

Hardwarenahe Software-Entwicklung

Prof. Dr. rer. nat. Peter Gerwinski

Hardwarenahe IT-Systeme
Campus Velbert/Heiligenhaus

Hardwarenahe Software-Entwicklung

Hardwarenahe Software-Entwicklung

- Rechnertechnik:
Wie funktioniert ein Computer?

Hardwarenahe Software-Entwicklung

- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Rechnertechnik:
Wie funktioniert ein Computer?

Hardwarenahe Software-Entwicklung

- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Hardwarenahe
Software-Entwicklung
- Rechnertechnik:
Wie funktioniert ein Computer?

Hardwarenahe Software-Entwicklung

- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Hardwarenahe
Software-Entwicklung:
Gerätetreiber
- Rechnertechnik:
Wie funktioniert ein Computer?

Hardwarenahe Software-Entwicklung

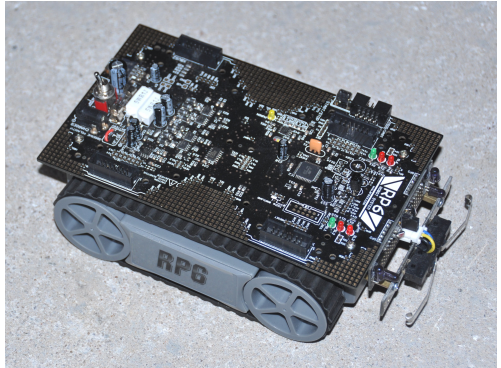
- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Hardwarenahe
Software-Entwicklung:
Gerätetreiber,
eingebettete Systeme
- Rechnertechnik:
Wie funktioniert ein Computer?

Hardwarenahe Software-Entwicklung

- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Hardwarenahe
Software-Entwicklung:
Gerätetreiber,
eingebettete Systeme,
Mikro-Controller
- Rechnertechnik:
Wie funktioniert ein Computer?

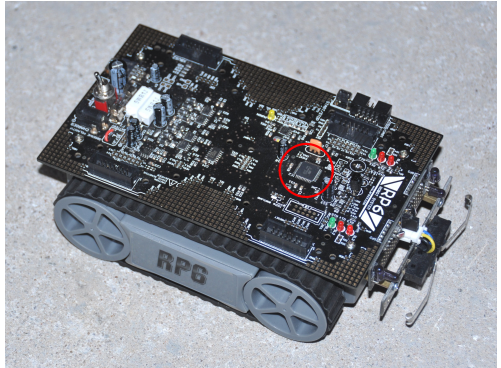
Hardwarenahe Software-Entwicklung

- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Hardwarenahe
Software-Entwicklung:
Gerätetreiber,
eingebettete Systeme,
Mikro-Controller
- Rechnertechnik:
Wie funktioniert ein Computer?



Hardwarenahe Software-Entwicklung

- Softwaretechnik:
Graphische
Benutzeroberflächen,
Datenbanken, ...
- Hardwarenahe
Software-Entwicklung:
Gerätetreiber,
eingebettete Systeme,
Mikro-Controller
- Rechnertechnik:
Wie funktioniert ein Computer?



Hardwarenahe Software-Entwicklung

Programmierung

Hardwarenahe Software-Entwicklung

Programmierung in C

Hardwarenahe Software-Entwicklung

Programmierung in C

- Kleinster gemeinsamer Nenner für viele Plattformen

Hardwarenahe Software-Entwicklung

Programmierung in C

- Kleinster gemeinsamer Nenner für viele Plattformen

Hardware und/oder Betriebssystem

Hardwarenahe Software-Entwicklung

Programmierung in C

- Kleinster gemeinsamer Nenner für viele Plattformen



Hardware und/oder Betriebssystem

- Programmierkenntnisse werden nicht vorausgesetzt,
aber schnelles Tempo

Hardwarenahe Software-Entwicklung

Programmierung in C

- Kleinster gemeinsamer Nenner für viele Plattformen



Hardware und/oder Betriebssystem

- Programmierkenntnisse werden nicht vorausgesetzt, aber schnelles Tempo
- Hardware direkt ansprechen

Hardwarenahe Software-Entwicklung

Programmierung in C

- Kleinster gemeinsamer Nenner für viele Plattformen



Hardware und/oder Betriebssystem

- Programmierkenntnisse werden nicht vorausgesetzt, aber schnelles Tempo
- Hardware direkt ansprechen
- Hardware effizient einsetzen

Hardwarenahe Software-Entwicklung

Was ist C?

Hardwarenahe Software-Entwicklung

Was ist C?

- kleinster gemeinsamer Nenner für viele Plattformen

Hardwarenahe Software-Entwicklung

Was ist C?

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

- kein „Fallschirm“

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

- kein „Fallschirm“
- kompakte Schreibweise

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

- kein „Fallschirm“
- kompakte Schreibweise

Unix-Hintergrund

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

- kein „Fallschirm“
- kompakte Schreibweise

Unix-Hintergrund

- Baukastenprinzip

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

- kein „Fallschirm“
- kompakte Schreibweise

Unix-Hintergrund

- Baukastenprinzip
- konsequente Regeln

Hardwarenahe Software-Entwicklung

Was ist C?

Etabliertes Profi-Werkzeug

- kleinster gemeinsamer Nenner für viele Plattformen
- leistungsfähig, aber gefährlich

„High-Level-Assembler“

- kein „Fallschirm“
- kompakte Schreibweise

Unix-Hintergrund

- Baukastenprinzip
- konsequente Regeln
- kein „Fallschirm“

1 Einführung in C

1 Einführung in C

1.1 Hello, world!

1 Einführung in C

1.1 Hello, world!

Text ausgeben

1 Einführung in C

1.1 Hello, world!

Text ausgeben

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    printf ("Hello, _world!\n");
```

```
    return 0;
```

```
}
```


1 Einführung in C

1.3 Zahlenwerte ausgeben

Wert ausgeben

```
#include <stdio.h>
```

```
int main (void)
```

```
{  
    printf ("Die_Antwort_lautet:_%d\n", 42);  
    return 0;  
}
```

1 Einführung in C

1.4 Elementares Rechnen

Wert einlesen

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int a;
```

```
    scanf ("%d", &a);
```

```
    printf ("Das_Doppelte_ist:_%d\n", 2 * a);
```

```
    return 0;
```

```
}
```

1 Einführung in C

1.4 Elementares Rechnen

Wert an Variable zuweisen

```
#include <stdio.h>
```

```
int main (void)
```

```
{
```

```
    int a;
```

```
    scanf ("%d", &a);
```

```
    a = 2 * a;
```

```
    printf ("Das_Doppelte_ist:_%d\n", a);
```

```
    return 0;
```

```
}
```