Experiment 6

Interfacing a Hyper Terminal to the Flight 86 Kit

Objective

The aim of this lab experiment is to interface a Hyper Terminal to 8086 processor by programming the 8251 USART.

Equipment

- Flight 8086 training board,
- PC with Flight86 software, and
- Download cable

Tasks to be Performed

Reading data from the keyboard and displaying it on the Hyper Terminal using RS-232 standard interface and asynchronous serial communication

6.1 Background

The Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communication with Intel's microprocessor families. It is used as a peripheral device and is programmed by the CPU to operate using many serial data transmission techniques.

The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream. It accepts serial data streams and converts them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the status of the USART at any time. The status includes data transmission errors and control signals.

Although the USART is capable of operating in synchronous and asynchronous modes, it is probable that most work will be carried out in the asynchronous mode. Therefore, asynchronous operation only will be described in this experiment.

The 8251 USART chip may operate its transmitter and receiver independently. It may even operate them at different speeds. However, on the FLIGHT-86 board, they both work at the same speed which can be programmed using Counter 2 of the 8253 PIT chip.

Registers

The 8251 USART chip has four register:

1. **The Data Register** is used to store recently received data byte or the data byte that is ready to be transmitted.

2. **The Mode Register** is used to set the operation mode of the 8251 chip.

3. **The Control Register** is used to send commands to the 8251 chip.

4. **The Status Register** reflects the current status of 8251 chip.

Table 6.1 shows the actual port addresses and allowed activities of these four registers in the FLIGHT-86 board.

Table 6.1: The 8251 Registers

Register	Activity Allowed	Actual Port Address
Data	Read/Write	18h
Mode	Write Only	1Ah
Control	Write Only	1Ah
Status	Read Only	1Ah

Control Words

The 8251 USART chip can be programmed using two types of control words:

(1) The Mode Instruction, which must follow a reset (internal or external) to specify the general operation of the chip. This control word can be sent through the MODE register according to the format shown in Figure 6.1 (a). The baud rate of the transmitter/receiver depends on the frequency of the input clock of the transmitter/receiver. For example, if the input clock runs at 9600Hz (i.e. 9600 cycles per second), then the 8251 can transmit/receive 9600 bits per second (i.e. 1 bit every clock cycle). The baud rate factor is used to adjust the baud rate by a certain factor (i.e. divide the baud rate by 1, 16 or 64).

(2) The Command Instruction, which defines the detailed operation. Command instructions can by sent after a mode instruction using the COMMAND register according to the format shown in Figure 6.1 (b). All command instructions must keep D6 zero. Setting this bit to 1 will reset the USART and make it ready to accept a new mode word.

Status Word

The CPU can examine the STATUS register at any time to determine the current condition of the 8251. The format of the 8251 status word is given in Table 6.2.

Sending Data

Provided the transmitter is enabled, as soon as a data byte is written to the DATA register, the 8251 will convert this to a serial form and send it to the serial output pin in the format specified. If the receiving device is not ready the data byte will be held and sent as soon as possible.

Sending Data

Provided that the receiver is enabled, the 8251 will receive a serial data byte sent to it from another device, and will store it in the DATA register. The CPU can identify if a character has been received (e.g. by checking the STATUS register) and read the data byte form the DATA register. Once the data byte is read the DATA register is flushed.

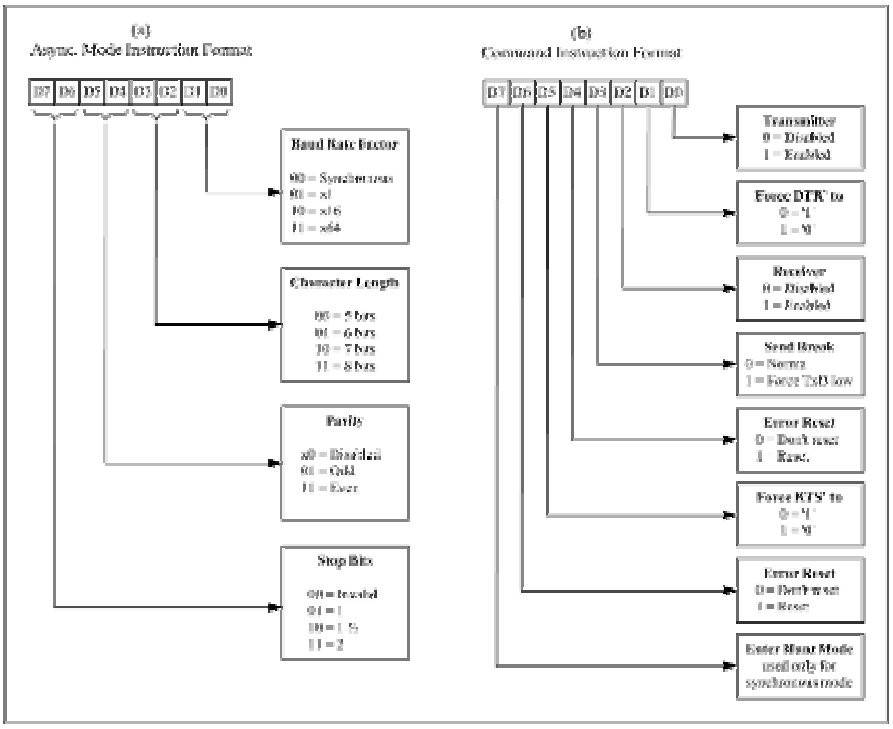

Figure 6.1: 8251 Control Words

Table 6.1: 8251 Status Word

BIT		Description
0	Transmitter Ready	This bit is set when the transmitter is ready to receive a new character for transmission from the CPU.
1	Receiver Buffer Full	This bit s set when a character is received on the serial input.
2	Transmitter Buffer Empty	This bit is set as soon as the USART completes transmitting a character and a new one has not been loaded in time. It goes low only while a data character is being transmitted by the USART.
3	Parity Error	This bit is set when parity is enabled and a parity error is detected in any received character.
4	Overrun Error	This bit is set when the CPU does not read a received character before the next one becomes available. However, the previous character is lost.
5	Framing Error	This bit is set when a valid stop bit (high) is not detected at the end of a received character.
6	Synchronous Detect/ Break Detect	Used for synchronous mode only
7	Data Set Ready	This is a general-purpose input bit that can be read by the CPU as part of the 8251status.

6.2 Programming the 8251 USART Chip

In order to program the 8251 USART chip in asynchronous mode, you need to do the following:

1. Set the receiver/transmitter input clock to the desired **baud rate**. In the FLIGHT-86, this is done by programming Counter 2 of the 8253 PIT chip.

2. Send asynchronous **mode instruction** with the desired format.

3. Send a **command instruction** to enable the transmitter/receiver.

Setting up the Baud Rate

Since the output of Counter 2 is connected to the input clocks of the transmitter and receiver, Counter 2 can be used to control the speed (baud rate) of transmission/reception. If we program Counter 2 in mode 3 and load it with some number N, then the transmitter/receiver can be operated at 14.7456/(N×6) MHz. In other words, the transmitter/receiver can transmit/receive $14.7456\times10^6/(N\times6)$ bits per second. For instance, to get a baud rate of 9600, we need to load Counter 2 with $14.7456\times10^6/(9600\times6) = 256$.

Sending a Mode Instruction

After a system RESET the MODE register must be the first to be set. Because the 8251 may be in an unknown state, it is normal practice to send the byte 00h to the COMMAND register three times. This will guarantee the 8251 COMMAND register is active. Now, the byte value 40h (D6 is 1) is sent to activate the MODE register. The desired asynchronous format may now be set up and sent to the MODE register.

Sending a Control Instruction

Once the mode has been programmed the 8251 will switch to the COMMAND register. Now, you can send a command instruction according to the format shown in Figure 6.1 (b) to enable the transmitter or the receiver or both.

Example 4.1: Write a program that continuously reads one character from the keyboard and displays it on the Hyper Terminal.

```
1    COMSEG SEGMENT BYTE PUBLIC 'CODE'
2    ASSUME CS:COMSEG, DS:COMSEG, ES:COMSEG, SS:COMSEG
3    ORG 0100H
4    Start:
4          ; send 00h three times to ensure that
5          ; the command register is active
6          MOV AL, 00h
7          CALL Send_Control_Word
8          CALL Send_Control_Word
9          CALL Send_Control_Word
10         ; send 40h to activate the mode register
11         ; (D6=1 internal reset)
12         MOV AL, 40h
13         CALL Send_Control_Word

14         ; send a mode instruction, namely
15         ; 0100 1101 = 4Dh
16         ; baud factor=1, 8-bit char, no parity, 1 stop bit
17         MOV AL,  4Dh ;mode register 0100 1110
18         CALL Send_Control_Word
19         ; send a command instruction to enable
20         ; both the transmitter and receiver
21         MOV AL,  37h
22         CALL Send_Control_Word

23         ; initialize 8253 PIT chip  (1011 0110 = B6h)
24         ; Counter2, load MSB then LSB, mode 3, binary
25         MOV AL, 0B6h
26         OUT 0Eh, AL

27         ; counter loaded with 0100h (baud rate = 9600)
28         MOV AL, 00h
29         OUT 0Ch, AL  ; first load low byte
30         MOV AL, 01h
31         OUT 0Ch, AL  ; now load high byte

32   L1:   IN   AL, 1Ah; read status word
33         TEST AL, 02h; check receiver buffer
34         JZ L1   ; if buffer is empty then try again
35         IN  AL, 18h ; else get the character in the buffer
36         OUT 18h, AL ; and output it to the Hyper Terminal
37         JMP L1        ; repeat the process

38   Send_Control_Word:
39           OUT 1Ah, AL  ; send the control word
40           MOV CX, 200h ; delay to ensure 8251 catches up
41   Delay: LOOP Delay
42           RET
43   COMSEG ENDS
44   END   start
```

Note that once you run the program of Example 6.1 on the FLIGHT-86, you need to close the F86GO program to allow the Hyper Terminal to communicate with the program through the serial cable. This is because the serial communication port may not be used by more than one application at the same time.

Exercises

6.1. Write a program to read a string from the keyboard and display it on the Hyper Terminal in a reverse order.

6.2. Write a program to read a decimal number (between 0 and 255) from the keyboard and display it on the Hyper Terminal as a binary number.