allows updating the firm-

ware of the modules via a simple drag&drop dialog.

Preferences/Properties

In the "Preferences" section, important settings can be made, such as the granularity of the debug messages of the plug-in.

In case of a problem with the AmbiComp plug-in, it is possible for the developer to send a message to the AmbiComp project team and attach the log file created by the plug-in. (The prerequisite for starting an email client from Java is JDK version 1.6 or higher.)

In the project-specific "Properties", the developer can easily switch between the two AmbiComp API variants miniAPI and CLDC API.

Eclipse Help / Cheat Sheets

Via the Eclipse Help function, the developer is informed about all aspects of the AmbiComp project.

Tutorials and examples are provided, which give a comprehensive and simple introduction to programming applications for AmbiComp with the help of the Eclipse plugin.

In addition, so-called "Cheat Sheets" help the developer in taking his first steps in programming for AmbiComp.

Further information about the AmbiComp plug-in can be found at:
<http://www.ambicomp.org>

