

UART

Maik Holzhey

Fakultät IV
Technische Universität Berlin

22. Mai 2014

Begriffsklärung

UART

Begriffsklärung

UART

Universal Asynchronous Receiver and Transmitter

serielle Schnittstelle - asynchroner Modus

serielle Schnittstelle - asynchroner Modus

seriell

serielle Schnittstelle - asynchroner Modus

seriell

Bitfolge wird auf **einer** Leitung
mit festgelegter Wortbreite
übertragen

serielle Schnittstelle - asynchroner Modus

seriell

asynchron

Bitfolge wird auf **einer** Leitung
mit festgelegter Wortbreite
übertragen

serielle Schnittstelle - asynchroner Modus

seriell

Bitfolge wird auf **einer** Leitung mit festgelegter Wortbreite übertragen

asynchron

Datenübertragung wird intern abgesprochen und ist **nicht** von einer Takt-Leitung (SCK) abhängig

Motivation - Warum überhaupt?

Motivation - Warum überhaupt?

- Kommunikation zwischen logischen Einheiten ermöglicht effiziente Entwicklung

Motivation - Warum überhaupt?

- Kommunikation zwischen logischen Einheiten ermöglicht effiziente Entwicklung
- Asynchrone Schaltkreise sind ökonomischer im Stromverbrauch (abschaltbar)

Motivation - Warum überhaupt?

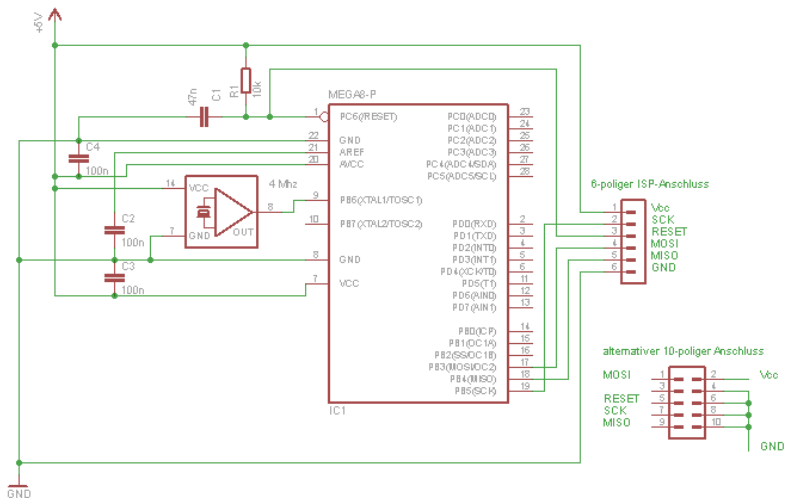
- Kommunikation zwischen logischen Einheiten ermöglicht effiziente Entwicklung
- Asynchrone Schaltkreise sind ökonomischer im Stromverbrauch (abschaltbar)
- Asynchrone Schnittstellen haben das Potential wesentlich schneller zu sein als synchrone

- 1 Hinführung
- 2 Hardware
- 3 Software
 - Baudrate
 - Datenübertragung
- 4 Handshake
 - Hardware Handshake
 - Software Handshake
- 5 Schlussbemerkung

2. Hardware

Erinnerung an den generellen Aufbau

Erinnerung an den generellen Aufbau



UART-Anschluss: Was ist neu/anders?

UART-Anschluss: Was ist neu/anders?

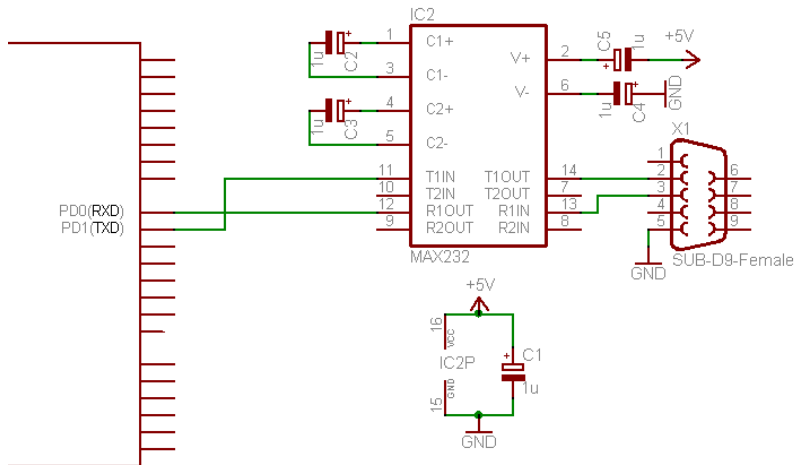


Abbildung: mikrocontroller.net/articles/AVR-Tutorial:_UART

Was brauchen wir?

Was brauchen wir?

- UART-spezifischen IC: MAX232 (Pegelwandler) und RS232

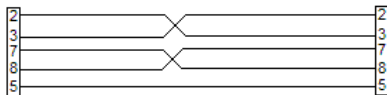
Was brauchen wir?

- UART-spezifischen IC: MAX232 (Pegelwandler) und RS232
- Kondensatoren zur Entkopplung der Versorgungsspannung

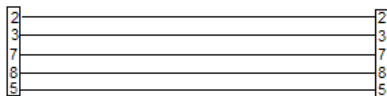
Was brauchen wir?

- UART-spezifischen IC: MAX232 (Pegelwandler) und RS232
- Kondensatoren zur Entkopplung der Versorgungsspannung
- SUB D9 Verbinder für 9-poliges Verlängerungskabel (Modem-Kabel)

Modemkabel



Nullmodem Kabel



Modem Kabel

Abbildung: mikrocontroller.net/articles/AVR-Tutorial:_UART

3. Software

Was wollen wir leisten?

Was wollen wir leisten?

- Synchronisierung im Falle der Datenübertragung (Start/Stop-Bits)

Was wollen wir leisten?

- Synchronisierung im Falle der Datenübertragung (Start/Stop-Bits)
- Festlegung der Bits pro Symbol

Was wollen wir leisten?

- Synchronisierung im Falle der Datenübertragung (Start/Stop-Bits)
- Festlegung der Bits pro Symbol
- Koordinierung der Datenverarbeitung

Émile Baudot - Baudrate



1

Émile Baudot (1845 - 1903)

Baudrate

Baudrate

Symbolrate

Baudrate

Symbolrate

- abgeleitet von [s] - Wörter pro Sekunde

Baudrate

Symbolrate

- abgeleitet von [s] - Wörter pro Sekunde
- Schrittfrequenz eines Datensignals

Baudrate

Symbolrate

- abgeleitet von [s] - Wörter pro Sekunde
- Schrittfrequenz eines Datensignals
- ein Schritt im Datensignal ist die Übertragung des kürzesten Datensignals (besteht mindestens aus einem Bit)

UART konfigurieren

UART konfigurieren

$$UBRR = \frac{\text{Taktfrequenz}}{16 \cdot \text{Baudrate}} - 1$$

UART konfigurieren - speziell bei ATmega

UART konfigurieren - speziell bei ATmega

$$UBRR = \frac{\text{Taktfrequenz} + \text{Baudrate} \cdot 8}{16 \cdot \text{Baudrate}} - 1 = \left(\frac{\text{Taktfrequenz}}{16 \cdot \text{Baudrate}} - 0.5 \right)$$

Fehlerrechnung

$$\%Fehler_{(Baudrate)} = \left(\frac{UBRR_{gerundet} + 1}{UBRR_{genau} + 1} - 1 \right) \cdot 100$$

Mögliche Frequenzen

Mögliche Frequenzen

- 1,8432 MHz (klassische Frequenz der PC-Baudrate)
- 3,6864 MHz
- 4,9152 MHz
- 6,1440 MHz
- 7,3728 MHz
- 9,8304 MHz
- 11,0592 MHz
- 12,2880 MHz
- 14,7456 MHz
- 18,4320 MHz
- 19,6608 MHz
- 22,1184 MHz

Mögliche Frequenzen

- 1,8432 MHz (klassische Frequenz der PC-Baudrate)
- 3,6864 MHz = $2 \cdot 1,8432\text{MHz}$
- 4,9152 MHz = $2 \cdot 2,4576\text{MHz}$
- 6,1440 MHz
- 7,3728 MHz = $4 \cdot 1,8432\text{MHz}$
- 9,8304 MHz = $4 \cdot 2,4576\text{MHz}$
- 11,0592 MHz
- 12,2880 MHz = $2 \cdot 6,144\text{MHz}$
- 14,7456 MHz = $8 \cdot 1,8432\text{MHz}$
- 18,4320 MHz = $3 \cdot 6,144\text{MHz}$
- 19,6608 MHz = $8 \cdot 2,4576\text{MHz}$
- 22,1184 MHz

Notwendigkeit des Quarzoszillators

Notwendigkeit des Quarzoszillators

- Baudrate muss sich durch ganzzahlige Teilung der Taktfrequenz ableiten lassen

Notwendigkeit des Quarzoszillators

- Baudrate muss sich durch ganzzahlige Teilung der Taktfrequenz ableiten lassen
- ansonsten entsteht eine Fehler in der Abtastung des seriellen Signals, der zu groß wird

Notwendigkeit des Quarzoszillators

- Baudrate muss sich durch ganzzahlige Teilung der Taktfrequenz ableiten lassen
- ansonsten entsteht eine Fehler in der Abtastung des seriellen Signals, der zu groß wird
- beim Baudratenquarz ist der Fehler gleich der Toleranz des Quarzes

Notwendigkeit des Quarzoszillators

- Baudrate muss sich durch ganzzahlige Teilung der Taktfrequenz ableiten lassen
- ansonsten entsteht eine Fehler in der Abtastung des seriellen Signals, der zu groß wird
- beim Baudratenquarz ist der Fehler gleich der Toleranz des Quarzes
- Quarzfrequenz muss wegen Abtastungsvorgängen das 16-fache der gewünschten Baudrate betragen

Codebeispiel: Baudrate

```
1 ;Quelle: mikrocontroller.net
2 .include "m8def.inc"
3 .def temp = R16
4
5 .equ F_CPU = 4000000           ; Systemtakt in Hz
6 .equ BAUD = 9600              ; Baudrate
```


Codebeispiel: Baudrate

```
1 ;Quelle: mikrocontroller.net
2 .include "m8def.inc"
3 .def temp = R16
4
5 .equ F_CPU = 4000000           ; Systemtakt in Hz
6 .equ BAUD = 9600              ; Baudrate
```

```
1 ; Berechnungen
2 .equ UBRR_VAL = ((F_CPU+BAUD*8)/(BAUD*16)-1) ; clever runden
3 .equ BAUD_REAL = (F_CPU/(16*(UBRR_VAL+1))) ; Reale Baudrate
4 .equ BAUD_ERROR = ((BAUD_REAL*1000)/BAUD-1000) ; Fehler 1/1000
5
6 .if ((BAUD_ERROR>10) || (BAUD_ERROR<-10))
7   .error "Systematischer Fehler der Baudrate grösser 1 Prozent "
8 .endif
```

Datenübertragung

Was machen wir jetzt mit den Daten?

Datentypen und Adressierung

Datentypen und Adressierung

- Senden von Daten via TXD-pin am μC

Datentypen und Adressierung

- Senden von Daten via TXD-pin am μC
- Empfangen von Daten via RXD-pin am μC

Datentypen und Adressierung

- Senden von Daten via TXD-pin am μC
- Empfangen von Daten via RXD-pin am μC
- Interrupts bieten sich zur Verfeinerung der Kommunikation im Programmablauf an.

Datentypen und Adressierung

- Senden von Daten via TXD-pin am μC
- Empfangen von Daten via RXD-pin am μC
- Interrupts bieten sich zur Verfeinerung der Kommunikation im Programmablauf an.
- Strings enden in C-Programmen mit 0x00

ASCII-Tabelle

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	€#32;	Space	64	40	100	€#64;	@	96	60	140	€#96;	`
1	1	001	SOH (start of heading)	33	21	041	€#33;	!	65	41	101	€#65;	A	97	61	141	€#97;	a
2	2	002	STX (start of text)	34	22	042	€#34;	"	66	42	102	€#66;	B	98	62	142	€#98;	b
3	3	003	ETX (end of text)	35	23	043	€#35;	#	67	43	103	€#67;	C	99	63	143	€#99;	c
4	4	004	EOT (end of transmission)	36	24	044	€#36;	\$	68	44	104	€#68;	D	100	64	144	€#100;	d
5	5	005	ENQ (enquiry)	37	25	045	€#37;	%	69	45	105	€#69;	E	101	65	145	€#101;	e
6	6	006	ACK (acknowledge)	38	26	046	€#38;	&	70	46	106	€#70;	F	102	66	146	€#102;	f
7	7	007	BEL (bell)	39	27	047	€#39;	'	71	47	107	€#71;	G	103	67	147	€#103;	g
8	8	010	BS (backspace)	40	28	050	€#40;	(72	48	110	€#72;	H	104	68	150	€#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	€#41;)	73	49	111	€#73;	I	105	69	151	€#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	€#42;	*	74	4A	112	€#74;	J	106	6A	152	€#106;	j
11	B	013	VT (vertical tab)	43	2B	053	€#43;	+	75	4B	113	€#75;	K	107	6B	153	€#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	€#44;	,	76	4C	114	€#76;	L	108	6C	154	€#108;	l
13	D	015	CR (carriage return)	45	2D	055	€#45;	-	77	4D	115	€#77;	M	109	6D	155	€#109;	m
14	E	016	SO (shift out)	46	2E	056	€#46;	.	78	4E	116	€#78;	N	110	6E	156	€#110;	n
15	F	017	SI (shift in)	47	2F	057	€#47;	/	79	4F	117	€#79;	O	111	6F	157	€#111;	o
16	10	020	DLE (data link escape)	48	30	060	€#48;	0	80	50	120	€#80;	P	112	70	160	€#112;	p
17	11	021	DC1 (device control 1)	49	31	061	€#49;	1	81	51	121	€#81;	Q	113	71	161	€#113;	q
18	12	022	DC2 (device control 2)	50	32	062	€#50;	2	82	52	122	€#82;	R	114	72	162	€#114;	r
19	13	023	DC3 (device control 3)	51	33	063	€#51;	3	83	53	123	€#83;	S	115	73	163	€#115;	s
20	14	024	DC4 (device control 4)	52	34	064	€#52;	4	84	54	124	€#84;	T	116	74	164	€#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	€#53;	5	85	55	125	€#85;	U	117	75	165	€#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	€#54;	6	86	56	126	€#86;	V	118	76	166	€#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	€#55;	7	87	57	127	€#87;	W	119	77	167	€#119;	w
24	18	030	CAN (cancel)	56	38	070	€#56;	8	88	58	130	€#88;	X	120	78	170	€#120;	x
25	19	031	EH (end of medium)	57	39	071	€#57;	9	89	59	131	€#89;	Y	121	79	171	€#121;	y
26	1A	032	SUB (substitute)	58	3A	072	€#58;	:	90	5A	132	€#90;	Z	122	7A	172	€#122;	z
27	1B	033	ESC (escape)	59	3B	073	€#59;	;	91	5B	133	€#91;	[123	7B	173	€#123;	{
28	1C	034	FS (file separator)	60	3C	074	€#60;	<	92	5C	134	€#92;	\	124	7C	174	€#124;	
29	1D	035	GS (group separator)	61	3D	075	€#61;	=	93	5D	135	€#93;]	125	7D	175	€#125;	}
30	1E	036	RS (record separator)	62	3E	076	€#62;	>	94	5E	136	€#94;	^	126	7E	176	€#126;	~
31	1F	037	US (unit separator)	63	3F	077	€#63;	?	95	5F	137	€#95;	_	127	7F	177	€#127;	DEL

Source: www.LookupTables.com

4. Handshake

Handshake

Handshake



Abbildung: wikipedia.org/wiki/Handshake

Motivation für zwei unterschiedliche Lösungen

Motivation für zwei unterschiedliche Lösungen

Hardware

Motivation für zwei unterschiedliche Lösungen

Hardware

- zuverlässige Umsetzung

Motivation für zwei unterschiedliche Lösungen

Hardware

- zuverlässige Umsetzung
- keine Selbstblockade möglich

Motivation für zwei unterschiedliche Lösungen

Hardware

- zuverlässige Umsetzung
- keine Selbstblockade möglich
- teuer und evtl. nicht realisierbar

Motivation für zwei unterschiedliche Lösungen

Hardware

Software

- zuverlässige Umsetzung
- keine Selbstblockade möglich
- teuer und evtl. nicht realisierbar

Motivation für zwei unterschiedliche Lösungen

Hardware

- zuverlässige Umsetzung
- keine Selbstblockade möglich
- teuer und evtl. nicht realisierbar

Software

- effiziente Umsetzung mit Code

Motivation für zwei unterschiedliche Lösungen

Hardware

- zuverlässige Umsetzung
- keine Selbstblockade möglich
- **teuer und evtl. nicht realisierbar**

Software

- effiziente Umsetzung mit Code
- keine zusätzlich Hardware nötig

Motivation für zwei unterschiedliche Lösungen

Hardware

- zuverlässige Umsetzung
- keine Selbstblockade möglich
- **teuer und evtl. nicht realisierbar**

Software

- effiziente Umsetzung mit Code
- keine zusätzlich Hardware nötig
- **keine hohe Zuverlässigkeit**

historische Lösung

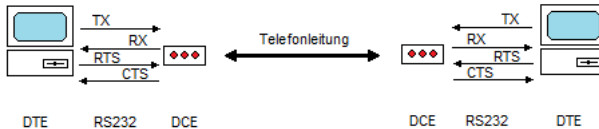


Abbildung: historischer Hardwarehandshake²

²mikrocontroller.net/articles/AVR-Tutorial:_UART

moderne Lösung

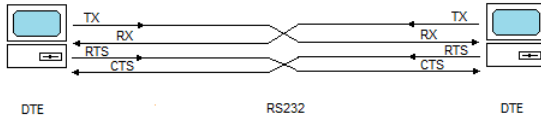


Abbildung: moderner Hardwarehandshake³

³mikrocontroller.net/articles/AVR-Tutorial:_UART

Beispielaufbau am ATmega16

Beispielaufbau am ATmega16

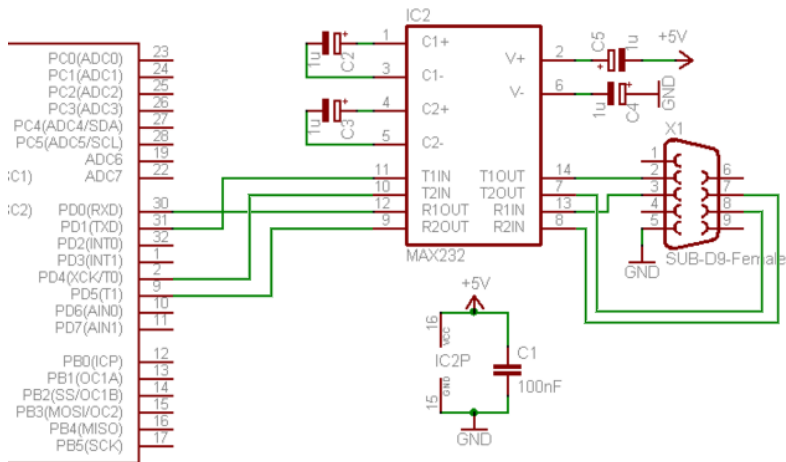


Abbildung: mikrocontroller.net/articles/AVR-Tutorial:_UART

Software Handshake - Generelle Idee

Software Handshake - Generelle Idee

- die Datenleitung selbst übernimmt die Flusskontrolle zwischen Sender/Empfänger

Software Handshake - Generelle Idee

- die Datenleitung selbst übernimmt die Flusskontrolle zwischen Sender/Empfänger
- beide Controller müssen exakt protokollieren, welcher Status anliegt (z.B. im Falle einer kurzfristigen Unterbrechung wichtig)

Software Handshake - Generelle Idee

- die Datenleitung selbst übernimmt die Flusskontrolle zwischen Sender/Empfänger
- beide Controller müssen exakt protokollieren, welcher Status anliegt (z.B. im Falle einer kurzfristigen Unterbrechung wichtig)
- XON (0x11) und XOFF (0x13) codieren Sende- und Empfangsbereitschaft (nicht zu verwechseln mit den Start/Stoppbits der Wörter!)

ASCII-Referenz

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	##32;	Space	64	40	100	##64;	@	96	60	140	##96;	`
1	1	001	SOH (start of heading)	33	21	041	##33;	!	65	41	101	##65;	A	97	61	141	##97;	a
2	2	002	STX (start of text)	34	22	042	##34;	"	66	42	102	##66;	B	98	62	142	##98;	b
3	3	003	ETX (end of text)	35	23	043	##35;	#	67	43	103	##67;	C	99	63	143	##99;	c
4	4	004	EOT (end of transmission)	36	24	044	##36;	\$	68	44	104	##68;	D	100	64	144	##100;	d
5	5	005	ENQ (enquiry)	37	25	045	##37;	%	69	45	105	##69;	E	101	65	145	##101;	e
6	6	006	ACK (acknowledge)	38	26	046	##38;	&	70	46	106	##70;	F	102	66	146	##102;	f
7	7	007	BEL (bell)	39	27	047	##39;	'	71	47	107	##71;	G	103	67	147	##103;	g
8	8	010	BS (backspace)	40	28	050	##40;	(72	48	110	##72;	H	104	68	150	##104;	h
9	9	011	TAB (horizontal tab)	41	29	051	##41;)	73	49	111	##73;	I	105	69	151	##105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	##42;	*	74	4A	112	##74;	J	106	6A	152	##106;	j
11	B	013	VT (vertical tab)	43	2B	053	##43;	+	75	4B	113	##75;	K	107	6B	153	##107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	##44;	,	76	4C	114	##76;	L	108	6C	154	##108;	l
13	D	015	CR (carriage return)	45	2D	055	##45;	-	77	4D	115	##77;	M	109	6D	155	##109;	m
14	E	016	SO (shift out)	46	2E	056	##46;	.	78	4E	116	##78;	N	110	6E	156	##110;	n
15	F	017	SI (shift in)	47	2F	057	##47;	/	79	4F	117	##79;	O	111	6F	157	##111;	o
16	10	020	DLE (data link escape)	48	30	060	##48;	0	80	50	120	##80;	P	112	70	160	##112;	p
17	11	021	DC1 (device control 1)	49	31	061	##49;	1	81	51	121	##81;	Q	113	71	161	##113;	q
18	12	022	DC2 (device control 2)	50	32	062	##50;	2	82	52	122	##82;	R	114	72	162	##114;	r
19	13	023	DC3 (device control 3)	51	33	063	##51;	3	83	53	123	##83;	S	115	73	163	##115;	s
20	14	024	DC4 (device control 4)	52	34	064	##52;	4	84	54	124	##84;	T	116	74	164	##116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	##53;	5	85	55	125	##85;	U	117	75	165	##117;	u
22	16	026	SYN (synchronous idle)	54	36	066	##54;	6	86	56	126	##86;	V	118	76	166	##118;	v
23	17	027	ETB (end of trans. block)	55	37	067	##55;	7	87	57	127	##87;	W	119	77	167	##119;	w
24	18	030	CAN (cancel)	56	38	070	##56;	8	88	58	130	##88;	X	120	78	170	##120;	x
25	19	031	EH (end of medium)	57	39	071	##57;	9	89	59	131	##89;	Y	121	79	171	##121;	y
26	1A	032	SUB (substitute)	58	3A	072	##58;	:	90	5A	132	##90;	Z	122	7A	172	##122;	z
27	1B	033	ESC (escape)	59	3B	073	##59;	;	91	5B	133	##91;	[123	7B	173	##123;	{
28	1C	034	FS (file separator)	60	3C	074	##60;	<	92	5C	134	##92;	\	124	7C	174	##124;	
29	1D	035	GS (group separator)	61	3D	075	##61;	=	93	5D	135	##93;]	125	7D	175	##125;	}
30	1E	036	RS (record separator)	62	3E	076	##62;	>	94	5E	136	##94;	^	126	7E	176	##126;	~
31	1F	037	US (unit separator)	63	3F	077	##63;	?	95	5F	137	##95;	_	127	7F	177	##127;	DEL

Source: www.LookupTables.com

Nachteile - Deadlock

Nachteile - Deadlock

- Zeitverzögerung in der Kommunikation durch die Datenleitung kann bei Kommunikationsabbruch zu Datenverlust führen

Nachteile - Deadlock

- Zeitverzögerung in der Kommunikation durch die Datenleitung kann bei Kommunikationsabbruch zu Datenverlust führen
- beide Seiten können sich mit XOFF endlos blockieren (Deadlock)

Tipps und Hilfestellungen

- mikrocontroller.net/UART
- Baudratenrechner im Internet

Baudratenrechner Internetbeispiel 1

www.wormfood.net/avrbaudcalc.php

WormFood's AVR Baud Rate Calculator

- Bit Rate Show table of all possible clock speeds, for a given bit rate
- Clock Freq. in Mhz Show table of all possible bit rates, for a given clock speed
- show additional synchronous mode information
- Show additional double speed mode information (U2X=1)
- Consider only 8-bit UBRR values as being valid
- Hide tables except for user supplied clock rate

Make Tables

32.768 KHz				1.8432 Mhz				2 Mhz				2.4576 Mhz				3.072 Mhz				3.2768 Mhz			
Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error
300	0	0x0006	2.5	300	193	0x017F	0.0	300	110	0x0149	0.0	300	511	0x01FF	0.0	300	839	0x027F	0.0	300	862	0x02AA	0.0
600	0	0x0003	12.5	600	191	0x018F	0.0	600	207	0x014F	0.0	600	255	0x01FF	0.0	600	319	0x011F	0.0	600	340	0x0114	0.0
1200	0	0x0001	17.5	1200	95	0x005F	0.0	1200	103	0x0067	0.0	1200	127	0x007F	0.0	1200	159	0x009F	0.0	1200	170	0x00AA	0.0
2400	0	0x0000	17.5	2400	47	0x002F	0.0	2400	51	0x0033	0.0	2400	63	0x003F	0.0	2400	79	0x004F	0.0	2400	84	0x0054	0.0
4800	0	0x0000	134.4	4800	23	0x0017	0.0	4800	25	0x001B	0.0	4800	31	0x001F	0.0	4800	39	0x0027	0.0	4800	42	0x002A	0.0
9600	0	0x0000	368.8	9600	11	0x000B	0.0	9600	12	0x000C	0.0	9600	15	0x000F	0.0	9600	19	0x0013	0.0	9600	20	0x0014	1.0
14400	0	0x0000	603.2	14400	7	0x0007	0.0	14400	8	0x0008	3.7	14400	10	0x000A	3.1	14400	12	0x000C	2.5	14400	13	0x000D	1.0
19200	0	0x0000	837.6	19200	5	0x0005	0.0	19200	5	0x0006	7.3	19200	7	0x0007	0.0	19200	9	0x000B	0.0	19200	10	0x000A	3.1
28800	0	0x0000	1306.8	28800	3	0x0003	0.0	28800	3	0x0003	7.3	28800	4	0x0004	6.4	28800	6	0x0006	5.4	28800	6	0x0006	1.0
57600	0	0x0000	1775.2	57600	2	0x0002	0.0	57600	2	0x0002	7.3	57600	3	0x0003	0.0	57600	4	0x0004	0.0	57600	5	0x0005	6.2
86400	0	0x0000	2712.8	86400	1	0x0001	0.0	86400	1	0x0001	7.3	86400	2	0x0002	13.2	86400	3	0x0003	10.4	86400	3	0x0003	13.2
96000	0	0x0000	3650.4	96000	1	0x0001	32.3	96000	1	0x0001	22.8	96000	1	0x0001	0.0	96000	2	0x0002	20.4	96000	2	0x0002	13.2
115200	0	0x0000	5325.6	115200	0	0x0000	0.0	115200	0	0x0000	7.3	115200	0	0x0000	23.4	115200	0	0x0000	20.4	115200	0	0x0000	13.2
230400	0	0x0000	11350.8	230400	0	0x0000	100.0	230400	0	0x0000	49.3	230400	0	0x0000	39.4	230400	0	0x0000	20.4	230400	0	0x0000	13.2
350000	0	0x0000	21607.6	350000	0	0x0000	133.4	350000	0	0x0000	100.0	350000	0	0x0000	82.4	350000	0	0x0000	30.4	350000	0	0x0000	22.4
3.579545 Mhz				3.6864 Mhz				3.93216 Mhz				4 Mhz				4.194304 Mhz				4.433619 Mhz			
Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error	Baud Rate	UBRR (dec)	UBRR (hex)	% of error
300	745	0x02E9	0.0	300	797	0x031F	0.0	300	818	0x0332	0.0	300	831	0x0340	0.0	300	871	0x0369	0.0	300	923	0x0369	0.0
600	372	0x0174	0.0	600	398	0x017E	0.0	600	409	0x0189	0.0	600	416	0x01A0	0.0	600	434	0x01B4	0.0	600	461	0x01C3	0.0

Baudratenrechner Internetbeispiel 2

← → C f www.gjlay.de/helferlein/avr-uart-rechner.html



AVR ATmega Baudraten-Rechner

Mit dem folgenden Rechner kann der Wert für das UBRR-Register eines AVR ATmega berechnet werden. Anzugeben ist die Frequenz f_{osc} , mit der der ATmega betrieben wird.

Für ungerade Taktraten führt Abrunden/Aufrunden/kaufmännisches Runden nicht immer zum kleinsten Fehler hin. Dieser Rechner rundet zum kleinsten Fehler hin.

UBRR-Einstellungen, welche eine Baudrate mit einem relativen Fehler ergeben, der betragsmässig nicht über der angegebenen Prozent-Grenze liegt, werden fett gedruckt.

JavaScript muss aktiviert sein.

f_{osc} = MHz max. Fehler = %

1 MHz	2 MHz	2.4576 MHz	3.6864 MHz
4 MHz	4.192 MHz	4.194304 MHz	7.3728 MHz
8 MHz	8.192 MHz	9.216 MHz	9.8304 MHz
10 MHz	12 MHz	11.0592 MHz	14.7456 MHz
16 MHz	18 MHz	18.432 MHz	
20 MHz	22.1184 MHz	24 MHz	24.576 MHz

Abkürzungsverzeichnis

DCE	Data Carrier Equipment
DTE	Data Terminal Equipment
CTS	Clear to Send
RTS	Request to Send
RXD	receive Data
TXD	transmit Data
UART	Universal Asynchronous Receiver and Transmitter
SPI	serial peripheral interface
ISP	in system programming
SCK	serial clock

Quellen

- Inhaltsquellen:

- 1 mikrocontroller.net/articles/AVR-Tutorial:_UART
- 2 mikrocontroller.net/articles/Baudratenquarz
- 3 wikipedia.org/wiki/Émile_Baudot

- Bildquellen:

- 1 Emile-Baudot: wikipedia.org/wiki/Émile_Baudot
- 2 mikrocontroller.net/articles/AVR-Tutorial:_UART
- 3 ASCII-Tabelle: asciitable.com
- 4 Handshake: wikipedia.org/wiki/Handshake

Danke fürs Zuhören! :)