1:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Authors : Noam Solan & Ronen Rozin

;

; Date : 14/11/18

;

; File : mpl.lab4.asm

;

; Hardware : 8051 based processor

;

; Description : Transmitting and receiving from UART

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CSEG AT 0h

**JMP** MAIN

CSEG AT 023h ; serial interrupt address

**JMP** SERINT

CSEG AT 0100h

MAIN**:**

CLR SM0 ; turn off SM0

**SETB** SM1 ; turn on SM1

**SETB** REN ; turn on REN

CLR RI ; must be reset after every input

CLR ET1 ; clear timer1 interrupt

ANL PCON**,** #01111111b

ANL TMOD **,** #00101111b ; timer mode 2

ORL TMOD **,** #00100000b ; timer mode 2

**MOV** TH1 **,** #112d ; setting baud rate to 2400

**SETB** **ES** ; enable serial interrupt

**SETB** TR1 ; enable timer1

**SETB** EA ; enable interrupts

**JMP** **$** ; loop

SERINT**:**

JBC RI**,** INPUT ; clear RI and continue getting input

RETI

INPUT**:**

**PUSH** ACC ;preserving ACC

**MOV** ACC**,** SBUF ; get value from SBUF

XRL A**,** #00100000b ;convert case

**MOV** SBUF**,** ACC ; send value to SBUF

**POP** ACC ;restoring ACC

RETI

END

2:

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Authors : Noam Solan & Ronen Rozin

;

; Date : 14/11/18

;

; File : mpl.lab4.asm

;

; Hardware : 8051 based processor

;

; Description : The program will begin transmitting A to the UART until it gets a new input.

;Then it will begin transmitting the new input.

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CSEG AT 0h

**JMP** MAIN

CSEG AT 023h ; serial interrupt address

**JMP** SERINT

CSEG AT 0Bh ; timer0 interrupt

**JMP** CONINT0

CSEG AT 0100h

MAIN**:**

**MOV** VALUE**,** #'A'

CLR SM0 ; turn off SM0

**SETB** SM1 ; turn on SM1

**SETB** REN ; turn on REN

CLR RI ; must be reset after every input

CLR ET1 ; clear timer1 interrupt

ANL PCON**,** #01111111b

ANL TMOD **,** #00101101b ; timer1 mode 2, timer0 mode 1

ORL TMOD **,** #00100001b ; timer1 mode 2, timer0 mode 1

**MOV** TH1 **,** #112d ; setting baud rate to 2400

**SETB** **ES** ; enable serial interrupt

**SETB** TR1 ; enable timer1

**SETB** TR0 ; enable counter

**SETB** ET0 ; enable interrupts for timer0

**SETB** EA ; enable interrupts

**JMP** **$** ; loop

SERINT**:**

JBC RI**,** INPUT ; clear RI and continue getting input

RETI

INPUT**:**

**PUSH** ACC ;preserving ACC

**MOV** A**,** ;SBUF saving new input

XRL A**,** #00100000b ;convert case

**MOV** VALUE**,** A ; get value from SBUF

**POP** ACC ;restoring ACC

RETI

CONINT0**:**

**MOV** TH0**,** #027h

**MOV** TL0**,** #0F2h ; setting interrupt every 5ms

**MOV** SBUF**,** VALUE ; send value to SBUF

RETI

DSEG AT 30h

VALUE**:** **DS** 1

END