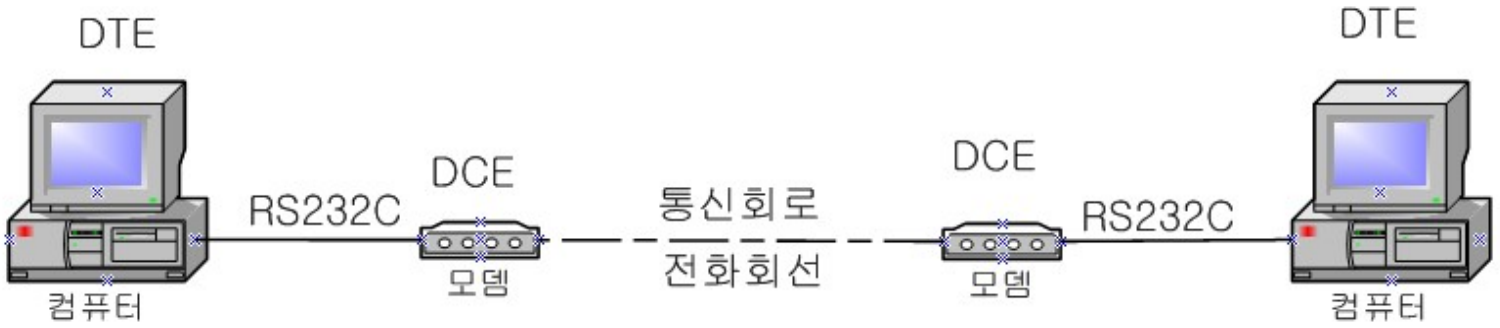
ATmega128

Serial Communication (RS-232C)

RS-232C

- EIA (Electronics Industries Association)
- DTE (Data Terminal Equipment)
- DCE (Data Communication Equipment)

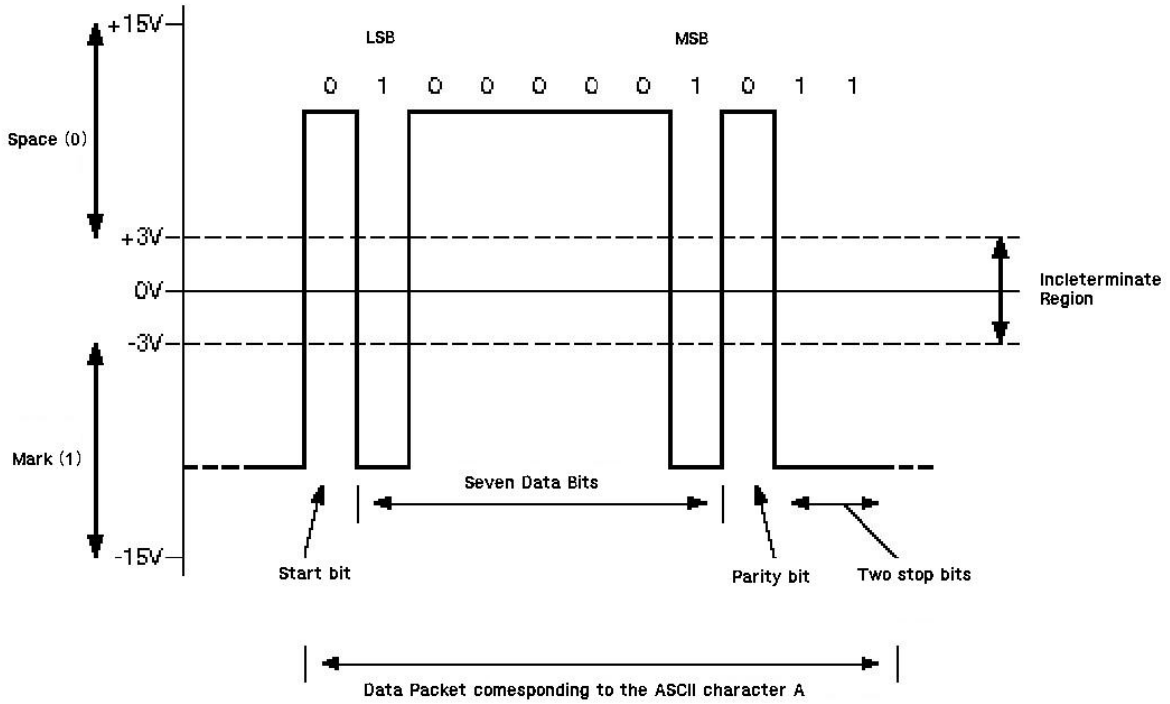


RS-232C Signals

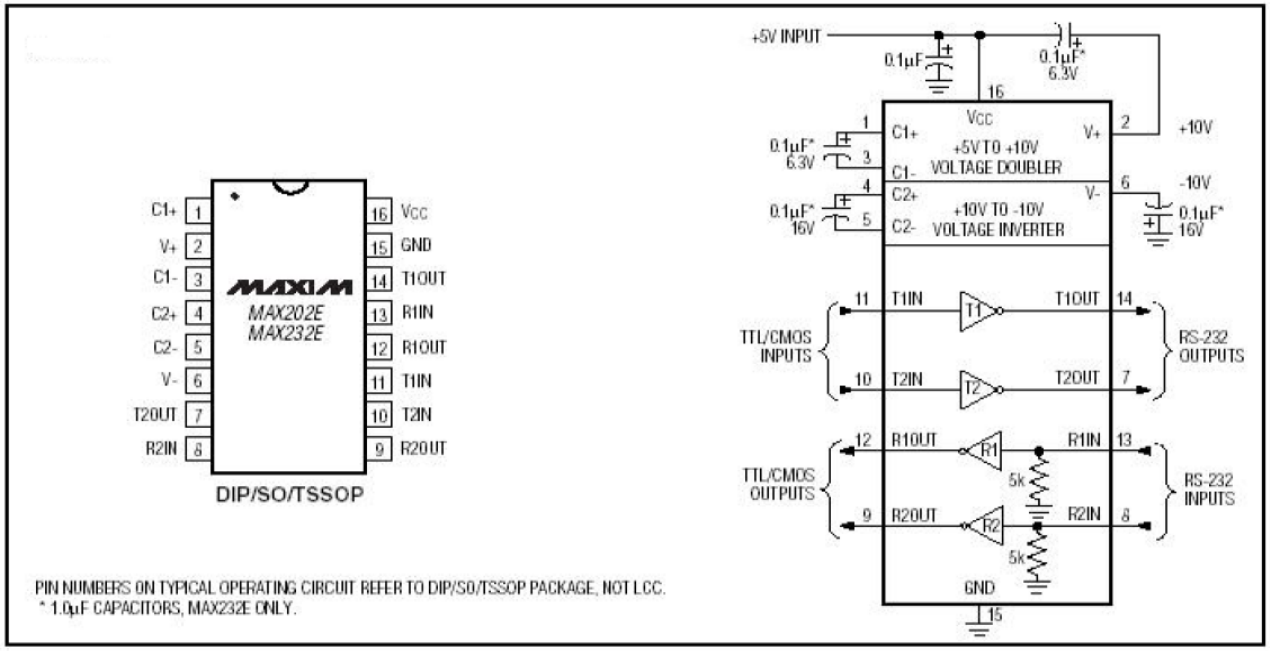
핀 번호 (Pin No.)	명 칭 (Signal Name)	신호방향 DTE-DCE	기 호 (Symbol)
1	기기 접지(Frame Ground)	-	FG
2	송신 데이터(Transmitted Data)	->	TXD
3	수신 데이터(Receive Data)	<-	RXD
4	송신 요구(Request to Send)	->	RTS
5	송신 허가(Clear to Send)	<-	CTS
6	통신기기 준비 완료(Data Set Ready)	<-	DSR
7	신호접지(Signal Ground)	-	SG
8	캐리어 검출(Data Carrier Detect)	<-	DCD
20	터미널 준비 완료(Data Terminal Ready)	->	DTR
22	Ring Indicator	<-	RI

RS-232C Signal

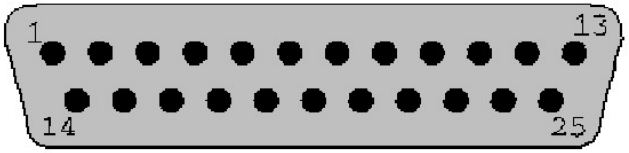
상 태(State)	‘L'	'H'
전압 범위(Voltage)	-25V ~ -3V	+3V ~ +25V
논 리(Logical Value)	1	0
명 칭(Name)	mark	space



MAX232

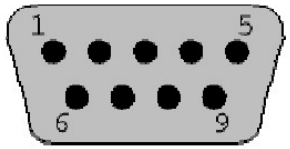

C1+ 1
V+ 2
C1- 3
C2+ 4
C2- 5
V- 6
T2OUT 7
R2IN 8
16 VCC
15 GND
14 T1OUT
13 R1IN
12 R1OUT
11 T1IN
10 T2IN
9 R2OUT
MAXIM
MAX202E
MAX232E
DIP/SO/TSSOP
+5V INPUT
0.1µF
0.1µF* 6.3V
VCC
C1+
V+
+10V
+5V TO +10V VOLTAGE DOUBLER
C1-
C2+
V-
-10V
+10V TO -10V VOLTAGE INVERTER
C2-
0.1µF* 16V
T1IN
T1OUT
T2IN
T2OUT
TTL/CMOS INPUTS
RS-232 OUTPUTS
R1OUT
R1IN
R2OUT
R2IN
TTL/CMOS OUTPUTS
RS-232 INPUTS
5k
GND
PIN NUMBERS ON TYPICAL OPERATING CIRCUIT REFER TO DIP/SO/TSSOP PACKAGE, NOT LCC.
* 1.0µF CAPACITORS, MAX232E ONLY.

D-SUB 25 Pin Connector



RS 232 Signals			
Pin	Description	Pin	Description
1	Earth Ground	14	Secondary TXD
2	TXD - Transmitted Data	15	Transmit Clock
3	RXD - Received Data	16	Secondary RXD
4	RTS - Request To Send	17	Receiver Clock
5	CTS - Clear Set Ready	18	Unassigned
6	DSR - Data Set Ready	19	Secondary RTS
7	GND - Logic Ground	20	DTR - Data Terminal Ready
8	DCD - Data Carrier Detect	21	Signal Quality Detect
9	Reserved	22	Ring Detect
10	Reserved	23	Data Rate Select
11	Unassigned	24	Transmit Clock
12	Secondary DCD	25	Unassigned
13	Secondary CTS		

D-SUB 9 Pin Connector

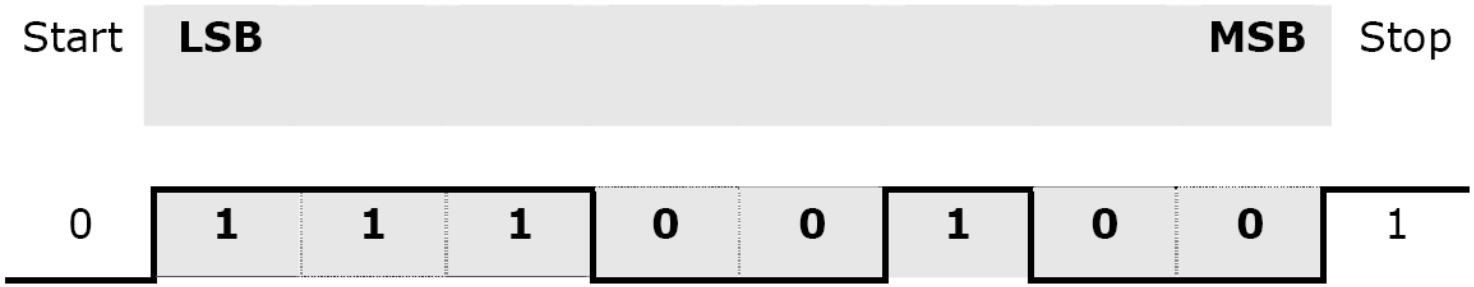


RS-574(IBM PC/AT) Signal			
Pin	Description	Pin	Description
1	DCD - Data Carrier Detect	6	Data Set Ready
2	RXD - Received Data	7	RTS - Request To send
3	TXD - Transmitted Data	8	CTS - Clear To Send
4	DTR - Data Terminal Ready	9	Ring Detect
5	GND - Logic Ground		

Bps/Baud Rate

- 300
- 600
- 1200
- 2400
- 4800
- 9600
- 19200
- 38400
- 57600
- 115200

Data Format



시리얼 모드1 전송방식으로 0x27 (0010 0111)을 전송할 때

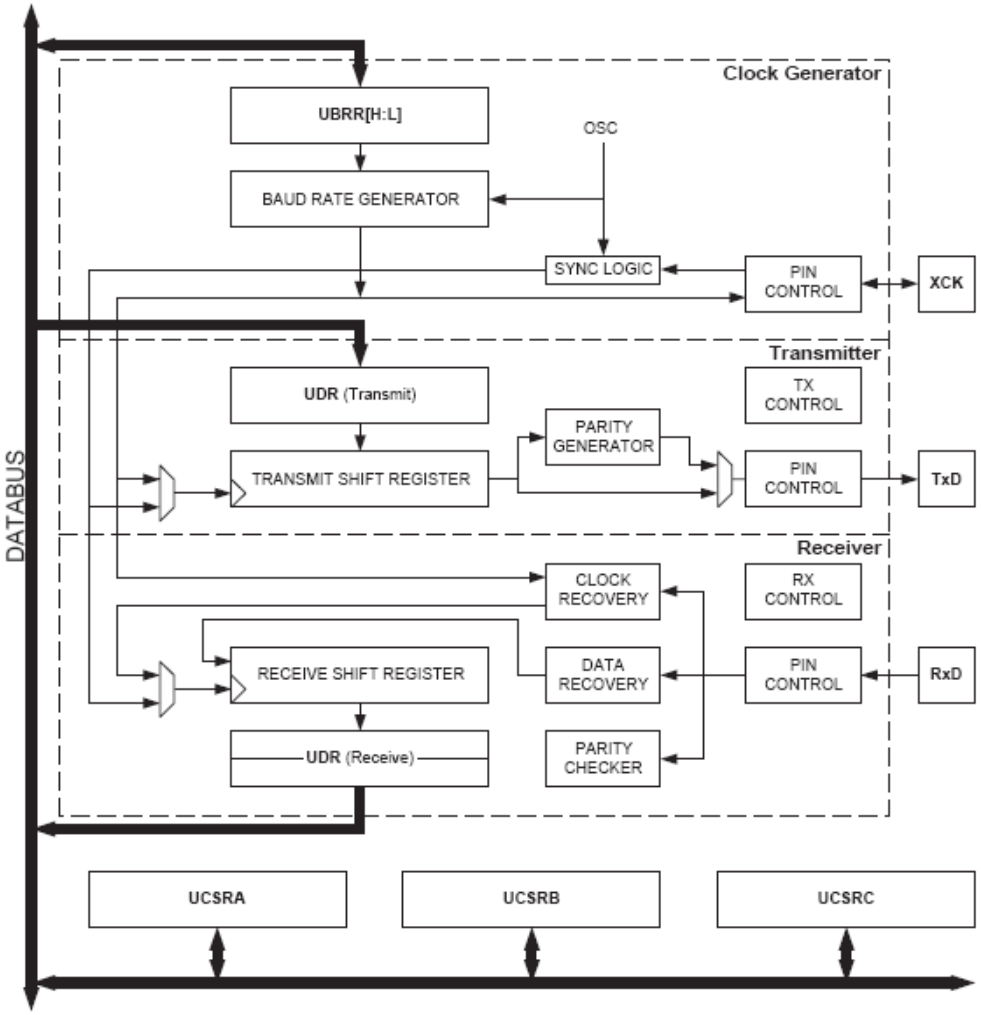
- N-8-1 : no parity, 8 bit data, one stop bit

USART

Universal Synchronous/Asynchronous Receiver Transmitter

- Synchronous or asynchronous serial communication
- Asynchronous serial communication(**)
  - not use a clock to validate data
  - serial interfaces are cheap, easy to use, very common
  - USB, replacing the serial com ports (on PC)
- Serial data is transferred a bit at a time
- USART communicates in a full-duplex mode (simultaneous xmit, rcv)
- ATmega128 : 2 USART's (USART0, USART1)

USART Block Diagram



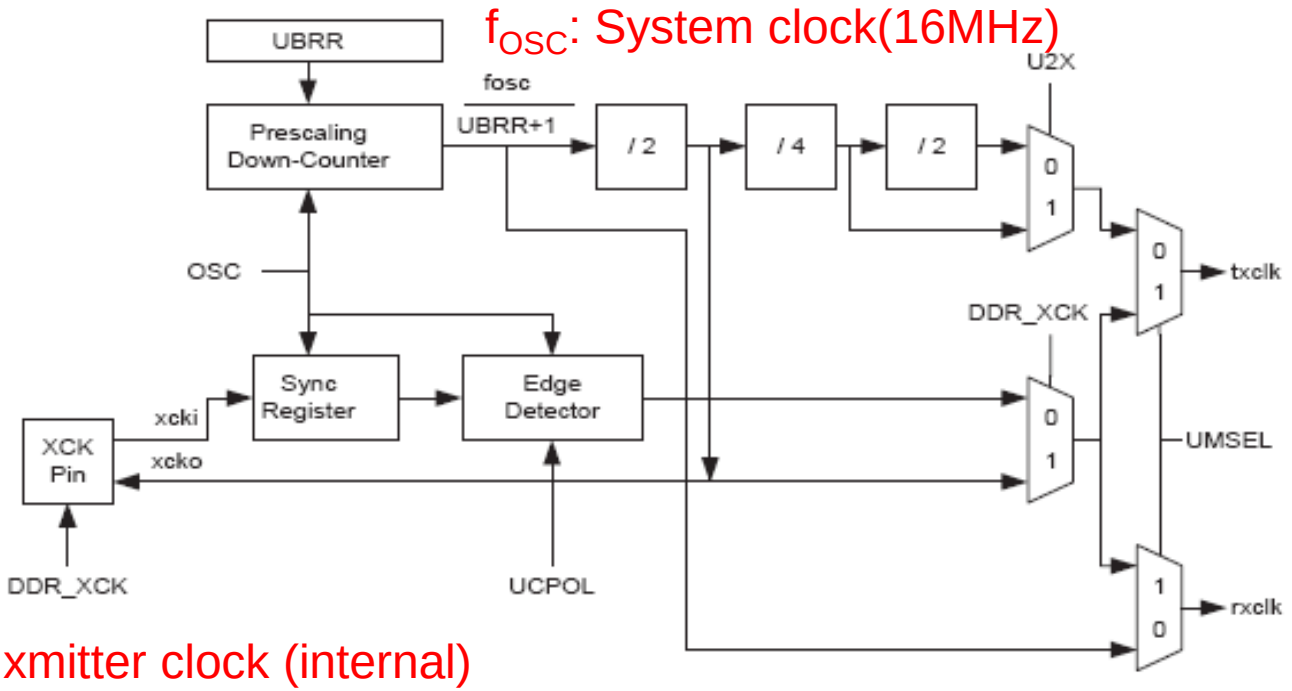
Clock Generator
XCK (Synchronous only)

Transmitter

Receiver

Clock Generation

- **Registers**
  - UMSEL bit in UCSRC : Asynch(0) or Synch(1)
  - U2X in UCSRA : Double speed asynch
  - DDR_XCK (UMSEL=1) : clock source is internal(Master) or external(Slave)



f_{OSC}: System clock(16MHz)

txclk: xmitter clock (internal)
rxclk: rcv clock (internal)

Internal Clock Generation

- Asynchronous/synchronous master modes of operation
- USART Baud Rate Register (UBRR)
  - Prescaling down-counter, loaded with UBRR each time the counter has counted down to zero (or UBRRL reg. is written)

Table 74. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	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	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{OSC} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRRL Registers, (0 - 4095)

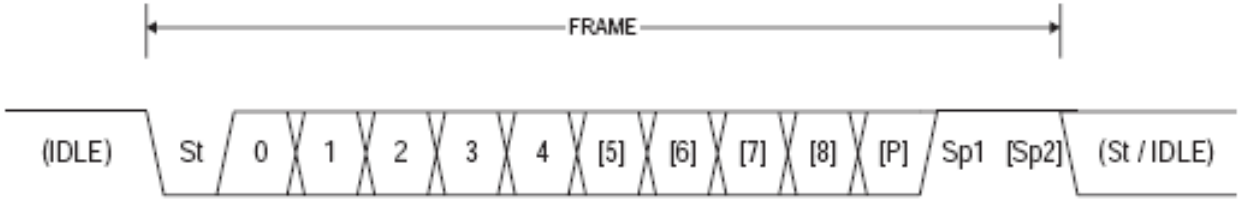
Baud Rate(bps)

Table 85. Examples of UBRR Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	f_{osc} = 16.0000 MHz				f_{osc} = 18.4320 MHz				f_{osc} = 20.0000 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	–	–	4	-7.8%	–	–	4	0.0%
1M	0	0.0%	1	0.0%	–	–	–	–	–	–	–	–
Max (1)	1 Mbps		2 Mbps		1.152 Mbps		2.304 Mbps		1.25 Mbps		2.5 Mbps	

1. UBRR = 0, Error = 0.0%

Serial Frame Formats



St	Start bit, always low. ☜ A frame starts with the start bit followed by the LSB data bit
(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 (RxD or TxD). An IDLE line must be high.

Every serial frame has at least
- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit (optional)
- 1 or 2 stop bits

Serial Frame Formats–Registers

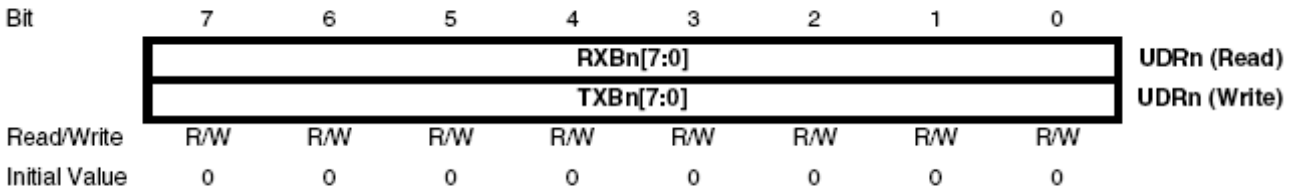
- USART Character Size (UCSZ2:0) Bits
- USART Parity mode (UPM1:0) bits
- USART Stop Bit Select (USBS) bit
  - Receiver ignores the 2nd stop bit
  - FE(Frame error) will only be detected when the first stop bit is 0

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	–	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

Registers

- **USARTn I/O Data Register – UDRn**
  - USARTn Transmit Data Buffer register and Receive Data Buffer register use the same I/O address(UDRn).



  - Transmit buffer can only be written when the UDREn flag in the UCSRAn is set
  - When data is written to the transmit buffer (xmitter is enabled), the transmitter will load the data into the Transmit Shift Register when the Shift register is empty

Registers-UCSRnA

- **USART Control & Status Register A (UCSRnA) (1/3)**

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e., does not contain any unread data). If the receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: USART Transmit Complete**

This flag bit is 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 (UDRn). The TXCn flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

Registers-UCSRnA

- **USART Control & Status Register A (UCSRnA) (2/3)**

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 5 – UDREn: USART Data Register Empty**

The UDREn flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn flag can generate a Data Register Empty interrupt (see description of the UDRIEn bit).

UDREn is set after a reset to indicate that the Transmitter is ready.

- **Bit 4 – FEn: Frame Error**

This bit is set if the next character in the receive buffer had a Frame Error when received. I.e. when the first stop bit of the next character in the receive buffer is zero. This bit is valid until the receive buffer (UDRn) is read. The FEn bit is zero when the stop bit of received data is one. Always set this bit to zero when writing to UCSRnA.

- **Bit 3 – DORn: Data OverRun**

This bit is set if a Data OverRun condition is detected. A data overrun occurs when the receive buffer is full (two characters), it is a new character waiting in the Receive Shift Register, and a new start bit is detected. This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

Registers-UCSRnA

- **USART Control & Status Register A (UCSRnA) (3/3)**

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 2 – UPEn: Parity Error**

This bit is set if the next character in the receive buffer had a Parity Error when received and the parity checking was enabled at that point (UPMn1 = 1). This bit is valid until the receive buffer (UDRn) is read. Always set this bit to zero when writing to UCSRnA.

- **Bit 1 – U2Xn: Double the USART Transmission Speed**

This bit only has effect for the asynchronous operation. Write this bit to zero when using synchronous operation.

Writing this bit to one will reduce the divisor of the baud rate divider from 16 to 8 effectively doubling the transfer rate for asynchronous communication.

- **Bit 0 – MPCMn: Multi-Processor Communication Mode**

This bit enables the Multi-processor Communication mode. When the MPCMn bit is written to one, all the incoming frames received by the USART Receiver that do not contain address information will be ignored. The transmitter is unaffected by the MPCMn setting. For more detailed information see "Multi-processor Communication Mode" on page 187.

Registers-UCSRnB

- **USART Control & Status Register B (UCSRnB) (1/3)**

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIEn: RX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the RXC flag. A USART Receive Complete interrupt will be generated only if the RXCIE bit is written to one, the global interrupt flag in SREG is written to one and the RXC bit in UCSRnA is set.

- **Bit 6 – TXCIE: TX Complete Interrupt Enable**

Writing this bit to one enables interrupt on the TXCn flag. A USARTn Transmit Complete interrupt will be generated only if the TXCIEn bit is written to one, the global interrupt flag in SREG is written to one and the TXCn bit in UCSRnA is set.

- **Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable**

Writing this bit to one enables interrupt on the UDREn flag. A Data Register Empty interrupt will be generated only if the UDRIEn bit is written to one, the global interrupt flag in SREG is written to one and the UDREn bit in UCSRnA is set.

Registers-UCSRnB

- **USART Control & Status Register B (UCSRnB) (2/3)**

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – RXENn: Receiver Enable**

Writing this bit to one enables the USARTn Receiver. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FEn, DORn and UPEn flags.

- **Bit 3 – TXENn: Transmitter Enable**

Writing this bit to one enables the USARTn Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and transmit buffer register do not contain data to be transmitted. When disabled, the transmitter will no longer override the TxDn port.

- **Bit 2 – UCSZn2: Character Size**

The UCSZn2 bits combined with the UCSZn1:0 bit in UCSRnC sets the number of data bits (character size) in a frame the Receiver and Transmitter use.

Registers-UCSRnB

- **USART Control & Status Register B (UCSRnB) (3/3)**

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – RXB8n: Receive Data Bit 8**

RXB8n is the ninth data bit of the received character when operating with serial frames with 9-data bits. Must be read before reading the low bits from UDRn.

- **Bit 0 – TXB8n: Transmit Data Bit 8**

TXB8n is the 9th data bit in the character to be transmitted when operating with serial frames with 9 data bits. Must be written before writing the low bits to UDRn.

Registers-UCSRnC

- **USART Control & Status Register C (UCSRnC) (1/4)**

Bit	7	6	5	4	3	2	1	0	
	–	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 7 – Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, these bit must be written to zero when UCSRnC is written.

- **Bit 6 – UMSELn: USART Mode Select**

This bit selects between Asynchronous and Synchronous mode of operation.

Table 77. UMSELn Bit Settings

UMSELn	Mode
0	Asynchronous Operation
1	Synchronous Operation

Registers-UCSRnC

- **USART Control & Status Register C (UCSRnC) (2/4)**

Bit	7	6	5	4	3	2	1	0	
	–	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 5:4 – UPMn1:0: Parity Mode**

These bits enable and set type of parity generation and check. If enabled, the Transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The Receiver will generate a parity value for the incoming data and compare it to the UPMn0 setting. If a mismatch is detected, the UPEn flag in UCSRnA will be set.

Table 78. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	(Reserved)
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

Registers-UCSRnC

- **USART Control & Status Register C (UCSRnC) (3/4)**

- **Bit 3 – USBSn: Stop Bit Select**

This bit selects the number of stop bits to be inserted by the Transmitter. The Receiver ignores this setting.

Table 79. USBSn Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bits

- **Bit 2:1 – UCSZn1:0: Character Size**

The UCSZn1:0 bits combined with the UCSZn2 bit in UCSRnB sets the number of data bits (character size) in a frame the Receiver and Transmitter use.

Table 80. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Registers-UCSRnC

- **USART Control & Status Register C (UCSRnC) (4/4)**

Bit	7	6	5	4	3	2	1	0	
	–	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

- **Bit 0 – UCPOLn: Clock Polarity**

This bit is used for synchronous mode only. Write this bit to zero when Asynchronous mode is used. The UCPOLn bit sets the relationship between data output change and data input sample, and the synchronous clock (XCKn).

Table 81. UCPOLn Bit Settings

UCPOLn	Transmitted Data Changed (Output of TxDn Pin)	Received Data Sampled (Input on RxDn Pin)
0	Rising XCKn Edge	Falling XCKn Edge
1	Falling XCKn Edge	Rising XCKn Edge

Registers–UBRRnL/H

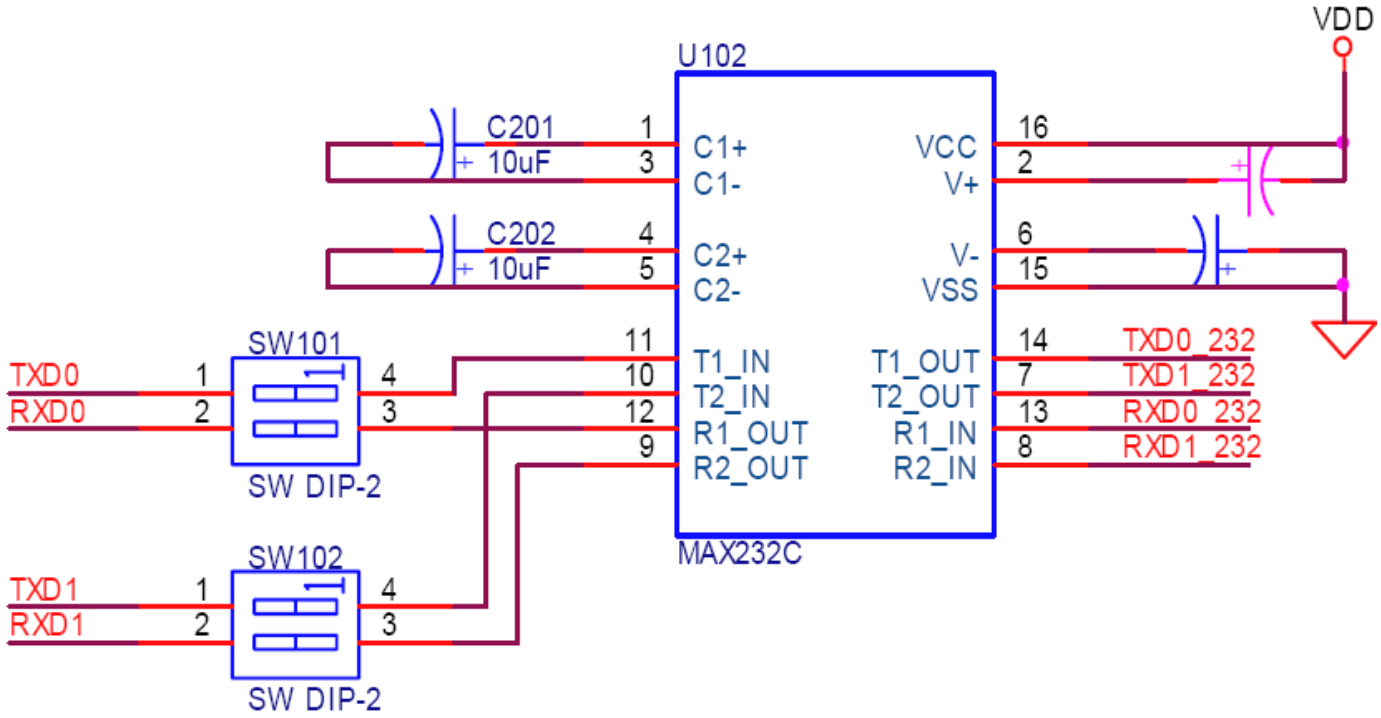
- **USART Baud Rate Registers – UBRRnL & UBRRnH**

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- **Bit 11:0 – UBRRn11:0: USARTn Baud Rate Register**

This is a 12-bit register which contains the USARTn baud rate. The UBRRnH contains the four most significant bits, and the UBRRnL contains the eight least significant bits of the USARTn baud rate. Ongoing transmissions by the transmitter and receiver will be corrupted if the baud rate is changed. Writing UBRRnL will trigger an immediate update of the baud rate prescaler.

Operating Mode	Equation for Calculating Baud Rate[1]	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	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$


VDD
U102
C201
10uF
C202
10uF
C1+
C1-
C2+
C2-
VCC
V+
V-
VSS
T1_IN
T2_IN
R1_OUT
R2_OUT
T1_OUT
T2_OUT
R1_IN
R2_IN
TXD0_232
TXD1_232
RXD0_232
RXD1_232
MAX232C
SW101
SW DIP-2
SW102
SW DIP-2
TXD0
RXD0
TXD1
RXD1

Interrupt Vectors in ATmega128

Table 23. Reset and Interrupt Vectors

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$0000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	$001C	TIMER1 OVF	Timer/Counter1 Overflow
16	$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	$0020	TIMER0 OVF	Timer/Counter0 Overflow

Interrupt Vectors in ATmega128

18	$0022	SPI, STC	SPI Serial Transfer Complete
19	$0024	USART0, RX	USART0, Rx Complete
20	$0026	USART0, UDRE	USART0 Data Register Empty
21	$0028	USART0, TX	USART0, Tx Complete
22	$002A	ADC	ADC Conversion Complete
23	$002C	EE READY	EEPROM Ready
24	$002E	ANALOG COMP	Analog Comparator
25	$0030(3)	TIMER1 COMPC	Timer/Countre1 Compare Match C
26	$0032(3)	TIMER3 CAPT	Timer/Counter3 Capture Event
27	$0034(3)	TIMER3 COMPA	Timer/Counter3 Compare Match A
28	$0036(3)	TIMER3 COMPB	Timer/Counter3 Compare Match B
29	$0038(3)	TIMER3 COMPC	Timer/Counter3 Compare Match C
30	$003A(3)	TIMER3 OVF	Timer/Counter3 Overflow

Interrupt Vectors in ATmega128

Table 23. Reset and Interrupt Vectors (Continued)

Vector No.	Program Address(2)	Source	Interrupt Definition
31	$003C(3)	USART1, RX	USART1, Rx Complete
32	$003E(3)	USART1, UDRE	USART1 Data Register Empty
33	$0040(3)	USART1, TX	USART1, Tx Complete
34	$0042(3)	TWI	Two-wire Serial Interface
35	$0044(3)	SPM READY	Store Program Memory Ready

Notes:
1. When the BOOTRST fuse is programmed, the device will jump to the Boot Loader address at reset, see "Boot Loader Support – Read-While-Write Self-Programming" on page 273.
2. When the IVSEL bit in MCUCR is set, interrupt vectors will be moved to the start of the Boot Flash section. The address of each interrupt vector will then be address in this table added to the start address of the boot Flash section.
3. The Interrupts on address $0030 - $0044 do not exist in ATmega103 compatibility mode.

MCUCR

Bit	7	6	5	4	3	2	1	0	
	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 1 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the interrupt vectors are placed at the start of the Flash memory. When this bit is set (one), the interrupt vectors are moved to the beginning of the Boot Loader section of the flash. The actual address of the start of the Boot Flash section is determined by the BOOTSZ fuses. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 273 for details. To avoid unintentional changes of interrupt vector tables, a special write procedure must be followed to change the IVSEL bit:

1. Write the Interrupt Vector Change Enable (IVCE) bit to one.
2. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling.

Note: If interrupt vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If interrupt vectors are placed in the Application section and Boot Lock bit BLB12 is programed, interrupts are disabled while executing from the Boot Loader section. Refer to the section "Boot Loader Support – Read-While-Write Self-Programming" on page 273 for details on Boot Lock bits.

MCUCR

- **Bit 0 – IVCE: Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below.

Assembly Code Example

```
Move_interrupts:
  ; Enable change of interrupt vectors
  ldi  r16, (1<<IVCE)
  out  MCUCR, r16
  ; Move interrupts to boot flash section
  ldi  r16, (1<<IVSEL)
  out  MCUCR, r16
  ret
```

C Code Example

```
void Move_interrupts(void)
{
  /* Enable change of interrupt vectors */
  MCUCR = (1<<IVCE);
  /* Move interrupts to boot flash section */
  MCUCR = (1<<IVSEL);
}
```

Sample Code(1)

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <string.h>
#define BYTE  unsigned char

unsigned char rxd_data=0x00;
unsigned char txd_data=0x00;

/* USART 0 receive interrupt vector */
SIGNAL(SIG_UART0_RECV)
{
      rxd_data = UDR0;
      txd_data = rxd_data;
      UDR0 = txd_data; /* echo back received data */
}
void TXChar(BYTE tx_data)
{
  /* wait until transmitter buffer is ready */
      while((UCSR0A&0x20) == 0x00);
      UDR0 = tx_data; /* send data to transmitter buffer */
}
```

Sample Code(2)

```
void PutChar(BYTE *databuf,BYTE num)
{
	BYTE CharCount;
	BYTE i;

	CharCount = num			;
	if( !CharCount ) CharCount = strlen( (char *)databuf );
	for(i=0; i<CharCount; i++){
		TXChar(databuf[i]);
	}
}
```

Sample Code(3)

```
int main(void)
{

    cli(); /* disable interrupt */

    MCUCR = 0x01; /* Enable change of interrupt vectors IVCE=1 */
    MCUCR = 0x00; /* interrupt vector to flash start address IVSEL=0 */

    /* USART0 initialization */
    UCSR0A = 0x00;
    UCSR0B = 0x98;
    UCSR0C = 0x06;
    UBRR0H = 0x00; /* baud rate 57600 UBRR0=16*/
    UBRR0L = 0x10;

    PutChar((unsigned char *)"Serial Communication OK\r\n", 0);

    sei(); /* enable interrupt */

    while(1);
}
```

Control Register Setting

UCSR0B = 0x98;

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – RXCIEn: RX Complete Interrupt Enable

Bit 4 – RXENn: Receiver Enable

Bit 3 – TXENn: Transmitter Enable

UCSR0C = 0x06;

Bit	7	6	5	4	3	2	1	0	
	–	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

Table 78. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	(Reserved)
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

Table 79. USBSn Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bits

Table 80. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Baud Rate

UBRR0H = 0x00; /* baud rate 57600 UBRR0=16*/

UBRR0L = 0x10;

Table 85. Examples of UBRR Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	f_{osc} = 16.0000 MHz				f_{osc} = 18.4320 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%
0.5M	1	0.0%	3	0.0%	–	–	4	-7.8%
1M	0	0.0%	1	0.0%	–	–	–	–
Max (1)	1 Mbps		2 Mbps		1.152 Mbps		2.304 Mbps	