USART

ATmega1284P

Networks and Embedded Software

Module 4.3.5

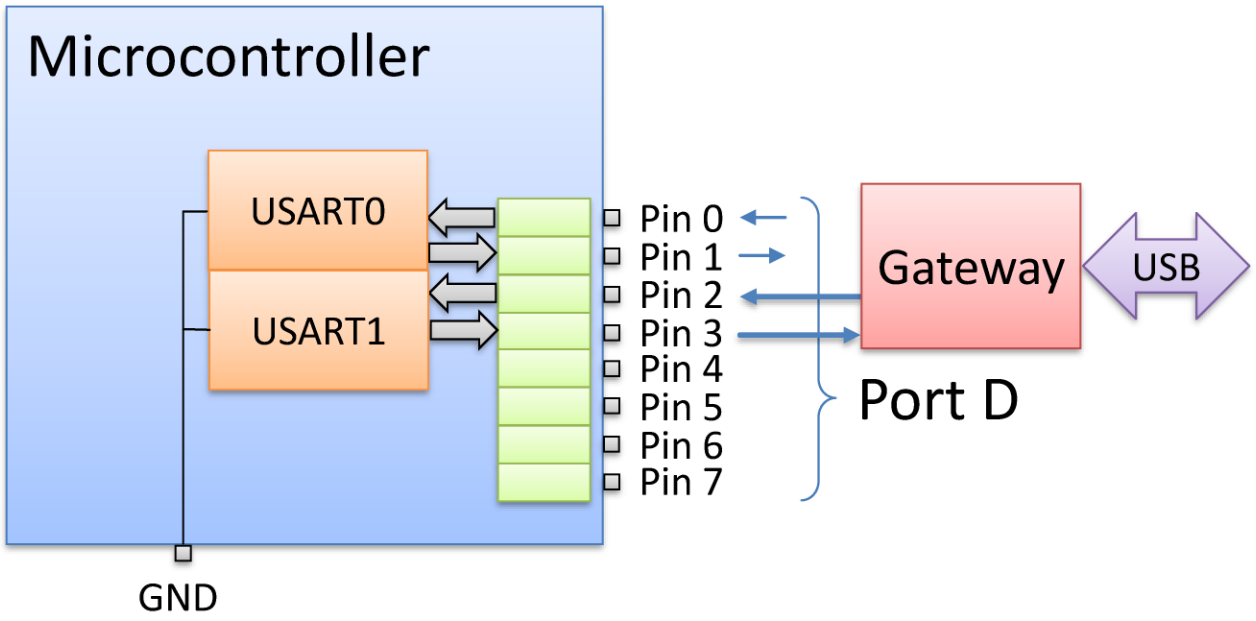
by Wolfgang Neff

USART (1)

- Implementation
  - Two USART modules
    - USART0 (RXD=PD0, TXD=PD1)
    - USART1 (RXD=PD2, TXD=PD3)
  - USART-to-USB gateway
    - Module: USART1
    - Configuration: 57600/8N1

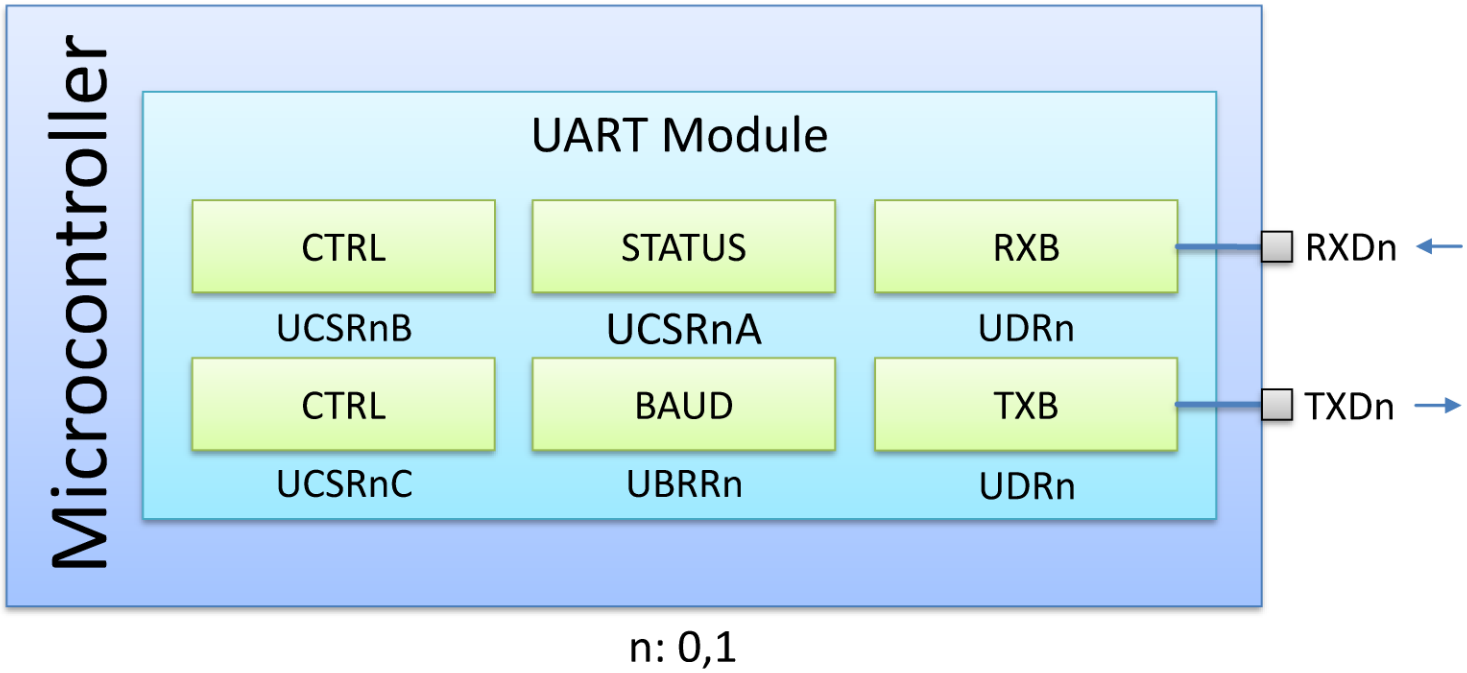
USART (2)

- Architecture



UART (3)

- Register Mapping



USART (4)

- Register Description
  - UDRn: USART I/O Data Register
    - Writes to Transmit Data Buffer Register (TXB)
    - Reads from Receive Data Buffer Register (RXB)
  - UCSRnA: USART Control and Status Register A
    - Bit 7 – RXC: USART Receive Complete
    - Bit 6 – TXC: USART Transmit Complete
    - Bit 5 – UDRE: USART Data Register Empty Flag

USART (5)

- Register Description (continued)
  - UCSRnB: USART Control and Status Register B
    - Bit 7 – RXCIE: Receive Complete Interrupt Enable
    - Bit 6 – TXCIE: Transmit Complete Interrupt Enable
    - Bit 5 – DREIE: Data Register Empty Interrupt Enable
    - Bit 4 – RXEN: Receiver Enable
    - Bit 3 – TXEN: Transmitter Enable

USART (6)

- Register Description (continued)
  - UCSRnC: USART Control and Status Register C
    - Bits 5:4 – USART Parity Mode
      - 0: No parity
      - 2: Even parity
      - 3: Odd parity
    - Bit 3 – SBMODE: Stop Bit Mode
      - 0: One stop bit
      - 1: Two stop bits

USART (7)

- Register Description (finished)
  - UCSRnC: USART Control and Status Register C
    - Bit 2:1 – CHSIZE: Character Size
      - Usually 3: 8 bits
      - Consult datasheet for details
  - UBRRn - USART Baud Rate n Register
    - Bit 11:0 – USART Baud Rate
      - See table below

USART (8)

- Baud Rate Table
  - F_CPU = 11059200 (default frequency)

Speed	UBRR
9600	71
14400	47
19200	32
38400	17
57600	11
115200	5

USART (9)

- Configuration Example (57600/8N1, no interrupts)
  - Configure RXD and TXD pins
    - `PORTD |= (1 << PORTD3);  // TXD - idle`
    - `DDRD |= (1 << DDD3);  // TXD - output`
    - `DDRD &= ~(1 << DDD2); // RXD - input`
  - Configure 8N1 asynchronous mode
    - ```
      UCSR1C = (
        (0 << UPM10) | (0 << USBS1) | (3 << UCSZ10)
      );
      ```

Pins are configured via the port module
      ```

USART (10)

- Configuration Example (continued)
  - Set speed to 57600 Baud (see Baud rate table)
    - `UBRR1 = 11;`
  - Enable receiver and transmitter
    - ```
      UCSR1B = (
          (1 << RXEN1) | (1 << TXEN1)
      );
      ```
      ```

USART (11)

- Polling
  - Synchronous Input/Output
    - Blocking read

```
while (!(UCSR1A & (1 << RXC1)));
data = UDR1;  // read from USART
```

    - Blocking write

```
while (!(UCSR1A & (1 << UDRE1)));
UDR1 = data;  // write to USART
```

  - Easy to implement but waste of performance

USART (12)

- Polling (continued)
  - Synchronous Input/Output
    - Non-blocking read
      -
```
if (UCSR1A & (1 << RXC1)) {
    data = UDR1;
}
```
    - Non-blocking write
      -
```
if (UCSR1A & (1 << UDRE1)) {
    UDR1 = data;
}
```
  - Performant but tricky. Good timing necessary.

USART (13)

- Register Summary

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UCSRnA	**RXC**	TXC	**UDRE**	FE	DOR	UPE	U2X	MPCM
UCSRnB	RXCIE	TXCIE	UDRIE	**RXEN**	**TXEN**	UCSZ2	RXB8	TXB8
UCSRnC	UMSEL[1:0]		**UPM[1:0]**		**USBS**	**UCSZ1**	**UCSZ0**	UCPOL
UBBRnL	-	-	-	-	**UBRR[11:8]**			
UBBRnH	**UBRR[7:0]**							
UDRn	**TXB/RXB[7:0]**							

USART (14)

- Interrupt Summary

Source	Description
USARTn_RX_vect	USART receive complete interrupt
USARTn_UDRE_vect	USART data register empty interrupt
USARTn_TX_vect	USART transmit complete interrupt