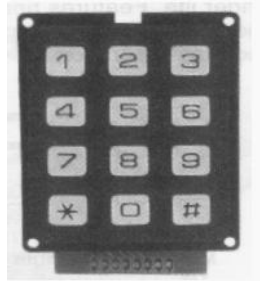


לוח מקשים 3X4



שרטוט לוח המקשים:

1	2	3
4	5	6
7	8	9
*	0	#

מבוא:

לוח מקשים 3X4 זהו התקן קלט, המקשר בין המשתמש למערכת. התקן זה כולל 12 לחצנים המסודרים ב-4 שורות ו-3 עמודות. הלוח כולל ספרות מ-0 עד 9 ובנוסף את המפסקים המסומנים ב * ו - # . קיימים מספר סוגי לוחות מקשים. אחד הנפוצים הוא **לוח מקשים מסוג מטריצה**. הלוח בנוי בצורת מטריצה של 4 שורות Rows ו 3 עמודות - Column. כאשר בכל מפגש של שורה עם עמודה מחובר לחצן, לחיצה על מקש מסוים מקצרת בין עמודה ושורה מסוימים. ללוח מקשים מסוג מטריצה כזו יש קונקטור בן 7 הדקים. היתרון בצורת מטריצה זו היא מס' קווים קטן. בפרויקטים רבים משמש לוח המקשים בכדי להכניס סיסמא או כדי לנהל דו שיח עם המשתמש.

מטריצת לוח המקשים:

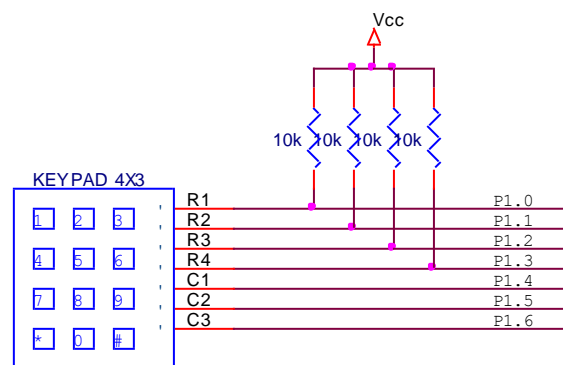
	1	2	3
R1			
	4	5	6
R2			
	7	8	9
R3			
	*	0	#
R4			
	C1	C2	C3

בשרטוט לא נראים המפסקים אבל ניתן להבין שכל לחיצה על מקש מקצרת בין שורה ועמודה בהתאם. לא יתכן מצב שאותה שורה ועמודה מקוצרות פעמיים מ-2 לחיצות מקש שונות.

טבלת הצטלבות השורות והעמודות:

המקש	עמודה (קו אנכי)	שורה (קו אופקי)
1	C1	R1
2	C2	R1
3	C3	R1
4	C1	R2
5	C2	R2
6	C3	R2
7	C1	R3
8	C2	R3
9	C3	R3
*	C1	R4
0	C2	R4
#	C3	R4

חיבור לוח המקשים במערכת:

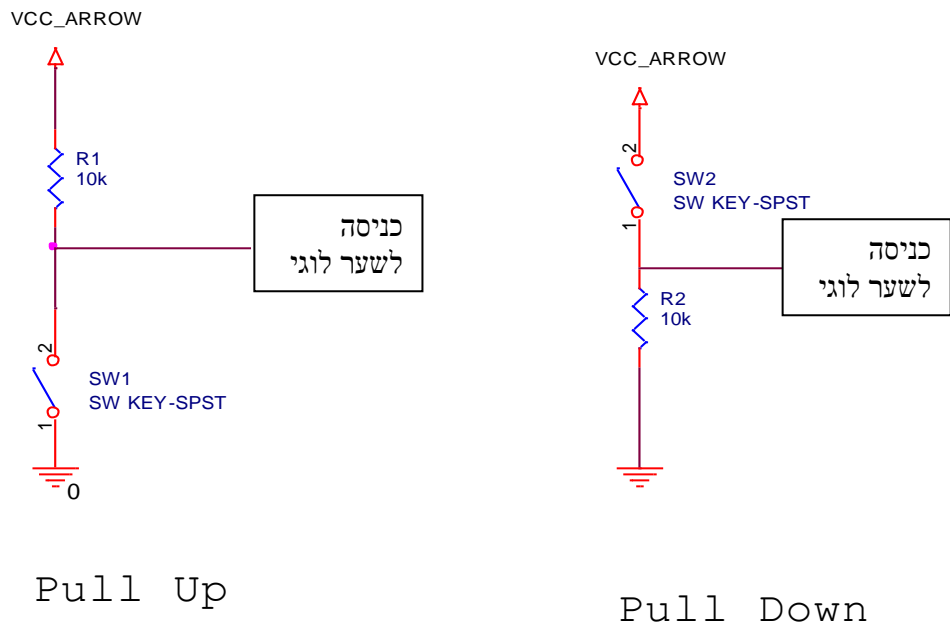


ניתן לראות ש 4 השורות ו 3 העמודות מתחברות לפורט 1 של המיקרו בקר. רגלי השורה – R_n משמשים כקלט ורגלי העמודה כפלט בצורה שתוסבר בהמשך. רגלי השורה מתחברים אל פורט 1 בעזרת נגדי משיכה למעלה – Pull Up Resistors שתפקידם לדאוג שבמצב שאין לחיצה על המפסק הרגל של הפורט תהיה ב '1' לוגי.

הצורך בנגדי Pull Up:

רצוי לזכור את הכלל שאין להשאיר רגלי כניסה של מעגל דיגיטאלי (במיוחד ממשפחת ה CMOS) באוויר כי אז מצבו הלוגי לא מוגדר ורעש יכול להעביר את הכניסה ממצב של 0 ל 1 או להיפך. יש

להגדיר לכניסה את מצבה הלוגי בכל מקרה. כאשר נחבר מפסק לכניסת שער לוגי והמפסק יהיה פתוח אז הרמה הלוגית איננה מוגדרת. לכן יש לחבר מפסק באחת משתי הצורות Pull Up או Pull Down. בשרטוט הבא רואים את שתי אפשרויות החיבור של מפסק אל כניסת שער לוגי.



במעגל הימני Pull Down, כאשר המפסק פתוח, כניסת השער נמצאת באדמה (מתח נמוך) ולחיצה על המפסק תעביר 1 אל השער. בשרטוט השמאלי Pull Up כאשר המפסק פתוח יש בכניסת השער '1' לוגי (גבוה) ולחיצה על המפסק תעביר '0' אל הכניסה.

לוח המקשים המתחבר אל פורט 1 מתחבר עם 4 נגדי Pull Up. כל אחד מקווי השורה (R1-R4) מחובר לנגד משיכה למעלה ולמקור המתח. כל עוד לא אולצה רמה לוגית שונה, נמצאים כל קווי השורה ב-'1' לוגי. שיטת העבודה כדי לאתר האם יש מקש לחוץ נקראת "אפס רץ".

ניתן לחבר את לוח המקשים, במקום למתח של 5 וולט, אל מתח '0' – אדמה - ואז במקום לעבוד עם '0' רץ ניתן לעבוד עם '1' רץ. אופן עבודה כזה עם הנגדים נקרא **Pull Down**.

בחירת ערכי הנגדים

הסיבה לבחירת ערכי הנגדים ל 10K היא: מצד אחד לא רוצים שהנגדים יהיו גדולים מידי ואז נקבל חלוקת מתח עם התנגדות הפורט ולא יהיה '1' לוגי בכניסת הפורט גם כאשר אף לחצן לא לחוץ. מצד שני לא ניקח נגדים קטנים מידי כי בזמן שלוחצים על אחד המפסקים, המפסק מעביר את ה '0' אל אחת מרגלי פורט 1 המשמש כשורה. '0' זה גורם למפל מתח של 5 וולט על הנגד. אם הוא קטן מידי יזרום דרכם זרם גדול שיגרום לחימום הנגדים ולבזבוז הספק או אפילו לשרוף את אותה רגל של פורט 1 - העמודה שבה יש את ה '0'. נגדים בסדר גודל של קילו אומים עד עשרות קילו אומים יעשו את תפקידם נאמנה.

הערה :

במערכות מיקרו בקרים הבנויות בטכנולוגיית NMOS אין צורך לחבר נגדי משיכה למעלה חיצוניים. מעבירים את הפורט למצב קלט ואז בתוך הרכיב מתחבר נגד משיכה למעלה (למעשה מתחבר טרנזיסטור FET עם ממתח V_{gs} שנותן התנגדות בסדר גודל של עשרות קילו אוהם).

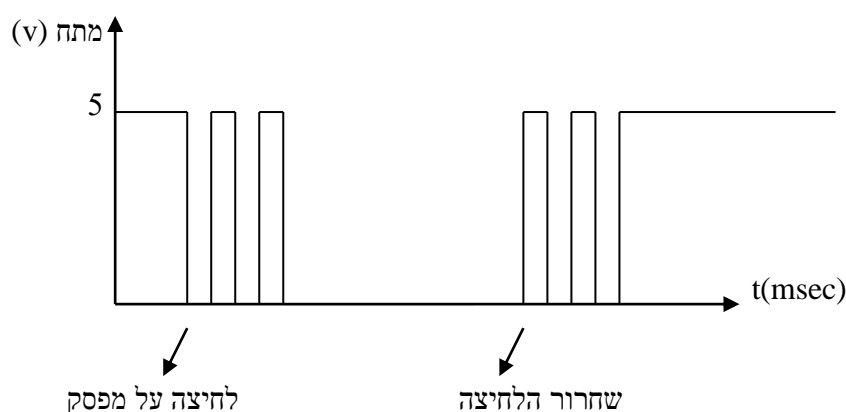
שיטת '0' רץ

בשיטה זו שמים '0' באחת העמודות ו '1' בשתי העמודות האחרות ואז בודקים את מצב השורות. אם 4 השורות נמצאות ב '1' אז אף מקש בעמודה זו איננו לחוץ ואז מעבירים את האפס לעמודה הבאה ושוב מבצעים את אותה הבדיקה על השורות. כך עושים גם לגבי העמודה השלישית. במידה ובאחת השורות מגלים שאין '1', כלומר יש '0' זה אומר שאחד המקשים לחוץ ואז נמצא בעזרת תוכנה באיזו שורה נמצא ה '0'. המפגש של העמודה אליה הוצאנו '0' עם השורה שבה קיבלנו את ה '0' נותן לנו את מספר המקש.

למשל, נניח שלחצנו על מקש '5', מקש זה נמצא בעמודה 2 ושורה 2. אנו נתחיל לסרוק את עמודות המקלדת, לשם כך נקבע '0' לוגי בעמודה מספר 1 ובשאר העמודות '1' לוגי. כעת נקרא את פורט 1 ונשים ב 4 הביטים הגבוהים שלו '1'. מכיוון שהמקש שנלחץ הוא לא מעמודה מס' 1 הערך שיקרא הוא FFh (כלומר אין מקש לחוץ). אנו נמשיך בסריקה וכעת נקבע '0' לוגי בעמודה 2 ובשאר העמודות '1' לוגי. כעת נקרא את פורט 1 ונשים '1' ב 4 הביטים הגבוהים ואז נראה שהערך שקיבלנו הוא FDH (נקבל '0' בשורה R2) ולא FFh. מכאן שהמקש הנלחץ הוא מעמודה זו. כעת נמצא את הסיבית שערכה '0' מעמודה זו וניתן לדעת שזה 5.

בעיית ריטוט המגעים (Bouncing) ופתרונם (De-bouncing)

כידוע בכל מפסק קיימת בעיה של ריטוט המופיע בזמן המעבר של המפסק. לחיצה על המפסק גורמת למגעיו לריטוט (להתחבר ולהתנתק) מספר פעמים עד להתייצבותם במצב החדש. גם שחרור המפסק גורם לתופעת הריטוט הנ"ל. הסיבה לריטוט היא קפיציות של המגעים. ריטוט המגעים נקרא גם בספרות **ניתור מגעים**. ריטוט זה יכול להימשך מספר מילי שניות. השרטוט הבא מתאר את התופעה.



היות וזמן המיקרו-בקר הינו מהיר יותר מזמן ההתייצבות של המפסק, המיקרו-בקר עלול לפרש את הריטוט כמספר לחיצות על אותו מקש.

קיימים מספר פתרונות לבעיית הריטוט. הבעיה יכולה להיפתר בתוכנה או בחומרה. פתרון בעזרת תוכנה לבעיה זו היא להוסיף השהייה לאחר כל הבחנה בשינוי מצב הלחצנים. השהייה זו אמורה להיות יותר גדולה מזמן ריטוט המגעים. בשיטה זו חייבים לדעת מהו זמן ריטוט המפסק אחרת

יכול להיווצר מצב ובו ניתן זמן ארוך מידי ובו לא יובחנו לחיצות מהירות על המקש. בצורה כללית ניתן לומר שאם זמן ניתור המגעים הוא מספר מילי שניות אז השהייה של כמה עשרות מילי שניות פותרת את הבעיה.

פתרון של חומרה הוא הוספת קבלים במעגל בין כל רגל שורה (R_n) לאדמה. כל לחיצה תדרוש פריקה או טעינה לפי טאו ותבטל את קפיצות המתח שבין '0' ל '1' .

תוכנית בשפת אסמבלי

התוכנית עובדת בשיטת ה '0' הרץ. בהתחלה בודקים האם יש מקש כלשהו לחוץ בכל המטריצה. אם כן מזהים את המקש. הבדיקה נעזרת במשתנה בשם col_counter (מונה עמודה) אשר מראה באיזו עמודה יש את ה '0'. התוכנית נותנת למקש 0 את הערך 10 ולמקש * את הערך אסקי '*'. ולמקש # את ערך האסקי '#'.
;

key1: ; בדיקה "בפעם אחת" האם יש מקש כלשהו לחוץ באחת העמודות
; אם כן אז מזהים באיזו עמודה ומיהו המקש

```
clr    c1
clr    c2
clr    c3
mov    a,p1
orl    a,#0f0h          ; לבטל את 4 הביטים הגבוהים
cjne   a,#0ffh,key2
ret
```

key2: ; בדיקת עמודה ראשונה

```
clr    c1
setb   c2
setb   c3
mov    col_counter,#0
mov    a,p1
orl    a,#0f0h
cjne   a,#0ffh,zihui
; בדיקת עמודה שנייה
```

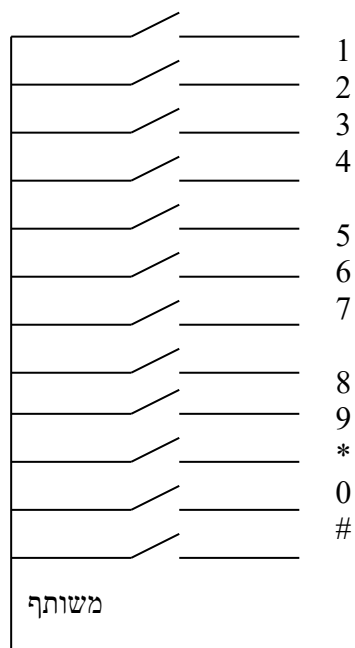
```
setb   c1
clr    c2
setb   c3
mov    col_counter,#1
mov    a,p1
orl    a,#0f0h
cjne   a,#0ffh,zihui
```

; בדיקת עמודה שלישית;

```
setb   c1
setb   c2
clr    c3
mov    col_counter,#2
mov    a,p1
orl    a,#0f0h
cjne   a,#0ffh,zihui
```

```
    mov    makash,#0ffh
    ret
zihui: jb    acc.0,chk_key4
        mov    a,#1
        ljmp   zihui1
chk_key4:
        jb    acc.1,chk_key7
        mov    a,#4
        ljmp   zihui1
chk_key7:
        jb    acc.2,chk_key10
        mov    a,#7
        ljmp   zihui1
chk_key10:
        jb    acc.3,no_key
        mov    a,#10
        sjmp   zihui1
no_key:ret
zihui1:add    a,col_counter
        cjne  a,#10,chk11
        mov    a,#'*'
        sjmp  key5
chk11: cjne  a,#11,chk12
        mov    a,#0
        sjmp  key5
chk12: cjne  a,#12,key5
        mov    a,'#'#
key5:  mov    makash,a
        ret
```

כאמור קיימים סוגים נוספים של לוחות מקשים. סוג אחד הוא של מגע משותף אחד לכל 12 המפסקים הנראה כך :



לחיצה על אחד המקשים מעביר את המתח שברגל המשותפת (או 0 Pull Down – או 1 Pull Up) אל המגע של המפסק הלחוצי. ללוח מקשים כזה לפחות 13 הדקים.

סוג נוסף של לוח מקשים הוא לוח מקשים מסוג מטריצה שורה עמודה עם מגע משותף. לחיצה על אחד המקשים מעבירה את המתח שבמגע המשותף אל השורה והעמודה המתאימה ללחצן. לדוגמא – אם שמים במגע המשותף אדמה אז לחיצה על מקש 5 מעבירה את האדמה אל R2 ואל C2. ללוח מקשים מסוג זה יש לפחות 8 הדקים.