

ATmega328P USART0

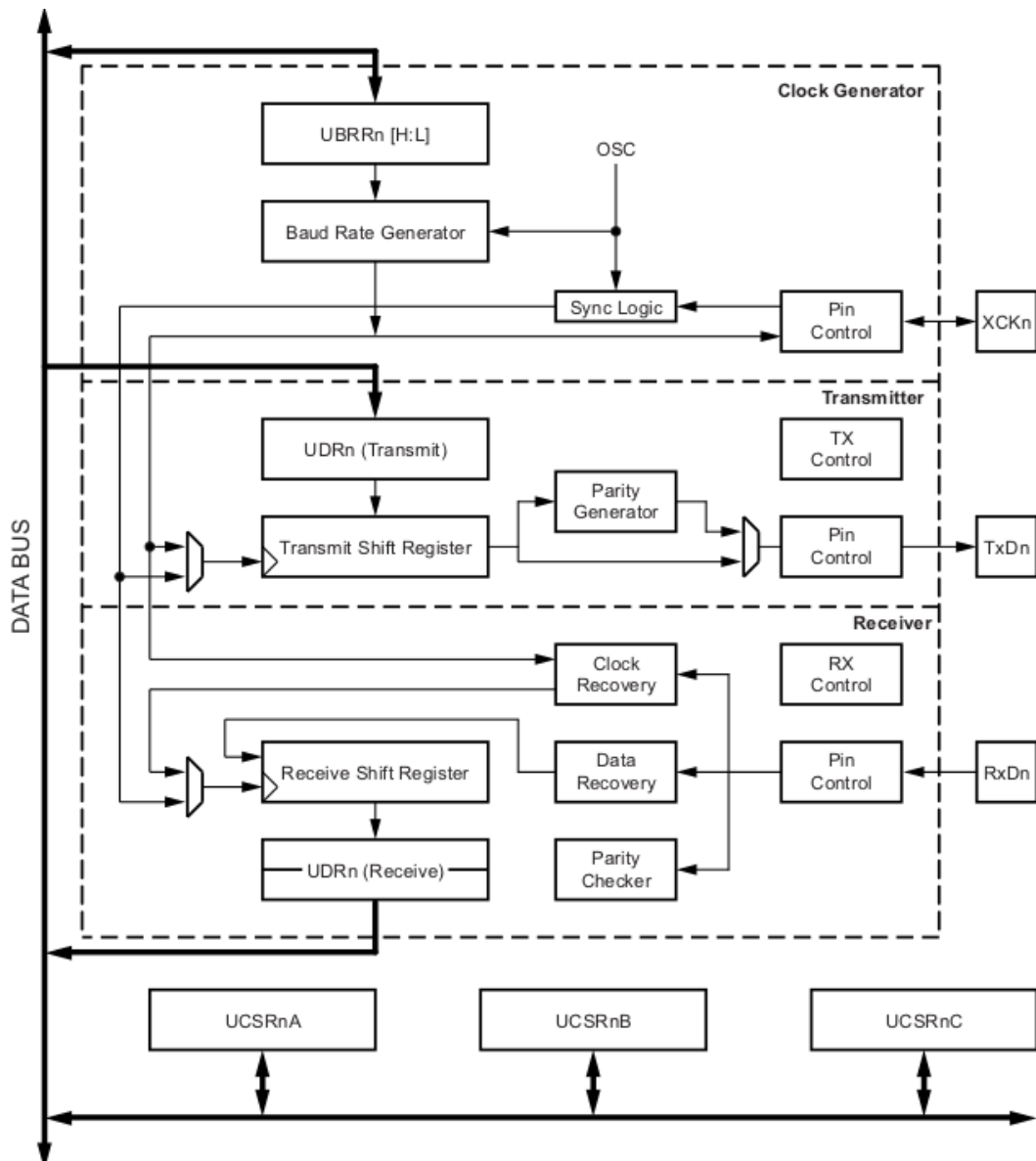
Narendiran S

July 27, 2020

1 Features

- Full duplex operation (independent serial receive and transmit registers).
- Asynchronous or synchronous operation
- High resolution baud rate generator
- Serial frame with 5,6,7,8,9 data bits and 1 or 2 stop bits
- Odd or even parity generator and checker by hardware
- Double speed asynchronous communication mode

2 Block Diagram



2.1 Clock Generator Block

- Consist of sync. Logic for external clock input for usage in sync. slave operation
- Consist of Baud rate Generator.
- Uses the *XCKn* pin for sync. Transfer mode

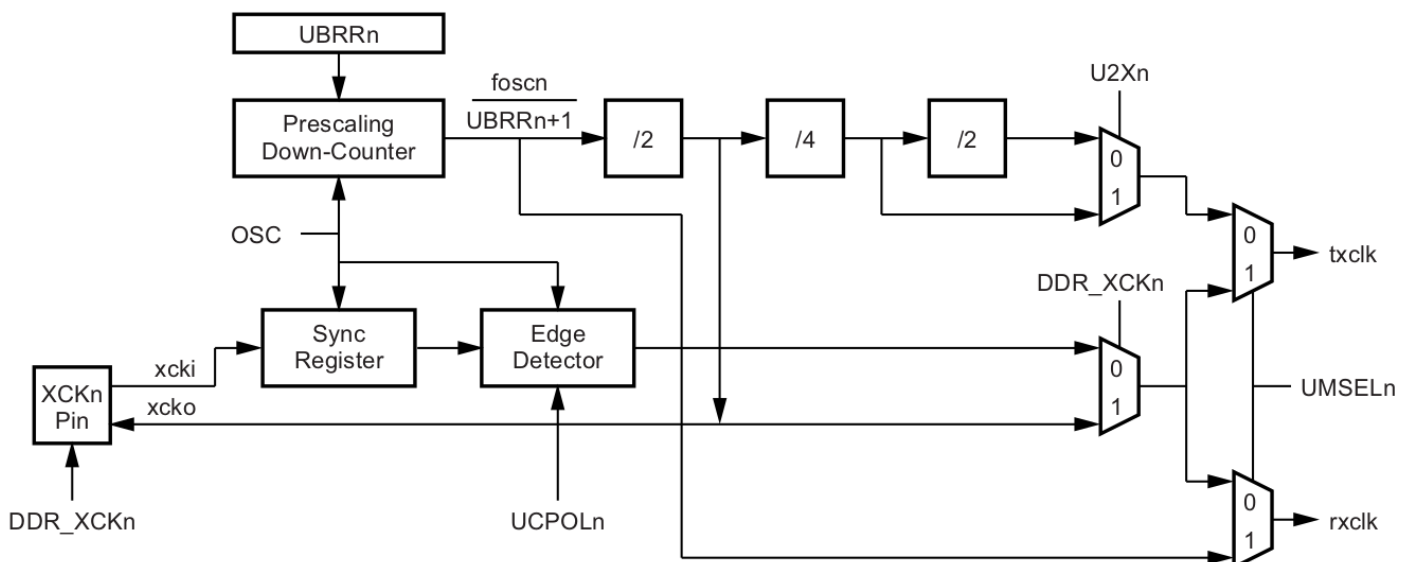
2.2 Transmitter Block

- Consist of single write buffer – continuous transfer of data without delay between frames
- Consist of Serial Shift register and Parity Generator
- Also, Control logic for handling different serial frame format.

2.3 Receiver Block

- Consist of Clock and data recovery unit – uses for Asynchronous reception
- Consist of Parity Checker, Control Logic, Shift Register, Two level Receiver buffer
- Can support frame error, data overrun parity error

3 Clock Generation



- Generates Base Clock for Transmitter and Receiver.
- USART supports four modes of clock operation
 - (i) Normal Asynchronous
 - (ii) Double Speed Asynchronous
 - (iii) Master synchronous
 - (iv) Slave synchronous
- Selection between Asynchronous and Synchronous is done by *UMSELn* bit in *UCSRnC* - USART Control and Status Register C.
- The Double Speed is selected by *U2Xn* bit in *UCSRnA* - USART Control and Status Register A.
- In Synchronous Mode, the master or slave mode is selected by *DDR_XSCn* bit direction. [external - slave mode; internal - master mode]

Signals	Description
txclk	Transmitte Clock
rxclk	Receiver Base Clock
xclki	Input from <i>XCK</i> pin - used for synchronous slave operation.
xelko	Clock output to <i>XCK</i> pin - used for synchronous master operation.
fosc	<i>XTAL</i> pin frequency (System clock).

3.1 Internal Clock Generation - The Baud Rate Generator

- Used for Asynchronous and Synchronous Master modes of operation.
- Programmed using *UBRRn* register.

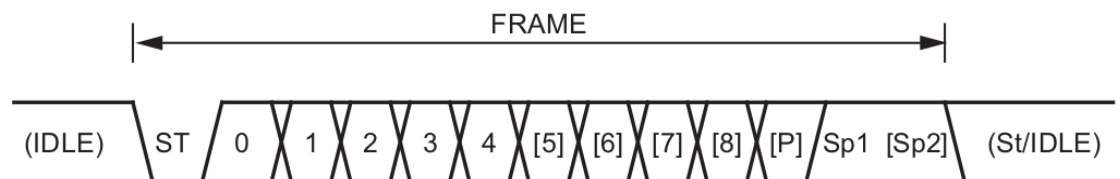
Operating Mode	UBRRn calculation
Asynchronous Normal Mode(<i>U2Xn</i> == 0)	$UBRRn = \frac{f_{osc}}{16 * BAUD} - 1$
Asynchronous Double Speed Mode(<i>U2Xn</i> == 1)	$UBRRn = \frac{f_{osc}}{8 * BAUD} - 1$
Synchronous Master Mode	$UBRRn = \frac{f_{osc}}{2 * BAUD} - 1$

3.2 External Clock

- Used by synchronous Slave mode.
- External clock input from *XCKn* pin is used and should

$$f_{XCK} < \frac{f_{osc}}{4}$$

4 Frame Format



St Start bit, always low.

(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

Sp Stop bit, always high.

IDLE No transfers on the communication line (*RxDn* or *TxDn*). An IDLE line must be high.

- A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking.
- The combinations can be
 - 1 start bit
 - 5 or 6 or 7 or 8 or 9 data bits
 - no or even or odd parity bits
 - 1 or 2 stop bits
- A frame starts with start bit followed by LSB data bits.
- Next the data bet can be from 5 to 9 ending with MSB data bits.
- Parity bits may be added if enabled.
- Finally, stop bit of 1 or 2 size is added.
- Generally, the line is idel with high Logic.

5 Register Description

UDRn – USART I/O Data Register n

7	6	5	4	3	2	1	0
RXB[7:0]							
TXB[7:0]							

UCSRnA – USART Control and Status Register n A

7	6	5	4	3	2	1	0
RXCn	TXCn	UDREN	FEn	DORn	UPEn	U2Xn	MPCMn

UCSRnB – USART Control and Status Register n B

7	6	5	4	3	2	1	0
RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	UCSZ _{n2}	RXB _n	TXB8 _n

UCSRnC – USART Control and Status Register n C

7	6	5	4	3	2	1	0
UMSEL _{n1}	UMSEL _{n0}	UPM _{n1}	UPM _{n0}	USBS _n	UCSZ _{n1}	UCSZ _{n0}	UCPOL _n

- **RXC_n** - USART Receive Complete - Set when there are unread data in receive buffer.
- **TXC_n** - USART Transmit Complete - Set when the entire frame in the transmit shift register has been shifted out and there are no new data currently present in the transmit buffer.
- **UDRE_n** - USART Data Register Empty - indicates if the transmit buffer is ready to receive new data. A one indicates buffer is empty and ready to transmit.
- **U2X_n** - Double the USART Transmission Speed - Affects only the asynchronous operation. One will increase the speed of transfer rate in asynchronous operation.
- **RXCIE_n** - RX Complete Interrupt Enable n - Writing one will enable Receive Complete interrupt.
- **TXCIE_n** - TX Complete Interrupt Enable n - Writing one will enable Transmit Complete interrupt.
- **UDRIE_n** - USART Data Register Empty Interrupt Enable n - Enable data register empty interrupt.
- **RXEN_n** - Receiver Enable - enable the receiver for reception.
- **TXEN_n** - Transmitter Enable - enable the Transmitter for Transmission.
- **UCSZ_{n[2:0]}** - Character Size n - select the number of data bits in a frame.
- **RXB8_n** - Receive Data Bit 8 n - it's the actual 9th bit received.
- **TXB8_n** - Transmit Data Bit 8 n - it's the actual 9th bit to be transmitted.
- **UMSEL_{n[1:0]}** - USART Mode Select - Select the mode.
- **UPM_{n[1:0]}** - Parity Mode - Disable or set the parity mode type.
- **USBS_n** - Stop Bit select - Selects the number of stop bits to be inserted by transmitter.

<i>UMSELn[1:0]</i>	Mode	<i>UPMn[1:0]</i>	Parity Mode	<i>USBSn</i>	Stop Bit(s)
00	Asynchronous USART	00	Disabled	0	1-bit
01	Synchronous USART	01	Reserved	0	2-bit
10	Reserved	10	Even Parity		
11	Master SPI	11	Odd Parity		

<i>UCSZn[2:0]</i>	Character Size
000	5-bit
001	6-bit
010	7-bit
011	8-bit
111	9-bit

UBRRnL and UBRRnH – USART Baud Rate Registers

15	14	13	12	11	10	9	8
-	-	-	-	UBRRn[11:8]			
UBRRn[7:0]							
7	6	5	4	3	2	1	0

UBRRn[11:0] - the actual 12-bit USART Baud Rate Registers.

6 Configurint USART

- First, the mode is selected by configuring the *UMSEL0[1:0]* bits in **UCSR0C** register.
- Next, the Baud rate is choosen and set in *UBRR0[11:0]* bits in **UBRR0H** and **UBRR0L** registers.
- Next, the frame format is set by configuring,
 - Data Length - by configuring *UCSZ0[2:0]* bit in **UCSR0B** and **UCSR0C** register.
 - Parity - by configuring *UPM0[1:0]* bit in **UCSR0C** register.
 - Stop bits - by configuring *USBS0* bit in **UCSR0C** register.
- Interrupt may be enabled by setting bits in **UCSR0A** register and ISR are wirtten.
- Finally, the Transmitter and Receiver are enabled by setting *TXEN0* and *RXEN0* bits in **UCSR0B**.
- The data can be sent by checking if the *UDRE0* bit is set in **UCSR0A** register and wiring the 8-bit data into **UDR0** register.
- The data can be received by checking if the *RXC0* bit is set in **UCSR0A** register and reading the 8-bit data from **UDR0** register.
- The code for a simple USART is seen below,

```
// Setting up the Mode
// Select the Asynchronous Master Mode.
// Setting UMSEL0[1:0] in UCSROC to 00
UCSR0C &= ~(1<<UMSEL00);
UCSR0C &= ~(1<<UMSEL01);

// setting up the Buad rate
// Due to The Clock rate being 8MHz, for a buad rate of 9600
// UBRR0 = (fosc / (16*BAUD)) -1
// So UBRR0 = (8000000 / (16 * 9600)) - 1 = 0x33
UBRR0H = 0x00;
UBRR0L = 0x33;

// setting up the Frame Format
```

```

// Let's select 8-bit data bits, no parity, and 1 stop bit
// 8 - bit data bits
// By selecting UCSZ0[2:0] in UCSROC and UCSROB register to be 011
UCSROB &= ~(1<<UCSZ02);
UCSROC |= (1<<UCSZ01);
UCSROC |= (1<<UCSZ00);
// No parity
// By selecting UPM0[1:0] in UCSROC to 00
UCSROC &= ~(1<<UPM01);
UCSROC &= ~(1<<UPM00);
// 1 stop bit
// By selecting USBS0 in UCSROC to 0
UCSROC &= ~(1<<USBS0);

// Disabling any interrupts
UCSROB &= ~(1<<7);
UCSROB &= ~(1<<6);
UCSROB &= ~(1<<5);

// Enabling Transmitter
UCSROB |= (1<<TXEN0);
// Enabling Receiver
UCSROB |= (1<<RXEN0);

```

```

void USART0sendChar(uint8_t data_)
{
    //CHECKING if transmitet buffer is empty
    while((UCSROA & (1<<UDRE0)) == 0x00){};
    UDRO = data_;
}

uint8_t USART0receiveChar()
{
    // wait for the data to be received
    while((UCSROA & (1<<RXC0)) == 0x00){};
    return UDRO;
}

```