UART

Zenit Music
13.05.2013

Inhalt

(1) Allgemeines
(2) Vorwissen(?)
(3) Funktionsweise des UART
(4) UART-RS232
(5) UART beim ATMEGA 32

Allgemeines

Allgemeines

- UART: Universal Asynchronus Reciever and Transmitter

Allgemeines

- UART: Universal Asynchronus Reciever and Transmitter
- Elektronischer Baustein

Allgemeines

- UART: Universal Asynchronus Reciever and Transmitter
- Elektronischer Baustein
- Realisierung von digitalen Schnittstellen

Allgemeines

- UART: Universal Asynchronus Reciever and Transmitter
- Elektronischer Baustein
- Realisierung von digitalen Schnittstellen
- Eigenständiges Bauelement oder Bestandteil eines anderen Bauteils

Allgemeines

- UART: Universal Asynchronus Reciever and Transmitter
- Elektronischer Baustein
- Realisierung von digitalen Schnittstellen
- Eigenständiges Bauelement oder Bestandteil eines anderen Bauteils
- Erster UART-Chip 1971

Allgemeines

- UART: Universal Asynchronus Reciever and Transmitter
- Elektronischer Baustein
- Realisierung von digitalen Schnittstellen
- Eigenständiges Bauelement oder Bestandteil eines anderen Bauteils
- Erster UART-Chip 1971
- Erste UARTs wenige 100Bit/s, heute mehrere Megabit/s

Vorwissen(?)

Vorwissen(?)

- Synchrone Datenübertragung:

Vorwissen(?)

- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung

Vorwissen(?)

- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung
- Asynchrone Datenübertragung:

Vorwissen(?)

- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung
- Asynchrone Datenübertragung:
  - Zeichen werden asynchron übertragen
  - Kein Taktsignal
  - Rahmen nötig

Vorwissen(?)

- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung
- Asynchrone Datenübertragung:
  - Zeichen werden asynchron übertragen
  - Kein Taktsignal
  - Rahmen nötig
- Baudrate:

Vorwissen(?)

- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung
- Asynchrone Datenübertragung:
  - Zeichen werden asynchron übertragen
  - Kein Taktsignal
  - Rahmen nötig
- Baudrate:
  - Symbole (Bit) pro Sekunde

Vorwissen(?)

- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung
- Asynchrone Datenübertragung:
  - Zeichen werden asynchron übertragen
  - Kein Taktsignal
  - Rahmen nötig
- Baudrate:
  - Symbole (Bit) pro Sekunde
- Vollduplex

Vorwissen(?)

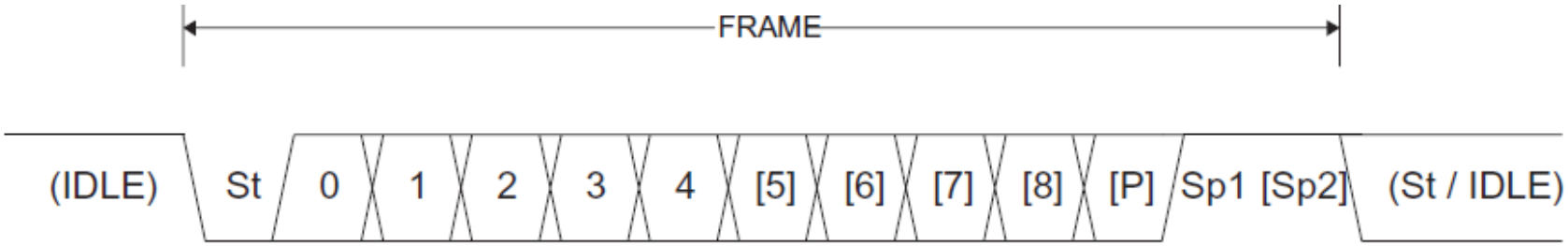
- Synchrone Datenübertragung:
  - Sender/Empfänger haben gleichen Takt
  - Zeitlich synchronisiert
  - Extra Taktleitung oder Taktrückgewinnung
- Asynchrone Datenübertragung:
  - Zeichen werden asynchron übertragen
  - Kein Taktsignal
  - Rahmen nötig
- Baudrate:
  - Symbole (Bit) pro Sekunde
- Vollduplex
  - Lesen/schreiben gleichzeitig

Funktionsweise des UART

Hardwarekomponenten

- Clock-generator
- I/O Schieberegister
- R/W Kontrolllogik
- Übertragungs-/Empfangspuffer
- FIFO-Puffer-Speicher

Rahmen-Format



Baudrate

- Die Baudrate muss eingestellt werden, da entscheidend für Empfänger
- ATMEGA: 2400bps bis 2,5Mbps (20MHz)

Bitrate	Bitdauer	9.600 bit/s	104 µs
50 bit/s	20,0 ms	19.200 bit/s	52,1 µs
110 bit/s	9,09 ms	38.400 bit/s	26,0 µs
150 bit/s	6,67 ms	57.600 bit/s	17,4 µs
300 bit/s	3,33 ms	115.200 bit/s	8,68 µs
1.200 bit/s	833 µs	230.400 bit/s	4,34 µs
2.400 bit/s	417 µs	460.800 bit/s	2,17 µs
4.800 bit/s	208 µs	500.000 bit/s	2,00 µs

Empfänger

Interne Clock: Abfrage bei jedem Clock-Pulse

Signal halbe Bitdauer low → Start

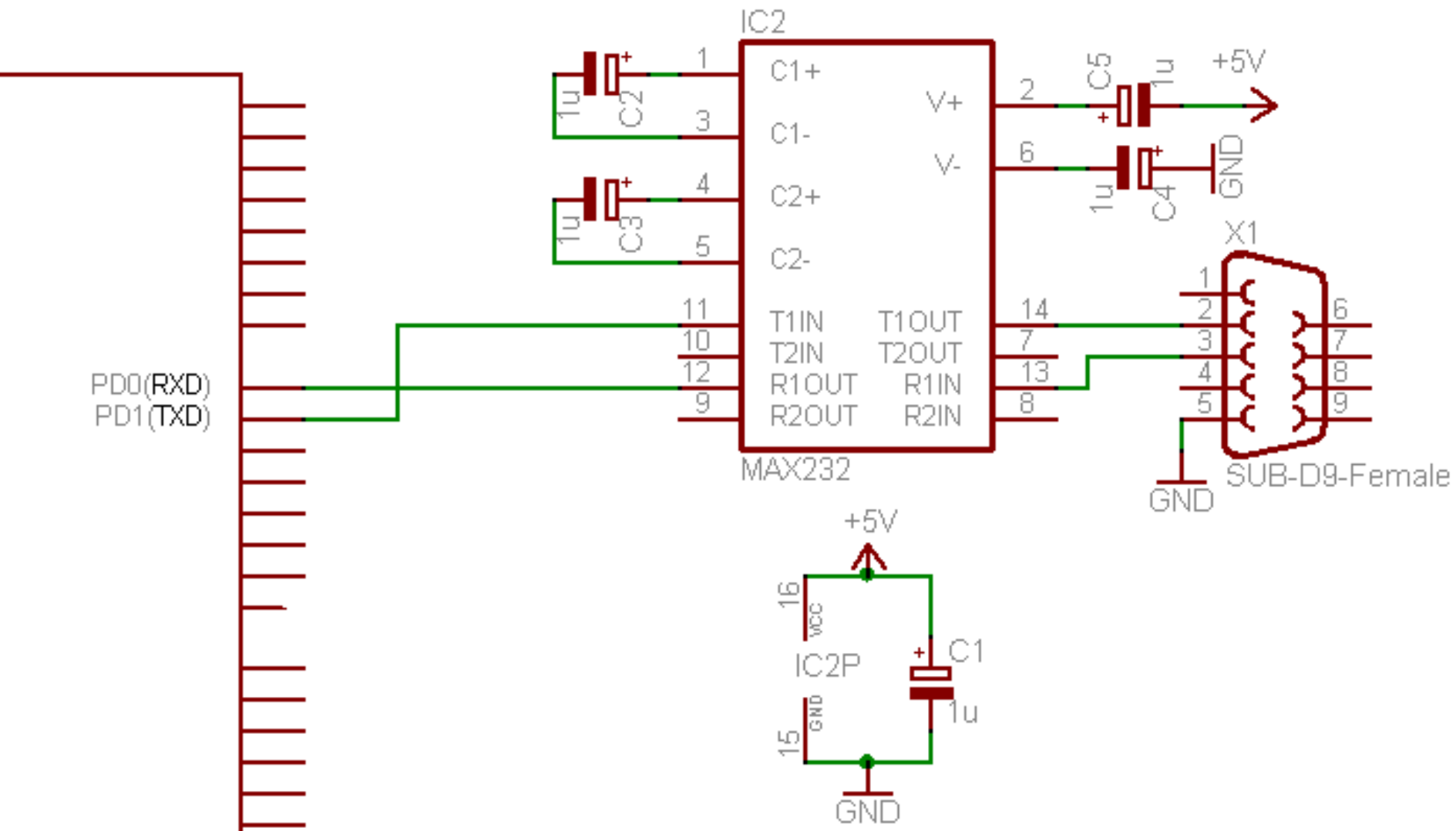
Warte eine Bitdauer → Abfrage (n-mal)

n-mal abgefragt →
Weiterleitung/Warten auf neues Startbit

Sender

- Wenn Daten in Schieberegister wird gesendet
- Busy-Status-Flag

UART→RS232


IC2
C1+
V+
C1-
V-
C2+
C2-
T1IN
T1OUT
T2IN
T2OUT
R1OUT
R1IN
R2OUT
R2IN
MAX232
C2
C3
C4
C5
1u
+5V
GND
X1
SUB-D9-Female
PD0(RXD)
PD1(TXD)
IC2P
C1
vcc
GND

UART beim ATMEGA

Register

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Register

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Beispiel:Initialisierung

```
void USART_Init( unsigned int baud )
{
  /* Set baud rate */
  UBRRH = (unsigned char)(baud>>8);
  UBRRL = (unsigned char)baud;
  /* Enable receiver and transmitter */
  UCSRB = (1<<RXEN)|(1<<TXEN);
  /* Set frame format: 8data, 2stop bit */
  UCSRC = (1<<URSEL)|(1<<USBS)|(3<<UCSZ0);
}
```

Quellen

- http://www.mikrocontroller.net/articles/AVR-Tutorial:_UART
- http://www.rn-wissen.de/index.php/UART
- http://www.physik.uni-regensburg.de/studium/edverg/elfort/C_KURS_Atmel_Programmieren%20htm/Der_UART_1.htm
- http://www.itwissen.info/definition/lexikon/universal-asynchronous-receiver-transmitter-UART.html
- http://www.elektronik-magazin.de/page/der-pegelumsetzer-max232-15


VIA 9GAG.COM