



Chapter 6

The 8051 Microcontroller

Serial Communication

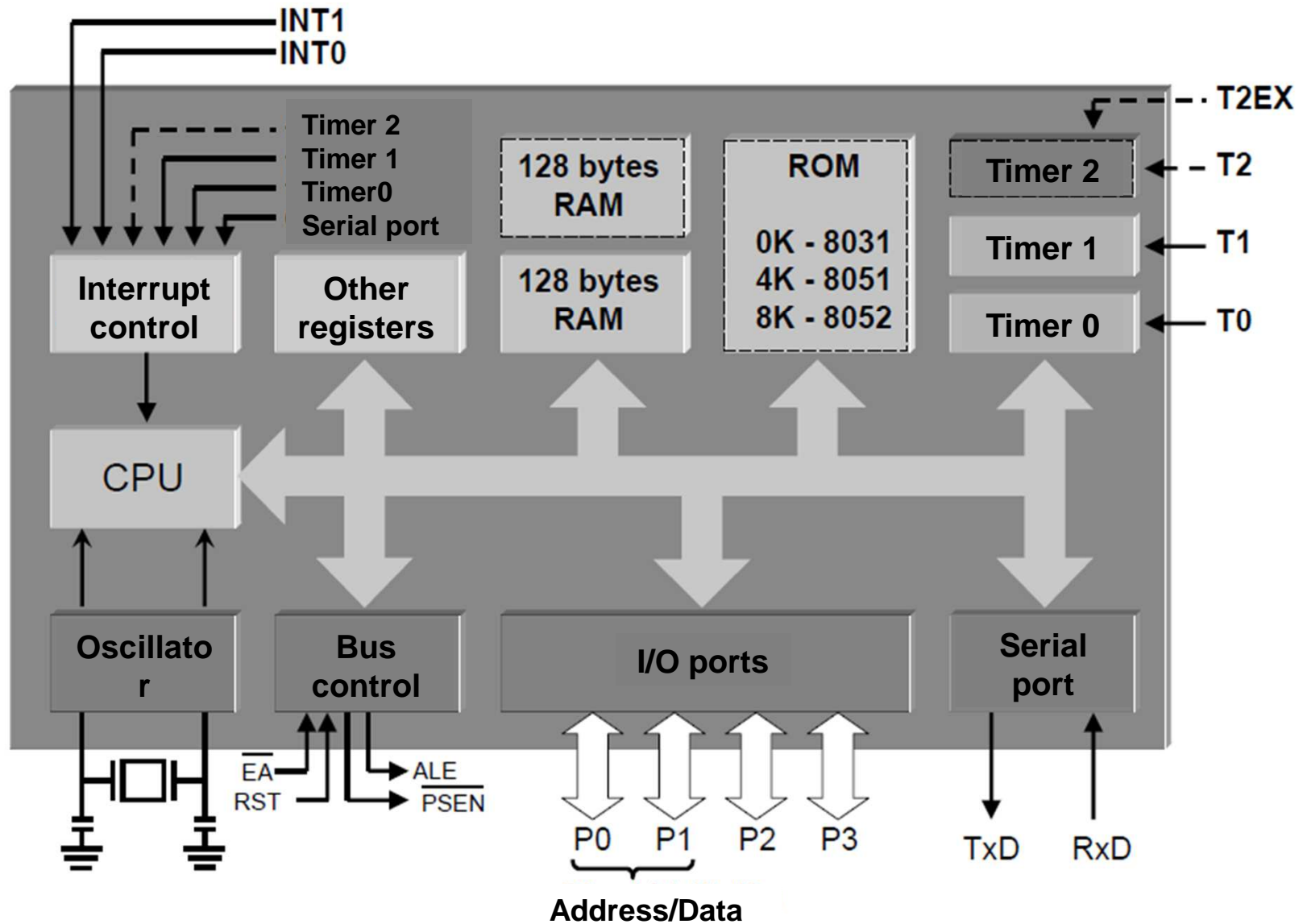


Faculty of Computer Science and Engineering
Department of Computer Engineering

Nguyen Quang Huy

huynguyen@cse.hcmut.edu.vn

Introduction



CPU

I/O

Timer

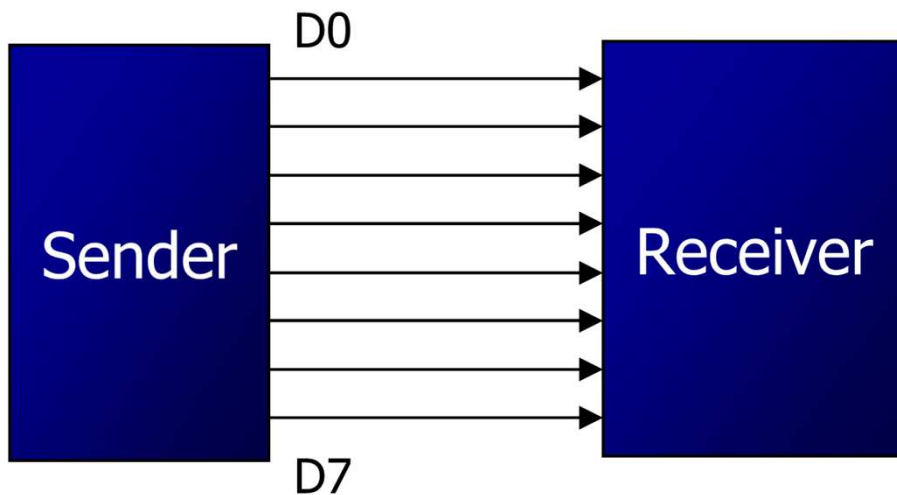
Interrupt

Serial

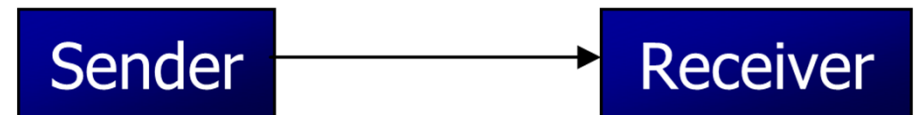
Parallel vs. Serial

- Transmitting: parallel-in serial-out shift register
- Receiving: serial-in parallel-out shift register

Parallel transfer



Serial transfer

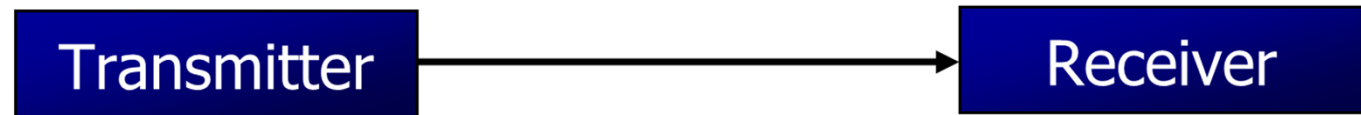


Synchronous vs. Asynchronous

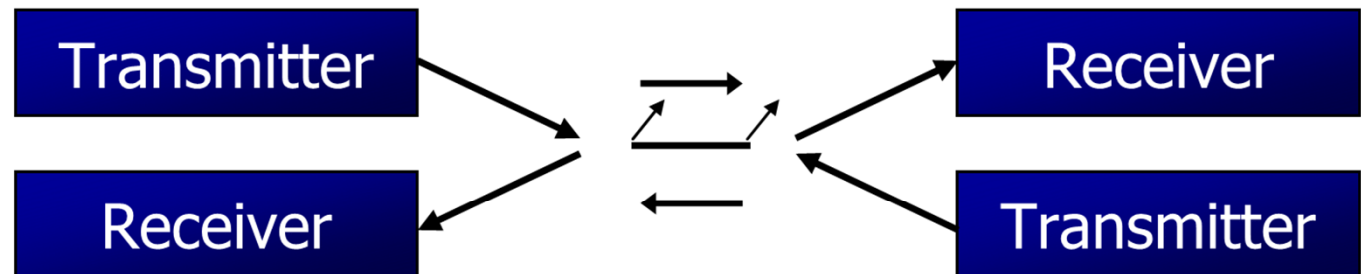
- Synchronous: transfer a block of data at a time
- Asynchronous: transfer a single byte at a time
- IC chips for serial communications
 - UART (Universal Asynchronous Receiver-Transmitter)
 - USART (Universal Synchronous-Asynchronous Receiver-Transmitter)

Communication Types

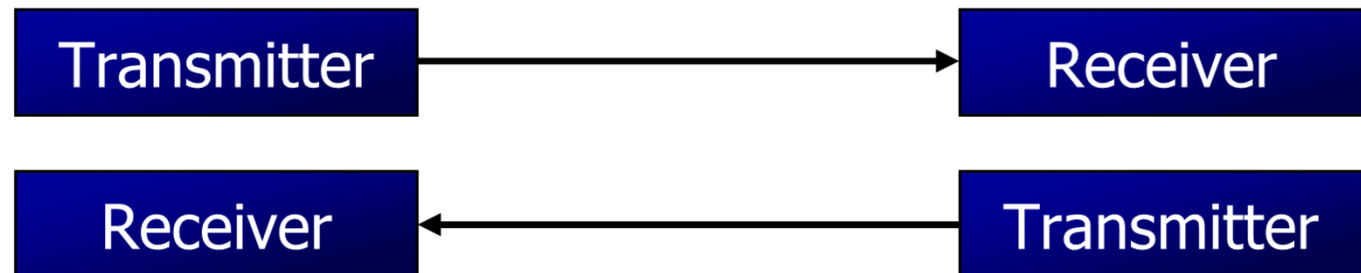
Simplex



Half Duplex

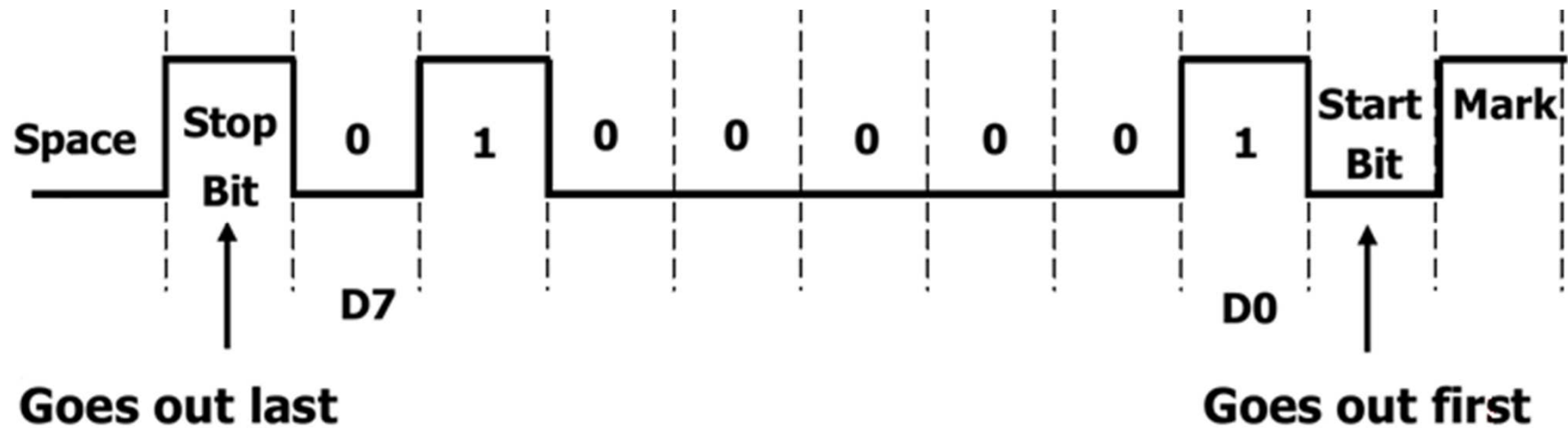


Full Duplex



Protocol

- A set of rules agreed by both the sender and receiver on
 - How the data is packed
 - How many bits constitute a character
 - When the data begins (start bit) and ends (stop bit)
 - *Framing*: between start and stop bits



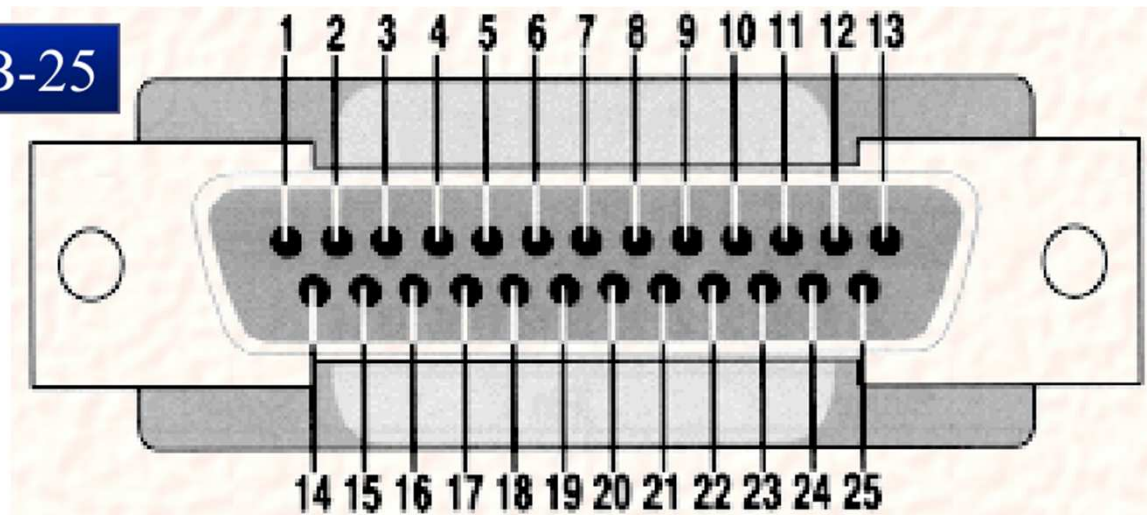
Data transfer rate

- **bps** (bits per second)
- **baud rate**
 - The number of signal changes per second
- As far as the conductor wire is concerned, the baud rate and bps are the same, and we use the terms interchangeably

RS232 Standard

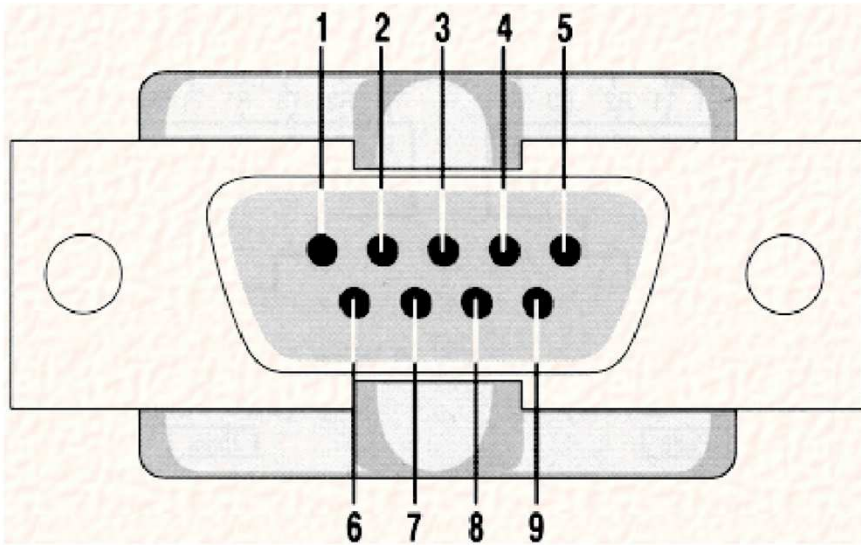
- In RS232
 - Bit 1: -3 ~ -25 V
 - Bit 0: +3 ~ +25 V
 - -3 to +3 is undefined

RS232 Connector DB-25



RS232 Standard

RS232 Connector DB-9

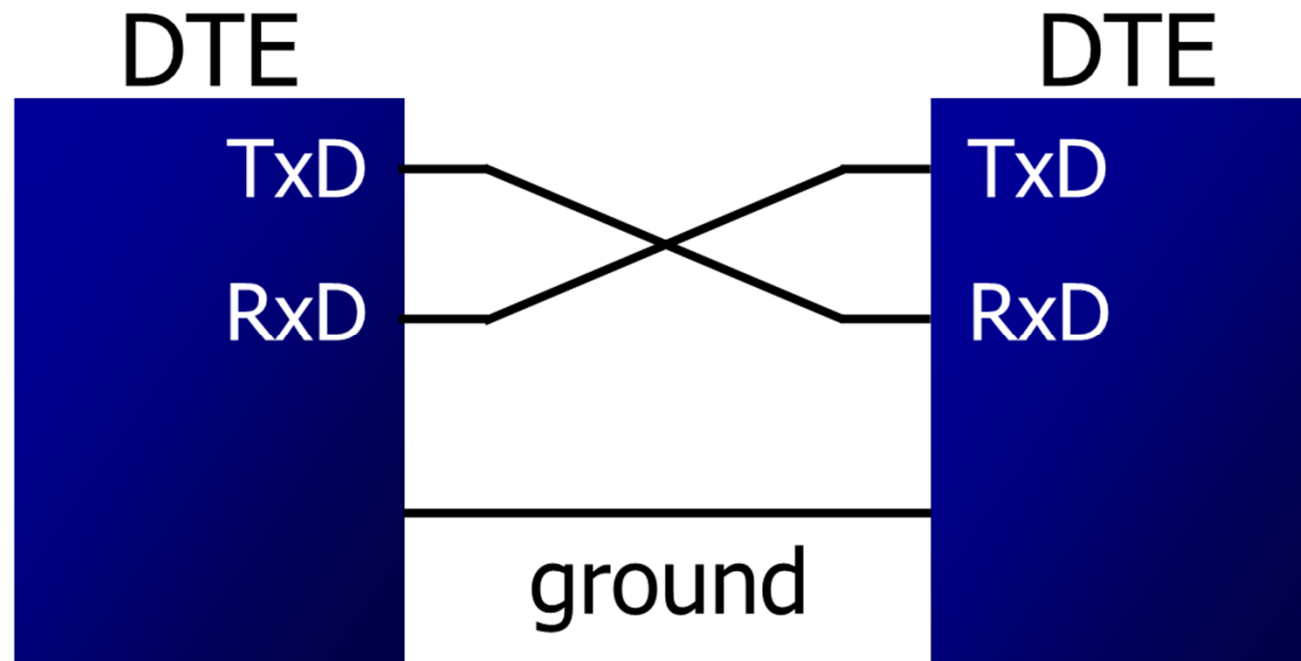


RS232 DB-9 Pins

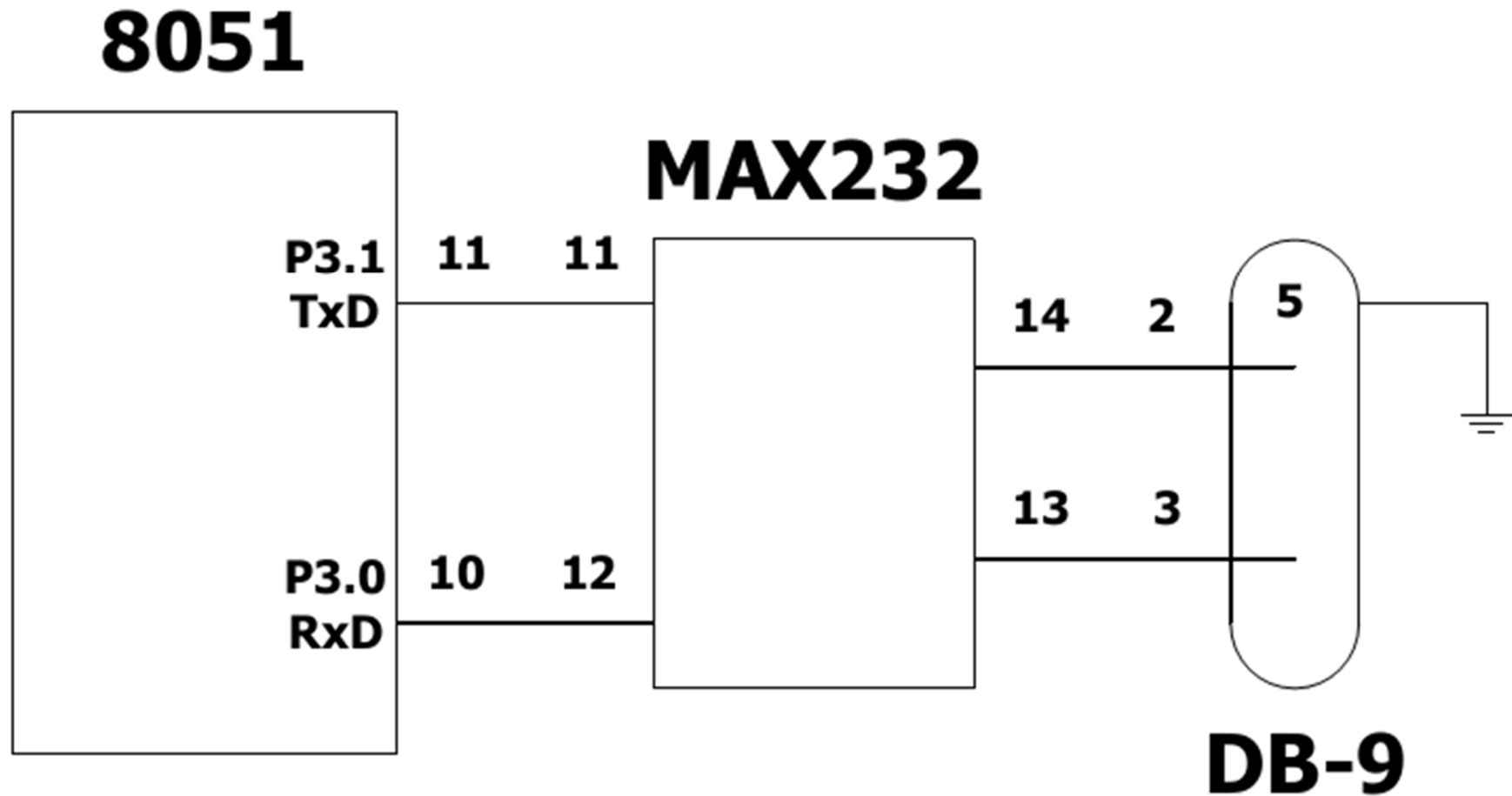
Pin	Description
1	Data carrier detect (-DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (-DSR)
7	Request to send (-RTS)
8	Clear to send (-CTS)
9	Ring indicator (RI)

Data Communication

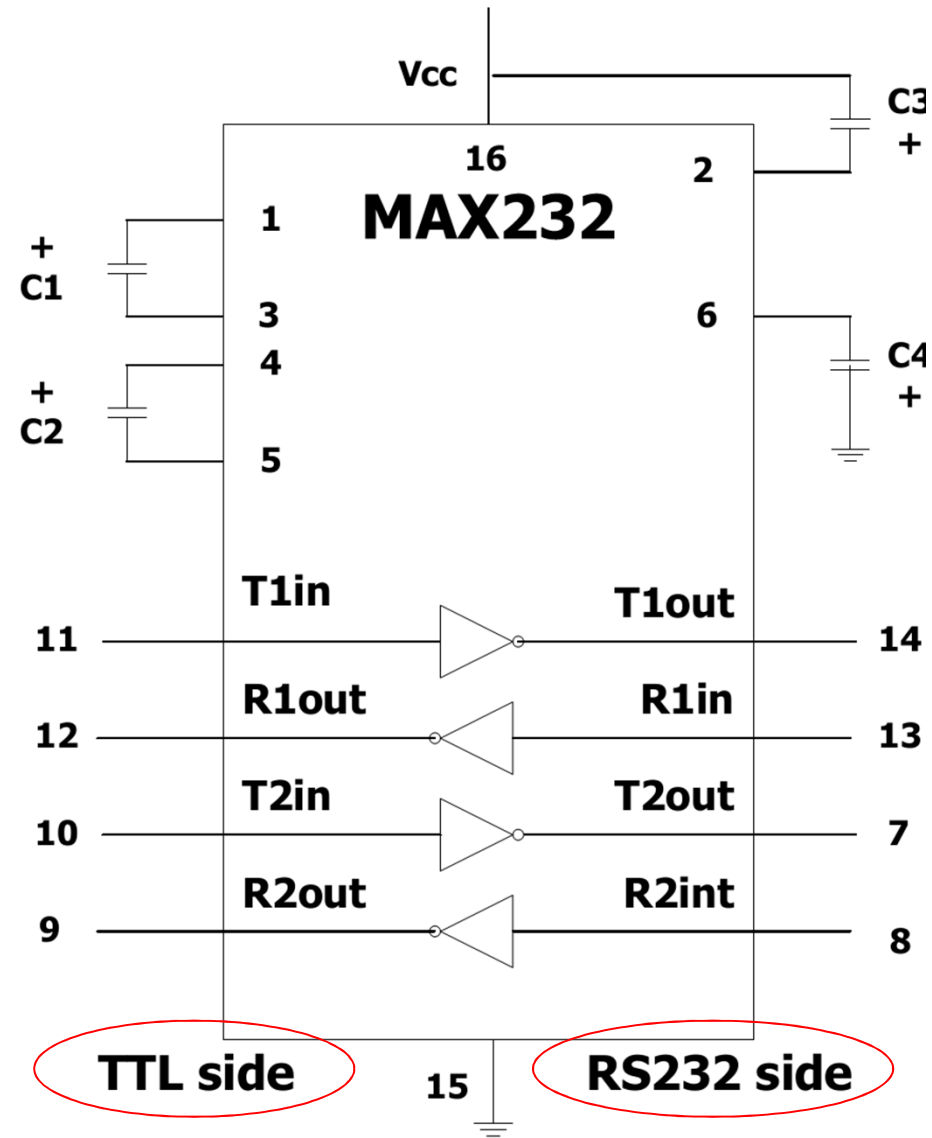
- Simplest connection between a PC and microcontroller requires a minimum 3 pins
 - TxD, RxD and GND



8051 Connection to RS232



MAX232



Serial Communication Programming

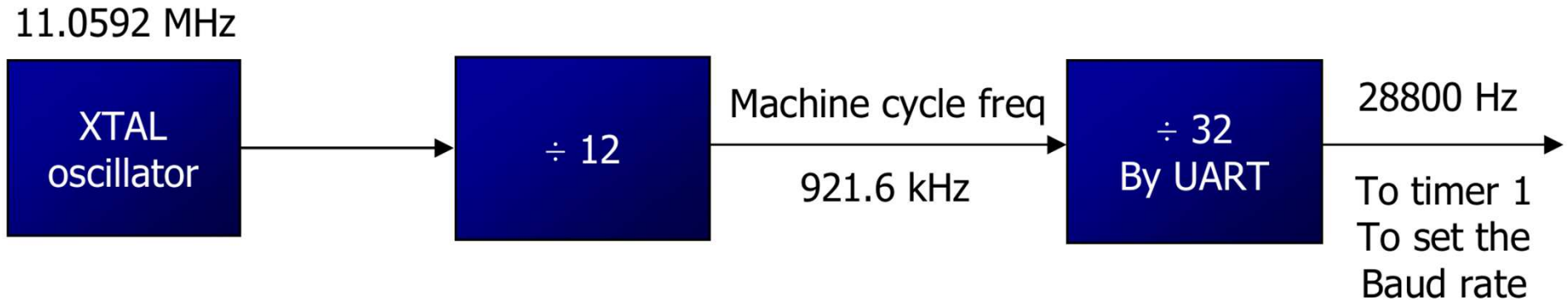
- Make sure that the baud rate of 8051 matches the baud rate of PC's COM port

Baud rate
110
300
600
1200
2400
4800

Baud rate
9600
14400
19200
38400
57600
115200

Serial Communication Programming

- Use **Timer 1** to set baud rate



$$f = 11.0592 / 12 = 921.6 \text{ kHz},$$

$$f_{\text{UART}} = 921.6 \text{ kHz} / 32 = 28,800 \text{ Hz}$$

$28,800 / 3 = 9600$ where -3 = FD (hex) is loaded into TH1

$28,800 / 12 = 2400$ where -12 = F4 (hex) is loaded into TH1

$28,800 / 24 = 1200$ where -24 = E8 (hex) is loaded into TH1

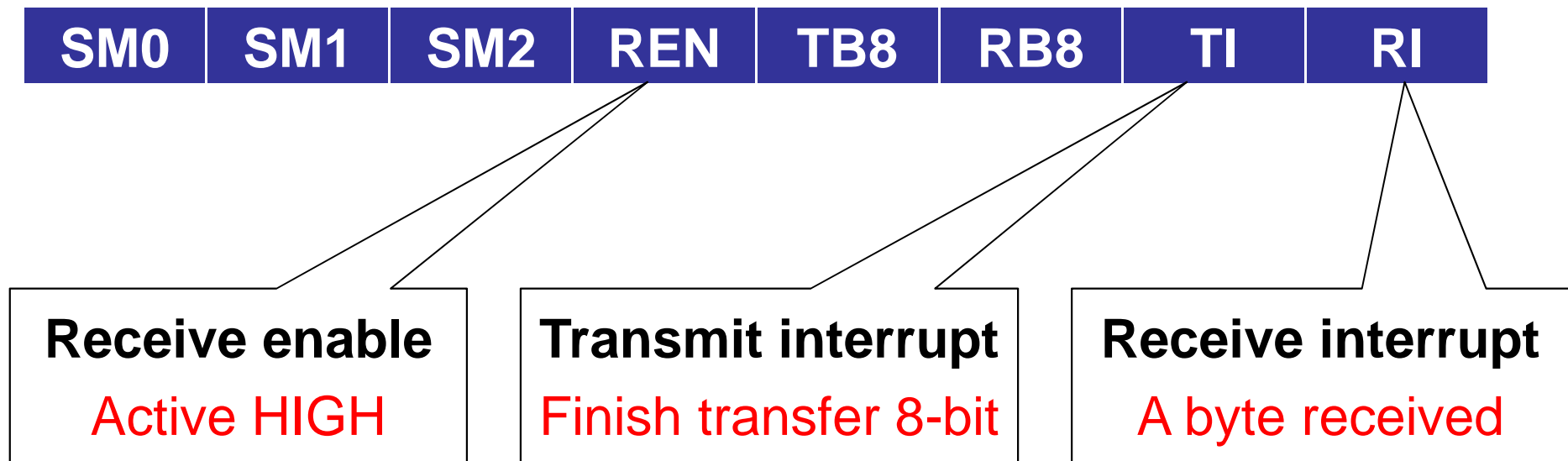
SBUF Register

- 8-bit register used solely for serial communication
 - Data to be transferred via TxD must be place in SBUF
 - SBUF holds the data received by 8051 RxD

```
MOV SBUF, #'D'    ;load SBUF=44h, ASCII for 'D'  
MOV SBUF,A        ;copy accumulator into SBUF  
MOV A,SBUF        ;copy SBUF into accumulator
```


SCON Register

- 8-bit register used to program the start bit, stop bit, and data bits of data framing, among other things



SCON Register

- SM2
 - This enables the multiprocessing capability of the 8051

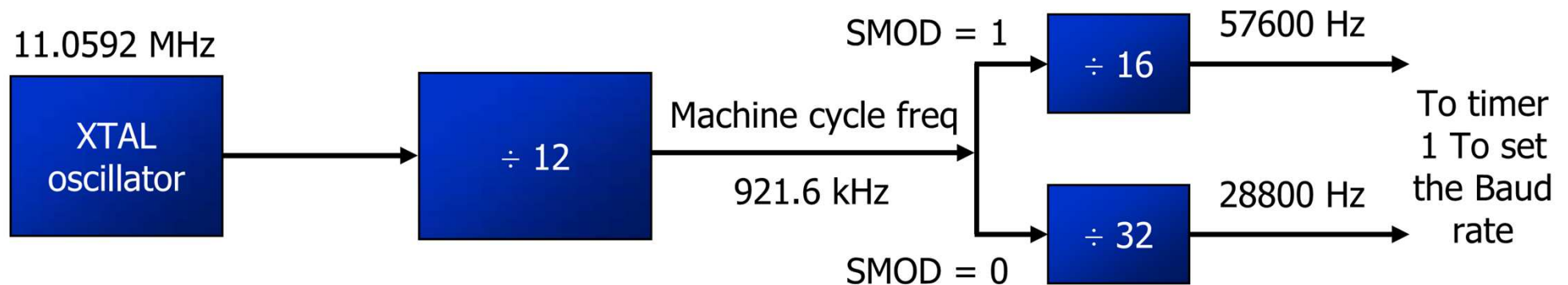
SM0	SM1	
0	0	Serial Mode 0
0	1	Serial Mode 1, 8-bit data, 1 stop bit, 1 start bit
1	0	Serial Mode 2
1	1	Serial Mode 3

Only mode 1 is
of interest to us

PCON Register

- 8-bit register
 - SMOD = 1 → double the baud rate

SMOD	---	---	---	GF1	GF0	PD	IDL
------	-----	-----	-----	-----	-----	----	-----



Example 1

- Write a C program for 8051 to transfer the letter “A” serially at 4800 baud continuously. Use 8-bit data and 1 stop bit

Example 1

```
#include <reg51.h>
void main(void){
    TMOD = 0x20;           //use Timer 1, mode 2
    TH1 = 0xFA;            //4800 baud rate
    SCON = 0x50;
    TR1 = 1;
    while (1) {
        SBUF = 'A';        //place value in buffer
        while (TI == 0);
        TI = 0;
    }
}
```

Example 2

- Program the 8051 in C to receive bytes of data serially and put them in P1. Set the baud rate at 4800, 8-bit data, and 1 stop bit.

Example 2

```
#include <reg51.h>

void main(void){
    unsigned char mybyte;

    TMOD = 0x20;           //use Timer 1, mode 2
    TH1 = 0xFA;            //4800 baud rate
    SCON = 0x50;
    TR1 = 1;               //start timer
    while (1) {             //repeat forever
        while (RI == 0);    //wait to receive
        mybyte = SBUF;       //save value
        P1 = mybyte;         //write value to port
        RI = 0;
    }
}
```

Reference

- “*The 8051 Microcontroller and Embedded Systems Using Assembly and C – 2nd*” - Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D.McKinlay
- “*The 8051 Microcontroller - 2nd*” - I. Scott Mackenzie, Prentice-Hall 1995