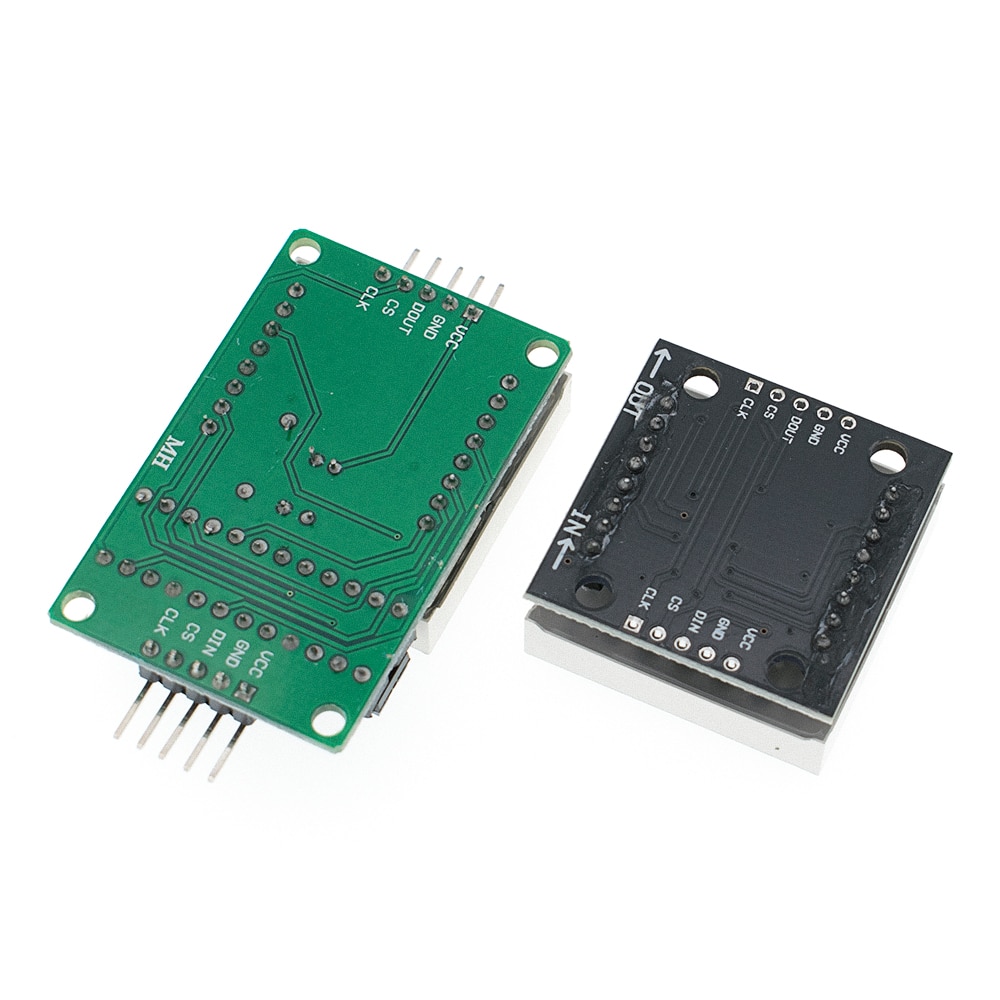
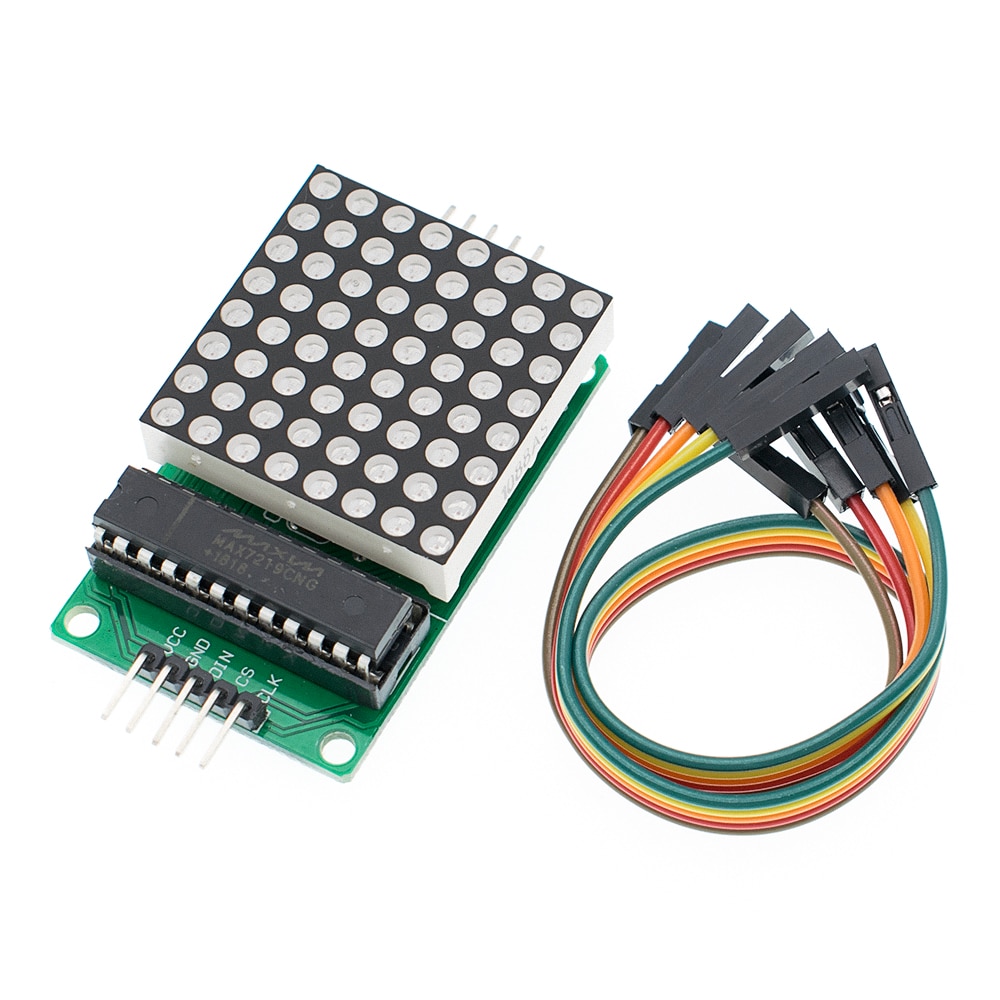
**תצוגת מטריצת נקודות לדים MAX7219**

1. **כללי**

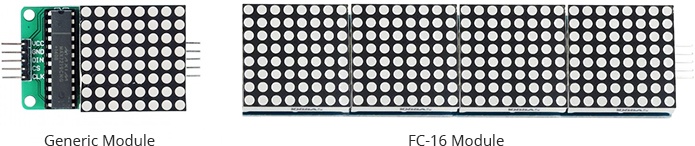
התצוגה שנראית באיור הבא מורכבת מ 2 מרכיבים : **1.** תצוגה של מטריצת נקודות לדים של 8x8 **2.** ג'וק MAX7219 .

איור 1 : תצוגת מטריצת לדים 8X8 . מצד שמאל צד הרכיבים ומצד ימין הצד המודפס.

תצוגת מטריצת נקודות הלדים היא מערך דו-ממדי של 8 על 8 לדים ( סה"כ 64 לדים) והיא משמשת לייצוג תווים, סמלים ותמונות. כמעט כל מסכי ה-LED המודרניים משתמשים במטריצות נקודות .

בתצוגה זו אנחנו שולטים עם רכיב MAX7219 (שהוא מעגל משולב – ג'וק ( Integrated Circuit - לשלוט בכל 64 הלדים.

ישנם מספר לוחות של MAX7219 זמינים, שניים מהפופולריים יותר - אחד הוא המודול באיור למעלה והשני הוא מודול FC- 16 שבו 4 לוחות מחוברים יחד ליחידה אחת כפי שנראה באיור הבא :



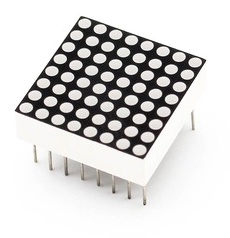
איור 2 : משמאל מודול כללי (גנרי) ומימין 4 מודולים ביחידה אחת .

בכתיבת המאמר נעזרתי באתר :

<https://lastminuteengineers.com/max7219-dot-matrix-arduino-tutorial/>

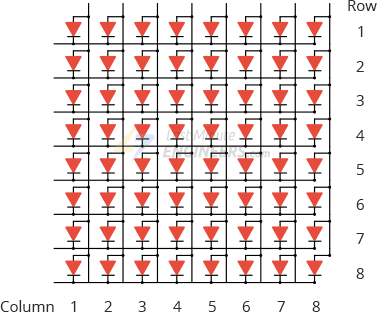
1. **תצוגת מטריצת הנקודות 8x8**

יחידת מטריצה טיפוסית של 8×8 נקודות בצבע יחיד ( יש מודולים בצבע אדום או ירוק או כחול ויש גם מטריצה RGB של שלושת הצבעים) כוללת 16 פינים, 8 לכל שורה ו-8 לכל עמודה. ( כמו שרואים באיור הבא) .



איור 3 : יחידה/מודל מטריצת הנקודות 8x8 .

הסיבה לכך שכל השורות והעמודות מחוברות זו לזו היא כדי להפחית את מספר הפינים הנדרשים. אם זה לא היה המקרה, יחידת מטריצה של 8×8 נקודות הייתה דורשת 65 פינים, אחד לכל נורית LED ואחד עבור אנודה משותפת או קתודה משותפת . על-ידי חיווט שורות ועמודות יחד מספיק רק 16 הדקים . טכניקה זו של שליטה במספר רב של נורות LED עם פחות פינים נקראת ריבוב – Multiplexing . האיור הבא מתאר את המבנה הפנימי של מטריצת הלדים בחיבור קטודה משותפת Common Catode ( הקטודה משותפת לשורה שלמה של 8 לדים).

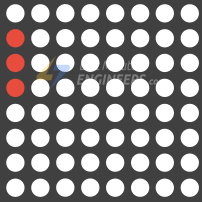


איור 4 : מבנה פנימי של מטריצת הנקודות 8x8 .

לדוגמה : כדי להדליק את הלד בשורה השנייה הרביעית משמאל יש לשלוח לכל השורות '1' חוץ משורה 2 לה נוציא '0' . לכל העמודות נשלח '0' חוץ מהעמודה הרביעית שלה נשלח '1' .

בטכניקה זו, כל עמודה מופעלת למשך פרק זמן קצר מאוד ובמקביל נורות LED בעמודה זו מוארות על ידי התייחסות לשורה המתאימה. העמודות מתחלפות כל כך מהר (מאות או אלפי פעמים בשנייה) עד שההתמדה של העין האנושית תופסת את התצוגה כמוארת במלואה. לכן רק מקסימום של שמונה נורות LED מוארות בכל פעם.

באיור הבא רואים כיצד מדליקים עמודה אחרי עמודה באופן איטי של עמודה בחצי שנייה. אם נדליק כל עמודה בקצב של מאות ואלפי פעם בשנייה נקבל תמונה יציבה ( כמו עיקרון של תמונה בטלוויזיה) .



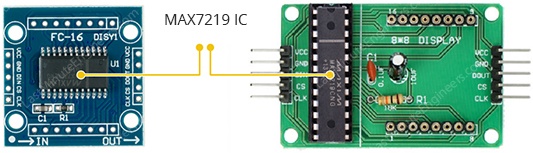
איור 5 : תיאור ההפעלה של עמודה אחרי עמודה במטריצה

1. **הרכיב MAX7219**

**הערה** : על הרכיב עצמו מבחינת האלקטרוניקה כמו מבנה, הסבר הדקים וכו' נסביר בפרק האחרון.

הייתרון בהורדת כמות ההדקים של הרכיב מ 65 ל 16 דורשת ריבוב וריענון של התצוגה כל הזמן כדי לשמור על התמונה יציבה. לשם כך קיים הרכיב MAX7219 שעושה עבורנו את כל עבודת הבקרה והרענון. כל שעלינו לעשות הוא לשלוח לו פקודות טוריות דרך ממשק SPI של 4 פינים והוא ידאג באופן אוטומטי לשאר.

האיור הבא מתאר את הרכיב MAX7219 במבנה DIP (מצד ימין ) עבור מודול יחיד וכיצד הוא נראה בטכנולוגיית SMD בכרטיס FC-16 עם 4 המודולים הבדידים מצד שמאל באיור.



איור 6 : רכיב MAX7219 עבור מודול יחיד (מצד ימין ) ובכרטיס FC-16 עם 4 המודולים הבדידים (מצד שמאל).

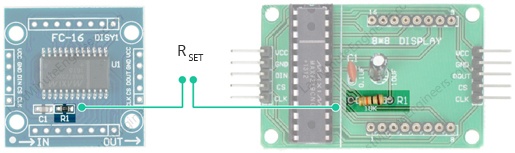
הרכיב יכול לשלוט באופן מלא ב-64 נורות LED בודדות – כולל שמירה על אותה בהירות, ולאפשר להתאים את הבהירות של נוריות ה-LED באמצעות חומרה או תוכנה (או שתיהן). לאחר עדכון התצוגה על ידי המיקרו-בקר, ה-MAX7219 דואג לכל העבודה של ריענון התצוגה ב-800 הרץ ועל ידי כך להסיר את הצורך של ריענון על ידי המיקרו-בקר, שיכול להיות פנוי לעשות דברים חשובים אחרים.

ניתן אפילו לכבות את התצוגה לצורך חיסכון בחשמל, ועדיין לשלוח לרכיב נתונים בזמן שהוא "כבוי". דבר טוב נוסף הוא שכאשר הוא מופעל, הוא שומר על נוריות ה-LED כבויות, כך שאין תצוגות משונות בשניות הראשונות של הפעולה.

שבב MAX7219 מתקשר באמצעות ממשק SPI, כך שהוא דורש רק 3 פיני נתונים כדי להתחבר למיקרו-בקר, ויתרה מכך ניתן לשרשר ( לחבר אחד אחרי השני) מספר מודולים יחד לתצוגה גדולה יותר באמצעות אותם 3 חוטים.

1. **קביעת מקסימום זרם ובהירות**

שבב/רכיב MAX7219 מאפשר להתאים את בהירות התצוגה עם חומרה או תוכנה (או שניהם). כדי לכוונן את הבהירות ברמת החומרה, המודול מגיע עם נגד (RSet) כפי שרואים באיור הבא :



איור 6 : הנגד הקובע את הזרם ובעצם את עוצמת ההארה והבהירות במודול הבודד (מימין) וב FC-16 (משמאל).

נגד זה אחראי על הגדרת הגבול העליון של הזרם המוזן לנורות ה- LED, ומכאן, הבהירות הכוללת של התצוגה.

הטבלה הבאה מציגה את ערכי הנגד שניתן להשתמש בהם בהתאם למתח ולזרם הקדמי עבור מטריצת ה- LED . לדוגמה, אם יש לנו נורית LED של 2V 20 mA, ערך הנגד שלך יהיה 28kΩ (הערכים הם ב- kΩ).

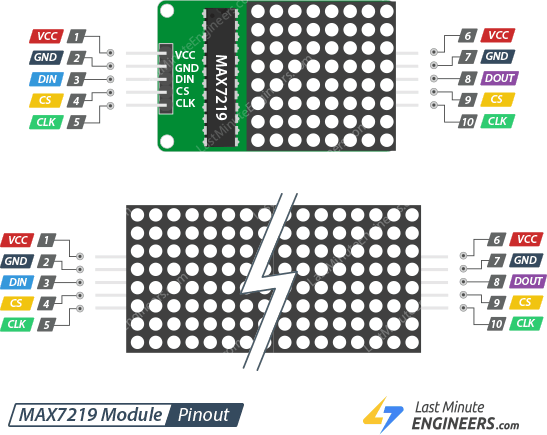
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Rset vs. Segment current and LED forward voltage* | | | | | |
| ISEG (mA)  הזרם דרך הלד | מתח הלדVLED (V) | | | | |
| 1.5 | 2.0 | 2.5 | 3.0 | 3.5 |
| 40 | 12.2 | 11.8 | 11.0 | 10.6 | 9.69 |
| 30 | 17.8 | 17.1 | 15.8 | 15.0 | 14.0 |
| 20 | 29.8 | 28.0 | 25.9 | 24.5 | 22.6 |
| 10 | 66.7 | 63.7 | 59.3 | 55.4 | 51.2 |

טבלה 1 : התאמת בהירות הלד בחומרה בעזרת הנגד Rset .

הסבר על שינוי הבהירות בתוכנה בסעיפים הבאים.

1. **הדקי מודול MAX7219**

לא משנה באיזו גרסה של המודול תבחר, יהיו שני מחברים במודול והם נראים באיור הבא :

איור 7 : סידור ההדקים

הפינים/הדקים בקצה אחד של המודול משמשים לתקשורת עם המיקרו-בקר. בקצה השני המודול מתחבר למודול נוסף דומה.

נתחיל עם הכניסות למודול בצד שמאל של האיור:

VCC  מתחבר ל- 5V. בשל צריכת הזרם הגבוהה של התצוגה (עד 1A בזרם מקסימאלי, מומלץ להפעיל אותה ישירות מספק הכוח החיצוני במקום מספק ה- 5V של הארדואינו. אחרת, יש להקפיד לשמור על הבהירות מתחת ל -50%, כך שמייצב המתח של ה Arduino לא יתחמם יתר על המידה.

GND  מתחבר לאדמה הכללית .

DIN  הוא ה - Data In. נחבר אותו לכל הדק דיגיטלי של המיקרו-בקר.

CS/LOAD הוא Chip Select (לפעמים מסומן כ-LOAD). נחבר אותו לכל הדק דיגיטלי של המיקרו-בקר.

CLK הוא הדק פולסי השעון מהמיקרו אל המודול. יש לחברו אותו לכל הדק דיגיטלי של המיקרו-בקר.

ההדקים בקצה השני של המודול (מצד ימין באיור ) משמשים כאשר רוצים להציג תצוגות שרשרת של כמה מודולים יחד.

VCC מתחבר ל- 5V במודול הבא.

GND  מתחבר לאדמה במודול הבא .

DOUT הוא Data Out ומתחבר להדק DIN של המודול הבא.

CS/LOAD  מתחבר ל CS/ LOASD במודול הבא.

CLK מתחבר ל CLK במודול הבא.

1. **חיבור MAX7219 אל ארדואינו**

בהתחלה נחבר את מתח הספק ל Vcc והאדמה . מכיוון שהתצוגה שואבת הרבה זרם נפעיל את המודול מספק הכוח החיצוני במקום מספק הכוח 5V מלוח Arduino. אם נחבר מודול MAX7219 יחיד ניתן להפעיל את המודול ישירות מהארדואינו אם כי לא מומלץ .

עכשיו נחבר את ההדקים המשמשים לתקשורת SPI. מכיוון שמודול MAX7219 דורש העברת נתונים רבה, הוא ייתן את הביצועים הטובים ביותר כאשר הוא מחובר להדקי SPI של החומרה במיקרו-בקר. הדקי SPI של החומרה הם הרבה יותר מהירים מאשר SPI תוכנה.

יש שים לב שלכל לוח Arduino יש הדקי SPI שונים שיש לחבר בהתאם. עבור לוחות Arduino כגון UNO/Nano V3.0 הדקים אלה הם דיגיטליים 13 (SCK), 12 (MISO), 11 (MOSI) ו 10 (SS) .

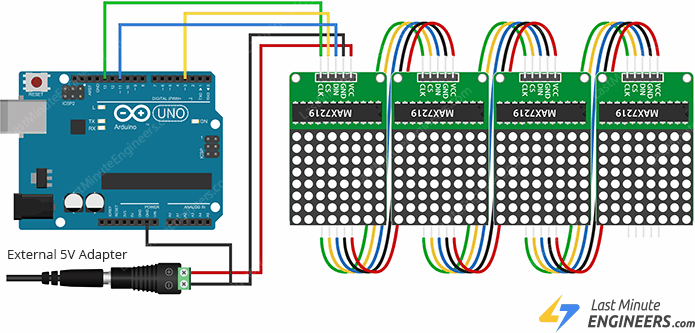
בארדואינו מגה ההדקים הם : 52 (SCK), 50 (MISO), 51 (MOSI) ו (SS) 53 .



איור 8 : הדקי SPI בארדואינו מגה

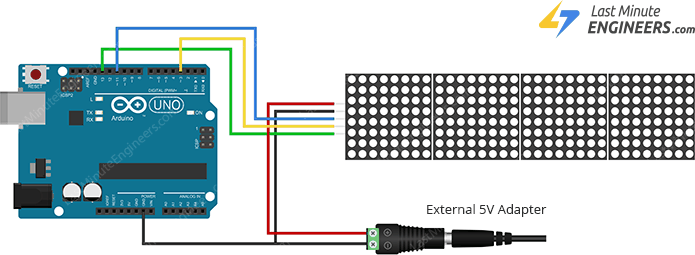
אם משתמשים בלוח Arduino אחר, מומלץ לבדוק את התיעוד הרשמי של כל כרטיס.

האירו הבא מתאר את החיבור של הארדואינו ל 4מודולים בדידים של MAX7219 ומתחתיו את החיבור למודול FC-16 .



מתאם חיצוני ל 5 וולט

למעלה - חיבור ל 4 מודולים בדידים ולמטה חיבור ל FC-16



מתאם חיצוני ל 5 וולט

איור 9 : חיבור הארדואינו ל4 מודולים של MAX7219 (למעלה) וחיבור אל FC-16 המכיל 4 מודולים

אם רוצים לשרשר צגים מרובים כדי ליצור תצוגה גדולה יותר נחבר את ה- DOUT של התצוגה הראשונה ל- DIN של התצוגה הבאה. הדקי ה VCC, GND, CLK ו- CS ישותפו כולם בין צגים.

1. **התקנת הספריות**

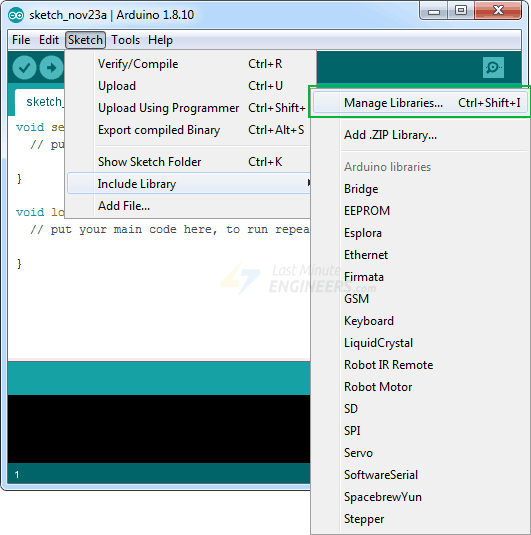
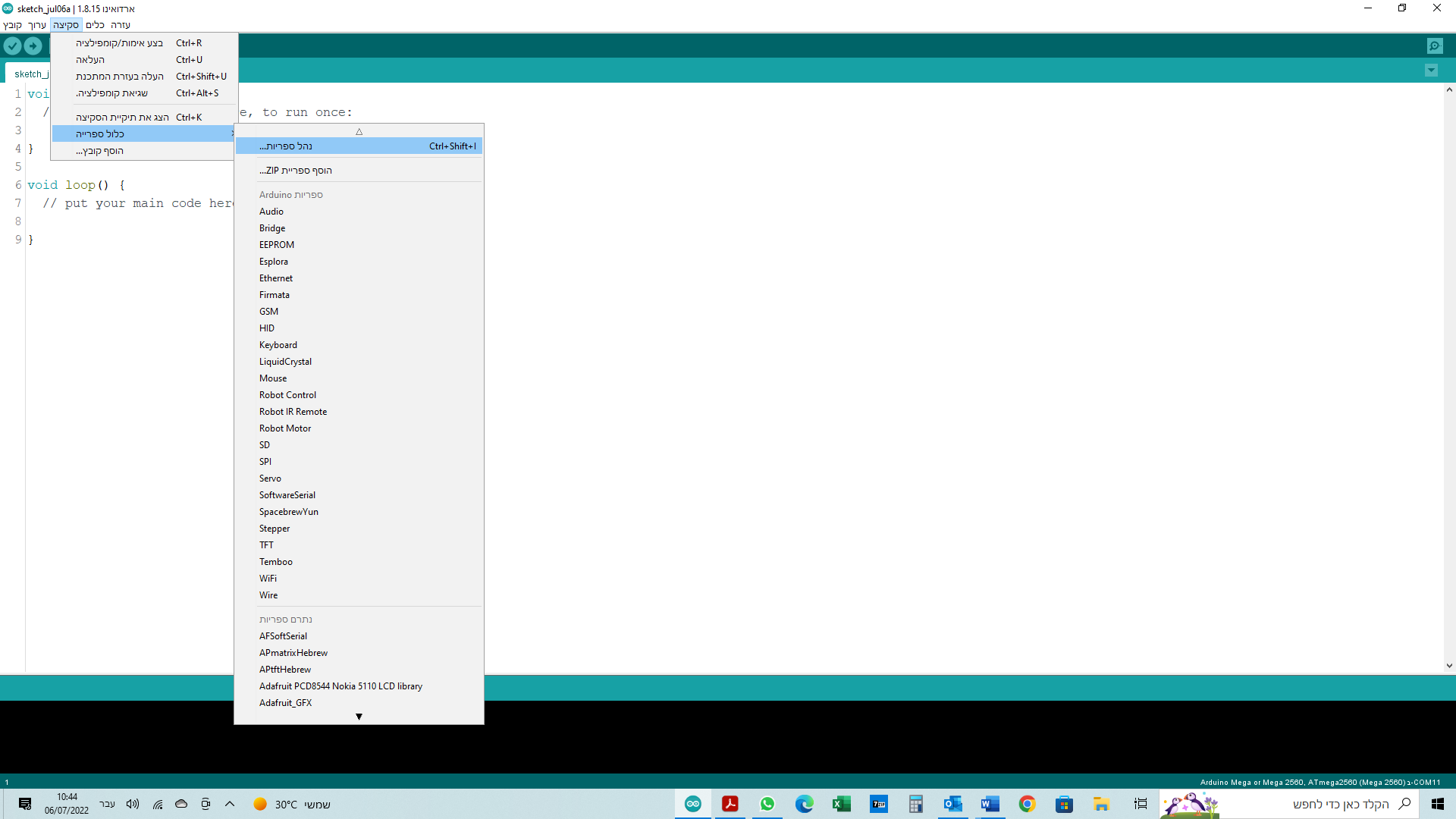
שליטה במודול MAX7219 היא עבודה לא קלה. למרבה המזל, ספריית MD\_Parola נכתבה כדי להסתיר את המורכבות של ה-MAX7219 כך שנוכל להוציא פקודות פשוטות כדי לשלוט בתצוגה.

כדי להתקין את הספרייה נווט אל Sketch > Include Library > Manage Libraries או בארדואינו עם גרסה עברית

סקיצה > כלול ספריה > ניהול ספריות...

המתן עד שמנהל הספרייה יוריד את אינדקס הספריות ויעדכן את רשימת הספריות המותקנות.

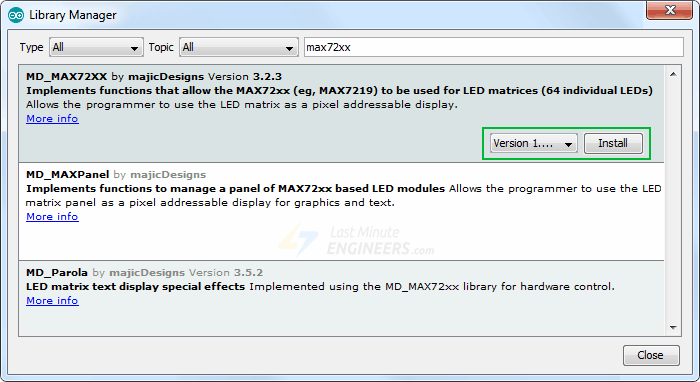
האיורים הבאים מתארים את התקנת הספריה:



איור 10 : ניהול ספריות להתקנת ספריית max72xx

נסנן את החיפוש על-ידי הקלדת 'max72xx'. צריכים להיות כמה ערכים. חפשו MD\_MAX72XX של MajicDesigns. לחץ על ערך זה ולאחר מכן בחר install (ראה באיור הבא מצד שמאל ) . בגרסאות חדשות יותר של ארדואינו נקבל את המסך הבא:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

איור 11 : התקנת הספרייה המתאימה ל MAX7219 בגרסה אנגלית משמאל ובעברית עם 1.8.15 מימין .

ספריית MD\_MAX72XX זו היא ספריה ספציפית לחומרה המטפלת בפונקציות ברמה נמוכה יותר. יש לשייך אותו לספריית MD\_Parola כדי ליצור הנפשות טקסט רבות ושונות כגון גלילה ואפקטים של טקסט ספרייט. התקן גם ספריה זו. בהתקנה בעברית עם גרסת 1.8.15 אחרי לחיצה על התקן מקבלים שאלה האם להתקין ספריות נוספות ויש לרשום אישור.

1. **תוכנה בסיסת להדפסת טקסט**

לתוכנית הראשונה נדפיס טקסט פשוט בתצוגה ללא כל אנימציה. נרשום את התוכנית הבאה שניתן למצוא בין הדוגמאות של הספרייה בתוכנית MD\_MAX72xx\_PrintText לפי האיור הבא :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

איור 12 : טעינת התוכנית MD\_MAX72xx\_PrintText

התוכנית עצמה נראית כך:

// Including the required Arduino libraries

#include <MD\_Parola.h>

#include <MD\_MAX72xx.h>

#include <SPI.h>

// Uncomment according to your hardware type

#define HARDWARE\_TYPE MD\_MAX72XX::FC16\_HW

//#define HARDWARE\_TYPE MD\_MAX72XX::GENERIC\_HW

// Defining size, and output pins

#define MAX\_DEVICES 4

#define CS\_PIN 3

// Create a new instance of the MD\_Parola class with hardware SPI connection

MD\_Parola myDisplay = MD\_Parola(HARDWARE\_TYPE, CS\_PIN, MAX\_DEVICES);

void setup() {

// Intialize the object

myDisplay.begin();

// Set the intensity (brightness) of the display (0-15)

myDisplay.setIntensity(0);

// Clear the display

myDisplay.displayClear();

}

void loop() {

myDisplay.setTextAlignment(PA\_LEFT);

myDisplay.print("Left");

delay(2000);

myDisplay.setTextAlignment(PA\_CENTER);

myDisplay.print("Center");

delay(2000);

myDisplay.setTextAlignment(PA\_RIGHT);

myDisplay.print("Right");

delay(2000);

myDisplay.setTextAlignment(PA\_CENTER);

myDisplay.setInvert(true);

myDisplay.print("Invert");

delay(2000);

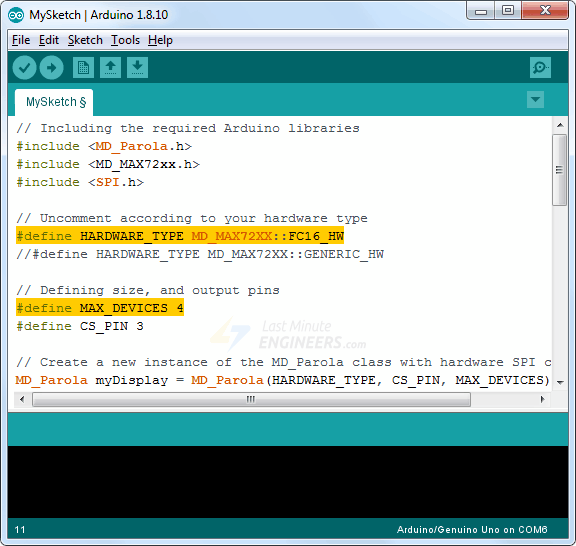
myDisplay.setInvert(false);

myDisplay.print(1234);

delay(2000);

}

נבצע 2 שינויים בתוכנית. יש לשנות את שני המשתנים הבאים.



איור 13 : שינויים של 2 משתנים בתוכנית

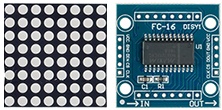
**המשתנה הראשון**: HARDWARE\_TYPE, אומר לארדואינו באיזו גרסה של המודול אתה משתמש.

הגדר את HARDWARE\_TYPE ל GENERIC\_HW אם אתה משתמש במודול שמגיע בדרך כלל עם מעגל מודפס ( PCB ) ירוק וג'וק MAX7219 במבנה DIP ( Dual-in-line package - אריזה דו צדדית בשורה) כפי שרואים באיור הבא.



איור 14 : מודול עם מעגל מודפס בצבע ירוק וג'וק במבנה DIP

נגדיר את HARDWARE\_TYPE ל FC16\_HW אם משתמשים במודול שמגיע בדרך כלל עם PCB כחול וג'וק MAX7219 בטכנולוגיית SMD ( Surface-mount Device - רכיב להרכבה על משטח) שמוצג להלן.

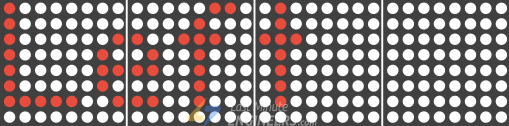


איור 15 : מודול עם מעגל מודפס בצבע כחול וג'וק במבנה SMD

**המשתנה השני**: עם המשתנה השני, MAX\_DEVICES, מגדירים את המספר של תצוגות מטריצת 8×8 נקודות הנמצאות בשימוש. מטריצה של 8×8 נחשבת כהתקן אחד, כך שאם רוצים לשלוט במודול 8×32 יש להגדיר MAX\_DEVICES עד 4 (צג של 8×32 מכיל 4 מעגלים-משולבים מסוג MAX7219).

בסיום השינויים מריצים את הסקיצה. לאחר העלאת הסקיצה נכוון את התצוגה כראוי כדי לראות את הפלט. אם נשתמש במודול גנרי נכוון אותו כך שה- MAX7219 IC יהיה למעלה. אם נשתמש במודול FC-16 נכוון אותו כך שצד ה- DIN יישאר בצד ימין.

אם הכול בסדר יתקבל הפלט הנראה באיור הבא :



איור 16 : הפלט המתקבל

1. **הסבר התוכנית לכתיבת טקסט**

בתחילה יש הנחיות לקומפיילר אילו קבצי כותר להכליל.כפי שצוין קודם ספריית MD\_MAX72XX מיישמת את הפונקציות הספציפיות לחומרה של מטריצת ה- LED בעוד שספריית MD\_Parola מיישמת את אפקט הטקסט. צריך לכלול את ספריית SPI שמגיעה מותקנת מראש ב IDE Arduino. ספרייה זו משמשת לתקשורת בין התצוגה לבין הארדואינו באמצעות SPI.

#include <MD\_Parola.h>

#include <MD\_MAX72xx.h>

#include <SPI.h>

לאחר מכן, עלינו לציין באיזו חומרה אנו משתמשים. מכיוון שאנו משתמשים במודול FC-16 אז HARDWARE\_TYPE מוגדר FC16\_HW. מספר המעגלים המשולבים MAX7219 שאנו משתמשים בו הוא 4, כך MAX\_DEVICES מוגדר ל-4. לבסוף, ההדק שאליה מחובר קו CS של התצוגה מוגדרת כהדק 3 בארדואינו .

#define HARDWARE\_TYPE MD\_MAX72XX::FC16\_HW

#define MAX\_DEVICES 4

#define CS\_PIN 3

לאחר מכן יוצרים מופע חדש של מחלקת MD\_Parola עם הפונקציה ()MD\_Parola. פונקציה זו דורשת שלושה פרמטרים, הראשון הוא סוג החומרה, השני הדק CS, והשלישי המספר המרבי של התקנים מחוברים.

MD\_Parola myDisplay = MD\_Parola(HARDWARE\_TYPE, CS\_PIN, MAX\_DEVICES);

בפונקציית ה setup() אנו מאתחלים תחילה את האובייקט עם הפונקציה ()begin. ניתן להגדיר את הבהירות של התצוגה עם הפונקציה ()setIntensity לה שולחים ערך בין 0 (בהירות מינימלית) ל 15 (בהירות מרבית). הפונקציה ()Clear מנקה את התצוגה.

void setup() {

myDisplay.begin();

myDisplay.setIntensity(0);

myDisplay.displayClear();

}

בפונקציית ה loop() מגדירים תחילה את היישור של הטקסט להדפסה עם הפונקציה ()TextAlignmentset . ניתן ליישר את הטקסט לשמאל, למרכז ולימין עם PA\_LEFT PA\_CENTER PA\_RIGHT בהתאמה.

לאחר מכן, המחרוזת 'שמאל' מודפסת עם myDisplay.print("Left") . יש שים לב למקם מרכאות " " סביב הטקסט מכיוון שאנו מדפיסים מחרוזת טקסט. כאשר ברצונך להדפיס מספרים אין צורך במרכאות כמו לדוגמה myDisplay.print(1234). ניתן גם להפוך את התצוגה באמצעות ערכת הפונקציה setInvert()

void loop()

{

myDisplay.setTextAlignment(PA\_LEFT);

myDisplay.print("Left");

delay(2000);

myDisplay.setTextAlignment(PA\_CENTER);

myDisplay.print("Center");

delay(2000);

myDisplay.setTextAlignment(PA\_RIGHT);

myDisplay.print("Right");

delay(2000);

myDisplay.setTextAlignment(PA\_CENTER);

myDisplay.setInvert(true);

myDisplay.print("Invert");

delay(2000);

myDisplay.setInvert(false);

myDisplay.print(1234);

delay(2000);

}

1. **התוכנית לגלילת טקסט**

כאשר רוצים להדפיס הודעה בתצוגת מטריצת הנקודות לעתים קרובות נגלה שהתצוגה קטנה מכדי להתאים להודעה כולה. הפתרון הוא להשתמש באפקט גלילת טקסט. הדוגמה הבאה מראה כיצד לגלול הודעה.

// Including the required Arduino libraries

#include <MD\_Parola.h>

#include <MD\_MAX72xx.h>

#include <SPI.h>

// Uncomment according to your hardware type

#define HARDWARE\_TYPE MD\_MAX72XX::FC16\_HW

//#define HARDWARE\_TYPE MD\_MAX72XX::GENERIC\_HW

// Defining size, and output pins

#define MAX\_DEVICES 4

#define CS\_PIN 3

// Create a new instance of the MD\_Parola class with hardware SPI connection

MD\_Parola myDisplay = MD\_Parola(HARDWARE\_TYPE, CS\_PIN, MAX\_DEVICES);

void setup() {

// Intialize the object

myDisplay.begin();

// Set the intensity (brightness) of the display (0-15)

myDisplay.setIntensity(0);

// Clear the display

myDisplay.displayClear();

myDisplay.displayScroll("Hello", PA\_CENTER, PA\_SCROLL\_LEFT, 100);

}

void loop() {

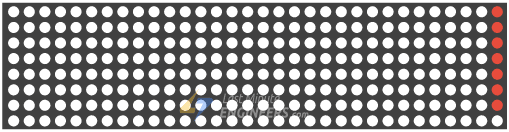
if (myDisplay.displayAnimate()) {

myDisplay.displayReset();

}

}

האיור הבא מתאר את גלילת הטקסט שנקבל כאשר נריץ את התוכנית :



איור 17 : גלילת טקסט

1. **הסבר התוכנית לגלילת טקסט**

החלק הראשון של הקוד בסוף פונקציית ה setup() הוא בדיוק זהה לדוגמה הקודמת. בסוף setup() משתמשים בפונקציה ()displayScroll.

myDisplay.displayScroll("Hello", PA\_CENTER, PA\_SCROLL\_LEFT, 100);

לפונקציה שולחים ארבעה ארגומנטים:

displayScroll(pText, align, textEffect, speed)

* pText – היא מחרוזת הטקסט שרוצים להציג ( בדוגמה כאן זה Hello) .
* align – מגדיר את יישור הטקסט עם אפשרות של השהיה . ניתן להשתמש באותן אפשרויות יישור כמו בדוגמה הקודמת כגון PA\_CENTER, PA\_LEFT או PA\_RIGHT.
* textEffect – מציין את אפקטי הגלילה. הגדרתו PA\_SCROLL\_LEFT תגלול את הטקסט שמאלה.
* Speed קובע את מהירות ההנפשה. המהירות היא הזמן באלפיות השנייה בין מסגרות האנימציה. זמן קצר גורם לאנימציה מהירה יותר.

מלבד זאת בפונקציית ה loop() רק שתי פונקציות משמשות ליצירת גלילת טקסט. ראשית, אנו משתמשים בפונקציה ()displayAnimate במשפט if. פונקציה זו גוללת את הטקסט ומחזירה True עם סיום הגלילה. בסיום הגלילה, אנו מאפסים את התצוגה עם הפונקציה ()displayReset, כך שנקבל גלילה רציפה.

void loop() {

if (myDisplay.displayAnimate()) {

myDisplay.displayReset();

}

}

1. **הסבר הרכיב MAX7219**

התמונות בפרק זה הם מדפי הנתונים של הרכיב בקישור :

<https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>

**12.1 הסבר כללי**

הרכיבים MAX7219 ו MAX7221 של חברת MAXIM מתחברים אל מיקרו-בקרים בעזרת 4 קווים ( מספיק גם 3 קווים) והם ג'וקים של קלט/פלט טוריים עם דוחפי זרם להפעלת תצוגות כמו 7-segment numeric LED (תצוגות 7 מקטעים) עם קטודה משותפת . הרכיב יכול להפעיל עד 8 תצוגות של 7 מקטעים , תצוגות Bar-graph או 64 לדים בדידים.

על השבב כלולים מפענח מ BCD לקוד B (להפעלת תצוגת 7 מקטעים ) , מעגלי סריקה , דוחפי זרם להפעלת מקטעים וספרות וזיכרון RAM סטטי של 8x8 המאחסן כל ספרה. נדרש נגד חיצוני אחד כדי להגדיר את זרם המקטע עבור כל נוריות ה- LED. ה-MAX7221 תואם לסוגי תקשורת - SPI™ , QSPI™ ו MICROWIRE™ ויש לדוחפי הזרם שלו הגבלה של ה slew-rate להקטנת EMI (הפרעות אלקטרו מגנטיות) .

ניתן לטפל ולעדכן ספרות בודדות מבלי לשכתב את התצוגה כולה. MAX7219/MAX7221 גם מאפשרים למשתמש לבחור פענוח B code – ( קוד B ) עבור כל סיפרה (הכוונה האם המספר יפוענח לבינארי מ 0 עד 9 ויכללו גם את הספרות ( E, H, L , P ) או ללא פענוח שבו לכל ביט בנתון יש הפעלה/כיבוי של הסגמנט המתאים ( ביט D7 עבור הנקודה העשרונית) .

ההתקנים כוללים מצב כיבוי בהספק נמוך של 150μA (low-power shutdown mode ), בקרת בהירות אנלוגית ודיגיטלית, אוגר מגבלת סריקה (scan-limit register ) המאפשר למשתמש להציג בין 1 ל-8 ספרות ומצב בדיקה – Test Mode - המדליק את כל נוריות ה-LED. עבור יישומים הדורשים פעולה עם 3V או הבהוב מקטעים קיים ג'וק שנקרא MAX6951 .

**12.2 יישומים**

● תצוגות בר-גרף Bar-Graph Displays ● בקרים תעשייתיים Industrial Controllers ● פאנלים של מודדים Panel Meters ● צגי מטריצת LED LED Matrix Displays .

**12.3 תכונות**

● ממשק טורי של 10 מגה-הרץ

● בקרה של מקטע LED בודדת

● אפשרות פענוח/ללא פענוח של סיפרה/ספרות

● כיבוי בהספק נמוך של 150μA (הנתונים נשמרים)

● בקרת בהירות דיגיטלית ואנלוגית

● התצוגה חשוכה בהפעלת חשמל

● דחיפת זרם לתצוגת LED עם קתודה משותפת .

● אריזה/זיווד של DIP ו- SO של 24 פינים

* 1. **תצורת ההדקים ומעגל יישום אופייני**

האיור הבא מתאר את תצורת ההדקים/פינים של הרכיב (מצד שמאל) ומעגל יישום אופייני (מצד ימין)





איור 18 : תצורת ההדקים (מצד שמאל) ומעגל יישום אופייני (מצד ימין).

**12.3.1 הסבר מעגל היישום ( צד ימין באיור)**

המיקרו בקר או המיקרו פרוססור נראה במלבן השמאלי. המלבן במרכז הוא הג'וק MAX7219 (או MAX7221 ). מעל הרכיב מחוברות 8 תצוגות של 7 מקטעים בחיבור קטודה משותפת.רואים שהמיקרו מחובר לרכיב בעזרת 3 קווים של תקשורת SPI והם DIN שהוא כניסת נתון טורי מהמיקרו ( MASTER) אל הרכיב (SLAVE - נקרא MOSI בתקשורת SPI ) , הקו השני נקרא LOAD/CS והקו השלישי הוא קו CLK של פולסי השעון שמסנכרן את הנתון הטורי לתוך הרכיב.

הנגד של 9.53KΩ המתחבר בין ה 5 וולט ובין הדק 18 של הרכיב – ISET – קובע את הזרם בכל סגמנט ומכאן שאת הבהירות של התצוגה.

מצד ימין של הרכיב רואים 2 יציאות, כל יציאה היא של 8 חוטים. בעזרת 8 היציאות העליונות בוחרים איזו תצוגת 7 מקטעים - מתוך 8 התצוגות - להדליק ובעזרת 8 הקווים התחתונים בוחרים מהי הסיפרה שתהיה בתצוגה .

**12.3.2 הסבר ההדקים**

הטבלה הבאה מתארת את שמות ההדקים ותפקידם.



טבלה 2 : שמות ההדקים ותפקידם.

**הדק 1 – DIN** – Data IN – כניסת נתון הטורי. הנתון הטורי נכנס לרגיסטר הזזה של 16 ביטים בעליית פולס השעון בהדק 13.

**הדקים 2 , 3 , 5 – 8 , 10 , 11 - DIG0-DIG7** - 8 קווים שקובעים איזו תצוגה מופעלת ודרכם נסגר (מצב sink ) זרם הקטודות המשותף לאדמה. במצב שהתצוגה לא מופעלת יש בקו 5 וולט ברכיב 7219 ואילו ב 7221 יש עכבה גבוהה.

**הדקים 4 ו 9** – GND – קווי אדמה. יש לחבר אותן לאדמה של המתח של ה 5 וולט .

**הדק 12** – ב **MAX7219** זהו מצב LOAD – טעינת נתוני כניסה. 16 סיביות הנתונים הטוריים האחרונים ננעלים במעבר מ 0 ל 1 של LOAD.

**ב MAX7211** - זהו CS – Chip Select – בחירת רכיב . נתונים טוריים נטענים באוגר ההזזה כאשר ש- CS נמוך.

16 סיביות הנתונים הטוריים האחרונים לננעלים במעבר מ 0 ל 1 של CS.

**הדק 13 – CLK** – כניסת שעון טורי. קצב מירבי של 10 מגה-הרץ. בקצה העולה – מעבר מ 0 ל 1 - של CLK, הנתונים מועברים לאוגר ההזזה הפנימי. בירידה מ 1 ל 0 של CLK, הנתונים נשמרים מתוך DOUT. ב - MAX7221 , כניסת ה-CLK פעילה רק כאשר ה-CS נמוך.

**הדקים 14 עד 17 , 20 עד 23** – **SEGA – SEGG,DP** – דוחפי הזרם לתצוגת 7 המקטעים והנקודה העשרונית שהם מקור הזרם לתצוגה. ב MAX7219 כאשר מקור הזרם של המקטע כבוי הוא נמצא ב - GND. ב MAX7221 הם בעלי עכבה גבוהה כאשר הם כבויים.

**הדק 18** – **ISET** – ההדק מתחבר ל - VDD באמצעות נגד (RSET) כדי להגדיר את זרם השיא של המקטע.

**הדק 19** – **V+** - מתח אספקה חיובי המתחבר ל +5V.

**הדק 24 – DOUT** - serial Data OUT - הנתונים ב - DIN תקפים ב DOUT 16 . ב 5 מחזורי שעון מאוחר יותר הדק זה משמש לשרשור של מספר MAX7219/MAX7221 והוא אף פעם לא בעכבה גבוהה

1. **דיאגרמה פונקציונאלית**

האיור הבא מתאר את המלבנים המרכיבים את הרכיב



איור 19 : דיאגרמה פונקציונלית של הרכיב

בחלק העליון של הדיאגרמה רואים מצד שמאל את 8 הדקי הסיגמנטים A עד G פלוס DP ( Decimal Point נקודה עשרונית ) . הדקים אלו יציגו את הספרה בתצוגה.

למעלה במרכז יש את ההדקים DIG0-DIG7 הקובעים מי מ 8 תצוגות 7 המקטעים מופעלת. ההפעלה של התצוגה עלי ידי 0 כי החיבור הוא בקטודה משותפת.

המלבן במרכז מצד שמאל הנקרא **CODE B ROM WITH BYPASS** - הוא מלבן האומר האם להמיר את קוד ה BCD לקוד 7 מקטעים או לעקוף אותו.

מתחתיו יש מלבן **8x8 DUAL-PORT SRAM**  - זהו זיכרון המכיל RAM סטטי בן 8 כתובות , בכל כתובת 8 ביטים , המכיל את 8 הספרות שרוצים להוציא לתצוגות.

מימין לשני המלבנים האחרונים יש 5 רגיסטרים :

* **SHUTDOWN REGISTER** – רגיסטר המחשיך את התצוגה (למשל בהפעלת חשמל).
* **MODE REGISTER** – מראה את אופן העבודה עם הרכיב.
* **INTENSITY REGISTER** – רגיסטר בהירות/עצמת הארה הקובע את הבהירות של הסיגמנט (בעזרת תוכנה).
* **SCAN-LIMIT REGISTER** – רגיסטר הגבלת סריקה שבעזרתו קובעים כמה תצוגות לסרוק ( מ 1 עד 8 ).
* **DISPLAY TEST REGISTER** – רגיסטר בדיקת תצוגה. קובע במצב בדיקה להדליק את כל הסיגמנטים בתצוגה.

מתחת ל 5 הרגיסטרים יש את **ADDRESS REGISTER DECODER** – מפענח כתובות הרגיסטר. הוא מקבל 4 ביטים בתקשורת טורית ובעזרת 4 ביטים אלו הוא יודע למי פונים ב SRAM וברגיסטרים.

מצד ימין יש את המלבן **INTENSITY PULSE WIDTH MODULATOR** – אפנן רוחב הדופק של הבהירות. הוא מקבל את מצב רגיסטר INTENSITY ובהתאמה מוציא רוחב פולס מתאים להדליק את הסגמנטים בבהירות הרצויה.

המלבן השני מצד ימין – MULTIPLEX SCAN CIRUITRY – מעגל סריקה בריבוב קובע איזו תצוגה מתוך 8 התצוגות מופעלת.

בצד שמאל יש את **SEGMENT CURRENT REFERENCE** – בעזרת הנגד החיצוני RSET קובע בחומרה את הבהירות של הסגמנטים.

בחלק התחתון מתואר רגיסטר ההזזה אליו שולחים את 16 הנתונים הטוריים.

1. **דיאגרמת זמנים**

האיור הבא מתאר דיאגרמת זמנים של שליחת 16 ביטים אל הרכיב.



איור 20 : דיאגרמת זמנים

מהאיור רואים שבכל עליית שעון מועבר ביט טורי ( מתחילים ב D15 שהוא ה MSB ). בסיום העברת כל 16 הביטים מעלים את קו LOAD/CS ל 1 והנתונים ננעלים ברגיסטר ההזזה .

**14.1 אופני הכתובות הטוריות**

עבור הרכיב MAX7219 נשלחים נתונים טוריים ב- DIN במנות/חבילות של 16 סיביות ומועברים לאוגר ההזזה הפנימי של 16 סיביות עם כל עלייה של CLK ללא קשר למצב העומס. עבור MAX7221 ה CS חייב להיות בנמוך כדי להכניס נתונים פנימה או החוצה. לאחר מכן הנתונים ננעלים באוגרי הספרה או הבקרה בעלייה של LOAD/CS . LOAD/CS חייב לעלות לגבוה במקביל לעליית השעון ה-16 או אחריו אך לפני עליית השעון הבאה אחרת הנתונים יאבדו. הנתונים ב- DIN מועברים דרך אוגר ההזזה ומופיעים ב DOUT 16 חמישה מחזורי שעון מאוחר יותר. הנתונים מועברים החוצה בירידה של CLK . הנתונים מסומנים כ- D0-D15 (בטבלה הבאה הנקראת פורמט הנתונים הטוריים). D8–D11 מכילים את כתובת הרגיסטר D0–D7 מכילים את הנתון ו-D12–D15 הם סיביות ללא משמעות – Don't Care . הביט הראשון שנעביר הוא D15 שהוא ה MSB .



טבלה 3 : פורמט הנתונים הטוריים

**14.2 רגיסטרי הבקרה והספרה**

הטבלה הבאה מפרטת את 14 אוגרי הספרות והבקרה הניתנים להתייחסות. אוגרי הספרות נמצאים ב SRAM בעל יציאה כפולה 8x8. ניתן לכתוב לכל ספרה ישירות . הנתונים ב SRAM נשמרים כל עוד V+ גדול מ 2V. חמישה אוגרי הבקרה מורכבים ממפענח אופן עבודה, עוצמת/בהירות תצוגה, מגבלת סריקה (מספר התצוגות שנסרוק), כיבוי ובדיקת תצוגה (כל נוריות ה-LED דולקות).



טבלה 4 : כתובות אוגרי הבקרה והספרה

* 1. **אופן כיבוי Shutdown Mode -**

כאשר ה-MAX7219 נמצא במצב כיבוי, מתנד הסריקה נעצר, כל מקורות הזרם של המקטעים נמשכים לאדמה ( נמצאים ב 0 ), וכל דוחפי הזרם של התצוגות נמשכים ל V+ ובכך מכבים את התצוגה. ה-MAX7221 זהה, למעט שמקורות הזרם עוברים לעכבה גבוהה. הנתונים באוגרי הספרה והבקרה נשארים ללא שינוי. ניתן להשתמש בכיבוי כדי לחסוך בחשמל או כאזעקה כדי להבהב את התצוגה על-ידי כניסה ויציאה רצופה של מצב כיבוי. עבור תצרוכת זרם מינימלי במצב כיבוי הכניסות הלוגיות צריכות להיות ברמה 0 או V+ (רמות CMOS-logic).

בדרך כלל, נדרשות פחות מ-250μs כדי שה-MAX7219/MAX7221 יעזוב את מצב הכיבוי. ניתן לתכנת את מנהל ההתקן של התצוגה למצב כיבוי, וניתן לעקוף את מצב הכיבוי על-ידי פונקציית בדיקת התצוגה – Display test .

* 1. **מצב הפעלת חשמל**

בעת הפעלת החשמל הראשונית, כל אוגרי הבקרה מתאפסים, התצוגה מתרוקנת/לא פועלת וה-MAX7219/MAX7221 נכנסים למצב כיבוי. יש לתכנת את הרגיסטרים של התצוגה לפני השימוש בתצוגה כי אחרת מוגדר בתחילה לסרוק ספרה אחת, ללא פענוח נתונים באוגרי הנתונים, ואוגר העוצמה יוגדר לערך המינימלי שלו.

* 1. **רגיסטר פענוח מצב**

אוגר מצב הפענוח מגדיר קוד BCD B (0 - 9 ו E, H, L, P) או פעולת אי-פענוח עבור כל ספרה. כל סיבית באוגר מתאימה לספרה אחת. לוגיקה גבוהה בוחרת פענוח קוד B בעוד לוגיקה נמוכה עוקפת את המפענח. דוגמאות לתבנית רישום הבקרה של מצב פענוח מוצגות בטבלה הבאה :



טבלה 5 : דוגמאות לרגיסטר מפענח מצב

כאשר נעשה שימוש במצב פענוח קוד B, המפענח מסתכל רק על הניבל התחתון ( 4 ביטים נמוכים) של הנתונים באוגרי הספרות (D3–D0), תוך התעלמות מהסיביות D4 – D6 . D7 שקובע את הנקודה העשרונית (SEG DP), אינו תלוי במפענח והוא בלוגיקה חיובית (D7 = 1 מפעיל את הנקודה העשרונית). הטבלה הבאה מפרטת את הגופן של קוד B.



טבלה 6 : הגופנים בקוד B .

הדוגמה הנוחה ביותר בטבלה היא הדלקת הספרות 0 ו 1 בתצוגה.

כדי לקבל 0 בתצוגת 7 המקטעים שולחים נתון D0 = D1 = D2 = D3 = 0 (ביטים 4 עד 6 לא רלוונטיים – Don't Care ) והמפענח דואג שהסגמנטים שידלקו יהיו מ A עד F ללא G . כדי לקבל 1 בתצוגה שולחים xxxx0001 והמפענח דואג שסיגמנים B ו C יידלקו.

כאשר לא נבחר פענוח, סיביות הנתונים D7-D0 תואמות לקווי המקטעים של MAX7219/MAX7221. הטבלה הבאה6 מציגה את התיאום אחד לאחד של כל סיבית נתונים לקו המקטע המתאים :



טבלה 7 : המצב ללא פענוח וההתאמה בין ביט הנתון לסגמנט

לדוגמה : כדי להדליק את הסגמנטים A עד D בלבד הנתון שנשלח הוא : 01111000 .

**14.6 בקרת הארה/בהירות והחשכה בין תצוגות**

MAX7219/MAX7221 מאפשר לשלוט בבהירות התצוגה באמצעות נגד חיצוני (RSET) המחובר בין V+ ל-ISET. זרם השיא שנותנים דוחפי הזרם של המקטעים הוא נומינלית פי 100 מהזרם הנכנס ל- ISET. נגד זה יכול להיות קבוע או משתנה כדי לאפשר כוונון בהירות בחומרה על ידי המשתמש. הערך המינימלי של הנגד צריך להיות 9.53kΩ אשר בדרך כלל מגדיר את זרם המקטע על 40mA. ניתן גם לשלוט בבהירות התצוגה באופן דיגיטלי – בתוכנה - באמצעות אוגר העוצמה/בהירות INTENSITY REGISTER .

שליטה דיגיטלית על בהירות התצוגה מסופקת על ידי אפנן רוחב פולס – PWM - פנימי, הנשלט על ידי הניבל התחתון של אוגר העוצמה. המודולטור – אפנן - משנה את קנה המידה של זרם המקטע הממוצע ב 16 צעדים ממקסימום של 31/32 עד 1/32 מזרם השיא שנקבע על ידי RSET (15/16 עד 1/16 ב-MAX7221). הטבלה הבאה מפרטת את תבנית רגיסטר העוצמה. זמן ההחשכה המינימלי בין תצוגה לתצוגה מוגדר ל-1/32 של מחזור.



טבלה 8 : פורמט רגיסטר עצמת בהירות

כתובת הרגיסטר היא 0xxA . רואים שבמצב xxxx0000 הבהירות תהיה מינימלית כי כל סיגמנט מזרים זרם במשך זמן של 1/32 מזמן הפולס של רוחב דופק ה PWMועצמת הבהירות של כל סגמנט תהיה נמוכה. עבור xxxx1111 כל סגמנט מזרים זרם במשך זמן של 31/32 מדופק ה PWM ולכן עצמת הבהירות תהיה מקסימלית.

**14.7 רגיסטר הגבלת סריקה**

אוגר הגבלת הסריקה מגדיר כמה ספרות של 7 מקטעים מוצגות מ-1 עד 8. הם מוצגים באופן מרובב עם קצב סריקת תצוגה טיפוסי של 800Hz עם 8 ספרות מוצגות. אם מוצגות פחות ספרות, קצב הסריקה הוא 8fOSC/N כאשר N הוא מספר הספרות שנסרקו ו fosc הוא קצב הסריקה. מאחר שמספר הספרות הסרוקות משפיע על בהירות התצוגה­ אין להשתמש באוגר הגבלת הסריקה כדי להחשיך חלקים מהתצוגה (כגון דיכוי אפס מוביל - leading zero suppression). הטבלה הבאה מפרטת את תבנית האוגר של הגבלת הסריקה.



טבלה 9 : פורמט הגבלת סריקה

הרגיסטר נמצא בכתובת 0xxB . לדוגמה אם יש לנו 8 תצוגות של 7 מקטעים ורוצים שכולן יעבדו נשלח לרגיסטר את הנתון : xxxxx111 .

אם אוגר הגבלת הסריקה מוגדר לשלוש ספרות או פחות, דוחפי הזרם של תצוגות בודדות יבזבזו כמויות גדולות של הספק. כתוצאה מכך, יש לכוון את הערך של נגד RSET בהתאם למספר הספרות המוצגות כדי להגביל את פיזור ההספק של דוחפי הזרם של ספרות בודדות. הטבלה הבאה מפרטת את מספר התצוגות ואת זרם המקטע המרבי המומלץ המתאים בעת השימוש דוחפי הזרם של התצוגות.



טבלה 10 : זרם סגמנט מרבי עבור 1, 2 או 3 תצוגות

העמודה השמאלית בטבלה היא כמות התצוגות והעמודה הימנית הזרם המקסימלי בסגמנט.

**14.8 רגיסטר בדיקת תצוגה**

אוגר בדיקת התצוגה פועל בשני מצבים: בדיקה רגילה ובדיקת תצוגה. מצב בדיקת תצוגה מפעיל את כל נוריות ה-LED על-ידי עקיפה, אך לא שינוי, של כל הבקרות ורגיסטרי הספרות (כולל אוגר הכיבוי). במצב בדיקת תצוגה, נסרקות 8 ספרות ומחזור העבודה הוא 31/32 (15/16 עבור MAX7221). הטבלה הבאה מפרטת את תבנית אוגר בדיקת התצוגה.



טבלה 11 : תבנית רגיסטר בדיקת תצוגה

הרגיסטר נמצא בכתובת 0xXF . שליחה של 1 לביט D0 של הרגיסטר מדליקה את כל בסגמנטים בתצוגה. במצב רגיל יש לשים בביט 0 ( שה גם המצב אחרי הפעלת חשמל).

**14.9 רגיסטר ללא פעולה No Op Register**

האוגר ללא פעולה ( קיצור של No Operation ) משמש בעת שרשור ( חיבור כמה רכיבים יחד) של MAX7219s או MAX7221s. יש לחבר את כל כניסות ה-LOAD/CS של הרכיבים יחד ואת DOUT ל-DIN ברכיבים סמוכים. DOUT הוא יציאה ברמה לוגית של CMOS שמתחבר בקלות ל DIN של חלקים מחוברים ברצף. לדוגמה, אם ארבעה MAX7219s מחוברים, אז כדי לכתוב לרכיב הרביעי יש לשלוח את המילה הרצויה של 16 סיביות, ואחריה שלושה קודים ללא פעולה (hex 0xX0XX ). כאשר LOAD/CS עולה ל '1' , הנתונים ננעלים בכל הרכיבים. שלושת הרכיבים הראשונים מקבלים פקודות no-op, והרביעי מקבל את הנתונים המיועדים.