Chapter 09

Programming in Assembly

Lesson 08

Programming Examples for Serial Port

Programming the SCON

- SCON has control as well as status bits and is bit addressable

RI at SCON^0 at Bit Address 98H

- Flag bit— indicates Received the character sent
- Must be reset at the interrupt service-routine for enabling next interrupt.

TI at SCON^1 at 99H

- Flag bit— indicates Transmitted character sent
- Must be reset at the interrupt service-routine for enabling next interrupt

RB8 at SCON^2 at 9AH

- Status bit— the Received 8th bit after b7 in mode 2 and 3

TB8 SCON^3 at 9BH

- 8th bit after b7 for Transmission in mode 2 and 3

REN at SCON^4 at 9CH

- Control bit for Rx enable

SM2 SCON^5 at 9DH

- Control Bit for Multi-processor communication mode 2 or 3
- Control bit to show when serial port receives a invalid character in mode 1

SM0 and SM1 at SCON^7 and SCON^6 at 9FH and 9EH

- Serial Mode 0, 1, 2 or 3 control bits

SM0-SM1 = 00 for mode 0.

SM0-SM1 = 01mode 1.

SM0-SM1 = 10 for mode 2.

SM0-SM1 = 11 for mode 3.

Write instructions to set Serial port in Mode 0

- Serial Mode 0: CLR SCON^7 and CLR SCON^6

or CLR 9FH and CLR 9EH

or ANL SCON, # 3F; Using AND operation upper 2 bits will change to 0

Write instructions to set Serial port in Mode 1

- Serial Mode 1: CLR SCON^7 and SETB SCON^6

or CLR 9FH and SETB 9EH

or ORL SCON, # 40H and ANL SCON, # 7FH

Write instruction to set SM2 multiprocessor communication control

- SM2 Set: SETB SCON^5

or SETB 9DH

or ORL SCON, # 20H; Using OR operation bit next to upper 2 bits will change to 1

Write instruction to Enable Receiver at serial port

- REN Set: SETB SCON^4

or SETB 9CH

or ORL SCON, # 10H; Using OR operation bit next to upper 3 bits will change to 1

Write instruction to set Enable TB 8 = 1 to enable appropriate action by the receiver at other end

- TB8 Set: SETB SCON^3

or SETB 9BH

or ORL SCON, # 08H; Using OR operation bit next to lower 3 bits will change to 1

Write instruction to reset TB 8 = 0 to enable appropriate another action by the receiver at other end

- TB8 Reset: CLR SCON^3

or CLR 9BH

or ANL SCON, # F7H

; Using AND operation bit next to lower 3 bits will change to 0

Write instructions to set Serial port in Mode 2

- Serial Mode 2: SETB SCON^7 and CLR SCON^6

or SETB 9FH and CLR 9EH

or ORL SCON, # 80H and ANL SCON, # BFH;

Write instruction to set Serial port in Mode 3

- Serial Mode 3: SETB SCON^7
- SETB SCON^6

or SETB 9FH and SETB 9EH

or ORL SCON, # C0H;

SBUF Register Address 99H

- The characters are sent through serial port by write operations at the SBUF
- The characters are received through serial port by read operations at the SBUF

Write instructions to transmit character A

- MOV SBUF, #65; 65d is ASCII code of A

Read Write instructions to read character at serial port at internal RAM 30H

- SETB SCON^4; Enable receiver
- MOV 30H, SBUF; Move SBUF to direct address 30H

Serial Asynchronous (UART) Communication Mode 3

- Variable UART rate of using T1 in 8051 and T2 in 8052
- UART communication between one MCU and peripheral devices or between one MCU and RS232C COM port in computer
- UART like output

Program for transmitting 32 bytes at 60H on serial UART of 8051 with 11.0592 MHz Xtal

- TB8 used. Baud rate = 1200
- Message be 32 bytes at 60H [Between 60H and 7FH]
- One instruction cycle time = 1 µs /(11.0592/12) = 1.085 µs.

Calculations for Baud Rate ser 1200

- Required T_{serial} period= 1/(1200)] = 32 × 26.024 µs
- Clock inputs for 26.024 µs = 26.024 × 11.0592/12 = 24 when PCO^7 SMOD bit = 0
- The timer 1 will timeout after 24 clock inputs
- TH1= –24 = E8H for 26 µs for 11.0592 MHz Xtal

Program timer in mode 2

- Timer Mode 2 provides for auto reload mode and TH1 loads automatically in TL1 on each timer 1 overflow

Program

- SM0–SM1 is set to 11 for programmable baud rate in mode 3
- TB8 extra bit after 8 data bits
- SM2 = 0,
- REN = 0 (receiver disabled)
- TI is TxD SBUF empty flag

Without use of ISR

- When the interrupt structure is not to be used, then a continuous check for TI setting required
- Then sending the next character using SendUART3Char routine

Use of ISR at vector address 0x0023

- The best way in mode 3
- SBUF send bytes using the serial interface interrupts and ISR
- TI is reset at ISR instruction
- ISR calls routine to save the character and rewriting the next character into SBUF

Header lines in Program

- ; Program SIUARTM3TxD for sending 32 bytes from pointer address in R5 and number of bytes in R7 on serial SI asynchronous interface using mode 3
- ; Raj Kamal Jan 31, 2010

Header lines in Program

- ; Runs on 8051 with 12 MHz XTAL
- : Initialises Bank 3
- ; R7 = 20H used for pointing to 32 bytes of message at 60H

Header lines in Program

- ; Uses A, R7, R6 and R5.
- ; R6 = 60H. It points to address of the message having 20 characters
- ; R0 initialises from R6 and is used as pointer register for sending bytes into SBUF.
- ; Modifies A, PSW, Bank 3 data
- :Org 0x0976

Program (Main)

- StartInitials: MOV R6, #60H ; Move pointer to the next address after the characters of the message
- MOV R7, #20H ; Move counter for characters of message
- MOV R1, #20H; Define initial pointer for TB8 bits between 20H and 3FH nternal RAM

Program (Main)

- SETB RS0
- SETB RS1; Define RS1-RS0 [resister set (bank)] = 3

Program (Main)

- MOV A, R6 ; Move pointer for a message in A
- MOV R0, A; Move pointer of characters in the message for transfer in R0
- MOV A, R7 ; Move counter for a message in A
- MOV R5, A ; Move counter of characters in the message for transfer in R4

Instructions SIUARTM3TxD and SIUART1200

- SIUARTM3TxD: MOV SCON, # C0H ; SCON bits = 1100 0000 UART set for transmission with TB8. Variable baud, and receiver disable
- SIUART1200: MOV TH1, #E8H; Timer 1 load for Baud rate 1200

Instructions for Initialization

- MOV TMOD, #20H; Set TMOD bits = 00 10 0000. No external Gating, and Internal timer 1 mode 2
- SETB EA ; Interrupt Initialization
- SETB ES ; Serial Port Interrupts Initialization
- SETB TR1; Set TR1 to start and run timer 1
- SETB ET1; T1 interrupts initialisation

Instruction for Call Routine SendChar for sending first character

- ACALL SendChar ; Call Send first character at 0x0800. Next character sent on ISR interrupts
- INC R0 ; Incrément R0
- JB ES – 2 ; On serial interrupt disabled then only Next Instruction
- EndWait : SJMP – 2 ; Wait forever

ISRSerial3Tx

- Vector Address 0x0023-0x002A (Only 8 bytes available for the ISR) for ISRSerial3Tx
- DJNZ R5, Send; **Decrease R5 if no more characters left; disable serial interrupts and disable timer [CCY =2]interrupts and return [CCY = 2] ELSE JUMP to Send a character**
- Disables serial interrupts and ISR run after last character

Call to Stop Character after last character sent

- ACALL StopChar; Call stop timer and interrupts
- RTI ; Return from interrupt

Send a character by calling SendChar3

- Send: ACALL SendChar3
- CLR TI; Clear TI for next transmission– CCY=2] [CCY= 2]
- RETI; Return from interrupt

SendChar3 to transmit a character

- SendChar3:CLR TI; Clear TI for next transmission– [CCY= 1]
- MOV SBUF, @R0; Send Character on TxD [CCY= 2]
- INC R0; Increase R0 to point to next [CCY=1]
- MOV A, #0FFH; MOV data FFH into A[CCY=1]

Routine SendChar3

- ADD A, @R1 ; Add bit at the byte address pointed by R1. C =1 if TB8 =1 [CCY=1]
- MOV TB8, C; MOV C bit in TB8 [CCY=2]
- INR R1; Increase pointer R1 address [CCY=1]
- RET; Return [CCY= 2]

StopChar for Masking Serial Interrupts

- StopChar:CLR ES; Stop Serial Interrupts [CCY= 1]CLR TR1; Stop timer [CCY= 1]
- RET; Return [CCY= 2]

Summary

We learnt

- Programming the SCON
- Baud rate set at 1200 using timer T1
- First Character sent from main
- Remaining Characters using ISRSerial3Tx

End of Lesson 08 on

Programming Examples for Serial Port