COMMUNICATION BETWEEN TWO MICROPROCESSOR

BY

SUBATHRA S

COMMUNICATION BETWEEN TWO MICROPROCESSORS

OBJECTIVE

To establish communication between the two 8085 Microprocessor by writing an assembly language program to transmitter and receiver

APPARATUS REQUIRED

- Single board microcomputer (VI Microsystems 85).
- RS – 232 Connector.
- Flat Ribbon Cable.

DESCRIPTION

When data are transmitted as voltage, the commonly used standard is known as RS – 232C. It is defined in reference to data terminal equipment (DTE) and data communication equipment (DCE) terminal and modern. Its voltage levels are not compatible with TTL logic levels. The rate of data transmission in RS – 232C, is restricted to a maximum of 20k baud and a distance of 50ft, for high speed data transmission, new standards RS - 422A and RS - 423A developed in recent years.

The RS – 232C is a 25 pin connector and its signals are divided in to four groups signals, control signals, timing signals and ground.

The minimum interface between a computer and a peripheral requires three lines 2, 3, and 7. These lines are defined in relation to the DTE, the terminal transmits on pin2, and receiver other hand, the DCE transmits on pin3, receiver on pin2. The transmit and receive lines cross, hence it is known as NULL – MODEM CONNECTION.

ALGORITHM

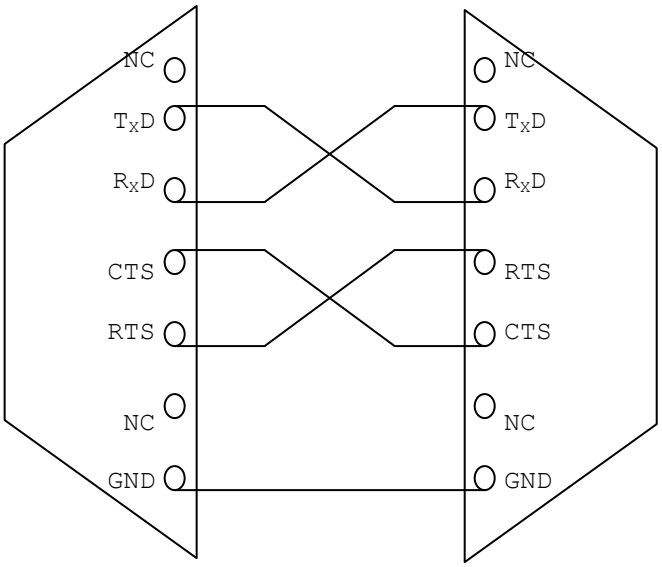
Transmitter

- Initialize the timer and USART of the transmitter.
- Set the starting address of data to be transmitted and the number of bytes to be transmitted.
- Program the USART for serial data transmission.
- Stop the execution.

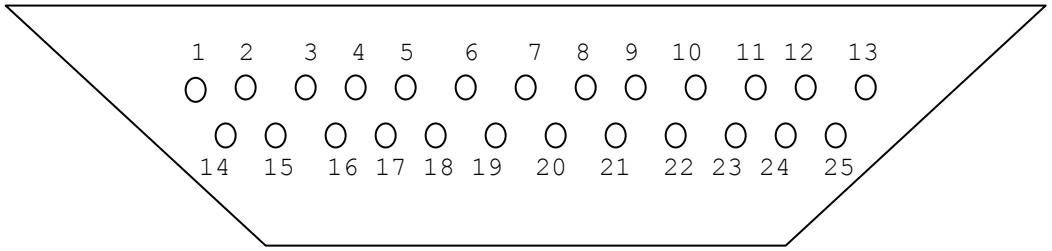
Receiver

- Intialize the timer and USART of receiving data.
- Set the starting address of data received.
- Program the USART for serial data reception.
- Receive the data from RS-232 pin and stop the execution.

NULL MODEM CONNECTION



RS 232 DETAILS - 25 PIN



PIN	SIGNALS
1	Protective ground
2	Transmitted data (TXD) -> DCE
3	Received data (RXD) -> DTE
4	Request to send (RST)->DCE
5	Clear to send (CTS) -> DTE
6	Data set ready (DSR) -> DTE
7	Signal ground
8	Received line signal detector
9,10	Reserved for data set testing
11	Unassigned
12	Secondary Clear to send
13	Secondary Recorded line signal detector
14	Secondary Transmitted data
15	Transmission signal element timing (DCE source)
16	Secondary Received data
17	Receiver signal element timing (DCE source)
18	Unassigned
19	Secondary request to send
20	DCE <- data terminal ready (DTR)
21	Signal quality detector
22	ring indicator

23	data signal rate selector (DTE/DCE source)
24	Transmit signal element timing
25	Unassigned

COMMAND INSTRUCTION FORMAT OF 8251A

DATA BIT		DESCRIPTION
D_0	**TXEN**	TRANSMIT ENABLE 1 – ENABLE ; 0 – DISABLE
D_1	**DTR**	DATA TERMINAL READY 'HIGH' WILL FORCE DTR OUTPUT TO 0
D_2	**RXE**	RECEIVE ENABLE 1 – ENABLE ; 0 – DISABLE
D_3	**SBRK**	SEND BREAK CHARACTER 1 = FORCE TXD "LOW" 0 = NORMAL OPERATION
D_4	**ER**	ERROR RESET 1 = RESET ERROR FLAGS OF, PE, OE, FE
D_5	**RTS**	REQUEST TO SEND "HIGH " WILL FORCE RTS OUPUT TO 0
D_6	**IR**	INTERNAL RESET "HIGH" RETURNS 8251A TO MODE INSTRUCTION FORMAT
D_7	**EH**	ENTER HUNT MODE 1 – ENABLE A SEARCH FOR SYNCRONOUS CHARACTER (HAS NO EFFECT IN ASYNCRONOUS MODE)

STATUS WORD FORMAT 8251A

DATA BIT		DESCRIPTION
D_0	**TXRDY**	TRANSMITTER READY TXRDY status bit has different meanings from the TXRDY output pin. The former is not conditioned by CTS & TXEN (i.e) TXRDY status bit = DB buffer empty TXRDY pin out = DB buffer empty. (CTS – 0) (TXEN – 1).
D_1	**RXRDY**	RECEIVER READY This bit indicates that the 8251A contains a character that is ready to be input to the CPU
D_2	**TXEMPTY**	TRANSMITTER EMPTY When the 8251A has no character to transmit this bit will go high
D_3	**PE**	PARITY ERROR The PE flag is set when a parity error is detected. It is reset by the ER bit of the command instruction. PE does not inhibit operation of the 8251A
D_4	**OE**	OVERFLOW ERROR The OE flag is set when the CPU does not read a character before the next one becomes available. OE is reset by the ER bit of the command instruction. OE does not inhibit the operation of 8251A however the previously overrun character is lost.
D_5	**FE**	FRAMING ERROR(Asynchronous mode only) The FE flag is set when a valid stop bit is not detected at end of every character. It is reset by the ER BIT of the command instruction. FE does not inhibit the operation of 8251A.
D_6	**SYNDET**	SYNC DETECT This pin is used in synchronous mode for syndet and is used in asynchronous mode for break detect.
D_7	**DSR**	DATA SET READY Indicates that the DSR is at zero level.

ASYNCRONOUS MODE OF 8251A

<table>
<tr><th colspan="2">DATA BIT</th><th>DESCRIPTION</th></tr>
<tr><td>D_0</td><td>B1</td><td rowspan="2">Baud rate Factor
<table><tr><th>B2</th><th>B1</th><th></th></tr><tr><td>0</td><td>0</td><td>Syncronous mode</td></tr><tr><td>0</td><td>1</td><td>1X</td></tr><tr><td>1</td><td>0</td><td>16X</td></tr><tr><td>1</td><td>1</td><td>64X</td></tr></table></td></tr>
<tr><td>D_1</td><td>B2</td></tr>
<tr><td>D_2</td><td>L1</td><td rowspan="2">Character Length
<table><tr><th>L2</th><th>L1</th><th></th></tr><tr><td>0</td><td>0</td><td>5 bits</td></tr><tr><td>0</td><td>1</td><td>6 bits</td></tr><tr><td>1</td><td>0</td><td>7 bits</td></tr><tr><td>1</td><td>1</td><td>8 bits</td></tr></table></td></tr>
<tr><td>D_3</td><td>L2</td></tr>
<tr><td>D_4</td><td>PEN</td><td>Parity Enable
1 – ENABLE
0 – DISABLE</td></tr>
<tr><td>D_5</td><td>EP</td><td>Even parity Generation Check
0 – ODD
1 – EVEN</td></tr>
<tr><td>D_6</td><td>S1</td><td rowspan="2">No of Stop bit
<table><tr><th>S2</th><th>S1</th><th></th></tr><tr><td>0</td><td>0</td><td>Invalid</td></tr><tr><td>0</td><td>1</td><td>1 bit</td></tr><tr><td>1</td><td>0</td><td>1.5 bits</td></tr><tr><td>1</td><td>1</td><td>2 bits</td></tr></table></td></tr>
<tr><td>D_7</td><td>S2</td></tr>
</table>

CONTORL WORD FORMAT 8253

<table>
<tr><th colspan="2">DATA BIT</th><th>DESCRIPTION</th></tr>
<tr><td>D_0</td><td>BCD</td><td>BINARY/BCD</td></tr>
<tr><td>D_1</td><td>M0</td><td rowspan="3">MODE
<table><tr><th>M2</th><th>M1</th><th>M0</th><th>MODE</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>2</td></tr><tr><td>0</td><td>1</td><td>1</td><td>3</td></tr><tr><td>1</td><td>0</td><td>0</td><td>4</td></tr><tr><td>1</td><td>0</td><td>1</td><td>5</td></tr></table></td></tr>
<tr><td>D_2</td><td>M1</td></tr>
<tr><td>D_3</td><td>M2</td></tr>
<tr><td>D_4</td><td>RL0</td><td rowspan="2">READ/LOAD
<table><tr><th>RL1</th><th>RL0</th><th></th></tr><tr><td>0</td><td>0</td><td>Latch</td></tr><tr><td>0</td><td>1</td><td>LSB</td></tr><tr><td>1</td><td>0</td><td>MSB</td></tr><tr><td>1</td><td>1</td><td>LSB/MSB</td></tr></table></td></tr>
<tr><td>D_5</td><td>RL1</td></tr>
<tr><td>D_6</td><td>SC0</td><td rowspan="2">SELECT COUNTER
<table><tr><th>SC0</th><th>SC1</th><th>CH#</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>2</td></tr><tr><td>1</td><td>1</td><td>X</td></tr></table></td></tr>
<tr><td>D_7</td><td>SC1</td></tr>
</table>

ASSEMBLY LANGUAGE PROGRAM

Transmitter				
ADDRESS	**LABEL**	**MNEMONICS**	**OPCODE/OPERAND**	**COMMENT**
4100		**MVI A,36$_H$**	3E 36	Initialization of 8253
4102		**OUT TMRCNT**	D3 0B	
4104		**MVI A,0A$_H$**	3E 0A	
4106		**OUT TMRCHO**	D3 08	
4108		**XRA A**	AF	
4109		**OUT TMRCHO**	D3 08	
410B		**XRA A**	AF	Resetting the 8251A
410C		**OUT UATCNT**	D3 05	
410E		**MVI A,40$_H$**	3E 40	
4110		**OUT UATCNT**	D3 05	
4112		**MVI A,4E$_H$**	3E 4E	Initialization of 8251A
4114		**OUT UATCNT**	D3 05	
4116		**MVI A,37$_H$**	3E 37	
4118		**OUT UATCNT**	D3 05	
411A		**LXI H,4300$_H$**	21 00 43	Load the input data in to transmitter memory
411D		**MVI C,05$_H$**	0E 05	Initialize the number of input data count
411F	**LOOP**	**IN UATCNT**	DB 05	Check 8251A TXEMPTY and then send the data
4121		**ANI 04$_H$**	E6 04	
4123		**JZ LOOP**	CA 1F 41	
4126		**MOV A,M**	7E	Move the recent input data to accumulator
4127		**INX H**	23	Increment the memory location to get the next data in to transmitter
4128		**OUT UATDAT**	D3 04	Transmit the data to receiver
412A		**DCR C**	0D	Decrement the input data count
412B		**JNZ LOOP**	C2 1F 41	Until all data are transmitted continue looping
412E		**HLT**	76	Stop the execution
Receiver				
4200		**MVI A,36$_H$**	3E 36	Initialisation of 8253
4202		**OUT TMRCNT**	D3 0B	
4204		**MVI A,0A$_H$**	3E 0A	
4206		**OUT TMRCHO**	D3 08	
4208		**XRA A**	AF	
420A		**OUT TMRCHO**	D3 08	
420B		**XRA A**	AF	Resetting the 8251A
420C		**OUT UATCNT**	D3 05	
420E		**MVI A,40$_H$**	3E 40	
4210		**OUT UATCNT**	D3 05	
4212		**MVI A,4E$_H$**	3E 4E	Initialization of 8251A
4214		**OUT UATCNT**	D3 05	

4216		**MVI A,37 H**	3E 37	
4218		**OUT UATCNT**	D3 05	
421A		**LXI H,4400 H**	21 00 44	Load the output data in to receiver memory
421D		**MVI C,05 H**	0E 05	Initialize the number of input data count
421F	**LOOP1**	**IN UATCNT**	DB 05	Check 8251A RXEMPTY and hence get the data and store the data
4221		**ANI 02 H**	E6 02	
4223		**JZ LOOP1**	CA 1F 42	
4226		**IN UATDAT H**	DB 04	Receive the data from transmitter
4228		**MOV M,A**	77	Move the recent input data to accumulator
4229		**DCR C**	0D	Decrement the receiver data count
422A		**INX H**	23	Increment the memory location to store the transmitted data
422B		**JNZ LOOP**	C2 1F 42	Until all data are received continue looping
422E		**HLT**	76	Stop the execution

EXECUTION

Transmitter

ADDRESS	DATA
4300_H	AA_H
4301_H	BB_H
4302_H	CC_H
4303_H	DD_H
4304_H	EE_H

Receiver

ADDRESS	DATA
4400_H	AA_H
4401_H	BB_H
4402_H	CC_H
4403_H	DD_H
4404_H	EE_H

REFERENCE

1. Ramesh S.Gaonkar, Microprocessor Architecture, Programming, and Applications, Fourth Edition, Penram International Publishing (India), 2000.

2. S.Subathra, “Advanced Microprocessor Laboratory”, Record work, Adhiparashakthi Engineering College, Melmaruvathur, October 2002

3. S.Subathra, “Programming in 8085 Microprocessor and its applications – An Innovative Analysis”, Technical Report, Adhiparashakthi Engineering College, Melmaruvathur, March 2003

4. Micro-85 EB, User Manual, Version – 3.0, CAT #M85 EB-002, VI Microsystems Pvt. Ltd., Chennai.