

פרק 12

פסיקות חומרה

מבוא לפרק

- ▶ עד כה נתקלנו במספר פסיקות DOS שמבצעות פעולות מול רכיבי חומרה
- ▶ טרם עסקנו בשאלה- איך מידע מהתקני החומרה מגיע אל המעבד?
- ▶ בפרק זה נבין לעומק את הדרך בה המעבד מתקשר עם התקני חומרה
 - המסלול שפסיקת חומרה עוברת מהחומרה עד המעבד
 - פורטים I/O ports
 - דוגמה מעשית: המקלדת

תרגיל מסכם פסיקות

- ▶ צרו תוכנית שעם הקשה על מקש כלשהו במקלדת מבצעת את הפעולות הבאות:
 - משמיעה צליל בתדר 440 הרץ
 - מדפיסה למרכז המסך ריבוע של 3×3 פיקסלים אדומים
- ▶ עם שחרור מקש המקלדת, תופסק השמעת הצליל ויימחק הריבוע מהמסך

Interrupts- רקע

- ▶ המעבד מקושר להתקנים חיצוניים
- ▶ התקנים אלו מייצרים אירועים חומרתיים
 - לחיצה / שחרור מקש במקלדת
 - הזזת העכבר
 - עדכון של הטיימר
- ▶ מבחינת המעבד- כל אירוע הוא בלתי צפוי בזמן
- ▶ יש צורך לעדכן את המעבד ולהפעיל קוד מתאים לטיפול במידע שמתקבל מהחומרה
 - קיימות שתי גישות לפתרון הבעיה:
 - Polling
 - Interrupts

- ▶ אחת לזמן מוגדר מראש, התוכנה שרצה על המעבד שואלת כל התקן חומרה אם יש לו מידע חדש
- ▶ התשאול מתבצע בצורה מעגלית ("round robin")
- ▶ התקן החומרה מסבילנות שיפנו אליו



polling

▶ יתרונות:

- גישה פשוטה יחסית למימוש
- אין צורך בחומרה נוספת.

▶ חסרונות:

- עיכוב בטיפול בהתקני חומרה
- בזבוז משאבי המעבד - רוב הזמן התשובה תהיה שאין מידע חדש

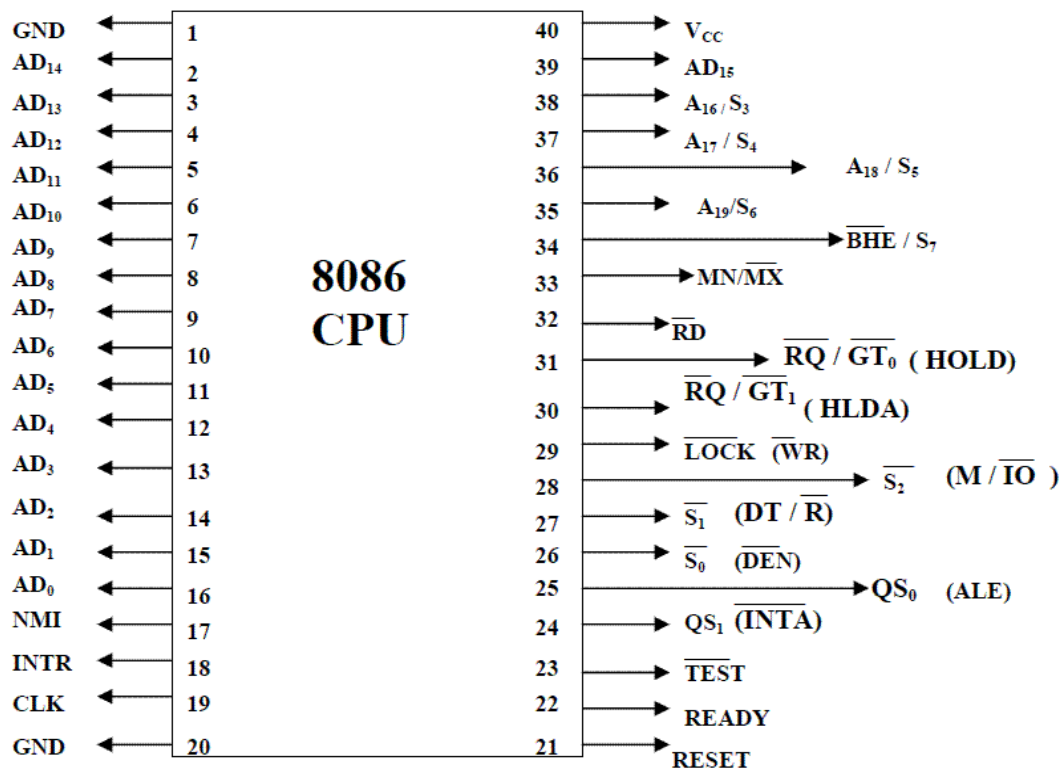
Interrupts

- ▶ מעכשיו, כשנכתוב interrupt נתכוון לפסיקת חומרה
- ▶ כשיש להתקן החומרה צורך בשירות, הוא שולח אות חשמלי
- ▶ המעבד קוטע את ריצת התוכנית, מטפל ב-interrupt ולאחר מכן חוזר לבצע את התוכנית



דיאגרמת רגליים של מעבד ה-8086

Pin Diagram of 8086



▶ רגל מספר 18
מסומנת כ-INTR,
מקבלת אינטרפטים
מרכיבי חומרה
חיצוניים

בקר האינטרפטים- PIC

► איך בעזרת רגל אחת המעבד יכול להיות קשור למספר רב של התקני חומרה?

- קיים רכיב נוסף שמרכז את כל האינטרפטים- בקר האינטרפטים
- Programmable Interrupt Controller
- ל-PIC שמונה רגליים שמחוברות להתקני חומרה

- IR0- טיימר

- IR1- מקלדת

- IR2- עכבר

- וכו'

- ל-PIC רגל שנקראת INT

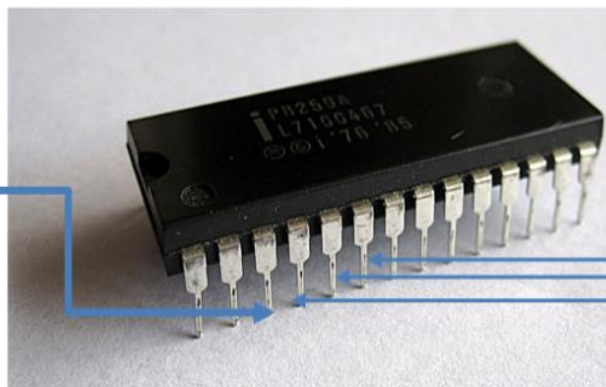
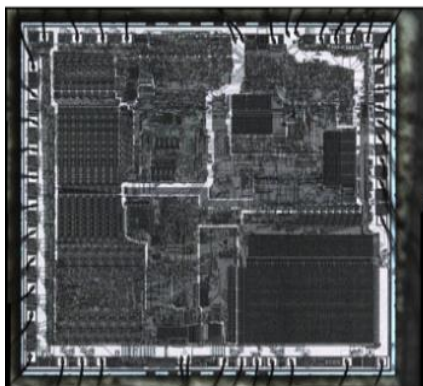
- מחוברת חשמלית אל המעבד



8259A PIC Microcontroller with all pins labeled.

מסלול האות החשמלי

מעבד



עכבר

מקלדת

טיימר

עם קבלת אינטרפט:

- ה-PIC שולח למעבד אות חשמלי שמסמן שיש אינטרפט
- שולח מידע לאזור מיוחד בזיכרון - I/O port - פורט
- מפסיק לשלוח אינטרפטים למעבד
- שומר אינטרפטים חדשים בתור
- מחכה לאות חשמלי end of interrupt מהמעבד
- חוזר לשלוח אינטרפטים למעבד

ה-PIC, המשך

▶ עד עכשיו תיארנו את ה-PIC כגורם מקשר בין רכיבי חומרה למעבד

▶ ה-PIC גם קובע עדיפויות להתקני החומרה

- משתמש הקיש על המקלדת

- הכור הגרעיני דיווח על בעיה

- ...במי צריך לטפל קודם?

▶ ב-PIC יש תור של אינטרפטים

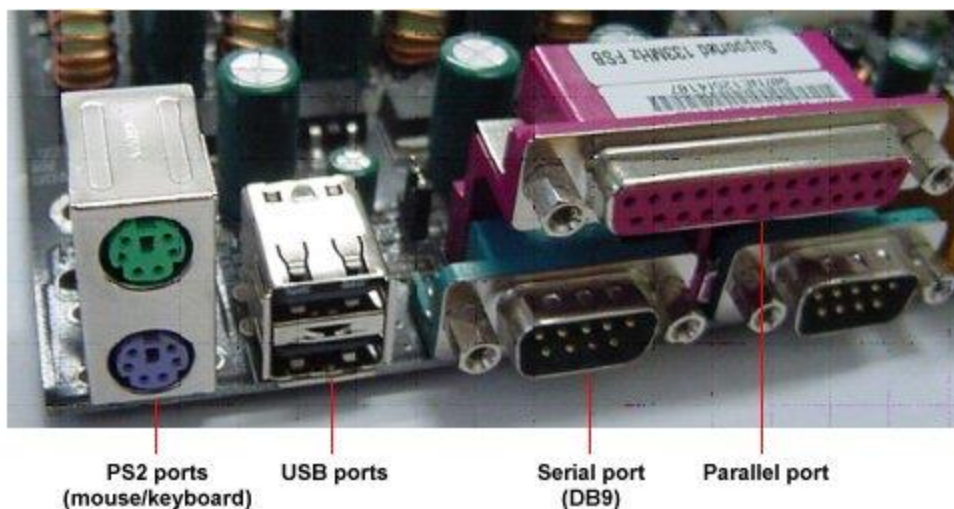
- מנוהל ע"פ עדיפויות

- יש אפשרות לאובדן אינטרפטים



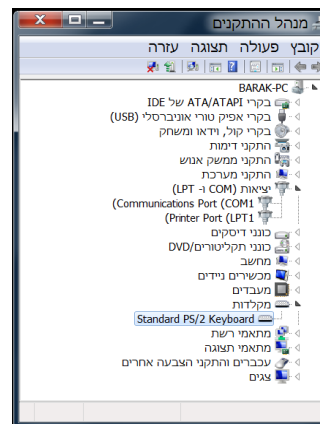
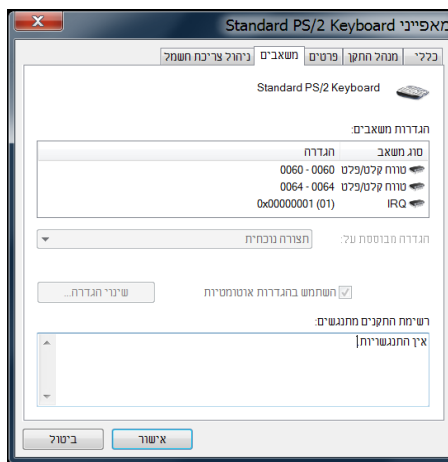
I/O Ports

- ▶ בפרק אודות מבנה המחשב סקרנו את הפסים (buses) השונים שיש למעבד, בין היתר ה-address bus
- ▶ למעשה למעבד ישנם שני address buses
 - פס ברוחב 20 ביט- לזיכרון ה"רגיל"
 - פס ברוחב 16 ביט- לזיכרון קלט פלט - I/O / Input / Output



תרגיל- מציאת פורט המקלדת

- ▶ מהו פורט התקשורת עם המקלדת? הדרכה:
 - מנהל ההתקנים (חיפוש תכנית- device manager)
 - בתוך מנהל ההתקנים, מסמנים את המקלדת ומקיישים עכבר ימני ואז "מאפיינים" properties
 - בוחרים "משאבים", מופיעים הפורטים שבשימוש



תקשורת עם I/O

▶ התקשורת עם זיכרון I/O עובדת כמו עם זיכרון רגיל, למעט מספר הבדלים:

- כתובת של זיכרון I/O נקראת פורט Port
- כתובת של זיכרון I/O מיוצגת ע"י 16 ביט בלבד
 - סה"כ יש 64K פורטים אפשריים
- פס הבקרה דואג שפקודות in, out, יתקשרו לזיכרון ה-I/O ואילו פקודות mov יתקשרו עם הזיכרון הרגיל

פקודות in, out

- ▶ פקודת in משמשת לקריאה מפורט
 - מעתיקה את המידע מהפורט אל ax או al
- ▶ פקודת out משמשת לכתיבה לפורט
 - מעתיקה את ax או al אל הפורט

in	al, 61h
out	61h, al

mov	dx, 300h
in	al, dx

- ▶ שיטות כתיבה:
 - ישירות: רושמים את מספר הפורט, בין 0-255 בלבד
 - בעקיפין: פורט מעל 255, משתמשים ב-dx

- ▶ נפרט את שרשרת הפעולות שמתבצעות מרגע ההקשה על מקש במקלדת ועד קבלת תו בזיכרון המעבד:
 - המקלדת יוצרת מידע- הקשה ושחרור של מקשים
 - מידע זה נקרא scan codes
 - המקלדת מקושרת לזיכרון ה-I/O של המעבד, פורט 60h
 - אינטרפט אוסף את ה- scan codes מפורט 60h ומעתיק אותם אל באפר (מערך בזיכרון) מיוחד
 - אינטרפט 9h
 - שם הבאפר הוא Type Ahead Buffer



דרכים מעשיות לתקשורת עם המקלדת

► עבודה ישירות מול הפורטים של המקלדת

- פסיקה מספר h 9

► פסיקה של BIOS

- חבילת תוכנה של אינטל

- Basic Input Output System

- פסיקה מספר h 16

- "עוטפת" את פסיקה h 9 עם קוד נוסף

► פסיקה של DOS

- פסיקה מספר h 21

- "עוטפת" את הקוד של BIOS

כיוון שיש פסיקות
זמינות לשימוש
ממקורות שונים,
ישנן מספר דרכים
לבצע את אותה
הפעולה

יצירת scan codes ושליחה למעבד

► במקלדת יש רכיב, ששולח ל-PIC קוד עם כל לחיצה ושחרור

- מצב המקשים הלחוצים חשוב לטובת פעולה הגיונית
- טבלת קודים- הן ללחיצה והן לשחרור

Key	Down	Up	Key	Down	Up	Key	Down	Up	Key	Down	Up
ESC	1	81	[{	1A	9A	, <	33	B3	center	4C	CC
1 !	2	82] }	1B	9B	. >	34	B4	right	4D	CD
2 @	3	83	Enter	1C	9C	/ ?	35	B5	+	4E	CE
3 #	4	84	Ctrl	1D	9D	R shift	36	B6	end	4F	CF
4 \$	5	85	A	1E	9E	PrtSc	37	B7	down	50	D0
5 %	6	86	S	1F	9F	alt	38	B8	pgdn	51	D1
6 ^	7	87	D	20	A0	space	39	B9	ins	52	D2
7 &	8	88	F	21	A1	CAPS	3A	BA	del	53	D3
8 *	9	89	G	22	A2	F1	3B	BB	/	E0 35	B5
9 (0A	8A	H	23	A3	F2	3C	BC	enter	E0 1C	9C
0)	0B	8B	J	24	A4	F3	3D	BD	F11	57	D7
- _	0C	8C	K	25	A5	F4	3E	BE	F12	58	D8
= +	0D	8D	L	26	A6	F5	3F	BF	ins	E0 52	D2
Bksp	0E	8E	;	27	A7	F6	40	C0	del	E0 53	D3
Tab	0F	8F	"	28	A8	F7	41	C1	home	E0 47	C7
Q	10	90	` ~	29	A9	F8	42	C2	end	E0 4F	CF
W	11	91	L shift	2A	AA	F9	43	C3	pgup	E0 49	C9
E	12	92	\	2B	AB	F10	44	C4	pgdn	E0 51	D1
R	13	93	Z	2C	AC	Num	45	C5	left	E0 4B	CB
T	14	94	X	2D	AD	SCRL	46	C6	right	E0 4D	CD
Y	15	95	C	2E	AE	home	47	C7	up	E0 48	C8
U	16	96	V	2F	AF	up	48	C8	down	E0 50	D0
I	17	97	B	30	B0	pgup	49	C9	R alt	E0 38	B8
O	18	98	N	31	B1	-	4A	CA	R ctrl	E0 1D	9D
P	19	99	M	21	B2	left	4B	CB	pause	E1 1D	-

שליחת scan code למעבד

- ▶ דוגמה: לחיצה על esc תגרום לשליחת קוד h 1. שחרור
מקש ה-esc יגרום לשליחת קוד h 81
- ▶ הפעולות שמתבצעות:
 - רכיב חומרה במקלדת שולח לפורט h 60 את ה-scan codes
 - ה-PIC מקבל דרך IR1 אינטרפט מהמקלדת
 - ה-PIC שולח למעבד אינטרפט, שאומר שיש מידע בפורט המקלדת
 - כתגובה לאינטרפט, המעבד מריץ ISR שמספרו h 9
 - ה-ISR מטפל בהעתקת ה-scan code אל ה-type ahead buffer
 - בסיום ריצת ה-ISR, המעבד שולח ל-PIC סימן end of interrupt

Type Ahead Buffer

- ▶ ה-ISR שמופעל ע"י אינטרפט 9h לוקח את ה- scan code ומתרגם אותו לקוד ASCII
 - התרגום תלוי באילו עוד מקשים היו לחוצים- לדוגמה A, a
- ▶ ה-ISR מעתיק את ה-scan code ואת קוד ה-ASCII אל באפר המקלדת
 - מחזיק עד 16 הקלדות
 - מיקום 001A:0040- מצביע על ראש הבאפר
 - מיקום 001C:0040- מצביע על זנב הבאפר
 - מיקום 001E:0040- 16 מילים (words)

Type Ahead Buffer - תרגיל

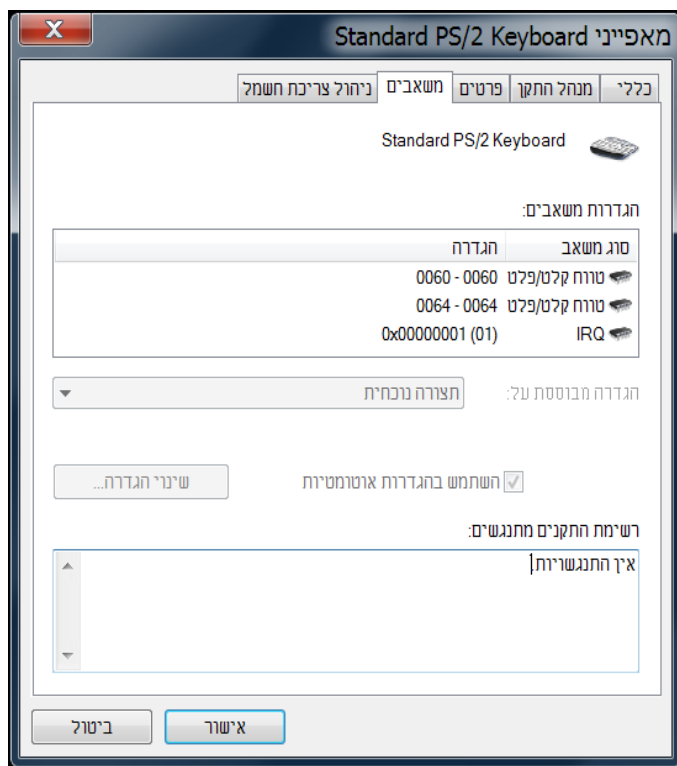
- ▶ ה-ISR של המקלדת מעתיק את ה-scan code אל ה-Type Ahead Buffer שמתחיל במיקום 0040:001Eh בזיכרון המעבד.
- ▶ כיתבו תוכנית שקוראת תו מהמקלדת (השתמשו ב-`int h 21` עם הקוד המתאים), הריצו את התוכנית ב-TD במצב `step by step`, וצפו בשינוי בזיכרון במיקום של ה-type ahead buffer. בתור קלט, הכניסו את התו 'a' ומצאו ב-type ahead buffer את ה-scan codes שלו.

Type Ahead Buffer - המשך

- ▶ אינטרפט המקלדת מגדיל את ערכו של הזנב ב-2 בתים
 - אם הערך יוצא מהבאפר- הוא מוחזר לתחילת הבאפר
 - אם ה"זנב" עוקף את ה"ראש"- המידע שבבאפר הולך לאיבוד
- ▶ פעולת הוצאה של נתון מהבאפר ("ניקוי"):
 - העתקה של המילה מראש הבאפר
 - קידום ערכו של הראש ב-2 בתים
 - ישנן מספר אפשרויות לביצוע:
 - ע"י המתכנת
 - ע"י פסיקת BIOS
 - ע"י פסיקת DOS
- ▶ כעת נראה איך מבצעים קריאה וניקוי הבאפר בכל השיטות

שיטה א' - ע"י פורטים של מקלדת

- ▶ פורט h60 - דרכו מקבלים scan codes
- ▶ פורט h64 - הודעות בקרה (האם יש scan code חדש)



פורטים של המקלדת- המשך

► קוד שבודק אם הגיע scan code חדש:

```
in      al, 64h      ; Read keyboard status port
cmp     al, 10b      ; Data in buffer ?
```

► קוד שקורא scan code מהפורט:

```
in      al, 60h
```

► קוד שבודק אם ה-scan code הוא לחיצה או שחרור מקש:

```
and     al, 80h      ; if the result is 0 a key is pressed
                        ; and if it is 1 a key is released
```

פורטים של המקלדת- קוד דוגמה

► קוד שקורא את המקלדת עד לחיצה על ESC:

WaitForData :

```
in      al, 64h
cmp     al, 10b
je      WaitForData
in      al, 60h
cmp     al, 1h
jne     WaitForData
```

```
; Read keyboard status port
; Data in buffer ?
; Wait until data available
; Get keyboard data
; Is it the ESC key ?
```

ESCPressed:

...

```
C:\TASM\BIN>keyboard
ESC key pressed
```

```
C:\TASM\BIN>012345678901234_
```

► הערה: כיוון שאיננו מנקים את הבאפר, רק 15 תווים יישמרו בו

שיטה ב'- פסיקת BIOS

▶ פסיקת BIOS מספר $16h$ נותנת כלים נוחים:

- בדיקת מצב המקלדת
- קריאת התו שהוקלד
- ניקוי באפר המקלדת

▶ פסיקה $16h$ עם קוד $ah=1$: סטטוס מקלדת

- 0 אם יש תו מוכן
- 1 אם אין תו מוכן
- al יקבל את ערך ה-ASCII של התו
- ah יקבל את ה-scan code

פסיקת BIOS- קוד דוגמה

WaitForData :

; check keyboard status

mov ah, 1

Int 16h

jz WaitForData

; read key

mov ah, 0

int 16h

; exit if ESC pressed

cmp ah, 1h

jne WaitForData

ESCPressed:

...

קריאה של ▶

תווים עד

שהמשתמש

מקיש על ESC:

פסיקת BIOS - תרגיל

▶ במשחקי מחשב שונים, המקשים wasd משמשים לתזוזת השחקן:

W = up ◦

A = left ◦

S = down ◦

D = right ◦

▶ כיתבו תוכנית שמאזינה למקלדת. אם הוקש אחד ממקשי wasd, יודפס למסך "Move up", "Move down" וכו'. אם הוקש מקש ה־Esc, התוכנית תצא. כל מקש אחר – התוכנה לא תעשה דבר. כדי לדמות משחק מחשב, השתמשו בפסיקת BIOS (אינה עוצרת את ריצת התוכנית בהמתנה לקלט).

שיטה ג' - פסיקת DOS

▶ פסיקה 21h עם קוד ah=0Ch , al=7h

- מנקה את באפר המקלדת
- קולטת תו מהמשתמש ללא echo (ללא הדפסה על המסך)
- בסיום, al מכיל את קוד ה-ASCII של התו
- חסרון: התוכנית עוצרת בהמתנה לקלט (בניגוד לפסיקת BIOS)

; Clear keyboard buffer and read key without echo

```
mov    ah,0Ch  
mov    al,07h  
int    21h
```

```
; make a sound
in      al, 61h
or      al, 00000011b
out     61h, al
mov     al, 0b6h
out     43h, al
mov     ax, 2394h
out     42h, al
mov     al, ah
out     42h, al
```

```
; stop sound
in      al, 61h
and     al, 11111100b
out     61h, al
```

▶ כיתבו תוכנית שברגע שנלחץ מקש כלשהו מוציאה צליל, ועם שחרור המקש מפסיקה את השמעת הצליל.

▶ הדרכה:

- התוכנית תשתמש בפסיקה 16h כדי לבדוק אם יש מידע חדש מהמקלדת. אם יש מידע חדש, התוכנית תבדוק בעזרת פורט 60h אם ה-scan code מתאים ללחיצה או לשחרור ובהתאם יופעל קטע הקוד שמשמיע צליל או קטע הקוד שמפסיק את השמעת הצליל.
- לשימושכם הקוד הבא, גורם לכרטיס הקול להשמיע צליל.
- לשימושכם הקוד הבא, גורם לכרטיס הקול להפסיק את השמעת הצליל.

תרגיל מסכם

- ▶ צרו תוכנית שעם הקשה על מקש כלשהו במקלדת מבצעת את הפעולות הבאות:
 - משמיעה צליל בתדר 110 הרץ
 - מדפיסה למרכז המסך ריבוע של 3×3 פיקסלים אדומים
- ▶ עם שחרור מקש המקלדת, תופסק השמעת הצליל ויימחק הריבוע מהמסך
- ▶ הדרכה:
 - התבססו על תרגיל המקלדת
 - פרק 12 בספר הלימוד- הדרכה איך קובעים תדירות צליל
 - פרק 12 בספר הלימוד- הדרכה איך מדפיסים פיקסל למסך

