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

;

; Authors : Noam Solan & Ronen Rozin

;

; Date : 21/11/18

;

; File : Ex1.asm

;

; Hardware : 8051 based processor

;

; Description : Transmitting and receiving from UART

;

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

CSEG AT 0000H

**JMP** MAIN

CSEG AT 0023H ; change serial int

**JMP** SER

CSEG AT 0003H ; timer 3 INT

**JMP** EX\_INT0

CSEG AT 0100H ; main

MAIN**:**

CLR SM0

**SETB** SM1

**SETB** REN

ANL PCON **,** #07FH ; baud rate set to 19200

ANL TMOD **,** #02FH

ORL TMOD **,** #020H

**MOV** TH1 **,** #00EEH

**SETB** TR1 ; timer1 is on

CLR ET1 ; disable timer1 int

**SETB** **ES** ; enable port int

**SETB** EA ; enable all int

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

SER**:**

CLR TI ; clear transmit for uart

JBC RI**,**TRANS\_INT ;turn RI off and send back the value

**PUSH** ACC

CLR A

MOVC A **,** @A**+**DPTR ; jump to the printing function of the letter that should be printed

**JZ** NULL ; when null, skip, and set ex0 for next press

**MOV** SBUF **,** A ; print the value of A

**INC** DPTR ; get next letter from string

**JMP** STR\_NOTFINISHED ; string is not finished.

NULL**:**

**SETB** EX0 ; Enable ext int 0

STR\_NOTFINISHED**:**

**POP** ACC

RETI

EX\_INT0**:** ; interrupt 0 function

CLR EX0 ;close interrupts

**MOV** DPTR **,** #int0Press ;move dptr to the addres of ths string

**SETB** TI ; create an interrupt that will cause the print

RETI

TRANS\_INT**:**

CLR RI ;no other interupts from uart while sending

**MOV** R2 **,** SBUF ; store input from uart

;swtich case for letters a-d

CJNE R2 **,** #'a' **,**bCheck

**MOV** DPTR **,** #aPress

**JMP** TRANS\_INT\_END

bCheck**:**

CJNE R2 **,** #'b' **,** cCheck

**MOV** DPTR **,** #bPress

**JMP** TRANS\_INT\_END

cCheck**:**

CJNE R2 **,** #'c' **,** dCheck

**MOV** DPTR **,** #cPress

**JMP** TRANS\_INT\_END

dCheck**:**

CJNE R2 **,** #'d' **,** TRANS\_INT\_END

**MOV** DPTR **,** #dPress

TRANS\_INT\_END**:**

**SETB** TI

RETI

CSEG AT 0300H

aPress**:** DB 'Alpha'

DB 13

DB 10

DB 0

bPress**:** DB 'Beta'

DB 13

DB 10

DB 0

cPress**:** DB 'Charlie'

DB 13

DB 10

DB 0

dPress**:** DB 'Delta'

DB 13

DB 10

DB 0

int0Press**:** DB 'This button should not be pressed!'

DB 13

DB 10

DB 0

END

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

;

; Authors : Noam Solan & Ronen Rozin

;

; Date : 21/11/18

;

; File : mpl.lab4.asm

;

; Hardware : 8051 based processor

;

; Description : Transmitting and receiving from UART

;

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

CSEG AT 0000H

**JMP** MAIN

CSEG AT 0023H ; change serial int

**JMP** SER

CSEG AT 0100H ; main

MAIN**:**

CLR SM0

**SETB** SM1

**SETB** REN

ANL PCON **,** #07FH ; baud rate set to 19200

ANL TMOD **,** #02FH

ORL TMOD **,** #020H

**MOV** TH1 **,** #00EEH

**SETB** TR1 ; timer1 is on

CLR ET1 ; disable timer1 int

**SETB** **ES** ; enable port int

**SETB** EA ; enable all int

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

SER**:**

CLR TI ; clear transmit for uart

JBC RI**,**TRANS\_INT ;turn RI off and send back the value

**PUSH** ACC

CLR A

MOVC A **,** @A**+**DPTR ; jump to the printing function of the letter that should be printed

**JZ** NULL ; when null, skip

**MOV** SBUF **,** A ; print the value of A

**INC** DPTR ; get next letter from string

NULL**:**

**POP** ACC

RETI

TRANS\_INT**:**

CLR RI ;no other interupts from uart while sending

MOV R2 , SBUF ; store input from uart

;swtich case for letters a-d

CJNE R2 , #'a' ,bCheck

MOV DPTR , #aPress

JMP TRANS\_INT\_END

bCheck:

CJNE R2 , #'b' , cCheck

MOV DPTR , #bPress

JMP TRANS\_INT\_END

cCheck:

CJNE R2 , #'c' , dCheck

MOV DPTR , #cPress

JMP TRANS\_INT\_END

dCheck:

CJNE R2 , #'d' , TRANS\_INT\_END

MOV DPTR , #dPress

TRANS\_INT\_END:

SETB TI

RETI

//data segment

CSEG AT 0300H

aPress: DB 'Alpha'

DB 13

DB 10

DB 0

bPress: DB 'Beta'

DB 13

DB 10

DB 0

cPress: DB 'Charlie'

DB 13

DB 10

DB 0

dPress: DB 'Delta'

DB 13

DB 10

DB 0

END