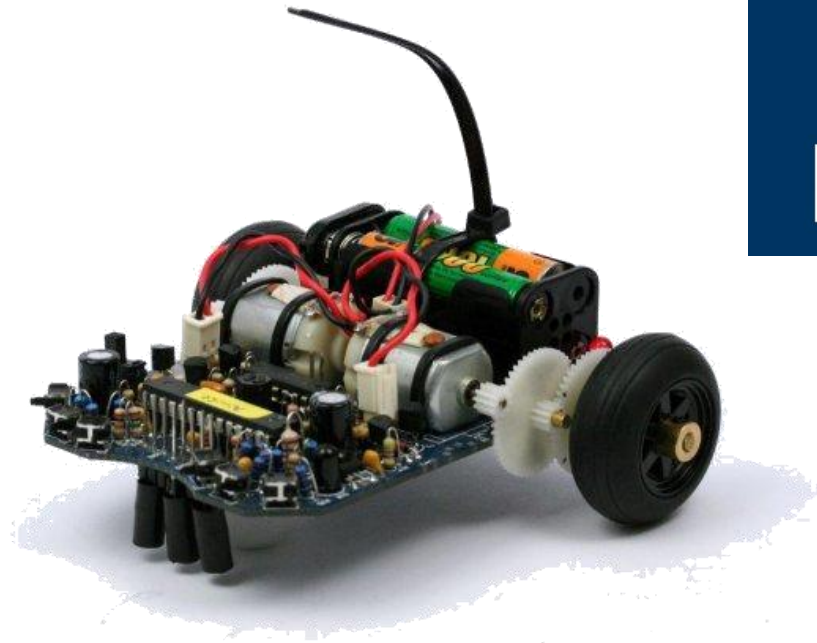


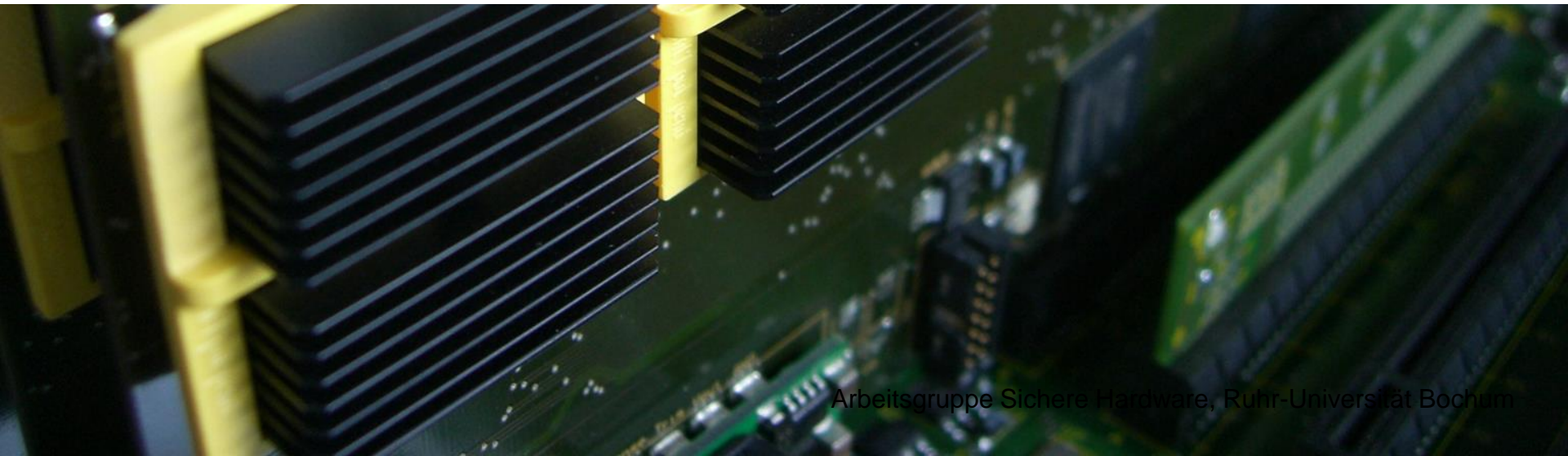
Eingebettete Prozessoren SS 2014

Übung 1: Der Asuro Roboter

Dipl.-Ing. Ingo von Maurich
Arbeitsgruppe Sichere Hardware
Horst Görtz Institut für IT-Sicherheit



17.04.2014



Agenda

- 1. Organisatorisches**
- 2. Installation der Entwicklungsumgebung**
- 3. Ein erstes Programm**
- 4. Debugging im AVR Studio**
- 5. Asuro-Roboter**
- 6. Ausleihe der Asuros**

1. Organisatorisches

- Übungstermin: Do. 12:15 – 13:00 Uhr im HID
- Ziele der Übung
 - Besprechung vergangener Hausaufgaben
 - Vorbereitung auf kommende Hausaufgaben
- Folien im Blackboard, Tafelanschrieb mitschreiben
- Hausaufgaben im Blackboard
- Normalerweise eine Woche Bearbeitungszeit
- Fragen bitte im Diskussionsforum (Blackboard)!

1. Organisatorisches

- Hausaufgaben werden korrigiert und benotet
 - Bis zu 10% Bonus für die Klausur
 - Einzel- und Gruppenaufgaben
 - Abgeschriebene Aufgaben werden mit 0 Punkten für alle Beteiligten bewertet
 - Bei Wiederholung alle Aufgaben 0 Punkte, ggf. Ausschluss vom Übungsbetrieb
- Bei Fragen zur Korrektur
Pascal Günther, Andreas Wolf, Jan Zimmer
eingebettet@crypto.rub.de

1. Organisatorisches

- Gruppenleiter reicht Gruppenaufgaben ein
 - Gruppenleiter: Nachname alphabetisch sortiert vorne
 - Beispielgruppe
 - **T. Anderson**, B. Appel, I. Büffel
- Jeder Teilnehmer muss einen Versuch einreichen, um Punkte zu bekommen (zur Not auch leer)!
- Zuletzt abgegebener Versuch wird gewertet
- Absenden nicht vergessen

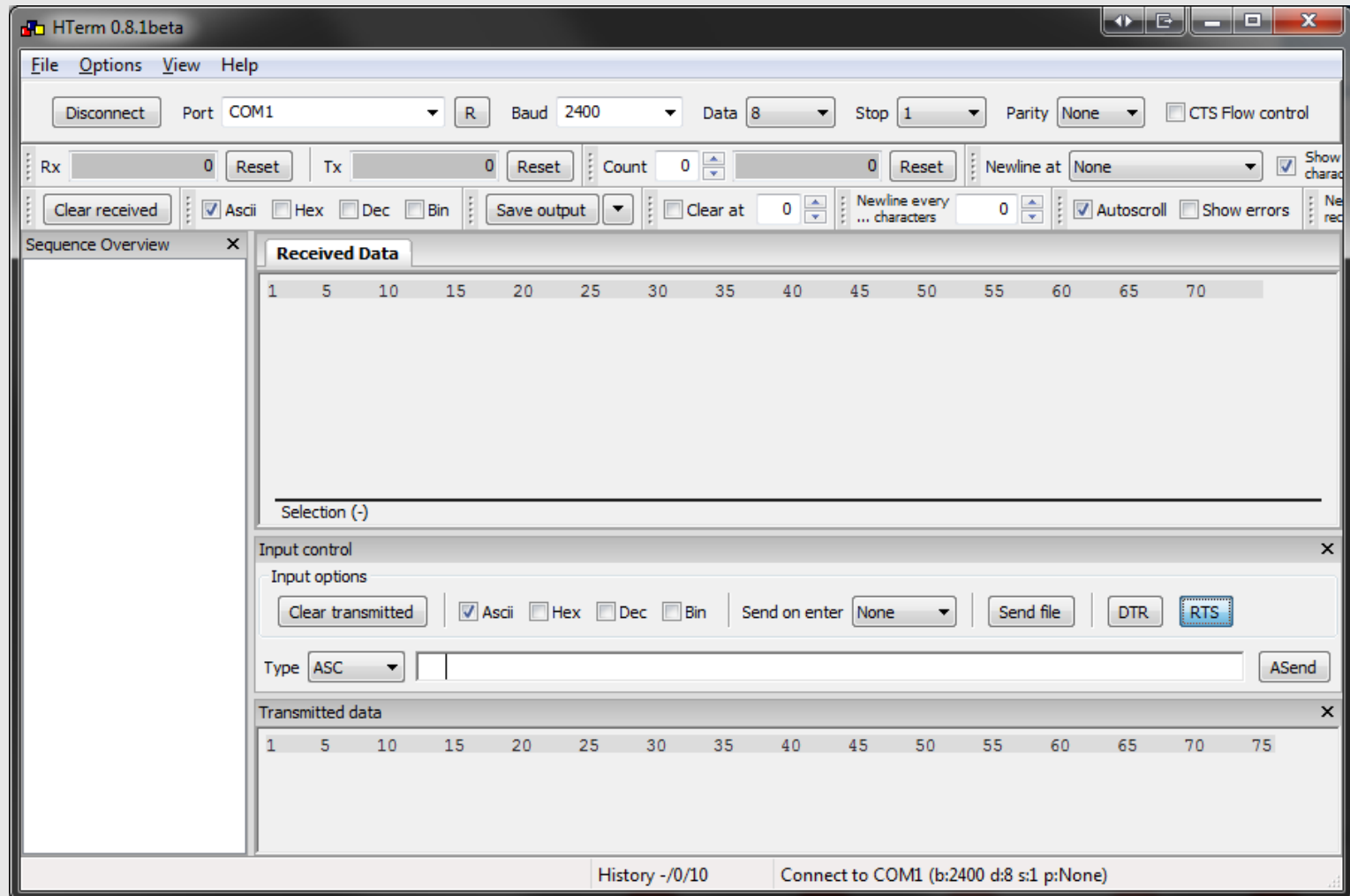
2. Installation der Entwicklungsumgebung

- Einrichtung der Kommunikationsschnittstelle
 - HTerm
 - IR-Transceiver
- Einrichtung der Programmierumgebung
 - AVR Studio
 - AVR Toolchain
 - AsuroLib
- Übertragung auf den Asuro
 - Flash Tool

2.1 Einrichtung der Kommunikationsschnittstelle

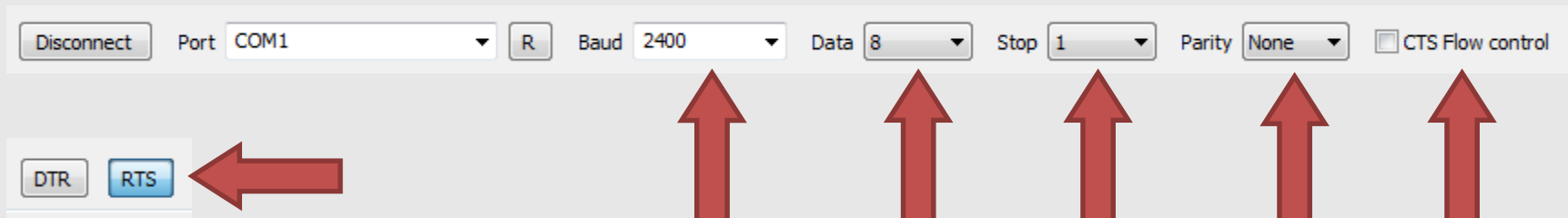
- HTerm ist ein Programm zur seriellen Kommunikation
- Sehr großer Funktionsumfang
- Windows und Linux Versionen verfügbar
- Herunterladen, keine Installation notwendig
- <http://www.der-hammer.info/terminal/>

2.1 Einrichtung der Kommunikationsschnittstelle



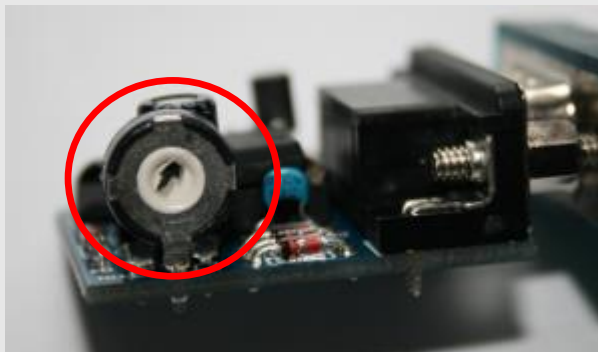
2.1 Einrichtung der Kommunikationsschnittstelle

- Wichtige Einstellungen für Kommunikation mit Asuro
 - Baud-Rate: 2400
 - Data: 8
 - Stop: 1
 - Parity: None
 - Keine CTS Flusskontrolle
 - RTS gesetzt



2.1 Einrichtung der Kommunikationsschnittstelle

- IR-Transceiver verbinden
- Ca. 10cm über weißes Blatt halten
- HTerm Zeichen senden lassen
- Kontrolle der empfangenen Zeichen
- Falls keine Übereinstimmung:
 - Vorsichtig mit einem Schraubenzieher am Trimmer drehen, testen, drehen, testen...



- Nicht notwendig bei USB-Transceivern

2.1 Einrichtung der Kommunikationsschnittstelle

- Falls keine serielle Schnittstelle vorhanden
 - Hat jemand aus der Gruppe einen PC mit serieller Schnittstelle?
 - Evtl. am Mainboard vorhanden nur nicht nach außen geführt?
- Anschluss per USB (wird nicht gestellt)
 - USB<->Seriell Adapter, bekannt dafür Probleme zu verursachen, evtl. zusätzliche Stromversorgung
 - USB-IR Transceiver ca. 20€
 - Asuro inkl. USB Transceiver ca. 50€

2.2 Programmierumgebung

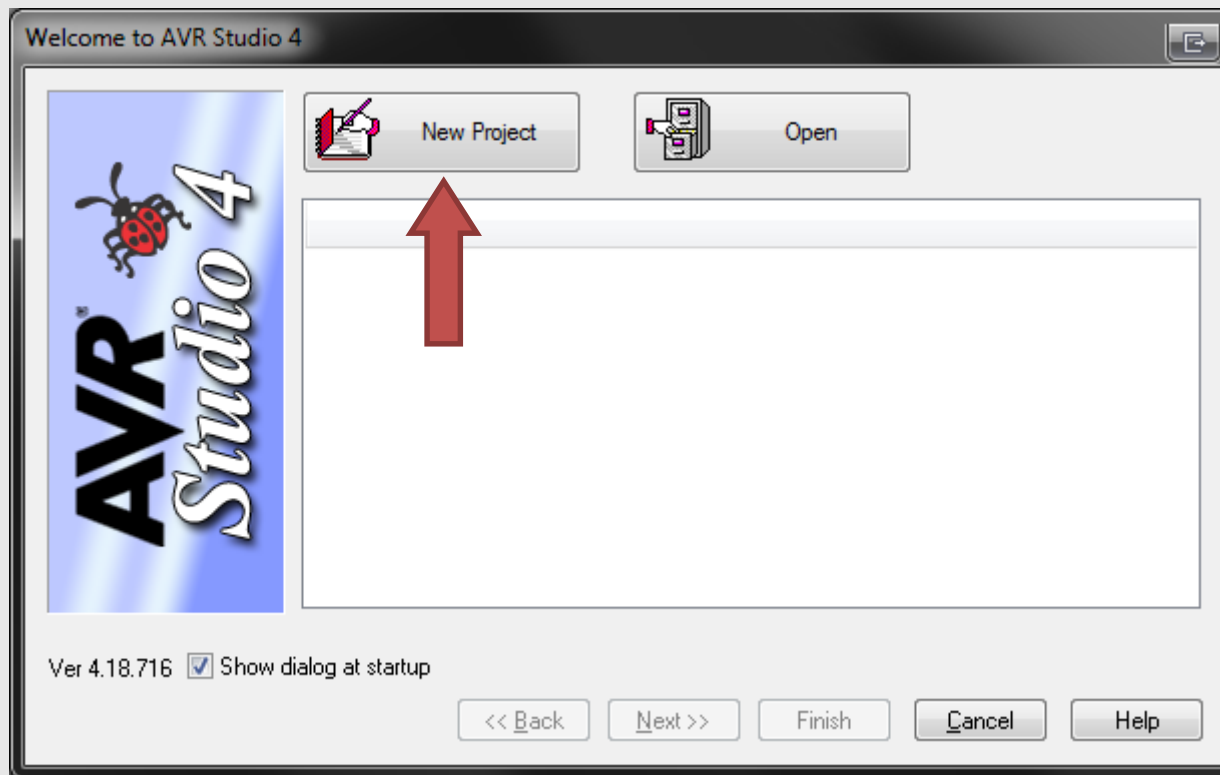
- Wir verwenden
 - AVR Studio **4.19**
 - AVR Toolchain **3.3.0**
- Herunterladen und installieren (Direktlink im Blackboard)
- Alternative
<http://www.atmel.com/tools/STUDIOARCHIVE.aspx>
- Registrierung für Download bei Atmel erforderlich, E-Mailadresse muss *gültig* sein
- Auf richtige Version achten!
Start -> Ausführen -> *cmd* -> *avr-gcc --version*
avr-gcc (AVR_8_bit_GNU_Toolchain_3.3.0_364) 4.5.1



2.2 Programmierumgebung

- Am Anfang verwenden wir eine C-Library
 - AsuroLib 2.8.0 RC1 (patched)
- Bitte aus dem Blackboard laden, da kleine Änderungen an der Library notwendig waren!
- Keine Installation notwendig, einfach entpacken
- Empfehlungen
 - Gleiche Partition wie AVR Studio & Toolchain
 - Keine Leerzeichen im Pfad

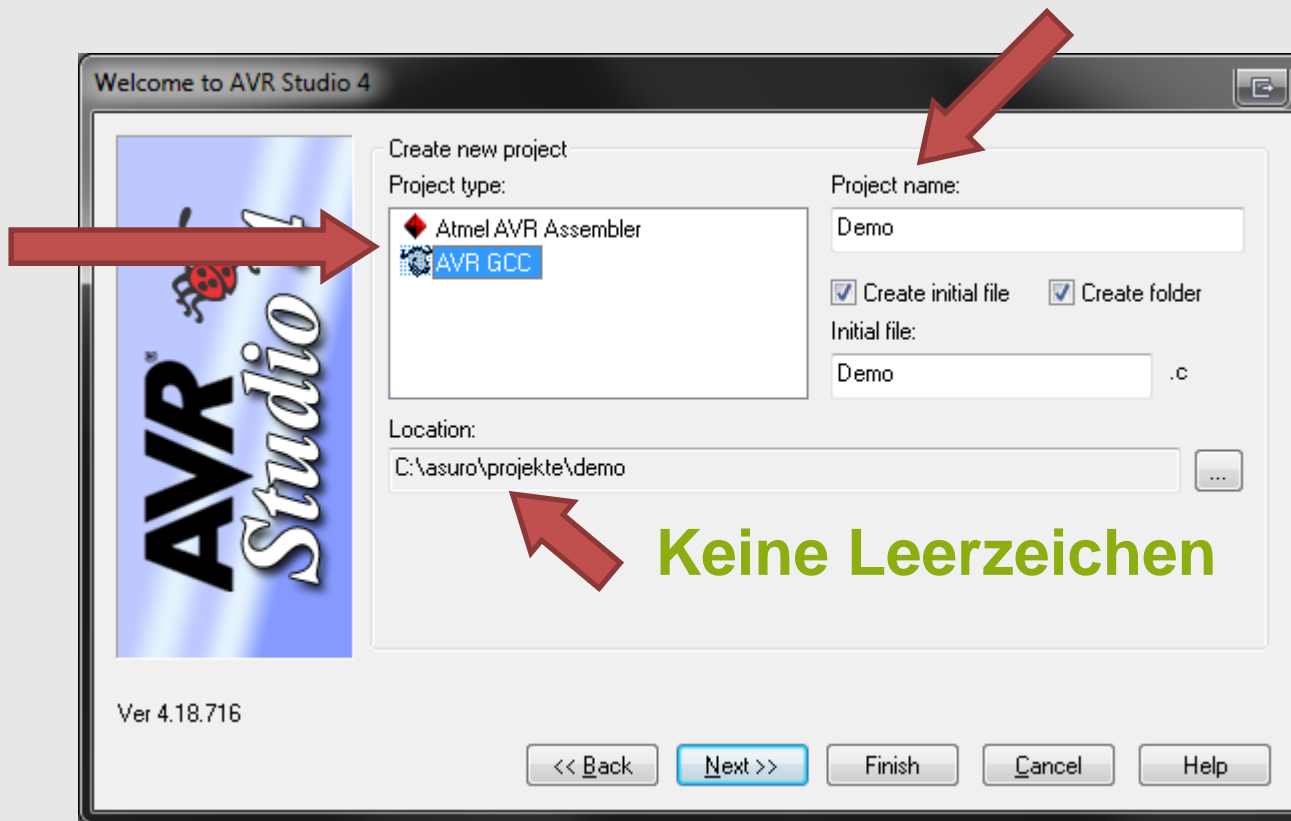
2.2 Programmierumgebung



2.2 Programmierumgebung

AVR
GCC

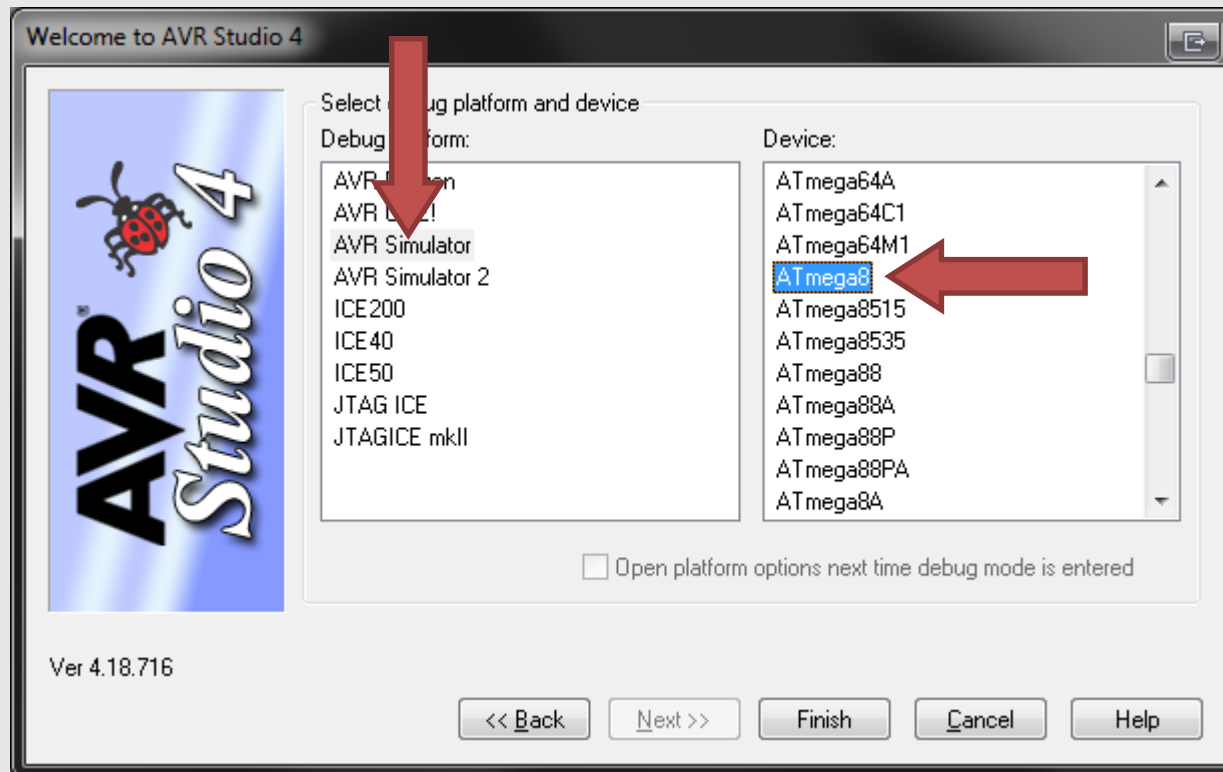
Name des Projekts



Keine Leerzeichen

2.2 Programmierumgebung

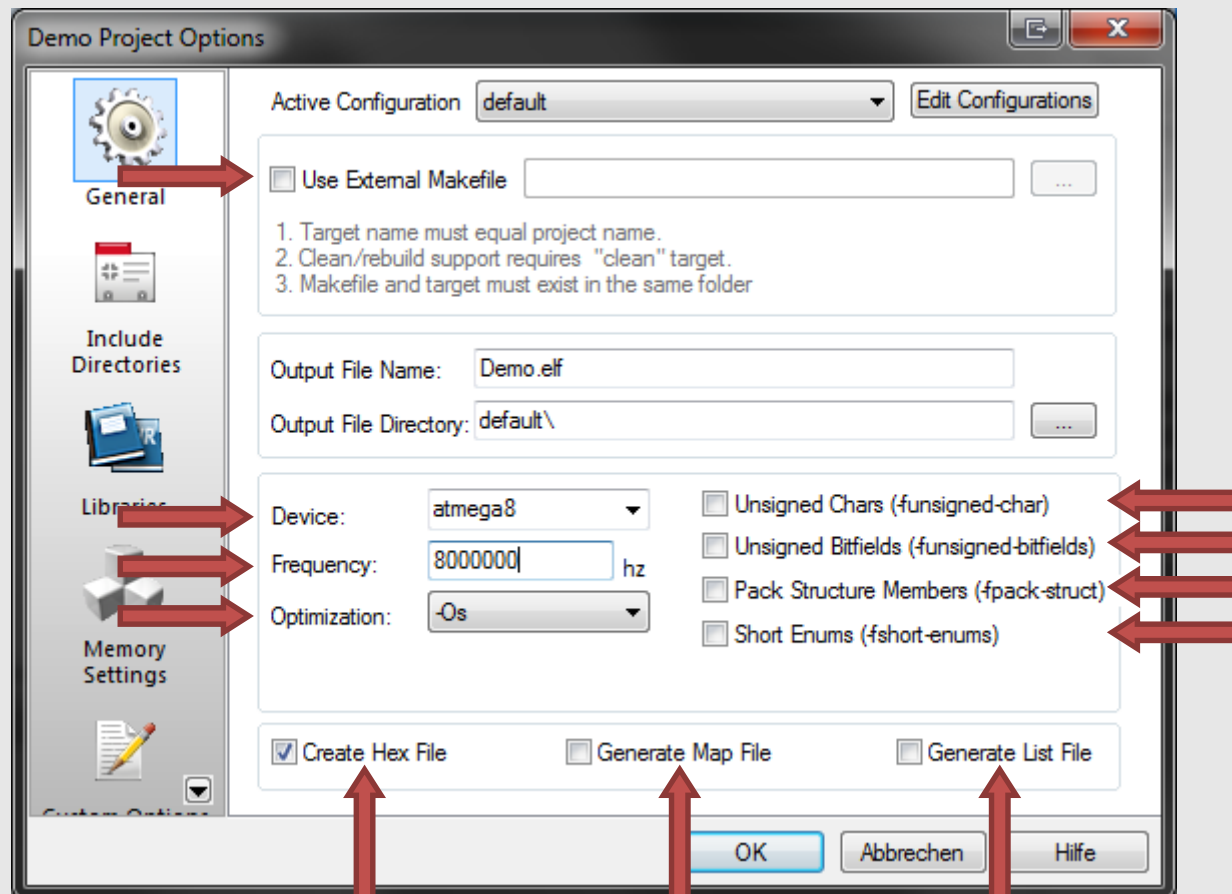
AVR Simulator



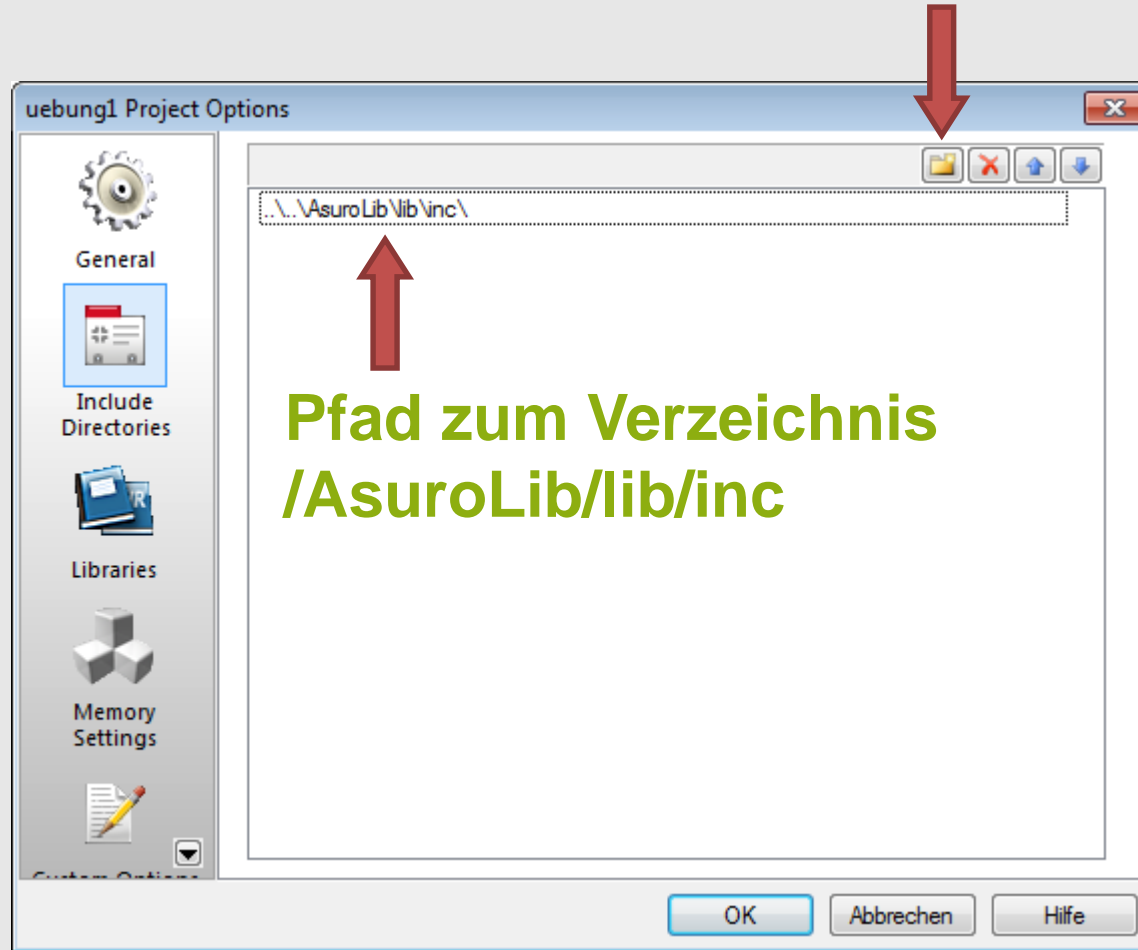
ATmega8

2.2 Programmierumgebung

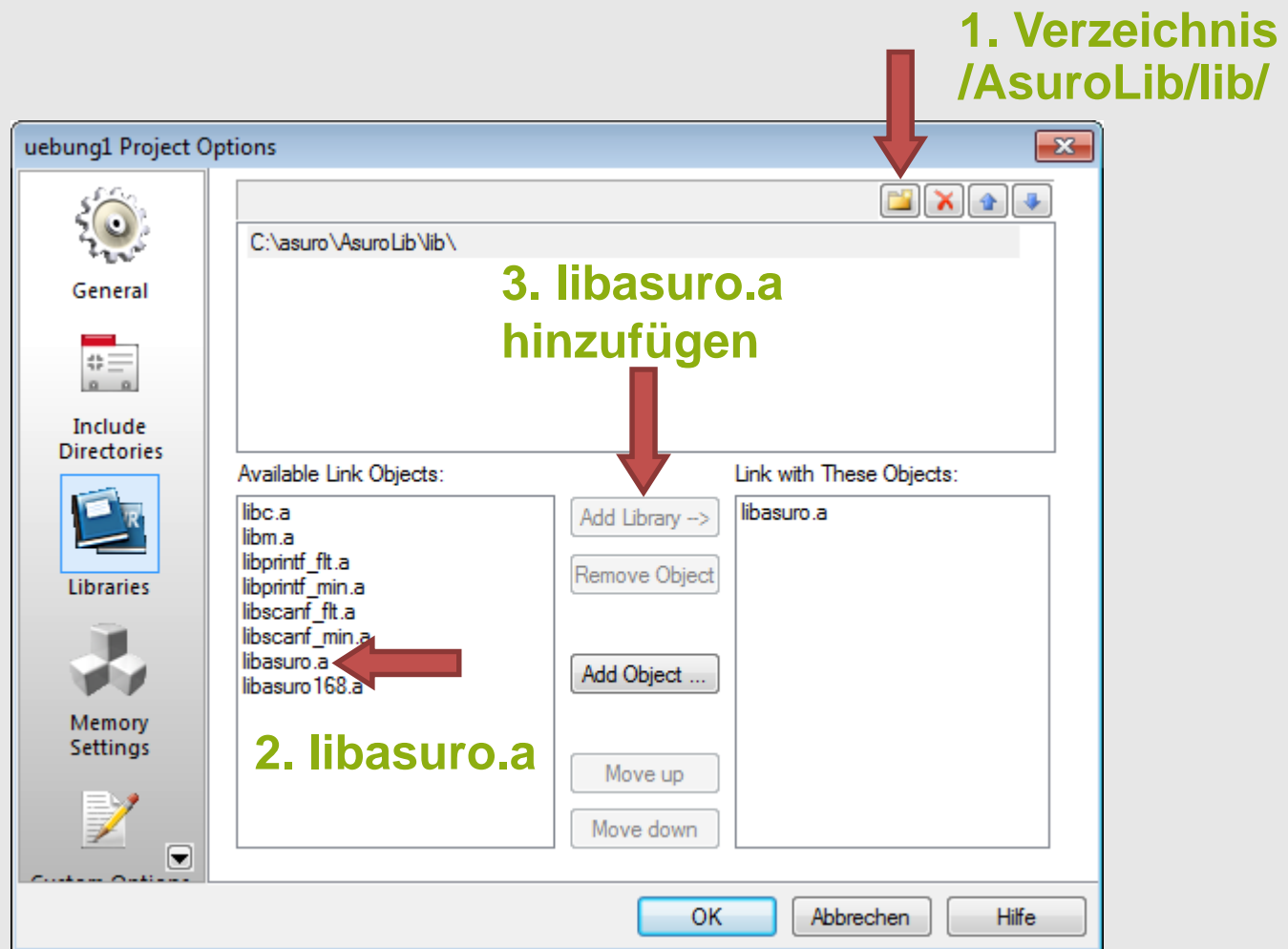
- Project → Configuration Options



2.3 Einbindung der AsuroLib

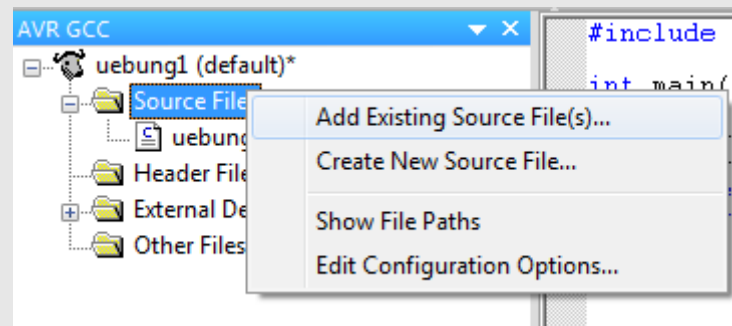


2.3 Einbindung der AsuroLib

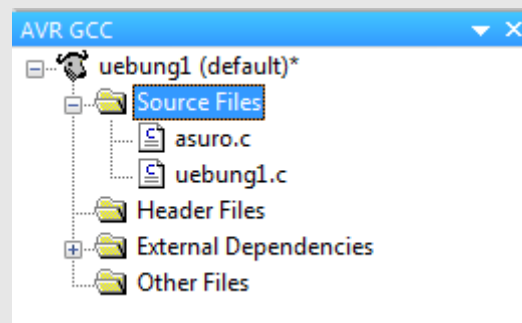


2.3 Einbindung der AsuroLib

- Hinzufügen der asuro.c zu den Source Files



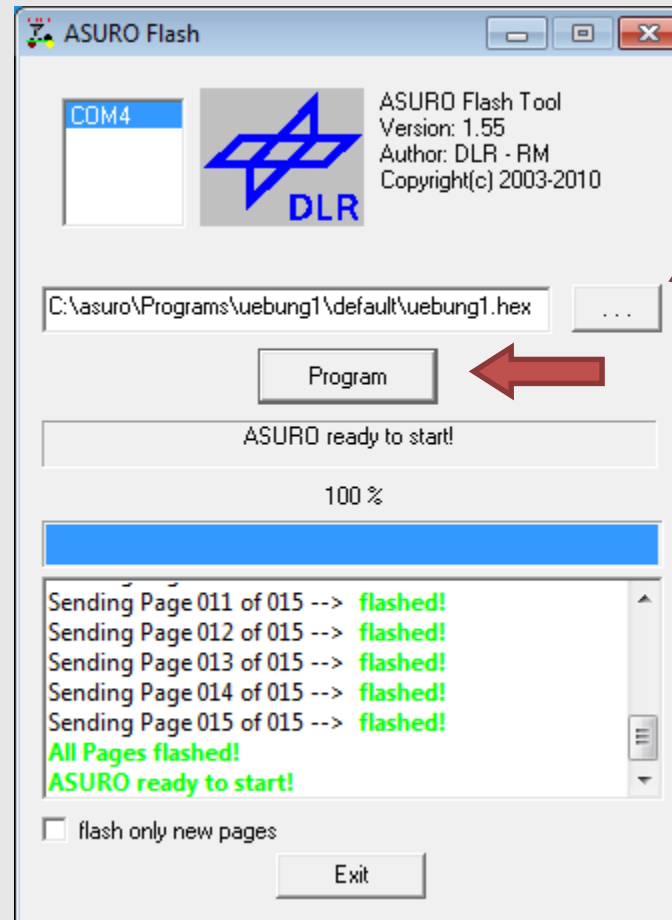
- Die asuro.c liegt im Verzeichnis /AsuroLib/lib



2.4 Übertragung auf den Asuro

- Flash.exe überträgt Programme auf den Asuro per Infrarot Schnittstelle
- Im Blackboard oder unter http://www.arexx.nl/arexx.php?cmd=goto&cparam=p_asuro_downloads.dat
- Empfehlung: neueste Version 1.55 verwenden
- Falls Probleme mit 1.55, ältere Versionen probieren

2.4 Übertragung auf den Asuro



1. *.hex Datei auswählen

2. „Program“ anklicken

3. Asuro einschalten

3. Ein erstes Programm

- Einfache Aufgabe: Asuro soll vorwärts fahren

```
#include <asuro.h>

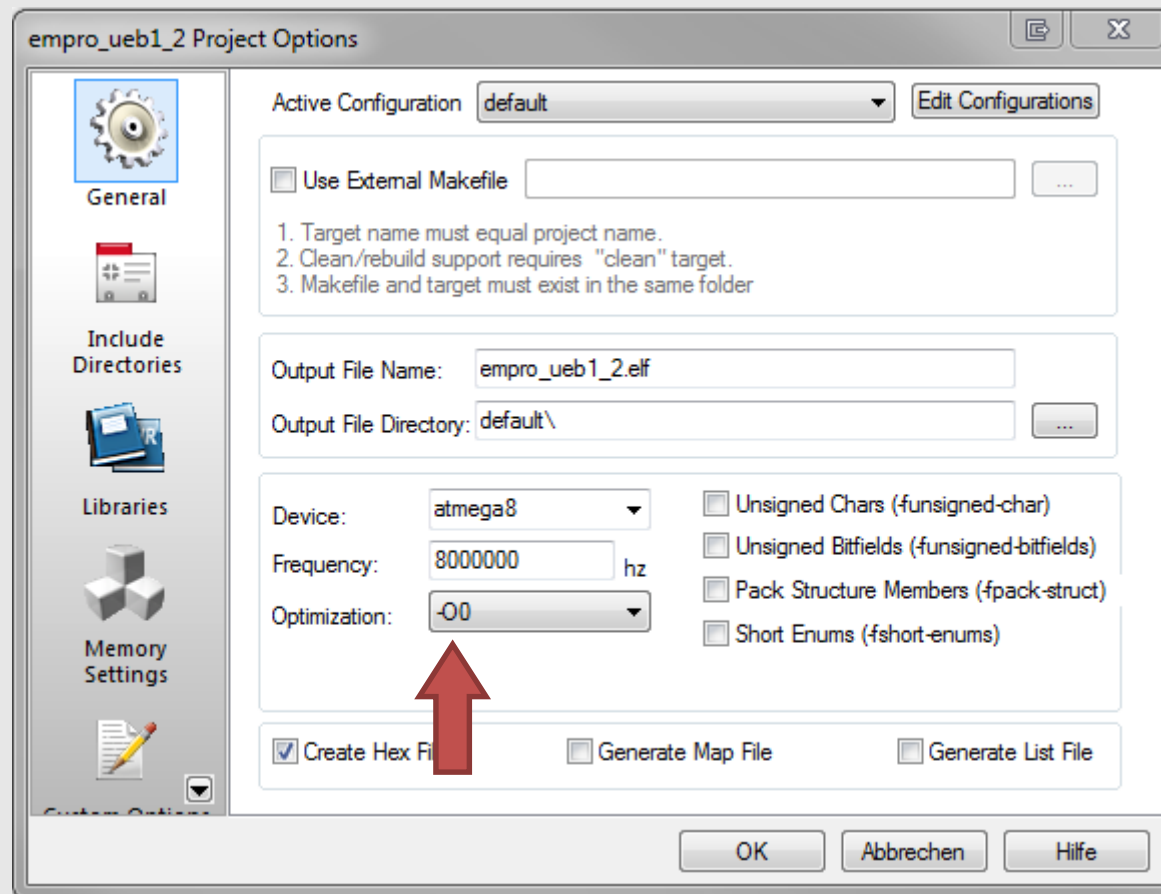
void main()
{
    Init();
    MotorDir(FWD, FWD);
    MotorSpeed(150, 150);
    while(1);
}
```

4. Debugging im AVR Studio

- AVR-Studio bietet die Möglichkeit, den Inhalt von Variablen während der Ausführung zu beobachten
- Breakpoints an beliebigen Stellen im Code
- Wichtig: Compiler-Optimierungen ausschalten -O0

4. Debugging im AVR Studio

- Deaktivierung der Compiler-Optimierungen (-O0)



4. Debugging im AVR Studio

Debugging starten

Programm schrittweise ausführen

Breakpoint

Variablen beobachten

The screenshot shows the AVR Studio interface with the following components:

- Processor Window:** Displays system variables such as Program Counter (0x000031), Stack Pointer (0x0457), X pointer (0x0000), Y pointer (0x0457), Z pointer (0x0000), Cycle Counter (123), Frequency (4.0000 MHz), Stop Watch (30.75 us), SREG, and Registers.
- Code Editor:** Contains the following C code:

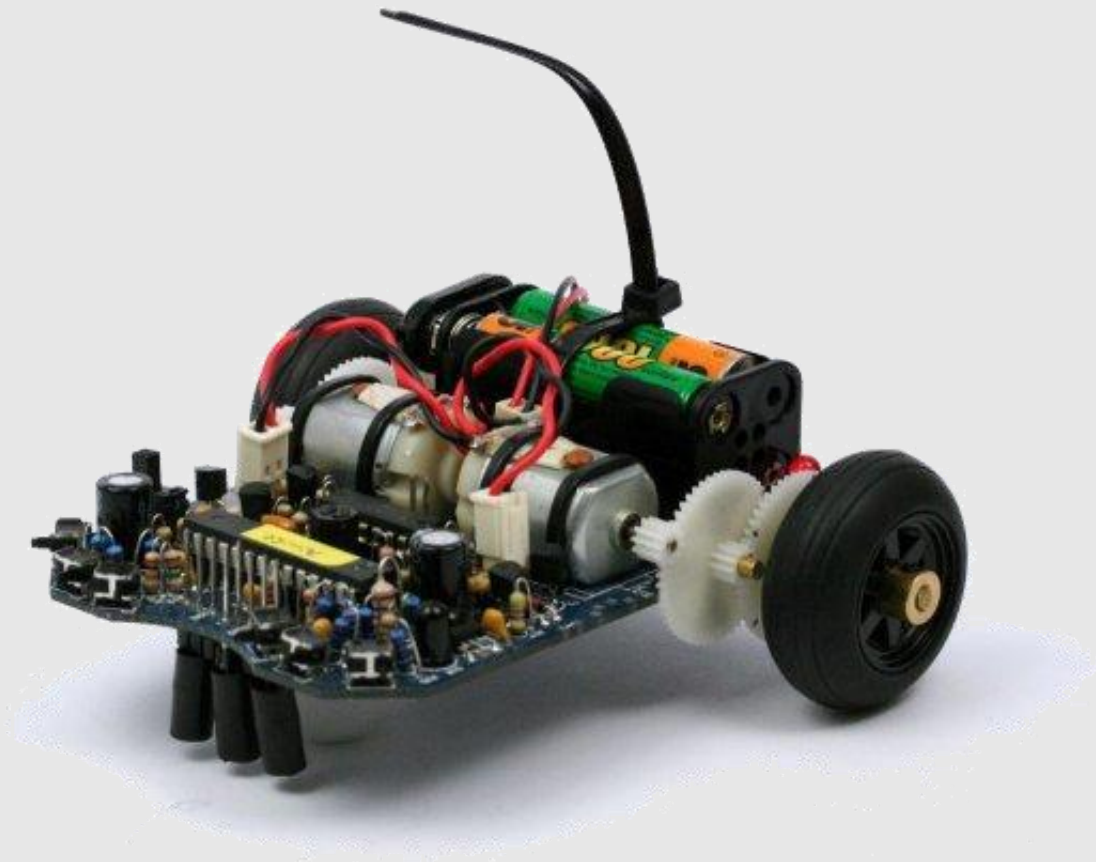
```
int main(void)
{
    int i,j;
    for(i=j=0;i<10;i++)
        j=j+i;
    while(1);
}
```

A yellow arrow points to the first line of the code, and a red dot indicates a breakpoint set on the opening curly brace of the main function.
- Watch Window:** A table for monitoring variables:

Name	Value
i	0x0002
j	0x0003

Red arrows point to the 'i' and 'j' rows in this window.
- Toolbar:** The top toolbar contains various icons for file operations, editing, and debugging. A red arrow points to the 'Step Into' (F7) icon, which is used for single-stepping through the code.

5. Asuro Roboter



5. Asuro Roboter

- Im Karton: Asuro, IR-Transceiver, Kabel
- Karton und Inhalt bitte **nicht** zerstören, ihr seid nach der Ausleihe für den Asuro verantwortlich
- Asuro ist empfindlich!
- Nicht
 - fallen lassen,
 - Elektronik berühren,
 - Flüssigkeiten drüber kippen, etc.
- Vorsicht beim Flashen! Asuro nicht an Tischkante stellen, sonst fährt er ggf. los und fällt

6. Ausleihe der Asuros

- In 3er Gruppen zusammenfinden
- Ausgefüllten Leihschein nach vorne bringen
- Asuro in Empfang nehmen und den Inhalt des Kartons auf Vollständigkeit prüfen
- Ab jetzt seid ihr für den Asuro verantwortlich
- Nach Ende der Vorlesung den Asuro zurückgeben, ansonsten behalten wir uns vor das Klausurergebnis zurückzuhalten, bis der Asuro bei uns eingetroffen ist

6. Ausleihe der Asuros

