

Mikroprozessortechnik

Praxisbezogene Aufgaben

Laboraufgaben

zur Bearbeitung mit dem Emulator oder Simulator

Stand 30.10.2011

Stefan Lehmann

Als Vorbereitung zu den nachfolgenden Übungen sollten folgende Punkte jeder Studentin und jedem Studenten k l a r sein. Falls noch Unklarheiten bestehen sollten, ist das Vorlesungsskript, die Datenblätter der Bausteine sowie diverse Bücher heranzuziehen. Eventuell sind auch Kommilitonen in der Lage Rede und Antwort zu stehen. Es muß der interne Aufbau (Registerstruktur) und der Befehlssatz bekannt sein.

Die folgenden kurzen Programme sollten deshalb mit Leichtigkeit erstellt werden können:

- 8- bzw. 16-Bit Addition und Subtraktion
- 8-Bit Multiplikation
- und für „Experten“ 16-Bit durch 8-Bit Division
- Programmierung von Schleifen, I/O-Zugriffen, Initialisierung

In der Regel können alle Aufgaben sowohl für das Z80-System als auch für den PIC gelöst werden. Port RA des PICs entspricht Port 0 beim Z80, Port RB dem Port 1. RA1 beim PIC entspricht also Bit 1 des Port 0 beim Z80.

Aufgabe 1

Es soll ein einfacher Binärzähler programmiert werden. Dieser besitzt 8 Ausgänge (Port RB bzw. PORT 1) und einen Zähl Eingang RA0. Ein Reset-Eingang (RA1) setzt den Zählerstand auf 0, ein Inhibit-Eingang (RA2) verriegelt den Zählereingang bei H-Pegel und gibt ihn bei Lowpegel wieder frei. Der Pin RA3 ist der Carryausgang des Zählers. Er ist für die Dauer eines Taktes aktiv hat also H-Pegel beim Zählerübergang von 0FFH auf 00H.

Aufgabe 1a

Ergänzen Sie das obige Programm so, daß ein zweistelliger Dezimalzähler (00 bis 99) entsteht. Auch hier wird der Carryausgang beim Überlauf von 99 auf 00 wieder aktiv.

Aufgabe 2

Schreiben Sie ein Programm, das eine 4-Bit-BCD-Information an Port RA in eine Bitinformation für eine Siebensegmentanzeige konvertiert. RA0 sei LSB. Der Port RB ist als Ausgang geschaltet mit folgender Zuordnung:

RB0	Segment A	---A---	0000 -> 0	1000 -> 8
RB1	Segment B	I I	0001 -> 1	1001 -> 9
RB2	Segment C	F B	0010 -> 2	1010 -> H
RB3	Segment D	I I	0011 -> 3	1011 -> E
RB4	Segment E	---G---	0100 -> 4	1100 -> L
RB5	Segment F	I I	0101 -> 5	1101 -> P
RB6	Segment G	E C	0110 -> 6	1110 -> .
RB7	Punkt	---D--- Pkt	0111 -> 7	1111 -> (Blank)

Aufgabe 3:

Eine Frequenz zwischen 1kHz und 10kHz soll durch

a) 2 oder 4

b) 3 oder 5

geteilt werden. Frequenzeingang ist RA0, Frequenz Ausgang RA1.

Aufgabe 4:

Eine Frequenz zwischen 1kHz und 10kHz mit einem Tastverhältnis von 1:1 so verdoppelt werden. An RA0 liegt die Eingangsfrequenz, an RA1 die verdoppelte Frequenz an.

Aufgabe 5:

In einem Signal, das aus Frequenzen zwischen 1kHz und 10kHz besteht, treten immer wieder kurze Störimpulse auf. Diese Störungen sind mind. 1µs und max. 50µs lang. Diese Störimpulse sollen per Software „herausgefiltert“ werden. Eine Gesamterzögerung zwischen dem Eingangs- und Ausgangssignal von weniger als 100µs ist tolerierbar. Es können sowohl positive als auch negative Störimpulse auftreten. (Zeichnen Sie ein Zeitdiagramm).

Aufgabe 6:

Schreiben Sie das Programm aus Aufgabe Nr. 3 so um, dass ein mit Störsignalen durchsetztes serielles Datensignal „entstört“ werden kann. Das Datensignal hat folgende Parameter: 9600 bit/s, 8 Datenbit, 2 Stopbits und keine Parität. Der TTL-Ruhepegel beträgt +5V, der aktive Pegel ist 0V. Eine 1 wird durch +5V, eine 0 durch 0V übertragen.

Aufgabe 7:

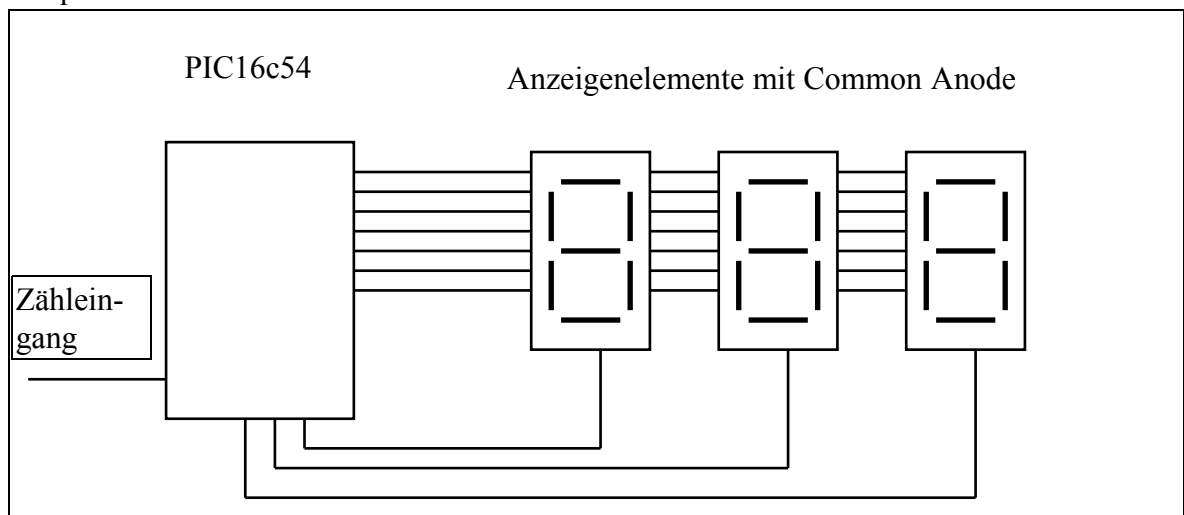
In einem Signalgemisch (RA0) kommen neben den Impulsen für die Information (50µs bis 500µs) auch Fehlimpulse vor. Diese Fehlimpulse haben eine Pulsdauer von unter 40µs. Der Ausgang RA1 soll dann auf High gehen, wenn das Verhältnis der Fehlimpulse zu Nutzimpulsen größer 10% wird.

Aufgabe 8:

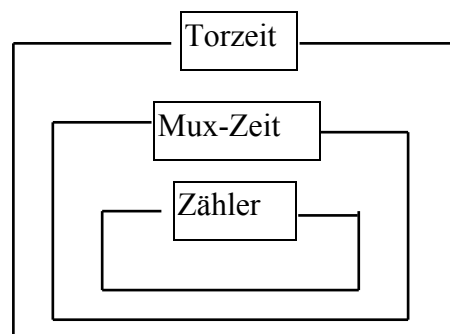
Ein Impulsgeber an einem Fahrrad gibt pro Radumdrehung 8 Impulse ab. Das Rad legt pro Umdrehung 1,60m zurück. Die 3-stellige gemultiplexte 7-Segment-LED-Anzeige soll den zurückgelegten Weg anzeigen. Das Rücksetzen der Zähler erfolgt durch einen Programmneustart mittels RESET. RB0 = A-Segment, RB1 = B-Segment usw. Einerstelle an bei RA0 = 1, mittlere Stelle an bei RA1 = 1 und linke Stelle an bei RA2 = 1.

Aufgabe 9

Frequenzzähler



Am Zähleingang RA3 liegt die zu messende Frequenz an. Der Frequenzbereich erstreckt sich von 0 Hz bis 99,9 (9,99) kHz. Die Multiplexfrequenz für die 3-stellige Anzeige soll ca. 100 Hz betragen (absoluter Wert unkritisch, die on-time pro Digit ca. 10msec). Die Torzeit soll per Software realisiert werden und entweder 0,5 s oder 1,0 s betragen, je nachdem ob nur eine oder beide Flanken des Meßsignals erfaßt werden.



Aufgabe 10 (nur für PIC)

Ein serieller Datenstring soll empfangen, kontrolliert und auf der Anzeige aus Aufgabe 3 angezeigt werden. Der Datenstring beinhaltet nur die Zeichen 0..9 und H, E, L, P und Blank. Die Datenübertragungsrate soll 4800 bps das Format 8 Datenbit ohne Parität haben. Der Datenstring wird eingerahmt durch STX und ETX.

z.B.

03	31	32	33	30	04
STX	1	2	3	Pos.d.	ETX
				Punktes	

Programmvariationen:

- a) Die Anzeige soll während des Empfanges dunkel sein
- b) Die Anzeige soll auch während des Empfangs gemultiplext werden.
- c) Der Datenstring wird durch eine Prüfsumme, die nach der Position des Kommas übertragen wird, ergänzt. Die Prüfsumme errechnet sich nach:

$$\{(\sum[b(x) \neq 0FFH]) \wedge 0FH\} \vee 30H$$

in den Grenzen von x=1 bis n (n = Anzahl der Informationsbytes, hier 4)
(b(x) sind die Datenbytes)

Bei einer seriellen Datenübertragung gelten folgende Festlegungen:

Ruhepegel	= HIGH
Startbit	= LOW
8 Datenbits	
Stopbit	= HIGH

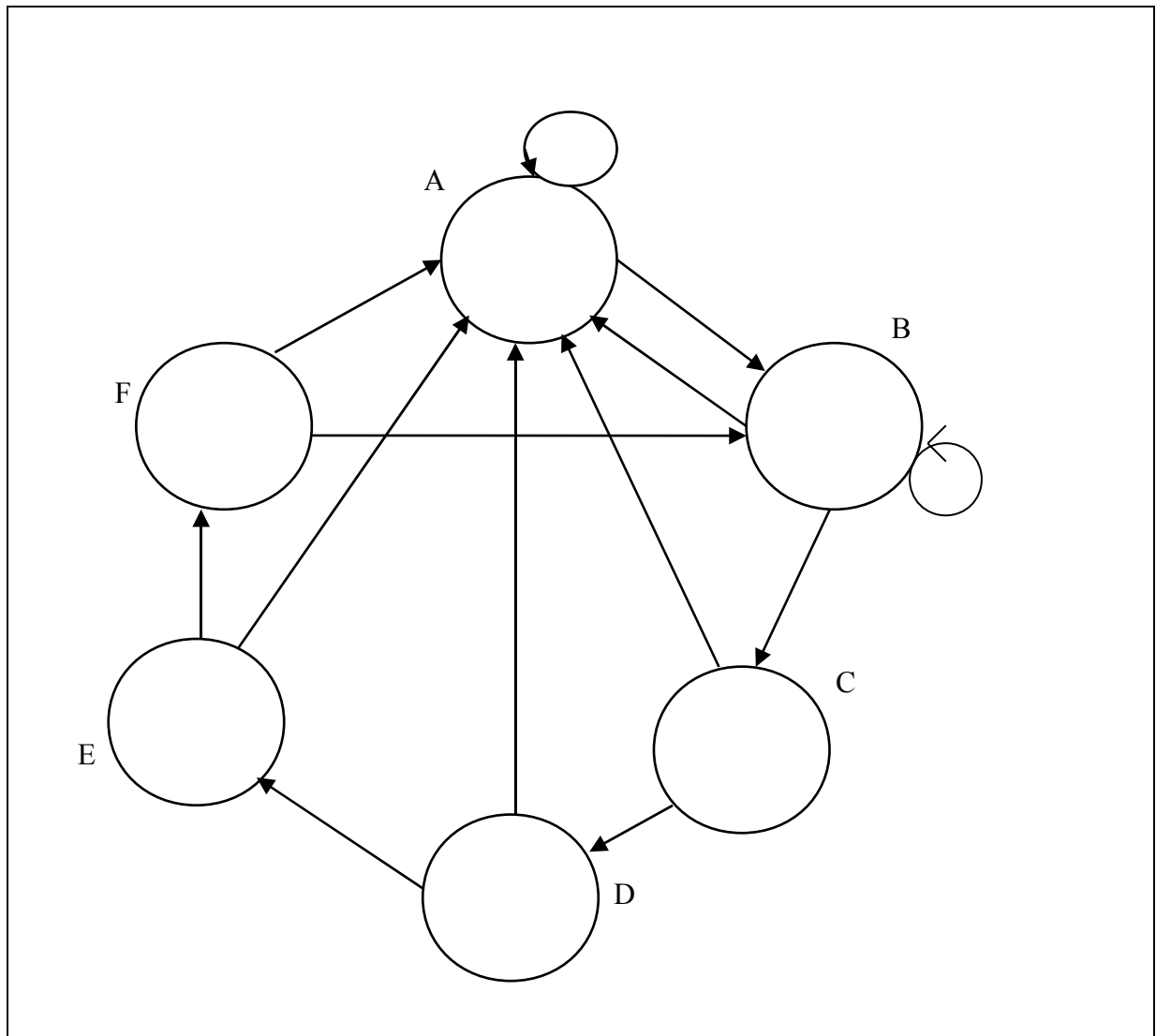
Aufgabe 11

Programmieren einer STATE-MACHINE

RB7 = 0 = Info an RA darf sich ändern

RB7 = 1 = Info an RA wird ausgewertet

Eingangsvariable RA3 .. RA0



Zustand A	IDLE	RB = X000 0000
Zustand B	WAIT	RB = X000 0110
Zustand C	STATE 1	RB = X000 1001
Zustand D	STATE 2	RB = X000 1111
Zustand E	STATE 3	RB = X010 1111
Zustand F	STATE 4	RB = X011 0000

Übergangsbedingungen

A -> B	RA = 0XXX
B -> A	RA = 0XX0 oder 1XXX
B -> C	RA = 0XX1
C -> A	RA = *
C -> D	RA = 0X11
D -> A	RA = *
D -> E	RA = 0100
E -> A	RA = *
E -> F	RA = 0101
F -> A	RA = *
F -> B	RA = 0111

* alle ungültigen Eingangsvariablen (Übergangsbedingungen) des entsprechenden Zustands

Hinweise:

Lösungsansatz in Tabellenform

Redundante Eingangsbits ausmaskieren.

Aufgabe 12:

Bild 1 zeigt eine Schaltung zur Steuerung der Helligkeit einer Halogenlampe. Diese Halogenlampe wird mit Hilfe einer Phasenanschnittsteuerung in ihrer Helligkeit verändert. Dazu wird die Lampe mittels TRIAC (Halbleiterschalter) ein- und ausgeschaltet. Der TRIAC kann nur gezündet (eingeschaltet) werden; um ihn auszuschalten, muss der Laststrom (entsp. dem Strom durch die Lampe) auf 0V zurückgehen, was bei einer Wechselspannung pro Periode zweimal erfolgt. Der TRIAC schaltet sich also in jedem Nulldurchgang der Wechselspannung selbst aus. Variiert man den Zündzeitpunkt, fließt durch die Lampe mehr oder weniger Strom. Wird beispielsweise der Zündzeitpunkt bei 10° gewählt, brennt die Lampe fast mit der vollen Helligkeit, bei 80° glimmt sie jedoch nur noch schwach. Die Fläche unter dem Kurvenzug entspricht der Leistung die die Lampe aufnimmt. Würde der Zündzeitpunkt sich laufend ändern, wäre ein unruhiges Flackern der Lampe die Folge. Aus diesem Grund muss der Zündzeitpunkt konstant zum Nulldurchgang gehalten werden. Dieser Nulldurchgang wird mit den Bauteilen R5, R6 und T1 detektiert und dem PIC zur Auswertung zugeleitet ($U_{BE} = 0,6V$). Der Spannungsverlauf nach dem Brückengleichrichter ist in Bild 2 skizziert.

Aufgabe:

Schreiben Sie ein Programm, das den Nulldurchgang erkennt und nach einer bestimmten Zeit (abhängig von der Stellung der Schalter S1..S4), den TRIAC einschaltet. Die Mindestzeit nach dem Nulldurchgang muss $3^\circ (= 166\mu s)$ betragen. Der positive Zündimpuls beträgt ebenfalls 3° . Damit ergeben sich folgende Zündzeitpunkte:

S4	S3	S2	S1	Zündzeitpunkt
0	0	0	0	TRIAC zündet nie, Lampe aus
0	0	0	1	$11,25^\circ$, Lampe brennt hell
0	0	1	0	$22,50^\circ$
0	0	1	1	$33,75^\circ$
0	1	0	0	$45,00^\circ$
0	1	0	1	$56,25^\circ$
0	1	1	0	$67,50^\circ$
0	1	1	1	$78,75^\circ$
1	0	0	0	$90,00^\circ$
1	0	0	1	$101,25^\circ$
1	0	1	0	$112,50^\circ$
1	0	1	1	$123,75^\circ$
1	1	0	0	$135,00^\circ$
1	1	0	1	$146,25^\circ$
1	1	1	0	$157,50^\circ$
1	1	1	1	$168,75^\circ$, Lampe glimmt schwach

An Hand der Tabelle erkennt man, dass nur durch betätigen des Schalters S1 die Lampe entweder aus ist, oder aber mit max. Helligkeit brennt. Sollen die Helligkeiten dazwischen gewählt werden, müssen auch die Schalter S2 bis S4 betätigt werden.

Aufgabe:

Skizzieren Sie den Spannungsverlauf am Kollektor des Transistors T1. Berechnen Sie die Impulslänge.

zu Aufgabe 12.

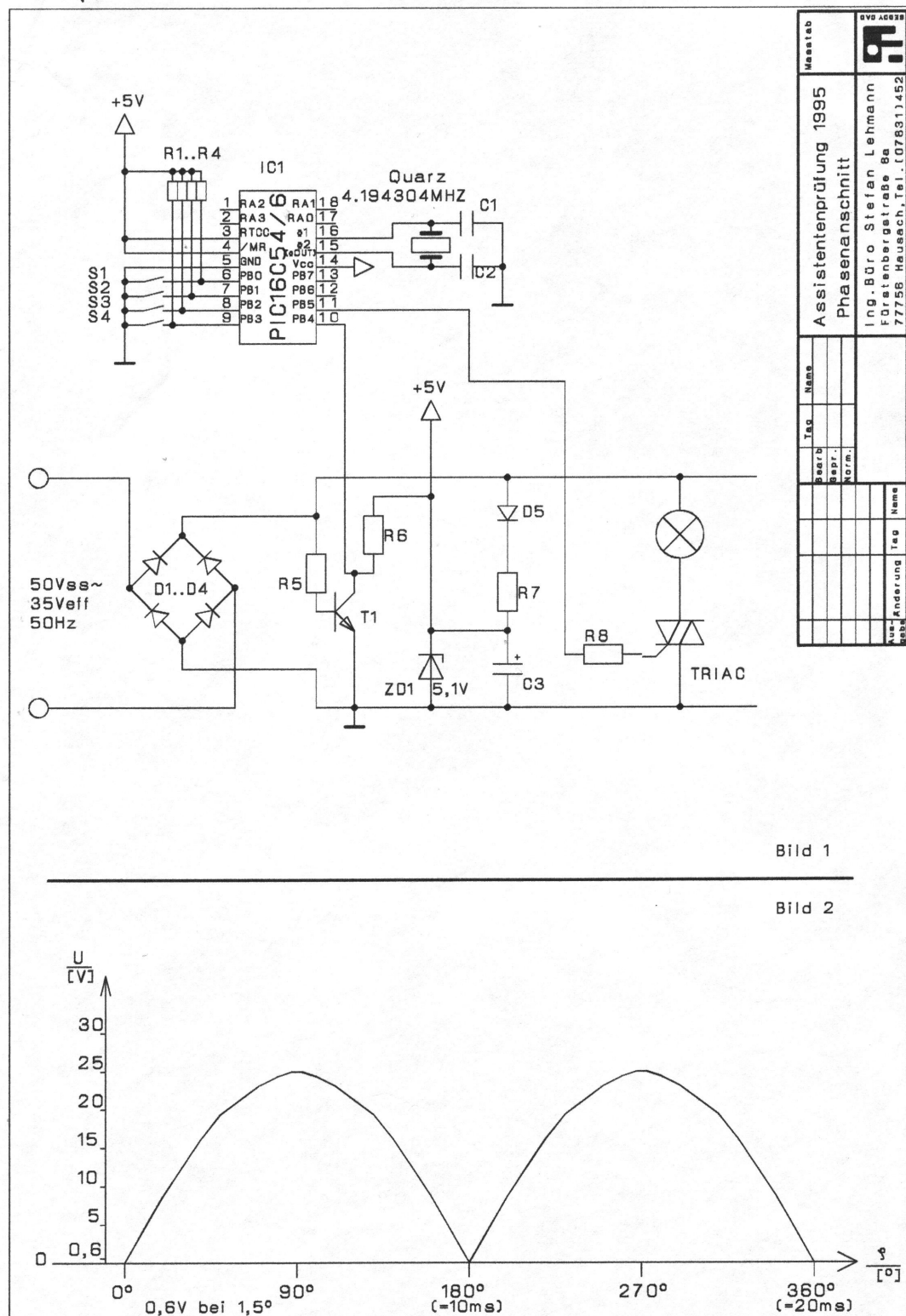
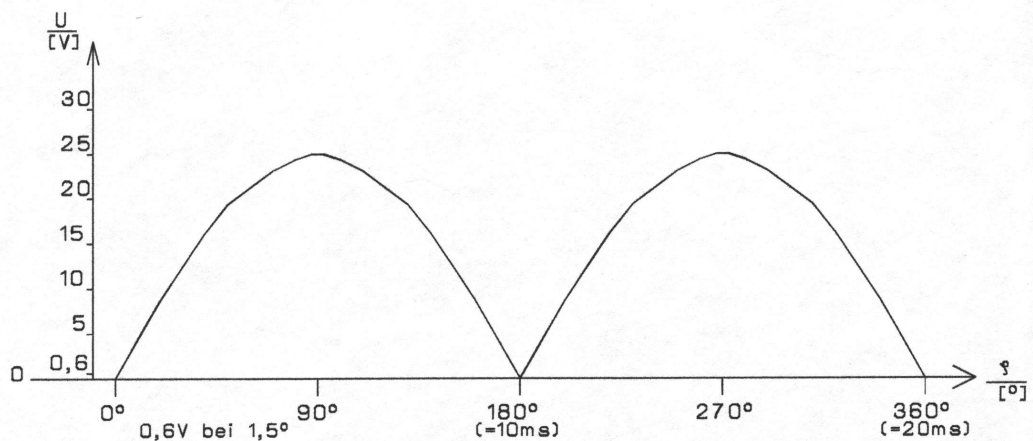


Bild 1



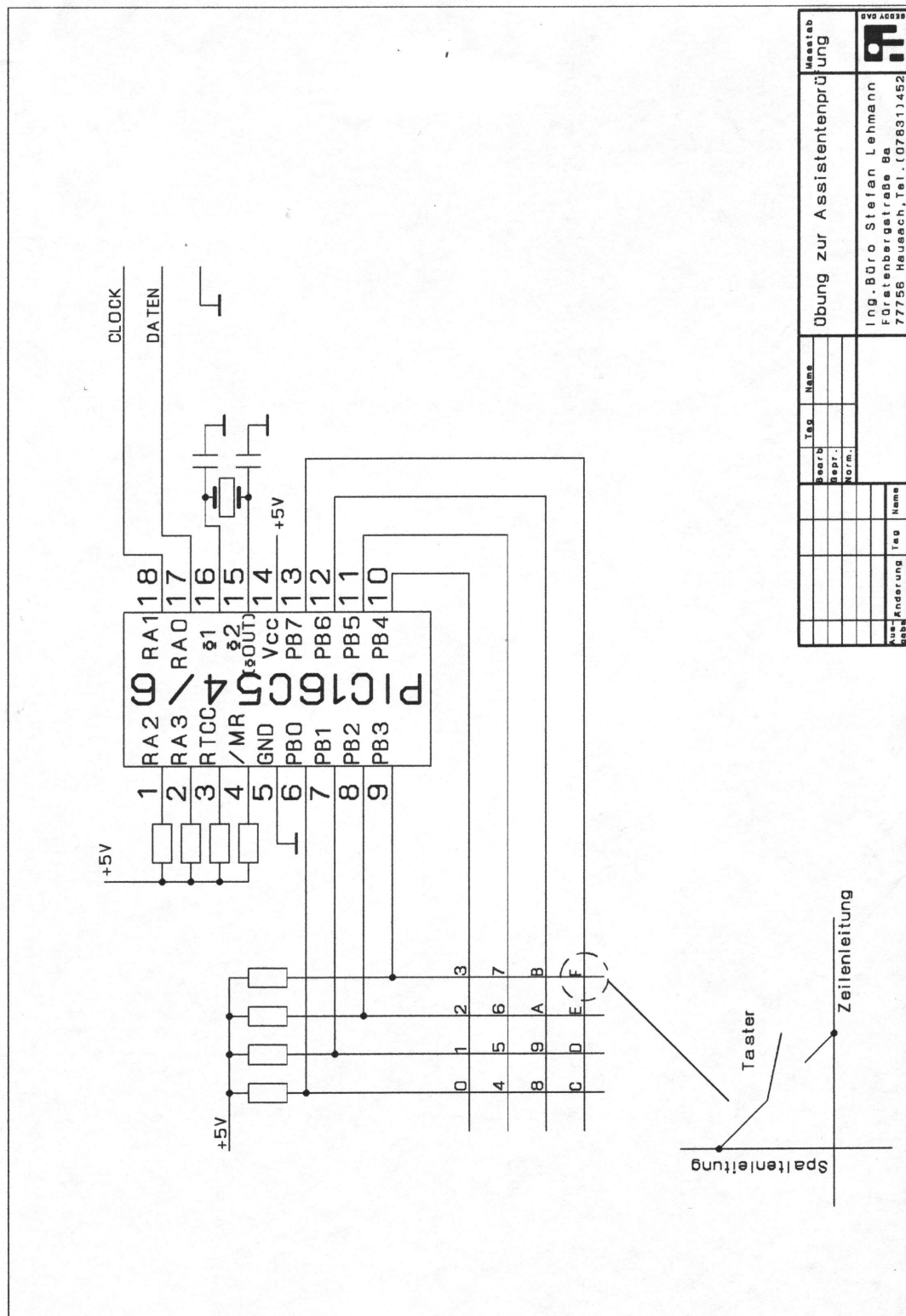
Aufgabe 13:

Die nachfolgende Schaltung zeigt einen PIC-Mikrocontroller mit einer 4 x 4 Matrixtastatur und einem seriellen synchronen TTL-Interface, wie es bei der Tastatur des PCs verwendet wird. Bei der Tastaturabfrage können per Software verschiedene Verhalten programmiert werden. Man spricht in diesem Zusammenhang von NO-ROLL-OVER-KEY, ROLL-OVER-KEY und n-ROLL-OVER-KEY. Sie unterscheiden sich darin, wie sie reagieren, wenn man die nächste Taste gedrückt, bevor die alte wieder freigelassen wurde. Die n-roll-over Variante lässt es sogar zu, dass bereits eine dritte Taste gedrückt werden kann, während die alten noch im aktiven Zustand (geschlossen) sind. Dieses unterschiedliche Verhalten wird rein softwaremäßig realisiert. Um überhaupt das gleichzeitige Drücken von 2 Tasten zu ermöglichen, müssen die einzelnen Leitungen entkoppelt werden. Dazu werden Dioden verwendet. Zeichnen Sie hierzu die Dioden in das Schaltbild an die richtige Stelle.

Das serielle, synchrone Übertragungsverfahren bedeutet, dass hier auf der Datenleitung die Daten seriell aufgelegt werden. Dazu wird immer dann, wenn die Daten gültig sind die Clockleitung aktiviert. Um ein Einschwingen der Pegel auf den Leitungen zu erlauben, wird die Clockleitung erst $1\mu\text{s}$ nach den Daten aktiv. Im Gegensatz dazu bleibt die Datenleitung noch $1\mu\text{s}$ nach Wegnahme des Clocksignals auf dem entsprechenden Pegel. Das Clocksignal hat eine Dauer von $8\mu\text{s}$. Zeichnen Sie das entsprechende Zeitdiagramm, das das ASCII-Zeichen „A“ überträgt. Ruhelpegel für Clock und Daten ist LOW.

Schreiben Sie ein Programm, das die Tastatur (1. no-roll-over-key ; 2. roll-over-key) einliest und die Daten dann mittels der seriellen, synchronen Schnittstelle weitersendet.

zu Aufgabe 13



Aufgabe 14:

Die nachfolgende Schaltung zeigt die Hardware für eine automatische Wasserfüllanlage. Dazu wird die Wasserzuführung mittels Magnetventil gesteuert. Das Magnetventil selbst wird über das Relais ein- und ausgeschaltet. Eine „1“ an RA0 schaltet das Relais und damit das Magnetventil ein; das Wasser läuft. Ein Durchflussmesser erfasst die Wassermenge und gibt pro Liter einen Impuls (zehn Impulse) ab. Diese Impulse werden über den Zähleringang RB7 eingelesen.

Das Befüllen wird durch das Drücken der Starttaste gestartet. Das Magnetventil wird geöffnet und die einfließenden Liter gezählt. Der Füllvorgang wird gestoppt, sobald die an den BCD-Schaltern eingestellte Literzahl erreicht ist. Das Einlesen dieses Maximalwertes soll immer dann erfolgen, wenn die Starttaste gedrückt wird. Diese zeitliche Verzögerung die sich daraus ergibt ist unerheblich.

Die LED an RA1 soll solange leuchten, wie das Wasser läuft. Der Taster „Manuell“ soll nicht berücksichtigt werden.

Erstellen Sie einen Ablaufplan.

Schreiben Sie das Programm in Assembler für den PIC 16C83.

Aufgabe 15:

In einem PIC 16C83 soll eine Digitaluhr mit Weckfunktion realisiert werden. Die Zeitbasis wird von der Netzfrequenz abgeleitet. Über einen Signalformer liegt an RA0 ein 100 Hz Signal an. An RA1 bis RA3 sind 3 Taster angeschlossen. Ist der Taster offen, liegt am entsprechenden Pin ein H-Signal (5V) an, wird der Taster geschlossen, fällt der Pegel auf 0V ab. Die Funktionen dieser Taster sind folgende:

RA1 weniger als 3 Sekunden gedrückt, bedeutet Alarm ein / aus
 mehr als 3 Sekunden gedrückt, Alarmzeit stellen
 mehr als 5 Sekunden gedrückt, Uhrzeit stellen

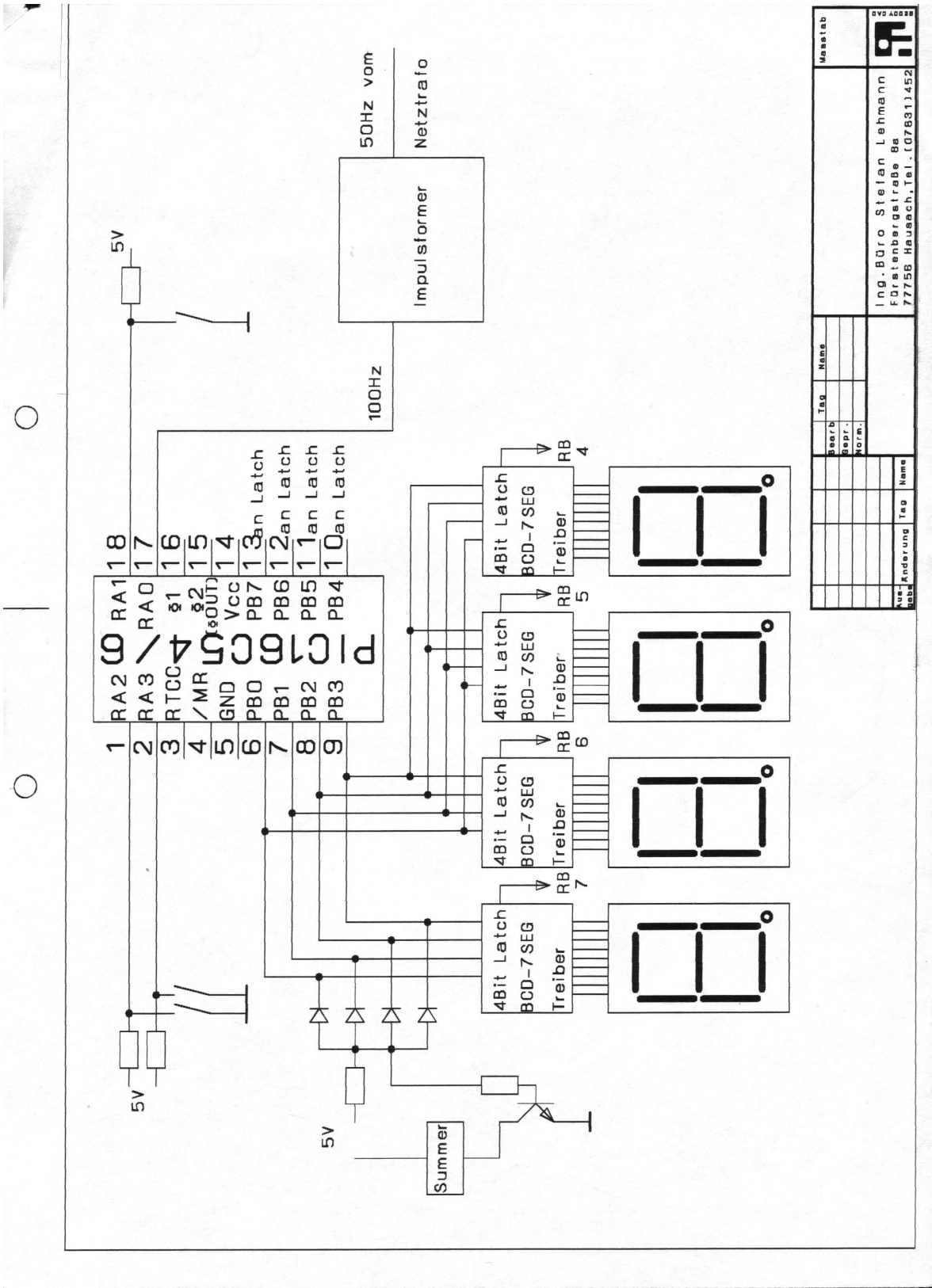
Nach dem Einstellen wird RA1 nochmals gedrückt um die Uhr zu starten.

RA2 Stunden hochzählen

RA3 Minuten hochzählen

Die Anzeige besteht aus 4 Siebensegmentanzeigen mit davorgeschalteten BCD-7-Segmentdecodern mit Latchfunktion. Um diese Anzeige anzusteuern, ist wegen der Latchfunktion kein Multiplex notwendig. Die BCD-Information wird auf die unteren 4 Bits von RB gelegt und mittels RB4 bis RB7 die gewünschte Stelle aktiviert. Hierzu ist ein positiver Impuls von mind. 5 μ s notwendig. Liegt am Dekoder ein Wert zwischen 10 (0AH) und 15 (0FH) wird diese Stelle dunkelgeschaltet.

Der Alarm wird aus der BCD-Information abgeleitet. Eine Hardware dekodiert die Bitkombination 1111 aus. Liegt diese an, wird ein Summer eingeschaltet.



Ausg.		Anderung		Tag		Name	

Aufgabe 16:

Eine Ampel an einer Straßenkreuzung mit Taster für die Fußgängerampel soll mittels Moore-Automat realisiert werden. Dabei soll die Verkehrsdichte a) nicht, b) mit die Zeitintervalle bestimmen.

- Problemanalyse mittels Zustandsdiagramm
- Umsetzung in Code z.B. Java, dabei sind die Automatenzustände als Integervariablen zu führen; damit sind boolsche Verknüpfungen möglich.
- graphische Simulation auf dem Bildschirm.

Programm für den PIC 16C83 schreiben. Hier sind u.U. noch Hardwareergänzungen notwendig.

Aufgabe 17:

Generieren Sie mittels PIC 16C83 die Synchronimpulse für ein Normvideosignal (BAS). Es sind für VS und HS getrennte Pins vorzusehen, die dann mittels zusätzlicher Logik verknüpft werden.

Schreiben Sie das entsprechende Programm und skizzieren Sie die Schaltung.

Erweitern Sie die Hardware um ein R2R-Netzwerk mit 4 Bits für das Grausignal, so dass auf dem Fernsehschirm 16 Graustufen dargestellt werden (Schaltbild). Erweitern Sie das Programm um diese Funktionen.

Aufgabe 18

Drehstromgenerator:

Mit einem PIC 16C83 soll ein 3-Phasen-Drehstromsignal erzeugt werden. Dabei wird pro Phase ein 3-Bit-DA-Wandler angeschlossen, der den digitalen Ausgabewert in eine äquivalente Spannung umsetzt. Die Frequenz wird mittels 4 Schalter in 16 Stufen von 0 Hz (Motor aus) bis 60 Hz (max. Motorgeschwindigkeit) eingestellt. Es gilt folgende Pinzuordnung:

RA0	Phase 1, Bit 0	RB0	Phase 2, Bit 0	RB3	Phase 3, Bit 0
RA1	Phase 1, Bit 1	RB1	Phase 2, Bit 1	RB4	Phase 3, Bit 1
RA2	Phase 1, Bit 2	RB2	Phase 2, Bit 2	RB5	Phase 3, Bit 2

RA3 Schalter 0 (niederwertigste Stelle)
RA4 Schalter 1
RB6 Schalter 2
RB7 Schalter 3 (höchstwertigste Stelle)

Die notwendigen Sinuswerte sind einer Tabelle zu entnehmen!

Lösungsansatz:

1. Die drei Ausgangssignale haben jeweils einen sinusförmigen Verlauf. Die entstehenden Stufen werden durch Tiefpassfilter geglättet.
2. Die drei Phasen sind starr miteinander gekoppelt. Der Phasenwinkel beträgt 120° .
3. Bei 3 Bit Auflösung ergeben sich 8 Stufen für die Schrittweite. Bei 360° pro Periode folgt, dass nach jeweils 45° ein neuer Ausgangswert berechnet und umgesetzt werden muss. (Hier ergibt sich ein Widerspruch! Welcher? Wie kann man diesen umgehen?).
180
4. Zur Realisierung werden 3 Zähler und 1 Timer (Hilfzähler) verwendet. Die Zähler zählen von 0 bis 7 und stellen den Index für den Tabellenzugriff dar.
5. Der Timer bestimmt die Geschwindigkeit mit der die übrigen Zähler inkrementiert werden, und bestimmt somit auch die generierte Frequenz. Je nach Schalterstellung (S1 bis S4) läuft dieser Timer schneller oder langsamer.
6. Bei 60 Hz läuft der Motor mit Maximalgeschwindigkeit, der Zähler bekommt somit die niedrigsten Startwerte geladen. Damit eine gewisse Grundgenauigkeit erreicht wird, sollte dieser Wert etwa 10 betragen.

Aufgabe:

- a) Wo liegt ein Widerspruch? Bitte entsprechend korrigieren.
- b) Ablaufdiagramm
- c) PIC-Assemblerprogramm

Aufgabe 19

Über die Ladezeit eines Kondensators kann man entweder die Kapazität oder den Widerstand ermitteln. Bei Microcontroller-Applikationen wird vor allem der Widerstand durch einen Sensor ersetzt. Liefert dieser ein der physikalischen Größe proportionalen Widerstand, kann man so Temperatur, Helligkeit, etc. messen.

Ein LDR hat einen Widerstandswert zwischen $10\ \Omega$ und $1\ \text{M}\Omega$. Der interessante Bereich sei zwischen $1\ \text{k}\Omega$ und $100\ \text{k}\Omega$. Schreiben Sie ein Programm das die entsprechende Ladezeit erfasst und den Widerstandswert anzeigt.

Bei diesem einfachen Verfahren geht die Genauigkeit des Kondensators mit in die Messung mit ein. Besteht eine Möglichkeit, den Einfluss des Kondensators zu eliminieren?

Skizze und PAP

Aufgabe 20

Entwickeln Sie ein Programm für eine Treppenhausbeleuchtung. Falls mehrere Taster notwendig sein sollten, werden diese alle als ODER-Schaltung angeschlossen. Die Taster sollen:

- a) nur das Licht einschalten und die voreingestellte Zeit ablaufen lassen
- b) wie a), aber wenn der Taster länger als 15 s gedrückt wird, soll auf Dauerlicht geschaltet werden. Ein erneuter kurzer Tastendruck schaltet das Dauerlicht wieder ab.

Über 4 Taster sollen Einzeiten zwischen 15 sec und 4 Minuten einstellbar sein.

Erstellen Sie ein Zustandsdiagramm und das Assemblerprogramm. Lösen Sie dabei auch folgendes Problem: wie erkennt man, dass nun das Dauerlicht ein- bzw. wieder ausgeschaltet ist?

Aufgabe 21

Schreiben Sie ein Programm, das einen Fahrstuhl steuert. Es sollen vorhanden sein: EG, 1.OG und 2.OG. Die gewünschte Fahrtrichtung soll wählbar sein. Im Aufzugskorb sind 3 Tasten für das Zielstockwerk. Der Motor wird über zwei Signale „links“ und „rechts“ angesteuert. Die Rückmeldung des erreichten Stockwerkes erfolgt über einfache Schließer. Die 12 I/O-Leitungen des 16C83 müssen ausreichend sein.

Anhang
ASCII-Tabelle

Dec	Hex	Zeichen	Dec	Hex	Zeichen	Dec	Hex	Zeichen	Dec	Hex	Zeichen
0	00	^@ NUL	32	20	Space	64	40	@	96	60	'
1	01	^A SOH	33	21	!	65	41	A	97	61	a
2	02	^B STX	34	22	“	66	42	B	98	62	b
3	03	^C ETX	35	23	#	67	43	C	99	63	c
4	04	^D EOT	36	24	\$	68	44	D	100	64	d
5	05	^E ENQ	37	25	%	69	45	E	101	65	e
6	06	^F ACK	38	26	&	70	46	F	102	66	f
7	07	^G BEL	39	27	‘	71	47	G	103	67	g
8	08	^H BS	40	28	(72	48	H	104	68	h
9	09	^I HT	41	29)	73	49	I	105	69	i
10	0A	^J LF	42	2A	*	74	4A	J	105	6A	j
11	0B	^K VT	43	2B	+	75	4B	K	107	6B	k
12	0C	^L FF	44	2C	,	76	4C	L	108	6C	l
13	0D	^M CR	45	2D	-	77	4D	M	109	6D	m
14	0E	^N SO	46	2E	.	78	4E	N	110	6E	n
15	0F	^O SI	47	2F	/	79	4F	O	111	6F	o
16	10	^P DEL	48	30	0	80	50	P	112	70	p
17	11	^Q DC1	49	31	1	81	51	Q	113	71	q
18	12	^R DC2	50	32	2	82	52	R	114	72	r
19	13	^S DC3	51	33	3	83	53	S	115	73	s
20	14	^T DC4	52	34	4	84	54	T	116	74	t
21	15	^U NAK	53	35	5	85	55	U	117	75	u
22	16	^V SYN	54	36	6	86	56	V	118	76	v
23	17	^W ETB	55	37	7	87	57	W	119	77	w
24	18	^X CAN	56	38	8	88	58	X	120	78	x
25	19	^Y EM	57	39	9	89	59	Y	121	79	y
26	1A	^Z SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	^[ESC	59	3B	;	91	5B	[123	7B	{
28	1C	^\ FS	60	3C	<	92	5C	\	124	7C	
29	1D	^] GD	61	3D	=	93	5D]	125	7D	}
30	1E	^^ RS	62	3E	>	94	5E	^	126	7E	~
31	1F	^_ US	63	3F	?	95	5F	_	127	7F	DEL