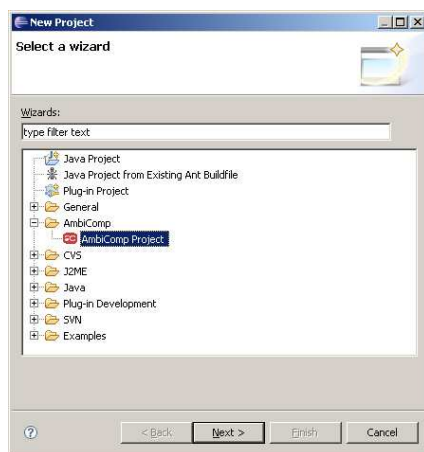


AmbiComp Eclipse Plugin



AmbiComp ist ein Forschungsprojekt, das in den Jahren 2006-2009 vom Bundesministerium für Bildung und Forschung gefördert wird. Ziel des Projekts ist, die Entwicklung von Software für eingebettete Systeme im Bereich der „Ambient Intelligence“ zu erleichtern. So soll gerade auch kleinen und mittelständischen Unternehmen ermöglicht werden, solche Geräte inklusive der dazu benötigten Software schnell zu entwickeln und zu vermarkten und die Interoperabilität mit Produkten anderer Hersteller zu gewährleisten.

AmbiComp entwickelt dafür unter anderem eine Software-Entwicklungs-Plattform auf Basis des freien Java-Werkzeugs Eclipse. Mit ihr ist es möglich, sehr einfach auch anspruchsvolle Ambient-Intelligence-Anwendungen zu entwickeln, zu testen und zu verteilen.



Features

- Installation, Update und Deinstallation sind über das Eclipse Framework einfach durchzuführen
- Projekterzeugung mit einem AmbiComp-Wizard
- Komfortables Java-Programmieren mit Eclipse-Unterstützung
 - Code-Vervollständigung
 - Code-Refactoring
 - Debugging
- Abbildung der kompletten Werkzeugkette in Eclipse
 - Programmierung in Java mit AmbiComp API
 - Transcodierung in AmbiComp-BLOB-Dateien
 - Übertragen von Anwendungen und Firmware Updates auf das Endgerät
- Spezielle AmbiComp-Views mit Drag&Drop-Unterstützung für
 - Informationen über die angeschlossenen/emulierten Geräte
 - Konfiguration von Emulatoren und „Otto“ (siehe Abb. unten)
 - Darstellung des gesamten Datenaustauschs
 - Übertragung von Anwendungen
- Automatisches Logging in eine Datei und Ausgabe in einer AmbiComp Console
- AmbiComp Preferences und Project Properties für benutzer-spezifische Voreinstellungen
- Eclipse-Hilfeseiten/Cheat Sheets mit Tutorials und Beispielcode
- Betriebssysteme: Linux und Windows, open source

Entwicklung von AmbiComp Anwendungen mit dem AmbiComp Eclipse Plugin

Das Eclipse Plugin ermöglicht es dem Anwender, auf einfache Art AmbiComp Anwendungen zu programmieren, zu compilieren, zu transcodieren und auf reale oder emulierte AICUs (AICU = Ambient Intelligence Control Unit) zu übertragen.

Des Weiteren wird eine Emulator-Software zur Verfügung gestellt, mit deren Hilfe auf einem Linux-Server AICUs emuliert werden können. Über einen USB/Backplanebus-Konverter können reale AICUs angeschlossen und gesteuert werden.

Hardware- und Software-Voraussetzungen

Möchte der Entwickler nicht nur den Emulator nutzen, sondern auch mit realer Hardware entwickeln, so benötigt er mindestens eine AICU. Diese setzt sich aus einem oder mehreren SMS (Sandwich-Modulen) zusammen (siehe auch: ergänzende Datenblätter) und kommuniziert über die „Otto“-Software-Komponente mit dem Rechner des Entwicklers über eine Socketverbindung. Des Weiteren wird ein Konverter benötigt, der zwischen USB und dem AICU-internen Kommunikationsbus (Backplanbus) konvertieren kann.

An Software benötigt der Entwickler lediglich das freie Werkzeug Eclipse (Version 3.4 empfohlen), sowie das ebenfalls frei verfügbare Java SDK (Version JDK 5 oder 6). In Eclipse muss nur noch das AmbiComp Plugin installiert werden.

(nähere Angaben siehe: <http://www.ambicomp.org> unter „Entwicklungswerkzeuge“).

Programmierung in Java

Auf jedem SM läuft eine ACVM (AmbiComp Virtual Machine) – eine angepasste Java Virtual Machine, die den erstellten Code interpretiert und ausführt. Die Anwendung wird in der komfortablen Programmiersprache Java geschrieben. Die zur Verfügung stehende AmbiComp API wird in zwei Varianten mit dem Eclipse Plugin ausgeliefert:

- Gebündelt mit einer kleineren Bibliothek (mini.jar) für Module mit wenig Speicherplatz und
- gebündelt mit einer umfangreicheren, komfortableren Bibliothek (cldc.jar), die sich an der CLDC API orientiert.

Je nach angeschlossenen Modul können dann Funktionalitäten von analogen/digitalen Pins, Ethernet oder Bluetooth genutzt werden.

Um eine AmbiComp Anwendung zu schreiben, erstellt der Entwickler zunächst ein neues AmbiComp-Projekt mit Hilfe des AmbiComp Projekt Wizards, in das bereits alle notwendigen Bibliotheken automatisch eingebunden werden. Nach Projekterstellung wird der Programmierer dann durch die für Java-Programmierung in Eclipse bekannten Funktionalitäten wie z.B. Code-Vervollständigung, einfaches Erstellen von Packages und Klassen, Refactoring, etc. unterstützt.

Bevor der Code von der ACVM auf einem SM ausgeführt werden kann, muss der compilierte Bytecode zunächst in ein sehr klein gehaltenes und effizientes AmbiComp-spezifisches Format transcodiert werden. Dies erfolgt über ein kontextsensitives Menü im Java-Projekt. Hierbei werden die sogenannten BLOB (Binary Large Object)-Dateien erzeugt.

Die BLOB-Dateien müssen nun auf das entsprechende Endgerät übertragen werden. Auch dieser Vorgang wird vom AmbiComp Plugin über grafische Oberflächen unterstützt.

Weitere Features

AICU List View

In einer AmbiComp-spezifischen Anzeige werden dem Entwickler alle wichtigen Informationen über eine AICU angezeigt, so z.B. die enthaltenen Module, Hardware- und Firmware-Versionen oder der vorhandene Speicherplatz. Die AICU List View dient auch als Quelle für den Zugriff auf die direkte Kommunikation mit einer AICU oder einem SM. Der Zugriff ist komfortabel über Drag&Drop realisiert.

AICU Communication View

Wird eine AICU/SM von der AICU List View in den Bereich der AICU Communication View gezogen, so werden in dieser Ansicht alle Daten angezeigt, die zwischen Entwicklungsrechner und AICU/SM ausgetauscht werden. Zusätzlich können hier einzelne Kommandos aufgerufen oder auch ganze BLOB-Dateien direkt übertragen werden. Debug-Meldungen der programmierten Anwendungen werden hier ebenfalls ausgegeben, so dass der Entwickler den Programmablauf auf einfache Weise kontrollieren kann.

Emulator View

Die Emulator View bietet umfangreiche Möglichkeiten, um mit Emulator und emulierten AICUs arbeiten zu können. Emulierte Geräte bieten genau die gleiche Funktionalität wie reale. So können AICUs konfiguriert, BLOB-Dateien übertragen und emulierte Geräte dann gestartet werden.

Otto Configuration View

In dieser Ansicht werden die Verbindungseinstellungen zu den Otto Software Komponenten auf Linux-Server bzw. Windows-Rechner getätigt. So kann das Eclipse Plugin mit mehreren „Otto“s auf unterschiedlichen Servern und den daran angeschlossenen emulierten und realen AICUs kommunizieren.

Fehlersuche/AmbiComp Console

In der AmbiComp Konsole werden alle wichtigen Nachrichten ausgegeben, die das AmbiComp Plugin betreffen. Die gleichen Informationen werden zusätzlich in eine Datei geschrieben, die der Entwickler bei Bedarf an die AmbiComp-Projektmitarbeiter senden kann (s.u.).

Firmware-Update

AmbiComp ist ein Forschungsprojekt, das ständig weiterentwickelt wird. Somit werden in regelmäßigen Abständen neue Ver-

sionen der ACVM veröffentlicht, die API optimiert und eventuelle Fehler beseitigt. Damit der Entwickler zeitnah von der Weiterentwicklung profitieren kann, ist es über das Eclipse Plugin möglich, die Firmware der Module auf einfache Art über einen Update Dialog zu aktualisieren.

Preferences/Properties

In den Voreinstellungen können wichtige Eingaben gemacht werden, so z.B. die Granularität der Debug-Ausgaben des Plugins. Hier kann der Entwickler auch durch einen einfachen Mausklick eine Nachricht an die AmbiComp-Projektentwickler senden, sowie die vom Plugin erstellte Log-Datei anhängen, sollten einmal Probleme mit dem AmbiComp Plugin auftreten (Voraussetzung für den Start des eMail-Client aus Java: mind. JDK 1.6).

In den projektspezifischen Properties kann der Entwickler schnell zwischen den beiden AmbiComp API Varianten miniAPI und CLDC API umschalten.

Eclipse Hilfe/Cheat Sheets

Über die Eclipse-Hilfeseiten erfährt der Entwickler alles über das AmbiComp-Projekt. Hier werden zielgruppenspezifische Beispielprogramme und Tutorials angeboten, die er dann nachprogrammieren kann, um ihm einen leichten Einstieg in die Programmierung für AmbiComp mit dem Eclipse-Plugin zu ermöglichen.

Über die sog. Cheat Sheets werden ihm weiterhin die ersten Schritte in die Programmierung für AmbiComp erleichtert.

Weitere Informationen zum AmbiComp Plugin:
<http://www.ambicomp.org>

