

Programmieren in C: Grundlagen

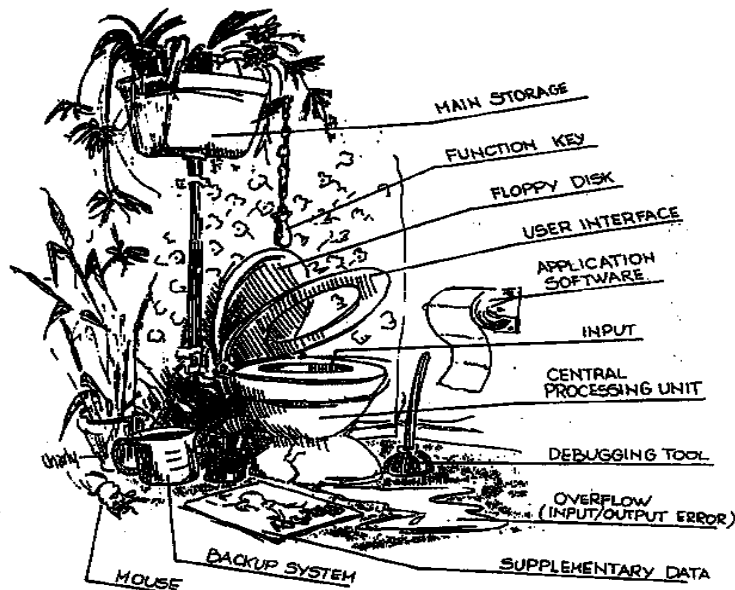
19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 1

Understanding the Computer Technology



19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 2

Programmieren in C

- Die Lehrveranstaltung erläutert die Entwicklung von Software exemplarisch am Beispiel der

➡ **Programmiersprache C**

- In diesem Kapitel wollen wir zeigen,

➡ **wie Zahlen und Daten im Computer dargestellt werden:**

- **positive und 2-Komplement Zahlen**
- **Buchstaben, Ziffern und Sonderzeichen.**

19.11.2004



Bedeutung der Programmiersprache C

- Heute gilt:

Ohne Programmieren keine Elektrotechnik .

➡ Zur Zeit enthalten viele elektronische Geräte (z.B. ABS, Fernseher) Mikroprozessoren, die in aller Regel in C programmiert werden.

➡ Die Internet Programmiersprache **JAVA** ähnelt der **Programmiersprache C**.

➡ Hardware wird heute mit der Programmiersprache **VHDL** entwickelt, die der **Programmiersprache C** ähnlich ist.

19.11.2004



Geschichte der Programmiersprache C

- Entwickelt von *D. Ritchie* in den 70er Jahren zur Implementierung des Betriebssystems *UNIX*.
- De Facto Standard war jahrelang das Buch: *Kernighan, B.W.; Ritchie, D.M.: C Programming Language, Prentice-Hall 1978*.
- 1989: C wird vom *ANSI (American National Standards Institute)* modifiziert und genormt (alter *ANSI C Standard*).
- 1996: Der C-Standard wird vom *ANSI (American National Standards Institute)* geringfügig modifiziert und von der *ISO (International Standards Organisation)* übernommen (*ANSI C, ISO-Standard C*).

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 5

Positive Zahldarstellung: Bit und Byte (1)

- Ein **Computer** verwendet intern ausschließlich die **binäre Darstellung von Zahlen, Daten und Befehlen**.
Bei der **binären Darstellung** gibt es nur die 2 Ziffern **0 und 1**.
- Die kleinste Informationseinheit (binäre Ziffer) ist **1 Bit (binary digit)**.
- 8 Bit bezeichnet man auch als **1 Byte**.

➡ 1 Byte = 8 Bit

Bit								1 Bit
Wertigkeit	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bitnummer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1 Byte = 8 Bit	1	0	1	1	1	0	1	0

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 6

Positive Zahldarstellung: Bit und Byte (2)

- Ein Computer verwendet intern **ausschließlich die binäre Darstellung** von Zahlen, Daten und Befehlen.

Bei der binären Darstellung gibt nur die 2 Ziffern 0 und 1.

- Das höchstwertige Bit nennt man **MSB (most significant bit)**. Es hat die **Wertigkeit 2^7** .
- Das niederwertigste Bit nennt man **LSB (least significant bit)**. Es hat die **Wertigkeit 2^0** .

Bit	MSB				1 Bit			LSB
Wertigkeit	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bitnummer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1 Byte = 8 Bit	1	0	1	1	1	0	1	0

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 7



Zahlbereich für positive Zahlen bei 1 Byte und bei 16 Bit

Welche größte und welche kleinste positive (vorzeichenlose) Zahl (*unsigned number*) kann man mit **1 Byte = 8 Bit** darstellen?

Welche größte und welche kleinste positive (vorzeichenlose) Zahl (*unsigned number*) kann man mit **16 Bit** darstellen?

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 8

✓ Zahlbereich für positive Zahlen bei 1 Byte

- positive 1-Byte-Zahl (*unsigned character*)

$$Z_p = \sum_{i=0}^{N-1} a_i \cdot 2^i \quad a_i = \begin{cases} 0 \\ 1 \end{cases} \quad N=8 \quad \text{1 Byte (unsigned character)}$$

größte
darstellbare Zahl

• **1 Byte** $a_i = 1, i = 0 \dots 7$

$$= 1111 \ 1111_2$$

$$= 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + \dots$$

$$\dots + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$= 2^8 - 1 = 255_{10}$$

kleinste
darstellbare Zahl

$$a_i = 0, i = 0 \dots 7$$

$$= 0000 \ 0000_2$$

$$= 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + \dots$$

$$\dots + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$= 0_{10}$$



✓ Zahlbereich für positive Zahlen bei 16 Bit

- positive 16-Bit Zahl (*unsigned int, unsigned short*)

$$Z_p = \sum_{i=0}^{N-1} a_i \cdot 2^i \quad a_i = \begin{cases} 0 \\ 1 \end{cases} \quad N=16 \quad \text{16 Bit (unsigned int, unsigned short)}$$

größte
darstellbare Zahl

• **16 Bit** $a_i = 1, i = 0 \dots 15$

$$= 1111 \ 1111 \ 1111 \ 1111_2$$

$$= 2^{16} - 1 = 65535_{10}$$

kleinste
darstellbare Zahl

$$a_i = 0, i = 0 \dots 15$$

$$= 0000 \ 0000 \ 0000 \ 0000_2$$

$$= 0_{10}$$



Zahldarstellung im Computer

- In einem Computer müssen im Allgemeinen außer **positiven ganzen Zahlen** auch
 - vorzeichenbehaftete ganze Zahlen (2-Komplement)**
 - vorzeichenbehaftete Gleitkommazahlen**
 - alphanumerischen Zeichen**
- dargestellt werden können.

Beispiele:

- positive ganze Zahlen:
0, 375, 2, 32767 (*unsigned char, unsigned short, unsigned long*)
- vorzeichenbehaftete ganze Zahlen:
-35, +70, 16000 (*signed char, signed short, signed long*)
- Gleitpunktzahlen (immer vorzeichenbehaftet):
0., -3.5, -0.7, -1.6e3, 2.E-4 (*float, double, long double*)
- alphanumerische Zeichen (z.B. Buchstaben):
"E1A/PR1" (*char*)

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 11

19.11.2004

2-Komplement Zahldarstellung

- Bei der Darstellung von relativen Zahlen, **interpretiert man das MSB mit negativer Wertigkeit.**

1 Byte als 2-Komplementzahl

Bit	MSB			1 Bit				LSB
Wertigkeit	-2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bitnummer	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1 Byte = 8 Bit	1	0	1	1	1	0	1	0

- Man bezeichnet diese Darstellung als **2-Komplementzahl** und in **C: als signed number** (vorzeichenbehaftete Zahl).
 - 1 Byte 2-Komplementzahlen liegen im Zahlbereich:
 $+127_{10} \dots -128_{10}$

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 12

19.11.2004



2-Komplementzahlbereich bei 16 Bit

Welche größte und welche kleinste vorzeichenbehaftete Zahl (*signed number*) kann man mit **16 Bit** darstellen?

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 13



2-Komplementzahlbereich bei 16 Bit

- 2-Komplement 16-Bit Zahl

$$Z_{2K} = -a_{N-1} \cdot 2^{N-1} + \sum_{i=0}^{N-2} a_i \cdot 2^i \quad a_i = \begin{cases} 0 \\ 1 \end{cases} \quad N=16 \quad \begin{matrix} \text{16 Bit} \\ \text{(signed int,} \\ \text{signed short)} \end{matrix}$$

größte
darstellbare Zahl

kleinste
darstellbare Zahl

- 16 Bit**

$$a_{15} = 0$$

$$a_{15} = 1$$

$$\begin{aligned} a_i = 1, i = 0 \dots 14 &= a_i = 0, i = 0 \dots 14 \\ 0111 \ 1111 \ 1111 \ 1111_2 &= 1000 \ 0000 \ 0000 \ 0000_2 \\ = 2^{15} - 1 = 32767_{10} &= -32768_{10} \end{aligned}$$

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 14

Größenordnungen

1 Kbyte (Kilobyte) = 1024 Byte = 2^{10} Byte

1 Mbyte (Megabyte) = 1024 Kbyte = 2^{20} Byte

1 Gbyte (Gigabyte) = 1024 Mbyte = 2^{30} Byte

1 Tbyte (Terabyte) = 1024 Gbyte = 2^{40} Byte

**Beachten Sie den Unterschied zwischen
1k (= $10^3 = 1000$) und 1K (= $2^{10} = 1024$)!**
**Beachten Sie die unterschiedliche Bedeutung von
1M (= 10^6) zu 1M (= $2^{20} = 1\,048\,576$)!**
Das gleiche Problem existiert für 1G und 1T!

19.11.2004



Binärzahlen und Hexadezimalzahlen (1)

- Nachteile der Binärdarstellung von Zeichen
 - ➡ **unübersichtlich bei großen Zahlen**
 - ➡ **benötigt viel Platz** z.B. auf einem Bildschirm.
- Übersichtlicher ist die äquivalente
 - ➡ **Hexadezimaldarstellung.**
 - Man fasst jeweils 4 Binärstellen zu einer neuen Ziffer zusammen.
 - Problem: Es werden 16 Ziffern benötigt.
 - Lösung:
Bei der Hexadezimaldarstellung verwendet man außer den Ziffern 0...9 zusätzlich die Ziffern
➡ **A, B, C, D, E, F**

19.11.2004



Binärzahlen und Hexadezimalzahlen (2)

Zahlsystem	dezimal	binär (dual)	hexadezimal
Basis	10	2	16
Ziffern	1,2,3,4,5,6,7,8,9,0	1, 0	1,2,3,4,5,6,7,8,9,A,B,C,D,E,F, 0
	0	0	0
	1	1	1
	2	10	2
	3	11	3
	4	100	4
	5	101	5
	6	110	6
	7	111	7
	8	1000	8
	9	1001	9
	10	1010	A
	11	1011	B
	12	1100	C
	13	1101	D
	14	1110	E
	15	1111	F
	16	10000	10
	17	10001	11
	18	10010	12
	19	10011	13
	20	10100	14
	21	10101	15
	22	10110	16
	:	:	:

- Zahleingabe in C:

Hexadezimalzahl:

→ 0x901..

Achtung:

0723... = Oktalzahl ! →

→ In C ist keine direkte Binärzahleingabe möglich.

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 17

19.11.2004



Binärzahlen und Hexadezimalzahlen

Wie lautet die Binärdarstellung und die Dezimaldarstellung von 0xaffe ?

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 18

19.11.2004

✓ Binärzahlen und Hexadezimalzahlen (1)

• **0xaffe** = C-Darstellung der Hexadezimalzahl **AFFE₁₆**

• Binärdarstellung:

$$A_{16} = 10_{10} = 1010_2$$

$$F_{16} = 15_{10} = 1111_2$$

$$E_{16} = 14_{10} = 1110_2$$

$$\Rightarrow 0xaffe = AF FE_{16} = 1010 \ 1111 \ 1111 \ 1110_2 \Leftarrow$$

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 19

✓ Binärzahlen und Hexadezimalzahlen (2)

• **0xaffe** = C-Darstellung der Hexadezimalzahl **AFFE₁₆**

$$Z_p = \sum_{i=0}^{N-1} a_i \cdot 16^i \quad a_i = \begin{cases} 0_{10} & (0_{16}) \\ 1_{10} & (1_{16}) \\ \vdots & \\ 15_{10} & (F_{16}) \end{cases} \quad N = 4$$

• Dezimaldarstellung:

$$A_{16} = 10_{10} \quad F_{16} = 15_{10} \quad E_{16} = 14_{10}$$

$$\begin{aligned} AF FE_{16} &= 10 \cdot 16^3 + 15 \cdot 16^2 + 15 \cdot 16^1 + 14 \cdot 16^0 \\ &= 40\,960 + 3840 + 240 + 14 \end{aligned}$$

$$\Rightarrow 0xaffe = AF FE_{16} = 45\,054_{10} \Leftarrow$$

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 20

Darstellung von Zahlen, Daten und Befehlen

- Ein Computer verwendet intern ausschließlich die **binäre Darstellung** von Zahlen, Daten und Befehlen.
- Die Zuordnung von Zeichen eines Zeichenvorrats zu denen eines anderen nennt man
 **Code**.
- Um Buchstaben oder Ziffern darzustellen verwendet man häufig den
 **ASCII-Code** (American Standard Code for Information Interchange).
- Binärdaten kann man nicht unmittelbar ansehen, ob sie Zahlen, Buchstaben oder Befehle darstellen.

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 21

Amerikanischer / deutscher ASCII-Code

MS- Stelle →	Hexadezimal	0	1	2	3	4	5	6	7
LS- Stelle ↓									
Hexa- dezimal	Binär → ↓ B ₃ B ₂ B ₁ B ₀	B ₆ B ₅ B ₄							
		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	0 0 0 0	NUL	DLE	SP	0	@	P	`	p
1	0 0 0 1	SOH	DC1	!	1	A	Q	a	q
2	0 0 1 0	STX	DC2	"	2	B	R	b	r
3	0 0 1 1	ETX	DC3	#	3	C	S	c	s
4	0 1 0 0	EOT	DC4	\$	4	D	T	d	t
5	0 1 0 1	ENQ	NAK	%	5	E	U	e	u
6	0 1 1 0	ACK	SYN	&	6	F	V	f	v
7	0 1 1 1	BEL	ETB	'	7	G	W	g	w
8	1 0 0 0	BS	CAN	(	8	H	X	h	x
9	1 0 0 1	HT	EM	)	9	I	Y	i	y
A	1 0 1 0	LF	SUB	*	:	J	Z	j	z
B	1 0 1 1	VT	ESC	+	;	K	[	Ä	{
C	1 1 0 0	FF	FS	,	<	L	\	Ö	
D	1 1 0 1	CR	GS	-	=	M	]	Ü	}
E	1 1 1 0	SO	RS	.	>	N	^	n	~
F	1 1 1 1	SI	US	/	?	O	_	o	DEL

Beispiel: "FH" = 100 0110₂ 100 1000₂ = 46₁₆ 48₁₆

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 22

MS DOS-Zeichensatz (1)

- Der Zeichensatz von **Microsoft DOS (MS DOS)** unterscheidet sich vom Windows-Zeichensatz.

Dies gilt für die Darstellung der **deutschen Umlaute** und **einiger Sonderzeichen**.

Dadurch treten bei der Entwicklung von Konsolprogrammen mit **MSVC** fehlerhafte Zeichen bei der Ausgabe von Text auf.

- Wir wollen deshalb in unseren Programmen **deutsche Umlaute** und **andere Sonderzeichen** bei der Ausgabe von Text vermeiden.

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 23

MS DOS-Zeichensatz

- **Microsoft DOS (MS DOS)**

benutzt je nach
eingestellter
Codepage einen
abweichenden
Zeichensatz:
Codepage 850
(deutsch, links)
bzw. Codepage
437 (englisch,
rechts)

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-		☐	☐	♥	♦	♣	♠	·	☐	◊	☐	♣	♠	♠	♠	♠
1-	▶	◀	↑		¶	§	—	↑	↓	→	←	↶	↷	▲	▼	
2-		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3-	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4-	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5-	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6-	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7-	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☐
8-	Ç	ù	é	â	ä	à	ç	ê	ë	è	ï	í	ì	Å	Ä	
9-	É	æ	Æ	ô	õ	ö	û	ü	ý	ÿ	Ö	Ø	ø	¥	×	ƒ
A-	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	™	½	¼	¿	«	»
B-	☐	☐	☐		†	‡	§	¶	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
C-	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
D-	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
E-	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
F-	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘

19.11.2004

Hochschule für angewandte Wissenschaften Hamburg
Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagard



Programmieren in ANSI C:
K1_C-Grundlagen.ppt, Seite 24



ASCII-Code

Welche ASCII-Zeichen stellt die
Hexadezimalzahlenfolge

45_{16} 31_{16} 42_{16} $2F_{16}$ 50_{16} 52_{16} 31_{16}

dar?

Hochschule für angewandte Wissenschaften Hamburg

Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:

K1_C-Grundlagen.ppt, Seite 25

19.11.2004



ASCII-Code

- Hexadezimalzahlenfolge:

45_{16} 31_{16} 42_{16} $2F_{16}$ 50_{16} 52_{16} 31_{16}

➡ ASCII-Zeichenfolge: "E1B/PR1"

Hochschule für angewandte Wissenschaften Hamburg

Prof. Dr.-Ing. U. Vogelsang, Prof. Dr. U. Sauvagerd



Programmieren in ANSI C:

K1_C-Grundlagen.ppt, Seite 26

19.11.2004