

כרטיס קולורדואינו – COLORDUINO

הכרטיס קולורדואינו הנראה באיור 1 מאפשר בקרה על מטריצת לדים מסוג RGB (אדום ירוק כחול). הוא תוכנן על ידי חברת iTead Studios והוא בנוי סביב המיקרו בקר ATmega328. הוא מבוסס על כרטיס קודם שנקרא RAINBOWDUINO של חברת SPEED STUDIO. חברת iTead דאגה שההדקים של הקולורדואינו יהיו תואמים עם הכרטיס הקודם, כך שיתאים לכל הפרויקטים הקודמים, אבל הגדילה את המפרט שלו לקבלת ביצועים טובים יותר. ניתן למצוא כרטיסים גם בצבעים ירוק או אדום או לבן. באיור מספר 1 נראה כרטיס הקולורדואינו משני הצדדים שלו.



איור 1 – כרטיס קולורדואינו משני הצדדים שלו

באיור מספר 2 נראה הכרטיס קולורדואינו שעליו מחוברת מטריצת הלדים

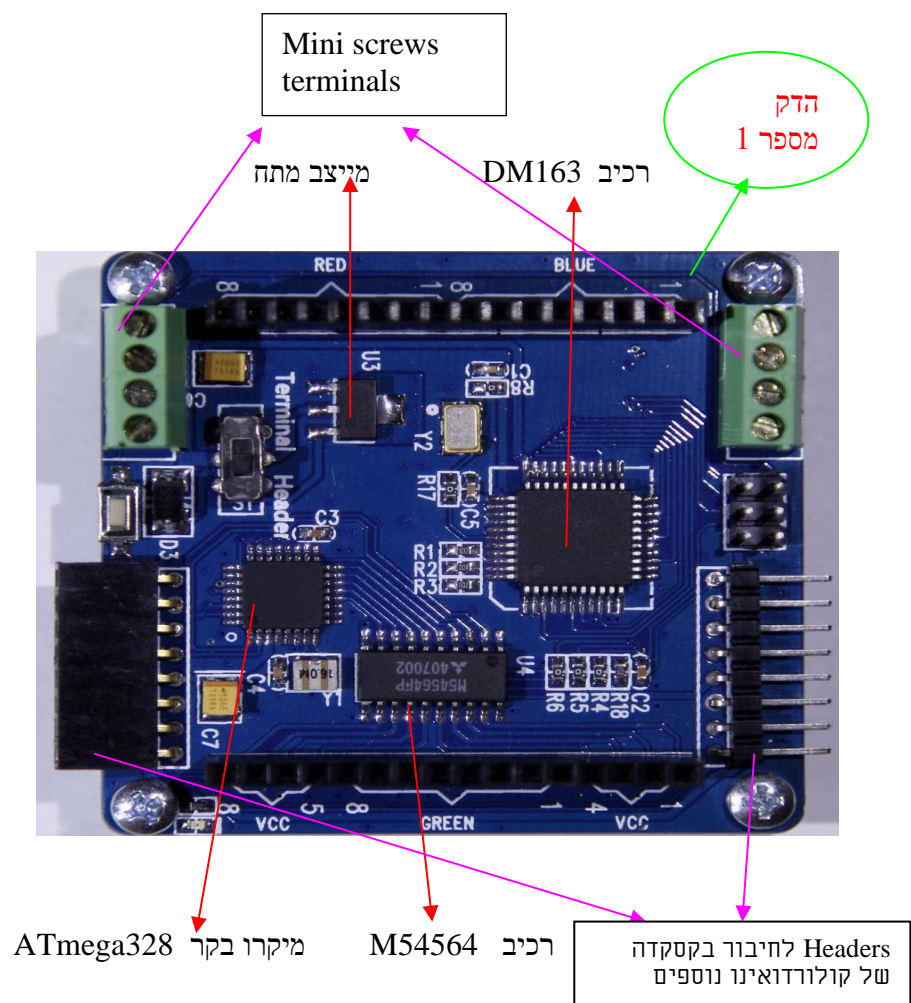


איור 2 – מבט על כרטיס הקולורדואינו עם מטריצת לדים של 8*8 המחוברת אליו.

מאפיינים

- מידות פיזיות : 60mm X 60mm X 1.6 mm
- תמיכה ב 8 ביט של צבע עם 6 ביטים של תיקון לכל צבע בכל נקודה
- תמיכה בחומרה של PWM 16MHz (אפנון רוחב דופק).
- אין צורך בתוספת של מעגלים חיצוניים נוספים.
- ממשק GPIO ו ADC.
- תקשורת UART ו I2C עם אפשרות לחיבור קסקדה של מעגלים נוספים בקלות.
- 24 ערוצים של זרם קבוע של 100 מילי אמפר כל אחד.
- 8 ערוצים של דוחפי זרם חזקים של 500 מילי אמפר כל אחד.

על כרטיס הקולורדואינו מספר מעגלים משולבים (ג'וקים), שאת תפקידם נסביר בהמשך, ומספר קונקטורים. באיור 3 מתוארים חלק מהרכיבים שבכרטיס (הפעם הכרטיס בצבע כחול). כדאי לשים לב לסימון הדק 1 של הצבע הכחול בצד ימין למעלה. זה מציין את הכיוון של חיבור המטריצה על הכרטיס.



איור 3 - כרטיס קולורדואינו עם פירוט הרכיבים

בכרטיס יש רכיב DM163 שהוא LED DRIVER (דוחף זרם ללדים) המאפשר עבודה עם PWM (Pulse Width Modulator - אפנון רוחב דופק) ומאפשר עבודה של כל אחד מערוצי ה RGB עם עומק צבע של 8 ביט ומכאן שבאופן תיאורטי יש אפשרות ליותר מ 16 מיליון צבעים שונים לכל פיקסל לפי החישוב הבא: 8 ביטים לכל צבע נותן 2^8 אפשרויות. היות וניתן לחבר לכל צבע את אחד או שני הצבעים הנוספים נקבל: $2^8 * 2^8 * 2^8 = 2^{24} = 2^4 * 2^{10} * 2^{10} = 16 * K * K = 16M = 16777216$ אפשרויות צבע שונות.

האם העין של האדם יכולה להבחין בין כל הצבעים ?? סביר להניח שלא. אבל ניתן לקבל בקרה טובה על הצבעים של הלדים במטריצה.

ברכיב DM163 יש 3 ערוצים של 6 ביט כל אחד של רגיסטרי תיקון. הם שימושיים לכיול הבהירות המתאימה של כל ערוץ בתוכנה על ידי קביעת הזרם בכל ערוץ ועל ידי כך לאיזון הצבע הלבן עבור תצוגות מטריצה שונות. הסיבה לכך היא שהלד בצבע האדום מאד דומיננטית יחסית לשאר כי נופל עליה מתח נמוך של כ 2.3 וולט בהולכה לעומת כ 3.4 וולט על הלדים בצבע הירוק והכחול (העזר באיור 9) ולכן נוהגים לתת תיקון בתכנה על ידי כך ששולחים לרגיסטרי התיקון ללד האדומה מספר קטן (כחצי) מזה של 2 הלדים בצבע הכחול והירוק. הפונקציה לתיקון מופעלת פעם אחת בתחילת התכנית ואז כאשר נדליק את 3 הצבעים נקבל את הצבע הלבן. אם לא נבצע את התיקון אז כאשר נדליק את 3 הצבעים נקבל במקום צבע לבן צבע הנוטה לאדום.

הממדים של לוח הקולורדואינו עצמו קטנים מעט ממטריצת לדים של $8*8$ (ראה איור 1) וניתן לחבר מספר כרטיסי קולורדואינו יחד בטור אחד אחרי השני ולקבל מבנה גדול יותר של מטריצות. הנתונים מועברים בין הלוחות בפרוטוקול תקשורת I2C או תקשורת טורית רגילה (הדקי TxD ו RxD) שכרטיסי הארדואינו השונים (אוננו, נאנו מגה וכו') תומכים בהן.

מתחי ספק הכוח יכולים להתחבר דרך קונקטורים (Headers) או דרך 2 סטים של הדקים הנסגרים על ידי מברג mini screw terminals- (ראה איור 1). מפסק קובע עם מי מהם נעבוד. לשני קצוות הכרטיס העבירו גם הדקים של TX, RX, DTS, שבעזרתם ניתן לצרוב את רכיב המיקרו בקר ATmega328 שבכרטיס וגם הדקים של SDA ו SCL שהם הדקי התקשורת I2C. לד קטנה בחזית הכרטיס מציינת שיש מתח ספק כוח, אם כי כאשר נרכיב את המטריצה של הלדים על כרטיס הקולורדואינו אז מטריצת הלדים תסתיר את הלד.

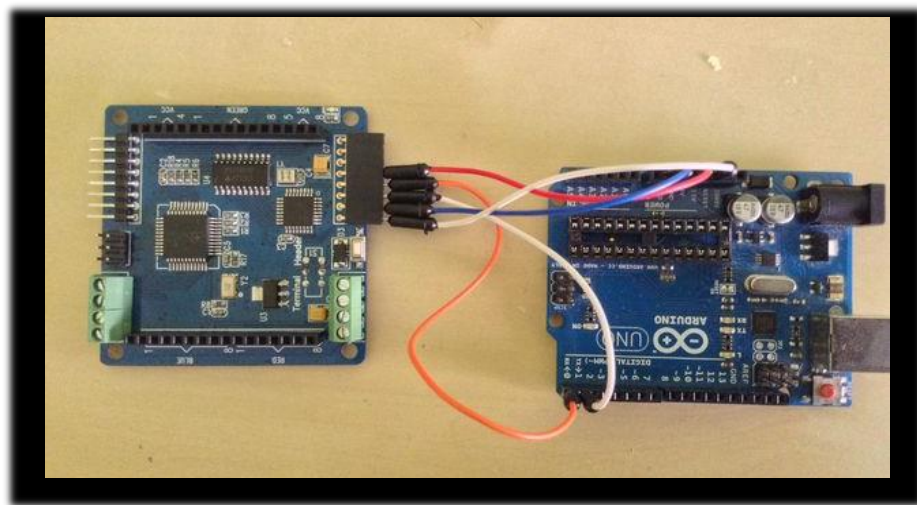
ה setup מאד פשוט. כאשר קונים כרטיס קולורדואינו חדש אז בדרך כלל אחרי שמרכיבים את מטריצת הלדים על כרטיס הקולורדואינו ונותנים 5 וולט לכרטיס, רצה תכנית demo הצרובה במיקרו בקר ATmega328. התכנית מריצה צורות של תבניות יפות במטריצת הלדים. בתכנת הארדואינו, בתפריט "קובץ File" ו "דוגמאות Examples", קיימת ספרייה של קולורדואינו עם תבניות שונות עבור המשתמש.

תכנות המיקרו בקר ATmega328 שבכרטיס הקולורדואינו

בכרטיס הקולורדואינו אין פורט USB , כך שלא ניתן להתחבר ישירות למחשב PC כמו בארדואינו "רגיל". כאשר רוצים לצרוב תכנית למיקרו בקר שבכרטיס ניתן להשתמש בכרטיס ארדואינו אונו.

מוציאים את המיקרו בקר שבכרטיס הארדואינו (ראה האיור 4 שבעמוד הבא) ומחברים בין הארדואינו לקולורדואינו בצורה הבאה (ניתן לעשות זאת כשמטריצת הלדים על הקולורדואינו ואין חובה לנתק את המטריצה מהכרטיס):

- RX בכרטיס ארדואינו ל RX בכרטיס קולורדואינו
- TX בכרטיס ארדואינו ל TX בכרטיס קולורדואינו
- ארדואינו RESET אל DTR בכרטיס קולורדואינו



- מתחי הספק 5 וולט והאדמה מהארדואינו בהתאמה אל הקולורדואינו.

איור 4 – תכנות המיקרו בקר שבקולורדואינו בעזרת ארדואינו אונו

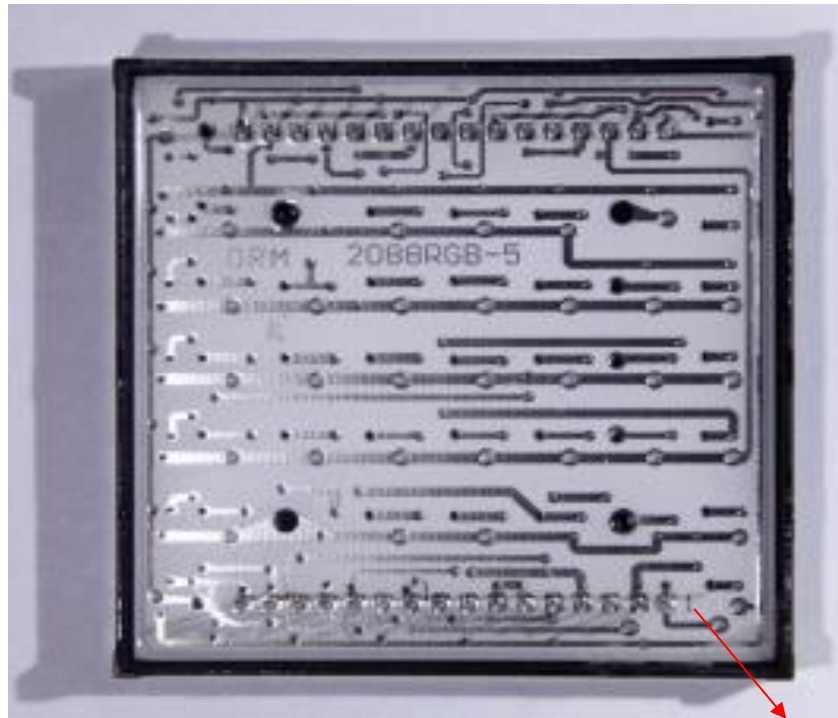
הדבר הופך את כרטיס הארדואינו אונו לצורב . ניתן לעלות את התכנית לקולורדואינו תוך שימוש רגיל בתכנת הארדואינו. יש לשנות בתוכנת הארדואינו בכלים (tools) את הלוח מארדואינו אונו ל Arduino Duemilanove. התוכנה תצרוב את התכנית למיקרו בקר ATmega328 שבכרטיס הקולורדואינו ולא לזה שבכרטיס הארדואינו.

חיבור כרטיס הקולורדואינו ומטריצת הלדים

אם קונים בנפרד את כרטיס הקולורדואינו ובנפרד את המטריצה 8*8 RGB המתאימה יש להרכיב את המטריצה על כרטיס הקולורדואינו בצורה הבאה :

בחלק האחורי של מטריצת הלדים יש סימן של '1' בצד ימין למטה באיור 5. יש לחבר את ההדק הזה אל כרטיס הקולורדואינו להדק 1 של הכחול (BLUE) – ראה איור 3 .

מטריצת הלדים חייבת להיות מסוג אנודה משותפת . כל שורה היא אנודה של 24 לדים
המסודרות ב 8 עמודות כשבכל עמודה 3 לדים בצבעי RGB (ראה איור 8).



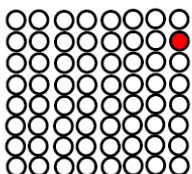
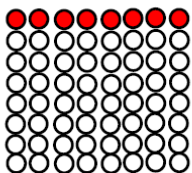
הסימון של הדק 1

איור 5 – החלק האחורי בתצוגת מטריצת הלדים עם הסימון '1'

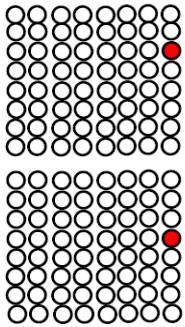
בהמשך נסביר בהרחבה את הג'וק DM163 שעושה את רוב העבודה של הדלקת הלדים במטריצה ועל ידי כך משחרר את המיקרו בקר לאפשרויות עבודה אחרות.

עיקרון הצגת נתון בתצוגת מטריצת הלדים:

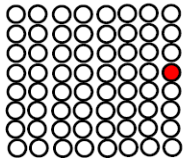
נניח שרוצים להציג את התו ה . בשורה הראשונה נרשום (FFH) 11111111



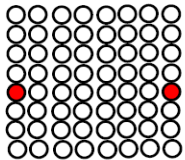
שורה שנייה : 00000001



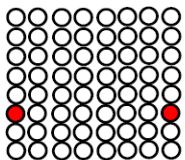
00000001 : בשורה השלישית



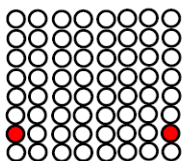
00000001 : בשורה הרביעית



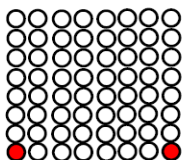
10000001 : שורה חמישית



10000001 : שורה שישית

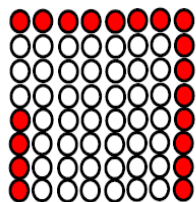


10000001 : שורה שביעית



10000001 : שורה שמינית

אם נעבור על השורות אחת אחרי השנייה במהירות רבה (למעלה מ 16 פעמים בשנייה) אז העין והמוח "יזכרו" רק את הנקודות בצבע האדום ונקבל את התו הרצוי ה. התופעה נקראת Persistence of Vision - ההתמדה בראייה בדומה להצגת תמונה בטלויזיה).



סכמה מלבנית של כרטיס קולורדואינו + מטריצת הלדים 8X8

סכמת המלבנים המרכיבה את כרטיס הקולורדואינו + מטריצת הלדים 8X8 נראית באיור 6.

כרטיס הקולורדואינו מורכב מ 3 יחידות עיקריות :

א. מיקרו בקר **ATmega238**. בעזרת התכנית שצרבנו לו הוא אחראי על כל המתרחש בכרטיס ובתצוגה.

הוא מתחבר אל דוחף הזרם DM54564 בעזרת 8 קווים ואל בקר התצוגה DM163 עם 5 קווים.

ב. **DM163** שהוא דרייבר ל 3 ערוצים בני 8 קווים כל ערוץ (8 ללדים האדומים, 8 ללדים הירוקים ו 8

ללדים הכחולים). הוא מקבל 192 ביטים מהמיקרו בקר ובהתאמה מדליק את הצבעים בלדים. 192

הביטים הם 24 בתים – bytes, המחולקים כך שכל 3 בתים שייכים לעמודה אחת שבה יש 3 צבעים

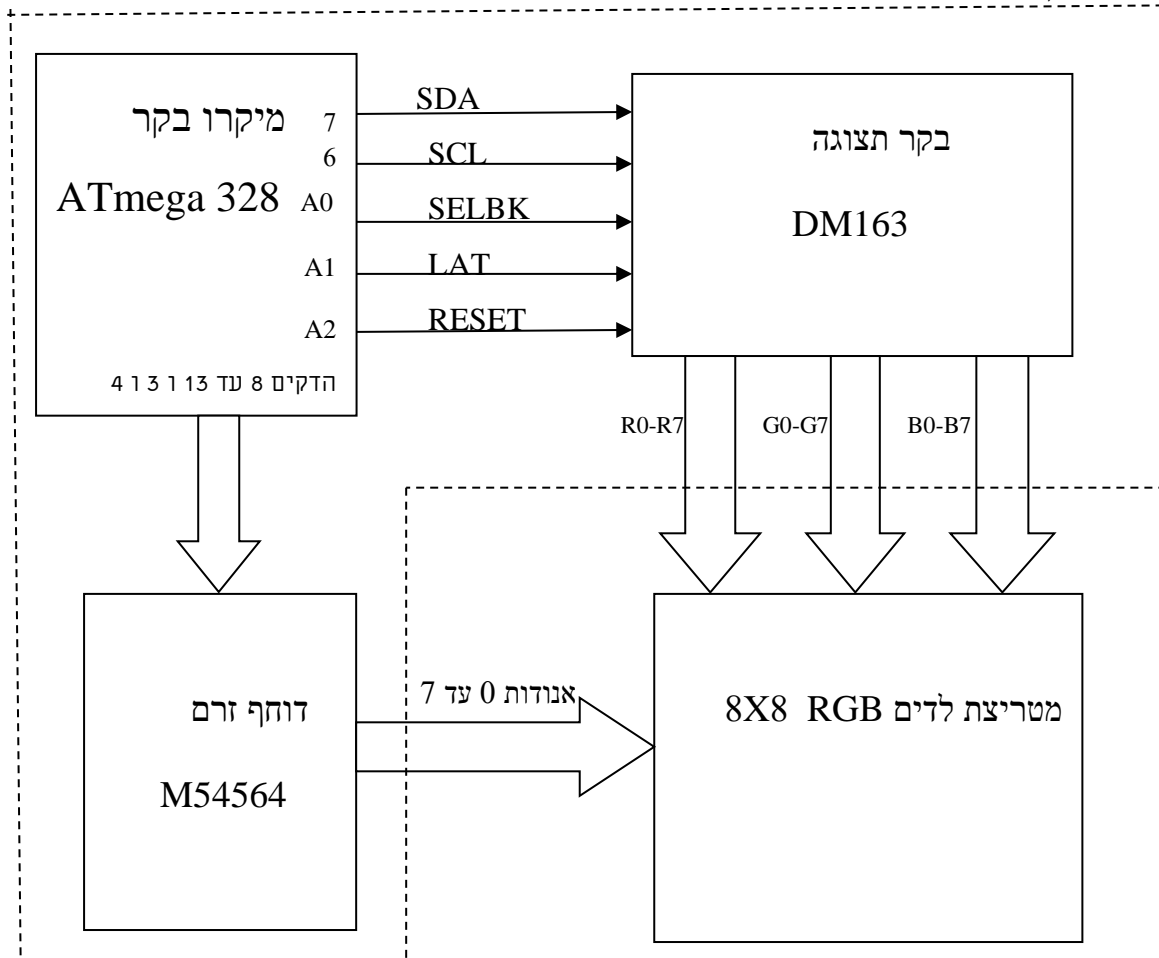
RGB (ביית עבור כל צבע).

ג. **M4564** - דוחף זרם של 8 ערוצים הנותן את המתח החיובי לאנודות של מטריצת הלדים.

מטריצת הלדים מסודרת בצורה של שורות ועמודות. ישנן 8 שורות ו 8 עמודות כאשר בכל עמודה 3 יציאות

(3 צבעים בכל עמודה). ישנן 8 אנודות, כאשר לכל אנודה יש 3*8 יציאות (ראה איור מספר 8).

כרטיס קולורדואינו



איור 6 – סכמה מלבנית של קולורדואינו + תצוגת מטריצת לדים 8X8 RGB

שרטוט חשמלי של הקולורדואינו עם תצוגת המטריצות

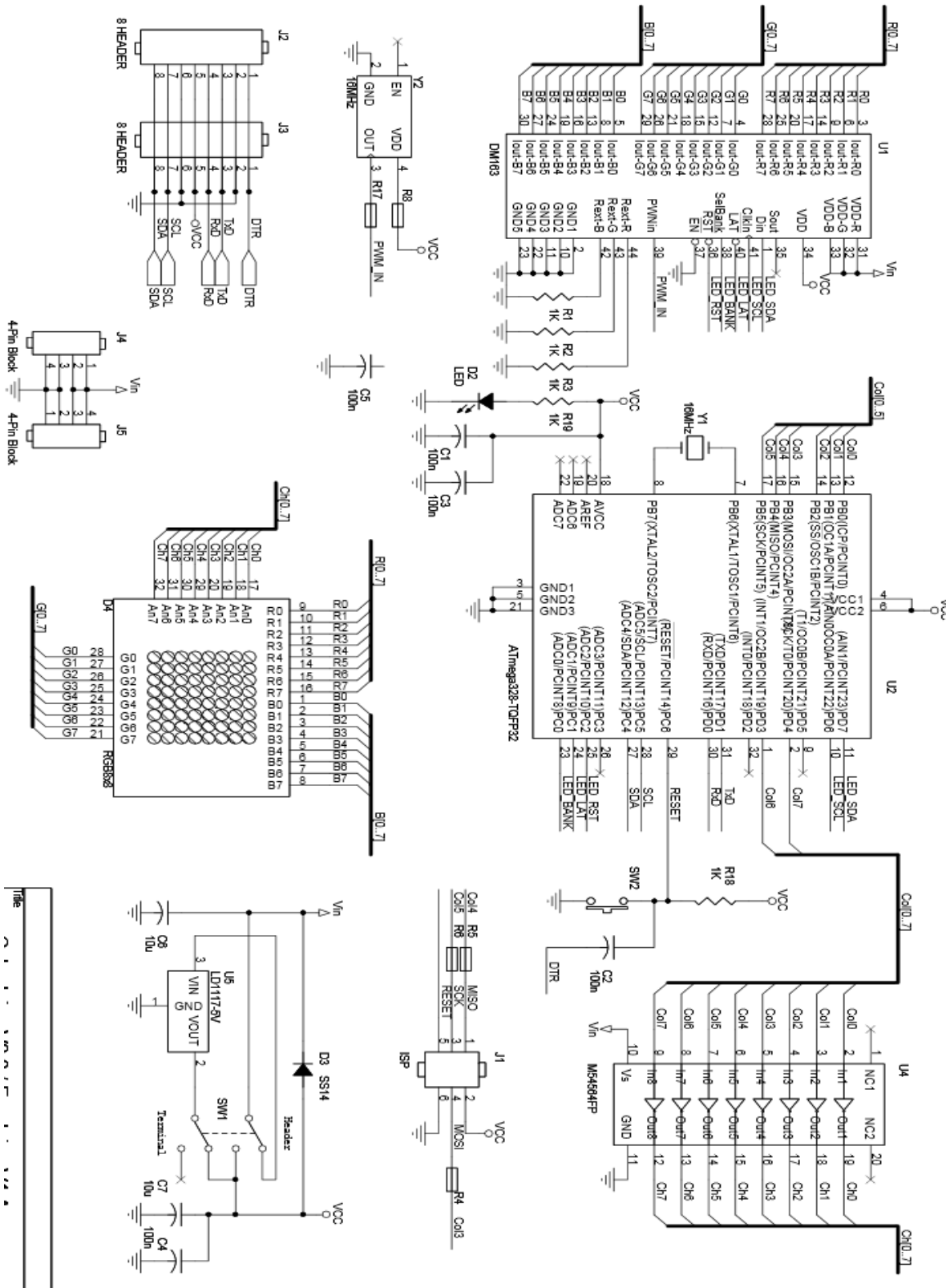
איור 7 -

סכימה

דוחף זרם
לקתודות
DM163
3*8 ערוצים

מיקרו בקר
Atmega328

דוחף זרם
לאנודות
M54564



חשמלית של כרטיס קולורדואינו עם מטריצת לדים 8X8 RGB הסבר כללי : (הסבר מפורט על כל רכיב בהמשך).

המיקרו בקר ATmega328 מתחבר בתקשורת טורית בעזרת 5 הדקים אל רכיב DM163 שהוא דוחף זרם ל 24 הקתודות של התצוגה (3*8). כמו כן הוא מתחבר בעזרת 8 הדקים אל רכיב M54564 שהוא דוחף הזרם ל 8 האנודות של התצוגה.

המיקרו שולח נתונים אל ה DM163 בעזרת ההדקים :

LED_SDA - הדק הנתון הטורי שמוציא המיקרו בקר אל ה DM163 (דוחף הזרם לקתודות). ההדק

הוא PD7 ברכיב המיקרו בקר של ה AT328 . הוא מתחבר להדק מספר 1 ברכיב ה DM163 .

LED_SCL - הדק השעון הטורי המסנכרן את הנתון הטורי לתוך אוגרי ההזזה שברכיב ה DM163 .

ההדק במיקרו בקר הוא PD6 והוא מתחבר להדק 41 של ה DM163.

LED_LAT - פקודת נעילת הנתון הטורי בנועלים שברכיב ה DM163 . זהו הדק PC1 (A1)

במיקרו בקר ומתחבר אל הדק 40 ב DM163 .

LED_BANK – הדק שבעזרתו המיקרו קובע האם לפנות לבנק 0 או בנק 1 ברכיב ה DM163. זהו

הדק PC0 (A0) במיקרו בקר והוא מתחבר להדק 38 ברכיב.

LED_RST - פקודת איפוס (אתחול) . בתחילת העבודה יש להוציא פקודת איפוס בהדק זה אל

הרכיב כדי להביא את כל האוגרים שבו למצב התחלתי ידוע. זהו הדק PC2 (A2) במיקרו בקר והוא

מתחבר להדק 36 ברכיב.

המיקרו שולח 8 קווים אל רכיב M54564 שהוא דוחף הזרם ל 8 האנודות. 8 הכניסות מתחברות , כל

אחת, למערכת עם 4 טרנזיסטורים (איור 11) ו 8 היציאות של הרכיב מתחברות אל 8 האנודות של

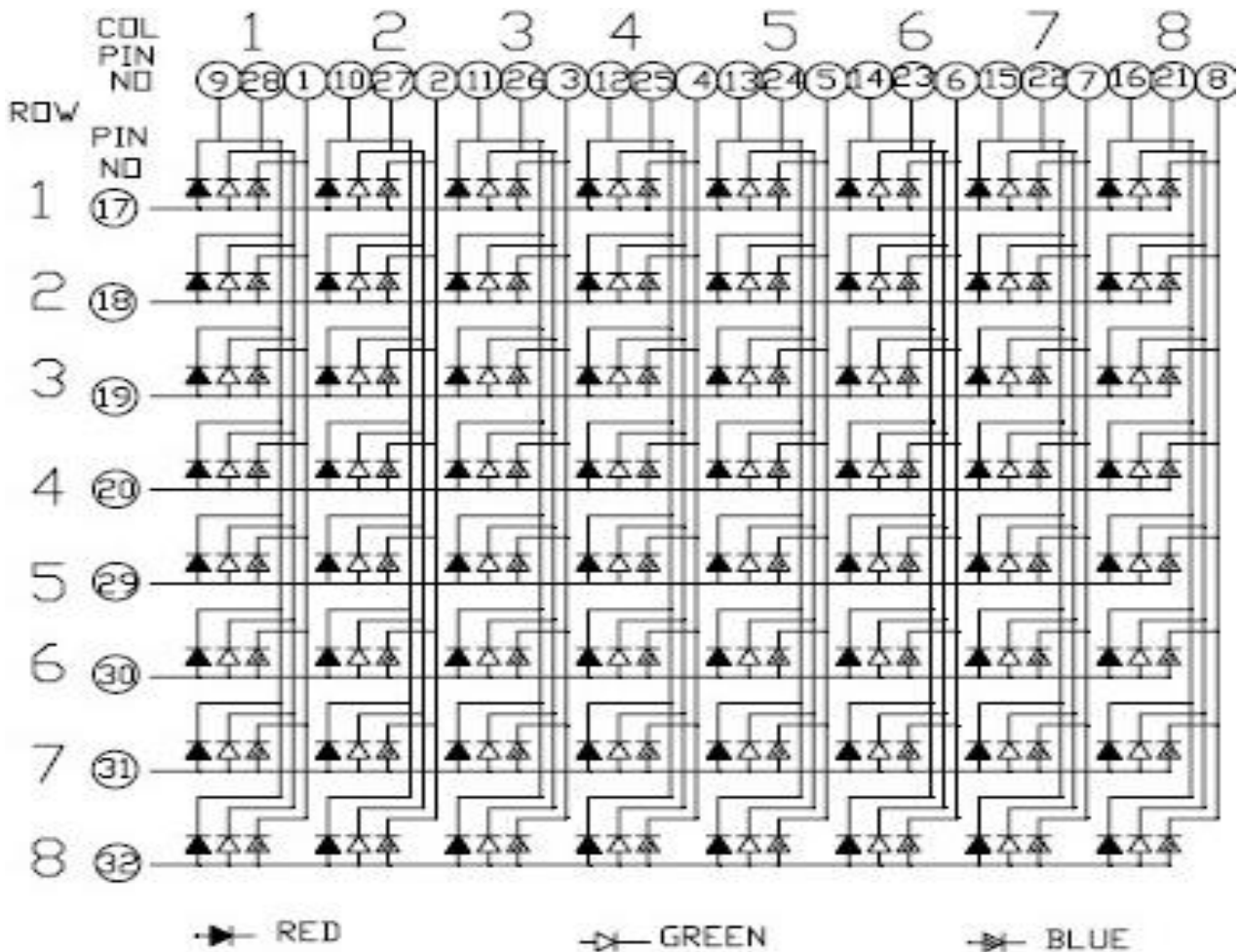
תצוגת המטריצה. המיקרו בקר מתחבר לדוחף הזרם עם ההדקים PB0~PB5 (הדקים דיגיטאליים 8 עד

13 - להפעלת האנודות מ 0 עד 5 בהתאמה) והקווים PD3 ו PD4 (הדקים דיגיטאליים 3 ו 4

להפעלת האנודות מ 6 ו 7 בהתאמה).

מטריצת התצוגה - 8X8 RGB led matrix

באיור 8 מתואר המבנה פנימי של התצוגה.



איור 8 – מבנה פנימי של המטריצה

התצוגה מחולקת לשורות ועמודות. השורות הן האנודות והעמודות הן הקתודות. בכל שורה (לכל אנודה משותפת) מתחברות 24 לדים. 8 לדים אדומים, 8 ירוקים ו 8 כחולים. לדוגמה: הדק מספר 17 הוא אנודה מספר 1. לאנודה זו מתחברות בעמודה 1 לד אדומה, ירוקה וכחולה בהדקים 9, 28 ו 1 בהתאמה.

לכל עמודה של צבע מתחברות גם 8 לדים נוספות. לדוגמה – לעמודה 1 – הדק 9 מתחברות כל הלדים האדומות המתחברות גם לאנודות האחרות.

דוגמה : אם נרצה להדליק את הLED הירוקה שמחוברת לאנודה הרביעית בעמודה הרביעית יש להוציא לאנודה בהדק 20 מתח חיובי ולקתודה בהדק 25 מתח '0'. כלומר יש להוציא לאנודות את הנתון הבינארי: 00001000 ולקתודות את הנתון הבינארי בן 24 הביטים :

111 111 111 111 101 111 111 111

באיור 9 מצורפים דפי הנתונים של הLEDים בתצוגת המטריצה:

Electro-Optical characteristic(Ta=+25°C)光电特性 (温度=+25°C)						
PRODUCTMODEL 产品型号	Emitted color 发光颜色	Forward voltage 正向电压 (VF) (IF=20MA)/VF		Luminous intensity 亮度 (mcd) (IF=20MA) VF		Dominant Wavelength 主波长 (nm) (IF=20MA)/λ d
		MIN	TYP	MIN	TYP	
GTM2088ARGB-23	RED	1.9	2.3	100	140	622~625
	GREEN	3.0	3.4	220	240	522.5~523.75
	BLUE	3.0	3.4	70	80	466.25~467.5
ABSOLUTE MAXIMUM RATINGS (Ta=25°C)最大额定值 (温度=25°C)						
Reverse voltage (V) 反向电压	Continuous Forward Current (mA) 瞬间正向电流	Peak forward Current (mA) 峰值正向电流	Power dissipation (mw) 消耗功率	Operation Temperature range (°C) 工作温度	Storage temperature Range (°C) 储藏温度	
8	50	70	150	-40~80	-40~80	

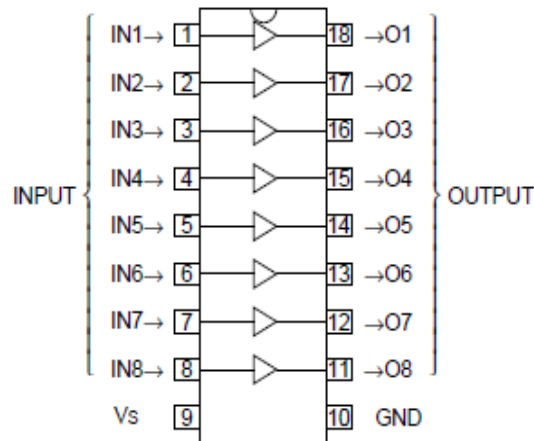
איור 9 – דפי נתונים של הLEDים בתצוגת המטריצה.

החלק העליון בטבלה הוא ערכים אופייניים ואילו החלק התחתון בטבלה הוא הערכים הנקובים המקסימאליים. Absolute Maximum Ratings. כל הערכים נמדדו בטמפרטורה של 25 מעלות צלסיוס.

עבור כל נתון יש ערך מינימאלי (- MINimum -) וערך אופייני (TYP- TYPically). לדוגמה - רואים שהמתח על LED אדומה בהולכה הוא כ-2.3 וולט. בזרם של 20 מילי אמפר היא מאירה באור של 140 mcd, שהיא קיצור של mili candle – אלפיות אור נר. אורך הגל של האור האדום הוא כ-622 עד 625 nm שהוא קיצור של ננו מטר ($1\text{nm} = 10^{-8}$). בשורת הערכים המקסימאליים רואים שהזרם המקסימאלי דרך כל LED הוא 50 מילי אמפר. אם היא עובדת עם דפיקים אז הזרם המקסימאלי הוא 70 מילי אמפר.

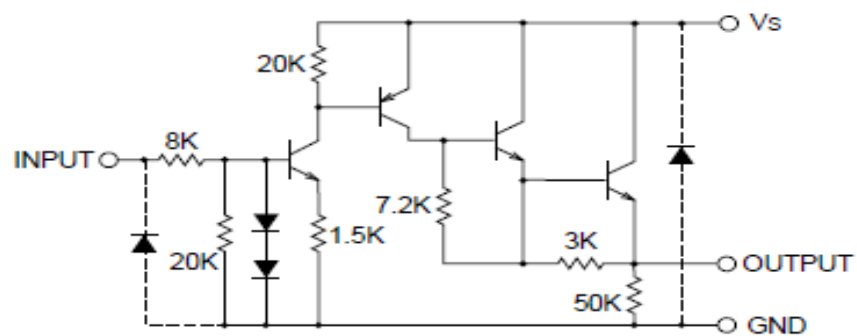
דוחף הזרם לאנודות - M54546

באיור 10 רואים את הדקי הרכיב M54546 המספק את המתח לאנודות.



איור 10 – הדקי הרכיב בזיווד של 18 הדקים

רואים שכל 8 הכניסות מחוברות בצד אחד של הרכיב ו 8 היציאות מהצד השני. הדק 9 של ה Vs הוא הדק המתח החיובי של הכרטיס והדק 10 הוא הדק האדמה. הכניסות מתחברות אל המיקרו בקר ATmega328 ואילו היציאות מתחברות אל האנודות של תצוגת המטריצה. אם נשים בכניסה '0' אז ביציאה נקבל '0' ואם נשים '1' אז ביציאה נקבל את המתח Vs. בכל כניסה יש מעגל הנראה באיור 11.



איור 11 - מעגל של אחד הערוצים

כאשר נשים '0' בכניסה - INPUT אז הטרנזיסטור Q1 בקטעון וגם Q2 בקטעון. אין זרם לבסיס Q3 ולכן גם הוא בקטעון. אין זרם לבסיס Q4 ולכן גם Q4 בקטעון. מכאן שאין זרם דרך טרנזיסטור זה והמתח על הנגד של ה 50K שבאמיטר שלו הוא 0 וולט. מכאן שביציאה OUTPUT יש מתח '0'.

כאשר נכניס '1' בכניסה אז Q1 יהיה ברוויה, Q2 יהיה בהולכה ויעביר זרם ל Q3 שייכנס גם הוא להולכה ויכניס את Q4 לרוויה. המתח ביציאה יהיה Vs פחות המתח Vcesat של טרנזיסטור Q4, כלומר כמעט Vs.

הדיודות המשורטטות עם קווים שבורים, אחת בכניסה והשנייה ביציאה, הן דיודות הנוצרות בתהליך ייצור הרכיב ומונעות כניסת מתחים הפוכים לרכיב.

באיור 12 נתונים הערכים הנקובים המקסימליים של הרכיב ומתחתיו הערכים המומלצים.

ABSOLUTE MAXIMUM RATINGS (Unless otherwise noted, Ta = -20 ~ +75°C)

Symbol	Parameter	Conditions	Ratings	Unit
VCEO	Collector-emitter voltage	Output, L	-0.5 ~ +50	V
Vs	Supply voltage		50	V
Vi	Input voltage		-0.5 ~ +30	V
Io	Output current	Current per circuit output, H	-500	mA
Pd	Power dissipation	Ta = 25°C, when mounted on board	1.79(P)/1.10(FP)	W
Topr	Operating temperature		-20 ~ +75	°C
Tstg	Storage temperature		-55 ~ +125	°C

RECOMMENDED OPERATING CONDITIONS (Unless otherwise noted, Ta = -20 ~ +75°C)

Symbol	Parameter	Limits			Unit	
		min	typ	max		
Vs	Supply voltage	0	—	50	V	
Io	Output current (Current per 1 circuit when 8 circuits are coming on simultaneously)	Duty Cycle P : no more than 8% FP : no more than 5%	0	—	-350	mA
		Duty Cycle P : no more than 55% FP : no more than 30%	0	—	-100	
VIH	"H" input voltage	2.4	—	25	V	
VIL	"L" input voltage	0	—	0.2	V	

איור 12 ערכים נקובים מקסימאליים וערכים מומלצים

DM163 - בקר התצוגה

8*3 CHANNEL CONSTANT CURRENT LED DRIVER

הרכיב הוא דוחף זרם ללדים המכיל את המרכיבים הבאים :

א. רגיסטרי הזזה

ב. נועלים (latches)

ג. 8*3 ערוצים של מעגלי זרם קבוע עם ערך זרם הנקבע על ידי 3 נגדים חיצוניים ו 64*256 רמת אפור של PWM (אפנון רוחב דופק) .

כל ערוץ מספק מקסימום זרם של עד 60 מילי אמפר. רמות הסקלה של האפור - **Gray Scale** (מוסבר בסעיף הבא) מופרדות בבנק 0 ובבנק 1 בהתאמה והם נבחרים על ידי הדק SELBK . בבנק 0 יש 6 ביטים (64 רמות) של סקלת האפור ובבנק 1 יש 8 ביטים של סקלת האפור (256 רמות) . ניתן לשתף את 2 הבנקים ולהשיג ביצוע של 8+6 ביטים של סקלת אפור. לחילופין ניתן לבחור האם לקבל בנק של 64 רמות אפור או 256 רמות.

ראינו שיש לנו כ 16.7 מיליון גוונים שונים ($3*8 = 24\text{bit}$ אפשרויות צבע - 16M) כאשר אנחנו רוצים להשתמש בטווח גוונים מושלם (זה נקרא **Color** או **Continues Tone**) . בסריקת תמונה בשחור-לבן בלבד (**LineArt**) או עם גווני אפור (**GrayScale**) אין צורך בשלושה בתים של מידע עבור כל פיקסל אלא סיבית אחת בסריקה של צבעי שחור-לבן כגון מסמכי טקסט, או שמונה סיביות לצורך גווני האפור (את גווני האפור ב RGB -ניתן לייצג רק על-ידי נתינת ערכים זהים בכל שלושת הרכיבים מה שאומר שיש 256 גוונים של אפור –ניתן לייצגם על ידי שמונה סיביות). במקרה שלנו נשתמש במושג סקלת אפור אבל באופן מעשי נקבל את כל ה 16M אפשרויות צבע.

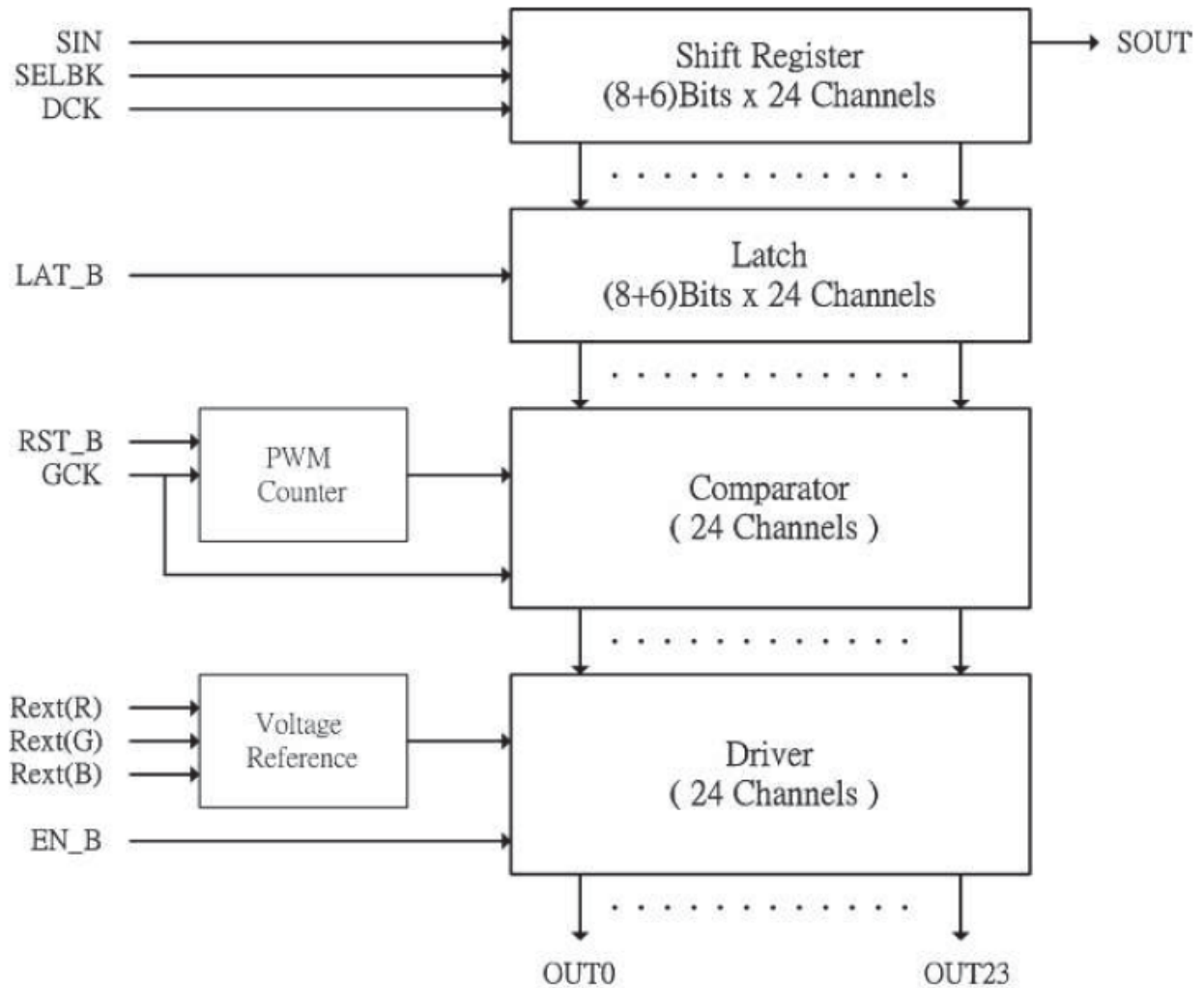
הרכיב יכול לשמש גם כבקר PWM לדוחפי זרם. כאשר VDDH מחובר ל VDD, כל אחת מ 24 הערוצים יכול לעבוד כמהפך לאות הדיגיטאלי.

מאפיינים

- 24 ערוצי יציאה.
- 8 + 6 ביטים של בקרת סקלת אפור של PWM
- זרם יציאה קבוע : 5 מילי אמפר עד 60 מילי אמפר.
- מתח ספק הכוח ללדים עד 17 וולט.
- VDD מ 3 עד 5 וולט.
- זרם יציאה משתנה הנקבע על ידי 3 נגדים חיצוניים.
- ארכיטקטורת הזזה טורית לסקלת אפור.

סכמה מלבנית

באיור מספר 6 נראית הסכימה המלבנית של הרכיב
הרכיב מורכב ממספר יחידות בסיסיות.



איור 6 – סכמה מלבנית של הרכיב

א. רגיסטרי הזזה Shift Registers

בחלק העליון של האיור יש את רגיסטרי ההזזה של 8 + 6 ביטים ל 24 הערוצים של הרכיב. הדק ה **SIN** (Serial IN) מצד שמאל למעלה באיור הוא ההדק בו נכניס את הנתון הטורי. הדק **DCK** (Data Clk) שמתחתיו הוא הדק השעון שמסנכרן את הנתון הטורי לתוך רגיסטרי ההזזה. הדק ה **SELBK** מפריד בין 2 קבוצות הבנק (בנק 0 ובנק 1). הדק **SOUT** הוא הדק שניתן לחבר אותו להדק **SIN** של רכיב מטריצת לדים נוספת בקסקדה.

ב. מערכת הנועלים - Latch

הנתונים מרגיסטרי ההזזה מתחברים אל מערכת נועלים. כאשר סיימנו להכניס את רצף הנתונים הרצוי מעלים את ההדק LAT_B (LATCH B) ל '1' ואז הנתונים שהכניסות הנועלים ננעלים ליציאות שלהם ומגיעים למשווים (COMPARATORS).

ג. משווים – Comparators

הכניסות של המשווה מושוות עם מונה אפנון רוחב הדופק - **PWM Counter** – ובהתאמה מוציאות נתונים אל דוחפי הזרם **Drivers** וכך נקבע עצמת ההארה המשתנה של הלידים.

ד. דוחפי הזרם Driver

מקבלים את הנתונים מהמשווים. היציאות של דוחפי הזרם מתחברים בהתאמה אל 3 הקבוצות של הלידים. מצד שמאל של דוחפי הזרם יש מתח ייחוס המתחבר ל 3 הדקים **Rext(R) Rext(G)** **Rext(B)**. להדקים אלו מחברים 3 נגדים חיצוניים של 1 קילו אום הקובעים את רמת הזרם של האדום ירוק וכחול בהתאמה. הסבר מפורט יותר עבור כל ערוץ - באיור 8.

תפקיד ההדקים

ההדק	תפקיד
VDDH(R) VDDH(G) VDDH(B)	הדקי הגנת יציאה. הם יכולים להתחבר כל אחד לחוד או למתח הספק של הLED (VLED)
VDD	הדק ה + של ספק הכוח
VSS	הדק האדמה (המינוס של ספק הכוח).
SIN	Serial Input הכניסה הטורית
SOUT	Serial Out - יציאה טורית לכרטיס קולורדואינו נוסף
DCLK	פולסי שיעון להכנסת הנתון הטורי שב SIN . העברת הנתון הטורי מתבצעת בעלית פולס השעון.
SELBK	SElect Bank - בחירת בנק. כאשר בהדק יש '1' הנתונים ייכנסו לבנק 1 של ה 8 ביט. אם בהדק יש '0' הנתון ייכנס לבנק 0 של ה 6 ביט.
LAT_B	במעבר מ '1' ל '0' הנתונים בשני רגיסטרי ההזזה ננעלים ביציאת הנועלים.
GCK	כניסת שעון לפעולת PWM
Rext(R) Rext(G) Rext(B)	External Resistor - נגד היצוני המתחבר בין ההדק לאדמה. הוא קובע את זרם הדרייבר עבור כל צבע. Rext(R) שולט על יציאות ה out שמספרן 0,3,6,9,12,15,18,21 Rext(G) שולט על יציאות ה out שמספרן 1,4,7,10,13,16,19,22 Rext(B) שולט על יציאות ה out שמספרן 2,5,8,11,14,17,20,23
IOUT 0 ~ 23	יציאות הזרם של הדרייבר ללדים
EN_B	Enable output - כאשר הדק כניסה זה ב '1' כל יציאות IOUT ב OFF (מצב שלישי).
RST_B	Reset . הרכיב מאותחל כאשר בהדק זה יש '0' . יש בהדק זה נגד pull up פנימי . לפני שימוש ברכיב יש לתת לו reset . אם כל ערוץ מתוכנן להפעיל מספר לדים יש לבצע reset לפני כל נעילת לדים כדי למנוע הבהוב.

ערכים נקובים מקסימליים וערכים מומלצים

באיור 7 רואים בטבלה הראשונה את הערכים הנקובים המקסימליים להפעלת הרכיב ומתחתיה הערכים המומלצים להפעלה.

Maximum Ratings (Ta=25°C, Tj(max) = 140°C)

CHARACTERISTIC	SYMBOL	RATING	UNIT
Supply Voltage	VDD	-0.3 ~ 7.0	V
Input Voltage	VIN	-0.3 ~ VDD+0.3	V
Output Current	IOUT	60	mA
Output Voltage	VOUT	-0.3 ~ 17	V
DCK Frequency	FDCK	20	MHz
GCK Frequency	FGCK	20	MHz
GND Terminal Current	IGND	1440	mA
Power Dissipation	PD	1.36 (QFP44); 3.63 (QFN40) (Ta=25°C)	W
Thermal Resistance	Rth(j-a)	84.42 (QFP44); 31.67 (QFN40)	°C/W
Operating Temperature	Top	-40 ~ 85	°C
Storage Temperature	Tstg	-55 ~ 150	°C

Recommended Operating Condition

DC Characteristics (Ta = 25°C)

CHARACTERISTIC	SYMBOL	CONDITION	MIN.	TYP.	MAX.	UNIT
Supply Voltage	VDD	—	3	—	5.5	V
Output Voltage	VOUT	—	—	—	17	V
Output Current	Io	OUTn	5	—	60	mA
	IOH	SERIAL-OUT	—	—	2	
	IOL	SERIAL-OUT	—	—	-2	
Input Voltage	VIH	—	0.8 VDD	—	VDD+0.2	V
	VIL	—	-0.2	—	0.2 VDD	

איור 7 – ערכים נקובים מקסימליים וערכים מומלצים

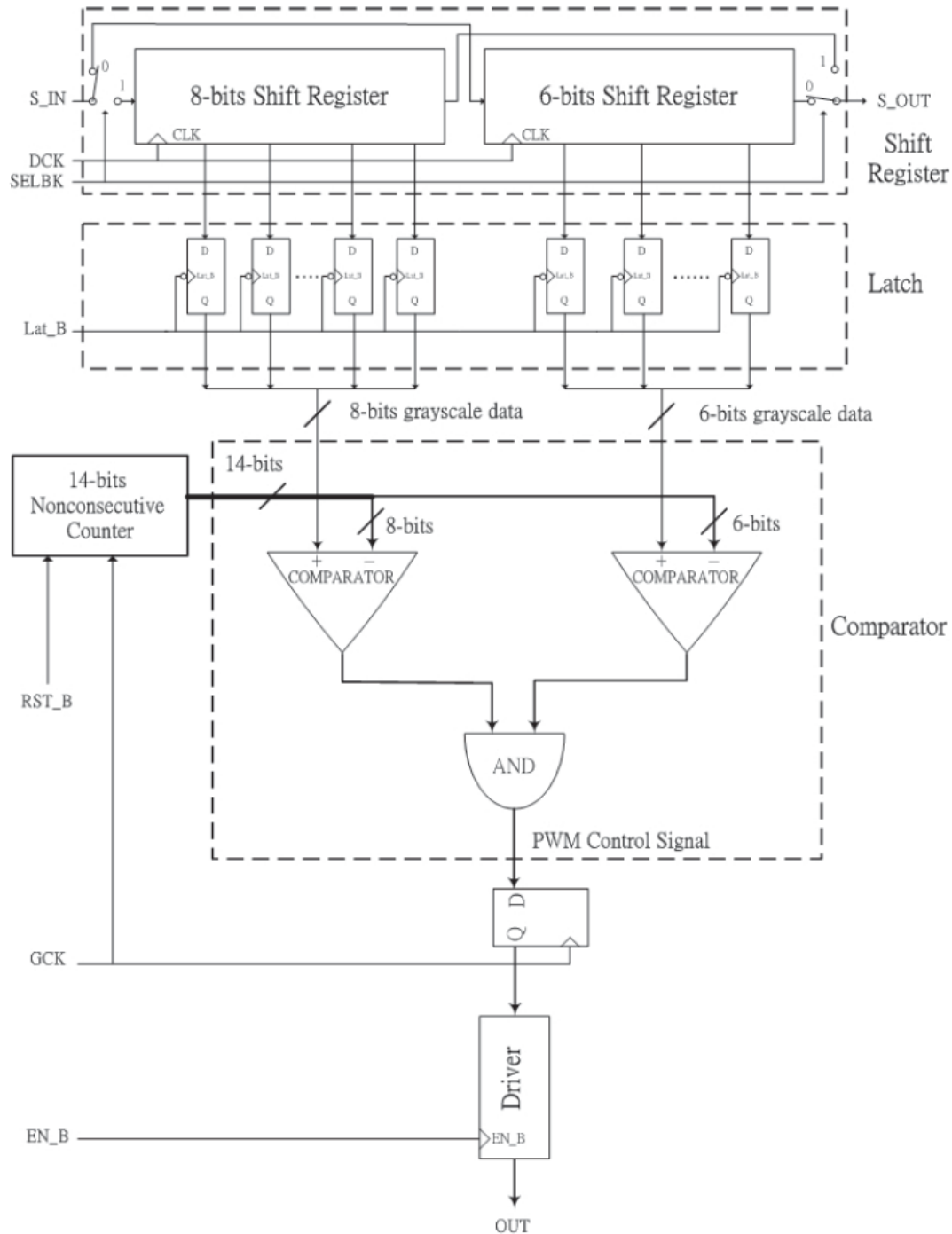
בטבלה העליונה:

- מתח הספק VDD – בין 7 ~ -0.3 וולט בטבלה העליונה. מתח כניסה Vin – בין -0.3 וולט ל VDD+0.3 וולט.
- זרם יציאה IOUT – מקסימום 60 מילי אמפר.
- מתח יציאה VOUT בין -0.3 וולט עד 17 וולט.
- תדר עבודה מקסימלי לפולסי השעון להזנת הנתון - FDCK – 20 מגה הרץ.
- תדר עבודה מקסימלי לפולסי השעון של ה PWM – FGCK – 20 מגה הרץ.
- זרם מקסימלי בהדק האדמה - IGND – 1440 מילי אמפר.
- לאחר מכן נתונים תצרוכת ההספק של הרכיב עבור 2 סוגי האריזה שמהם הוא יכול להיות בנוי - PD – בטמפרטורת החדר של 25 מעלות צלסיוס. לאחר מכן מופיעים ההתנגדות הטרמית שלו (מבטאת את יכולת פיזור החום שלו) ואח"כ מופיעות תחום טמפרטורת העבודה ותחום טמפרטורת האחסנה.

בטבלה שלמטה מופיעים המתח המומלץ לעבוד עם מתח ספק הכוח VDD, מתח היציאה VOUT, זרם היציאה בהדק IOUT ובהדק SOUT במצב גבוה ונמוך ומתח הכניסה לכל אחד מההדקים במצב גבוה ונמוך.

סכמה מפורטת של ערוץ

באיור 8 רואים את החלקים הבאים :



איור 8 - סכמה מפורטת של אחד הערוצים

א. רגיסטרי הזזה

נמצאים בחלק העליון של האיור. הרגיסטר השמאלי של 8 ביט והימני של 6 ביט. כניסת ה S_IN היא של הנתון הטורי. הנתון יגיע לאחד מרגיסטרי ההזזה. ההדק SELBK (SElect BanK) קובע לאיזה רגיסטר (או בנק) יגיע הנתון. במצב של '0' בהדק ה SELBK, הנתון מגיע לרגיסטר ההזזה של ה 6 ביט (בנק 0) ואילו במצב של '1' הנתון מגיע לרגיסטר של ה 8 ביט (בנק 1). גם הדק ה S_OUT שמצד ימין (לחיבור מספר כרטיסי קולורדואינו בהתאמה), מושפע מהדק ה SELBK ומתחבר לרגיסטר ההזזה המתאים.

הדק ה DCK - הוא הדק השעון המסנכרן את הנתון הטורי לתוך רגיסטר ההזזה המתאים.

ב. הנועלים - LATCH

נמצאים מתחת לרגיסטרי ההזזה. כל נועל הוא D-FF. לאחר הכנסת הנתון נותנים פקודת נעילה בהדק Lat_B ואז הנתון מועבר מכניסות הנועלים ליציאות של הנועלים ומגיע למשווים-COMPARATOR.

ג. המשווים - Comparator

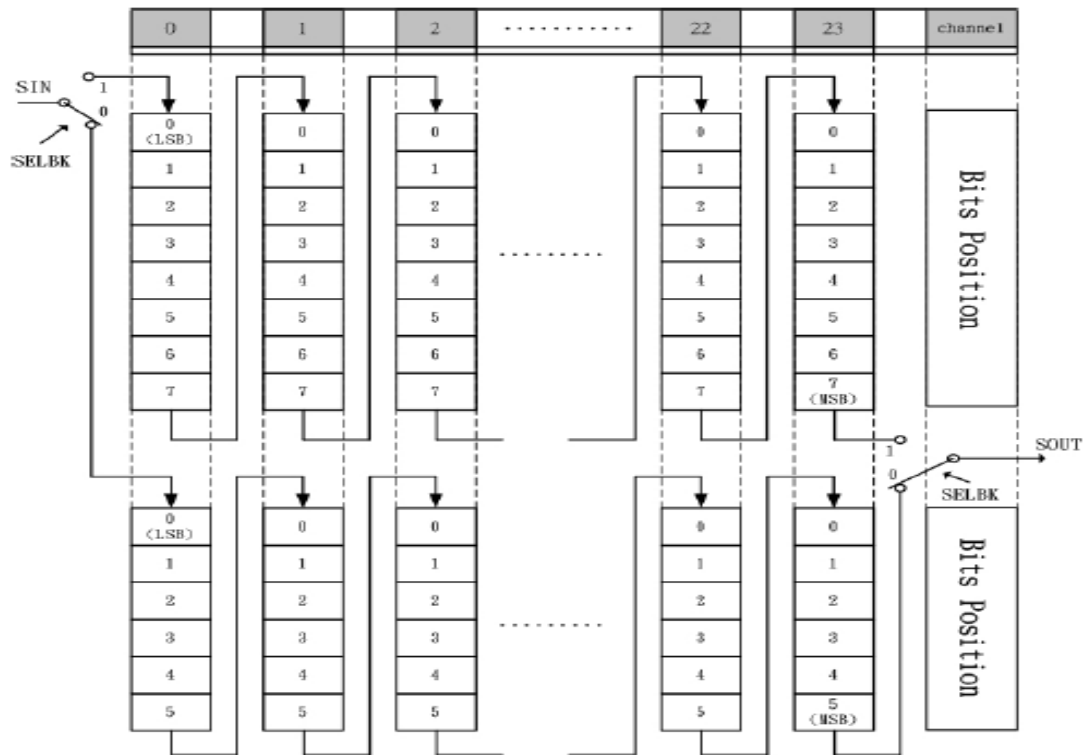
מקבלים מצד אחד את מצב מונה בן 14 ביטים ומצד שני את הנתון. כאשר מצב המונה קטן מהנתון יוצא '1' מהמשווה וכאשר המונה גדול מנתון יוצא '0'. היציאה מגיעה לשער AND שמבצע פעולת AND בין 2 הבנקים. יציאת השער היא אות בקרה של PWMControl Signal – PWM. אות זה מגיע ל-D-FF שמעביר את הנתון שב D שלו כאשר מגיע אות GCK שהוא פולס שעון בפעולת PWM שמעבירה את האות לדוחפי הזרם.

ד. דוחפי הזרם - DRIVER

מקבל את האות מהדק Q של ה D_FF ואם יש אפשרות ב EN_B (0 בהדק) אז האות יוצא ל OUT.

רגיסטרי ההזזה

באיור 9 נראים רגיסטרי ההזזה.

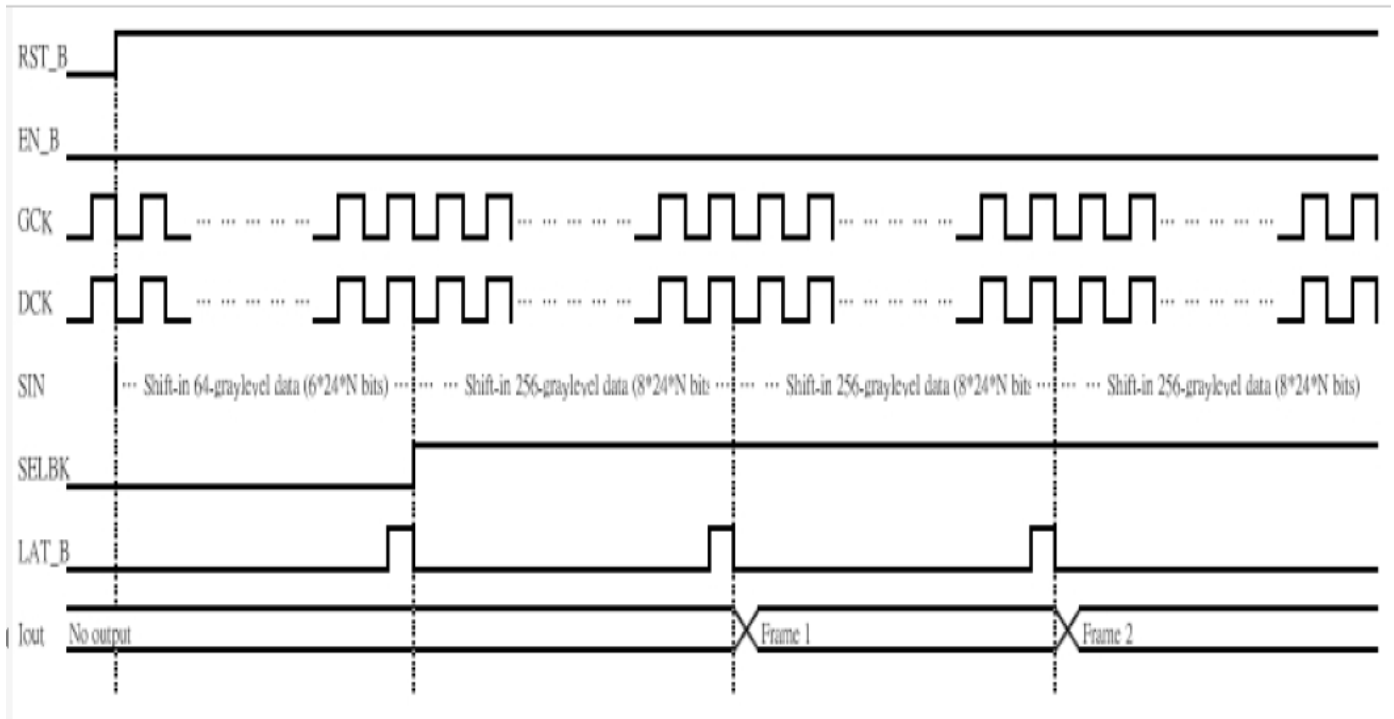


איור 9 – רגיסטרי ההזזה

בחלק העליון של האיור מתוארים הערוצים, מערוץ 0 ועד 23. בכל ערוץ יש בחלק העליון את בנק 1 עם ה 8 ביט ובתחתון את בנק 0 עם ה 6 ביט. ארכיטקטורת הזזה טורית זו היא First In First Out - FIFO - הראשון שנכנס הוא הראשון שיוצא. ביט ה MSB - זה שבערוץ ה 23 הוא ביט הנתון הראשון שנכנס להזזה ואילו ביט ה LSB, הביט הראשון בערוץ הראשון הוא הביט שנכנס אחרון לרגיסטרי ההזזה. גם כאן רואים שהדק SELBK במצב 0 מנתב את הנתון לבנק 0 של ה 6 ביט ובמצב 1 הנתון הטורי מנותב לבנק 1 של ה 8 ביט.

תיאור הכנסת נתונים ל 64 רמות אפור כגורמי תיקון ו 256 רמות אפור של תמונה

איור מספר 10 מתאר דיאגרמת זמנים של אותות הכניסה לרכיב כאשר רוצים 64 רמות אפור עבור התיקון (רמת הלבן) ו 256 רמות אפור של תמונה. מניחים שמספר תצוגות מטריצה מחוברות בטור. ההדקים המשורטטים באיור הם הדק RST_B שנותן איפוס כללי



איור מספר 10 – דיאגרמת הכנסת נתונים ל 64 רמות של תיקון ו 256 רמות אפור של תמונה

תהליך הכנסת הנתונים יהיה לפי השלבים הבאים:

א. שמים בהדק SELBK=0 (כדי לעבוד עם בנק 0) והכנסת הנתונים של 6 ביטים של תיקון לרגיסטר ההזזה. סה"כ נכניס 24 בייט של 6 ביט כל אחד לקביעת הצבע – סה"כ $24 \cdot 6 = 144$ ביטים. אם יש 2 כרטיסי קולורדואינו מחוברים בטור (קסקדה), נכניס כמות כפולה של ביטים - 288. יש לזכור ש SOUT של כרטיס אחד הוא SIN של הכרטיס אחריו ומכאן שהביט הראשון שנכנס יגיע אחרי 288 הזזות ויהיה ביט ה MSB בערוץ 24 של של הכרטיס השני. אם יש כרטיס קולורדואינו נוסף אז יש להעביר עוד 144 ביטים וכך הלאה. המספר N באיור מראה כמה כרטיסי קולורדואינו מחוברים בטור.

ב. מעבירים את הדק LAT_B ל'1' כדי שהנתון יעבור מכניסת הנועלים וייעל ביציאתם ואז מורידים את ההדק ל'0'.

ג. מעבירים את הדק SELBK ל'1' כדי לעבוד עם בנק 1.

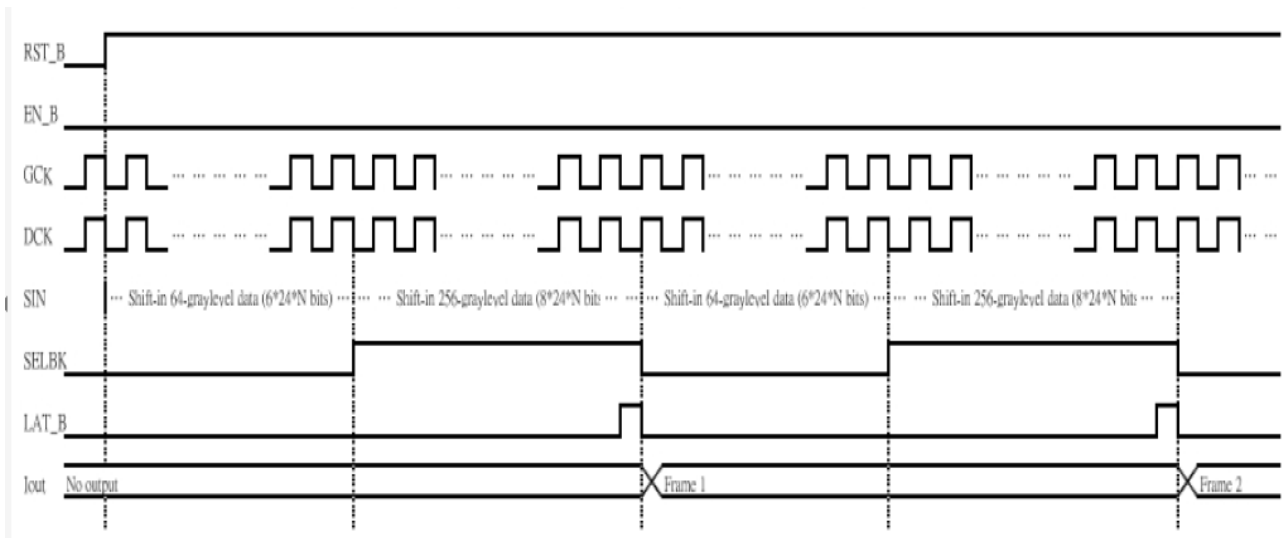
ד. מבצעים הכנסת נתונים של התמונה (image) של 8 ביט. הנתונים שנכניס לרגיסטרי ההזזה הם : יש 8 שורות של 8 פיקסלים . לכל שורה של 8 פיקסלים יש 3 צבעים (Red Green Blue) - אדום, ירוק, כחול) ומכאן שיש 24 ביטים. היות ויש 8 שורות אז כמות הבתים היא $24 * 8 = 192$. סה"כ יש 192 ביטים להעביר ברגיסטרי ההזזה. אם יש כרטיס נוסף אז כמות הביטים היא כפולה וכך הלאה.

ה. מעבירים את הדק LAT_B ל'1' כדי שהנתון יעבור מכניסת הנועלים ויינעל ביציאתם ואז מורידים את ההדק ל'0' .

ו. חוזרים על שלבים ד ו ה .

תיאור הכנסת נתונים ל 8 ביטים + 6 ביטים לתמונה

באיור 12 מתוארת דיאגרמת זמנים כאשר גם ה 8 ביטים וגם ה 6 ביטים משמשים לתמונה.



איור 12 - דיאגרמת זמנים כאשר גם ה 6 ביטים וגם ה 8 ביטים משמשים לתמונה.

אחרי שהרכיב היה ב '0' וקיבל אתחול –reset, מעלים את הדק ה RST_B ל 1 ומתחילים להכניס נתונים ולהזיז אותם.

תהליך הכנסת הנתונים יהיה לפי השלבים הבאים:

א. שמים בהדק $SEKBK=0$ (כדי לעבוד עם בנק 0) והכנסת הנתונים של 6 ביטים של נתונים של התמונה לרגיסטר ההזזה. סה"כ נכניס 24 בייט של 6 ביט כל אחד לקביעת הצבע – סה"כ $24 * 6 = 144$ ביטים. אם יש 2 כרטיסי קולורדואינו המחוברים בטור (קסקדה), נכניס כמות כפולה של ביטים - 288. יש לזכור ש SOUT של כרטיס אחד הוא SIN של הכרטיס אחריו ומכאן שהביט הראשון שנכנס יגיע אחרי 288 הזזות ויהיה ביט ה MSB בערוץ 24 של הכרטיס השני. אם יש כרטיס קולורדואינו

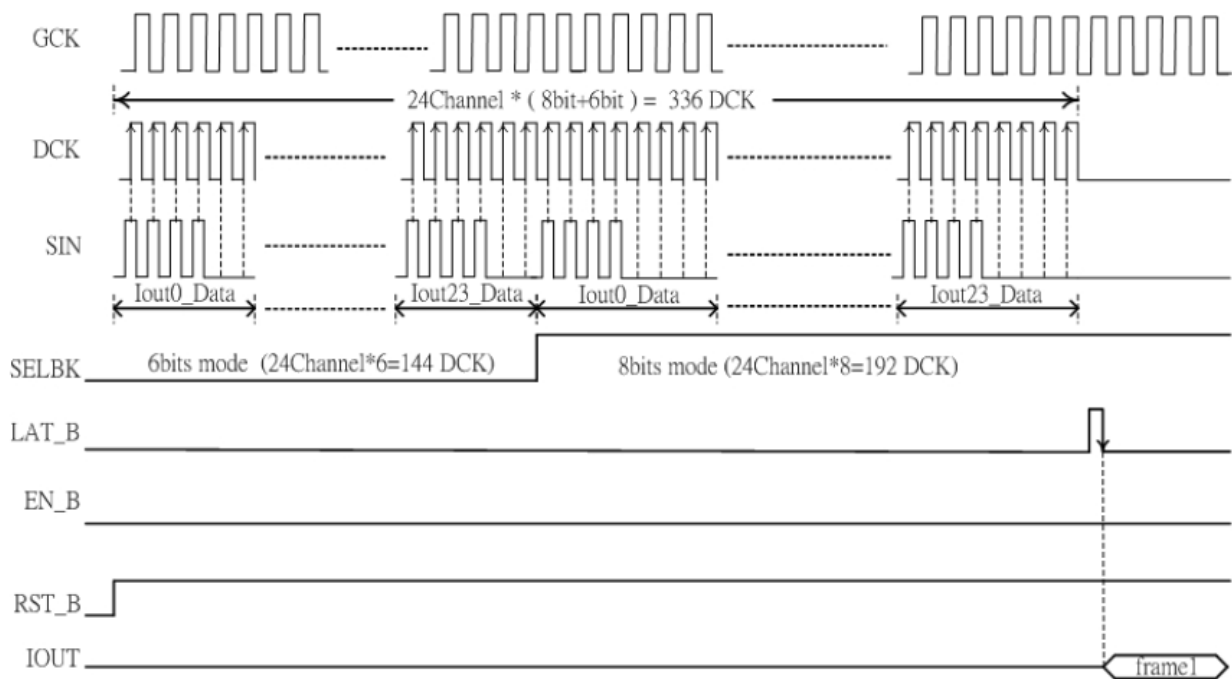
נוסף אז יש להעביר עוד 144 ביטים וכך הלאה. המספר N באיור מראה כמה כרטיסי קולורדואינו מחוברים בטור.

ב. מעבירים את הדק SELBK ל '1' כדי לעבוד עם בנק 1.

ג. מבצעים הכנסת נתונים של התמונה (image) של 8 ביט. הנתונים שנכניס לרגיסטרי ההזזה הם : יש 8 שורות של 8 פיקסלים . לכל שורה של 8 פיקסלים יש 3 צבעים (Red Green Blue) - אדום, ירוק, כחול) ומכאן שיש 24 ביטים. היות ויש 8 שורות אז כמות הבתים היא $24 = 3 * 8$. סה"כ יש $192 = 24 * 8$ ביטים להעביר ברגיסטרי ההזזה. אם יש כרטיס נוסף אז כמות הביטים היא כפולה וכך הלאה.

ד . מעבירים את הדק LAT_B ל '1' כדי שהנתון יעבור מכניסת הנועלים ויינעל ביציאתם ואז מורידים את ההדק ל '0' . רואים שמופיע מסגרת תמונת המסך הראשון בתצוגה (frame1) . ה. חוזרים על שלבים א עד דלקבלת מסכים נוספים.

באיור 13 מתוארת דיאגרמת זמנים מפורטת של הכנסת הנתונים והזתם.



איור 13 - דיאגרמת זמנים מפורטת של הזזת הנתונים

פולסי השעון הסינכרוניים של ה DCK עובדים על עליה מ '0' ל '1' . ההדק LAT_B נועל את הנתונים בירידה מ '1' ל '0' .

סה"כ יש להכניס 336 ביטים כדי להכניס את הנתונים עבור מסגרת תמונה אחת לפי החישוב הבא : 6 ביטים כפול $24 = 144$ בתים (המספר 24 הוא לכל שורה יש 8 בתים ו 3 צבעים GB) . 8 ביטים * 24 בתים הם 192 ביטים.

336 = 144 + 192 ביטים.

בדוגמה שבשרטוט רואים שהנתון שהכנסנו לרגיסטר של ה 6 ביט הוא 001111 (ה MSB ראשון) ולרגיסטר של ה 8 ביט הכנסנו 00001111. מכאן שזרם היציאה הממוצע יהיה :

$$I_{OUT} * 15/64 * 15/256$$

הנוסחה שהשתמשנו לחישוב זרם היציאה הממוצע בהדק I_{OUT} כל שהוא :

$$I_{OUT \text{ avg}} = I_{OUT} * \text{BANK0}/64 * \text{BANK1}/256$$

כאשר I_{OUT} הוא זרם הייחוס ותלוי בנגד R_{ext} . הוא ניתן לחישוב מהנוסחה:

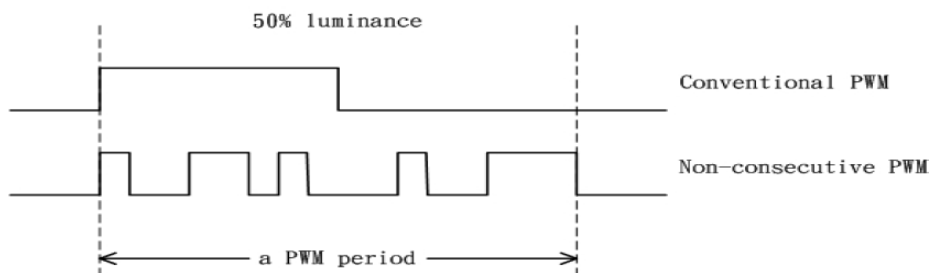
$$I_{OUT} = 47 * V_{R_{ext}} / R_{ext}$$

PWM ייחודי

הרכיב DM163 כולל מונה PWM שונה ומכאן שלצורות הגל שלו מאפיינים שונים יחסית למונה PWM רגיל.

1. מונה לא רציף

צורת הגל של המונה נראית באיור 14 :



איור 14 - צורת הגל של מונה לא רציף.

בצורת הגל העליונה רואים עבודה עם PWM רגיל ואילו בשנייה עם המונה הלא רציף. בשני המקרים מדובר בעצמת הארה של 50%. צורת הגל העליונה מראה שחצי מחזור PWM יש הולכת זרם וחצי מחזור השני לא. צורת הגל התחתונה מראה שמחזור PWM מתפרש על כל המחזור וגורם לפולסי ביניים. סיכום של כל פולסי הביניים נותן 50% של הארה כמו ב PWM הרגיל. פרישה זו עוזרת למנוע הבהוב הלדים בעצמת סקלת אפור נמוכה.

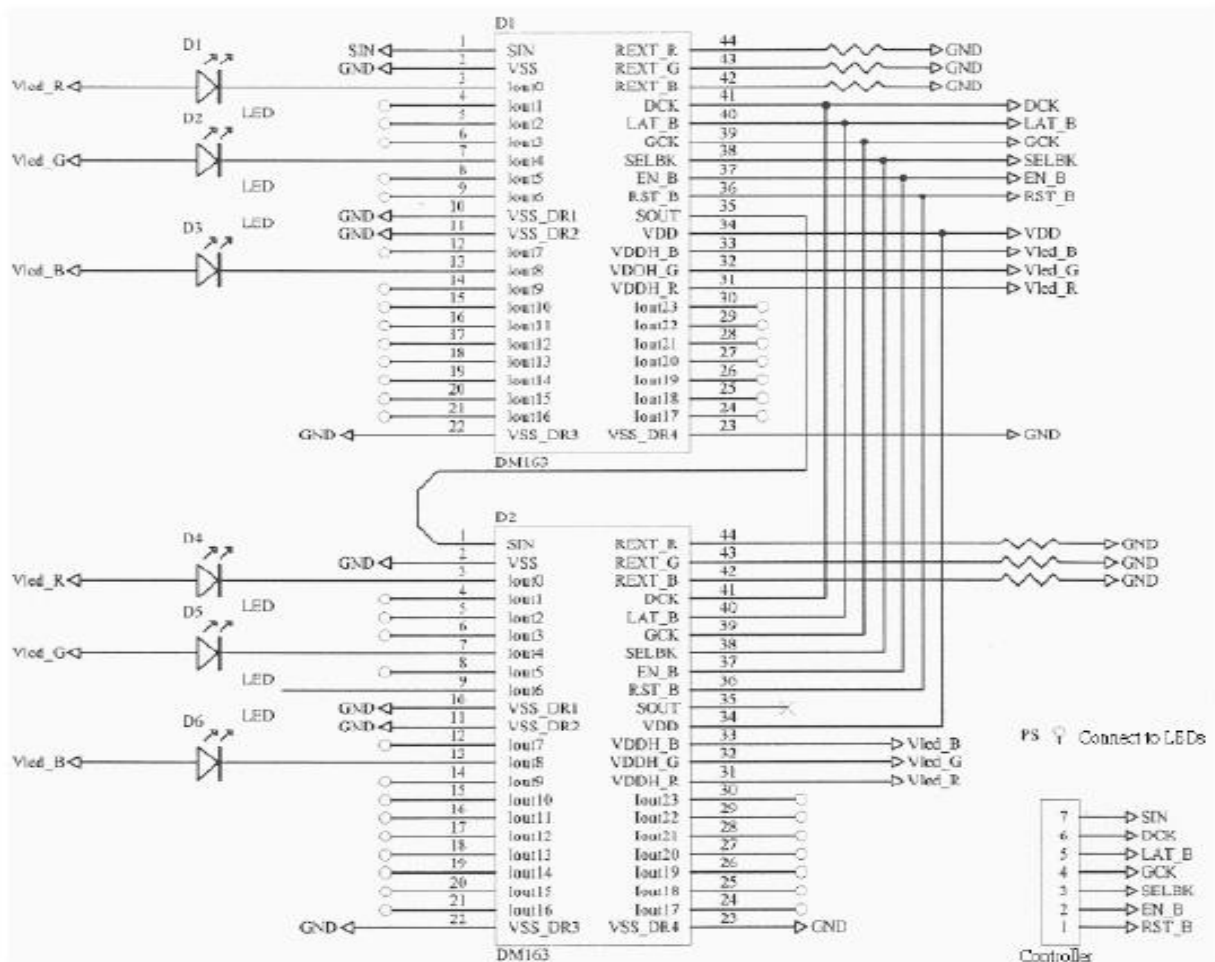
2. משווה 8 + 6 ביטים.

יציאת המשווה שבאיור 8 תהיה בגובה רק כאשר הערך בהדק '+' של המשווה גדול מהערך בהדק ה '-'. כאשר גם 8 הביטים וגם 6 הביטים של המשווה יהיו בגובה יהיה זרם ביציאת הערוץ. כאשר הערך ב '+' יהיה שווה או קטן מהערך ב '-' יהיה 0 ביציאת המשווה.

כתוצאה מתכנון המשווה המיוחד הרכיב מציג אופייני יציאה בשני מקרים. במקרה הראשון יציאות הרכיב יהיו ב OFF כאשר אחד משני הבנקים (של ה 6 ביט או ה 8 ביט) מלא באפסים. במקרה השני כאשר כל הביטים בשני הבנקים נמצאים בגבוה. יציאות הרכיב נותנות את מקסימום ההארה (אבל לא 100% . בגלל אופי תכנון המשווה אות הבקרה של ה PWM יהיה 0 בתנאי שכל הביטים בכל ה FF (דלגלים) של 2 הבנקים יהיו ב '1' . מכאן שאות ה PWM יהיה 0 ל 2^6+2^8 פולסי שעות CGK במקום כל הזמן גבוה.

דוגמה לחיבור 2 רכיבים בקסקדה

באור 15 מתואר חיבור 2 רכיבי DM163 בקסקדה (בטור אחד אחרי השני).



איור 15 – חיבור שני רכיבי DM163

- הדק SOUT של הרכיב העליון מחובר להדק SIN ברכיב התחתון.
- ההדקים המחוברים במקביל בשני הרכיבים הם : DCK , LAT_B , GCK , SELBK , EN_B , RST_B .

- צריך להיות מחובר אל VDDH_R/G/B בהתאמה. ה Vled_R/G/B מתחזי ספק הכוח ללדים של האדום / ירוק / כחול .
- VSS_DR הוא הדק האדמה של הלדים ואפשר לחבר אותו ל VSS .

באיור 16 נתונים דפי הנתונים של הרכיב M54564

MITSUBISHI SEMICONDUCTOR <TRANSISTOR ARRAY>
M54564P/FP
 8-UNIT 500mA SOURCE TYPE DARLINGTON TRANSISTOR ARRAY

DESCRIPTION

M54564P and M54564FP are eight-circuit output-sourcing Darlington transistor arrays. The circuits are made of PNP and NPN transistors. Both the semiconductor integrated circuits perform high-current driving with extremely low input-current supply.

FEATURES

- High breakdown voltage ($BV_{CE0} \geq 50V$)
- High-current driving ($I_o(max) = -500mA$)
- With output pulldown resistance (Driving available with fluorescent display tube)
- Driving available with PMOS IC output or with TTL output
- Wide operating temperature range ($T_a = -20$ to $+75^\circ C$)
- Output current-sourcing type

APPLICATION

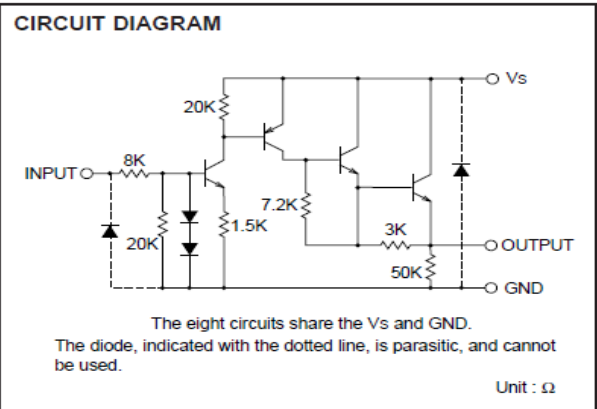
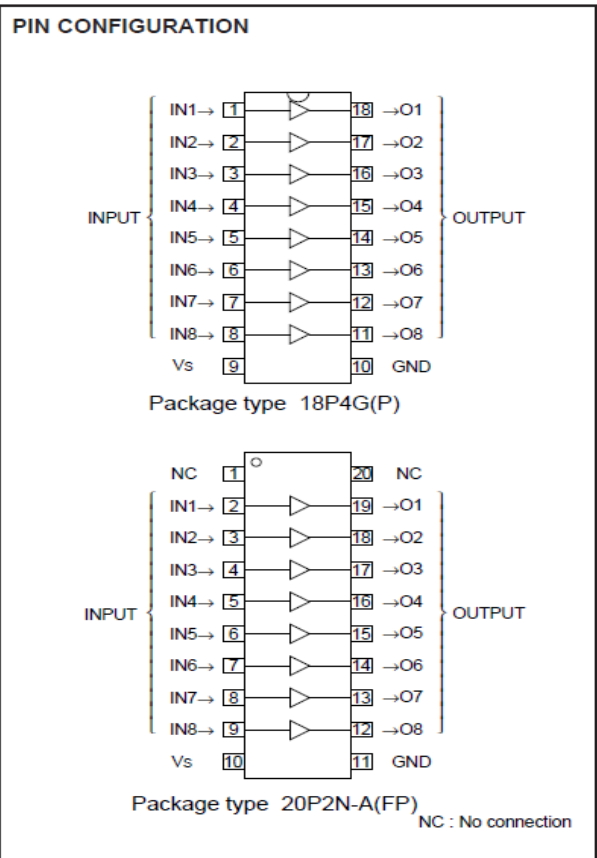
Drives of relays, printers, LEDs, fluorescent display tubes and lamps, and interfaces between MOS-bipolar logic systems and relays, solenoids, or small motors

FUNCTION

The M54564P and M54564FP each have eight circuits, which are made of input inverters and current-sourcing outputs. The outputs are made of PNP transistors and NPN Darlington transistors. The PNP transistor base current is constant. Resistance of $50k\Omega$ is connected between each output pin and GND, making these ICs suitable for fluorescent display tubes. V_s and GND are used commonly among the eight circuits.

Output current is 500mA maximum. Supply voltage V_s is 50V maximum.

The M54564FP is enclosed in a molded small flat package, enabling space-saving design.



ABSOLUTE MAXIMUM RATINGS (Unless otherwise noted, Ta = -20 ~ +75°C)

Symbol	Parameter	Conditions	Ratings	Unit
V _{CEO}	Collector-emitter voltage	Output, L	-0.5 ~ +50	V
V _S	Supply voltage		50	V
V _I	Input voltage		-0.5 ~ +30	V
I _O	Output current	Current per circuit output, H	-500	mA
P _d	Power dissipation	Ta = 25°C, when mounted on board	1.79(P)/1.10(FP)	W
T _{opr}	Operating temperature		-20 ~ +75	°C
T _{stg}	Storage temperature		-55 ~ +125	°C

RECOMMENDED OPERATING CONDITIONS (Unless otherwise noted, Ta = -20 ~ +75°C)

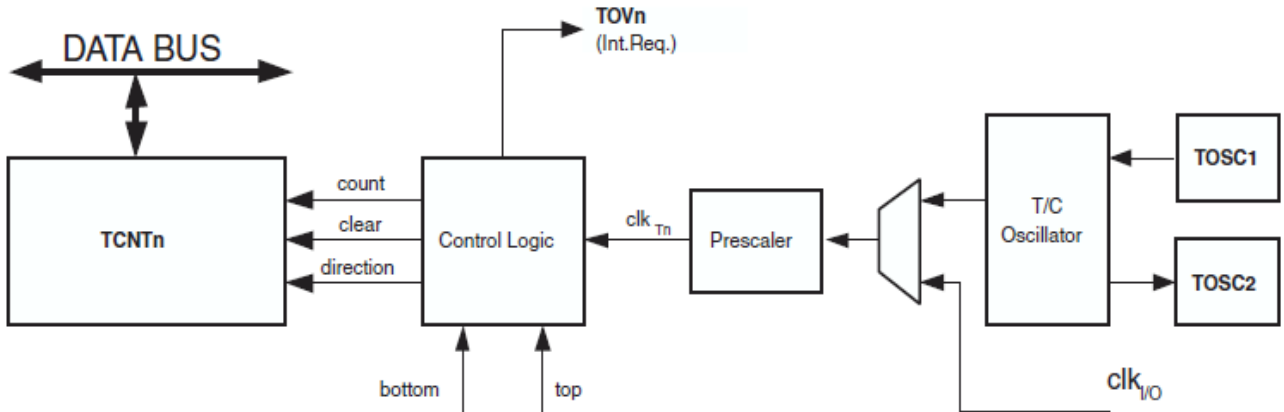
Symbol	Parameter	Limits			Unit	
		min	typ	max		
V _S	Supply voltage	0	—	50	V	
I _O	Output current (Current per 1 circuit when 8 circuits are coming on simultaneously)	Duty Cycle P : no more than 8% FP : no more than 5%	0	—	-350	mA
		Duty Cycle P : no more than 55% FP : no more than 30%	0	—	-100	
V _{IH}	"H" input voltage	2.4	—	25	V	
V _{IL}	"L" input voltage	0	—	0.2	V	

איור 16: דפי נתונים של דוחף הזרם לאנודות M54564

תוכנה

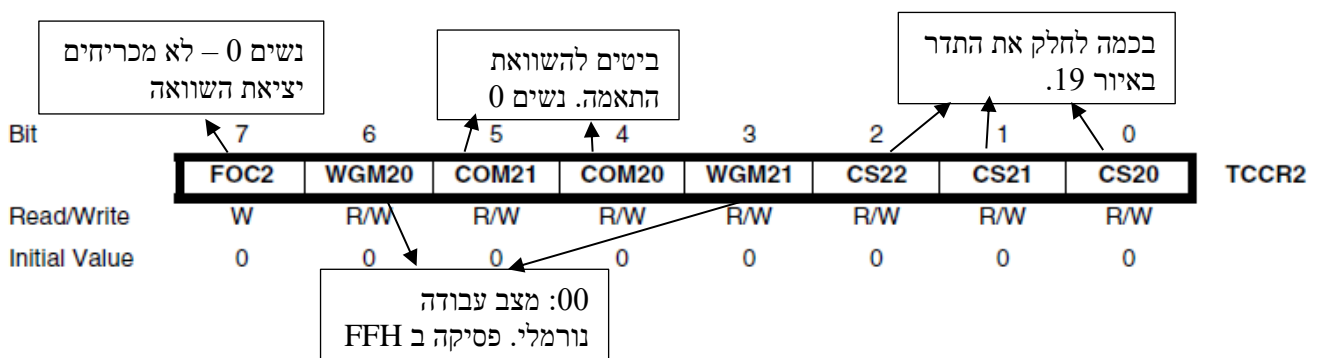
ניתן על קצה המזלג כיוונים של התוכנה שיש לרשום.
 כפי שהסברנו מקודם, כאשר מכניסים 3 צבעים (RGB) – 3 בתים - עבור כל לד (שניתן לקרוא לו גם פיקסל של תמונה) אז לשורה שבה יש 8 פיקסלים סה"כ הביטים שהכנסנו הם: $192 = 8 * 3 * 8$
 ביטים. אחרי שהכנסנו 192 ביטים לשורה אחת יש לעבור ולהכניס 192 ביטים עבור 8 הלדים בשורה הבאה וכך הלאה עד השורה השמינית ושוב להתחיל מהשורה הראשונה עד ה 8 וחוזר חלילה.
 באופן תאורטי כדי לקבל תמונה - מסגרת מטריצה שלמה של 8 על 8 - יש לרענן את התצוגה למעלה מ 16 פעמים בשנייה. באופן מעשי נרענן את המריצה בקצב גבוה בהרבה בצורה הבאה:
 נפעיל טיימר שייתן לנו כ 640 פסיקות בשנייה, כלומר כל $1/640 = 1.56 \text{mSec}$ נבצע ריענון לאחת מתוך 8 השורות. אם נחלק $640/8$ נקבל 80 וזה אומר שכל שורה מרועננת 80 פעם בשנייה וכך נקבל תצוגה יציבה ולא מרטטת.

טיימר 2 של המיקרו בקר AT328 הוא זה שקובע את קצב הריענון. הארדואינו עובד עם תדר של 16MHz. תדר זה מגיע למחלק תדר אותו נקבע בעזרת תכנה לחלק ב 128 ונקבל ביציאת מחלק התדר תדר של : $16 \cdot 10^6 / 128 = 125000\text{Hz}$. תדר זה מגיע אל טיימר 2 שהוא בן 8 ביטים. איור 17 מתאר את המבנה של טיימר 2.



איור 17 : מבנה טיימר 2

המונה/טיימר עצמו נמצא בצד שמאל בשרטוט – TCNTn שהוא קיצור של TimerCounter כאשר n במקרה שלנו הוא 2. המקור של פולסי הספירה מגיע מצד ימין בשרטוט. הפולסים יכולים להיות פנימיים מהגביש של 16 מגה הרץ (מלבן TOSC1) או חיצוניים מהקו clk_{io}. אנחנו נכניס את התדר הפנימי. מלבן ה Prescaler – קדם כיוול. זהו מחלק תדר שנשלט על ידי המתכנת והוא מחלק את התדר שנכנס במספר שאנחנו רוצים. במקרה שלנו נחלק ב 128. ראה איור 18 ו 19 ..



איור 18 : מבנה רגיסטר הבקרה של טיימר 2 - TCCR2

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{T2S} /(No prescaling)
0	1	0	clk _{T2S} /8 (From prescaler)
0	1	1	clk _{T2S} /32 (From prescaler)
1	0	0	clk _{T2S} /64 (From prescaler)
1	0	1	clk _{T2S} /128 (From prescaler) ← הבחירה שלנו
1	1	0	clk _{T2S} /256 (From prescaler)
1	1	1	clk _{T2S} /1024 (From prescaler)

איור 19 : הביטים שקובעים את חלוקת התדר של ה Prescaler .

נשים בטיימר את המספר 61 ואז הוא יספור מ 61 עד שיגיע למספר 255 . הפולס הבא יגרום לגלישה ולבקשת פסיקה של טיימר 2 . בפונקציית הפסיקה נטען את הטיימר שוב ב 61 ונבצע תכנית המדליקה את המצב של שורה מסוימת. בפסיקה הבאה נפעיל את השורה הבאה וכך הלאה.
 בפונקציית ה `setup()` נזמן פונקציה לאתחול טיימר 2 :

```
void Timer2Init()
{
  // Mhz... תדר הגביש בארדואינו 16
  // Timer2 (8bit) אתחול :
  // החלוקה ב: CS22 CS21 CS20 : ערכי חלוקת התדר prescaler
  //           0    0    0    stopped
  //           0    0    1    /1
  //           0    1    0    /8
  //           0    1    1    /32
  //           1    0    0    /64
  //           1    0    1    /128
  //           1    1    0    /256
  //           1    1    1    /1024
  // טיימר 2 מתקדם ב 1 כל 16MHz / prescaler
  // אם נחלק ב 128 נקבל תדר פולסי שעון של 125 קילו הרץ
  // Timer Counter Control Register=4Dh
  TCCR2B |= ((1<<CS22)|(1<<CS20));
  TCCR2B &= ~(1<<CS21);
  TCCR2A &= ~(1<<WGM21) | (1<<WGM20); // אופן עבודה נורמל
```

```

ASSR |= (1<<AS2); // Asynchronous Status Register משתמשים בשעון פנימי ולא חיצוני
TIMSK2 |= ((1<<TOIE2) | (0<<OCIE2B)); // Timer Interrupt MasK בגלישה 2
TCNT2 = 0xff;
}

```

פונקציית הפסיקה של טיימר 2 תיראה :

```

/*
// אם נשים במונה 61 אז הוא יספור 195=256-61 פולסי שעון בין הפסיקות
// מכאן שתדר הריענון יהיה 641 = 125000/195 . ומכאן שכל שורה אחת מ 8
השורות מתרעננת כ 80 פעם בשנייה
*/
ISR(TIMER2_OVF_vect) //פסיקה Timer2
{
    TCNT2 = 61;
    refreshOneLine(); // פונקציה המפעילה כל פעם שורה אחת מתוך ה 8
}

```

כדי להציג תמונה שלמה בת 8 שורות כשבכל שורה יש 8 קבוצות של 3 בתים הקובעים את הצבע לכל אחת מהעמודות יש לפתוח מערך בצורה הבאה : (דוגמה של רישום הספרה 0 בצבע אדום).

```

const unsigned char redNumbers[][8][8][3]= {

```

```

    {
        // 0 red
        {{0,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {0,0,0}},
        {{255,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {255,0,0}},
        {{255,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {255,0,0}},
        {{255,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {255,0,0}},
        {{255,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {255,0,0}},
        {{255,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {0,0,0}, {255,0,0}},
        {{0,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {255,0,0}, {0,0,0}}
    },
    |
    |
    |
};

```

כדי להציג את הספרה 0 בצבע ירוק נרשום בכל המקומות שרשום : {255,0,0} נרשום {0,255,0}

```
// קולורדואינו
/*
led_sda pd7 - Digital Pin 7 led_scl pd6 - Digital Pin 6
led_rst pc2 - a2 led_lat pc1 - a1 led_bank pc0 - a0
col0->col5 = pb0->pb5 - Digital Pin 8->13
col6 - pd3 D.pin 3 col7 - pd4 D.pin 4
anode1 8 anode2 9 anode3 10 anode4 11
anode5 12 anode6 13 anode7 3 anode8 4
*/
byte whiteColor[]={36,63,63}; // צבעים התחלתיים לכיול הלבן בגלל שוני בין הצבעים במטריצה
int row; // מרה על איזו שורה עובדים
byte color=4; // באיזה צבע רוצים להציג
const byte tzvaeem[8][3] = {
// לבן ירוק כחול אדום כחול אדום ירוק אדום ירוק אדום חושך
{0,0,0},{255,0,0},{0,255,0},{0,0,255},{255,255,0},{255,0,255},{0,255,255},{255,255,255}
};

const byte digits[10][8] = {
// 0
{B00111100,
B01000010,
B01000010,
B01000010,
B01000010,
B01000010,
B01000010,
B01000010,
B00111100},
// 1
{B00001000,
B00011000,
B00101000,
B01001000,
B00001000,
B00001000,
B00001000,
B00001000,
B00111110},
// 2
{B00111100,
B01000010,
B00000010,
```


B00000010,
B00111100,
B01000000,
B01000000,
B00111110},

// 3

{B01111100,
B00000010,
B00000010,
B01111100,
B00000010,
B00000010,
B00000010,
B01111100},

// 4

{B01000010,
B01000010,
B01000010,
B01000010,
B00111100,
B00000010,
B00000010,
B00000010},

// 5

{B00111110,
B01000000,
B01000000,
B01000000,
B00111100,
B00000010,
B00000010,
B01111100},

// 6

{B00100000,
B01000000,
B01000000,
B01000000,
B01111100,
B01000010,
B01000010,

```
B00111100},  
// 7  
{B01111100,  
B00000010,  
B00000010,  
B00111100,  
B00000010,  
B00000010,  
B00000010,  
B00000010},  
// 8  
{B00111100,  
B01000010,  
B01000010,  
B00111100,  
B01000010,  
B01000010,  
B01000010,  
B00111100},  
// 9  
{B00111100,  
B01000010,  
B01000010,  
B00111100,  
B00000010,  
B00000010,  
B00000010,  
B00111100},  
};
```

את שאר התכנית נשאיר למשתמש לרשום כיד דמיונו הטובה עליו.