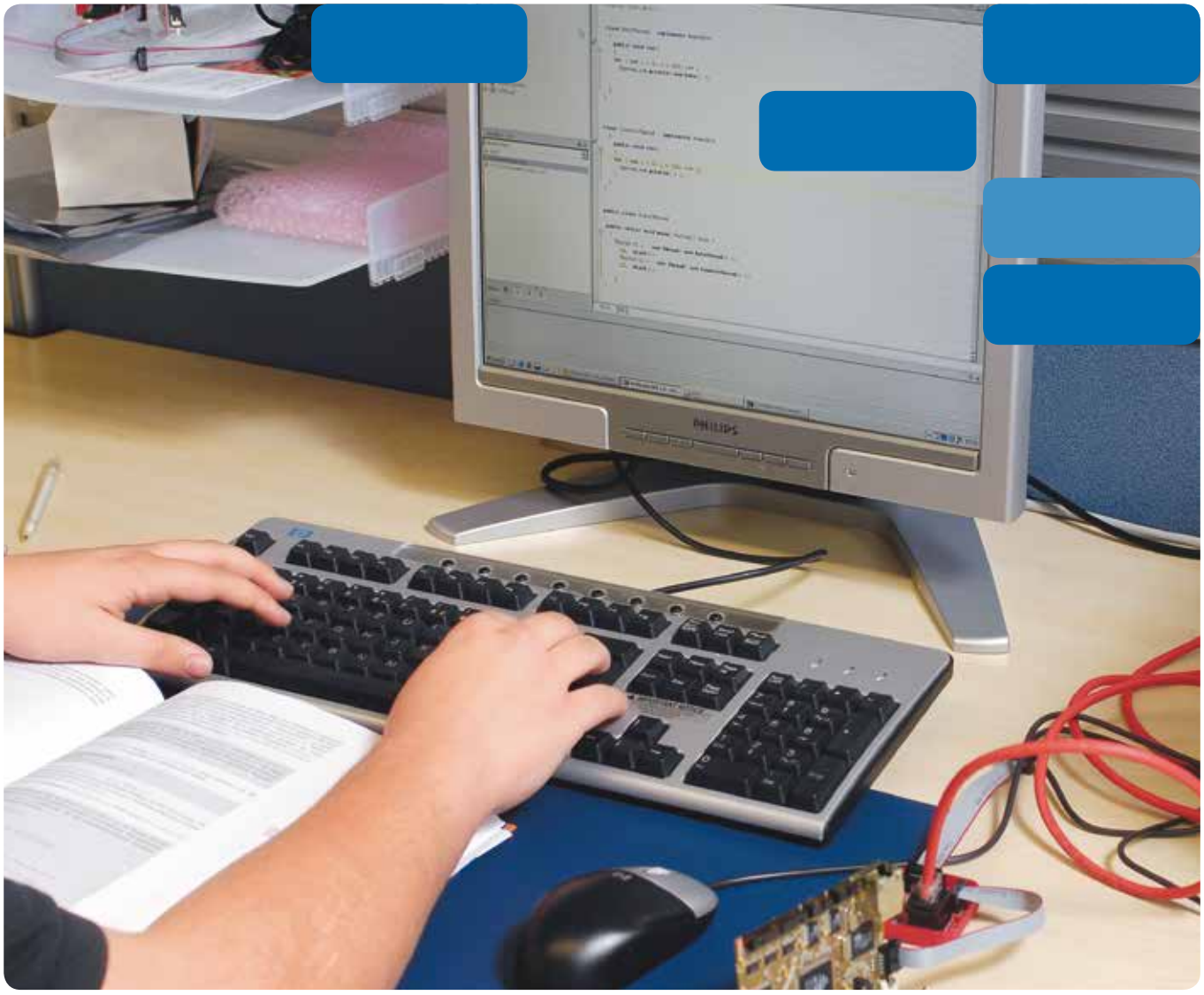
Mikrocontrollertechnik

ETBK 3



Inhalt

3 ETB3 Mikrocontrollertechnik
3.1 Verständnis Mikrocontroller 7
3.2 Analyse und Entwurf 105
3.3 Realisierung / Codierung 133
3.4 Softwaretest 142
3.5 Dokumentation 149
3.6 Präsentation 167

EDITIONS SWISSMEM

SWISSMEM

Herausgeberin: Edition Swissmem
5. Auflage 2018

Bezugsquelle:
Swissmem Berufsbildung
Brühlbergstrasse 4
8400 Winterthur

Telefon Vertrieb 052 260 55 55
Fax Vertrieb 052 260 55 59

www.swissmem-berufsbildung.ch
vertrieb.berufsbildung@swissmem.ch

Copyright Text, Zeichnung und Ausstattung:
© by Swissmem, Zürich

Alle Rechte vorbehalten. Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Verwertung in andern als den gesetzlich zugelassenen Fällen bedarf der vorherigen schriftlichen Einwilligung des Herausgebers.

SWISSMEM

An der Ausarbeitung dieses Lehrgangs waren beteiligt:

Baltisberger Martin, Müller Martini Druckverarbeitungssysteme, Zofingen
Bays Didier, CFPT, Petit-Lancy
Habegger Olivier (Projektverantwortung), Swissmem Berufsbildung, Winterthur
Inhelder Jürg, Projektleitung für die 5. Auflage 2018, Winterthur
Manni Daniel, Bobst S.A., Mex
Schmied Toni, CPLN, Neuchâtel
Simon Liechti, ICT Berufsbildungscenter, Bern

Mai 2018 Swissmem Berufsbildung

Zeichenerklärungen, Inhaltlicher Aufbau

Zeichenerklärung



Diese Variante ist zweckmässig. Im Sinne der Optimierung des Produktes suchen wir die stärkste Lösung.



Brauchbare Lösung. Sicher sind noch bessere Varianten zu finden!



Diese Lösung ist ungeeignet. Überlegen Sie, aus welchem Grund diese Lösung nicht befriedigt und suchen Sie eine bessere Variante.



Lösen Sie diese Aufgabe mit dem geeignetsten Hilfsmittel.



Lernziele



Wichtige Hinweise



Information



Informationen im Web: www.swissmem-elearning.ch

Notieren Sie hier die zutreffenden Informationen, wie nationale oder internationale Normen, Betriebsnormen, Titel von Fachbüchern, Betriebsanleitungen usw.

Inhaltlicher Aufbau

Der Lehrgang ist nach der gleichen Struktur wie der Kompetenzen-Ressourcen-Katalog aufgebaut.

Der Ressourcenaufbau ist wie folgt gegliedert:

Aktivierung
Jede Ausbildungseinheit beginnt mit Grundsatzfragen, welche den momentanen Wissensstand erfassen.

Theorie / Übungen
Der Theorieteil beinhaltet neben der Theorie auch Fragen und/oder Übungen, welche die Lernenden lösen müssen.

Repetition
Als Abschluss des Ressourcenaufbaus sind Repetitionsfragen zu beantworten. Diese dienen der Festigung des Lernstoffs.

Inhaltsverzeichnis

3.1 Verständnis Mikrocontroller

3.1.0 Binärtechnische Grundlagen **7**
Darstellung von Informationen 8
Binäre Logik – Maskieren 15
Binäre Arithmetik 18
3.1.1 Mikrocontroller evaluieren **27**
Was ist ein Mikrocontroller 28
Aufbau eines Mikrocontrollers 29
Systemaufbau 31
Speicherbausteine 35
Ansteuerung und Adressierung 38
Programmieren eines Mikrocontrollers 42
3.1.2 Hochsprache anwenden **45**
Programmiersprachen 46
Hochsprachen 48
Einführung in die Hochsprache C 51
Operatoren 52
Variablen 53
Programmschleifen 56
Programmverzweigungen 58
Funktionen 60
Erstes EVA-Programm 63
3.1.3 Programmentwicklungswerkzeuge **65**
Integrierte Entwicklungsumgebung (IDE) 68
Assembler und Hochsprache (Compilierung) 70
Direkte Programmierung 70
Linker 71
Globale Projektabwicklung 72
Beispiel für Dateien 73
Präsentation einer IDE 76
3.1.4 Software Engineering **81**
System-Design: Vom Pflichtenheft zum Test 82
Einfaches Übungsbeispiel 84
Analyse und Entwurf 89
Gültigkeitsbereich der Daten 91
Datendeklaration 92
Dokumentation 95
3.1.5 Mikrocontrollersysteme in Betrieb nehmen **101**
Aufbau/Struktur einer Mikrocontroller-Software 101
Datenblätter interpretieren 102
Testprogramme für Mikrocontroller schreiben 103

3.2 Analyse und Entwurf

3.2.1. Informationen beschaffen **105**
Aufgaben analysieren 105
3.2.2 Hardwarekonzepte **106**
3.2.3 Graphische Darstellung erstellen **107**
Struktogramm «Nassi-Shneiderman» 108
Programmablaufplan (PAP)/Flussdiagramm 115
Struktogramm im Vergleich mit Flussdiagramm 118
Zustandsdiagramm, State-Event-Diagramm 123
Symbole 123
Zustandsautomaten entwerfen 124
Umsetzungsvorschläge für bestimmte Programmfunktionen 127

Inhaltsverzeichnis

Schalter, zustandsgesteuert 127
Schalter, flankengesteuert 128
Blinken 130
Getaktetes Programm 131

3.3 Realisierung / Codierung

3.3.1 Compiler und Debugger konfigurieren und einsetzen 134
Kommentare im C-Quellcode 134
Verbalisierung 136
Codierung in C 136

3.4 Softwaretest

3.4.1 Testprotokolle 143
Beispiele 144

3.5 Dokumentation

3.5.1 Softwaredokumentation erstellen 149
Pflichtenheft 152
Projektplanung 154
Steuerungsdiagramm 156
Zustandsdiagramm 158
Zustandstabelle 158
Struktogramm 159
Codierung 160
Programmtest 163

3.6 Präsentation

3.6.1 Software präsentieren 167

3.1.0 Binärtechnische Grundlagen

Binärtechnische Grundlagen verstehen

1) Zählen Sie die verschiedenen arithmetischen Basen auf, die in der Elektronik häufig verwendet werden:

2) In welchen Bereichen wird Binärtechnik eingesetzt?

3) Geben Sie mehrere Beispiele zur Darstellung von binären Zuständen an:

4) Geben Sie mehrere Beispiele von Datenträgern an, die in der Binärtechnologie zur Speicherung von Informationen eingesetzt werden.

5) Wie könnte man in der Elektronik eine binäre Information darstellen?

3.1.0 Binärtechnische Grundlagen

Darstellung von Informationen

Die Begriffe digital und analog

Die meisten der in der Natur vorkommenden Grössen (Helligkeit, Temperatur, Druck, Geschwindigkeit, …) sind innerhalb eines gewissen Bereichs stetig. Man bezeichnet sie als **analoge** Grössen. Sie sind mit analoger Schaltungstechnik fassbar. Einfache Helligkeitsregler, aber auch Spannungsregler in Stromversorgungsgeräten können auf diese Weise arbeiten. Analoge Grössen sind durch Digitalcomputer nur in digitalisierter Form zu verarbeiten. Sie müssen vor der Eingabe in den Computer in **digitale** Grössen umgewandelt werden.

Einfache Digitalisierungsvorgänge sind z.B. die Unterteilung der:
– Zahlengeraden in positive und negative ganze Zahlenintervalle
– Zeit in Stunden, Minuten, Sekunden
– Temperatur in Kelvin, usw.



Musik wird heute meist digital gespeichert und verkauft. Mit dem MP3-Format der Fraunhofer-Gesellschaft belegt eine Minute Musik 1,4 Megabyte.

Wählen wir die Intervalle zwischen diesen diskreten Werten fein genug, so merken wir mit unseren menschlichen Sinnen schliesslich nichts mehr vom Digitalisierungsvorgang. Allerdings steigt mit der feineren Unterteilung der Bedarf an Rechnerleistung. Zudem kann der Digitalcomputer nur binäre Werte, d.h. solche mit nur zwei erlaubten Zuständen, verarbeiten. Digitale Grössen müssen deshalb mit mehreren Binärstellen codiert werden. Einige Beispiele von notwendigen Verarbeitungsgeschwindigkeiten zeigen, dass auch moderne Mikroprozessoren bald einmal überfordert sind:

– Heizungsregelung	10 Binärwerte/s
– Manuelle Tastatureingabe	100 b/s
– Sprachübertragung	100 kb/s
– Werkzeugmaschinenpositionierung	1 Mb/s
– Farbbildverarbeitung	10 Mb/s
– HDTV-Übertragung	1,4…3,4 Mb/s
– UHDTV-Übertragung (4K)	20…30 Mb/s

Digitale oder gar binäre Systeme lassen sich in der Natur nur per Definition finden, d.h. dass wir natürlichen Vorgängen willkürlich eine «Ja/Nein»- Aussage zuordnen. Oft wird dann zwar im alltäglichen Sprachgebrauch mehr oder weniger bildhaft ein hartes Ja/Nein etwas «weniger binär» umschrieben:
– Tag oder Nacht (dämmriger Tag, stockdunkle Nacht)
– Kalt oder warm (Eiskälte, Affenhitze)
– Tot oder lebendig (mausetot, quicklebendig)

3.1.0 Binärtechnische Grundlagen

Das einzelne Bit

Informationsmengen können gemessen werden, die Masseinheit heisst Bit.
1 Bit ist die kleinste Informationsmenge (Antwort auf eine Frage, welche nur zwei Antworten zulässt – ja oder nein).

Der Begriff Bit ist die Abkürzung für «**b**inary digi**t**» (binäre Ziffer) und wird in der Praxis auch für binäre Signale und binäre Zustände verwendet.

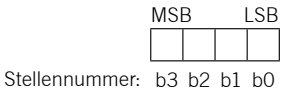
Datenworte

Mehrere Bits können organisatorisch zu einem Datenwort zusammengefasst werden. Die Bits werden nummeriert, wobei das niederwertigste Bit LSB (**l**east **s**ignificant **b**it) die Stellennummer 0 erhält und rechts dargestellt wird. Das höchstwertige Bit wird mit MSB (**m**ost **s**ignificant **b**it) bezeichnet.

Nibble

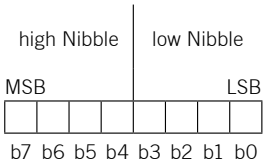
Ein Nibble besteht aus vier Bits und ist ein halbes Byte (englisch «to nibble» = anknabbern).

Andere Bezeichnungen für Nibble sind:
– Tetrade «lateinisch tetra = vier»
– Halbbyte



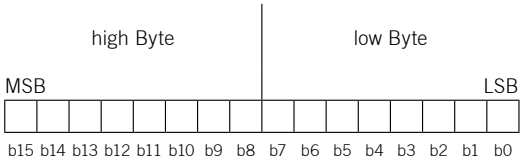
Byte

Ein Byte besteht aus acht Bit und gliedert sich in zwei Nibble zu je vier Bit.



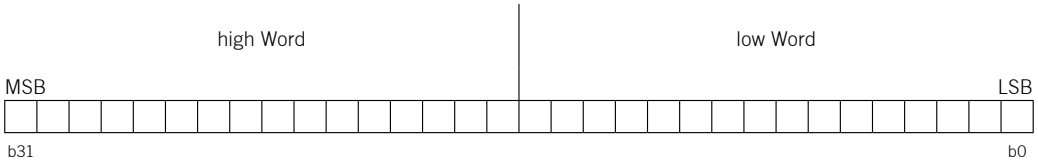
Word

Ein Word besteht aus zwei Byte.



Long Word

Viele Mikrocomputer können Datenworte von 32 Bit = 2 Word verarbeiten.



3.1.0 Binärtechnische Grundlagen

Bezeichnungen für Wertigkeiten

Folgende Bezeichnungen sind für die Wertigkeiten gebräuchlich:

– LSB (least significant bit)	niederstwertiges Bit
– Low Byte	niederwertiges Byte
– Low Word	niederwertiges Word
– MSB (most significant bit)	höchstwertiges Bit
– High Byte	höherwertiges Byte
– High Word	höherwertiges Word

Schreibweise

Sollen 8, 16 oder gar 32 Bit dargestellt werden, ist dies bei einer Abfolge von lauter Nullen und Einsen sehr unübersichtlich. Deshalb werden Datenworte oft in hexadezimaler Schreibweise notiert. Dabei wird beispielsweise aus dem binär dargestellten Byte 0011 1011 der hexadezimal notierte entsprechende Wert 3B. Mit bestimmten vor- oder nachgestellten Zeichen (z.B. 3Bh) kann bei Mikrocomputersystemen bei der Programmentwicklung zwischen verschiedenen Eingabe- und Anzeigeformen umgeschaltet werden.

Die Einführung von verschiedenen Schreibweisen für Zahlen bringt gleichzeitig die Möglichkeit der Verwechslung mit sich. Deshalb hat sich eingebürgert, dass die Schreibweise bei Zahlen mit vor- oder nachgestellten Buchstaben angegeben wird. Bei einer Zahl ohne Angabe wird die Dezimale Schreibweise angenommen.

Schreibweise	**Darstellung**
Binär	10B oder 10b oder 10_b oder %10 oder $10_{(2)}$
Dezimal	10D oder 10d oder 10_d oder 10 oder $10_{(10)}$
Hexadezimal	10H oder 10h oder 10_h oder 0x10 oder $10 oder $10_{(16)}$

Codes Einleitung

Computer und andere elektronische Digitalschaltungen können grundsätzlich nur binäre Signale verarbeiten. Grössen der realen Umwelt (analoge Signale, Druckzeichen usw.) müssen daher intern als digitale Werte in Datenworten dargestellt werden.

Die Vereinbarung, welche jedem Element einer Menge (z.B. Messwert) ein Element einer anderen Menge (z.B. Digitalwert) zuordnet, nennt man den Code.

Um ein Bitmuster im Computer interpretieren zu können, muss man den Code für das betreffende Datenwort kennen. Gleiche Bitmuster können je nach Code völlig unterschiedliche Bedeutung haben.
Bei gewichteten Codes stellt jedes einzelne Bit für sich einen bestimmten Wert dar. Bei ungewichteten Codes kann nur das ganze Datenwort zusammen interpretiert werden.

Ungewichtete Codes

Typische Beispiele sind Zeichencodes:
Wenn ein Computer Texte verarbeitet (speichern, drucken usw.), stellt er intern jedes Zeichen als Datenwort dar. Je nach Gerätehersteller, Sprache usw. kommen verschiedene Codes zur Anwendung.

Der «**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange» ist der am weitesten verbreitete Textcode. Er ist aus dem amerikanischen Telexcode hervorgegangen und arbeitet mit 7 Bit-Datenworten. Erweiterte Codes verwenden teilweise 8 Bit oder definieren zusätzliche länder- und anwendungsspezifischen Zeichen.

3.1.0 Binärtechnische Grundlagen

ASCII-Code

100 1101 ⇔ M
110 0001 ⇔ a
101 1011 ⇔ [
010 0001 ⇔ !

ASCII-Tabelle Standard

Dec	Hex	Steuerzeichen / Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End transm. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

3.1.0 Binärtechnische Grundlagen

MS-DOS Code-Tabelle

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	∑
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	6E	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	π
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	256	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	n
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	2
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	□