**Microprocessor and microcontroller lab**

**Digital assignment 4**

**Q1:** Witte an 8051 program to get data from port PO and send it to port P1continuously while an interrupt will do the following: Timar will toggle the P2.1 bit every 100 microseconds

**Code**:

ORG 0000H

LIMP MAIN // Jump to main.

ORG 000BH // timer 0 Interrupt vector label

CPL P2.1 //Toggle P2.1 pin

RETI //Return from ISR

ORG 0030H // After Vector Table Space

MAIN: MOV TMOD #02H //set Timer 0 in mode 2

MOV PO, ROFFH // Set PD as I/P port

MOV THO, #-92

MOV IE, #100000108 //enable Timer O

SETB TRO // Start the Timer

BACK: MOV A, PO //Get data from PO.

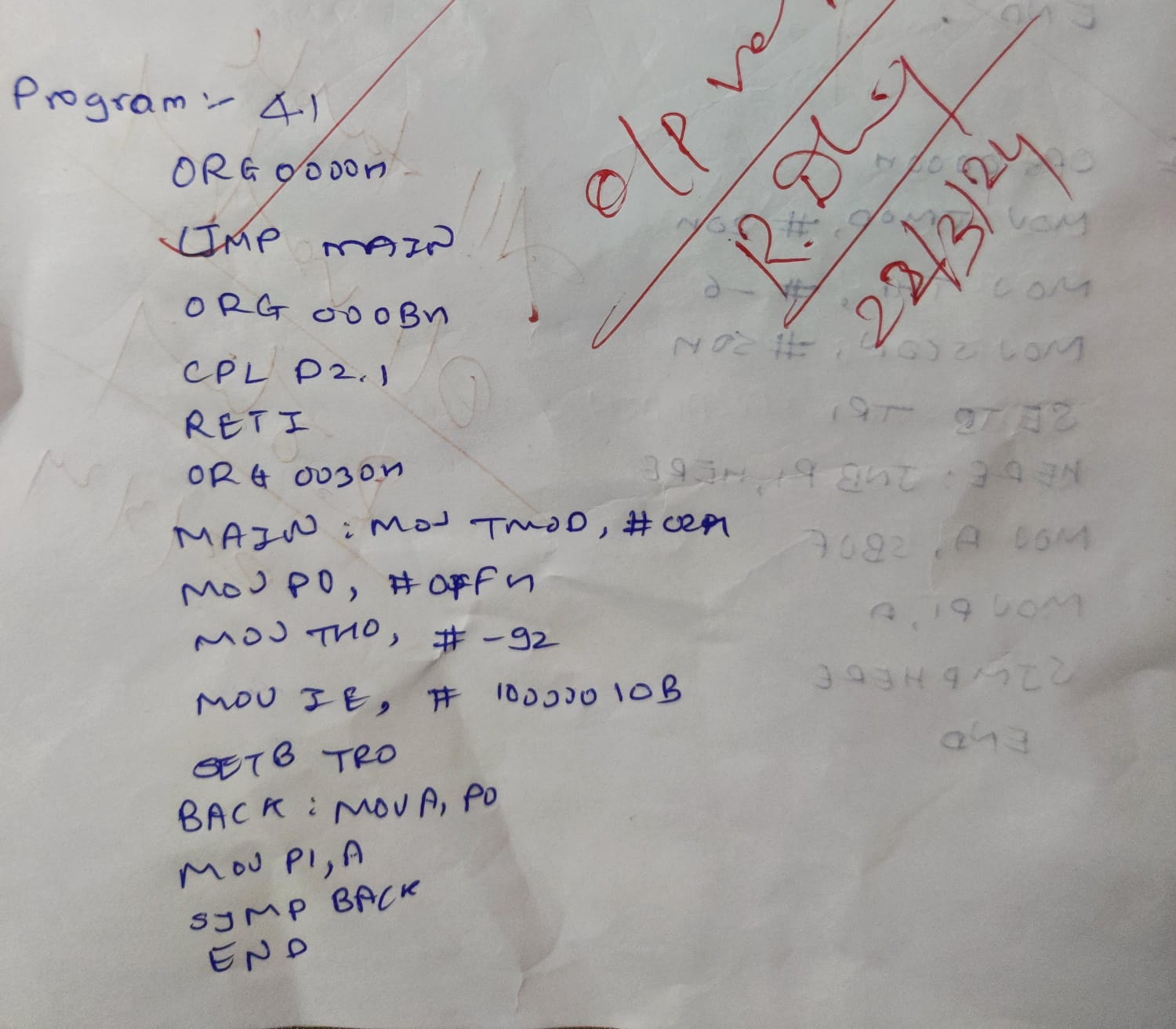
MOV P1, A //move data PD to P1

SIMP BACK //Loop it unless interrupted

END

**Conclusion:**

The assembly program successfully transfers the name serially at a baud rate of 9600 with 8-bit data and one stop bit. The transmitted data can be observed in the serial window of the simulator.



Output:

**Q2:** Write a program that continuously get 8-bit data from P1 and sends it to P2 while the INTI pin is connected to a switch which is normally high. Whenever it goes LOW, it should turn ON a LED. The LED is connected to P0.3 and is normally off. When it is turned ON, it should stay ON for fraction of second. As long as the switch is pressed LOW, the LED should stay ON.

**Code:**

ORG 0000H

LJMP MAIN ;by-pass interrupt vector table ISR for INTI to turn on LED

ORG 0013H ;INTI ISR

SETB P0.3 ;turn on LED

MOV R3,1255

BACK: DJNZ R3, BACK ;keep LED on for a while

CLR P0.3 ; turn off the LED RETI return from ISR

;MAIN program for initialization

ORG 30H

MAIN: MOV P1,#0FFH ;P1 AS INPUT

MOV P2,#00H ; P2 AS OUTPUT

CLR P0.3 LED: ;OUTPUT

HERE: MOV IE, #100001008 ;enable external INT 1

MOV A, P1

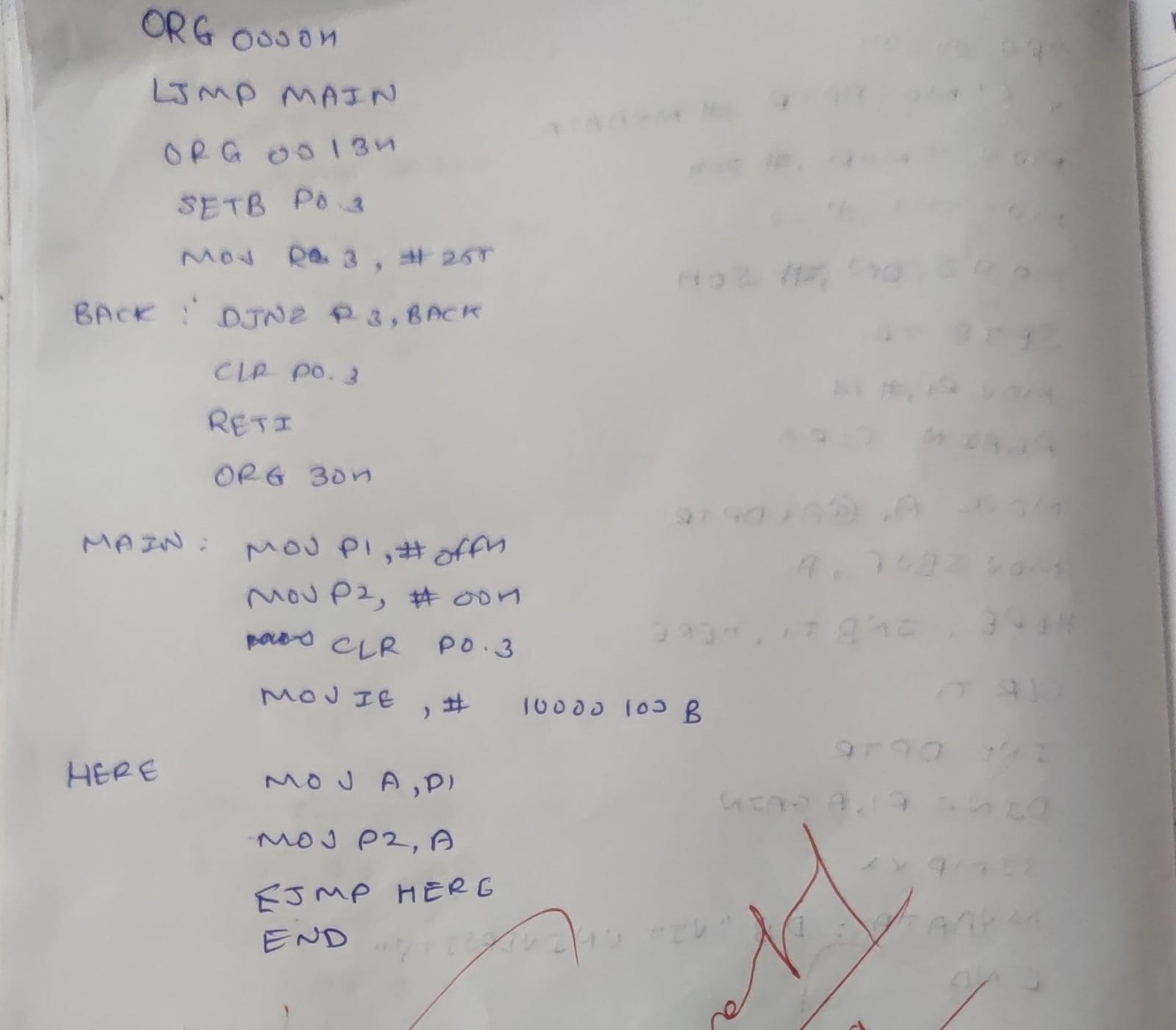
MOV P2,A

SJMP HERE ;stay here until get interrupted

END

**Conclusion:**

The program continuously monitors the data from port P1 and sends it to port P2. When the INT1 pin, connected to a switch, goes low, indicating it's pressed, the LED connected to pin P0.3 turns on for a brief moment. The LED remains on as long as the switch is pressed.



**Q3:**

**Code:**

ORG 0000H

LUMP MAIN

ORG 03H

LJMP TIMER

ORG 30H

MAIN:

MOV P0, FOFFH

MOV P1, OOH

MOV THO, #-92

CLR P2.1

MOV IE, #818

LOOP:

MOV A, PO

MOV P1, A

SJMP LOOP

TIMER:

SETB P2.1

SETB TRO

HERE: JNB TEO, HERE

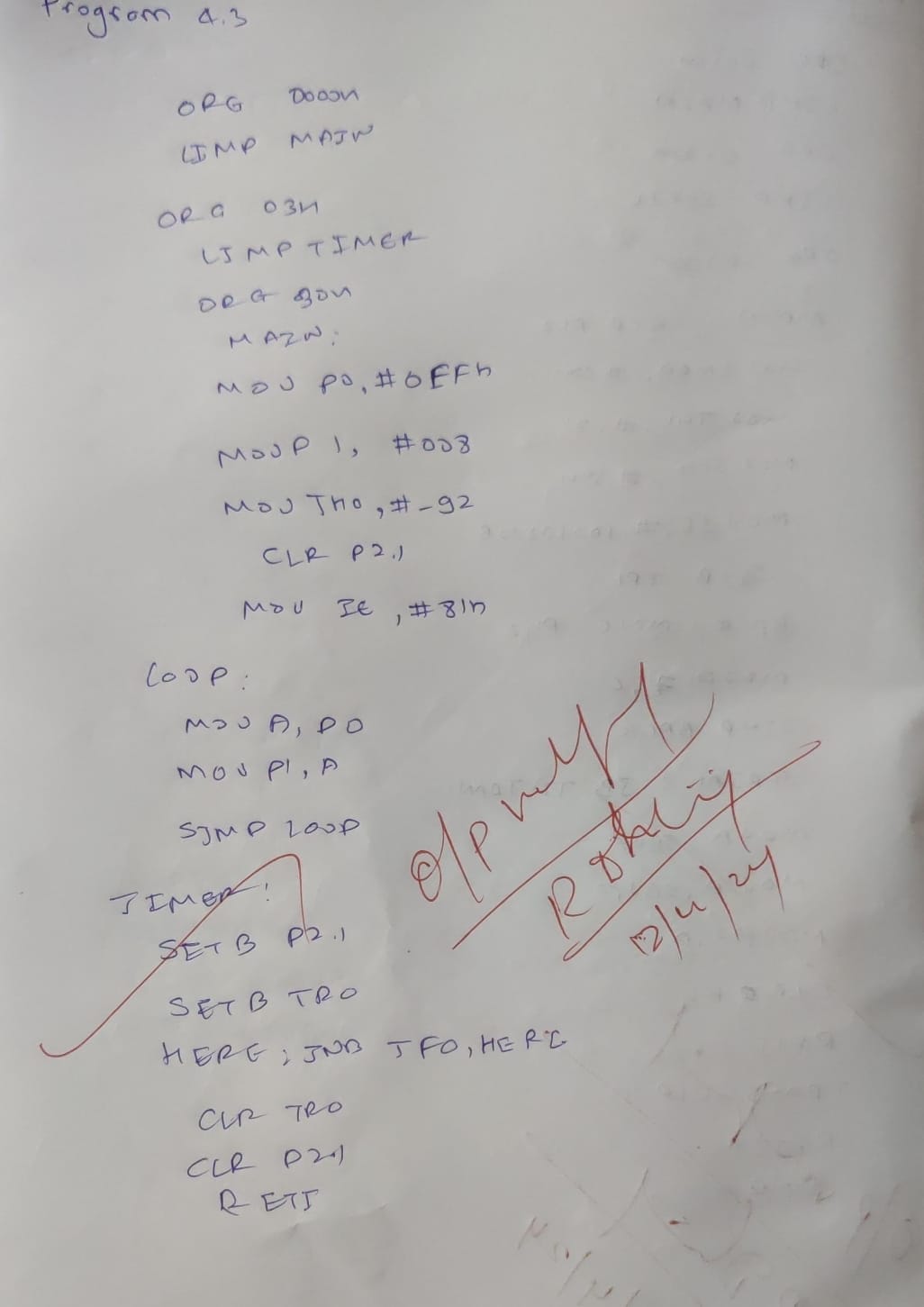
CLR TRO

CLR P2.1

RETI

**Conclusion:**

The program initializes ports P0 and P1, sets the timer TH0, and sets up interrupts. In the main loop, it continuously reads data from port P0 and sends it to port P1. Additionally, there's a timer interrupt that turns on a pin (P2.1) and waits until timer T0 overflows before turning it off.



**Q4**: Write an 8051 program to get data from a single bit of P1.2 and send it to P1.7 continuously while an interrupt will do the following: A serial interrupt service routine will receive data from a PC and display it on P2 ports. 9600 BAUD RATE.

**Code:**

ORG 0000H

LJMP MAIN

ORG 0023H ; serial interrupt vector table

LJMP SERIAL

ORG 0030H ;-- after vector table space

MAIN:SETB P1.2 ; -- P1.2 made as input pin

MOV TMOD,#20H ; -- timer 1 mode 2

MOV TH1,#-3 ;-- set baud rate 9600

MOV SCON ,#50H ; -- one stop bit MOV IE,#10010000B ; -- serial int. enabled. SETB TR1 ;-- Timer 1 stared.

BACK:MOV C,P1.2 MOV P1.7,C

SJMP BACK SERIAL:JB TI,TRANS MOV A,SBUF

MOV P2,A CLR RI

RETI TRANS:CLR TI RETI

END

**Conclusion:**

The 8051 program continuously reads data from bit P1.2 and sends it to bit P1.7. Meanwhile, a serial interrupt service routine receives data from a PC and displays it on port P2. The baud rate for serial communication is set to 9600.

