Rajdhani

DATE / /

Experiment :- 3

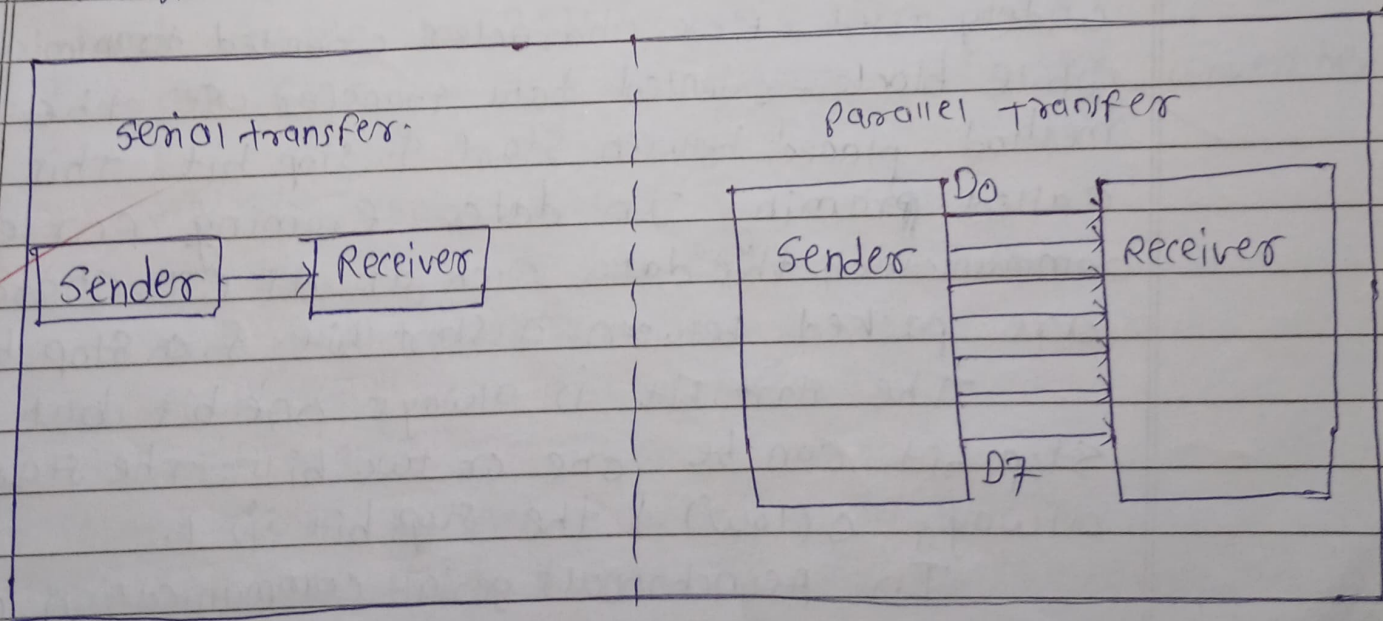
- **AIM** :- Serial communication using 8051.

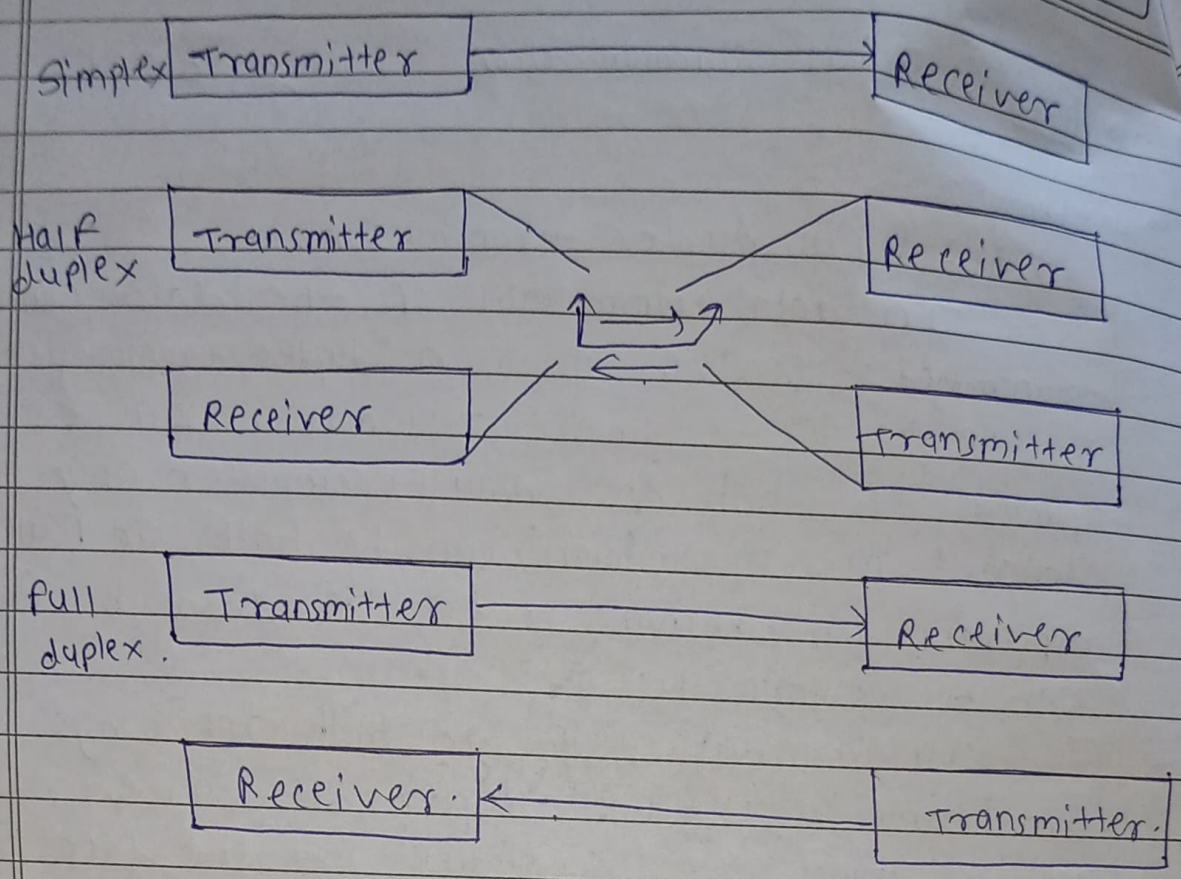
- **Theory :**

Half & full duplex transmission :

In data transmission if the data can be transmitted & receive it is a duplex transmission This is in the contact to simplex transmission such as with printers in which the ~~counter~~ computer only sends data Duplex transmission can be half or Full duplex depending on whether or not the data transfer can be simultaneous.

If the data can go both ~~ran~~ ways at the same time it is full duplex of course full duplex requires two wire conductors for the data lines one for transmission & one for reception in order to transfer & receive data simultaneously.





- Start & stop bits:

Asychronous serial data communication is widely used for character oriented transmission while block-oriented data transfer use the synchronous method. placed between start & stop bit. This is called framing In data framing for asynchronous communication the data such as ASCII characters are packed between a start bit & a stop bit.

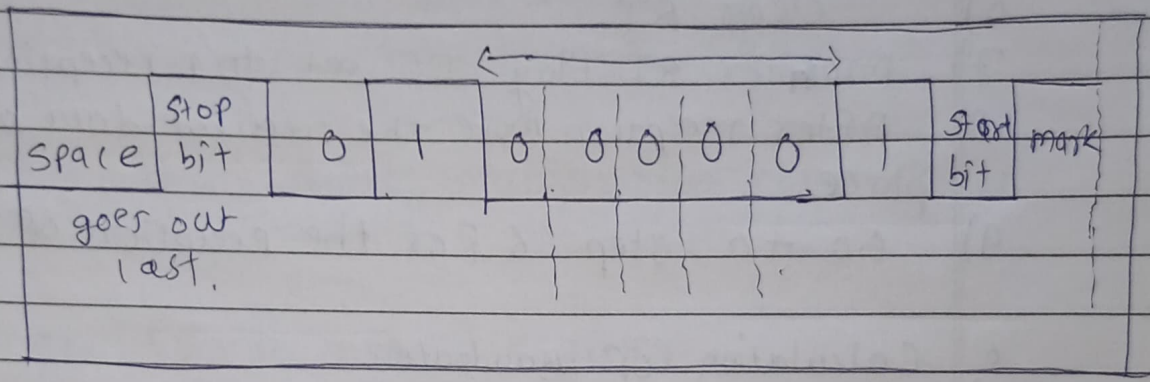
The start bit is always one bit but the stop bit can be one or two bits. The start bit is always 0 (low) & the stop bit is 1.

In Asynchronous serial communication, peripheral chips & modems can be programmed for data that is 7 or 8 bit wide this is in addition to the number of stop bit 1 or 2 while in order system ASCII character were 7 bit in recent year due to

Rajdhani

DATE / /

The extended CII character 8 bit data has become common In some older System due to the slowness of the receiving mechanical device two stop bits are used to give the device sufficient time to organize itself before transmission of the next byte.



- programming steps:
- Transmission:

1) load TMOD by 20H (Timer 1 in mode 2)
2) Crystal Freq / 12 → ans / 32 → output / Desired BR → Consider ans-ve = convert hex → 2's complement.
3) load TH1 by value from the steps.
4) load SCON Register
5) Start timer 1.
6) Clear T1.
7) Load SBUF by data which is to be transfer.
8) monitor TI Flag 1 bit If set to 1 transmission is Completed.
9) Go to step 6 for transmission of next byte.

- Reception:
1) load TMOD by 20H.
2) Same as or similar to transmission.
3) Load TH1
4) Load SCON Register.
5) Start timer 1.
6) clear RI.
7) Monitor RI flag If set to 1 reception is complete
8) After reception move the received data at another place.
9) Go to step 6 for the reception of next byte.

- Calculation of Baudrate:

1) 9600:

$$\frac{11.0592}{12} = 0.9216\ MHz = 921.6\ KHz$$

$$\frac{921.6}{32} = 3 \quad = -3$$

1st complement = + 1111 1100 = 1111 1101
1 = F D
(FD)H

(9600)$_H$ = FDH

- Mode 1 - 8 bit UART.

1) In this Mode we can transmit data through TXD pin & receive data through RXD pin.
2) In this mode we can transmit or receive data in full duplex mode.

Rajdhani
DATE / /

3) In this mode during transmission & reception start & Stop bits are added
4) So we can transmit 10 bits in mode 1 8 bit UART.
5) In this mode baud rate is variable & it can be calculated as follows:

$$BR = \frac{\text{Oscillator frequency}}{12 \cdot (256 - TH1)} \times \frac{2^{SMOD}}{32}$$

6) In this mode multiprocessor communication possible.

o SCON (serial control) register.
The SCON register is an 8 bit register used to program the start bit stop bit & data bit of data framing among other things.

0	1	0	1	0	0	0	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 SCON.7 - Serial port mode specifier
SM1 SCON.6 - Serial port mode specifier
SM2 SCON.5 - Used for multiprocessor communication
REN SCON.4 - Set/cleared by software to enable.
TB8 SCON.3 - NOT widely used.
RB8 SCON.2 - Not widely used.
TI SCON.1 - Transmit interrupt flag.
RI SCON.0 - Receive interrupt flag.

Rajdhani
DATE / /

- Timer 1 TH1 register values for various Baud

Baud rate	TH1 (Decimal)	TH1 (Hex).
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8.

A) Write a program to transfer "A" serially at 9600 baud rate continuously.

- Algorithm:

1) Load the value in TMOD #20H
2) Load the value in SCON #50H (mode1)
3) Load the value in TH1, #FDH (9600 baud rate)
4) Start timer 1
5) character "A" is move in SBUF
6) clear TI bit.
7) continuous ~~th~~ transfer character "A" using Loop JMP.

Program:

```
        ORG   0000 H
        MOV   TMOD, #20H
        MOV   SCON, #50H
        MOV   TH1, #FDH
        SETB  TR1
BACK :  MOV SBUF, #'A"
Next :  JNB   TI, Next
        CLR TI
        SJMP  BACK
        END.
```

B) Write a program to transfer "yes" serially at 9600 Baud rate continuously.

Algorithm:

1) load the value in TMOD, #20H.
2) load the value in TH1, #-3
3) Load the value in SCON #50H
4) Start timer 1 TR1
5) Move the character "Y" in A & give delay.
6) Move the character "E" in A & give delay.
7) Move the character "S" in A & give delay.
8) Move value A in SBUF
9) Check TI bit is zero or not.
10) Clear TI
11) Return to program.

- Program :

```
        ORG   0000H
        MOV   TMOD, #20H
        MOV   TH1, #-3
        MOV   SCON, #50H
        SETB  TR1
AGAIN:  MOV A, #"y"
        ACALL delay
        MOV A, #"E"
        Acall delay
        MOV A, #"S"
        Acall delay
        SJMP AGAIN
Delay : MOV SBUFF, A
HERE :  JNB TI, HERE
        CLR TI
        RET.
```

Conclusion :-

In this experiment we conclude that How to transfer Data serially with serial communication using 8051 Microcontroller.