

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

FAKULTÄT FÜR MASCHINENBAU UND MECHATRONIK

C-Programmierung mit Eclipse

Prof. Dr.-Ing. Reiner Kriesten
Michael Lautenschläger B.Eng.
18. September 2012

Inhaltsverzeichnis

1	Einleitung	1
1.1	Integrierte Entwicklungsumgebung	1
1.2	Eclipse	2
2	Installation von Eclipse	3
2.1	Installation von Wascana	3
2.2	Installation in einzelnen Schritten	4
3	Erstellen eines ersten Programms	7
4	Hinweise zu Eclipse	11

1 Einleitung

Im heutigen Alltag begegnen wir an vielen Stellen Anwendungen, die ohne eine Programmierung nicht möglich wären. Ein Wecker mit einem digitalen Display hilft uns beim Aufstehen, die Vollautomaten-Kaffeemaschine brüht uns den Kaffee. Nicht nur in Computern, bei denen es uns am meisten bewusst ist, sondern auch in vielen anderen Geräten steckt eine elektronische Logik. Es handelt sich hierbei um die eingebetteten Systeme, die in vielen Haushaltsgeräten, aber auch in allen Fahrzeugen oder Flugzeugen zum Einsatz kommen. Je nach Anwendung sind diese eingebetteten Systeme ihrem Aufgabengebiet angepasst.

Wie auch in den herkömmlichen Computern werden diese eingebetteten Systeme neben der Hardware-Logik auch programmiert. Hier ist es entscheidend möglichst platzsparend und Effizient vorzugehen, da sich bei diesen Massenprodukten die kleinsten Erweiterungen von Bauteilen (wie Speichern) stark auf den Preis auswirken. **C** ist eine solch effiziente Programmiersprache, die außerdem sehr stabil arbeitet. Es existieren auch weitere Programmiersprachen für eingebettete Systeme, auf diese soll in diesem Dokument jedoch nicht weiter eingegangen werden.

Dieses Dokument selbst soll lediglich als Hilfestellung dienen eine IDE (siehe Kapitel 1.1) zu installieren und sie zu benutzen. Kenntnisse der Programmiersprache C werden hier nicht vermittelt.

1.1 Integrierte Entwicklungsumgebung

Als Basis zum Programmieren wird eine integrierte Entwicklungsumgebung (IDE) verwendet. Sie soll vor allem dem Programmierer die Arbeit erleichtern. Dies wird dadurch erreicht, dass ein Texteditor, Compiler, Linker, Debugger und verschiedene Quelltext-formatierungsfunktionen in einer Plattform zusammengeführt werden. Was sie im Einzelnen bedeuten, soll an dieser Stelle jedoch nicht erläutert werden.

IDEs gibt es für verschiedene Programmiersprachen. So existieren beispielsweise für die Programmiersprache C die folgenden IDEs:

- Code::Blocks
- C-Free
- NetBeans

- Microsoft Visual Studio
- Eclipse

Dabei stellen die aufgezählten IDEs nur einen kleinen Teil der Gesamtmenge dar, auf welche hier aber nicht näher eingegangen wird.

1.2 Eclipse

In diesem Dokument wird ausschließlich die IDE *Eclipse* behandelt. Eclipse ist ein frei verfügbares Programmierwerkzeug zur Entwicklung von Software. Neben dessen Hauptanwendung in Java (Programmiersprache) eignet es sich auch zur Entwicklung von eingebetteter Software in C.

Im Folgenden wird eine Installationsanleitung für das Betriebssystem Windows und die ersten Schritte in Eclipse vorgestellt.

2 Installation von Eclipse

Für eine C-Programmierung in Eclipse müssen verschiedene Tools installiert werden. Diese können entweder einzeln von <http://www.eclipse.org/>, oder über das Gesamtpaket von <http://wascana.sourceforge.net/> heruntergeladen werden.

Das Paket von Wascana beinhaltet bereits:

- Eclipse IDE: Frei verfügbare Entwicklungsumgebung
- C/C++ Development Tool: Da Eclipse mehrere Programmiersprachen unterstützt und dessen Hauptanwendung Java ist, müssen Erweiterungs-Tools für andere Programmiersprachen eingebunden werden. Dieses Tool (C/C++ Development Tool) ist für die Programmierung in C bestimmt.
- MinGW: Hierbei handelt es sich um einen Compiler. Er ist eine Portierung des GNU-Compilers (aus Linux) in das Windows Betriebssystem.
- Java Runtime Environment: Die Java Runtime wird benötigt um Eclipse selbst auszuführen. Es handelt sich hierbei um eine Art virtuelle Maschine, die es ermöglicht ein Java-Programm (in diesem Fall Eclipse) auszuführen.

2.1 Installation von Wascana

In diesem Kapitel wird die Installation mit dem Paket von Wascana beschrieben und im nächsten Kapitel die Installation in einzelnen Schritten. Hierfür wird die Datei „wascana-mingw-0.9.3.exe“ (Stand: 19.04.2012) von der Wascana-Homepage heruntergeladen.

Die Installation wird durch einen Doppelklick auf die heruntergeladene Datei ausgeführt. In dieser Beschreibung werden die empfohlenen Konfigurationen des Setup-Assistenten beibehalten.

Nach der Installation wird Eclipse gestartet. Beim ersten Starten muss der Arbeitsordner für Eclipse definiert werden. Durch das Setzen des Hakens bei „Use this as the default and do not ask again“ wird diese Frage bei wiederholtem Ausführen von Eclipse unterdrückt. Diese Auswahl ist in der Abbildung 2.1 dargestellt.

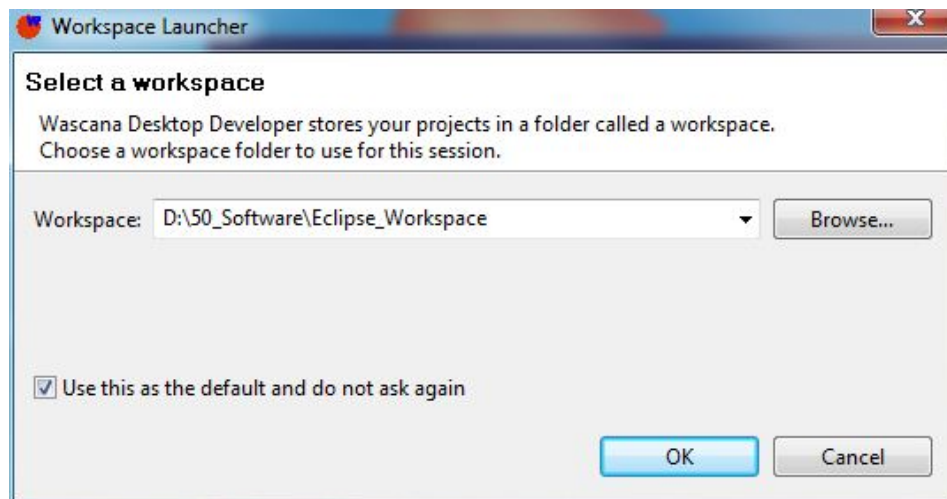


Abbildung 2.1: Arbeitsordner-Auswahl in Eclipse

2.2 Installation in einzelnen Schritten

In Kapitel 2.1 wurde beschrieben, wie Eclipse mit dem „Wascana-Paket“ durchgeführt wird. Wie eine Installation in einzelnen Schritten abläuft wird in diesem Kapitel gezeigt.

Zunächst werden die folgenden Pakete benötigt (diese wurden bereits zu Beginn des Kapitels 2 beschrieben):

- Eclipse Classic
- CDT Master
- MinGW

Die ersten beiden Pakete können von der Eclipse Homepage heruntergeladen werden (<http://www.eclipse.org/>) und das Paket MinGW von <http://www.mingw.org/>. Diese Pakete Eclipse Classic und CDT Master werden im ersten Schritt entpackt. Es wird empfohlen den Ordner von Eclipse Classic in das Programmverzeichnis von Windows („C:\Programme\...“) zu kopieren.

HINWEIS: Anders als von Windowsinstallationen gewohnt, muss für das Programm Eclipse Classic an dieser Stelle keine Installationsdatei gestartet werden. Die Installation wird hier durch das Entpacken und Kopieren durchgeführt.

Von der Datei „eclipse.exe“ kann eine Verknüpfung erstellt werden, welche auf dem Desktop platziert wird (dieser Schritt ist nicht notwendig, erleichtert jedoch das ausführen von Eclipse).

Nun wird MinGW installiert. Dies erfolgt durch ausführen der heruntergeladenen Datei, wie in diesem Fall „mingw-get-inst-20120421.exe“. Lediglich an zwei Stellen weicht die Installation von den voreingestellten Werten ab.

Die erste Abweichung ist bei der Wahl der Kataloge. Hier wird eine Aktualisierung über das Internet durchgeführt. Die dafür erforderliche Einstellung ist in Abbildung 2.2 dargestellt. Nach dem Akzeptieren der Lizenz-Vereinbarung werden die einzelnen

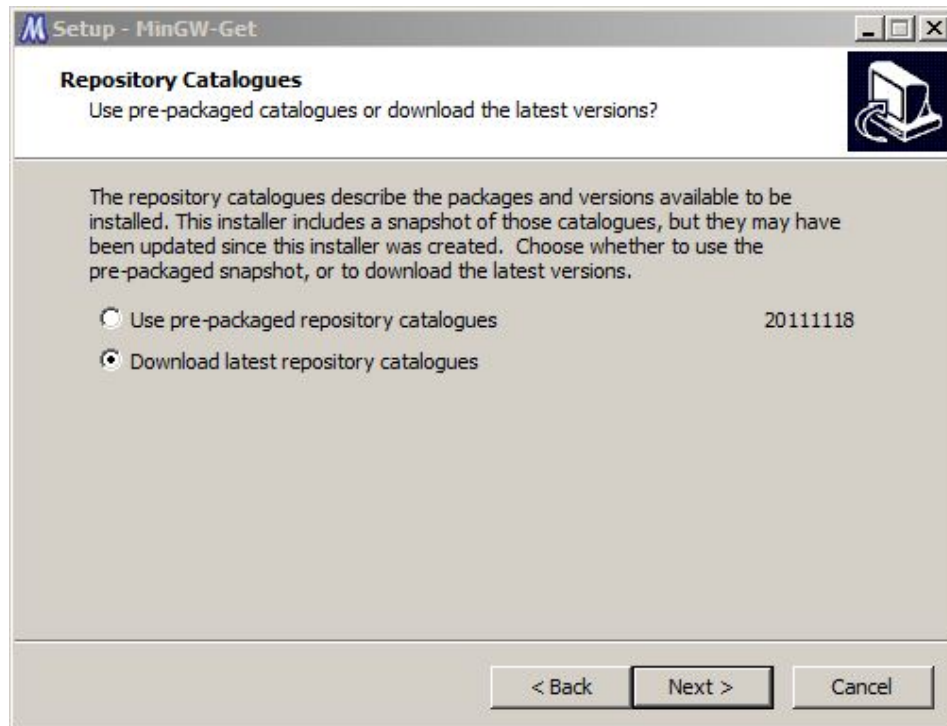


Abbildung 2.2: Installation für MinGW-Kataloge

Komponenten ausgewählt. Hierfür werden nur die in Abbildung 2.3 ausgewählten Komponenten benötigt. Nun wird Eclipse durch einen Doppelklick auf die zuvor erstellte Verknüpfung ausgeführt. Bevor aber in C programmiert werden kann, muss das Tool CDT in Eclipse eingebunden werden.

Dazu wird über → Help → Install New Software... der Software-Manager von Eclipse geöffnet. Nun wird über die Schaltfläche „Add...“ der entpackte Ordner von CDT ausgewählt. Hier müssen folgende Erweiterungen ausgewählt werden:

- CDT Main Features
 - C/C++ Development Tools
 - C/C++ Development Tools SDK
- CDT Optional Features
 - C/C++ Debugger Services Framework (DSF) Examples

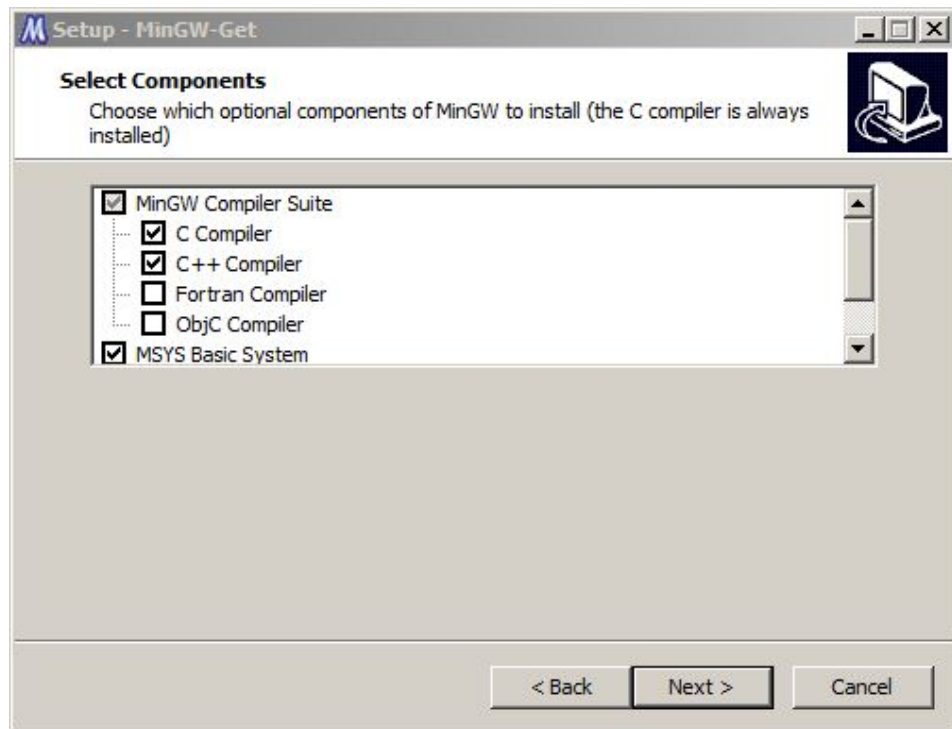


Abbildung 2.3: Installationseinstellungen für MinGW-Komponenten

- C/C++ GCC Cross Compiler Support
- C/C++ GDB Hardware Debugging
- Eclipse Debugger for C/C++
- Miscellaneous C/C++ Utilities
- TCF
 - Target Communication Framework (Incubation)
 - Target Communication Framework (Incubation), CDT Extender
- Uncategorized
 - P2 C/C++ Toolchain Installer

Über → Next und Akzeptieren der Lizenz-Vereinbarung werden die Erweiterungen eingebunden. Damit wird die Installation abgeschlossen und Eclipse kann verwendet werden.

3 Erstellen eines ersten Programms

Für die erste Anwendung in Eclipse wird das aus der Informatik bekannte Programm „Hello World“ gewählt. Dazu wird zunächst ein Projekt erstellt. Die Erstellung des Projektes kann auf die folgenden drei Arten durchgeführt werden.

- a. Rechtsklick in „Project Explorer“ → New → C Project
- b. Navigationsleiste „File“ → New → C Project
- c. Über eine der Schaltflächen in der Navigationsleiste → New → C Project

In jeder der drei Vorgehensweisen wird ein C Project erstellt, welches das in Abbildung 3.1 dargestellte Fenster aufruft. Unter „Project name:“ muss ein Projektname angegeben werden, welcher hier „Hallo“ lautet. Der zuvor ausgewählte Arbeitsordner ist hier vorausgewählt, kann aber auch verändert werden. Für die „Project types“ wird „Executable“ ausgewählt und in „Toolchain“ „MinGW GCC“. Bei MinGW GCC („Minimalist GNU for Windows“) handelt es sich um den GNU-Compiler, der von Linux in das Windows Betriebssystem portiert wurde. Nun können entweder über „Next“ weitere Einstellungen getroffen werden, oder wie in diesem Beispiel durch das Klicken von „Finish“, das Projekt erstellt werden. Im nächsten Schritt wird eine Quelldatei eingefügt. Hierfür stehen wiederum die drei genannten Vorgehensweisen zur Verfügung. An dieser Stelle wird nur eine der drei Vorgehensweisen beschrieben. Durch einen Rechtsklick auf den Projekt-Ordner „Hallo“ und anschließendem Klicken auf New → Source File wird das in Abbildung 3.2 dargestellte Fenster geöffnet. Es muss lediglich der Name der Quelldatei angegeben werden, welcher hier „World.c“ lautet.

HINWEIS: Beim Erstellen der Quelldatei muss auf die Dateiergung geachtet werden. Der Name muss mit einem „.c“ abschließen, damit eine Quelldatei für ein C-Programm erstellt wird.

In diesem Fenster werden keine weiteren Einstellungen getroffen. Es kann also durch Betätigen von „Finish“ geschlossen werden.

Nun wird das Programm für „Hello World“ geschrieben. Projekt und Programm werden in der Abbildung 3.3 dargestellt.

Vor dem Kompilieren des Programms, muss die Datei gespeichert werden (→ File → Save). Das Kompilieren erfolgt dann durch das Betätigen des „Hammer-Symbol“. Zum anschließenden Ausführen des Programms wird die Schaltfläche mit dem grünen „Wiedergabepfeil“ (Run Hallo) betätigt. Darauf erscheint in der „Console“ (im unteren Bereich von Eclipse) die Ausgabe des Programms. Diese Ausgabe ist auch in Abbildung 3.4 dargestellt.

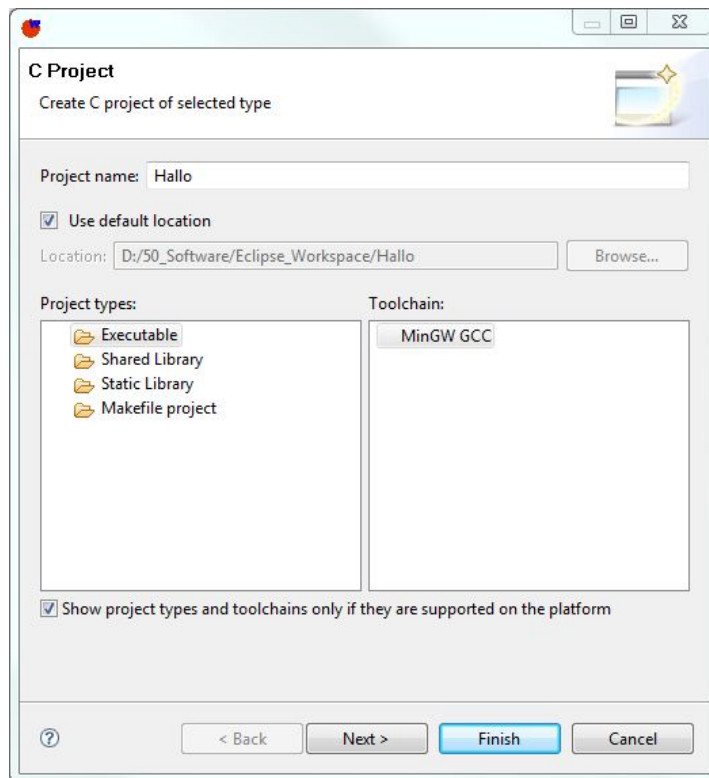


Abbildung 3.1: Erstellen eines ersten Projektes

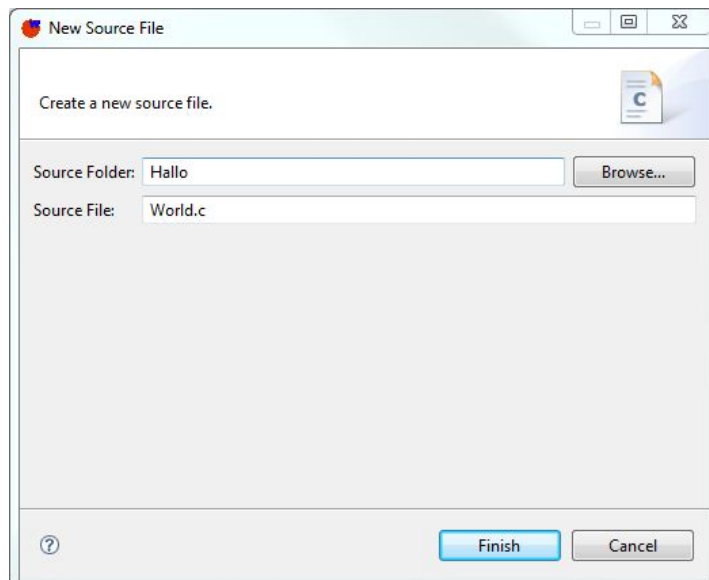


Abbildung 3.2: Erstellen einer Quelldatei

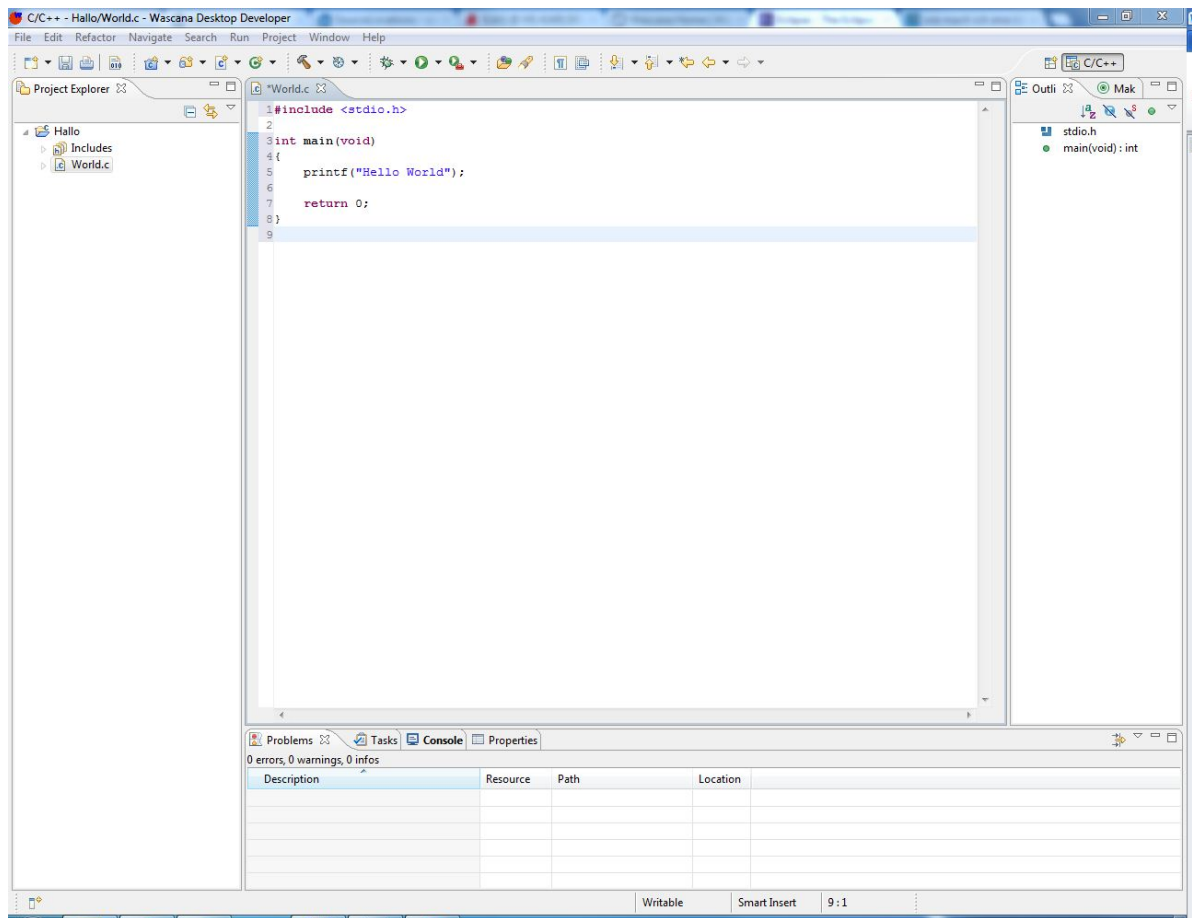


Abbildung 3.3: Hello World in Eclipse

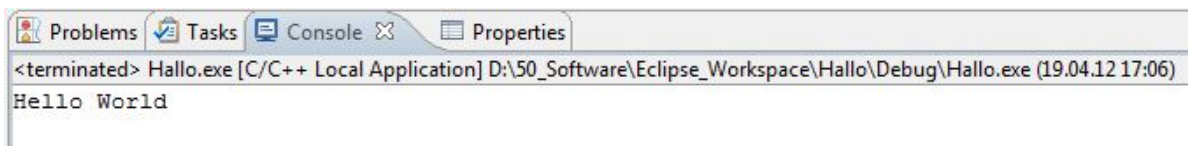


Abbildung 3.4: Konsolenausgabe in Eclipse

Der Debug wird in Eclipse über das Betätigen der Schaltfläche mit dem grünen Käfer ausgeführt. Um die einzelnen Schritte im Debug-Mode zu durchlaufen, kann der Shortcut „F6“ betätigt werden.

Für den Debug-Mode wird automatisch das „Perspective“ auf „Debug“ geändert. Um auf die ursprüngliche Darstellung von Eclipse zu gelangen, kann oben rechts über die Doppelpfeile die Ansicht gewählt werden. Wie in Abbildung 3.5 zu sehen ist lautet die ursprüngliche Ansicht „C/C++“.

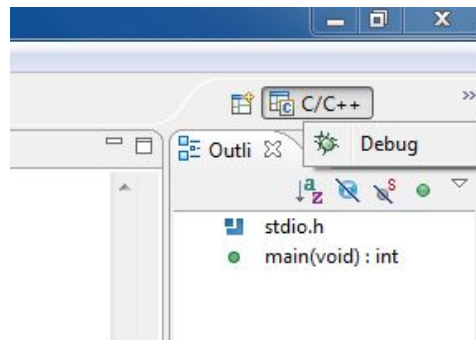


Abbildung 3.5: Wechseln der Perspektiven

4 Hinweise zu Eclipse

Zum Abschluss gibt es noch einige Hinweise für einen guten Start mit Eclipse.

Warnung beim Rückgabe-Wert

Wird die main-Funktion mit `void main(void)` aufgerufen, erscheint eine Warnung. Dies hängt mit der Definition der main-Funktion zusammen, die wie folgt lautet:

```
1  int main(int argc, char *argv[])
2  {
3      return 0;
4  }
```

Um die Warnung zu vermeiden, wird `int` als Rückgabewert gewählt (`int main...`). Dieser Rückgabewert (`int`) hat zur Folge, dass ein `return`-Wert in die Funktion eingefügt werden muss. Die Argumente `int argc` und `char *argv[]` müssen jedoch nicht zwingend aufgeführt werden. Die main-Funktion kann also folgendermaßen ausgeführt werden:

```
1  int main(void)
2  {
3      return 0;
4  }
```

Warnung durch fehlende Leerzeile

Eine weitere Warnung tritt auf, wenn nach dem Schließen der main-Funktion (durch die geschweifte Klammer) keine Leerzeile folgt. In Abbildung 3.3 ist diese Leerzeile in Zeile 9 zu erkennen.

Die Zeilen können über die folgenden Einstellungen sichtbar gemacht werden: → Window → Preferences → General → Editors → Text Editors → Show line numbers.

Hinweise zu Bildschirmausgaben und Tastaturabfragen

Bei der Verwendung der genannten Toolkette kann der Effekt auftreten, dass die Konsolenausgaben nicht instantan nach der Ausführung des entsprechenden Befehls (`printf`) erscheinen. Vielmehr wird das Programm erst „zu Ende“ ausgeführt und damit erscheinen die Ausgaben erst nach dem Verlassen der `main`-Routine.

Dieser Sachverhalt fällt dem Anwender nicht auf im Run-Mode von Programmen, welche „schnell“ terminieren. Hingegen ist der beschriebene Effekt im Debug-Modus genauso sichtbar und unerwünscht wie in Programmen, welche durch die Verwendung einer Endlosschleife nicht terminieren sollen.

Verantwortlich für die fehlende Konsolenausgabe ist die Tatsache, dass die Ausgabe durch `printf` zwar in den Ausgabepuffer gelangt, dieser aber erst nach Programmbeendigung geleert wird. Und somit ergibt sich auch die Abhilfemaßnahme: nach einem `printf`-Befehl wird durch die Anweisung `fflush(stdout)` das Leeren des Puffers erzwungen¹.

Analog kann der Eingabepuffer geleert werden durch die Anweisung `fflush(stdin)`. Eine sinnvolle Anwendung ergibt sich, um vom Benutzer falsch eingegebene Werte zu korrigieren und eine erneute Eingabe via `scanf` zu erlauben.

¹ Diese Vorgehensweise ist zwar nicht in einem definierten C-Standard enthalten, funktioniert aber in der Regel problemlos