USART

ATmega4809

Networks and Embedded Software

Module 4.3.5

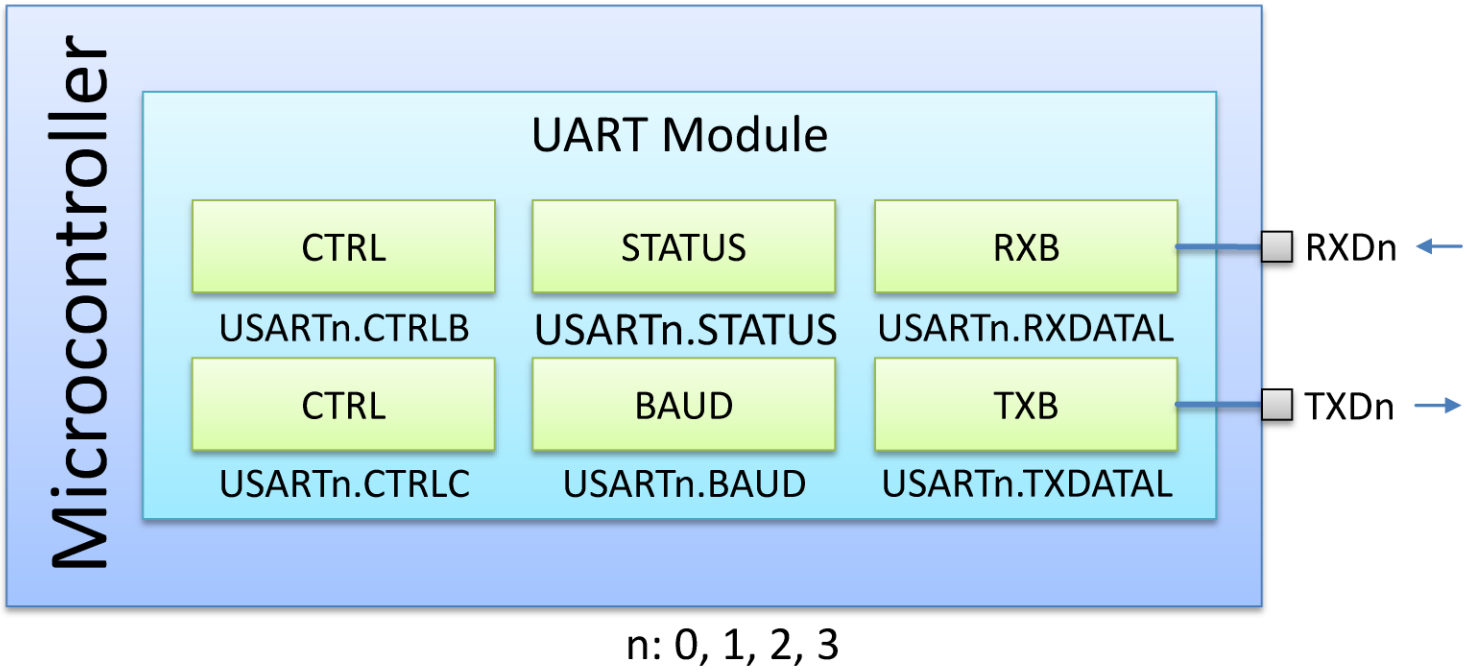
by Wolfgang Neff

USART (1)

- Implementation
  - Four USART modules
    - USART0 (RXD=PA0, TXD=PA1)
    - USART1 (RXD=PC0, TXD=PC1)
    - USART2 (RXD=PF0, TXD=PF1)
    - USART3 (RXD=PB0, TXD=PB1)
  - Virtual COM Port via Embedded Debugger
    - Module: USART1
    - Configuration: 115200/8N1

UART (2)

- Register Mapping



USART (3)

- Register Description
  - RXDATAL: Receiver Data Register Low Byte
  - TXDATAL: Transmit Data Register Low Byte
  - STATUS: USART Status Register
    - Bit 7 – RXCIF: USART Receive Complete Interrupt Flag
    - Bit 6 – TXCIF: USART Transmit Complete Interrupt Flag
    - Bit 5 – DREIF: USART Data Register Empty Flag

USART (4)

- Register Description (continued)
  - CTRLA: USART Control Register A
    - Bit 7 – RXCIE: Receive Complete Interrupt Enable
    - Bit 6 – TXCIE: Transmit Complete Interrupt Enable
    - Bit 5 – DREIE: Data Register Empty Interrupt Enable
  - CTRLB: USART Control Register B
    - Bit 7 – RXEN: Receiver Enable
    - Bit 6 – TXEN: Transmitter Enable

USART (5)

- Register Description (continued)
  - CTRLC: USART Control Register C
    - Bits 7:6 – CMODE: USART Communication Mode
      - Usually `USART_CMODE_ASYNCHRONOUS_gc`
      - Consult datasheet for details
    - Bits 5:4 – PMODE: Parity Mode
      - `USART_PMODE_DISABLED_gc`: No parity
      - `USART_PMODE_EVEN_gc`: Even parity
      - `USART_PMODE_ODD_gc`: Odd parity

USART (6)

- Register Description (finished)
  - CTRLC: USART Control Register C (continued)
    - Bit 3 – SBMODE: Stop Bit Mode
      - 0: One stop bit
      - 1: Two stop bits
    - Bit 2:0 – CHSIZE: Character Size
      - Usually `USART_CHSIZE_8BIT_gc`
      - Consult datasheet for details
  - BAUD: USART Baud Rate Register

USART (7)

- Baud Rate Table
  - F_CPU = 3333333 (default frequency)

Speed	BAUD
9600	1389
14400	926
19200	694
38400	347
57600	231
115200	116

USART (8)

- Configuration Example (115200/8N1, no interrupts)
  - Configure RXD and TXD pins
    - `PORT.OUT |= PIN1_bm;  // TXD - idle`
    - `PORT.DIR |= PIN1_bm;  // TXD - output`
    - `PORT.DIR &= ~PIN0_bm; // RXD - input`
  - Configure 8N1 asynchronous mode
    - ```
      USART.CTRLC = (
          USART_CHSIZE_8BIT_gc |
          USART_PMODE_DISABLED_gc
      );
      ```

Pins are configured via the port module
      ```

USART (9)

- Configuration Example (continued)
  - Set speed to 115200 Baud (see Baud rate table)
    - `USART.BAUD = 116;`
  - Enable receiver and transmitter
    - ```
      USART.CTRLB = (
          USART_RXEN_bm |
          USART_TXEN_bm
      );
      ```
      ```

USART (10)

- Polling
  - Synchronous Input/Output
    - Blocking read

```
while (!(USART.STATUS & USART_RXCIF_bm));
data = USART.RXDATAL;  // read from USART
```

    - Blocking write

```
while (!(USART.STATUS & USART_DREIF_bm));
USART.TXDATAL = data;  // write to USART
```

  - Easy to implement but waste of performance

Active Wait.
Do Nothing

USART (11)

- Polling (continued)
  - Asynchronous Input/Output
    - Non-blocking read

```
if (USART.STATUS & USART_RXCIF_bm) {
    data = USART.RXDATAL;
}
```

    - Non-blocking write

```
if (USART.STATUS & USART_DREIF_bm) {
    USART.TXDATAL = data;
}
```

  - Performant but tricky. Good timing necessary.

USART (12)

- Register Summary

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RXDATAL	**DATA[7:0]**							
RXDATAH	RXCIF	BUFOVF	-	-	-	FERR	PERR	DATA[8]
TXDATAL	**DATA[7:0]**							
TXDATAH	-	-	-	-	-	-	-	DATA[8]
STATUS	**RXCIF**	TXCIF	**DREIF**	RXSIF	ISFIF	-	BDF	WFB
CTRLA	RXCIF	TXCIF	DREIF	RXSIF	LBME	ABEIE	RS485[1:0]	
CTRLB	**RXEN**	**TXEN**	-	SFDEN	ODME	RXMODE[1:0]		MPCM

USART (13)

- Register Summary (continued)

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTRLC	**CMODE[1:0]**		**PMODE[1:0]**		**SBMODE**	**CHSIZE[2:0]**		
BAUD	**BAUD[7:0]**							
	BAUD[15:8]							
DBGCTRL	-	-	-	-	-	-	-	DBGRUN
EVCTRL	-	-	-	-	-	-	-	IREI
TXPLCTRL	TXPL[7:0]							
RXPCTRL	-	RXPL[6:0]						

USART (14)

- Interrupt Summary

Source	Description
USARTn_RXC_vect	USART receive complete interrupt vector
USARTn_DRE_vect	USART data register empty interrupt vector
USARTn_TXC_vect	USART transmit complete interrupt vector