CSE 315
Microprocessors & Microcontrollers

Tanvir Ahmed Khan

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology.

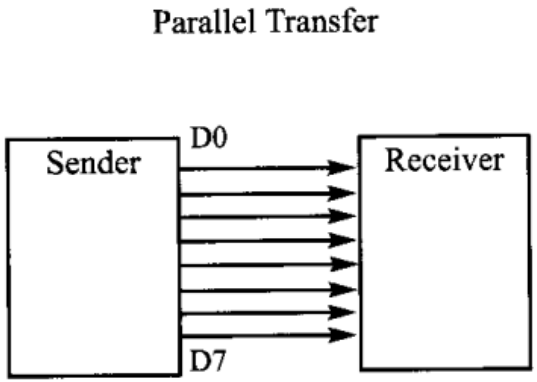
October 28, 2014

Recap

Data Communications

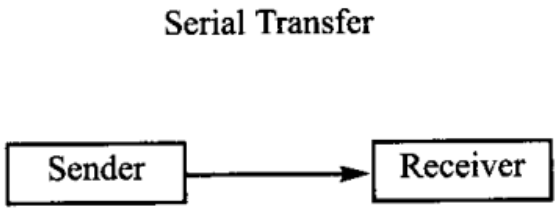
- Computers transfer data in 2 ways,
  - Parallel Communication
  - Serial Communication

Parallel Data Communications



- multiple wire lines are used to transfer data
- Advantage,
  - Speed
- Disadvantage,
  - Distance cannot be great
- Example, computer to printer data transfer

Serial Data Communications



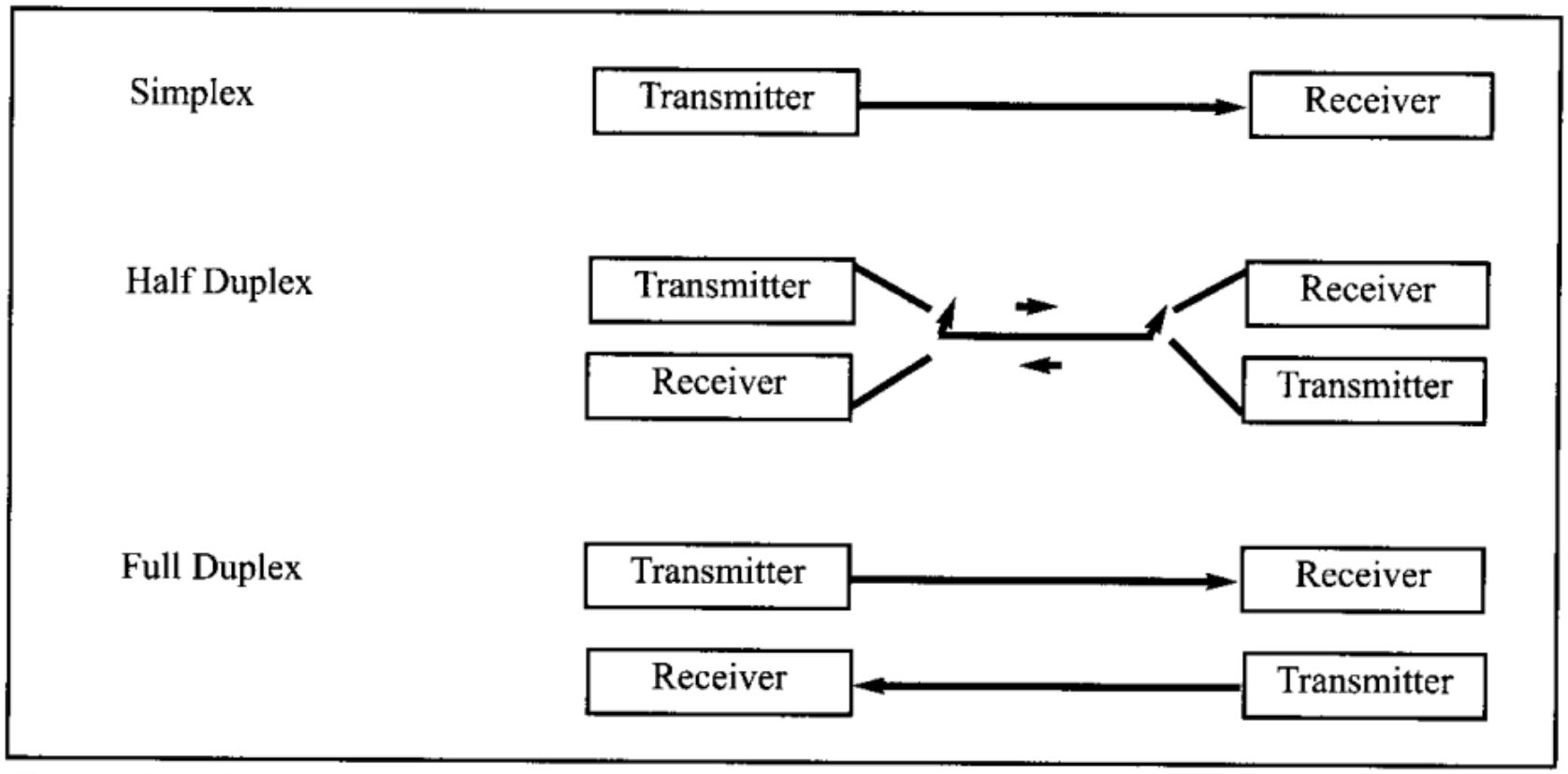
- data is sent one bit at a time
- Advantage,
  - larger distances
  - cheaper
  - fewer I/O pins
  - easy synchronization
- Disadvantage,
  - relatively slower
- Example, USB

Synchronization in Serial Communications

Two Methods

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

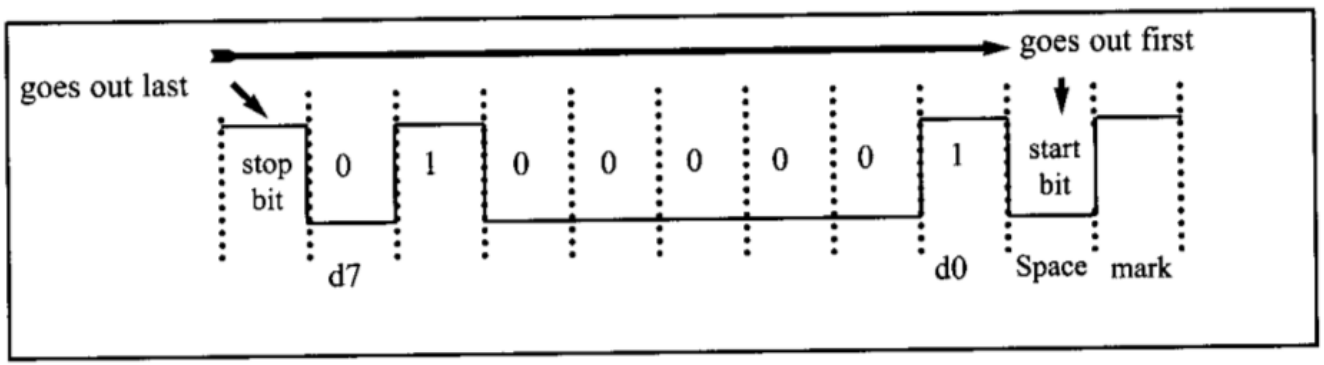
Simplex, Half- & Full-Duplex Data Transfer



Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits
- *Start bit*
  - always one bit
  - always 0(low)
- *Stop bit*
  - can be one or two bits
  - always 1(high)



Framing of 'A'(0*x*41)

Data Transfer Rate

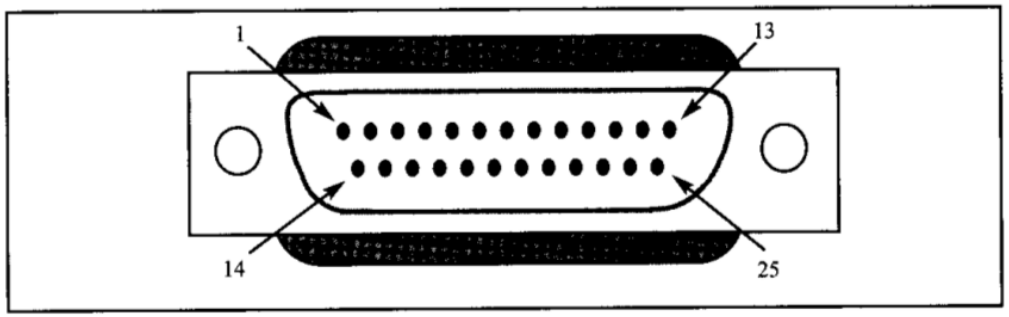
- *bps*
  - bits per second
- *baud rate*
  - number of signal changes per second
- generally, bps != baud rate
  - for some coding system, bps == baud rate

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard
- **not compatible with TTL family**
  - $0 = +3$ to $+25$ V
  - $1 = -3$ to -25 V
- we will need voltage converter
  - MAX232

RS232 Standards

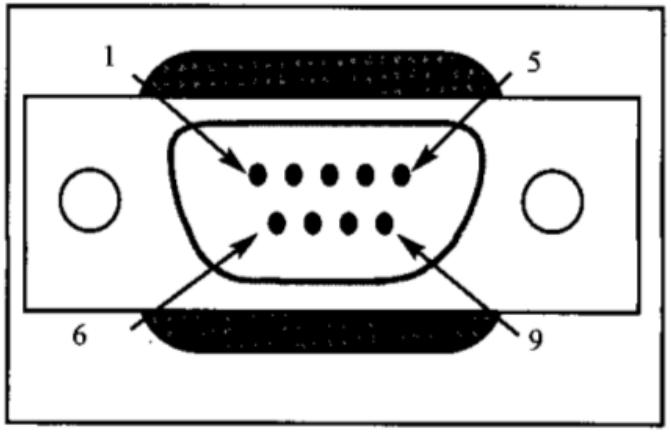
Continued



Original RS232 Connector DB-25

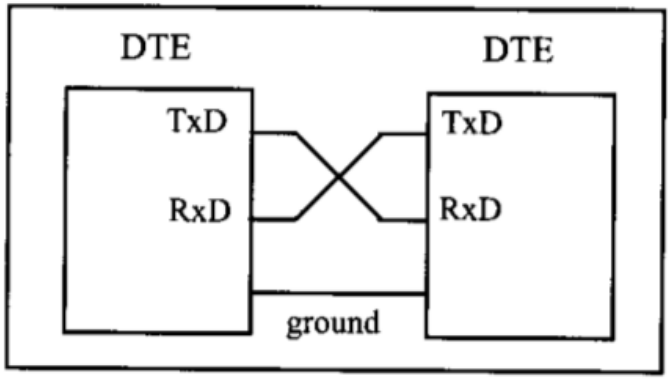
RS232 Standards

Continued



9-pin Connector for DB-9

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)



Today's Topic

ATmega16/32 Serial Communication Programming

Today's Problem

Hack the MCQ Exam

Problem Specification

Problem Specification

- Both Receive & Transmit

Problem Specification

- Both Receive & Transmit
- Frame,

Problem Specification

- Both Receive & Transmit
- Frame,
  - 1-byte(8-bit) char data

Problem Specification

- Both Receive & Transmit
- Frame,
  - 1-byte(8-bit) char data
  - 1 start bit

Problem Specification

- Both Receive & Transmit
- Frame,
  - 1-byte(8-bit) char data
  - 1 start bit
  - 1 stop bit

Problem Specification

- Both Receive & Transmit
- Frame,
  - 1-byte(8-bit) char data
  - 1 start bit
  - 1 stop bit
- Baud Rate, 4800

Problem Specification

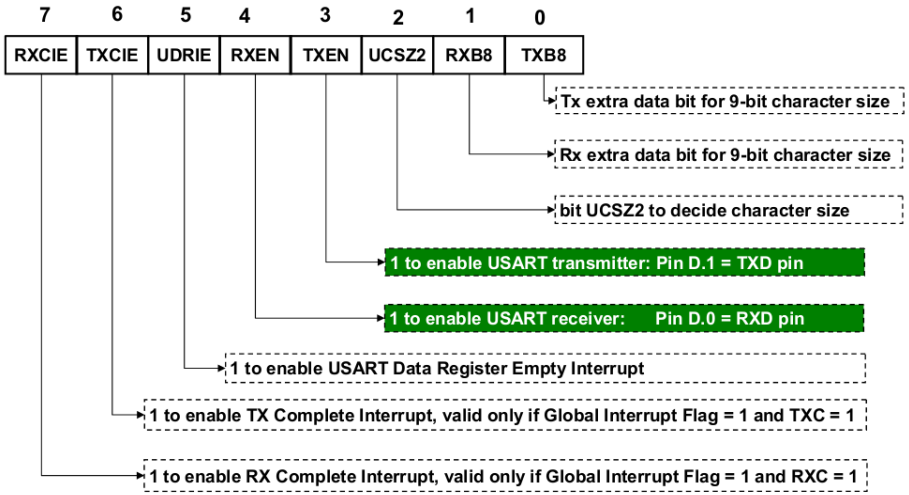
- Both Receive & Transmit
- Frame,
  - 1-byte(8-bit) char data
  - 1 start bit
  - 1 stop bit
- Baud Rate, 4800
- First Receive & then, Send

Problem Specification

- Both Receive & Transmit
- Frame,
  - 1-byte(8-bit) char data
  - 1 start bit
  - 1 stop bit
- Baud Rate, 4800
- First Receive & then, Send
- PC Serial COM port follows RS232 Standard

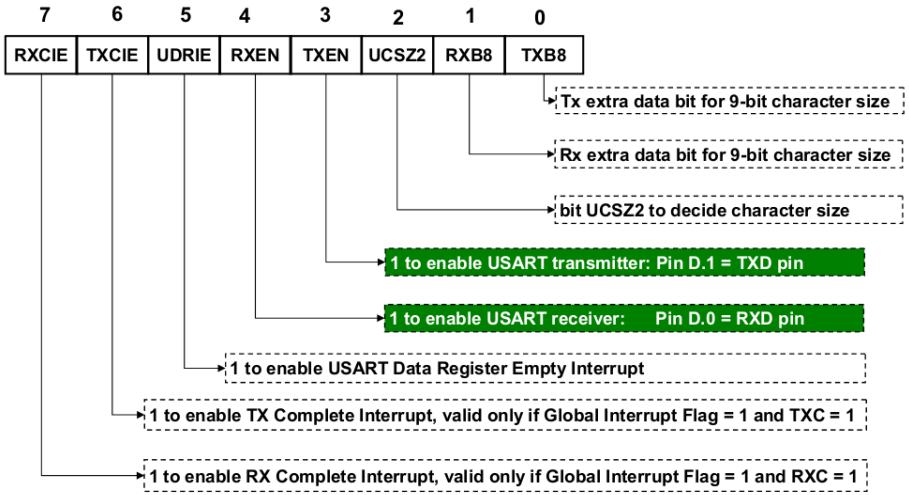
Enable Receive & Transmit

Enable Receive & Transmit



UCSRB Register

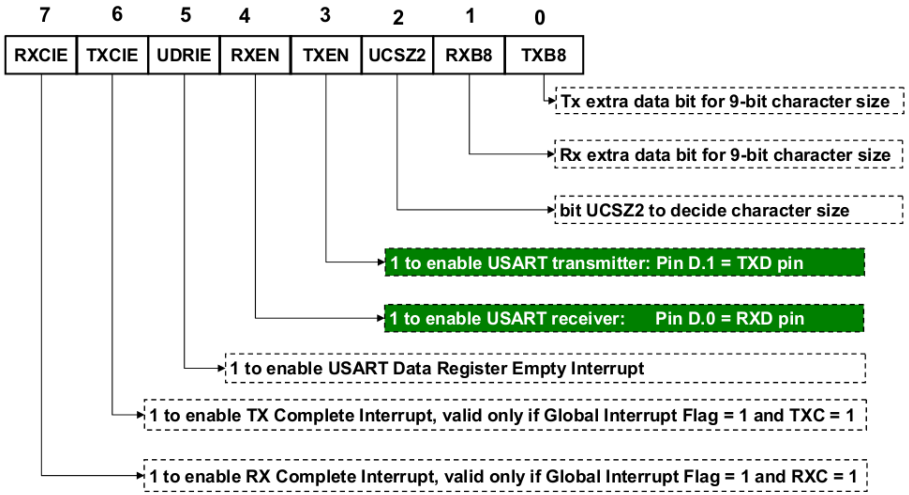
Enable Receive & Transmit



UCSRB Register

- `UCSRB = UCSRB | (1 << RXEN);`

Enable Receive & Transmit

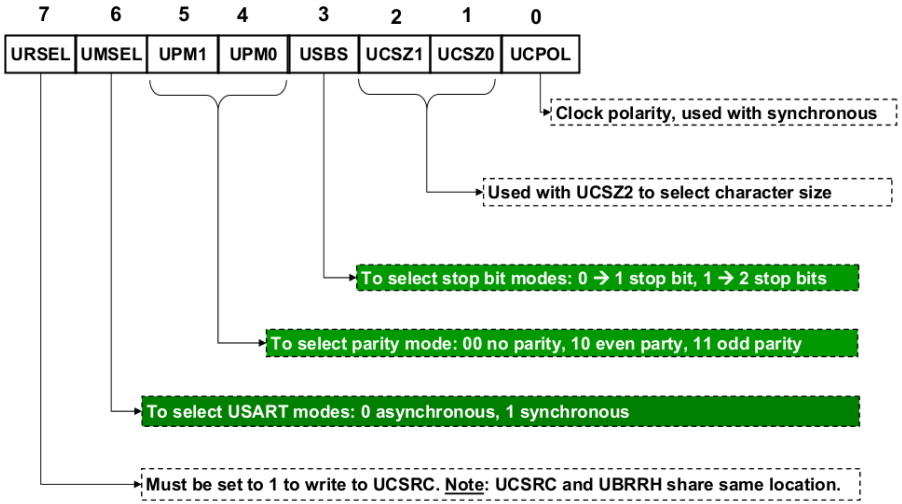


UCSRB Register

- `UCSRB = UCSRB | (1 << RXEN);`
- `UCSRB = UCSRB | (1 << TXEN);`

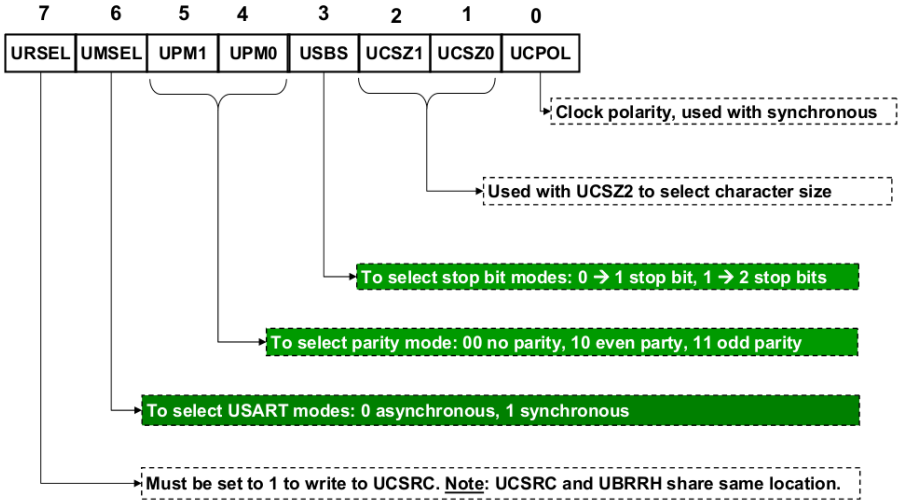
Framing

Framing



UCSRC Register

Framing



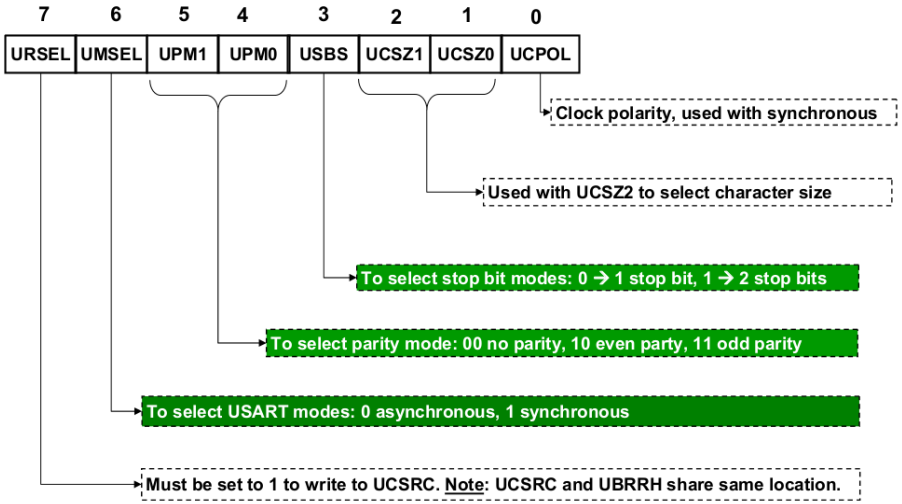
UCSRC Register

Table 11-5: Values of UCSZ2:0 for Different Character Sizes

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	1	1	9

Note: Other values are reserved. Also notice that UCSZ0 and UCSZ1 belong to UCSRC and UCSZ2 belongs to UCSRB

Framing



UCSRC Register

Table 11-5: Values of UCSZ2:0 for Different Character Sizes

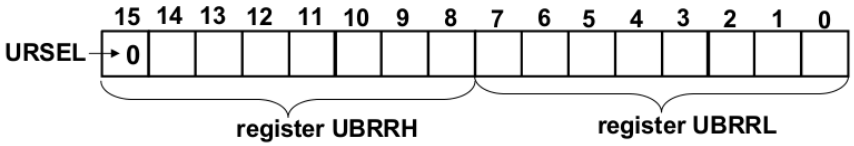
UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5
0	0	1	6
0	1	0	7
0	1	1	8
1	1	1	9

Note: Other values are reserved. Also notice that UCSZ0 and UCSZ1 belong to UCSRC and UCSZ2 belongs to UCSRB

- `UCSRC = UCSRC | (1 << UCSZ1) | (1 << UCSZ0) | (1<<URSEL);`

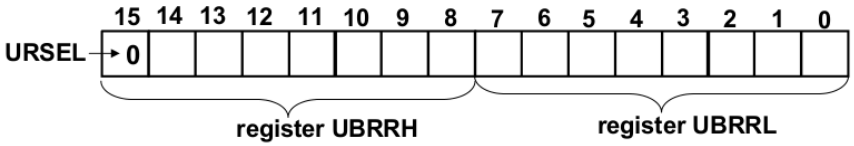
Configuring Baud Rate

Configuring Baud Rate



UBRR Register

Configuring Baud Rate

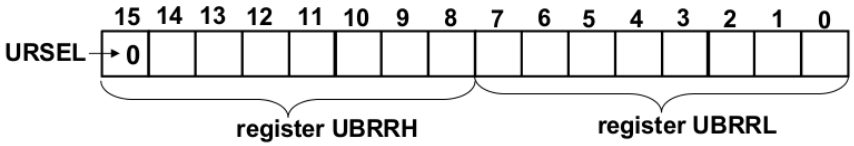


UBRR Register

$$\text{baud rate} = \frac{\text{system clock frequency (Hz)}}{16(\text{UBRR} + 1)}$$

$$\text{UBRR} = \frac{\text{system clock frequency (Hz)}}{16 \times \text{baud rate}} - 1$$

Configuring Baud Rate



UBRR Register

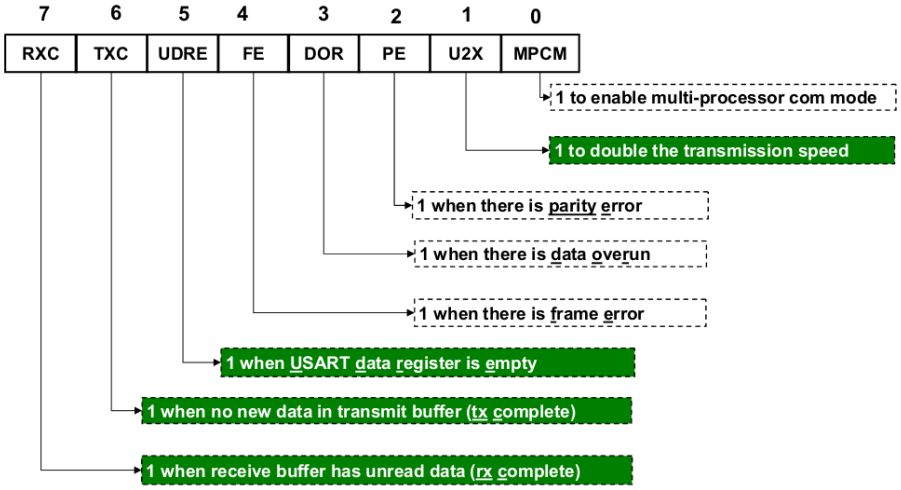
$$\text{baud rate} = \frac{\text{system clock frequency (Hz)}}{16(\text{UBRR} + 1)}$$

$$\text{UBRR} = \frac{\text{system clock frequency (Hz)}}{16 \times \text{baud rate}} - 1$$

- `UBRRl = 0x67;`

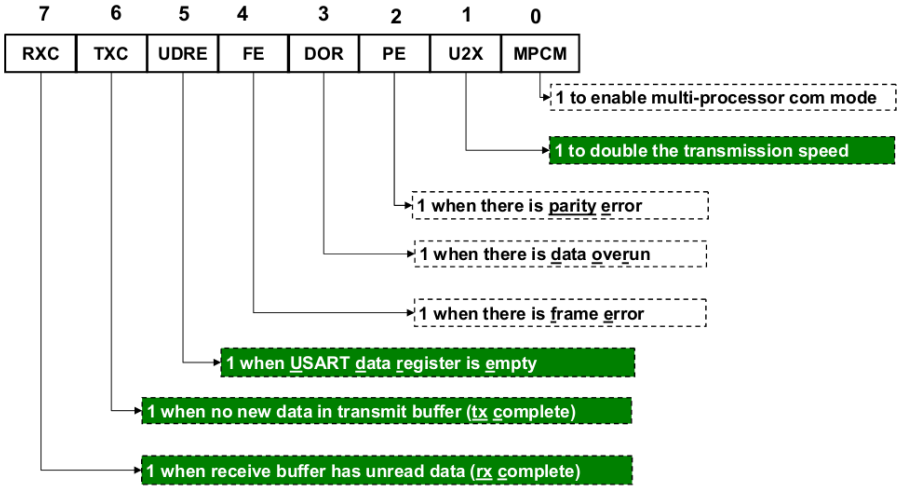
Wait, Receive & then, Wait, Send

Wait, Receive & then, Wait, Send



UCSRA Register

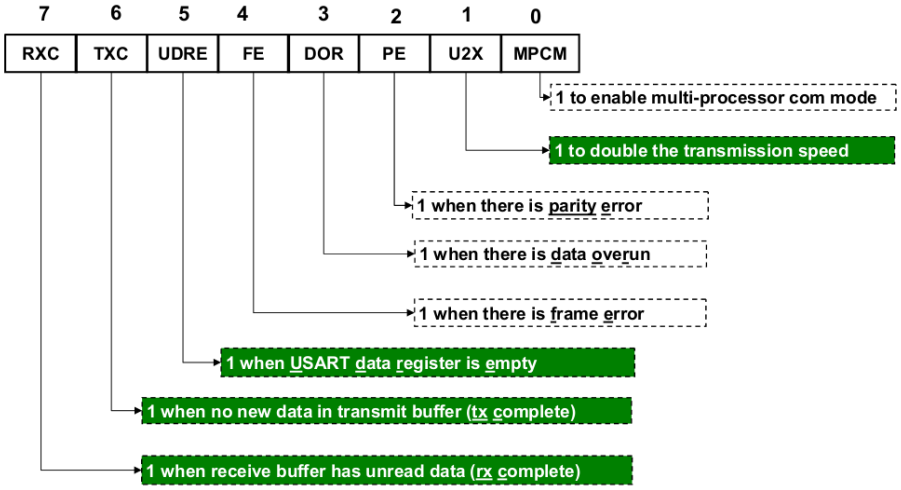
Wait, Receive & then, Wait, Send



UCSRA Register

- `while((UCSRA & (1 << RXC)) == 0);`

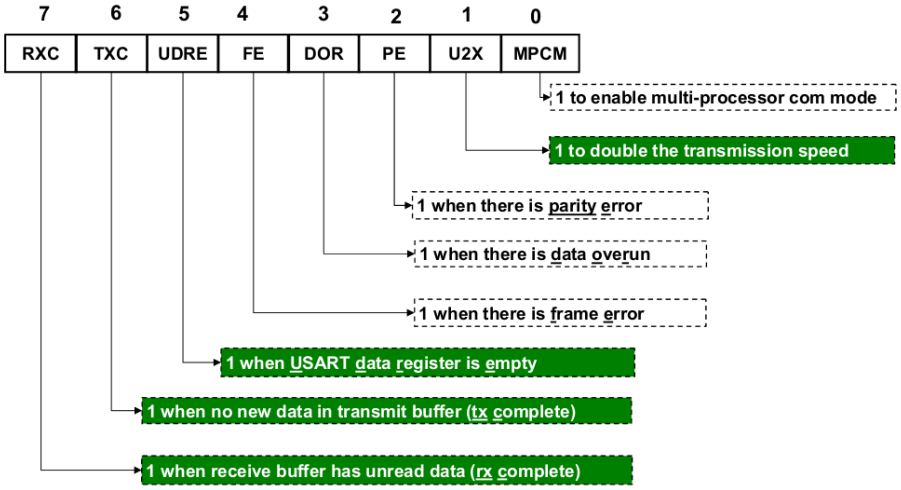
Wait, Receive & then, Wait, Send



UCSRA Register

- `while((UCSRA & (1 << RXC)) == 0);`
- `ch = UDR;`

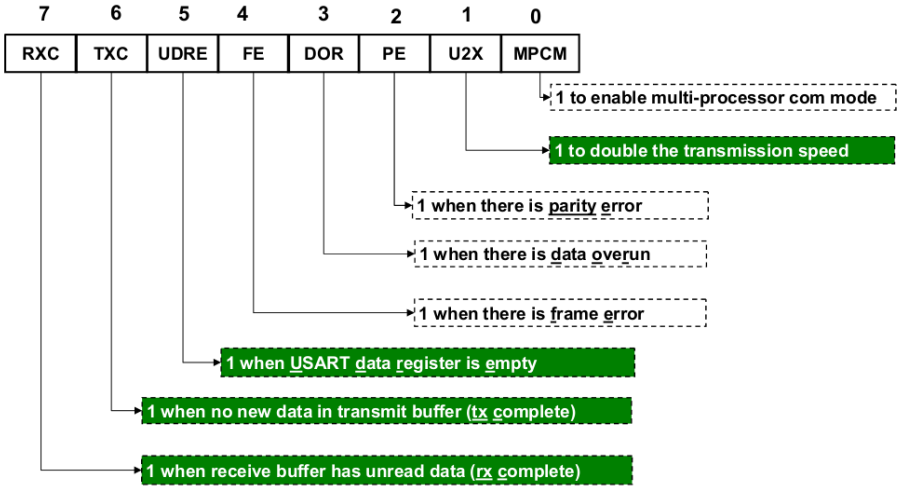
Wait, Receive & then, Wait, Send



UCSRA Register

- `while((UCSRA & (1 << RXC)) == 0);`
- `ch = UDR;`
- `while((UCSRA & (1 << UDRE)) == 0);`

Wait, Receive & then, Wait, Send



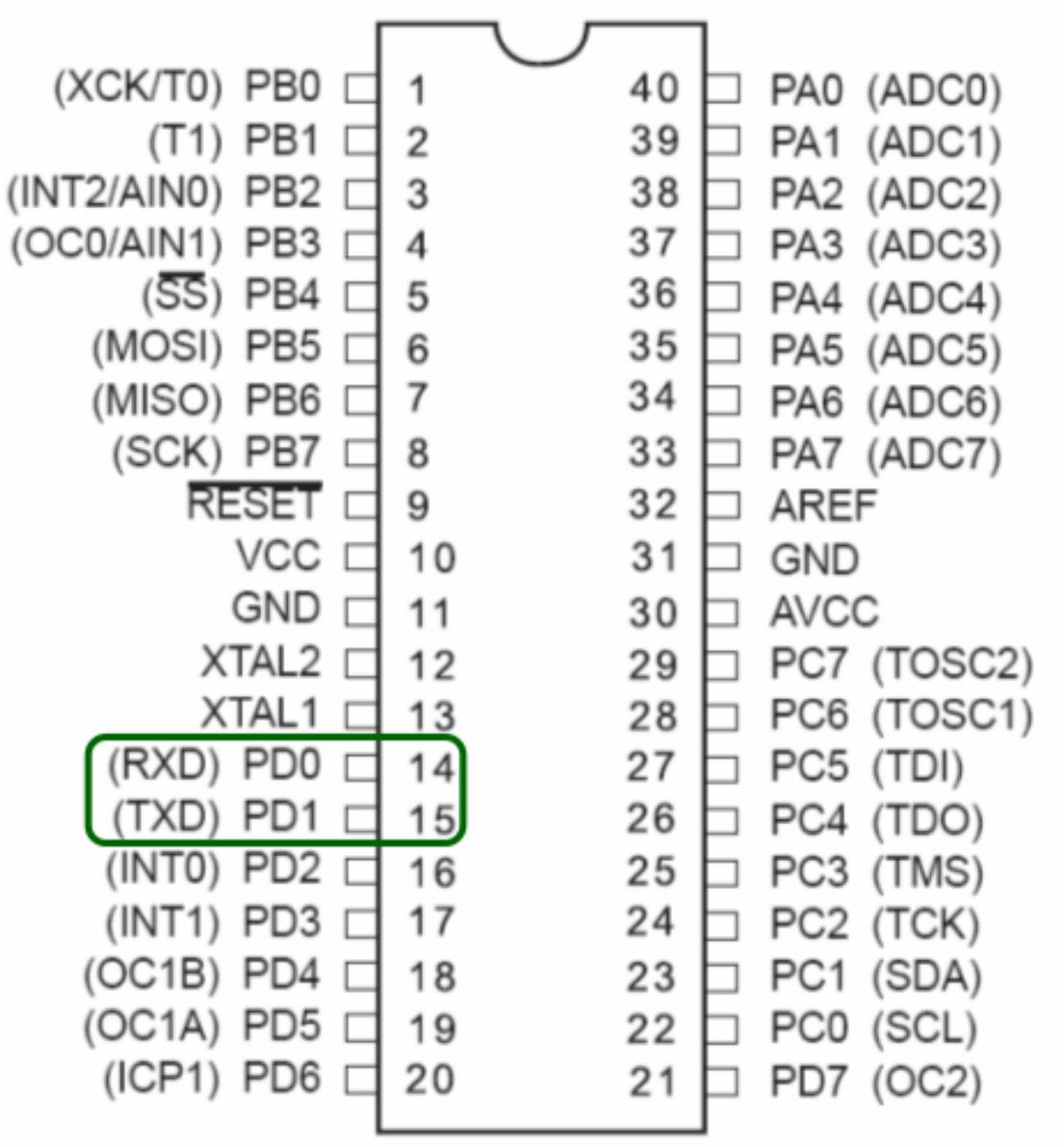
UCSRA Register

- `while((UCSRA & (1 << RXC)) == 0);`
- `ch = UDR;`
- `while((UCSRA & (1 << UDRE)) == 0);`
- `UDR = result[(ch - '0') % 10];`

The Program

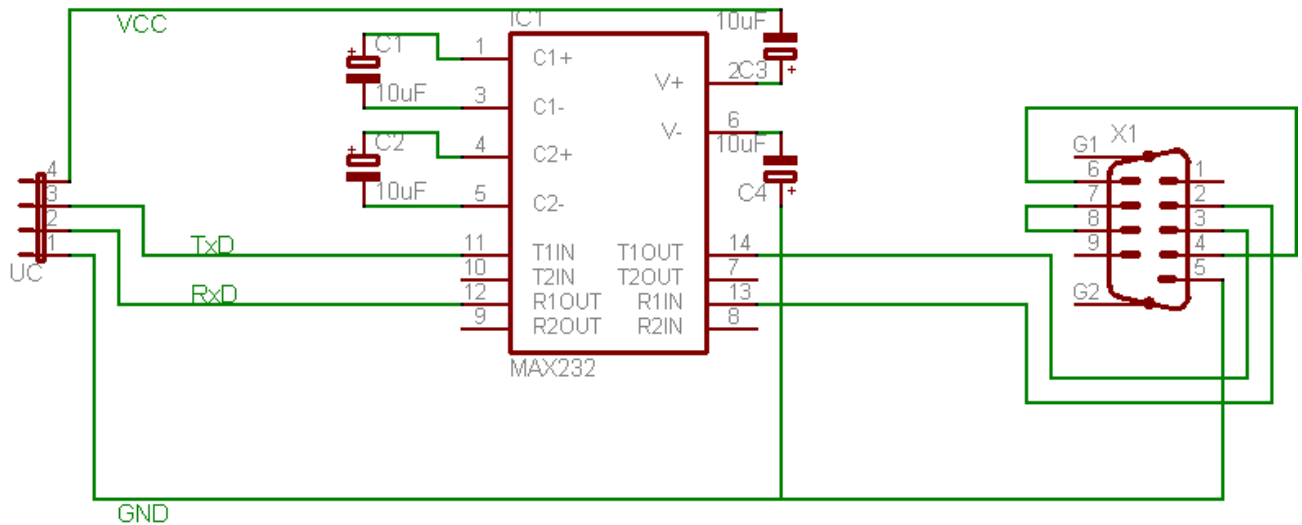
```
 4
 5 #include <avr/io.h>
 6 unsigned char result[] = "adcdacbdcb";
 7 int main(void)
 8 {
 9     unsigned char ch;
10     //initialize USART receive
11     UCSRB = UCSRB | (1 << RXEN);
12     //initialize USART transmit
13     UCSRB = UCSRB | (1 << TXEN);
14     //8-bit character per frame with 1 start & 1 stop bit
15     UCSRC = UCSRC | (1 << UCSZ1) | (1 << UCSZ0) | (1<<URSEL);
16     //Baud Rate is 4800
17     UBRRl = 0x67;
18     while(1)
19     {
20         /*
21         * Receive a char
22         */
23         while((UCSRA & (1 << RXC)) == 0);//wait for receive complete
24         ch = UDR;
25         /*
26         * Send the corresponding answer char
27         */
28         while((UCSRA & (1 << UDRE)) == 0);//wait until data register is empty
29         UDR = result[(ch - '0') % 10];
30     }
31     return 0;
32 }
```

ATmega16/32 RX & TX



MAX232

The Voltage Converter



Reference

- The avr microcontroller & embedded system, *Chapter 11*
  - Muhammad Ali Mazidi
  - Sarmad Naimi
  - Sepehr Naimi