

תצוגת גביש נוזלי (LCD – LIQUID CRYSTAL DISPLAY)

תצוגת הגביש הנוזלי מורכבת מתמיסות אורגניות במצב נוזלי שמשנות את תכונותיהן האופטיות בהשפעת שדה חשמלי.

ישנם כמה סוגים של תצוגות LCD:

1. תאורה טבעית- צג אשר חייב להיות מואר מחוץ על מנת לראות את התצוגה.
2. תאורה מלאכותית- ה- LCD מואר ע"י נורה פנימית קבועה שמחוברת אליו.
3. תאורה אחורית- ה- LCD מואר ע"י משטח פלורסנטי המותקן מאחוריו.

התכונה החשובה ביותר של תצוגת הגביש הנוזלי היא הגמישות הרבה שלה, ניתן לפתל אותה מבלי לשבור את המבנה המסודר שלה.

כיום משתמשים בתג"נ בשעוני יד, מחשבי כיס, מכשירים סלולאריים וכו'. הגביש הנוזלי המקובל כיום הוא גביש נוזלי נימתי. מעין סולם, שכל שלב בו הוא מולקולה אורגנית הבנויה בצורת שרשרת ארוכה. שרשראות אלה מונחות זו לצד זו ומחוברות במבנה הגבישי.

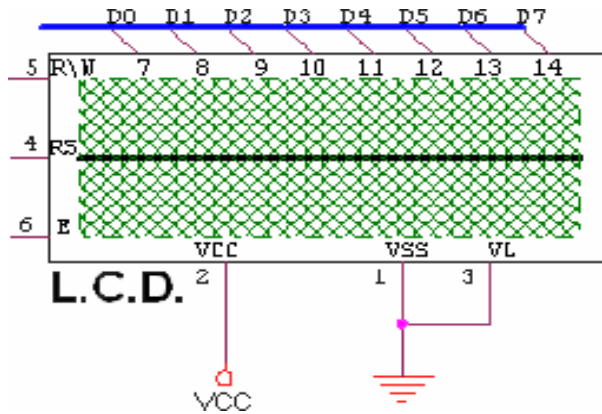
עקרון פעולת הגביש הנוזלי הוא שינוי צבעו מצבע שקוף לשחור בהשפעת מתח חילופין בין קצוות הגביש. כל נקודה בתצוגה מחוברת לשורה ולעמודה משני קצותיה, בעזרת מוליכים המעבירים אליה את האות הנדרש לשינוי הצבע. על מנת ששינוי הצבע של כל נקודה ייעשה בלי תלות בנקודות האחרות, יש לבצע סריקה מחזורית. המסך נסרק שורה אחר שורה כאשר בכל שורה משתנות רק הנקודות הדרושות ליצירת הסימנים הרצויים.

צבע הנקודה (שחור או שקוף) נקבע על פי מצב העמודה המתאימה באותו רגע כאשר העמודה תהיה ב- '1' היא תגרום לשינוי צבע הנקודה. כאשר היא תהיה ב- '0' אותה נקודה לא תראה (תהיה שקופה). כדי שעין האדם לא תבחין בהבהובים, תדר הסריקה יהיה לפחות 30KHZ לשורה, כלומר כל שורה חייבת להיבחר בזמן של $33\mu\text{SEC}$.

מערכת תצוגת ה- LCD כוללת גם מעגל בקר השולט על התצוגה. מעגל זה כולל זיכרון RAM פנימי שאליו כותבים את הנתונים ומעגל פענוח המתרגם את הנתון הבינארי אותו אנו רוצים להציג למספרים בינאריים עבור העמודות השונות של התצוגה. החיבור לרכיב זה נעשה בעזרת 14 כניסות.

מאפיינים:

- צריכת זרם נמוכה ביותר (nA).
- משקל קל.
- אורך חיים גבוה.
- ניתן לייצור בצורות שונות.



תצוגת LCD (שרטוט ותמונה)

תיאור מפורט של רגלי ה-LCD

רגל VSS – רגל אדמת המעגל.

רגל VDD – רגל המחוברת למתח ההזנה של 5V.

רגל RS – רגל זו תפקידה לבחור באחד משני האוגרים.

כאשר מספקים לרגל זו '0' האוגר ישמש כאוגר הוראות וכאשר ברגל זו '1' אוגר זה ישמש כאוגר הנתונים, משמש להפרדה בין תהליך תכנות התצוגה RS=1, ותהליך כתיבה לתצוגה

רגל R/W - תפקיד רגל זו הוא לקבוע האם יכתבו או יקראו נתונים. כאשר מספקים '1' לרגל זו הנתונים המאוכסנים ב RAM של התג"נ יקראו. כאשר מספקים '0', יכתבו נתונים לRAM של התג"נ.

רגל E – רגל אפשרור הרכיב. ב-'0' הרכיב לא יאופשר. ב-'1' הרכיב יאופשר.

רגליים DB₀-DB₇ – זהו פס נתונים של התצוגה ברוב של 8 סיביות. תפקיד פס הנתונים הוא העברת נתונים או פקודות אל התג"נ מהמיקרו בקר ולהיפך.

Instruction Register – אוגר הוראות

אוגר זה תפקידו לאכסן הוראות הקובעות את אופן הפעולה של התג"נ.

הוראות אלה יכולות להיות: הזזת סמן, ניקוי התצוגה, כיבוי תצוגה, הדלקת התצוגה וכו'.

Data Register – אוגר הנתונים

אוגר זה הינו אוגר זמני אשר בו מאכסנים את כל הנתונים שאותם רוצים להציג על התצוגה.

כדי להציג נתון שולחים את התו של ב ASCII.

פעולה	R/W	RS
פניה לאוגר ההוראות, כתיבת הוראה ל-LCD	0	0
פניה לאוגר הנתונים, כתיבת נתונים ל-LCD	0	1
קריאה של מונה הכתובות ודגל העיסוק מה-LCD	1	0
פניה לאוגר הנתונים, קריאת נתונים מה-LCD	1	1

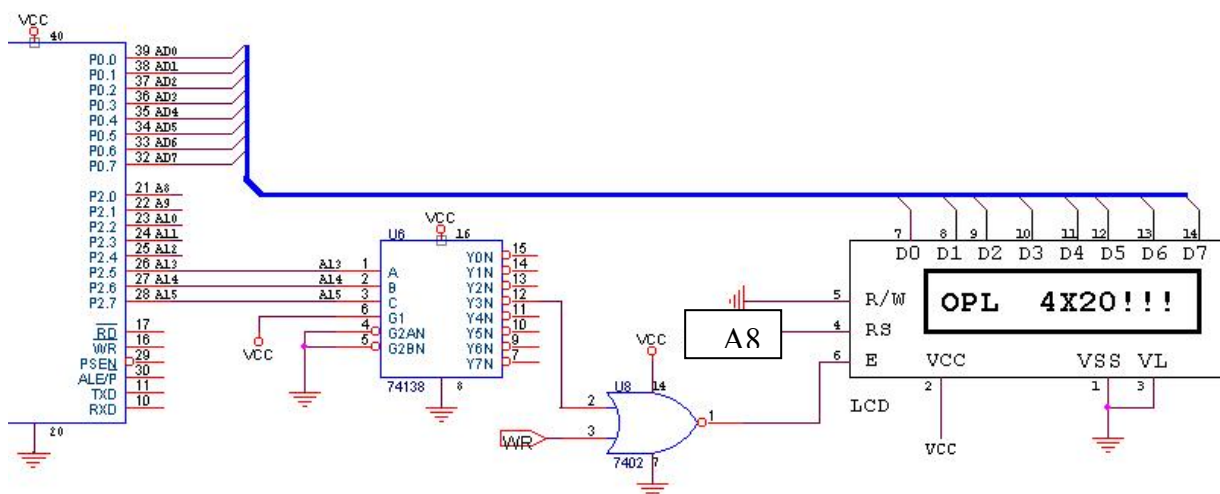
טבלת מצבי האוגרים והקריאה/כתיבה ב-LCD בתלות ברגליים RS ו-R/W

טבלת כתובות של התווים התצוגה

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
שורה 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93
שורה 2	0C0	0C1	0C2	0C3	0C4	0C5	0C6	0C7	0C8	0C9	0CA	0CB	0CC	0CD	0CE	0CF	0D0	0D1	0D2	0D3
שורה 3	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F	A0	A1	A2	A3	A4	A5	A6	A7
שורה 4	0D4	0D5	0D6	0D7	0D8	0D9	0DA	0DB	0DC	0DD	0DE	0DF	0E0	0E1	0E2	0E3	0E4	0E5	0E6	

אופן ההפעלה של התצוגה

על מנת לעבוד עם הגביש יש לתכנת את אופן העבודה RS=0. למשל: אחרי כל שליחת נתון הסמן יזוז ימינה, הסמן יהבהב כולא, וכו'. תהליך התכנות נעשה רק פעם אחת בתחילת התוכנית. כעת ניתן לשלוח תווים בקוד ASCII ותצוגה והם יופיעו אחד אחרי השני. בד"כ התווים נשלחים במחרוזת כפי שמוצג בתכניות לדוגמא.



פונקציות להפעלת התצוגה

1. תיכנות צג ה-LCD
2. השהייה בין כתיבה לכתיבה על פי דרישות הייצור.
3. ניקוי התצוגה (מחיקת מסך) ושליחת הסמן לצד שמאל שורה ראשונה.
4. קביעת מיקום הסמן (מס' שורה ומס' תו).
5. שליחת מחרוזת תווים אל התצוגה.
6. שליחת תו יחיד אל התצוגה.

//=====CONFIG THE LCD=====

```

void lcd_config() //Initialization of The LCD
{
    delay_lcd();
    XBYTE[LCD_REG]= 0x38; //8bit,2 line,5*7 dot
    delay_lcd();
    XBYTE[LCD_REG]=0x0e; // display on ,cursor on,cursor blink
    delay_lcd();
    XBYTE[LCD_REG]= 0x1; //clear_display
    delay_lcd();
    XBYTE[LCD_REG]= 0x6; // increment cursor,no display shift
}
    
```

1

```
//=====LCD=Delay=====
void delay_lcd()
{
int timer=200;
while(timer--);
}
```

2

```
//=====LCD CLR=====
void lcd_clr()
{
delay_lcd();
XBYTE[LCD_REG]=1;
}
```

3

```
//=====LCD LINE NUMBER=====
void lcd_line(char line_number,char p)
{
switch(line_number)
{
case 1:
delay_lcd();
XBYTE[LCD_REG]=(0x80+p);
break;
case 2:
delay_lcd();
XBYTE[LCD_REG]=(0xc0+p);
break;
}
}
```

4

```
//=====DISPLAY A STRING ON THE LCD=====
void lcd_string(char str[])
{
int i=0;
while(str[i])
{
delay_lcd();
XBYTE[LCD_DATA]=str[i++];
}
}
```

5

```
//=====DISPLAY THE CHARACTER ON THE LCD=====
void lcd_display(char character)
{
delay_lcd();
XBYTE[LCD_DATA]=character ;//Send The character to the LCD
}
```

6

