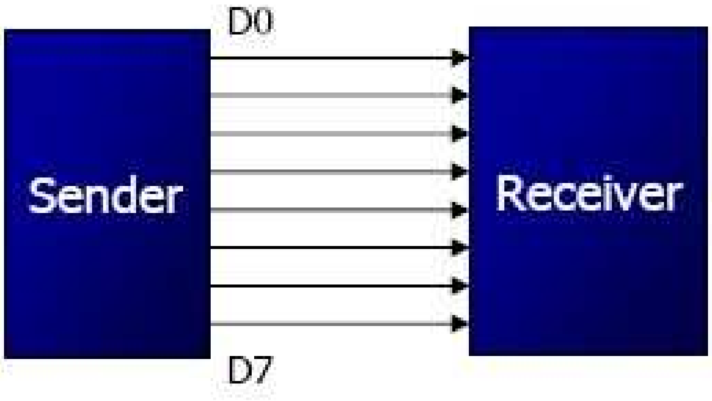
Serial Communication in AVR

Parallel: Often 8 or more lines are used.

Example: printer

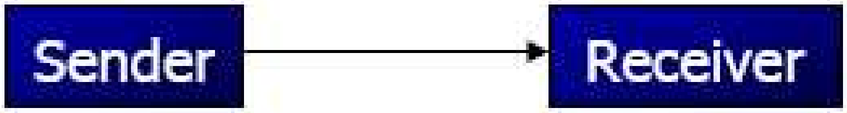


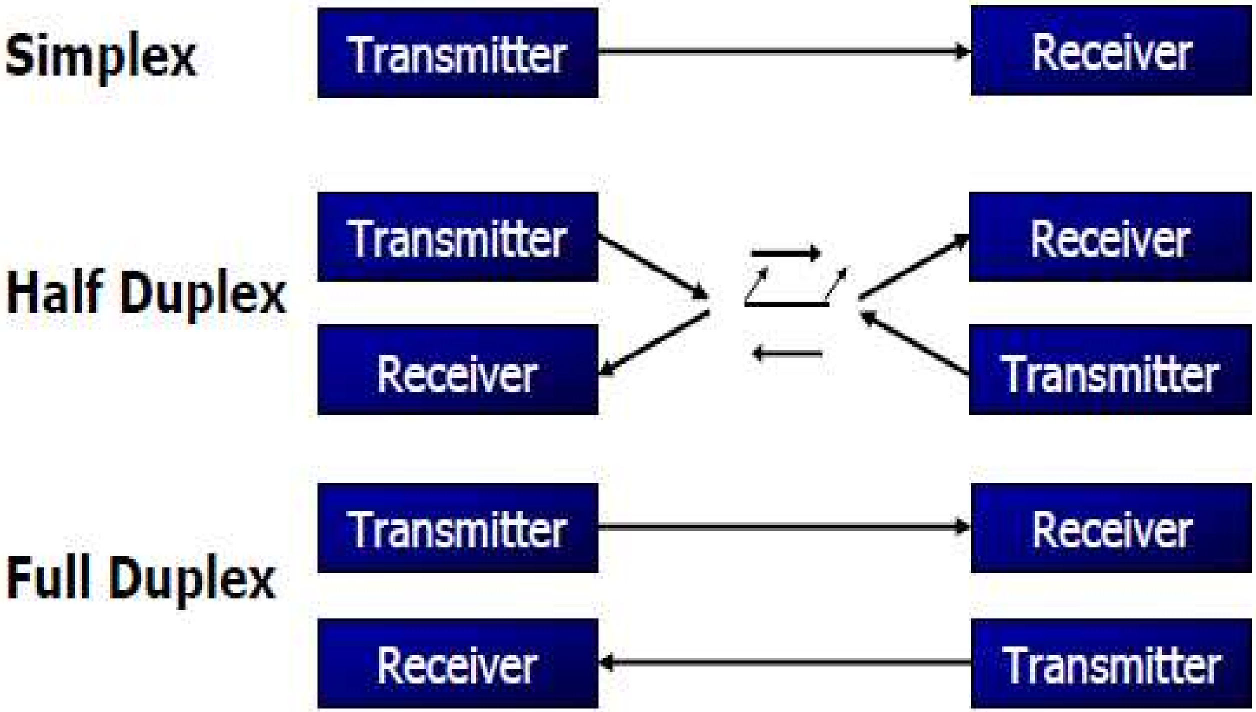
Serial

To transfer to a device located many meters away, the serial method is used.

The data is sent one bit at a time.

Example: Telephone lines





Simplex

Transmitter

Receiver

Half Duplex

Transmitter

Receiver

Receiver

Transmitter

Full Duplex

Transmitter

Receiver

Receiver

Transmitter

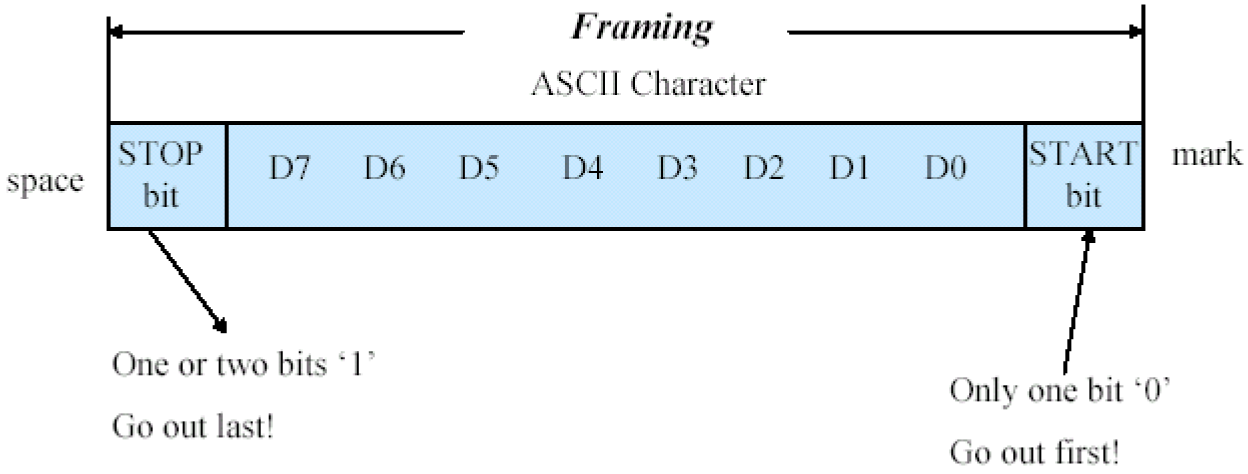
Serial data communication uses two methods:

- Synchronous method transfers a block of data at a time
- Asynchronous method transfers a single bit at a time

- Special IC chips: UART (universal asynchronous Receiver transmitter)
- USART (universal synchronous-asynchronous Receiver-transmitter)

Asynchronous Communication

- Start and stop bits



Framing

ASCII Character

space

STOP bit	D7	D6	D5	D4	D3	D2	D1	D0	START bit

mark

One or two bits ‘1’

Go out last!

Only one bit ‘0’

Go out first!

UART (Universal Asynchronous Receiver Transmitter) or USART (Universal Synchronous Asynchronous Receiver Transmitter) is one of the basic interface. This interface provide a cost effective simple and reliable communication between one controller to another controller or between a controller and PC.

RS-232 (Recommended Standard 232) is a standard for serial binary data signals connecting between a DTE (Data terminal equipment) and a DCE (Data Circuit-terminating Equipment).

RS-232 Voltage Levels

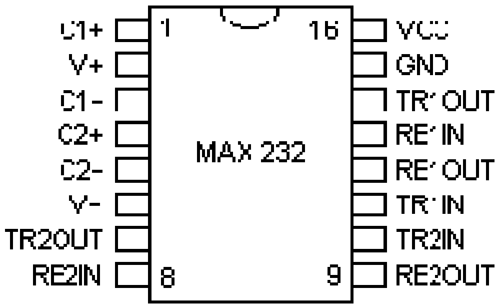
- The RS-232 standard defines the voltage levels that correspond to logical one and logical zero levels. Valid signals are plus or minus 3 to 25 volts.
- The range near zero volts is not a valid RS-232 level
- Logic one is defined as a negative voltage, the signal condition is called marking, and has the functional significance of OFF.
- Logic zero is positive, the signal condition is spacing, and has the function ON.
- So a Logic Zero represented as +3V to +25V and Logic One represented as -3V to -25V.

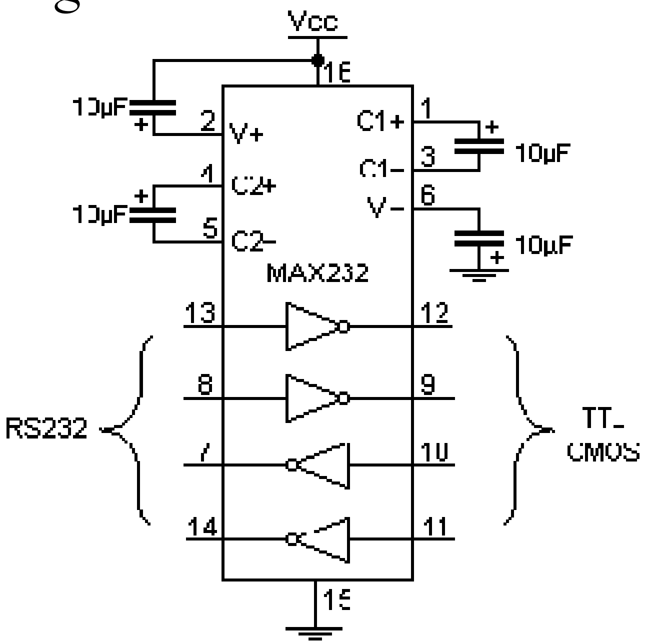
RS-232 Level Converters

- Usually all the digital ICs works on TTL or CMOS voltage levels which cannot be used to communicate over RS-232 protocol. So a voltage or level converter is needed which can convert TTL to RS232 and RS232 to TTL voltage levels.
- The most commonly used RS-232 level converter is MAX232. This IC includes charge pump which can generate RS232 voltage levels (-10V and +10V) from 5V power supply.
- It also includes two receiver and two transmitters and is capable of full-duplex UART/USART communication.

MAX 232

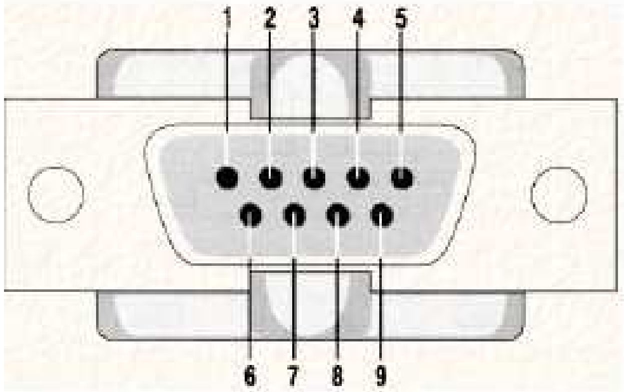
- To communicate over UART or USART, we just need three basic signals which are namely, RXD (receive), TXD (transmit), GND (common ground). So to interface MAX232 with 8051 microcontroller we just need the basic signals.

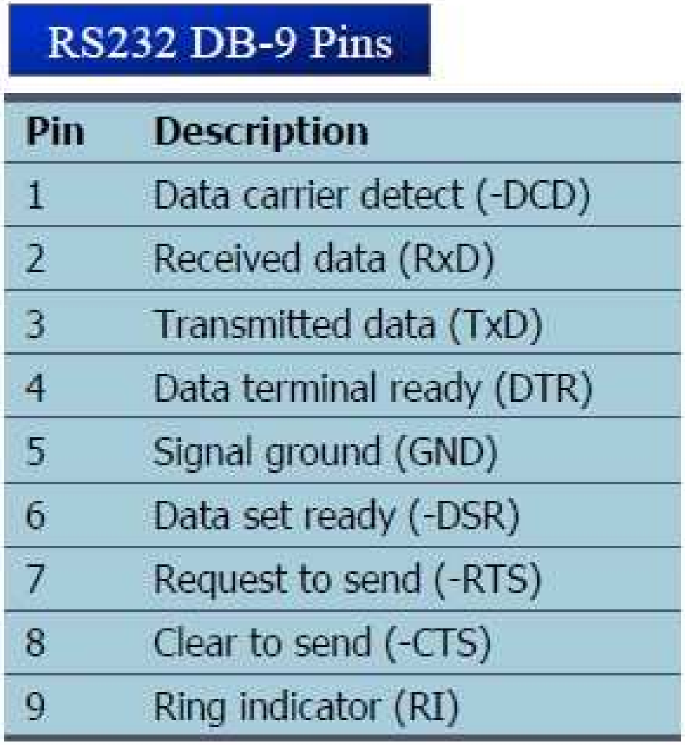




DB 9 Connector

RS232 Connector DB-9





RS232 DB-9 Pins

Pin	Description
1	Data carrier detect (-DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (-DSR)
7	Request to send (-RTS)
8	Clear to send (-CTS)
9	Ring indicator (RI)

UART in AVR

- Port D pin is used as UART pin dedicated on PD0 and PD1 known as TX and RX respectively
- Baud rate is an important parameter while working with serial communication
- Master and slave must have same baud rate
- Five registers are associated with AVR UART
- UDR (USART Data register) , UCSRA (USART Control Register), UCSRB, UCSRC and UBRR (UART Baud Rate Register)

The control registers specify:

- The mode of operation: synchronous or asynchronous
- Parity bit: odd or even
- Information unit: 5,6,7,8, or 9 bits
- Information unit separation: how to specify the transmission of a word starts and stops?
- Transmission rate

Parity bit

- A way to detect error during data transmission
  - Due to external noises
- Even parity= the number of 1s must be an even number
- Add an extra bit to the 8-bit data, called parity bit
- How does it work?
  - if the number of 1s is already even, set it to 0, otherwise to 1
  - Send the parity bit with data
  - If the other side detects odd number of 1s, there is something wrong

UDR

- **USART data register**

- **Actually two registers: one for the transmit direction, the other for receive direction**
- **Share the same address and name**
  - – When UDR is read, the data received from the serial line is returned
  - – When a data is written to UDR, it is directed to the transmit line

UBRR (UART Baud Rate Register)

- Used for communication between two serially connected device e.g. PC and Microcontroller
- Baud rate is programmable in AVR varies from 1200 bps to 115200 bps
- Crystal Frequency is an important parameter for baud rate calculation
- UBBR is a 16 bit register and can be used as UBRRL and UBRRH as two 8 bit register

Steps for baud rate calculation

- Desired Baud rate = Crystal Frequency/(16(X+1)
  - X is the value to be loaded into UBRR

- Can be modified to
  - X = (Crystal Frequency/(16*Baudrate))-1

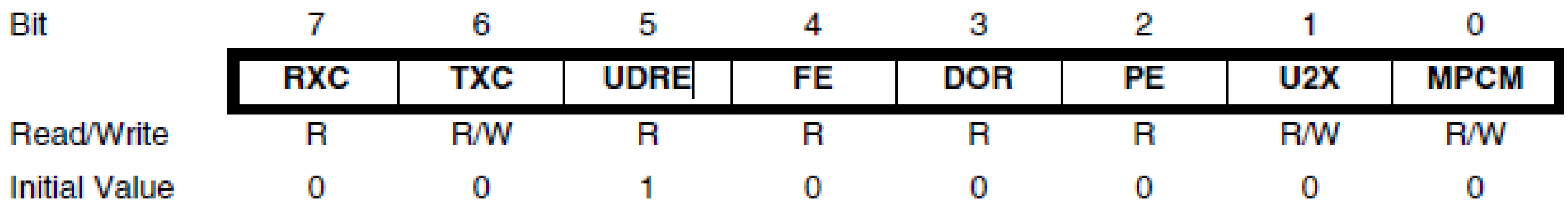
Baud rate for 8MHZ crystal

Baud rate	UBRR (decimal value)	UBRR (Hex value)
1200	415	19F
2400	207	CF
4800	103	67
9600	51	33
19200	25	19
38400	12	C

UCSR

- UCSR are 8 bit registers used for controlling serial communication in AVR
- Three USART control registers are used UCSRA, UCSRA, UCSRB, UCSRC

UCSRA



Bit	7	6	5	4	3	2	1	0
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
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 – RXC: 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

• Bit 6 – TXC: 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 (UDR).

• Bit 5 – UDRE: USART Data Register Empty

The UDRE Flag indicates if the transmit buffer (UDR) is ready to receive new data. If UDRE is one, the buffer is empty, and therefore ready to be written.

• Bit 4 – FE: Frame Error

This bit is set if the next character in the receive buffer had a Frame Error when received.

Bit 3 – DOR: Data OverRun

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..

- **Bit 2 – PE: Parity Error**

This bit is set if the next character in the receive buffer had a Parity Error.

- **Bit 1 – U2X: Double the USART Transmission Speed**

This bit only has effect for the asynchronous operation. Writing this bit to one will reduce the divisor of the baud rate divider from 16 to 8 effectively doubling

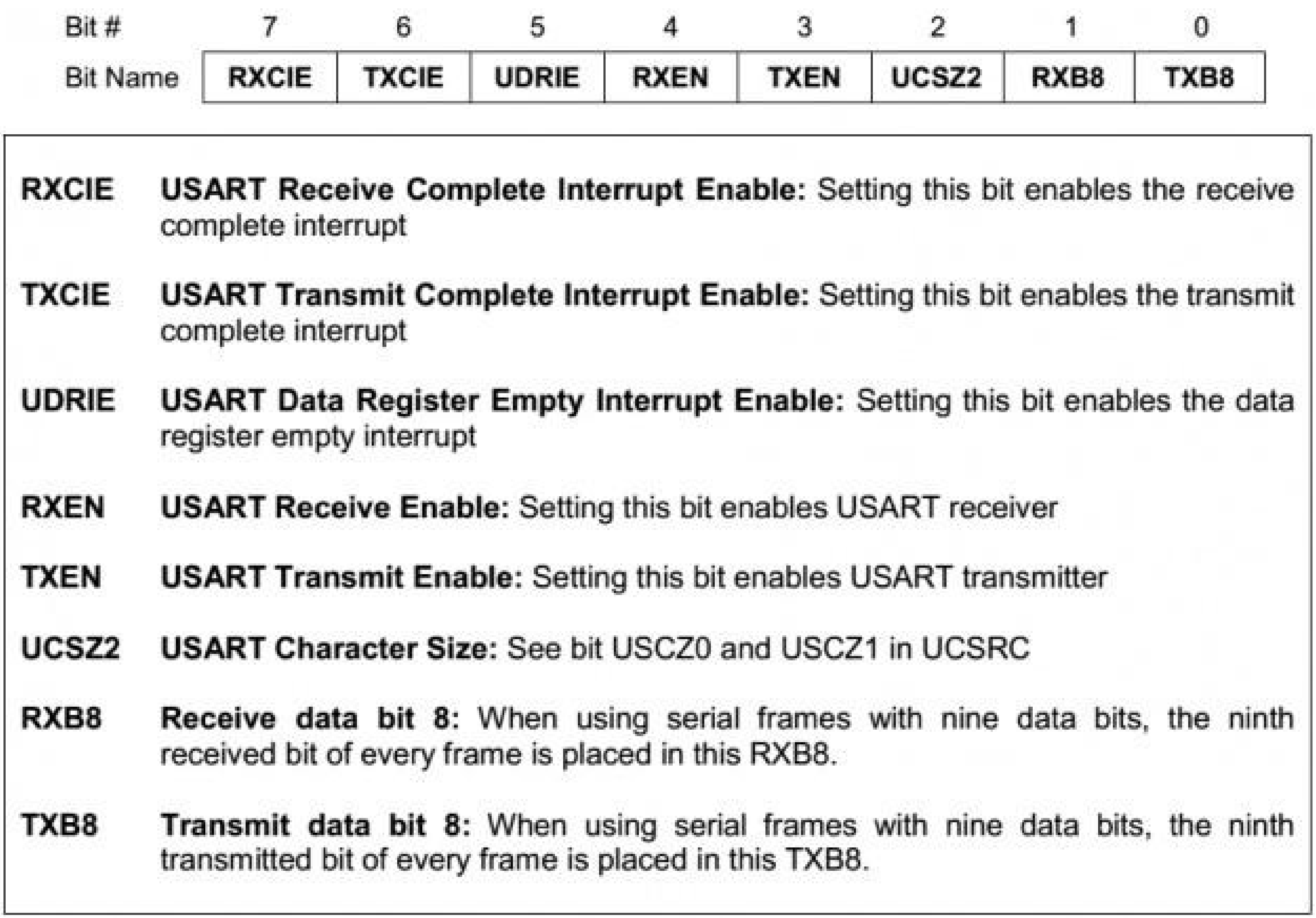
the transfer rate for asynchronous communication.

- **Bit 0 – MPCM: Multi-processor Communication Mode**

This bit enables the Multi-processor Communication mode.

- Bit 7 is set when no new data is received in receive buffer. It is used to generate receive complete interrupt
- Bit 6 is used to indicate that complete data are transmitted. It is used to generate transmit complete interrupt
- Bit 5 is USART Data register Empty. Used to indicate that the transmit buffer is ready to receive new data
- Bit 4 Frame error
- Bit 3 Data Overrun. Occurs when RXB and RX buffer are full and yet new data are available to read
- Bit 2 Parity Error
- Bit 1 U2X Double the transmission speed
- MPCM

UCSRB



Bit #	7	6	5	4	3	2	1	0
Bit Name	**RXCIE**	**TXCIE**	**UDRIE**	**RXEN**	**TXEN**	**UCSZ2**	**RXB8**	**TXB8**

RXCIE **USART Receive Complete Interrupt Enable:** Setting this bit enables the receive complete interrupt

TXCIE **USART Transmit Complete Interrupt Enable:** Setting this bit enables the transmit complete interrupt

UDRIE **USART Data Register Empty Interrupt Enable:** Setting this bit enables the data register empty interrupt

RXEN **USART Receive Enable:** Setting this bit enables USART receiver

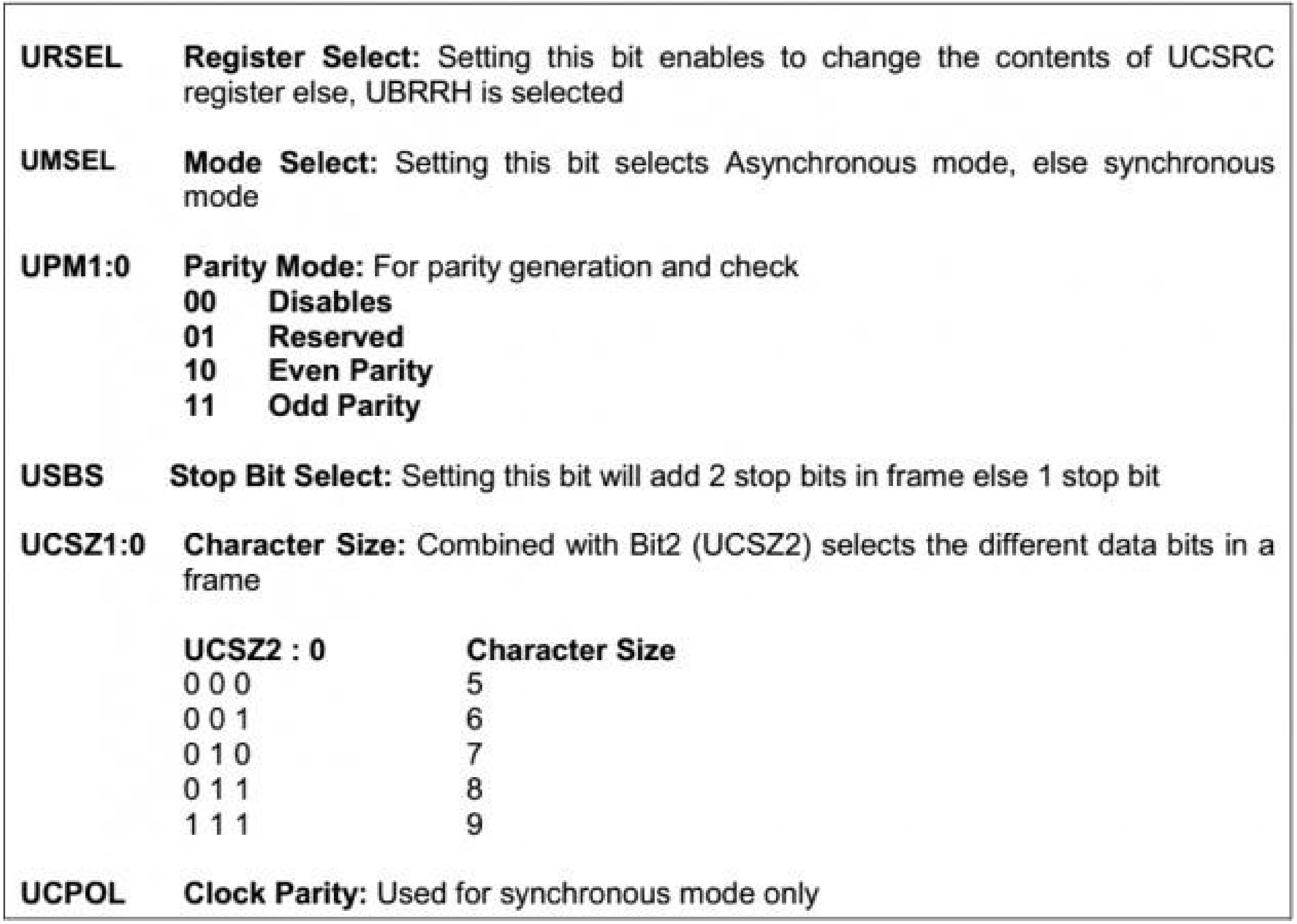
TXEN **USART Transmit Enable:** Setting this bit enables USART transmitter

UCSZ2 **USART Character Size:** See bit USCZ0 and USCZ1 in UCSRC

RXB8 **Receive data bit 8:** When using serial frames with nine data bits, the ninth received bit of every frame is placed in this RXB8.

TXB8 **Transmit data bit 8:** When using serial frames with nine data bits, the ninth transmitted bit of every frame is placed in this TXB8.

7	6	5	4	3	2	1	0
URSEL	UMSEL	UPM1	UPM0	USBS	USCZ1	USCZ0	UCPOL



URSEL **Register Select:** Setting this bit enables to change the contents of UCSRC register else, UBRRH is selected

UMSEL **Mode Select:** Setting this bit selects Asynchronous mode, else synchronous mode

UPM1:0 **Parity Mode:** For parity generation and check

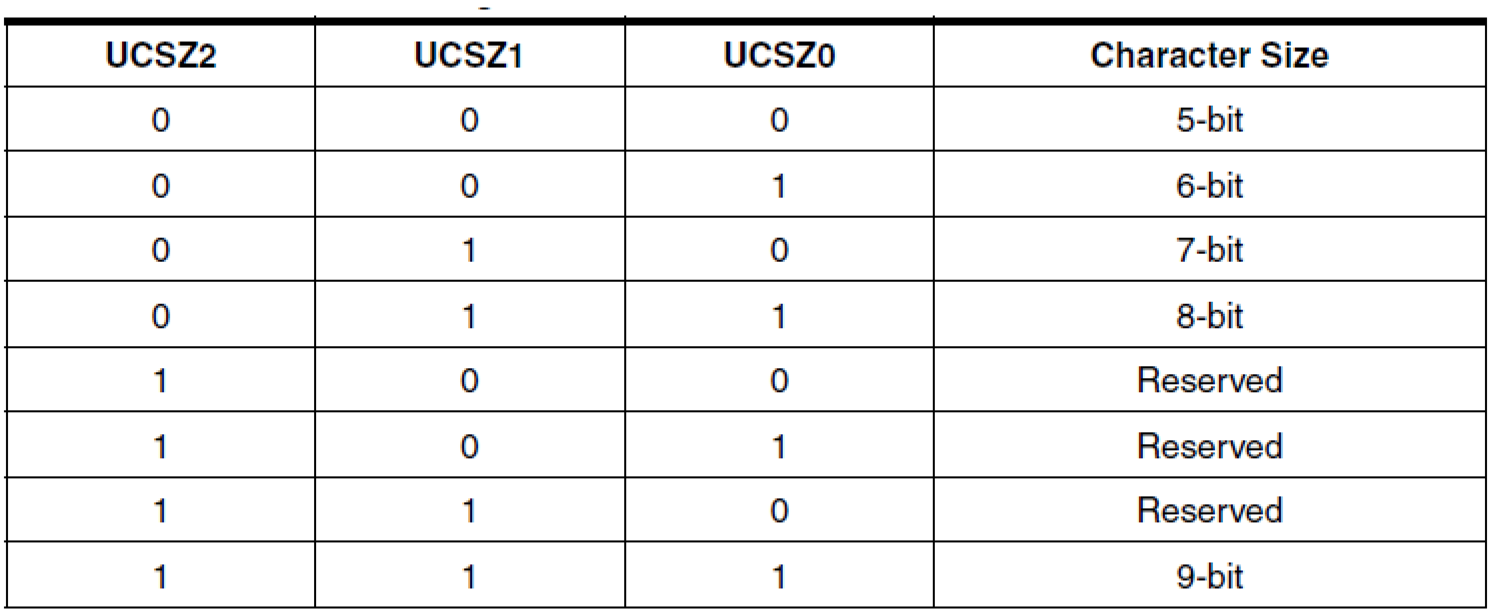
00	**Disables**
01	**Reserved**
10	**Even Parity**
11	**Odd Parity**

USBS **Stop Bit Select:** Setting this bit will add 2 stop bits in frame else 1 stop bit

UCSZ1:0 **Character Size:** Combined with Bit2 (UCSZ2) selects the different data bits in a frame

UCSZ2 : 0	**Character Size**
0 0 0	5
0 0 1	6
0 1 0	7
0 1 1	8
1 1 1	9

UCPOL **Clock Parity:** Used for synchronous mode only



UCSZ2	UCSZ1	UCSZ0	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

Steps to transmit data serially in AVR

Load UCSRB with specific value

Load UCSRC for required data format

Load UBRR register for the required baudrate

Monitor UDRE bit of UCSRA for transmission and reception