

Hard- und Software-Engineering / Automatisierungstechnik HS1 / AU1

SS 2009

Praktikumsaufgaben

Stand: 9. 6. 09

Versuch 1

Einführung in die C-Programmierung Atmel AVR.

1. Kennenlernen des grundsätzlichen Entwicklungsganges,
2. Darstellen von Zeit mittels Software,
3. elementare Anwendungsprogrammierung.

Grundlagen:

- AVR Studio 4,
- GNU C-Compiler,
- Starterkit AVR STK 500.

Grundkonfiguration: Port C mit den LEDs verbinden. Alle Ports auf Ausgang.

Aufgabe 1: Zeitdarstellung (1). Auf Port A abwechselnd 00H und FFH ausgeben. Kontrolle: mittels Oszilloskop. Wie lange dauert die kürzeste E-A-Schleife? (Im Versuch 2 werden wir die gleiche Aufgabe mittels Assemblerprogrammierung lösen.)

Aufgabe 2: Zeitdarstellung (2). Eine Funktion MILLISEC, die n Millisekunden darstellt, wobei der Parameter n eine 16-Bit-Binärzahl ist (hierzu brauchen wir zunächst eine Schleife die eine einzige Millisekunde darstellt). Funktionskontrolle wie bei Aufgabe 1.

Aufgabe 3: Lauflicht (1). Zyklische aufeinanderfolgende Erregung der LEDs in eine Richtung.

Aufgabe 4: Lauflicht (2). Zyklische aufeinanderfolgende Erregung der LEDs abwechselnd in beiden Richtungen.

Aufgabe 5: elementare Ein- und Ausgabe (1). Port D mit den Tasten verbinden und auf Eingabe programmieren. Jede Betätigung einer Taste muß dazu führen, daß die entsprechende LED leuchtet.

Aufgabe 6: elementare Ein- und Ausgabe (2). Port D mit den Tasten verbinden und auf Eingabe programmieren. Jede Betätigung einer Taste muß dazu führen, daß die entsprechende LED blinkt.

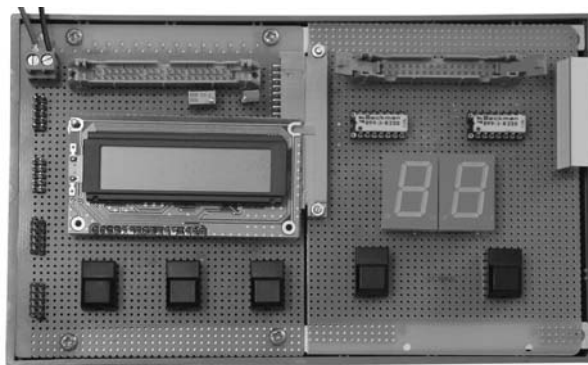
Aufgabe 7: Gebäudesystemtechnik (1). Ein- und Ausschalten des elektrischen Lichts. Mit einer der Tasten soll die gesamte LED-Reihe ein- und ausgeschaltet werden (erste Betätigung: ein, zweite Betätigung: aus usw.).

Aufgabe 8: Gebäudesystemtechnik (2). Jede LED soll durch Betätigen der jeweils benachbarten Taste ein- und ausgeschaltet werden, und zwar unabhängig von den anderen.

Aufgabe 9: elektronischer Würfel (1). Die LEDs des Starterkits sollen die Zahlen 1 bis 6 anzeigen, und zwar im Sinne einer Balkenanzeige (1 = 10000...; 2 = 110000...; 3 = 111000... usw.). Eine der Taster ausnutzen. Solange dieser Taster betätigt wird, ist zyklisch modulo 6 zu zählen.

Aufgabe 10: elektronischer Würfel (2). Siebensegmentanzeige mittels Übungsplattformen UeIDE 04 mit angesteckter Siebensegmentanzeige UeSSTa 04a (oder Nachfolger). Es ist eine Taste und eine Siebensegmentanzeige auszunutzen und ein Port (z. B. Port C) zu belegen. Die Wandlungstabelle BCD auf Siebensegment ist bereits vorhanden.

7	6	5	4	3	2	1	0
KEY#	G#	F#	E#	D#	C#	B#	A#



Aufgabe 11: Erweiterung von Aufgabe 10 auf das Ziehen von Lottozahlen. Statt von 1 bis 6 soll der Zähler von 1 bis 49 zählen. Hierzu werden beide Siebensegmentanzeigen belegt.

Versuch 2

Einführung in die Assemblerprogrammierung Atmel AVR. Elementare Anwendungsprogrammierung / die serielle Schnittstelle des Atmel AVR.

1. Die serielle Schnittstelle (1). Textausgabe.
2. Die serielle Schnittstelle (2) / einfache Interruptserviceroutinen (ISRs). Texteingabe.
3. Darstellen von Zeit mittels Software (Übungsstunden).

Versuchsordnung:

- Starterkit Atmel STK 500. Ankopplung an PC über die zweite serielle Schnittstelle. Bedienung über Hyperterminal. Interne Verbindungen: LEDs an Port C, Tasten an Port A. Anschließen der seriellen Schnittstelle nicht vergessen ...
- Programmiersprache: Assembler (Atmel Developer Studio).

Aufgabe 1: Ausgeben von Zeichen (Festtext) auf ein Bildschirm-Terminal (Windows-Hyperterminal).

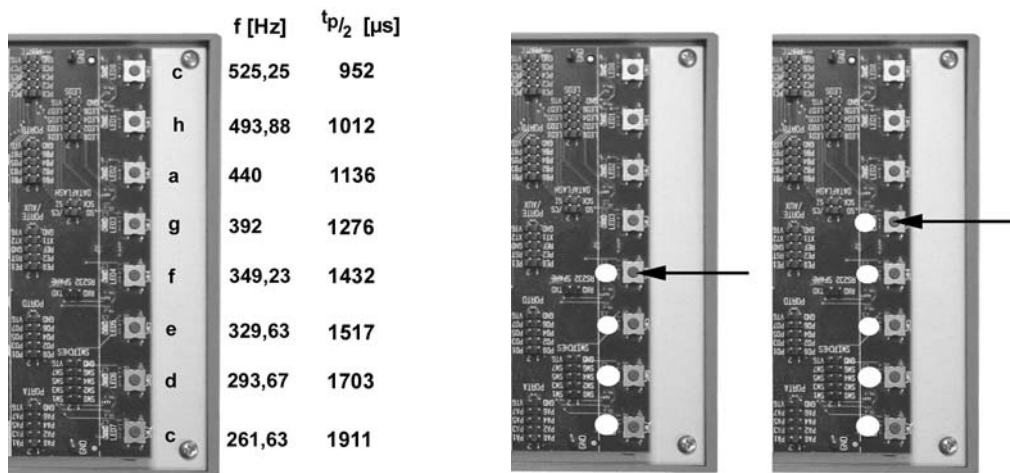
Aufgabe 2: Eine einfache Interruptserviceroutine (ISR), die das empfangene Zeichen auf den LEDs des Starterkits darstellt.

Aufgabe 3: Musikinstrument (1). Die Tasten werden zyklisch abgefragt. Eine betätigte Taste veranlaßt die Abgabe eines Tonsignals (Ports A und D; aktiv Low), das über Lautsprecher oder Schallgeber hörbar gemacht werden kann. Das Tonsignal ist eine symmetrische Rechteckwelle (Tastverhältnis 50:50) mit jeweils bestimmter Frequenz (gemäß chromatischer Tonleiter (s. nachstehend unter Kurzausbildung Musik; Frequenzangaben in Hz). Wenn kein Ton abgegeben wird, Ausgänge auf High (damit der angeschlossene Lautsprecher keinen Gleichstrom zieht).

Aufgabe 4: Musikinstrument (2). Spielen von Noten bei Bedienung über die serielle Schnittstelle. Jede Note soll durch Eingeben eines Zeichens an der PC-Tastatur ausgelöst werden. Da es sich im Grunde um eine Fleißaufgabe handelt, genügt es, die Achtelnoten der Tonleiter (s. Kurzausbildung) zu implementieren.

Aufgabe 5: Impulserzeugung / Pulsweitenmodulation. Mit dem Atmel-Starterkit sind variabel lange Impulse mit konstanter Periodendauer zu erzeugen (Grundlage der Pulsweitenmodulation (PWM)). Alles so einstellen (ggf. über Simulator), daß die Periodendauer ungefähr 2,5 ms entspricht (400 Hz). Bedienung: Hyperterminal. Zeichen C = alles Null; Zeichen + = erhöhen, Zeichen - = vermindern. Die gesamte Periodendauer entspricht acht Impulsschritten. Ausgabe: über Port C an die LEDs:

- a) Helligkeitssteuerung der gesamten LED-Reihe.
- b) als Bandskala: als Bei 1/8 Periodendauer ist nur die erste LED an, bei 2/8 die erste und die zweite usw.



Kurzausbildung Musik

Musik ist nicht einfach nur mit Geräusch verbunden, sondern sie reizt zum gleichnishafte Ansehen der dionysischen Allgemeinheit (F. Nietzsche); sie ist "... darin von allen anderen Künsten verschieden, daß sie nicht Abbild der Erscheinung, oder richtiger, der adäquaten Objectität des Willens, sondern unmittelbar Abbild des Willens selbst ist und als zu allem Physischen der Welt das Metaphysische, zu aller Erscheinung das Ding an sich darstellt." (A. Schopenhauer). Wir fangen allerdings erst einmal ganz klein an ...



Eine Taktlänge entspricht 2s unterteilt in 4 Schläge à 500ms bei einem Tempo von 120 bpm

<u>Noten</u>				<u>Pausen</u>				<u>Tempo: 120 bpm</u>
○	=	ganze Note	= 4/4 = 2000 ms	■	=	ganze Pause	= 4/4 = 2000 ms	
♩	=	halbe Note	= 2/4 = 1000 ms	▬	=	halbe Pause	= 2/4 = 1000 ms	
♪	=	viertel Note	= 1/4 = 500 ms	⏏	=	viertel Pause	= 4/4 = 500 ms	
♪	=	achtel Note	= 1/8 = 250 ms	⏏	=	achtel Pause	= 4/4 = 250 ms	
♫	=	sechzehntel Note	= 1/16 = 125 ms	⏏	=	sechzehntel Pause	= 4/4 = 125 ms	
<u>andere Zeichen:</u>								
				:	=	Wiederholungszeichen		

Note	Frequenz (Hz)	Halbperiode (μs)	$\frac{1}{16}$ Note (125 ms) entspricht ... Perioden
c	525,25	952	66
h	493,88	1012	62
a	440	1136	55
g	392	1276	49
f	349,23	1432	44
e	329,63	1517	41
d	293,67	1703	37
c	261,63	1911	33

Versuch 3

Einführung in die Assemblerprogrammierung Atmel AVR. A/D-Wandler / Zähler und Zeitgeber / LCD-Punktmatrixanzeige.

Aufgabe 1: A/D-Wandler. Digitale Anzeige einer variablen Spannung (binär auf den LEDs des Starterkits, hexadezimal auf Bildschirm-Terminal).

Versuchsordnung: Einspeisen einer Spannung vom Labornetzgerät in Port A, Bitposition 0.

Achtung: Vor dem Anschließen Netzgerät auf 0 V. Nicht über 5 V gehen!!!

Aufgabe 2: Zähler/Zeitgeber. Wir verwenden den Counter/Timer 0 und probieren verschiedene Betriebsarten aus:

- a) Rechteckwellenerzeugung mit Compare Output Mode.
- b) Pulsweitenmodulation im einfachen Modus (Fast PWM Mode).
- c) Pulsweitenmodulation im phasenrichtigen PWM-Modus.

Ausgabe über Pin OC1 = Port B3. Richtungsregister (DDR) auf Ausgabe stellen!

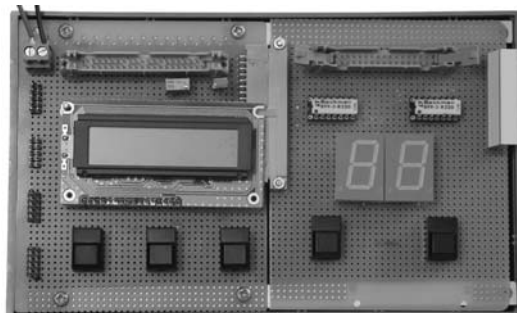
Für die Zählweite definieren wir ein Register, das anfänglich auf einen mittleren Wert (128) geladen wird. Um die Einstellung zu verändern, nutzen wir das Hyperterminal über die serielle Schnittstelle und die in Versuch 2 geschriebene Interruptserviceroutine. Z. B. Zeichen 0 = aus (Wert 0) , + = beschleunigen, - = verringern.

Weitere Demonstrationen: (1) Drehzahlsteuerung, (2) Helligkeitssteuerung der LEDs, (3) Nutzung als D-A-Wandler.

Aufgabe 3: LCD-Punktmatrixanzeige.

Versuchsordnung:

- Starterkit Atmel STK 500. Ankopplung an PC über die 2. serielle Schnittstelle. Bedienung über Hyperterminal.
- Übungstafel UeSSTa 04a. Anschluß an Starterkit: Daten: Port C, Steuersignale: Port B. Einzelheiten s. Kurzbeschreibung UeSSTa 04a.



Teilaufgabe 1: Stellen Sie auf dem Dotmatrixdisplay (2 Zeilen zu 16 Zeichen) dar:

1. ein einzelnes Zeichen (Probe dafür, daß die Initialisierung geklappt hat),
2. einen einzeiligen Festtext nach eigener Wahl,
3. einen zweizeiligen Festtext anc eigener Wahl.

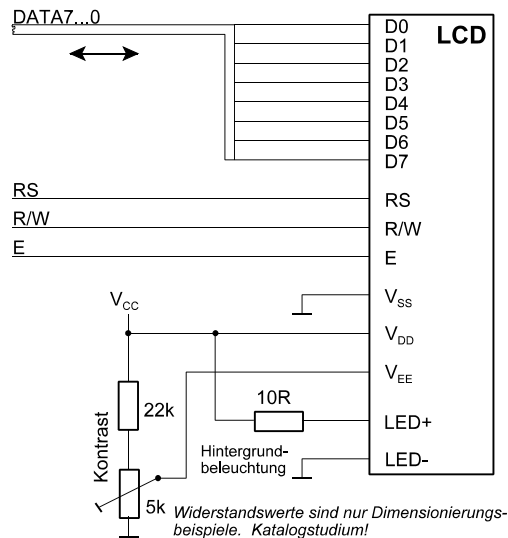
Bei der LCD-Ansteuerung beschränken wir uns auf reine Ausgabeabläufe, wobei die jeweilige Kommandoausführungszeit im Prozessor abgewartet wird.

Teilaufgabe 2: Nutzung der seriellen Schnittstelle:

1. Erweitern Sie die Interruptserviceroutine (ISR) der seriellen Schnittstelle so, daß die empfangenen Zeichen auf der LCD-Anzeige dargestellt werden (einfachste Funktion; noch ohne Schönheiten).
2. Erweitern Sie die ISR auf einen selbsttätigen Wrap Around (Übergang in die zweite Zeile, wenn Ende der ersten Zeile erreicht, Übergang an den Anfang, wenn Ende der 2. Zeile erreicht). Falls noch Zeit ist: die Tasten Backspace und Enter vernünftig auswerten (Rückschritt, Anfang der neuen Zeile), so daß sich die Anzeige wie ein kleines Terminal verhält.

Kurzausbildung LCD-Display

Interface:



Datenbus: Port C, Steuersignale: Port B.

Port C:

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Port B (1)

Betrifft Übungsplattform UeIDE 04 und AVR-Portadapter AVRPA 05.

7	6	5	4	3	2	1	0
ENTER	-	DOWN	ENTER	E (STB)	RW	-	RS
Eingang	x	Eingang		Ausgang		x	Ausgang

x = frei nutzbar

Port B (2)

Betrifft LED/LCD-Übungstafel 03a (UeLED/LCD 03a).

7	6	5	4	3	2	1	0
LED DATA	LED STROBE	LCD D/I (RS)	LCD R/W	LCD E (STB)	KEY RIGHT	KEY MIDDLE	KEY LEFT
OUT					IN		

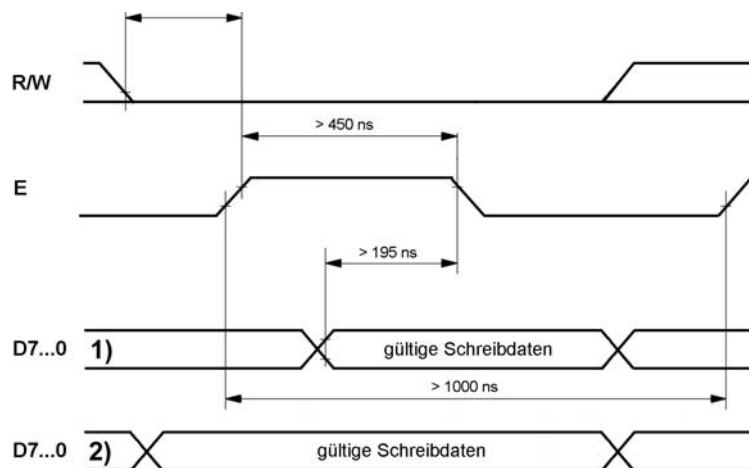
E: LCD-Erlaubniseingang (Enable, Strobe). 0 = kein LCD-Zugriff, 1 = LCD-Zugriff.

R/W: LCD-Zugriffssteuerung, 0 = Schreiben, 1 = Lesen.

RS: LCD-Registerauswahl: 0 = LCD-Steuerregister, 1 = LCD-Datenregister.

Der typische Ablauf eines Schreibzugriffs:

1. RS je nach Zugriff einstellen. R/W auf 0.
2. Schreibdaten auf den Bus legen.
3. von Schritt 1 an müssen wenigstens 140 ns vergangen sein. E-Impuls erzeugen (mindestens 450 ns).
4. Ggf. Schreibdaten vom Bus nehmen und Bus freigeben.
5. Beginn des nächsten Zugriffs: der nächste E-Impuls darf frühestens 1 µs nach Schritt 3 ausgelöst werden.



Die Zeichendarstellung wird durch Folgen entsprechender Kommandos aufgebaut. Nach dem Senden eines Kommandos ist die Anzeige zunächst mit dessen Ausführung beschäftigt und kann kein weiteres Kommando annehmen (Besetztzustand). Wir implementieren folgenden Ablauf:

- das erste Kommando übertragen,
- lange genug warten (gemäß maximaler Ausführungszeit (s. Tabelle)),
- das zweite Kommando übertragen,
- lange genug warten usw.

Initialisierung:

Nach dem Rücksetzen Kommandos gemäß folgender Tabelle übertragen:

Kommando	Steuerl.		Datenbyte								Anmerkungen
	RS	R/W	7	6	5	4	3	2	1	0	
Function Set	0	0	0	0	1	1	1	0	0	0	8-Bit-Betrieb. 2 Zeilen, Zeichenraster 5 • 8
Clear Display	0	0	0	0	0	0	0	0	0	1	Datenspeicher löschen. Cursor auf erste Zeichenposition (links (oben))
Display On/Off Control	0	0	0	0	0	0	1	1	1	1	Anzeige ein, Cursorsdarstellung ein, Cursor blinken
Entry Mode Set	0	0	0	0	0	0	0	1	1	0	Cursoradresse zählt aufwärts (Autoincrement); kein Schieben

Zeichentabelle (ASCII):

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20:	!	"	#	\$	%	&	'	<	>	*	+	,	-	.	/	
30:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50:	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
60:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70:	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Datenspeicheradressierung:

Anzeige	1. Zeile	2. Zeile	3. Zeile	4. Zeile
2 • 16	00H...0FH	40H...4FH		
2 • 20	00H...13H	40H...53H		
2 • 24	00H...17H	40H...57H		
2 • 40	00H...27H	40H...67H		
4 • 16	00H...0FH	40H...4FH	10H...1FH	50H...5FH
4 • 20	00H...13H	40H...53H	14H...27H	54H...67H

Es gibt keinen automatischen Übergang von Zeile zu Zeile. Vor dem Eintragen in eine bestimmte Zeile jeweils Datenspeicheradresse setzen (Kommando *DD RAM Adrs Set*).

Kommandoübersicht:

Kommando	Steuerl.		Datenbyte								Beschreibung	max. Ausführungszeit ²⁾	
	RS	R/W	7	6	5	4	3	2	1	0			
Clear Display	0	0	0	0	0	0	0	0	0	0	1	gesamte Anzeige löschen. Datenspeicheradresse auf Null	1,52 / 1,64 ms
Return Home	0	0	0	0	0	0	0	0	0	1	*1)	Datenspeicheradresse auf Null. Anzeige an Originalposition (keine Verschiebung). Datenspeicherinhalt bleibt erhalten	1,52 / 1,64 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S		Einstellung der Bewegungsrichtung des Cursors (I/D). Wahl zwischen Cursorbewegung und Verschieben der Anzeige (S)	37 / 40 µs
Display On/Off Control	0	0	0	0	0	0	1	D	C	B		Anzeige ein/aus (D), Cursor ein/aus (C), Blinken an Cursorposition ein/aus (B)	37 / 40 µs
Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*1)	*1)		Cursorbewegung und Verschieben der Anzeige	37 / 40 µs
Function Set	0	0	0	0	1	DL	N	F	*1)	*1)		Einstellung der Zugriffsbreite (DL), der Zeilenzahl (N) und des Zeichensatzes (F)	37 / 40 µs
CG RAM Adrs Set	0	0	0	1	Zeichengeneratoradresse					Adresse des ladbaren Zeichengenerators einstellen		37 / 40 µs	
DD RAM Adrs Set	0	0	1	Datenspeicheradresse					Datenspeicheradresse einstellen		37 / 40 µs		
Busy Flag / Adrs Read	0	1	BF	Adreßzähler					Busy-Bit (BF) und aktuelle Adreßzählerbelegung lesen		-		
CG RAM / DD RAM Data Write	1	0	zu schreibendes Byte					Schreiben in ausgewählten Speicher		37 / 40 µs			
CG RAM / DD RAM Data Read	1	1	gelesenes Byte					Lesen des ausgewählten Speichers		37 / 40 µs			

1): bedeutungslos (don't care); 2): 44780U / herkömmliche Ausführungen