

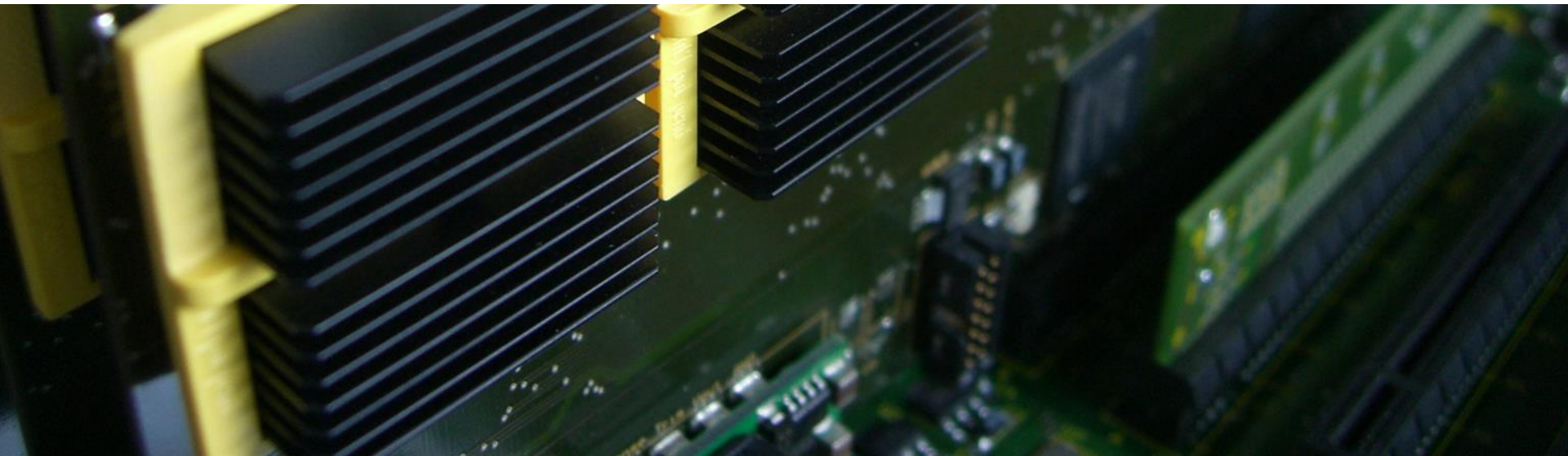
Eingebettete Prozessoren

SS 2013

Übung 13: Serielle Schnittstelle und Ausblick

Dipl.-Ing. Thomas Pöppelmann
Arbeitsgruppe Sichere Hardware
Horst Görtz Institut für IT-Sicherheit

18.07.2013



Agenda

- 1. Serielle Schnittstelle**
- 2. Sicherungssysteme**
- 3. Motivation und Ausblick**

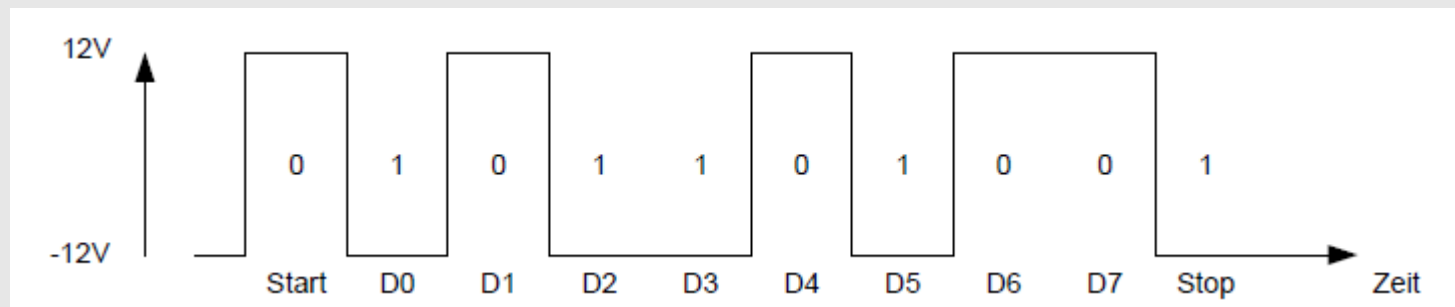
1. Serielle Schnittstelle

Ziel

- Ziel ist eine bidirektionale Kommunikationsverbindung zwischen dem Asuro/AVR und dem PC herzustellen
 - Übertragen von Kommandos
 - Senden von Statusmeldungen
- Serielle Schnittstelle ist einfach zu benutzen und in vielen Geräten verbaut
 - Früher in der Telekommunikation benutzt (in den 1960er Jahren erarbeitet). Viele Steuersignale werden heute nicht mehr benötigt.

Serielle Schnittstelle

- RS232 ist eine verbreitete, asynchrone (kein zusätzliches Taktsignal), bidirektional Schnittstelle
- Pegelorientiert mit positiven und negativen Spannungen
- Sendeleitung *TxD* , Empfangsleitung *RxD* und Masse *GND*
- Optional: *CTS/RTS* für Flusskontrolle
- Einbettung der Nutzdaten erfolgt in Frames
- Separate Hardwareeinheit (Kapitel 19. USART) für serielle Kommunikation – Beispiel im Skript/Blackboard
- USART = Universal Synchronous/Asynchronous Receiver Transmitter



Startbit

Daten: 0x2D= 00101101

Stopbit

Parallel vs. Seriell

- Serielle Schnittstelle: Ein Bit wird gleichzeitig übertragen
- Parallele Schnittstelle: Mehrere Bits werden gleichzeitig über mehrere Leitungen übertragen
- Vorteile Serielle Schnittstelle:
 - Es sind nur wenige Schnittstellenpins/Leitungen erforderlich.
 - Kein Übersprechen wie bei der parallelen Übertragung durch lokal nah zusammenliegende Busleitungen.
 - Keine unterschiedlichen Laufzeitverzögerungen zwischen den einzelnen Leitungen eines parallelen Bus (bus skew).
 - Durch Elimination des Bus Skews sind insgesamt höhere Taktfrequenzen auf der einzelnen Leitung möglich, wodurch sich höhere Übertragungsraten ergeben.
- Nachteile:
 - Höherer Hardware und Protokollaufwand durch die serielle, zeitlich nacheinander erfolgende Kodierung der Bits auf einer Leitung.
 - Synchronisationsaufwand

Konfiguration USART

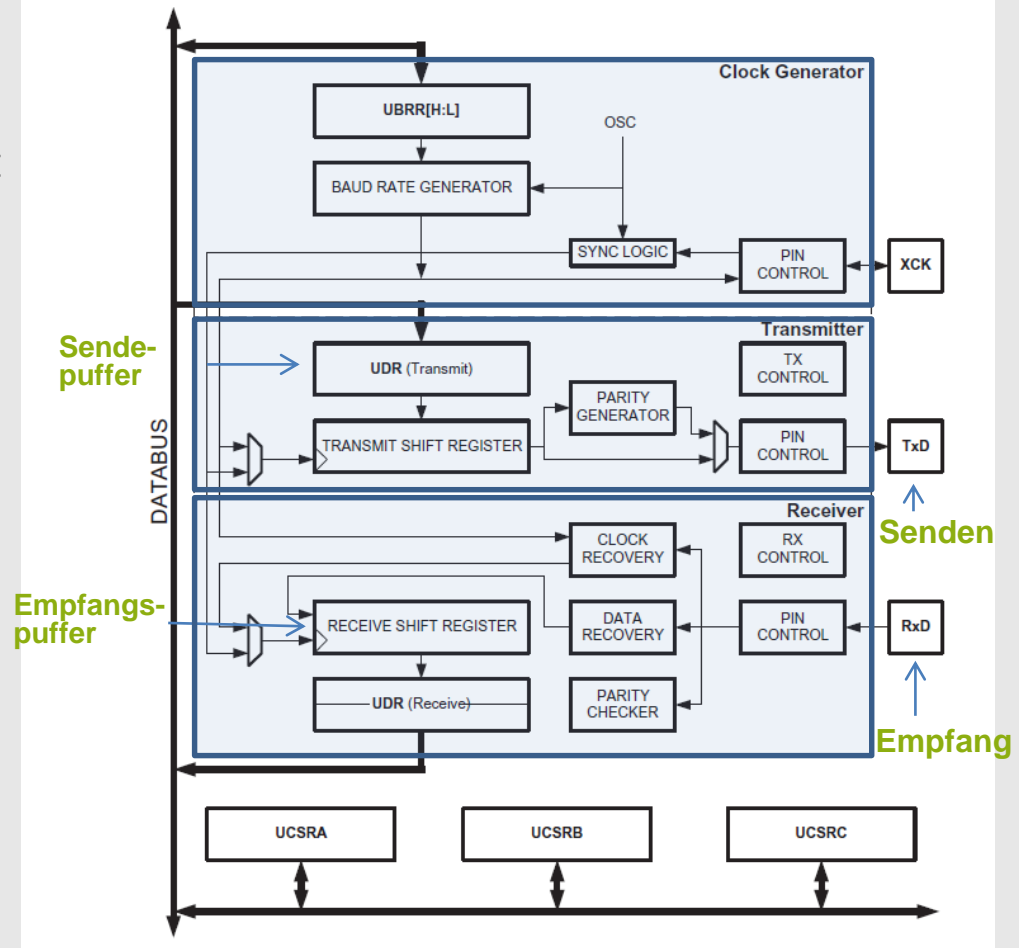
- Eigenschaften des USART im AVR μ C ermöglicht synchrone und asynchrone Übertragung
- Konfiguration der Baudrate
 - Baudrate: Einheit für die Symbolrate. 1 Baud ist die Geschwindigkeit, wenn 1 Symbol pro Sekunde übertragen wird
 - Bitrate: Menge an übertragenen Daten je Zeiteinheit in Bit je Sekunde. Nicht jedes Symbol wird zum Übertragen eines Datenbits verwendet (z.B. Parität)
 - Einstellung der Baudrate über das 16 Bit UBBR-Register
 - $$\text{UBBR} = \left\lfloor \frac{f_{\text{CPU}}}{16 * \text{Baudrate}} - 1 \right\rfloor$$
 - Beispiel: Bei $f_{\text{CPU}} = 8 \text{ MHz}$ und 2400 Baud/s , setze $\text{UBBR} = 207$
- Konfiguration des Frameformats mit Hilfe des Kontrollregisters UCSR

Aufbau USART

Drei Blöcke:

- Clock Generator
 - Write buffer, serial Shift Register, Parity Generator and logic for frame formats
- Transmitter
 - Clock recovery, parity checker, control logic, Shift Register und two level receive buffer (UDR)
- Receiver
 - Clock recovery, parity checker, control logic, Shift Register und two level receive buffer (UDR)
- Kontrollregister werden zwischen den Blöcken geteilt

Figure 19-1. USART Block Diagram⁽¹⁾



USART initialisieren

- USART initialisieren
 - Baudrate (hier in r17:r16)
 - Sender und Empfänger aktivieren
 - Frameformat

USART_Init:

; Setze baud rate

out UBRRH, r17

out UBRRL, r16

; Aktiviere Empfänger und Sender

ldi r16, (1<<RXEN)|(1<<TXEN)

out UCSRB, r16

; Setze Frameformat (8 Datenbits, kein Paritätsbit, 1 Stopbit)

LDI r16, (1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0)

OUT UCSRC, r16

Bit senden

- Sender aktivieren
- Übertragungsstatus löschen
- Warten bis der Sendepuffer leer ist
- Daten senden

```
.MACRO UART_SEND_REG
    SBI UCSRB, TXEN                ; aktiviere den Sender
    SBI UCSRA, TXC                ; lösche den Übertragungsstatus

    WAIT_EMPTY_BUFFER:           ; warte, bis der Sendepuffer wieder leer ist
        SBIS UCSRA, UDRE
        RJMP WAIT_EMPTY_BUFFER

    OUT UDR, @0                  ; Daten senden durch Laden des Sendepuffers
    WAIT_TRANSMIT_COMPLETE:      ; warte, bis die aktuelle Sendeoperation abgeschlossen ist
        SBIS UCSRA, TXC
        RJMP WAIT_TRANSMIT_COMPLETE

    CBI UCSRB, TXEN              ; deaktiviere den Sender
.ENDMACRO
```

Asuro spezifisches zum USART

- IR Diode benötigt eine 36kHz Trägerfrequenz die mit Timer2 erzeugt werden kann. Infos z.B. unter:
 - <http://www.roboternetz.de/community/threads/45551-Verst%C3%A4ndnisproblem-beim-IRCollisionTest-Testprogramm?p=436695#post436695>
- Komplettes Beispiel im Skript und Blackboard

```

; ##### ISR Routine
TIMER2_OVF:
    LDI TEMP, 0x25 ; setze den Timer vor, um den duty-cycle von 50% zu erhalten (ca. 111 Taktzyklen OC2=0)
    OUT TCNT2, TEMP
    RETI

[...]

; ##### Konfiguration
LDI TEMP, (1<<IR_TX)          ; IR-TX als Ausgang für das 36 kHz-Trägersignal der IR-Diode einstellen
OUT DDRB, TEMP
LDI TEMP, (1<<TOIE2)          ; Interrupt des Timer2 (als Signalgenerator des 36 kHz-Signals) aktivieren
OUT TIMSK, TEMP
; Timer 2 aktivieren (Fast PWM, OC2 bei Match setzen, bei maximalem Wert löschen, PRESCALER CLK/1 setzen => Einstellung CS02:CS01:CS00 = 001)
LDI TEMP, (1<<WGM21) | (1<<WGM20) | (1<<COM20) | (1<<COM21) | (1<<CS20)
OUT TCCR2, TEMP

; Einstellung des Timer 2 zur Erzeugung der Frequenz f=36kHz (setze OC2 für 111 Takte t_ein=111*125ns=13,875us Periode T mit 50% duty cycle:
; T=2*t_ein=27,750us, f=1/T=36,036 kHz)
LDI TEMP, 0x91
OUT OCR2, TEMP

```

2. Sicherungssysteme

Sicherungssysteme

- Aktives System Monitoring umfasst
 - Überwachung der Temperatur des Prozessors oder des Gesamtsystems
 - Z.B. mittels eines Temperatursensors/ADC
 - Überwachung der Energieversorgung (Brown-Out Detection)
 - Nutzen eines ADCs bzw. Komparators, der die Eingangsspannung mit einer nicht zu unterschreitenden Referenzspannung vergleicht
 - Wichtig bei batteriebetrieben oder kritischen Systemen
 - Manipulationsschutz
 - Erkennen von Manipulationen am System z.B. durch Sicherungsschalter
 - Schutz gegen Systemabstürze
 - Watch-Dog Timer

Watchdog Timer - Sicherungssysteme

- Watch-Dog Timer (WDT): Totmannschaltung, die Softwareabstürze und Deadlocks eines Programmes in Hardware überwachen kann
- Programm muss die ordnungsgemäße Funktionsweise in regelmäßigen Abständen dem Prozessor bestätigen
- Realisiert als unabhängiger, zusätzlicher 10 Bit Zeitgeber
 - Einstellbarer Vorteiler
 - Bei Überlauf erfolgt automatischer Reset des Prozessors
- WDT muss regelmäßig zurückgesetzt werden
 - Kann zu Problemen führen, wenn der WDT nicht korrekt zurückgesetzt wird

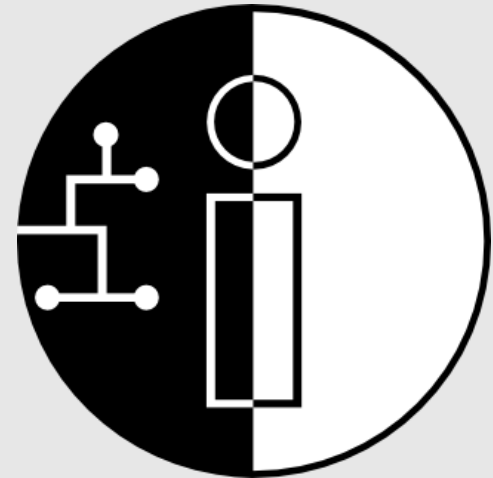
3. Motivation und Ausblick

Motivation

- Eingebettete Prozessoren ist nur eine einsemestrige Vorlesung
- Einige Themen haben wir nicht oder nur sehr kurz behandelt
- Grundlagenkenntnisse aus der Vorlesung sollten ausreichen um sich selbständig in die Thematik weiter einzuarbeiten (bei Interesse)
- Auf den nächsten Folien: Tipps und Ausblick

Basteln – Das Labor

- Verein in Bochum mit eigenen Räumen
- Raum zum experimentieren, diskutieren und zur Projektverwirklichung in angenehmer Atmosphäre
- Viele uC und FPGA Projekte (z.B. AVR-Crypto Lib)
- <http://www.das-labor.org/>



Basteln – Evaluierung

- Der Asuro ist ein recht umfangreiches Evaluationsboard
 - Kann erweitert werden (Ultraschall)
- Es existieren viele Evaluationsboard für interessantere und umfangreichere Projekte (Preis 20€-1000€)
- Sehr preiswerte Angebote sind z.B. das AVR-NETIO und das AVR-Funkmodul von Pollin
 - Ermöglicht Funkkommunikation oder den AVR als Web Server zu benutzen
 - Es muss etwas gebastelt und evtl. auch gelötet werden



Basteln – Evaluierung

- Arduino-Plattform
 - Einfaches realisieren von uC Projekten
- Raspberry Pi
 - ARM basiertes Entwicklungsboard
 - Mächtiger Prozessor
 - Sehr günstig
 - Anwendung als Embedded Webserver, Multimedia Server



Basteln – Ressourcen im Netz

- Mikrocontroller.net
 - Großes Forum
 - Viele Infos zum AVR + Beispielcode
- roboternetz.de
 - Großes Forum
 - Infos zum Roboterbau (z.B. auch Asuro)
- Fingers-welt.de (nicht nachmachen)
 - Selbstbau Röntgengeräte, Käsefräsen, Kaffeemaschinen, alle mit einem uC als Basis

Basteln - Beispiel

- Waschmaschinenüberwachung
 - In Bochumer Wohnheim umgesetzt
 - AVR-Board mit Netzwerkbuchse
 - Stromzange mit Analogausgang angeschlossen an den uC
 - Webserver auf dem AVR zeigt aktuellen Stromverbrauch an
 - Programm auf einem “richtigen” Webserver wertet die Daten aus und zeigt den Stromverbrauch der vergangen Stunde an

Nie wieder in den Keller
laufen wenn die
Waschmaschine noch
nicht fertig oder voll ist!



www.phdcomics.com

Studieren/Forschen/Arbeiten

- Lehrveranstaltungen mit uC Bezug an der RUB (Auszug)
 - Praktikum Embedded Smartcard Microcontrollers (BSc/MSc)
 - Praktikum Mess- und Regelschaltungen mit Mikrocontrollern (MSc)
 - Praktikum Seitenkanalangriffe (MSc)
- Viele andere Veranstaltungen zu eingebetteten Systemen
 - IKV, FPGA Praktikum, Krypto auf Prog. HW, Vorlesung Seitenkanalangriffe
- Mehr Assembler?
 - Programmanalyse (MSc) bei Thorsten Holz

Studieren/Forschen/Arbeiten

- Möglichkeit einer Abschlussarbeit mit uC Bezug
 - Implementierung von Verfahren
 - Seitenkanalangriffe auf uC
- Werksstudent
- Studentische Hilfskraft (Hiwi) werden (bringt Ruhm und etwas Geld)
 - **Sehr gutes Klausurergebnis zwingend erforderlich!**
 - Korrekturhiwi: Korrigiert die eingesandten Übungen
 - Forschungshiwi: Unterstützt Mitarbeiter bei Forschungsprojekten

Studieren/Forschen/Arbeiten - Beispiel

Publikation entstanden aus dem „Embedded Smartcard Microcontrollers“ Praktikum:

Evaluation of SHA-3 Candidates for 8-bit Embedded Processors

Stefan Heyse, Ingo von Maurich, Alexander Wild, Cornel Reuber, Johannes Rave, Thomas Pöppelmann, Christof Paar, Thomas Eisenbarth

2nd SHA-3 Candidate Conference, August 23-24, 2010, University of California, Santa Barbara, USA.

Ende

Viel Erfolg beim Asuro Führerschein und der Klausur!