

Projektlabor

USART

Gliederung

1. Allgemeines
 - Definition
2. Funktionsweise
3. Beispiel am ATmega 16
4. Fazit
5. Quellen

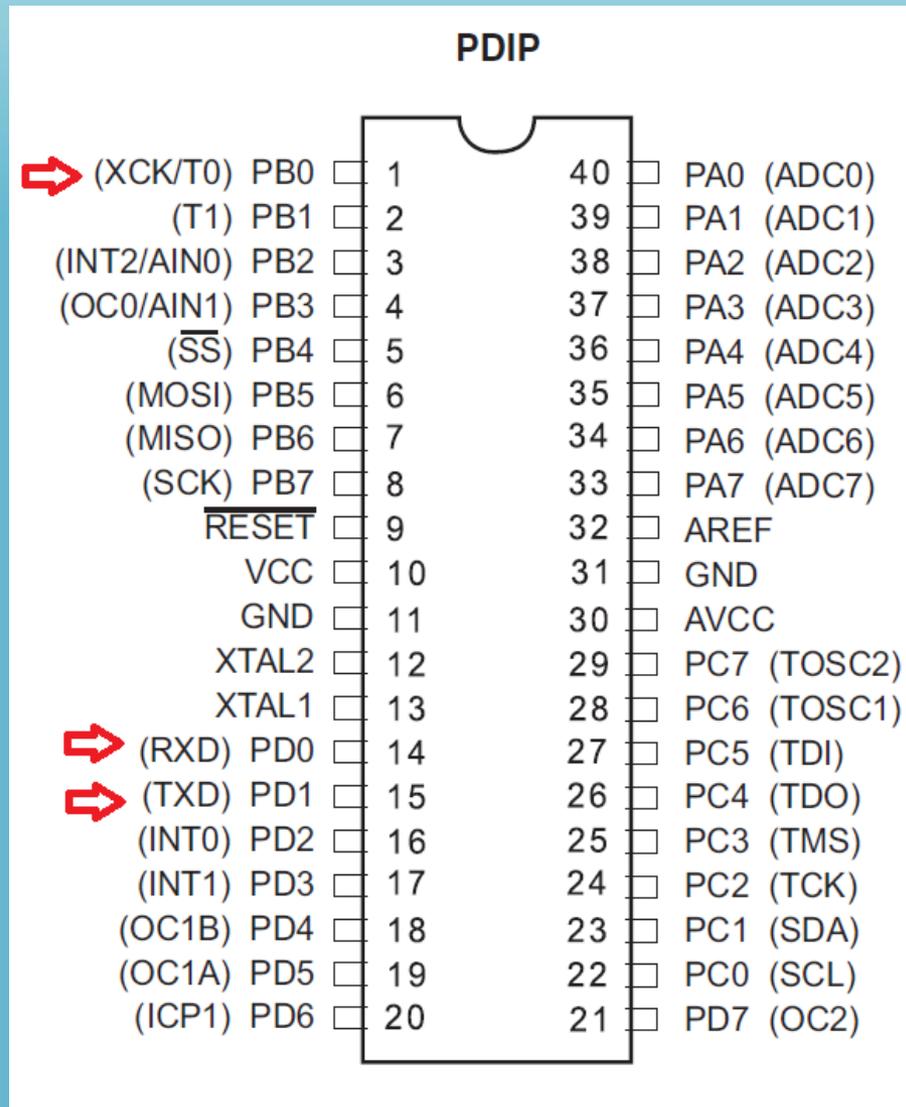
1. Allgemeines

- Was ist USART?
 - Schnittstelle die synchrone / asynchrone Kommunikation zwischen Mikrocontrollern und PCs erlaubt
 - „**U**niversal **S**ynchronus and **A**synchronus Serial **R**eceiver and **T**ransmitter“
 - Datenpaketgröße 5-9 Bit + Start-, Parity- und Endbit

2. Funktionsweise

- Benötigte Bauteile zum Senden, Empfangen und Synchronisieren
 - Senden PD0 TxD („Transmit-Data“)
 - Empfangen PD1 RxD („Receive-Data“)
 - Synchronisieren PBO XCK (gemeinsame Taktfrequenz)

2. Funktionsweise



2. Funktionsweise

- Asynchrone Übertragung
 - Ohne Taktleitung
 - „Synchronisierung“ erfolgt durch Start- und Stopbit, sowie der eingestellten Baudrate (Symbole pro Sekunde)

2. Funktionsweise

- Synchrone(Master/Slave) Übertragung
 - Mit Taktleitung
 - „Synchronisierung“ erfolgt durch Start- und Stopbit, der eingestellten Baudrate(Mastermode) und der Taktleitung

3. Beispiel am Atmega16

- Bit-Setzung in Registern
 - **UBRR**(USART Baud Rate Register)
 - Bestimmt bei welcher Baudrate, im normalen asynchronen Modus übertragen wird
 - Double Speed Mode im Asynchronen Betrieb
 - Durch setzen von U2X in Register **UCSRA**(USART Control and Status Register A)

3. Beispiel am Atmega16

- Baud-, UBRR-Berechnung

Operating Mode	Equation for Calculating Baud Rate	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

3. Beispiel am Atmega16

- Baudratenberechnung Beispiel

Double Speed Mode

$F_{osc} = 20 \text{ MHz}$

UBRR = 4



$$\text{Baudrate} = \frac{20\text{MHz}}{8 \cdot (4+1)}$$

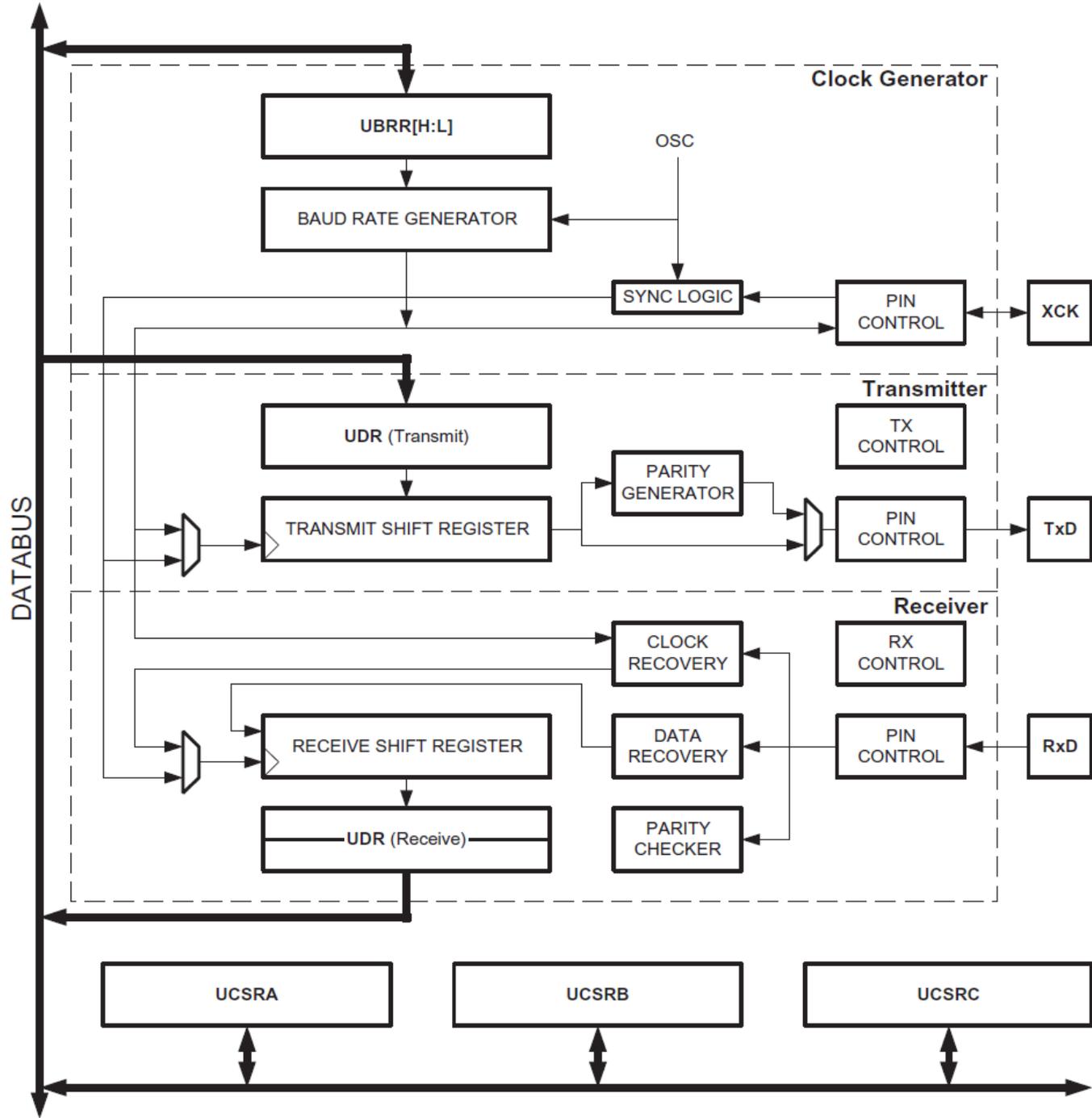
$$\text{Baudrate} = 500.000 \text{ bit/s}$$

3. Beispiel am Atmega16

- Bit-Setzung in Registern
 - Synchroner Modus wird aktiviert wenn **UMSEL** (USART Modus Select) auf „1“ gesetzt ist
 - XCK als Eingang(Slave), oder Ausgang(Master)

3. Beispiel am Atmega16

- Der USART Transmitter
 - Wird aktiviert wenn TXEN (Transmit Enable) in UCSRB (USART Control and Status Register B) gesetzt wird
 - Durch Schreiben eines Bytes in UDR (USART Data Register) wird dieses Byte über den USART ausgegeben
- Der USART Receiver
 - Wird aktiviert wenn RXEN(Receive Enable) in UCSRB (USART Control and Status Register B) gesetzt wird
 - Empfängt der USART Daten, so wird das RXC-Flag gesetzt



4. Fazit

- Vergleich SPI und USART
 - SPI ist immer synchron, USART nicht zwingend
 - SPI als auch USART benötigen drei Leitungen zur Kommunikation
 - SPI ist ein Kommunikationsprotokoll, USART hingegen Hardware mit keinem fest definiertem Protokoll
- Eignet sich gut zum Debuggen, da ständig Statusmeldungen an den Rechner gesendet werden
- USART ist einfach zu benutzen, da die Hardware vieles selbständig macht
- hohe Datenraten

5. Quellen

- Mikrocontroller.net
[http://www.mikrocontroller.net/articles/AVR-Tutorial: UART](http://www.mikrocontroller.net/articles/AVR-Tutorial:_UART)
<http://www.mikrocontroller.net/articles/UART>
<http://www.mikrocontroller.net/mc-project/Pages/AVR/USART/usart.html>
- Tschallener.net <http://www.tschallener.net/AVR/USART1.pdf>
- Wikipedia.org
[http://de.wikipedia.org/wiki/Universal Asynchronous Receiver Transmitter](http://de.wikipedia.org/wiki/Universal_Asynchronous_Receiver_Transmitter)
- Hsg-kl.de <http://www.hsg-kl.de/faecher/inf/netze/material/bitfuerbit.html>
- Rn-wissen.de <http://www.rn-wissen.de/index.php/UART>
- Datenblatt des ATmega 16

VIELEN DANK FÜR DIE AUFMERKSAMKEIT

Kritik ?