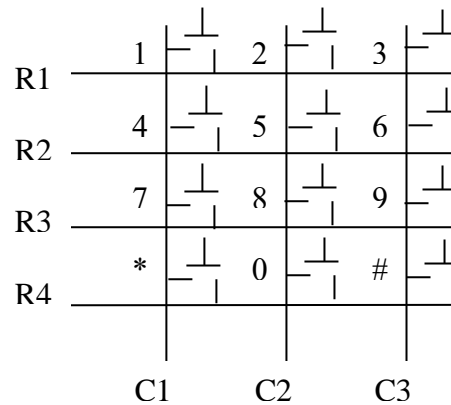
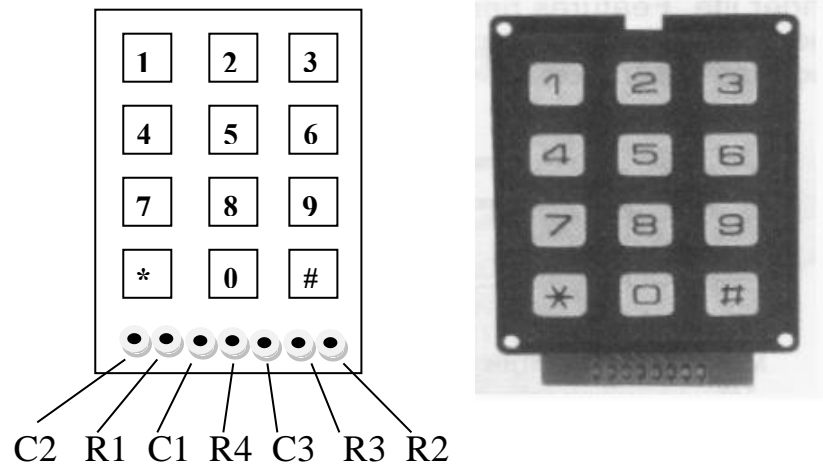


ארדואינו - חיבור לוח מקשים 4*3

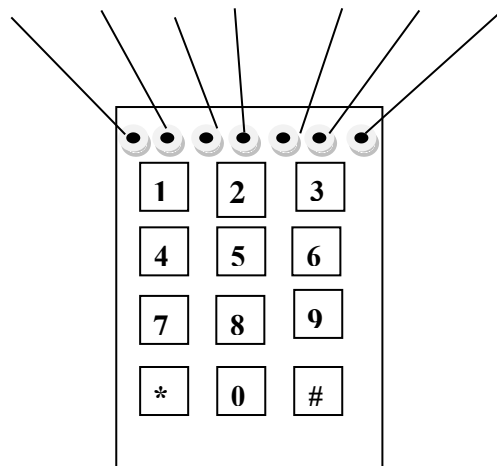
לוח המקשים אותו נחבר מתואר בשרטוט מספר 1



שרטוט 1 : מצד ימין - תמונה של לוח מקשים 4*3 . מצד שמאל - תצורת ההדקים. למטה – מבנה פנימי.

יש לשים לב ללוחות מקשים שבהם המחבר נמצא מעל המקשים וסידור ההדקים שונה !!

הדקי לוח המקשים R1 R2 R3 R4 C1 C2 C3



בסוף ההסברים מתואר גם לוח מקשים של 4*4.

1. מבוא:

לוח מקשים 3X4 זהו התקן קלט, המקשר בין המשתמש למערכת מיקרו. התקן זה כולל 12 לחצנים המסודרים ב-4 שורות ו-3 עמודות. את השורות - ROWS, מסמנים ב R1 עד R4. את העמודות – COLOUMN מסמנים ב C1 עד C3. כל לחיצה על מקש מקצרת בין שורה ועמודה בהתאם.

הלוח כולל ספרות מ-0 עד 9 ובנוסף את המפסקים המסומנים ב * ו - #.

קיימים מספר סוגי לוחות מקשים. אחד הנפוצים הוא הלוח בשרטוט 1 מסוג מטריצה שורות עמודות. בכל מפגש של שורה עם עמודה מחובר לחצן, לחיצה על מקש מסוים מקצרת בין עמודה ושורה מסוימים. היתרון בצורת מטריצה זו היא מס' קווים קטן לעומת לוח מקשים שבו יש 13 קווים. אחד משותף ואחד עבור כל אחד מ 12 המקשים.

ללוח מקשים מסוג מטריצה יש קונקטור (מחבר) בן 7 הדקים הנראה בשרטוט 1 בחלק העליון משמאל. בפרויקטים רבים משמש לוח המקשים בכדי להכניס סיסמא או כדי לנהל דו שיח עם המשתמש. הסבר מפורט על חיבור לוח מקשים אל מיקרו בקר ניתן למצוא באתר כאן בקישור:

http://www.arikporat.com/projects/key_board.doc

2. טבלת הצטלבות השורות והעמודות:

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

טבלה 1 : הצטלבות שורה עמודה בלחיצה על מקש

כדאי לציין שיש לוחות מקשים שבהם יש בקונקטור 13 הדקים. הדק אחד הוא משותף ועוד 12 הדקים עבור כל מקש. לחיצה על אחד המקשים מקצרת בין ההדק המשותף לאחד מהדקי הקונקטור. בגלל כמות ההדקים הגדולה השימוש בלוח מקשים זה הוא דל.

3. עיקרון גילוי מקש לחוץ : שיטת '0' רץ

נעזר בשרטוט מספר 2 להבנת הגילוי של המקש הנלחץ.

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

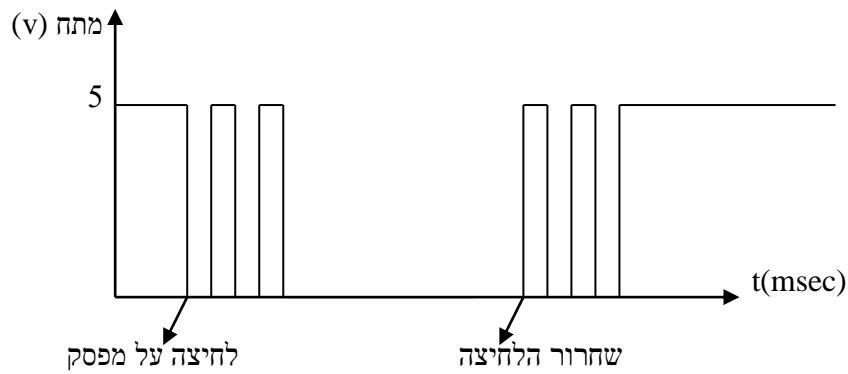
שרטוט מספר 2 : מבנה פנימי של לוח מקשים
נגדיר את העמודות כפלט ואת השורות כקלט.

נשים '0' באחת העמודות ו '1' בשתי העמודות האחרות ($C1=0, C2=C3=1$) ואז בודקים את מצב השורות. אם 4 השורות נמצאות בגבוה '1' ($R1=R2=R3=R4=HIGH$) , אז אף מקש בעמודה זו איננו לחוץ ואז מעבירים את האפס לעמודה הבאה ($C1=C3=1, C2=0$) ושוב מבצעים את אותה הבדיקה על השורות. כך עושים גם לגבי העמודה השלישית ($C1=C2=1, C3=0$). במידה ובאחת השורות מגלים שאין HIGH '1' , כלומר יש LOW '0' , זה אומר שאחד המקשים לחוץ ואז נמצא בעזרת תוכנה באיזו שורה נמצא ה LOW '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 פורט1 ונשים ב 4 הביטים הגבוהים שלו '1'. מכיוון שהמקש שנלחץ הוא לא מעמודה מס' 1 הערך שיקרא הוא $R1=R2=R3=R4=HIGH$ (כלומר אין מקש לחוץ). אנו נמשיך בסריקה וכעת נקבע '0' לוגי בעמודה 2 ובשאר העמודות '1' לוגי. כעת נקבל $R2=0, R1=R3=R4=HIGH$. מכאן שהמקש הנלחץ הוא מעמודה זו. כעת נמצא את הסיבית שערכה '0' מעמודה זו וניתן לדעת שזה 5.

4. בעיית ריטוט (ניתור) המגעים (Bouncing) ופתרונה (De-bouncing)

בכל מפסק קיימת בעיה של ריטוט המופיע בזמן המעבר של המפסק. לחיצה על המפסק גורמת למגעיו לרטוט , לנתר , (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.



שרטוט 3 : ריטוט (ניתור) המגעים

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'.

לוח מקשים נוסף נראה בתמונה 1 שבאיור הבא:

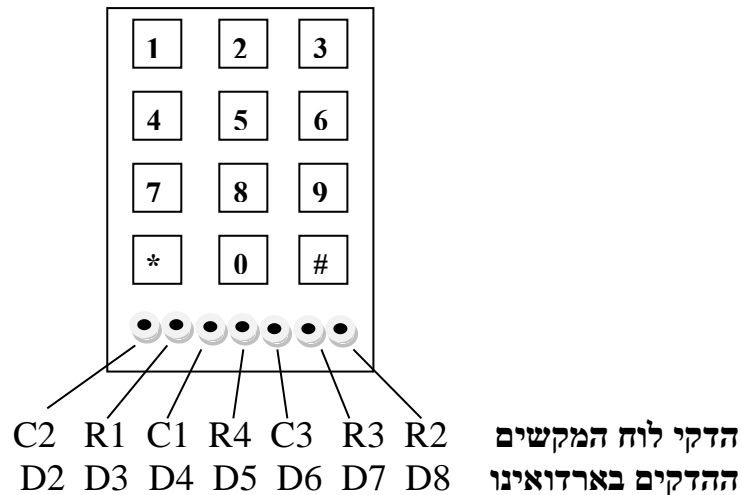


תמונה 1 : לוח מקשים 4*4 .

בלוח מקשים הזה סדר ההדקים הוא : R1 R2 R3 R4 C1 C2 C3 C4

5. דוגמה לחיבור לוח מקשים אל הארדואינו

חבר את לוח המקשים לפי השרטוט הבא :



שרטוט 4 : דוגמת חיבור ההדקים של לוח המקשים אל ארדואינו

6. תכנה

כפי שהסברנו במבוא לארדואינו יש פונקציות רבות לטיפול בהתקני קלט פלט. בין השאר יש לו קובץ כותר עם פונקציות לטיפול בלוח מקשים שנקראת Keypad.h. הקובץ נמצא ב :

<http://playground.arduino.cc/uploads/Code/keypad.zip>

זהו קובץ דחוס ויש לפתוח אותו לספריית libraries של ה arduino. (c:/.../arduino/libraries/)

נסביר מספר מושגים הקשורים לפונקציות אלו :

א. יצירת constructor (אתחול עצמים) למחלקה (class) בשם Keypad.

```
Keypad(makeKeymap(userKeymap), row[], col[], rows, cols)
```

הגדרת קבוע מטיפוס byte בשם rows ואתחולו ל 4 //

const byte cols = 3; // הגדרת קבוע מטיפוס byte בשם cols ואתחולו ל 3

char keys[rows][cols] = { // הגדרת מערך דו ממדי מטיפוס char שבו יש 4 שורות ו 3 עמודות

```
{'1','2','3'},
```

```
{'4','5','6'},
```

```
{'7','8','9'},
```

```
{' ','0','#'}
```

```
};  
// הגדרת מערך מטיפוס byte בשם rowPins שבו 4 שורות לקשר בין 4 השורות של לוח המקשים  
// ולאילו הזקים דיגיטאליים הם מחוברים בכרטיס הארדואינו. כאן שורה ראשונה להדק D3 שורה  
// שנייה להדק D4 וכך הלאה לפי שרטוט 4  
byte rowPins[rows] = {3, 8, 7, 5}; //connect to the row pinouts of the keypad  
// קישור בין הזקי העמודות בלוח המקשים ובין ההזקים הדיגיטאליים של הארדואינו.  
byte colPins[cols] = {4, 2, 6}; //connect to the column pinouts of the keypad  
// חיבור כל ההגדרות לאובייקט.  
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols );
```

ב. פונקציות המטפלות בלוח מקשים

1.ב

void begin(makeKeymap(userKeymap))

מאתחלת את מפת לוח המקשים שקיימת בתכנה שתהיה שווה למפת לוח המקשים של המשתמש .

2.ב

char waitForKey()

הפונקציה ממתינה עד ללחיצה על מקש (חוסמת תכניות אחרות !).

3.ב

char getKey()

הפונקציה מחזירה את ערך המקש שנלחץ (אם נלחץ).

4.ב

KeyState getState()

מחזירה את מצבו של הנוכחי של כל מקש. ארבעת המצבים האפשריים הם : **IDLE** – סרק (לא נלחץ),
PRESSED (נלחץ), **RELEASED** - שוחרר ו **HOLD** – במצב של לחיצה.

5.ב

boolean keyStateChanged()

קיים החל מגרסה 2.0 . נותן אפשרות לדעת מתי המקש שינה את מצבו ממצב אחד לאחר. לדוגמה :
במקום לבדוק ערך מקש ניתן לבדוק מתי מקש נלחץ.

6.ב

setHoldTime(unsigned int time)

הפונקציה קובעת את הזמן במילי שניות שעל המשתמש ללחוץ על המקש עד שמצב **HOLD** מתקבל .

7.ב.

addEventListener(keypadEvent)

מפעילה אירוע אם היה שימוש בלוח המקשים.

8.ב.

setDebounceTime(unsigned int time)

הפונקציה קובעת את הזמן במילי שניות עד שנקבל keypad/keyEvent (מקש נלחץ / אירוע מקש). זהו זמן ההשהיה כדי להתגבר על ניתור המגעים (debounce).

ג. תכנית הקולטת מלוח מקשים ומציגה את ערך המקש במוניטור

```
#include <Keypad.h> // להכליל את קובץ הכותר עבור לוח המקשים

const byte ROWS = 4; // הדרת קבוע מטיפוס ביית עבור השורות ובו הערך 4
const byte COLS = 3; // הגדרת קבוע מטיפוס ביית עבור העמודות שבו הערך 3
// הגדרת מערך דו ממדי 4*3 (4 שורות ו 3 עמודות) מטיפוס תווי שבו ערכי האסקי הנתונים
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
// קישור בין 4 השורות של המקלדת ו ההדקים הדיגיטאליים בארדואינו אליו מחוברים
byte rowPins[ROWS] = {3, 8, 7, 5};
byte colPins[COLS] = {4, 2, 6}; // קישור בין עמודות לוח המקשים להדקים בארדואינו/

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();
  // השורה הברה מדפיסה רק אם יש ערך במשתנה key
  if (key != NO_KEY){
    Serial.println(key); // הדפסת המקש במוניטור ויורדים שורה
  }
}
```

ד. חיבור לוח מקשים נוסף שבו תווים רצויים

נדגים הוספה של לוח מקשים עם אפשרות כתיבה של תווים ולא רק ספרות.

```
// תכנית המפעילה 2 לוחות מקשים שונות
#include <Keypad.h>
char key=1,key1,key2;
//----- הגדרות ללוח מקשים 1 -----
const byte ROWS1 = 4; // שורות 4
const byte COLS1 = 3; // עמודות 3
char keys1[ROWS1][COLS1] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};
// קישור בין השורות והעמודות בלוי המקשים להדקים הדיגיטאליים של הארדואינו
byte rowPins1[ROWS1] = {3, 8, 7, 5};
byte colPins1[COLS1] = {4, 2, 6};
Keypad keypad1 = Keypad( makeKeymap(keys1), rowPins1, colPins1, ROWS1,
COLS1 );

// הגדרות ללוח מקשים 2
const byte ROWS2 = 4; // שורות 4
const byte COLS2 = 3; //3 עמודות
char keys2[ROWS2][COLS2] = {
  {'a','b','c' },
  {'d','e','f'},
  {'g','h','i' },
  {' ','\n','#'} // המקש 0 מוריד שורה. המקש * נותן רווח
};
// קישור בין השורות והעמודות של המקלדת וההדקים הדיגיטאליים בארדואינו אליו מחוברים
// כאן חיברנו רק לוח מקשים אחד שיציג גם ספרות וגם תווים ולכן יש לו אותו קישור הדקים
byte rowPins2[ROWS2] = {3, 8, 7, 5};
byte colPins2[COLS2] = {4, 2, 6};
Keypad keypad2 = Keypad( makeKeymap(keys2), rowPins2, colPins2, ROWS2,
COLS2 );

void setup(){
  Serial.begin(9600);
}

void loop(){
  switch (key) {
    case 1:
      key1 = keypad1.getKey();
      if(key1 != NO_KEY)
      {
        if( key1 =='#')
          key=2;
        else
          Serial.print(key1);
      }
      break;
  }
}
```



```
case 2:
  key2 = keypad2.getKey();
  if(key2 != NO_KEY)
  {
    if( key2 =='#')
      key=1;
    else
      Serial.print(key2);
  }
  break;
}
}
```

תרגול

1. רשום תכנית שתפעיל לוח מקשים $4*4$ שנראה כמו לוח מקשים $4*3$ רק שבעמודה הרביעית יש את התווים A בשורה הראשונה, B בשורה השנייה, C בשורה השלישית ו D בשורה הרביעית. את המקש הלחוץ יש להראות במוניטור.
2. רשום תכנית המדמה כספת. בתחילת התכנית המשתמש יתבקש להקיש קוד לפתיחת הכספת. אם הקוד נכון אז תופיע הודעה: הקוד נכון. הכספת פתוחה. במידה והקוד שגוי יירשם הקוד גוי. נסה שנית. אחרי הקשת קוד שגוי 3 פעמים תופיע ההודעה. טעית 3 פעמים בהקשת הקוד. הכספת נעולה. את הקוד הקש בעזרת לוח המקשים ואת ההודעות הצג בעזרת תכנית המוניטור.
3. רשום תכנית של מחשבון. המשתמש יקיש מספר בעזרת לוח המקשים וייראה את המספר על תכנית המוניטור. המקש * יסמן מכפלה ואילו המקש # ישים שמקש השוויון = .
4. רשום תכנית שתיתן מעבד תמלילים פשוט שבו תמונת 4 תמונות של לוחות מקשים. הצג את התווים במוניטור.
לוח ראשון הוא של הספרות 0 עד 9 (לוח רגיל) + * . התו # מעביר ללוח מקשים שני .
לוח שני הוא התווים a עד i . התו * קיים כרגיל והמקש 0 אומר לרדת שורה. המקש # מעביר ללוח שלישי.
לוח שלישי הוא התווים j עד r . התו * קיים כרגיל והמקש 0 אומר לרדת שורה. המקש # מעביר ללוח הרביעי.
לוח רביעי הוא התווים s עד z . המקשים 8 ו 9 ללא תפקיד. התו * קיים כרגיל והמקש 0 אומר לרדת שורה. המקש # מעביר ללוח הראשון.