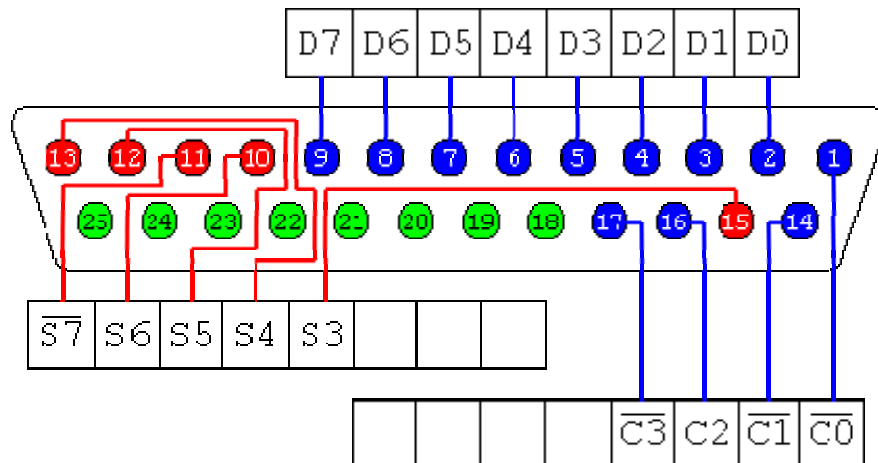


הסברים לפרויקט

פורט המקבילי של המחשב - יציאת מדפסת

היציאה המקבילית של המחשב כוללת 3 פורטים:

- יציאת DATA: בגודל 8 סיביות בכתובת 0x0378
- כניסת STATUS: בגודל 5 סיביות בכתובת 0x0379
- יציאת CONTROL: בגודל 4 סיביות בכתובת 0x037a



הערה: חלק מהסיביות הפוכות לוגית ויש להתחשב בכך בתוכנה.

כדי לגשת לפורטים בשפת C משתמשים בפקודות הבאות:

- שליחת מידע (Data_out) לפורט יציאה בכתובת 0x0378

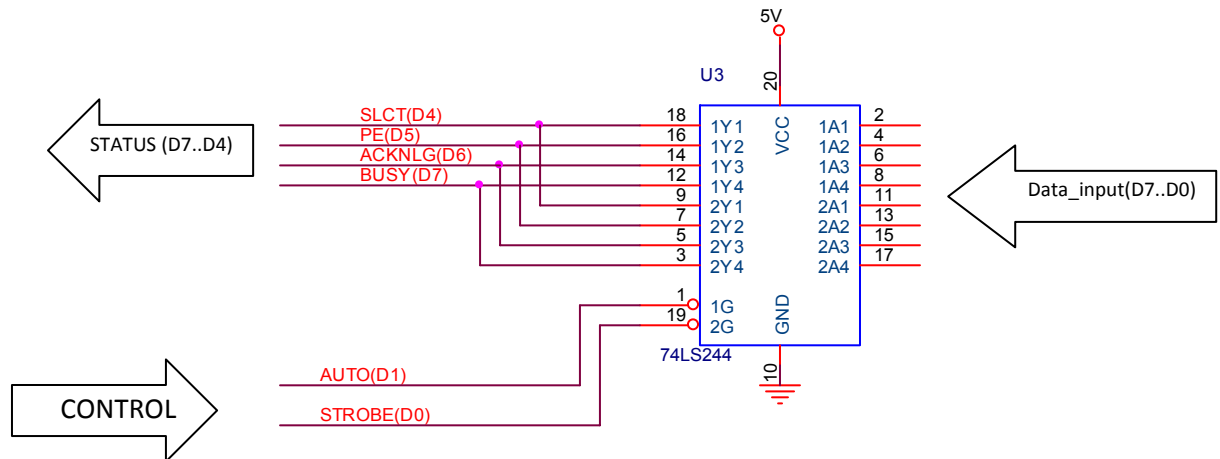
```
outportb(0x0378,Data_out);
```

- קבלת מידע מפורט כניסה בכתובת 0x0379 למשתנה Data_in

```
Data_in=inportb(0x0379);
```

קליטת נתון בגודל 8 סיביות מיציאת המחשב

כיוון שיציאת המחשב כוללת כניסה בגודל 5 סיביות בלבד, יש צורך לקרוא פעמיים מידע בגודל 4 סיביות ולבצע פעולות לוגיות לקבלת מידע של 8 סיביות. לשם כך משתמשים ברכיב 74LS244 לפי החיבור הבא:



הרכיב מקבל בכניסות A מידע של 8 סיביות ומעביר 4 סיביות בלבד בהתאם לבקרה G

\bar{C}_1 - CONTROL(D1)	\bar{C}_0 - CONTROL(D0)	STATUS(D7..D4)	
$\bar{C}_1 = 0$	$\bar{C}_0 = 1$	Data_input(D7..D4)	העברת 4 סיביות עליונות
$\bar{C}_1 = 1$	$\bar{C}_0 = 0$	Data_input(D3..D0)	העברת 4 סיביות תחתונות

הפונקציה in_port דואגת לשלוח את בקרה לממיר A/D להתחלת המרה ואותות בקרה לקליטת 4 סיביות עליונות ותחתונות ולהכניס אותם למשתנה in_s.

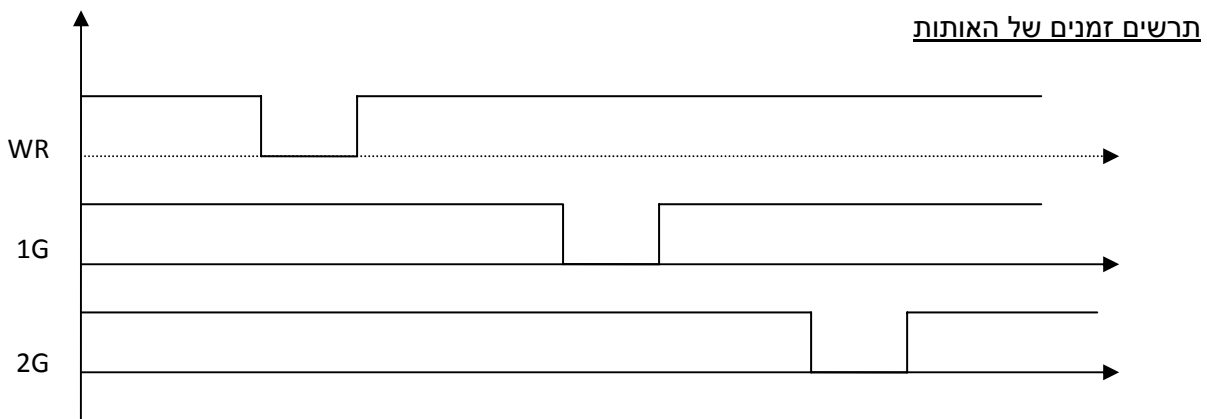
```

unsigned char in_port(void)
{
    unsigned char in_s;
    {
        outportb(CONTROL, 0x04);
        delay(1);
        outportb(CONTROL, 0x00); // שליחת 0 לוגי לממיר להתחלת המרה
        delay(1);
        outportb(CONTROL, 0x04);
        delay(10);
        outportb(CONTROL, 0x06); // 74LS244 1G בקרה לרוגל
        delay(1);
        in_s = (( inportb(STATUS)&0xF0)^0x80)>>4; //A/D ADC0804 עליונות
        delay(1);
        outportb(CONTROL, 0x05); //74LS244 2G בקרה לרוגל
        delay(1);
        in_s = ((inportb(STATUS)&0xF0)^0x80)|in_s; //A/D ADC0804 תחתונות
        outportb(CONTROL, 0x04);
        return in_s;
    }
}

```

האותות לממיר A/D ADC0804 ולרכיב 74LS244 buffer

מידע ל- CONTROL	$C_3C_2C_1C_0$	$WR_{ADC0804} = C_2$	$1G_{74244} = \overline{C_1}$	$2G_{74244} = \overline{C_0}$
0X04	0 1 0 0	1	1	1
0X00	0 0 0 0	0	1	1
0X06	0 1 1 0	1	0	1
0X05	0 1 0 1	1	1	0



הסבר קליטת 4 סיביות תחתונות

```
in_s=(( inportb(STATUS)&0xF0)^0x80)>>4;
```

מיסוך המידע עם הערך $0xf0=1111000b$, היפוך סיבית D7 (\bar{S}_7 של ה-STATUS) בעזרת פעולת XOR: 0x80 והזזה ימינה של 4 הסיביות: $>>4$

הסבר קליטת 4 סיביות עליונות

```
in_s=((inportb(STATUS)&0xF0)^0x80)|in_s;
```

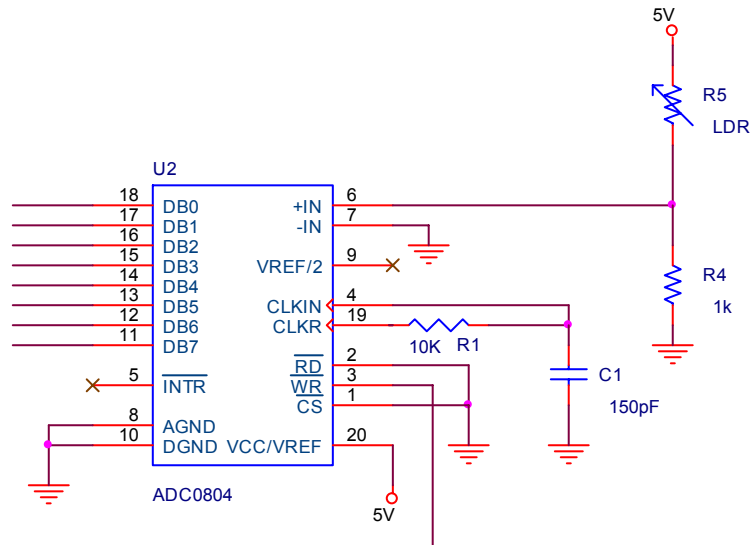
מיסוך המידע עם הערך $0xf0=1111000b$, היפוך סיבית D7 (\bar{S}_7 של ה-STATUS) בעזרת פעולת XOR: 0x80 ופעולת OR עם הערך הקודם (4 סיביות תחתונות) לאחר שתי הפקודות הנ"ל in_s יקבל מידע של 8 סיביות.

שליחת נתון מהמחשב

מידע המקבילי של 8 סיביות נשלח לפורט DATA בעזרת הפונקציה הבאה:

```
void out_port(unsigned char value)
{
    outportb(DATA,value);
}
```

הסבר פעולת המרת אור לערך דיגיטלי



רכיב ה-LDR הוא נגד המשנה את התנגדותו בהתאם לאור, ככל שעוצמת ההארה יותר גבוהה ההתנגדות יותר נמוכה והמתח שמגיע לכניסה IN גדול יותר.

המתח הנ"ל מומר מערך אנלוגי לדיגיטלי (DB7..DB0) ע"י הרכיב A/D.

הקשר בין הערך האנלוגי והדיגיטלי הוא לפי הנוסחה הבאה:

$$data = \frac{V_{IN}}{V_{REF} / 2^8} = \frac{256 \cdot V_{IN}}{5}$$

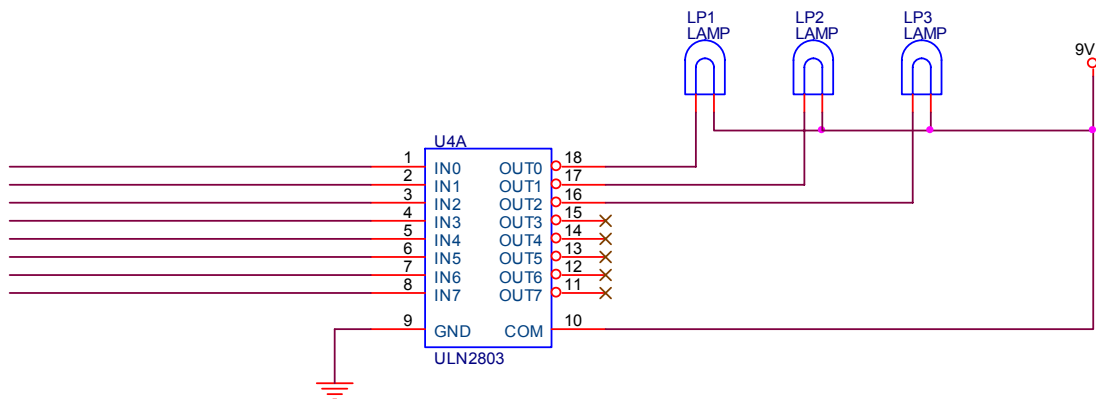
לדוגמא: עבור $V_{IN} = 2v$ נקבל מידע עשרוני 102 או 0x66.

תפקיד R1 ו-C1 הוא לקבוע תדר תנודות לפעולת הממיר העובד בשיטת קירוב מתקדם.

פעולת הממיר היא לפי השלבים הבאים:

1. שליחת 0 לוגי מהמחשב לרגל WR להתחלת המרה.
2. השהיה עד לסיום ההמרה של הרכיב (ההשהיה נקבעת ע"י התוכנה).
3. קריאת המידע הדיגיטלי ממוצא הממיר.

הפעלת הנורות



3 הנורות מחוברות דרך דוחף ULN2803 המורכב מ- 8 טרנזיסטורי דרליגנטון שתפקידו לספק זרם מתאים לכל נורה (הרכיב נחוץ כי למחשב אין יכולת לספק זרם לנורות).

כאשר מגיע 1 לוגי לכניסה IN טרניסטור הדרליגנטון בתוך הרכיב נכנס למצב רוויה וביציאה OUT נקבל מתח של בערך 1V ולכן המתח שיסופק לנורה יהיה : $9v-1v=8v$ והנורה תדלק. במצב שהכניסה 0 לוגי הדרליגנטון יכנס למצב קטעון והזרם במוצא הרכיב יהיה 0A והנורה תכבה.