

Anhang

Register des Kommunikationsbausteins 8250

Tabelle: Registerübersicht des Bausteins 8250

DLAB	A2	A1	A0	Code	Register
0	0	0	0	RBR	Empfänger Halte-Register (nur lesen)
0	0	0	0	THR	Sender Halte-Register (nur schreiben)
0	0	0	1	IER	Interrupt Enable Register
X	0	1	0	IIR	Interrupt Identifikations Register (nur lesen)
X	0	1	1	LCR	Leitungs Control Register
X	1	0	0	MCR	Modem Control Register
X	1	0	1	LSR	Leitungs Status Register
X	1	1	0	MSR	Modem Status Register,
X	1	1	1	SCR	Scratch Register, 8 Bit Schreib-Lese-Register
1	0	0	0	DLL	Baudratengenerator, Teilerfaktor LSB
1	0	0	1	DLM	Baudratengenerator, Teilerfaktor MSB

Tabelle: Empfangsregister

7	6	5	4	3	2	1	0	Empfangsregister (nur Lesen) BaseAdr+0
x	x	x	x	x	x	x	x	8 Bit Datenwort

Tabelle: Sendehalterregister

7	6	5	4	3	2	1	0	Sendehalterregister (nur Schreiben) BaseAdr+0
x	x	x	x	x	x	x	x	8 Bit Datenwort

Tabelle: Interrupt-Enable Register

7	6	5	4	3	2	1	0	Interrupt Enable Register BaseAdr+1
0	0	0	0	x	x	x	1	Ermöglicht Interrupt bei ankommenden Daten
0	0	0	0	x	x	1	x	Ermöglicht einen Interrupt bei leeren Sendehalteregister
0	0	0	0	x	1	x	x	Ermöglicht Interrupt bei Leitungsstatusänderungen
	0	0	0	1	x	x	x	Ermöglicht Interrupt bei Änderungen des Modemstatus.

Tabelle: Interrupt Identifikations Register

7	6	5	4	3	2	1	0	Interrupt Identifikationsregister BaseAdr+2
0	0	0	0	0	X	X	1	kein Interrupt
0	0	0	0	0	1	1	0	Empfänger Status Interrupt durch Overrun-Fehler, Parity-Fehler, Framing-Fehler oder Interruptabbruch. - Interrupt durch lesen von LSR beenden
0	0	0	0	0	1	0	0	Interrupt durch neues Datenwort - RBR, und damit Datenwort lesen
0	0	0	0	0	0	1	0	Interrupt durch freies Sende-Halte-Register - neues Byte durch schreiben in das THR versenden
0	0	0	0	0	0	0	0	Modem Status Interrupt durch \CTS, \DSR, \RI oder \DCD - Interruptquelle durch lesen des MSR ermitteln.

Tabelle: Leitungskontroll Register

7	6	5	4	3	2	1	0	Leitungs Kontroll Register BaseAdr+3
x	x	x	x	x	x	0	0	Wortlänge 5 Bit
x	x	x	x	x	x	0	1	Wortlänge 6 Bit
x	x	x	x	x	x	1	0	Wortlänge 7 Bit
x	x	x	x	x	x	1	1	Wortlänge 8 Bit
x	x	x	x	x	0	x	x	Es wird immer ein Stopbit erzeugt
x	x	x	x	x	1	x	x	Bei 5 Datenbits werden 1½ Stopbits erzeugt, sonst 2 Stopbits
x	x	x	x	0	x	x	x	Kein Paritäts-Bit
x	x	x	0	1	x	x	x	ungerade Parität
x	x	x	1	1	x	x	x	gerade Parität
x	x	1	x	1	x	x	x	Die Parität der vorhergehenden Paritätseinstellung wird invertiert
x	1	x	x	x	x	x	x	Unterbricht die Sendeprozedur wenn logisch 1 gesetzt.
0	x	x	x	x	x	x	x	Bit 7 muß logisch 0 sein um RBR, THR und IER lesen oder schreiben zu können.
1	x	x	x	x	x	x	x	Bit 7 muß 1 sein um die Teilerregister des Baudratengenerators ansprechen zu können.

Tabelle: Modemkontroll Register

7	6	5	4	3	2	1	0	Modem Kontroll Register BaseAdr+4
0	0	0	x	x	x	x	0	\DTR Ausgang ist HIGH
0	0	0	x	x	x	x	1	\DTR Ausgang ist LOW
0	0	0	x	x	x	0	x	\RTS Ausgang ist HIGH
0	0	0	x	x	x	1	x	\RTS Ausgang ist LOW
0	0	0	x	x	0	x	x	\IOUT1 Ausgang ist HIGH
0	0	0	x	x	1	x	x	\IOUT1 Ausgang ist LOW
0	0	0	x	0	x	x	x	\IOUT2 Ausgang ist HIGH
0	0	0	x	1	x	x	x	\IOUT2 Ausgang ist LOW
0	0	0	0	x	x	x	x	LOOP ist ermöglicht
0	0	0	1	x	x	x	x	LOOP ist nicht möglich

Tabelle: Leitungsstatus Register

7	6	5	4	3	2	1	0	Leitungs Status Register BaseAdr+5
0	x	x	x	x	x	x	1	DR / ist HIGH wenn ein Byte in den Empfangsbuffer geschrieben wurde
0	x	x	x	x	x	1	x	OE / Überlauffehler, zeigt an, daß noch nicht gelesene Daten überschrieben wurden.
0	x	x	x	x	1	x	x	PE / wird HIGH wenn ein Paritätsfehler erkannt wurde
0	x	x	x	1	x	x	x	FE / Zeigt bei HIGH an, daß kein gültiges Stopbit erkannt wurde
0	x	x	1	x	x	x	x	BI / Break Interrupt wird High wenn länger als eine Ein-Wort-Übertragung der Dateneingang auf logisch 0 gehalten wurde.
0	x	1	x	x	x	x	x	THRE / Baustein ist bereit neue Daten zu senden
0	1	x	x	x	x	x	x	TEMT / wird HIGH wenn sowohl THR als auch TSR leer sind.

Tabelle: Modem Status Register

7	6	5	4	3	2	1	0	Modem Status Register BaseAdr+6
x	x	x	x	x	x	x	1	DCTS / zeigt an, daß ICTS seinen Status gewechselt hat
x	x	x	x	x	x	1	x	DDSR / zeigt an, daß IDSR seinen Status gewechselt hat
x	x	x	x	x	1	x	x	TERI / zeigt an, daß IRI von LOW nach HIGH gewechselt hat
x	x	x	x	1	x	x	x	DDCD / zeigt an, daß IDCD seinen Status gewechselt hat
x	x	x	1	x	x	x	x	CTS / zeigt den Status von ICTS an
x	x	1	x	x	x	x	x	DSR / zeigt den Status von IDSR an
x	1	x	x	x	x	x	x	RI / zeigt den Status von IRI an
1	x	x	x	x	x	x	x	DCD / zeigt den Status von IDCD an

Tabelle: Scratch Register

7	6	5	4	3	2	1	0	Scratch Register BaseAdr+7
x	x	x	x	x	x	x	x	8-Bit Datenwort

Tabelle: Baudraten Register

7	6	5	4	3	2	1	0	Baudraten Register BaseAdr+0/ DLAB = 1 für Low-Byte BaseAdr+1/ DLAB = 1 für High-Byte
x	x	x	x	x	x	x	x	8-Bit Wort Low-Byte oder High-Byte

ANSI-C Funktionsübersicht

Quelle: /BOR 91/, /KER 88/ und /MAG 92/

Tabelle: ANSI-C Funktionsübersicht

Funktions- name	PO- SIX	UNIX V7	UNIX S.V 3	NeXT GNU- C	MI- NIX V.1.5	Cohe- -rent V.3.2	DOS MSC 6.0	DOS BC 3.1	Atari TC 2.0
abort	X	X	X	X	X	X	X	X	X
abs	X	X	X	X	X	X	X	X	X
acos	X	X	X	X		X	X	X	X
asctime	X	X	X	X	X	X	X	X	X
asin	X	X	X	X		X	X	X	X
assert	X	X	X	X	X	X	X	X	X
atan	X	X	X	X		X	X	X	X
atan2	X	X	X	X		X	X	X	X
atexit							X	X	X
atof	X	X	X	X		X	X	X	X
atoi	X	X	X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S.V 3	NeXT GNU- C	MINIX V.1.5	Cohe- -rent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
bsearch	X				X	X	X	X	X
	POSIX	UNIX V7	UNIX S.V 3	NeXT GNU- C	MINIX V.1.5	Cohe- -rent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
calloc	X	X	X	X	X	X	X	X	X
ceil	X	X	X	X		X	X	X	X
clearerr	X	X	X	X		X	X	X	X
clock			X				X	X	X
cos	X	X	X	X	X	X	X	X	X
cosh	X	X	X	X		X	X	X	X
ctime	X	X	X	X	X	X	X	X	
	POSIX	UNIX V7	UNIX S.V 3	NeXT GNU- C	MINIX V.1.5	Cohe- -rent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
difftime							X	X	X
div							X	X	X
	POSIX	UNIX V7	UNIX S.V 3	NeXT GNU- C	MINIX V.1.5	Cohe- -rent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
exit	X	X	X	X	X	X	X	X	X

exp	X	X	X	X		X	X	X	X
	POSIX	UNIX V7	UNIX S.V.3	NeXT GNU-C	MINIX V.1.5	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
fabs	X	X	X	X		X	X	X	X
fclose	X	X	X	X	X	X	X	X	X
feof	X	X	X	X		X	X	X	X
ferror	X	X	X	X		X	X	X	X
fflush	X	X	X	X	X	X	X	X	X
fgetc	X	X	X	X	X	X	X	X	X
fgetpos							X	X	X
fgets	X	X	X	X	X	X	X	X	X
floor	X	X	X	X		X	X	X	X
fmod	X		X				X	X	X
fopen	X	X	X	X	X	X	X	X	X
fprintf	X	X	X	X	X	X	X	X	X
fputc	X	X	X	X	X	X	X	X	X
fread	X	X	X	X	X	X	X	X	X
free	X	X	X	X	X	X	X	X	X
freopen	X	X	X	X	X	X	X	X	X
frexp	X	X	X	X	X	X	X	X	X
fscanf	X	X	X	X	X	X	X	X	X
fseek	X	X	X	X	X	X	X	X	X
fsetpos						X		X	X
ftell	X	X	X	X	X	X	X	X	X
fwrite	X	X	X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S.V.3	NeXT GNU-C	MINIX V.1.5	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
getc	X	X	X	X	X	X	X	X	X
getchar	X	X	X	X	X	X	X	X	X
getenv	X	X	X	X	X	X	X	X	X
gets	X	X	X	X	X	X	X	X	X
gmtime	X	X	X	X	X	X	X	X	
	POSIX	UNIX V7	UNIX S.V.3	NeXT GNU-C	MINIX V.1.5	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
isalnum	X	X	X	X	X	X	X	X	X
isalpha	X	X	X	X	X	X	X	X	X
iscntrl	X	X	X	X	X	X	X	X	X
isdigit	X	X	X	X	X	X	X	X	X
isgraph	X		X		X		X	X	X
islower	X	X	X	X	X	X	X	X	X

isprint	X	X	X	X	X	X	X	X	X
ispunct	X	X	X	X	X	X	X	X	X
isspace	X	X	X	X	X	X	X	X	X
isupper	X	X	X	X	X	X	X	X	X
isxdigit	X		X	X	X		X	X	X
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
labs							X	X	X
ldexp	X	X	X	X		X	X	X	X
ldiv						X	X	X	X
localeconv							X	X	
localtime	X	X	X	X	X	X	X	X	X
log	X	X	X	X		X	X	X	X
log10	X		X	X		X	X	X	X
longjmp	X	X	X	X	X	X	X	X	X
lseek	X	X	X	X	X	X	X	X	X
malloc	X	X	X	X	X	X	X	X	X
memchr			X		X	X	X	X	X
memcmp			X		X	X	X	X	X
memcpy			X		X	X	X	X	X
memmove					X	X	X	X	X
memset			X		X	X	X	X	X
mktime					X		X	X	X
modf	X	X	X		X	X	X	X	X
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
onexit							X		
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
perror	X	X	X	X	X	X	X	X	X
pow	X	X	X	X		X	X	X	X
printf	X	X	X	X	X	X	X	X	X
putc	X	X	X	X	X	X	X	X	X
putchar	X	X	X	X	X	X	X	X	X
puts	X	X	X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
qsort	X	X	X	X	X	X	X	X	X

	POSIX	UNIX V7	UNIX S.V.3	NeXT GNU- C	MINIX V.1.5	Cohe- rent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
rand	X	X	X	X	X	X	X	X	X
realloc	X	X	X	X	X	X	X	X	X
remove	X			X			X	X	X
rename	X			X	X	X	X	X	X
rewind	X	X	X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S.V.3	NeXT GNU- C	MINIX V.1.5	Cohe- rent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
scanf	X	X	X	X	X	X	X	X	X
setbuf	X	X	X	X	X	X	X	X	X
setjmp	X	X	X	X	X	X	X	X	X
setlocale							X	X	
setvbuf			X	X			X	X	X
signal	X	X	X	X	X	X	X	X	X
sin	X	X	X	X	X	X	X	X	X
sinh	X	X	X	X		X	X	X	X
sprintf	X	X	X	X	X	X	X	X	X
sqrt	X	X	X	X		X	X	X	X
srand	X	X	X	X	X	X	X	X	X
sscanf	X	X	X	X	X	X	X	X	X
strcat	X	X	X	X	X	X	X	X	X
strchr	X		X		X	X	X	X	X
strcmp	X	X	X	X	X	X	X	X	X
strcoll					X	X	X	X	
strcpy	X	X	X	X	X	X	X	X	X
strcspn	X		X		X	X	X	X	X
strerror				X	X	X	X	X	X
strlen	X	X	X	X	X	X	X	X	X
strncat	X	X	X	X	X	X	X	X	X
strncmp	X	X	X	X	X	X	X	X	X
strncpy	X	X	X	X	X	X	X	X	X
strpbrk	X		X		X	X	X	X	X
strrchr	X		X		X	X	X	X	X
strspn	X		X		X	X	X	X	X
strstr	X				X	X	X	X	X
strtod			X				X	X	X
strtok	X		X		X	X	X	X	X
strtol			X		X		X	X	X
strtoul					X		X	X	X
strxfrm					X	X		X	

system	X	X	X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
tan	X	X	X	X		X	X	X	X
tanh	X	X	X	X		X	X	X	X
time	X	X	X	X	X	X	X	X	X
tmpfile	X		X	X			X	X	X
tmpnam	X		X	X	X	X	X	X	
tolower	X		X	X	X	X	X	X	X
toupper	X		X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
ungetc	X	X	X	X	X	X	X	X	X
	POSIX	UNIX V7	UNIX S V 3	NeXT GNU-C	MINIX V.15	Coherent V.3.2	Dos MSC 6.0	Dos BC 3.1	Atari TC 2.0
vfprintf			X	X	X		X	X	X
vprintf			X	X			X	X	X
vsprintf			X	X	X		X	X	X

Erweiterte ASCII-Zeichentabelle (IBM-PC)

Tabelle: Erweiterter ASCII-Zeichensatz Teil 1

DEC	HEX	Char	DEC	HEX	Char	DEC	HEX	Char
0	0		43	2B	+	86	56	V
1	1		44	2C	,	87	57	W
2	2		45	2D	-	88	58	X
3	3	♥	46	2E	.	89	59	Y
4	4	♦	47	2F	/	90	5A	Z
5	5	♣	48	30	0	91	5B	[
6	6	♠	49	31	1	92	5C	\
7	7	•	50	32	2	93	5D]
8	8		51	33	3	94	5E	^
9	9		52	34	4	95	5F	_
10	A		53	35	5	96	60	'
11	B		54	36	6	97	61	a
12	C		55	37	7	98	62	b
13	D		56	38	8	99	63	c
14	E		57	39	9	100	64	d
15	F		58	3A	:	101	65	e
16	10		59	3B	;	102	66	f
17	11		60	3C	<	103	67	g
18	12		61	3D	=	104	68	h
19	13		62	3E	>	105	69	i
20	14		63	3F	?	106	6A	j
21	15		64	40	@	107	6B	k
22	16		65	41	A	108	6C	l
23	17		66	42	B	109	6D	m
24	18	↑	67	43	C	110	6E	n
25	19	↓	68	44	D	111	6F	o
26	1A	→	69	45	E	112	70	p
27	1B	←	70	46	F	113	71	q
28	1C		71	47	G	114	72	r
29	1D		72	48	H	115	73	s
30	1E		73	49	I	116	74	t
31	1F		74	4A	J	117	75	u
32	20		75	4B	K	118	76	v
33	21	!	76	4C	L	119	77	w

34	22	"	77	4D	M	120	78	x
35	23	#	78	4E	N	121	79	y
36	24	\$	79	4F	O	122	7A	z
37	25	%	80	50	P	123	7B	{
38	26	&	81	51	Q	124	7C	
39	27	'	82	52	R	125	7D	}
40	28	(83	53	S	126	7E	~
41	29)	84	54	T	127	7F	
42	2A	*	85	55	U			

Tabelle: Erweiterter ASCII-Zeichensatz Teil 2

DEC	HEX	Char	DEC	HEX	Char	DEC	HEX	Char
128	80	Ç	171	AB	½	214	D6	ö
129	81	ü	172	AC	¼	215	D7	×
130	82	è	173	AD	¡	216	D8	ø
131	83	á	174	AE	«	217	D9	ù
132	84	ä	175	AF	»	218	DA	ú
133	85	à	176	B0	°	219	DB	û
134	86	å	177	B1	±	220	DC	ü
135	87	ç	178	B2	²	221	DD	ý
136	88	ê	179	B3	³	222	DE	þ
137	89	ë	180	B4	´	223	DF	ß
138	8A	è	181	B5	µ	224	E0	α
139	8B	ï	182	B6	¶	225	E1	β
140	8C	ì	183	B7	·	226	E2	Γ
141	8D	ì	184	B8	¸	227	E3	π
142	8E	Ä	185	B9	¹	228	E4	Σ
143	8F	Å	186	BA	º	229	E5	σ
144	90	É	187	BB	»	230	E6	μ
145	91	œ	188	BC	¼	231	E7	τ
146	92	Æ	189	BD	½	232	E8	φ
147	93	ô	190	BE	¾	233	E9	Φ
148	94	ö	191	BF	¿	234	EA	Ω
149	95	ò	192	C0	À	235	EB	δ
150	96	ù	193	C1	Á	236	EC	∞
151	97	ú	194	C2	Â	237	ED	∅
152	98	ÿ	195	C3	Ã	238	EE	ε
153	99	Û	196	C4	Ä	239	EF	∩

154	9A	Ü	197	C5	Ä	240	F0	≡
155	9B	ø	198	C6	Æ	241	F1	±
156	9C	£	199	C7	Ç	242	F2	≥
157	9D	¥	200	C8	È	243	F3	≤
158	9E		201	C9	É	244	F4	∫
159	9F	f	202	CA	Ê	245	F5	J
160	A0	á	203	CB	Ë	246	F6	÷
161	A1	í	204	CC	Ï	247	F7	≈
162	A2	ó	205	CD	Í	248	F8	°
163	A3	ú	206	CE	Î	249	F9	•
164	A4	ñ	207	CF	Ï	250	FA	·
165	A5	Ñ	208	D0	Ð	251	FB	√
166	A6		209	D1	Ñ	252	FC	
167	A7		210	D2	Ö	253	FD	²
168	A8	ı	211	D3	Ó	254	FE	
169	A9		212	D4	Ô	255	FF	
170	AA	ı	213	D5	Õ			

Lösungen zu den Übungsaufgaben

Übungsaufgabe 1:

```
#include <stdio.h>

int leseZahl();
void zeichneBalken(int);

int leseZahl()
{ int z;
  scanf("%d",&z);
  return z;
}

void zeichneBalken(int z)
{ int i;
  printf("\r\n");
  for(i=0;i<z;i++)
    { printf("*");
    }
  printf(" Wert:%d\r\n",z);
}

main()
{ int z1,z2,z3,z4;
  z1=leseZahl();
  z2=leseZahl();
  z3=leseZahl();
  z4=leseZahl();
  zeichneBalken(z1);
  zeichneBalken(z2);
  zeichneBalken(z3);
  zeichneBalken(z4);
  return 0;
}
```

Übungsaufgabe 2

Der Wertebereich des Typ `int` geht von -32767 bis 32766. Damit liegt der Wert 40000 außerhalb dieses Bereichs. Der Datentyp `int` ist also nicht ausreichend.

Den geforderten Wertebereich decken z.B. die ganzzahligen Typen `unsigned int` oder `long` ab.

Übungsaufgabe 3

```
#include <stdio.h>

int leseZahl();
void zeichneBalken(int);
void zeichneMittelWert(int);
int mittelWert( int, int, int, int);

int leseZahl()
{ int z;
  scanf("%d",&z);
  return z;
}

void zeichneBalken(int z)
{ int i;
  printf("\r\n");
  for(i=0;i<z;i++)
  { printf("*");
  }
  printf(" Wert:%d\r\n",z);
}

void zeichneMittelWert(int z)
```

```
{ int i;
  printf("\r\n");
  for(i=0;i<z;i++)
    { printf("**");
      }
  printf(" Mittelwert:%d\r\n",z);
}

int mittelWert(int i1, int i2, int i3, int i4)
{ long mw = i1 + i2 + i3 +i4;
  return (int)(mw / 4);
}

main()
{ int z1,z2,z3,z4;
  int mittelwert;

  z1=leseZahl();
  z2=leseZahl();
  z3=leseZahl();
  z4=leseZahl();

  mittelwert=mittelWert(z1,z2,z3,z4);

  zeichneBalken(z1);
  zeichneBalken(z2);
  zeichneBalken(z3);
  zeichneBalken(z4);
  zeichneMittelWert(mittelwert);
  return 0;
}
```

Übungsaufgabe 4

```
#include <stdio.h>

int leseZahl();
void zeichneBalken(int, int);
void zeichneMittelWert(int);
int mittelWert( int, int, int, int);

int leseZahl()
{ int z;
  scanf("%d",&z);
  return z;
}

void zeichneBalken(int z, int mw)
{ int i;
  printf("\r\n");
  for(i=1;i<=z;i++)
    { if(i == mw)
      printf("M");
      else
      printf("*");
    }
  if( (i-1) <mw)
    { for( ;i<mw;i++)
      printf(" ");
      printf("M");
    }
  printf(" Wert:%d\r\n",z);
}

void zeichneMittelWert(int z)
{ int i;
  printf("\r\n");
  for(i=0;i<z;i++)
```

```
        { printf("*");
        }
        printf(" Mittelwert:%d\r\n",z);
    }

int mittelWert(int i1, int i2, int i3, int i4)
{ long mw = i1 + i2 + i3 +i4;
  return (int)(mw / 4);
}

main()
{ int z1,z2,z3,z4;
  int mittelwert;

  z1=leseZahl();
  z2=leseZahl();
  z3=leseZahl();
  z4=leseZahl();

  mittelwert=mittelWert(z1,z2,z3,z4);

  zeichneBalken(z1, mittelwert);
  zeichneBalken(z2, mittelwert);
  zeichneBalken(z3, mittelwert);
  zeichneBalken(z4, mittelwert);
  zeichneMittelWert(mittelwert); /* kann wegfallen */
  return 0;
}
```

Literaturverzeichnis

- /BOO 91/ Object-oriented Design with Applications
G. Booch
Benjamin-Cummings, Redwood City, USA, 1991
- /BOR 91/ Handbuch zu BORLANDC 3.0
Borland GmbH
Borland GmbH, München, 1991
- /COA 91/ Object-oriented Analysis
Yourdon Press, New Jersey, USA, 1991
- /ELL 90/ The annotated C++ Reference Manual
M. A. Ellis, B. Stroustrup
Addision-Wesley Publishing Company, Reading, USA, 1990
- /FIE 91/ Objekt-orientierte Programmierung in der Automatisierung
J. Fiedler, K.F. Rix, H. Zöllner
VDI-Verlag, Düsseldorf, 1991
- /KER 88/ The C Programming Language, 2. Edition
B. W. Kernighan, D. M. Ritchie
Prentice Hall, Englewood Cliffs, USA, 1988
- /MAG 92/ Magazin für Computertechnik
Ausgabe 1/92, Kartei
Heise Verlag, Hannover, 1992
- /WAR 91/ Strukturierte Systemanalyse von Echtzeit-Systemen
Paul T. Ward, Stephen J. Mellor
Carl Hanser Verlag, München, 1991

Index

#

#define, 27

#else, 30

#endif, 30

#ifdef, 30

A

A/D-Umsetzer, 170

Abfragebetrieb, 230

Abstrakte Daten, 250

Abtastfrequenz, 216

Abtastreglung, 221

Adreßberechnung, 82

Adreßoperator, 73

Analyse, 18; 236

Anker, 202

ANSI-Norm, 1

Anweisung, 14

Array, 53

ASCII-Codierung, 9

Asynchrone Übertragung, 163

B

Basistypen, 6

Bedienerführung, 225

bedingte Übersetzung, 30

Bereichsüberprüfung, 83

Betriebssystem, 111

Binder, 2

Bitfeld, 68

Bitoperationen, 35

Block, 14

break, 41

bsearch(), 133

Buchstabe, 112

Buchstaben, 8

C

calloc(), 106

Casting, 15

const, 11

D

D-Regler, 215

Datei, 128

Dateiverwaltung, 118

Dateizeiger, 127

Datenbreite, 7

Datenspeicher, 179

Datenstack, 180

Datenstruktur, 196

Datentyp, 72

Datentypen, 6
DDC, 218
Deklaration, 12
Dekrement, 36
Dereferenzoperator, 73; 86
Design, 18
Digitaler Regler, 216
Digitalregler, 212
disable(), 142
do...while(), 43
double, 10
Drucker, 163
dynamische Speicheranforderung, 108
dynamische Variable, 102

E

E/A-Baustein, 138
Echtzeitsystem, 236
Echtzeituhr, 143
Einbinden von Dateien, 29
Eingabefeld, 88
enable(), 142
enum, 63
Ereigniszähler, 143
exit(), 132

F

Far Pointer, 193
Fehlerquelle, 76
Fehlerquellen, 237

Feld, 53; 54
Feldzugriffsoperators, 87
fgets(), 128
FIFO, 190
FIFO-Datenbuffer, 182
FILE, 126
Fließkommazahlen, 10
float, 10
fopen(), 126
FOPEN_MAX, 128
for(), 45
Formatbezeichner, 120
Formatzeichenkette, 119
fprintf(), 126
Fragmentierung, 108
free(), 105
fscanf(), 126
Funktion, 4; 13
Funktionsaufruf, 14; 17
Funktionsprototyp, 13

G

getchar(), 43
getkey(), 228
getvect(), 142
Gleitpunktzahlen, 10
Grenzfrequenz, 171
Gültigkeit, 15

H

Headerdatei, 109; 111; 243

Heap, 102

I

I/O-Port, 139

if()...else, 37

Impulsgenerator, 143

include, 29

Index, 53

Initialisierung, 12; 166

Inkonsistenz, 249

Inkrement, 36

inport(), 138

int, 7

Integralregler, 214

Interrupt, 140; 168

interrupt, 141

Interruptroutine, 224

K

kbhit(), 228

Kommentar, 12

Kommunikationsadapter, 163

Konstante, 11

Kontrollfluß, 37

Kopieren, 114

L

Laufzeitoptimierung, 239

Lebensdauer, 15; 16; 57

Linker, 2

Liste, 198

LValue, 34

M

main(), 5

Makro, 26

Makrodefinition, 28

Makroname, 28

malloc(), 103

Maschinendatensatz, 86

Maschinensprache, 241

memcpy(), 85; 97

memmove(), 115

memset(), 115

Meßreihe, 121; 129

Mikrokontroller, 109; 133

Mikroprozessor, 67

Mittelwert, 210

Modul, 248; 254

Moduszeichenkette, 127

Multibytebuchstaben, 9

N

Name, 3

Null Pointer Assignment, 75

NULL-Zeiger, 103

Nullzeiger, 74; 127

O

Offset, 139

On-Line Hilfe, 1

Operator, 32

outport(), 138

P

P-Regler, 213

parallele Schnittstelle, 150

Parameter, 13

Peripherie, 150

PID-Verhalten, 212

Polling, 230

Portbaustein, 152

Präprozessor, 26

printf(), 118

Priorität, 32

Proportionalbereich, 213

Prozeß, 171

Prozeßereignis, 140

Q

qsort(), 133

Qualifizierer, 7

R

realloc(), 107

Reentranz, 102

Referenz, 72

Regelabweichung, 214

Registersatz, 166

Regler, 212

Rekursion, 102

return, 15

Ringbuffer, 186

RS-232-C, 164

Rückgabewert, 13

S

Sample and Hold, 171

scanf(), 122

Schlüsselworte, 3

Schnittstellen, 138

Schrittmotor, 143

Segment, 139

Seiteneffekte, 102

serielle Schnittstelle, 163

setvect(), 142

size_t, 103

sizeof(), 52

Softwareentwicklung, 236

Speicher, 188

Speichermodell, 193

Speicherverwaltung, 104

Stack, 78; 238

Standardbibliothek, 109

Stapelspeicher, 180

static, 16; 57; 102

Statistik, 210

stddef.h, 110

strcat(), 116

strcpy(), 97

stricmp(), 132

String, 113

strlen(), 94; 117

strncpy(), 115

struct, 48

Struktur, 48

Suchen, 114

switch(), 40

T

Tastaturabfrage, 229

Tastaturereignis, 227

Timer, 143

tolower(), 113

toupper(), 113

typedef, 61

Typkonvertierung, 15

Typprüfung, 124

Typzeichen, 120

U

Umlaute, 10

union, 65

Unterbrechung, 12

unwahr, 35

V

V.24, 163

Verbunddatentyp, 48; 65

Vergleich, 114

Verzweigung, 37

void, 6; 84

volatile, 12

Vorrang, 32

W

Wandler, 170

Wertebereich, 111

while(), 42

Z

Zahlenkonstante, 12

Zählschleife, 45

Zeichenkette, 92; 113

Zeichenkettenkonstante, 93

Zeichensatz, 10

Zeiger, 72

Zeiger auf Funktionen, 87

Zeiger auf Zeiger, 85

Zeigerarithmetik, 81

Ziel, 114

Zuweisung, 76; 85; 95

Zykluszeit, 172; 224