

Hardwarenahe Softwareentwicklung

Einstieg Embedded Systems

IDE (Integrated Development Environment)

Lernziele

Sie sind in der Lage:

- ▶ Die Einsatzgebiete von Embedded Systems zu erklären.
- ▶ Den Entwicklungsablauf für die hardwarenahe Softwareentwicklung zu erklären.
- ▶ Die Entwicklungsumgebung anzuwenden.



Inhaltsverzeichnis

1. Definitionen und Einsatzgebiete

Definitionen

Einsatzgebiete

2. Entwicklungsablauf

3. Entwicklungsumgebungen

Definitionen und Einsatzgebiete

Definitionen

Hardwarenahe Softwareentwicklung heisst:

- ▶ Die Software wird nicht auf einem PC oder Laptop ausgeführt, sondern auf einem Embedded System.
- ▶ Die Zielhardware (Target) ist ein Microcontroller.
- ▶ Beispiel: Ihr Programm läuft nicht auf dem Laptop, sonder auf dem Leguan-Board.

Definition Embedded System / Embedded Computing:

[SwissT.net]

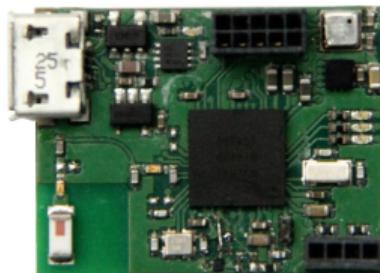
Unter Embedded Computing versteht man den Einsatz von Rechnersystemen als integrierter Bestandteil einer Anlage, eines Gerätes oder einer Maschine. Embedded Computing umfasst industrielle Betriebssysteme, Software-Tools, Applikationssoftware und Hardware-Plattformen und dient der Steuerung von Prozessen, der automatisierten Erfassung, Verarbeitung und Aufbereitung von Daten.

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



Quelle: www.consumerreports.org



Quelle: BFH-TI

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



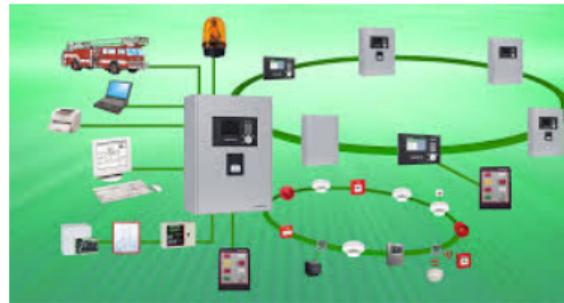
Quelle: Cardiac Surgery, University of Michigan



Quelle: Bayer Vital GmbH

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



Quelle: Securiton



Quelle: tagesanzeiger.ch

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



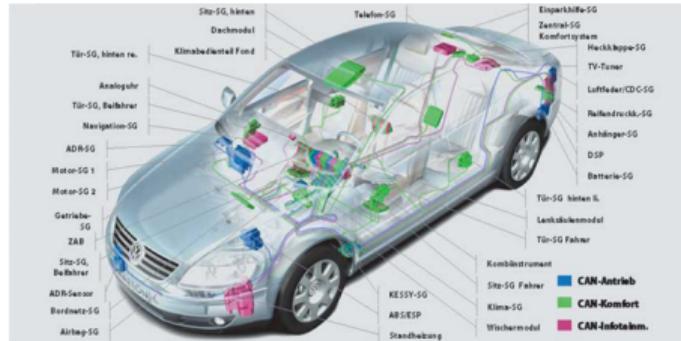
Quelle: www.industr.com



Quelle: BOSCH Rexroth

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



Quelle: homepages.thm.de



Quelle: eda.admin.ch

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



Quelle: BFH-TI / I3S



Quelle: Eversys

Einsatzgebiete

- ▶ Kommunikation
- ▶ Medizinaltechnik
- ▶ Sicherheitstechnik
- ▶ Industrieautomation
- ▶ Verkehrstechnik
- ▶ Consumer-Electronic
- ▶ Energietechnik



Quelle: www.die-partner.net



Quelle: www.photovoltaik.org

Entwicklungsablauf

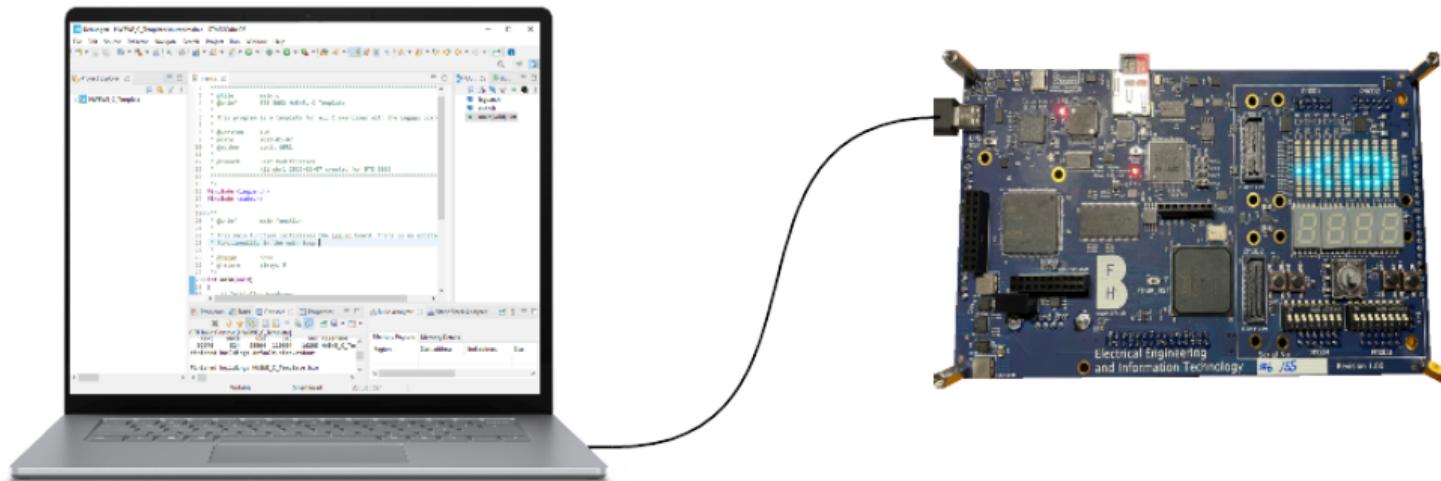
Entwicklungsablauf I

Für die Hardwarenahe Softwareentwicklung werden folgende Tools verwendet:

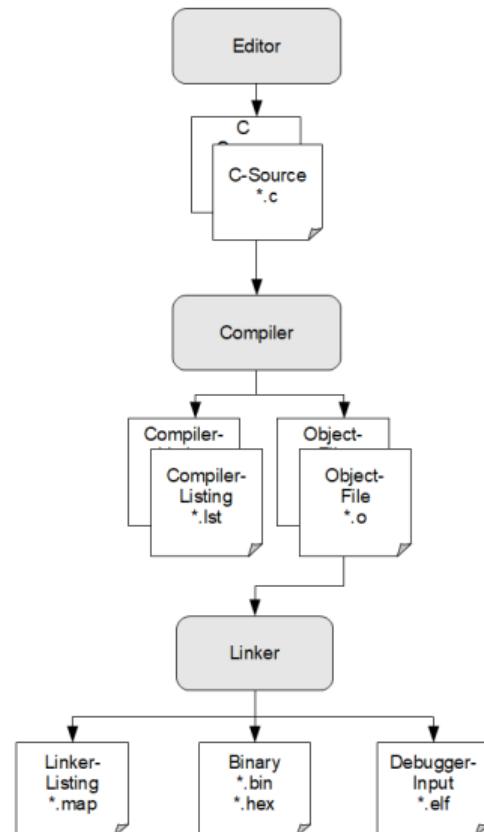
- ▶ Editor
- ▶ Assembler
- ▶ Compiler
- ▶ Linker/Locator
- ▶ Simulator / Debugger

Entwicklungsablauf II

Die Tools werden auf einem PC/Laptop ausgeführt und erzeugen ausführbare Programme für die Zielhardware.



Ablauf einer Software-Entwicklung in C



Editor

- Mit Hilfe des Editors wird der Quellcode geschrieben.
- Syntax-sensitiv (farbige Schlüsselwörter, Kommentare usw.).
- Beispiel: Mit dem Editor wurde folgender C-Code erstellt:

```
int main( void )
{
    /* Initialize Hardware */
    CUBEMX_Init() ;
    /* Initialize Leguan board */
    LEGUAN_Init() ;

    /* Set logging output destination to be the LCD */
    LOG_SetDestination(LCD_Stream) ;

    /* Main loop */
    for ( ; ) {

    }

    return (0) ;
}
```

Compiler

- Übersetzt den C-Code in ein Objekt-File.
- Erzeugt ein Listing (optional).
- CPU-spezifisch

Linker / Locator

- ▶ Aufgabe des Linker/Locator:
 - ▶ alle Objekt-Files eines Projektes “zusammenbinden“ (Linker).
 - ▶ absolute Adressen für Funktionen und Variablen zuweisen (Locator).
- ▶ Erzeugte Dateien:
 - ▶ Download in den Programmspeicher (.bin)
 - ▶ Debugger (.elf)
 - ▶ List- und MAP-Files
- ▶ Das elf-Format (Executable and Linking Format):
 - ▶ Standard Debug-Exec-Format.
 - ▶ Einbinden von: Direkt ausführbarer Code, Objektcode, Bibliotheken, Speicherabbilder, Debuginformationen.
 - ▶ Kann mit “objdump.exe“ eingesehen werden.

Simulator / Debugger

Simulator-Mode:

- ▶ Frühes Projektstadium.
- ▶ Hardware noch nicht verfügbar,
- ▶ oder hardwareunabhängige Algorithmen.

Debugger-Mode:

- ▶ Auf der Hardware (Target).
- ▶ Test in Echtzeit.

Entwicklungsumgebungen

Übersicht

Entwicklungsumgebungen / Integrated Development Environment (IDE), beinhalten

- ▶ Editor
- ▶ Toolchain (Compiler, Linker)
- ▶ Tools für das Debugging
- ▶ Weitere Tools für die effiziente Programmierung (statische Code-Analyse, Pprofiling ...)

Entwicklungsumgebungen auf dem Markt:

- ▶ Vom Chip-Hersteller, nur für seine Microcontroller-Familien, oft gratis. Bsp: STM32CubeIDE für STM32 CPUs.
- ▶ Chip-Hersteller unabhängig, oft kostenpflichtig. Bsp. Arm Development Studio oder Segger Embedded Studio für alle Cortex-Mx CPUs.

STM32CubeIDE

