MICROPROCESSOR BASED SYSTEMS

CMP 410

LECTURE 8_B

https://AssemSite8.wix.com/site8
Using “MikroC PRO 8051” to write embedded C-programs

With embedded libraries
Example

Design an embedded system (draw block diagram and write both assembly and MikroC code [without ISR]) for reading message “Are you OK?” that stored in the program memory of AT89C51 (starting at address 100H) and sending it to a computer via its serial port with baud rate 9600 (TH1 = FDH).
Serial Control (SCON)

<table>
<thead>
<tr>
<th>Modes</th>
<th>SM0</th>
<th>SM1</th>
<th>SM2</th>
<th>REN</th>
<th>TB8</th>
<th>RB8</th>
<th>TI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

AT89C51

Computer

“Are you OK?”

P3.1

Flash M
INCLUDE 89C51.mc
ORG 0H
SJMP MAIN

ORG 30H

MAIN:

MOV TMOD, #20H ;timer 1 is auto reload (mode 2)
MOV TH1, #FDH ; adjust 9600 baud rate
MOV SCON, #40H ;Ser mode1 > 8-bit, 1 start, 1 stop

SETB TR1 ;start timer 1

MOV DPTR,#MY_message1 ;load dp0tr with Meg location

Chr_Again:
CLR A
MOVC A,@A+DPTR ; move Chr_code to acc
JZ STOP ; chek if Msg end
CALL SEND ; start sending via serial port
INC DPTR ; next Chr
SJMP Chr_Again

SEND: MOV SBUF,A ;sending data via serial port

WAIT: JNB TI, WAIT ;stay in loop until complete sending
CLR TI ; clear trans interrupt flag
RET

STOP: SJMP $ ; YOU CAN SEND ANOTHER MSG

ORG 100H
MY_message1: DB "ARE YOU OK?",0
```c
void main() {
    UART1_Init(9600); // Initialize UART
    Delay_ms(50); // for UART module stabilizing
    UART1_Write_Text("ARE YOU OK?");
    while (1) { // Endless loop
        ...
    }
}
```
Example

Draw hardware block diagram and write both assembly & MikroC codes (without ISR) to design an embedded system (operates as UART Transceiver) based on AT89C51.

It receives data-bytes from a computer, complements them and send them to the computer with baud rate 9600 ($\text{TH1} = \text{FDh}$).
Using microchip MAX233 to interface between computer and 8051 via serial port
INCLUDE 89C51.mc

ORG 0H

SJMP MAIN

ORG 30H

MAIN:

    MOV TMOD, #20H ; timer 1 is auto reload (mode 2)
    MOV TH1, #FDH ; adjust 9600 baud rate
    MOV SCON, #50H ; Ser mode1 > 8-bit, 1 start, 1 stop
    SETB TR1 ; start timer 1

Receive_Chr:

WAIT_1:   JNB RI, WAIT_1 ; stay in waiting until complete Receiving
    MOV A, SBUF ; Receiving data via serial port
    CLR RI ; clear Receive interrupt flag

;---------------------------------
    CPL A ; complement Received data

SEND:     MOV SBUF, A ; sending data via serial port

WAIT_2:   JNB TI, WAIT_2 ; stay in waiting until complete sending
    CLR TI ; clear transmit interrupt flag

SJMP Receive_Chr ; ANOTHER Chr.
Using the MikroC SW program

```c
char Data_Bytes;

void main() {
    UART1_Init(9600);  // Initialize UART
    Delay_ms(100);     // Wait for UART stabilize

    while (1) {        // Endless loop
        if (UART1_Data_Ready()) {  // If data is received,
            Data_Bytes = UART1_Read();  // read the received data,
            UART1_Write(~ Data_Bytes);  // and send inverse data via UART
        }
    }
}
```
Example

Design an embedded system based on AT89C51 (draw block diagram and write MikroC code) for sending message “Are you OK?” via the serial port with baud rate 9600 and display it on the LCD.
4 bits Data & Commands
Using the MikroC SW program

```c
// Lcd module connections
sbit LCD_RS at P2_0_bit;    // Register Select pin
sbit LCD_EN at P2_1_bit;   // Enable pin
sbit LCD_D4 at P2_2_bit;    // 4 data input pins
sbit LCD_D5 at P2_3_bit;
sbit LCD_D6 at P2_4_bit;
sbit LCD_D7 at P2_5_bit;
// End Lcd module connections

char msg1[] = "ARE YOU OK?";

void main() {
    Lcd_Init();       // Initialize Lcd
    Lcd_Cmd(_LCD_CLEAR);   // Clear display
    Lcd_Cmd(_LCD_BLINK_CURSOR_ON);
    UART1_Init(9600);   // Initialize UART module
    Delay_ms(100);      // UART module to stabilize
    UART1_Write_Text(msg1);
    while (1) {         // Endless loop
        Lcd_Out(1,1,msg1); // Write text in first row
        Delay_ms(1000);   // Clear display
        Lcd_Cmd(_LCD_CLEAR);
        Delay_ms(500);
    }
}
```