

שאלות כלליות ושאלות מבחניות חיצוניות על המיקרו C8051F380

- הערות :** 1. חלק מהשאלות נלקח ממבחנים משנים קודמות (בסוגריים רשום השנה של המבחן). באתר יש פתרונות לשאלות אלו אבל הפתרון מתאים למיקרו 8051 . בחלק מהפתרונות יש לבצע התאמה למיקרו C8051F380 .
2. לשאלות 72 עד 82 מוצמד פתרון . יש לפתור אותן לבד ואחר כך להשוות את הפתרון שלכם לפתרון כאן.
3. בחלק מהאיורים לשאלות משורטט המיקרו בקר 8051 ואפשר להתייחס אליו כ C8051F380.
4. עבור כל התרגילים שבהם יש לרשום תוכניות יש להיעזר בפקודות הבאות :

שמירה על צורה אחידה למתן שמות אוגרים וטיפוסים של משתנים, יש להצהיר על ספריות הבאות:

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
```

פונקצייה לאתחול המעבד:

```
void Init_Device(void);
```

אין צורך לכתוב את מילות האתחול לאוגרים לביטול ה-watchdog וקביעת תדר הבקר, אלא להניח שהן קיימות כחלק ממימוש הקוד בפונקצייה Init_Device .

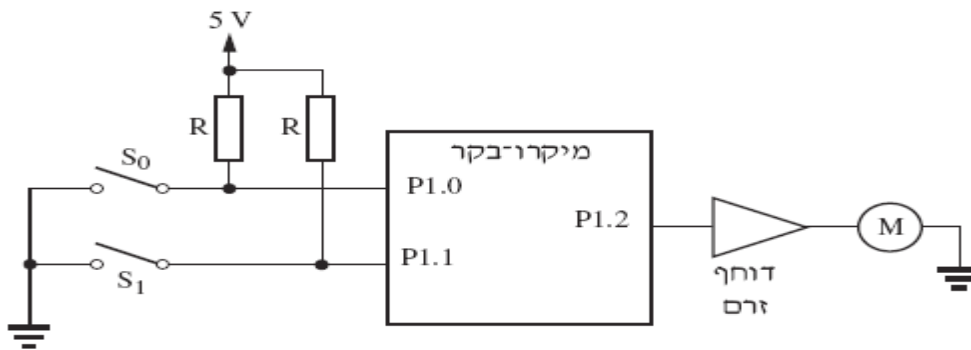
יש להניח שפונקציות השהייה במילי-שניות ומיקרו-שניות קיימות ואין צורך להצהיר עליהן או לממש אותן.

```
void delay_ms(unsigned int ms);
void delay_us(unsigned int us);
```

1. ערוך השוואה בין כל הרכיבים במשפחת ה MCS 51 (רכיבים המסתיימים ב 1 ב 2 וב 3). שים דגש על: א. זיכרון תכנית פנימי ב. זיכרון ה DIRECT ג. כמות הטיימרים. ד. כמות מקורות הפסיקה
2. מה ההבדל בין מיקרו פרוססור ומיקרו בקר ?
3. אילו יחידות נמצאות בתוך ה 8051 ומה תפקידן ? (הכוונה לטיימרים, פסיקות וכו').
4. תאר את תהליך הבאת הפקודה ב 8051 כולל פסים ורגלי חומרה המשתתפים בתהליך.
5. מהם הבנקים הקיימים ב 8051 ?
6. היכן נמצאת המחסנית ב 8051. איזה רגיסטר מציין את מיקומה?
7. מהי פסיקה? אפשר פסיקה או מיסוך? עדיפות ?
8. כמה ואילו סוגי פסיקה יש למיקרו C8051F380 ?
9. אילו 4 רגיסטרים נוטלים חלק בזמן פסיקות ? פרט תפקיד כל ביט ברגיסטרים אלו.
10. רשום פקודות בשפת C51 שיאפשרו פסיקות חיצוניות בלבד .
11. הסבר את ההבדל בין מצב COUNTER לTIMER .

12. אילו רגיסטרים לוקחים חלק בשליטה על הטיימרים ? הסבר כל ביט ברגיסטרים.
13. אילו אופני עבודה קיימים לכל טיימר ? צייר סכמה מלבנית של טיימר באופן 0 ו 1 .
14. חשב כל כמה זמן נקבל פסיקה באופן עבודה 1 של אחד הטיימרים. תדר הפולסים הנכנסים לספירה 12 מגה-הרץ.
15. עבור גביש של 12 מגה-הרץ. תכנן מערכת לספירת מאית שנייה.
16. מהו ההבדל בין BAUD לbps (סל"ש) בתקשורת טורית.
17. צייר כיצד נראה בתקשורת טורית שידור התו 4BH עם ביט 1 של התחלה וסיום ללא בדיקת זוגיות.
18. איזה רגיסטר שולט על התקשורת הטורית ? הסבר כל ביט.
19. מה הקשר של רגיסטר PCON לתקשורת טורית ?
20. עבור תדר גביש של 11.0592 מגה-הרץ. תכנן תקשורת בקצב של 19200 BAUD.
21. העבר את תוכן הבלוק 40H ועד 4FH לכתובת 60H ועד 6FH (באזור ה-DIRECT).
22. החלף בין תוכן הבלוקים 50H ועד 5FH עם 60H ועד 6FH באזור ה-DIRECT.
23. מלא את בלוק הכתובות מ 1000H ועד 10FFH בתוכן 0,1,2... עד FF בהתאמה.
24. העבר את הבלוק 1000H ועד 1200H ל 2000H ועד 2200H.
25. החלף בין הבלוקים 1000H ועד 1200H עם 2000H ועד 2200H.
26. במועדון מפורסם יש דלת כניסה ויציאה נפרדות. בכל דלת גלאי שכאשר נכנס (או יוצא) אדם הגלאי נותן פולס (ירידה לאפס וחזרה ל1): רשום תוכנית שתמצא כמה אנשים נכנסו למועדון וכמה אנשים נמצאים בו כרגע. (חבר כל יציאת גלאי לאחת מכניסות הרכיב). א. עם שאילתה. ב. עם פסיקות.
27. את השאלה הקודמת פתור בעזרת ה COUNTERS של המיקרו בקר .
28. בפורט 1 של ה8051 מחוברים 3 מפסקים. לכל מפסק מחובר נגד PULL UP שבמצב רגיל נותן "1" לוגי. כאשר נלחץ על אחד המפסקים נקבל אפס בכניסה המתאימה. המפסקים מחוברים ל P1.0 P1.1 P1.2 . רשום תוכנית שתדליק 8 לדים הנמצאים בפורט 2 אם לפחות שניים מתוך שלושת המפסקים לחוצים (שים לב – מפסק לחוץ נותן '0' והלדים מחוברים באנודה משותפת).
29. באיור לשאלה מתואר מיקרו-בקר שאליו מחוברים מתגים S0 ו-S1 ומנוע M. כתוב קטע תכנית בשפת C של המיקרו-בקר, שיפעיל את המנוע בהתאם לתנאים שבטבלה:

מנוע	S0	S1
אינו מופעל	פתוח	פתוח
מופעל	פתוח	סגור
מופעל	סגור	פתוח
אינו מופעל	סגור	סגור

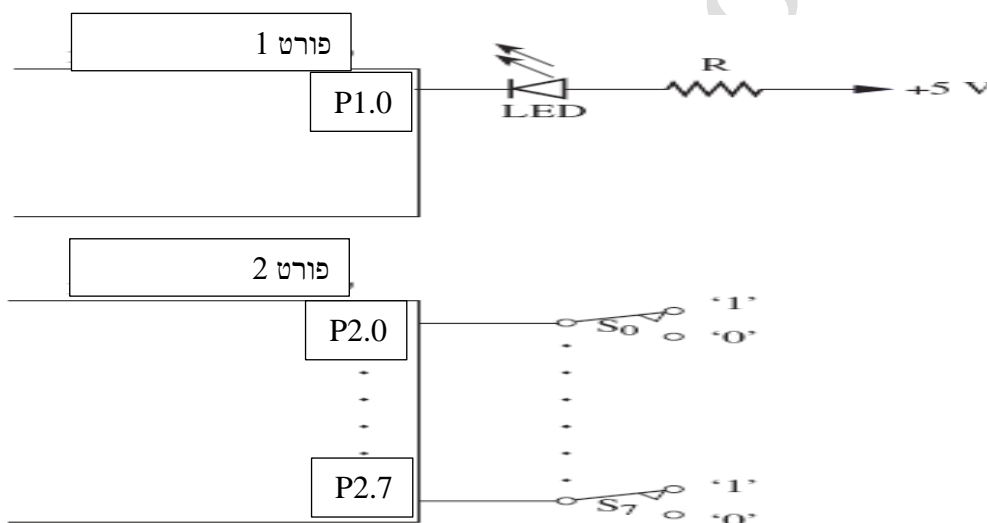


30. כתוב קטע תכנית בשפת C אשר יוסיף 10 לערכו של המשתנה X אם מתקיימים כל התנאים הבאים:

- ערכו של המשתנה SUM שווה לערכו של המשתנה UP.
- ערכו של המשתנה X גדול מ-3.
- ערכו של המשתנה X קטן או שווה ל-3.

אם אף אחד מהתנאים אינו מתקיים, קטע התוכנית יפחית 5 מערכו של X.

31. באיור לשאלה מתוארים נורית LED, המחוברת אל P1.0, ושמונה מתגים, המחוברים פורט 2. של המיקרו-בקר C8051F380.



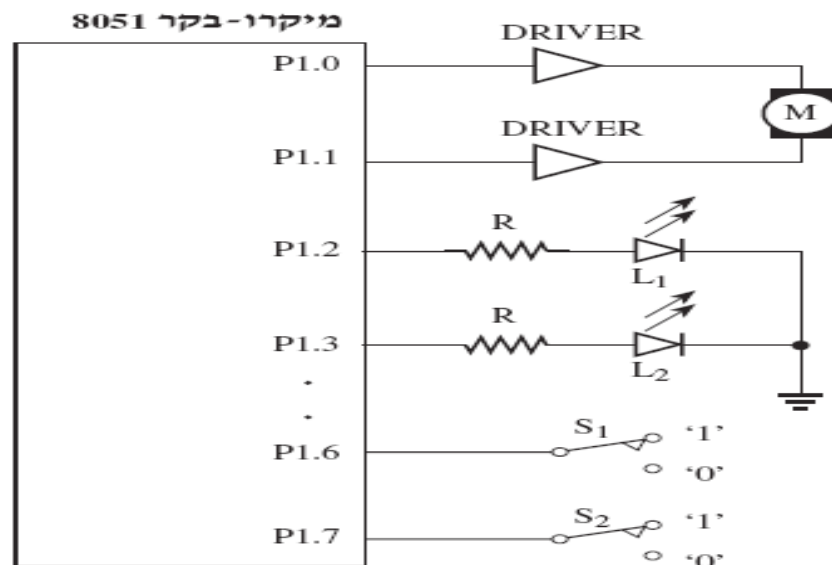
כתוב תוכנית בשפת C51, שתגרום לנורית ה-LED להבהב מספר פעמים, בהתאם לנתון הנקלט דרך שמונת המתגים

(S7,...,S0). למשל, אם הנתון הנקלט הוא 10H (בעשרוני) נורית ה-LED תבהב 16 פעמים בקצב של 0.5 שנייה.

הערה: נתונה פונקציית השהייה הנקראת `delay_ms(value)` שמבצעת השהייה של אלפיות שנייה למספר שיש ב-`value`.

השתמש בפונקציה הזו לקביעת משך זמן ההדלקה ומשך זמן הכיבוי בכל הבהוב של נורית ה-LED.

32. באיור לשאלה מתוארים מנוע M, שתי נוריות LED L1 ו L2 ושני מתגים S1 ו S2, המחוברים למיקרו-בקר 8051.



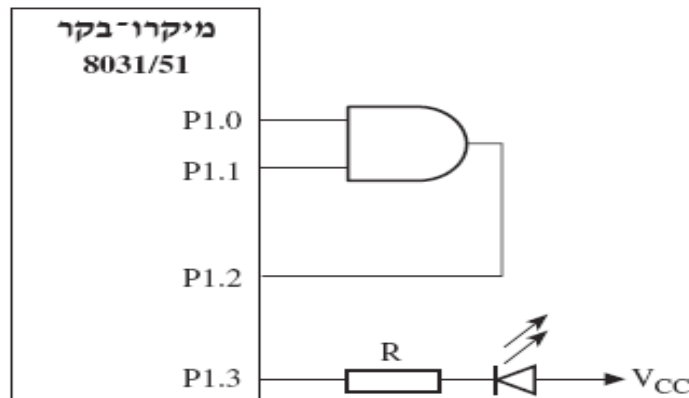
פעולת המנוע M נקבעת לפי מצב ההדקים P1.0 ו-P1.1, כמפורט בטבלה שלהלן:

M	P1.1	P1.0
מנוע לא מסתובב	0	0
המנוע מסתובב בכיוון השעון	0	1
המנוע מסתובב נגד כיוון השעון	1	0
המנוע לא מסתובב	1	1

כתוב תת-שגרה בשפת גם בשפת ASM-51 וגם בשפת ב51, שתבדוק את מצב המתגים S1 ו S2 תפעיל את המנוע M ותדליק את נוריות הLED - L1 ו L2 על-פי הטבלה שלהלן:

M	L2	L1	S2	S1
המנוע לא מסתובב	OFF	OFF	0	0
המנוע מסתובב בכיוון השעון	OFF	ON	0	1
המנוע מסתובב נגד כיוון השעון	ON	OFF	1	0
המנוע לא מסתובב	ON	ON	1	1

33. באיור לשאלה משמש המיקרו-בקר אמצעי לבדיקה של שער AND. אם השער תקין-נורית ה-LED דולקת ואם הוא לקוי-נורית ה-LED כבויה. רשום פונקציה בשפת C51 שתבצע בדיקת השער.



34. נתון בלוק של מספרים אשר כתובת ההתחלה שלו היא 20H באזור הנתונים הפנימי ואורכו 10H בתים. כל נתון בבלוק מאוחסן בבית אחד. כתוב תוכנית שתבצע את הפעולות האלה:

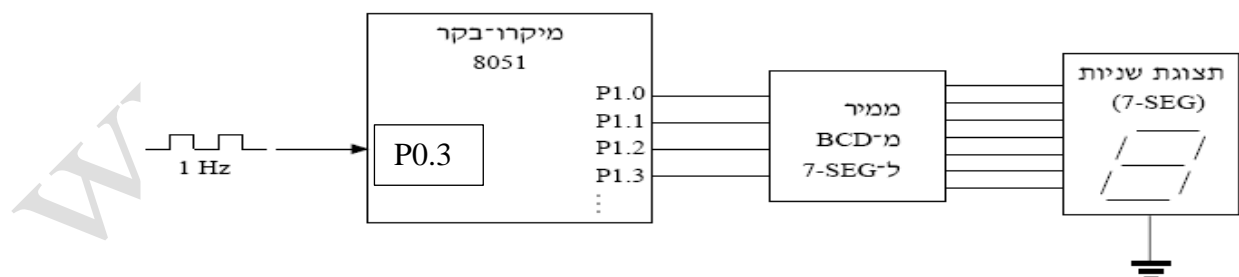
- תמנה את כמות המספרים האי-זוגיים, שהם גם שליליים, בבלוק נתונים זה.
- תציב את תוצאת המנייה בתא 30H.

35. כתוב תוכנית שתבצע את הפעולות האלה:

- תקרא נתון שגודלו ארבע סיביות מתא הזיכרון ב XRAM שכתובתו 200H (ידוע שערך הנתון נע בין 00H ל-0FH).
- תבצע המרה של הנתון למספר עשרוני בקוד BCD ותציב את המספר המתקבל בתא הזיכרון שכתובתו 201H.

36. כתוב תוכנית שתגרום למערכת לבדוק את ערכה של הסיבית בהדק P1.0. אם ערכה של סיבית זו הוא '1' יש להוציא בהדק P1.7 דופק יחיד שרוחבו צריך להיות בין כ 50 מיקרו-שניות. אם ערך הסיבית זו הוא '0' יש לאפס את ערכו של הדק P1.7.

37. באיור לשאלה מתוארת מערכת המשמשת כשעון שניות בתחום שבין 0 עד 9 שניות. להדק $\overline{INT0}$ של המיקרו-בקר חובר שעון שתדרו 1Hz.



- כתוב פקודות שיאפשרו קבלת פסיקה חיצונית במבוא $\overline{INT0}$ בלבד בכל ירידה של דופק-שעון.
- כתוב תוכנית שתציג בתצוגת השניות (7-SEG) ספרה מ-0 עד 9 בסדר עולה, בירידה של כל דופק-שעון. הספרות תוצגנה באופן מחזורי (בדופק אחרי 9 התצוגה מתאפסת ומתחילה ספירה חדשה).

.38

א. הסבר כיצד קורא המיקרו-בקר 8051 משני מרחבי כתובות של 64kB (גם מזיכרון תוכניתוגם מזיכרון נתונים).

ב. יש לכתוב תוכנית אשר:

- תאפשר פסיקות חיצוניות INT0 ו INT1 המופעלות בדרכון רמה גבוהה (1).
- תקבע עדיפות גבוהה יותר לפסיקה INT1.

.39

טבלה 1 מציינת את ערכי תאי הזיכרון 101H – 107H ב-RAM החיצוני (ב C8051F380 4 קילו בתים ראשונים של ה XRAM נמצאים בתוך הרכיב) . לאחר הפעלת הפונקציה shinui ערכי התאים משתנים לערכים המופיעים בטבלה 2 - (ספרות ההקסה מתחלפות). א. כתוב את הפונקציה shinui .

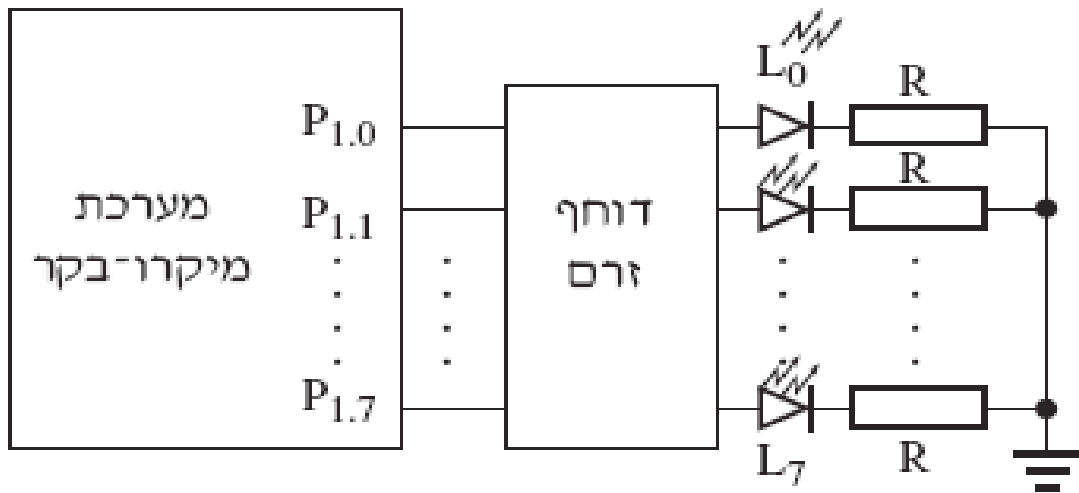
טבלה 1

כתובת	תוכן
1007H	27H
1006H	26H
1005H	25H
1004H	24H
1003H	23H
1002H	22H
1001H	21H

טבלה 2

כתובת	תוכן
1007H	72H
1006H	62H
1005H	52H
1004H	42H
1003H	32H
1002H	22H
1001H	12H

.40 באיור לשאלה מתוארת מערכת מיקרו-בקר המבוססת על המיקרו-בקר C8051F380:

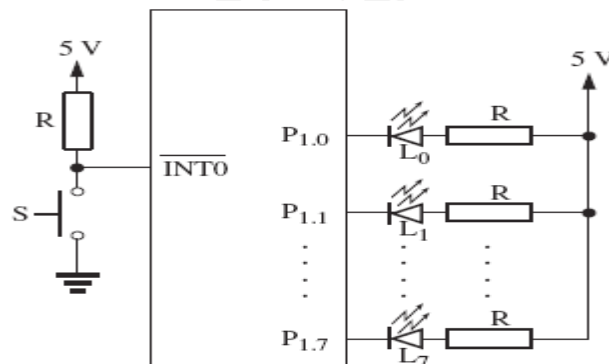


בטבלה שלהלן מצוינים הנתונים המאוחסנים בתאי זיכרון ה-RAM החיצוני שכתובותיהם 100H – 103H:

כתובת התא	100H	101H	102H	103H
ערך	10H	18H	23H	40H

רשום תוכנית שמסכמת את תוכן 4 הכתובות ומוציאה ללדים את התוצאה. אילו לדים דולקים ואילו כבויים בסיום התוכנית?

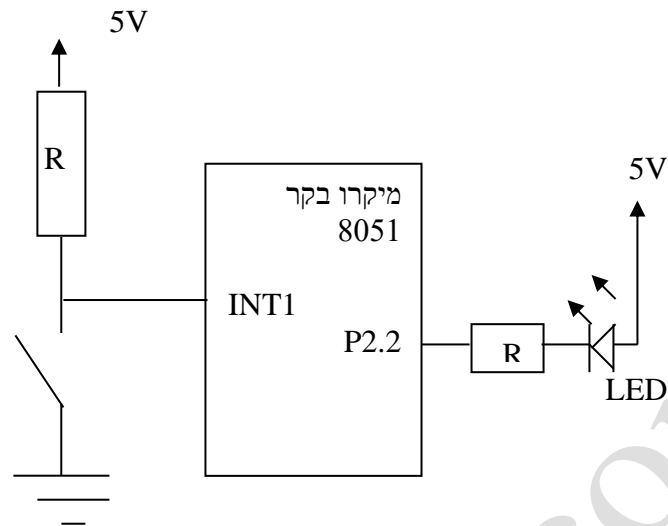
41. באיור לשאלה מתוארת מערכת מיקרו-בקר המבוססת על המיקרו-מעבד C8051F380.



- כתוב תת-שגרה המאפשרת שימוש בפסיקה $\overline{int0}$.
- כתוב תת-שגרה המגיבה לפסיקה $\overline{int0}$. בכל פסיקה תתבצע הזזה אחת של הנורית הדולקת. כלומר, בפסיקה הראשונה תידלק הנורית L_1 ושאר הנוריות תהיינה כבויים. בפסיקה השנייה תידלק הנורית L_2 ושאר הנוריות תהיינה כבויים, וכן הלאה עד הפסיקה השביעית. בפסיקה השמינית תידלק הנורית L_0 ושאר הנוריות תהיינה כבויים, וחוזר חלילה.

42. ממבחן בשנת 2010

באיור לשאלה נתונה נורית LED המחוברת להדק P2.2 של המיקרו בקר 8051, ומפסק המחובר להדק הפסיקה INT1 שלו.



להלן תכנית בשפת C של המיקרו 8051.

```

1.  #include <8051.h>
2.  void init(void)
3.  {
4.      TCON=0x04;
5.      IE = 0x84;
6.  }
7.  void INT1_ISR (void) interrupt 2
8.  {
9.      unsigned int i;
10.     P2_2 =0 ;
11.     for(i=0;i<64000;i++);
12.     P2_2=1;
13. }
14. void main (void)
15. {
16.     init( );
17.     P2_2=1;
18.     while(1);
19. }
```

- א. הסבר את הפקודות שבשורות 7, 9, 12, 16, 18. ב. הסבר את משמעות השורות 4 ו 5 בתוכנית. ג. הסבר מה מבצעת התוכנית.

שאלה 43 (ממבחן טכנאים משה"ח 2012)

להלן תכנית בשפת C של המיקרו-בקר 8051 העושה שימוש ב-UART.

```

1.      #include<8051.h>
2.      void main(void)
3.      {
4.          TMOD=0x20;
5.          SCON=0x40;
6.          TH1=0xFD;
7.          TL1=0xFD;
8.          TR1=1;
9.          TI=0;
10.         while(1)
11.         {
12.             SBUF=0x38;
13.             while(!TI);
14.             TI=0;
15.         }
16.     }

```

- א. הסבר את ההוראות שבשורות 9, 10 ו-13.
- ב. הסבר את המשמעות של ההוראות שבשורות 4 + 5 בקוד התכנית. היעזר בדף הנוסחאות של המיקרו-בקר לצורך ניתוח מילות הבקרה.
- ג. מה קובעים תכני האוגרים TH1 ו-TL1?
- ד. סרטט מחזור אחד של האות היוצא מן ההדק TxD, כפונקציה של הזמן. ציין בסרטוטך את מתח האות (אינך נדרש לציין את זמני המעבר).

שאלה 44

- א. כתוב תכנית בשפת C של המיקרו בקר 8051 שתבצע את הפעולות הבאות: 1. תקלוט מספר מכתובת 100h בזיכרון הנתונים החיצוני (ברכיב C8051F380 יש 4 קילו בתים ברכיב עצמו). 2. תספור את כמות ה'1' בנתון שנקלט. 3. תשמור את מספר ה'1' בכתובת 101h בזיכרון הנתונים החיצוני.
- ב. נתון לך מערך (בלוק) של 0x10 בתים מטיפוס U8 (תווי לא מסומן) החל מכתובת 0x40 בזיכרון הנתונים הפנימי. התוכנית תספור את כמות ה'0' בכל ביט ותשמור את התוצאה במערך הנמצא החל מכתובת 0x60 בזיכרון הנתונים הפנימי.

שאלה 45

כתוב תכנית בשפת C של המיקרו-בקר 8051, שתקלוט **עשרה** נתונים מהמפתח PI באמצעות פסיקה חיצונית, ותאחסן אותם בזיכרון הפנימי של המיקרו-בקר. בכל פעם שתקבל פסיקה חיצונית בהדק INT1 – ייקלט נתון **אחד** (בגודל בית) מהמפתח PI. הנתונים שייקלטו יאוחסנו בזה אחר זה בבלוק נתונים שגודלו 10 בתים, המתחיל בכתובת 40H בזיכרון הפנימי של המיקרו-בקר.

על התכנית לעבוד לפי האלגוריתם הבא:

1. התכנית הראשית תאפשר פסיקה חיצונית דרך INT1 ותמתין לפסיקות.
2. תת-שגרה תופעל בכל פעם שהדק הפסיקה INT1 יורד מ-'1' ל-'0'.
על תת-השגרה לבצע את הפעולות שלהלן:
 - 2.1 לחסום באופן זמני קבלת פסיקות נוספות.
 - 2.2 לקלוט נתון מהמפתח PI.
 - 2.3 לשמור את הנתון בבלוק הנתונים.
 - 2.4 אם בלוק הנתונים מלא – תופסק קליטת נתונים נוספים על-ידי חסימת הפסיקות. אם בלוק הנתונים אינו מלא – יאופשרו פסיקות נוספות.

שאלה 46

כתוב תוכנית שתבצע את הדברים הבאים:

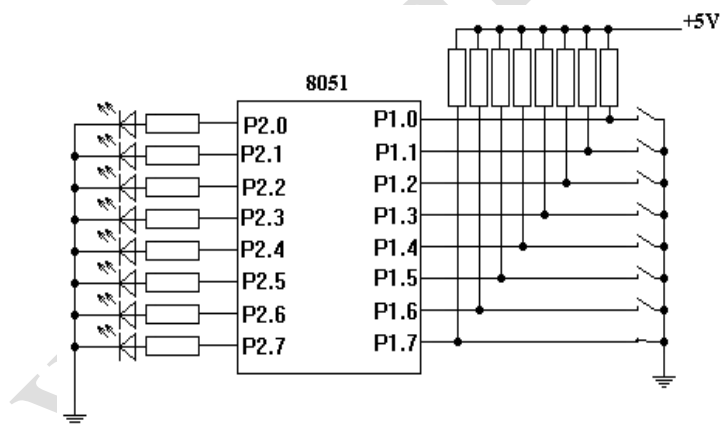
1. תקלוט נתון השמור בכתובת 40H בזיכרון הפנימי של המיקרו-בקר.
2. תספור את מספר הסיביות שערכן '0' בנתון שנקלט.
3. תשמור את מספר הסיביות שערכן '0' בכתובת 41H בזיכרון הפנימי של המיקרו-בקר.

שאלה 47

מערכת מיקרו מחשב המבוססת על מעבד מסוג 8051 כוללת ממשק קלט/פלט כדלקמן: 8 מפסקים, 8 נוריות מסוג LED ו-2 לחצנים המחוברים לקווי פסיקה.

- הלחצנים מיועדים להפעלת פסיקות (במעבר מ-1 ל-0) באופן הבא:
- לחצן PB1 גורם לפסיקה המיועדת לקריאת מצב המפסקים והפעלת הנוריות בהתאם: הדלקת LED ("0") עבור מפסק סגור ("0") ולהיפך.
 - לחצן PB2 גורם לפסיקה המיועדת להפיכת מצב הנוריות ללא תלות במצב המפסקים.
 - לפסיקה המתקבלת מלחצן PB1 עדיפות גבוהה יותר.

- א. תאר באופן סכמתי את מרכיבי המערכת והחיבורים ביניהם.
 - סמן את כל קווי הקלט/פלט וכיווניהם.
 - ציין במפורש לאיזה PORT במעבד מחוברים קווי הפסיקה.
- ב. כתוב שתי רוטינות שירות עבור הפסיקות המתקבלות מהלחצנים.
 - רשום במפורש את מילות הבקרה וערכי האתחול והתכנות ב-HEX בלבד!
 - ציין את כל הכתובות הרלוונטיות לכתיבת התכנית.
 - הסבר בקצרה כל שורה בתכנית.
- ג. הסבר כיצד אפשר למנוע תופעה של ריטוט הלחצנים (Debouncing) באמצעות תוכנה.

שאלה 48 נתון המעגל הבא והתוכנית בשפת C שלה.

```
void main (void) {
    unsigned char x, i, switch[]={0,1,2,3,4,5,6,7};
    unsigned char LEDs[]={0x0,0x0F,0xF0,0x03,0x0C,0x30,0xC0,0xFF};
    while (1)
    {
```

```
x = P1&7;
for (i = 0; i <8; i++)
{
    if (x == switch [i])
    {
        P2= LEDs [i];
        break;
    } } }
```

א. הסבר כל שורה באופן כללי . ב. מה מטרת התוכנית ? ג. אילו מפסקים מתוך ה 8 , משפיעים על LEDs ובאיזה אופן (אילו LEDs יידלקו במצבים השונים של המפסקים)?

שאלה 49 למערכת שבשאלה הקודמת נרשמה התוכנית :

```
unsigned char count (unsigned char num)
{
    unsigned char i, z = 0;
    for (i = 0; i <8; i++)
    {
        if ((num&1) == 1)
            z++;
        num = num>>1;
    }
    return (z);
}

void main(void)
{
    while(1)
        P2 = count(P1);
}
```

א) מה מטרת הפונקציה count()?

ב) הסבר כל שורה באופן כללי ואת תפקידה בתוכנית

ג) אילו LEDs יידלקו אם המפסקים נמצאים במצב המצויר (מפסק אחד סגור והשאר פתוחים)?

שאלה 50

1. להלן נתונה תכנית שמפעילה את הפורט הטורי - UART - של c8051f380: תדר הגביש 48 מגה הרץ.

```
void main(void)
```

```
{
    unsigned char x = 'A';
    SCON = 0x10;
    TMOD = 0x20;
    CKCON=8;
    TH1 = 0X30;
    TR1 = 1;
    while(x <= 'Z')
    {
        SBUF = x++;
        while(TI==0);
        TI = 0;
    }
}
```

א. הסבר כל שורה באופן כללי. ב. מה מטרת התוכנית? ג. באיזה Mode עובד הפורט הטורי? ד. מה קצב השידור של הפורט הטורי? ה. באיזה Mode עובד Timer1, ומה תפקידו בפורט הטורי? ו. מה ההבדל בין הפקודה SBUF = ++x; לבין הפקודה SBUF = x++;

שאלה 51 כתוב תוכנית בשפת C51 היוצרת בהדק P1.6 גל ריבועי בתדר 125 Hz. לספירת זמן השהיה יש להשתמש ב-Timer0 הפועל ב-Mode1. ידוע שתדר הפולסים הנכנסים למונה 12 מגה הרץ (תדר המערכת 48 מגה הרץ).

שאלה 52 (ממבחן טכנאים משרד החינוך 2013)

להלן תכנית בשפת C של המיקרו־בקר 8051. תדר הגביש של המיקרו־בקר הוא 12 MHz.

```

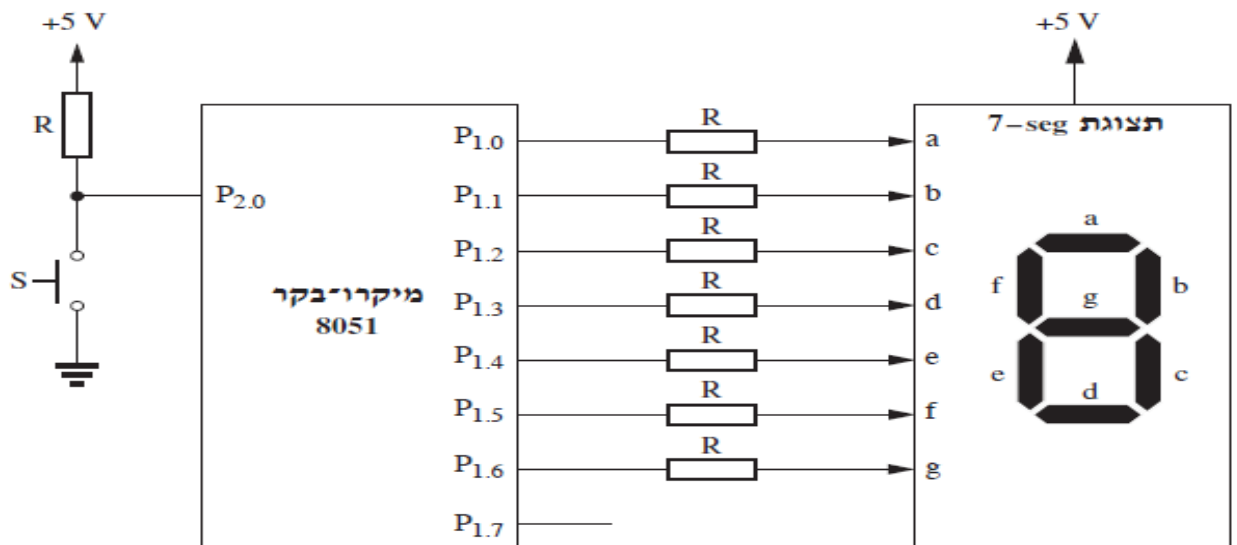
1.  #include<REG51.H>
2.  sbit P1_7=0x97;
3.  bit b=0;
4.  void t0() interrupt 1
5.  {
6.      TR0=0;
7.      P1_7=~P1_7;
8.      if (b==0)
9.      {
10.         TH0=0xFF;
11.         TL0=0xF0;
12.         b=1;
13.      }
14.      else
15.      {
16.         TH0=0xFF;
17.         TL0=0xD0;
18.         b=0;
19.      }
20.      TR0=1;
21.  }

22.  void main()
23.  {
24.      IE=0x82;
25.      TMOD=0x01;
26.      TH0=0xFF;
27.      TL0=0xD0;
28.      P1_7=1;
29.      TR0=1;
30.      while(1);
31.  }

```

- א. הסבר ההוראות שבשורות 3, 4, 7 ו 30
- ב. הסבר את המשמעות של ההוראות שבשורות 24 עד 29. היעזר בדף הנוסחאות של המיקרו־בקר לניתוח מילות הבקרה.
- ג. 1. סרטט את צורת האות המופק בהדק P1_7 כפונקציה של הזמן. 2. האם הזמן שבו האות נמצא במצב גבוה ('1') במהלך מחזור אחד שווה לזמן שבו האות נמצא במצב נמוך ('0') באותו המחזור? נמק את תשובתך.

נתונה תצוגת 7 מקטעים המתחברת לפורט 1. להדק P2.0 מחובר לחצן S שמצבו הרגיל פתוח



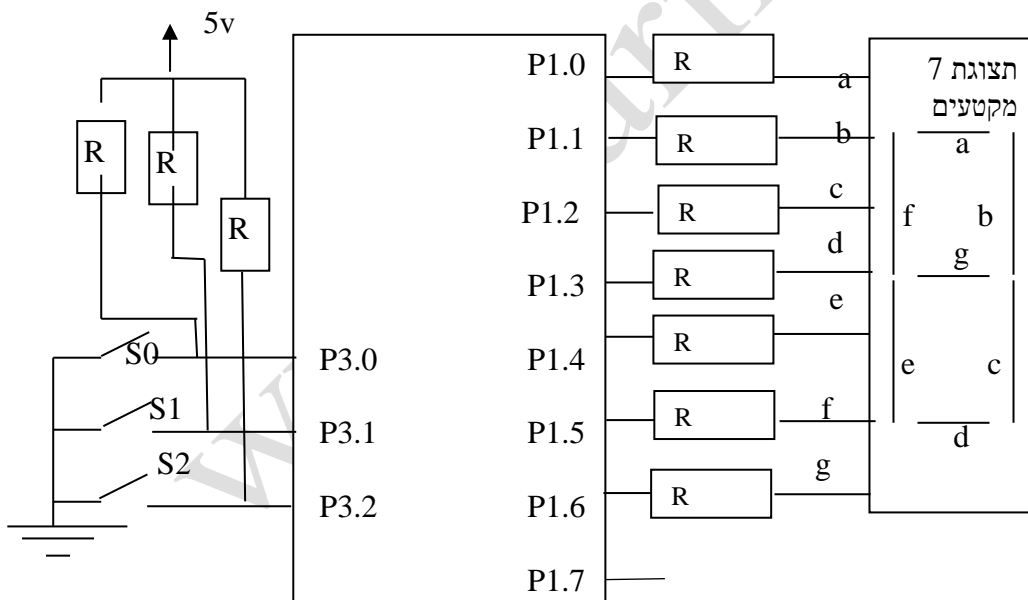
כתוב תכנית בשפת C51 שתבצע את הפעולות שלהלן :

1. תבדוק את מצב הלחצן S
2. אם הלחצן סגור – תופיע הספרה 4 בתצוגת 7 המקטעים עד שהלחצן ייפתח.
3. אם הלחצן פתוח – התצוגה חשוכה עד שהלחצן ייסגר.

שאלה 54 ממבחן בשנת 2014 באיור לשאלה נתונה תצוגת 7 מקטעים מסוג CC (קתודה משותפת) המחוברת להדקים

P1.0 ÷ P1.6

להדקים P3.0 ÷ P3.2 מחוברים המפסקים S0 ÷ S2.



כתוב תת שגרה בשפת הסף של המיקרו בקר 8051 או תכנית בשפת C שלו, שתבצע את הפעולות : 1. תבדוק כמה

מהמפסקים סגורים 2. תציג את מספר המפסקים הסגורים בתצוגת 7 המקטעים.

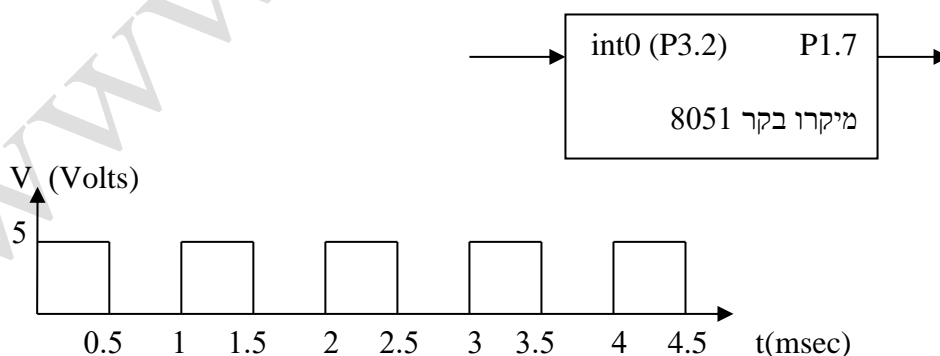
שאלה 55 ממבחן בשנת 2014

```

1.  #include <8051.h>
2.  unsigned char C;
3.  sbit P1_7 = 0x97 ;
4.  void int0 ( ) interrupt 0
5.  {
6.      C++;
7.      if ( C == 0xfe ) C=0;
8.  }
9.  void main ( )
10. {
11.     C = 0 ;
12.     P1_7 = 1 ;
13.     EX0 = 1 ;
14.     EA = 1 ;
15.     TCON = 0x01 ;
16.     while ( 1 )
17.     {
18.         if ( ( C % 2 == 0 ) ) P1_7 = 1 ;
19.         else P1_7 = 0 ;
20.     }
21. }

```

באיור הבא מתואר האות שנכנס להדק $\overline{\text{int0}}$ (P3.2) של המיקרו בקר.



- הסבר את ההוראות שבשורות 2, 3, 4, 16.
- הסבר את משמעות השורות 13, 14 ו 15. היעזר בדף הנוסחאות של המיקרו בקר לניתוח מילת הבקרה.
- העתק למחברתך את האות בהדק P3.2 וסרטט מתחתיו, בהתאמה, את צורת האות בהדק P1_7 במהלך ביצוע התוכנית.

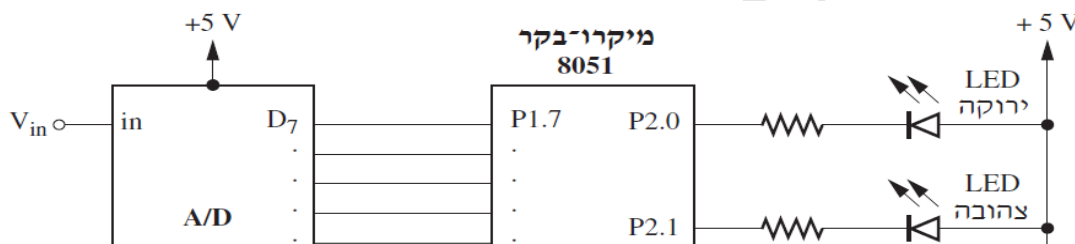
7. משנים את ההוראה שבשורה 18 להוראה הבאה : $P1_7 = 1 ; \text{ if } ((C \% 3 = 0))$. כיצד ישפיע שינוי זה על אות המוצא בהדק P1_7 ? נמק .

שאלה 56 ממבחן בשנת 2014

נתון בלוק נתונים המתחיל בכתובת 50H בזיכרון הפנימי של המיקרו בקר 8051 וגודלו עשרים בתים. בלוק הנתונים מכיל ערכים שלמים ממינוס 9 ועד פלוס 9 .
כתוב תכנית בשפת C51 שתבצע את הפעולות :
1. תאפס את תאי הזיכרון הפנימי שתוכנם הוא מספר שלילי . 2. תסכם את הנתונים החיוביים בבלוק ותציב את הסכום בתא שכתובתו 4fH בזיכרון הפנימי.

שאלה 57 שנת 2015

באיור לשאלה נתונה מערכת מיקרו-בקר 8051 המחוברת לממיר אות תקבילי לאות ספרתי (A/D). כושר ההבחנה של הממיר הוא 0.02 V , כלומר: כאשר המתח במבוא הממיר הוא 0 V – המספר במוצא הממיר יהיה 00000000 , וכל תוספת של 0.02 V תגדיל את המספר במוצא הממיר ב-1 . למיקרו-בקר חוברו שלוש נוריות LED : נורית LED אדומה להדק P2.2 , נורית LED צהובה להדק P2.1 ונורית LED ירוקה להדק P2.0 .



כתוב תכנית בשפת C של המיקרו-בקר 8051 שתבצע את הפעולות שלהלן:

1. תקלוט את דגימת המתח מממיר ה-A/D לתוך משתנה מתאים.
2. I. אם דגימת המתח שווה ל-4.5 V או גדולה יותר – תידלק הנורית הירוקה.
II. אם דגימת המתח גדולה מ-4.0 V וקטנה מ-4.5 V , תידלק הנורית הצהובה.
III. אם דגימת המתח שווה ל-4.0 V או קטנה יותר – תידלק הנורית האדומה.

24

להלן תכנית בשפת C של המיקרו־בקר 8051 (תדר הגביש של המיקרו־בקר הוא MHz)

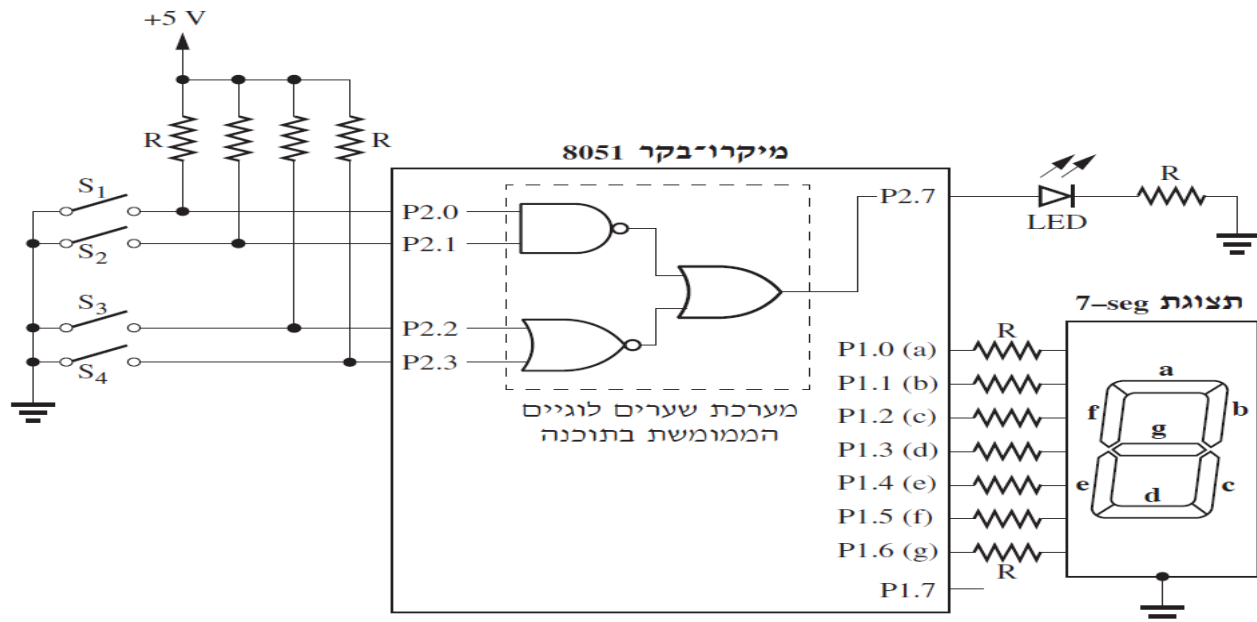
```

1. #include <8051.h>
2. char tav;
3. unsigned char i = 7;
4. char msg[7] _at_ 0x40;
5. void main ()
6. {
7.     TMOD = 0x20;
8.     SCON = 0x10;
9.     TH1 = 0xfd;
10.    TL1 = 0xfd;
11.    TR1=1;
12.    while(i)
13.    {
14.        i--;
15.        RI=0;
16.        while (!RI);
17.        tav = SBUF;
18.        msg[i]=tav;
19.    }
20. }
```

- א. הסבר את ההוראות שבשורות 4, 11, 12 ו-18.
- ב. הסבר את המשמעות של ההוראות שבשורות $8 \div 7$. היעזר בדף הנוסחאות של המיקרו־בקר לצורך ניתוח מילות הבקרה.
- ג. הסבר מה מבצעת התכנית.

שאלה 59 (2015)

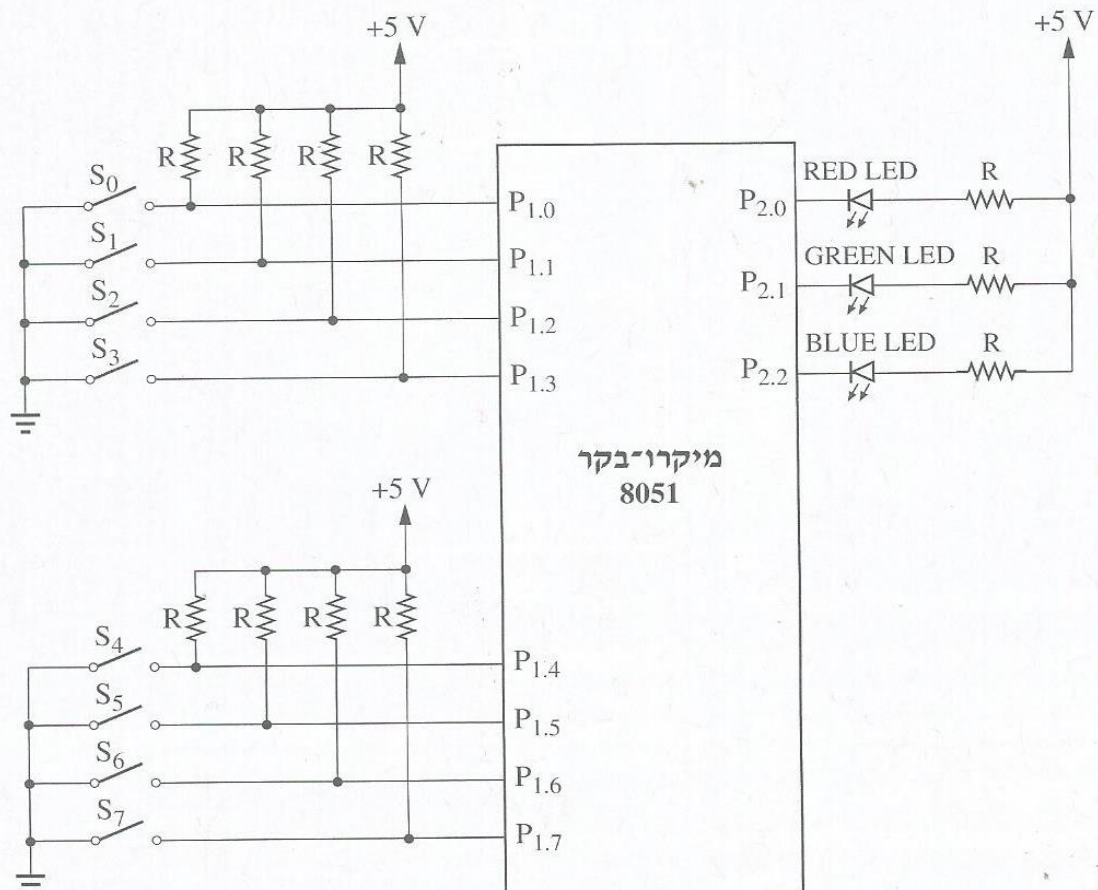
באיור לשאלה נתונה מערכת של שערים לוגיים הממומשת בתוכנה על-ידי המיקרו־בקר 8051. המערכת מופעלת באמצעות המתגים $S_1 \div S_4$, היכולים לספק לה את כל הצירופים האפשריים. פונקציית המוצא של מערכת השערים הלוגיים מתקבלת בהדק P2.7. נורית ה-LED המחוברת להדק P2.7 נדלקת או נכבית בהתאם לערך המתקבל במוצא של מערכת השערים הלוגיים. ההדקים P1.0 ÷ P1.7 מחוברים לתצוגת שבעה מקטעים (7-seg) מסוג CC.



כתוב תכנית בשפת-הסף של המיקרו־בקר 8051 או תכנית בשפת C שלו, שתבצע את הפעולות שלהלן:

1. תממש בתוכנה את הפונקציה הלוגית המתוארת באיור.
2. תפיק את הרמה הלוגית במוצא ההדק P2.7, בהתאם לערך המתקבל במוצא של מערכת השערים הלוגיים.
3. תציג בתצוגת שבעת המקטעים (7-seg) 0 או 1, בהתאם לרמה שהתקבלה בהדק P2.7.

באיור לשאלה □ נתון מיקרו־בקר 8051. ההדקים $P_{1.0} \div P_{1.3}$ שלו חוברו, בהתאמה, למפסקים $S_0 \div S_3$, וההדקים $P_{1.4} \div P_{1.7}$ שלו חוברו, בהתאמה, למפסקים $S_4 \div S_7$. כמו־כן חוברו נורית LED אדומה להדק $P_{2.0}$, נורית LED ירוקה להדק $P_{2.1}$, ונורית LED כחולה להדק $P_{2.2}$.



□ איור לשאלה

כתוב תכנית בשפת C של המיקרו־בקר 8051, שתבצע את הפעולות שלהלן:

1. תקלוט את מצב המפסקים $S_0 \div S_7$.
2. אם כל המפסקים פתוחים – היא תדליק את נורית ה־LED הכחולה בלבד.
3. אם לפחות אחד מהמפסקים סגור – התכנית תבדוק אם מצב המפסקים $S_0 \div S_3$ זהה למצב המפסקים $S_4 \div S_7$, בהתאמה. אם הוא זהה – התכנית תדליק את נורית ה־LED הירוקה בלבד. אם הוא אינו זהה – התכנית תדליק את נורית ה־LED האדומה בלבד.

```

2. {
3.     CH='A';
4.     I=0;
5.     TMOD=0x20;
6.     SCON=0x40;
7.     TH1=0xFD;
8.     TL1=0xFD;
9.     TR1=1;
10.    TI=0;
11.    IE=0x81;
12.    IT0=1;
13.    while(1)
14.    {
15.        SBUF=CH;
16.        while(!TI);
17.        TI=0;
18.    }
19. }

```

א. הסבר את ההוראות שבשורות 1, 3, 6 ו-26.

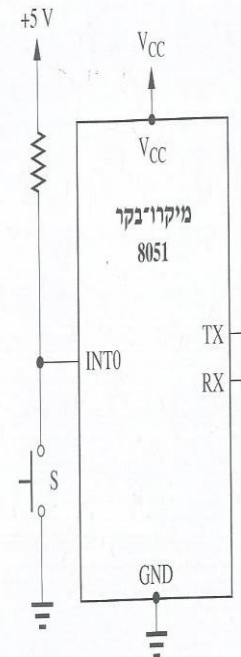
ב. הסבר את המשמעות של השורות 22 ÷ 15 בקוד התכנית. היעור בדף הנוסחאות של המיקרו-בקר לצורך ניתוח מילות הבקרה.

ג. העתק למחברתך את הטבלה שלהלן, השלם בה את הנתונים שישודרו במוצא ה-UART בכל אחד מן המצבים, ונמק את תשובותיך.

התעלם מבעיית הריטוטים העשויים להתרחש בעת לחיצה על הלחצן S.

נימוק	הנתונים שישודרו במוצא ה-UART
	לפני הלחיצה הראשונה
	לאחר הלחיצה הראשונה
	לאחר 2 לחיצות
	לאחר 3 לחיצות
	לאחר 4 לחיצות
	לאחר 5 לחיצות
	לאחר 6 לחיצות
	לאחר 7 לחיצות

איור לשאלה 61: נתון לחצן S המחובר להדק המבוא INTO במיקרו-בקר 8051. מצבו הרגיל של הלחצן הוא פתוח (N.O.).



איור לשאלה 61

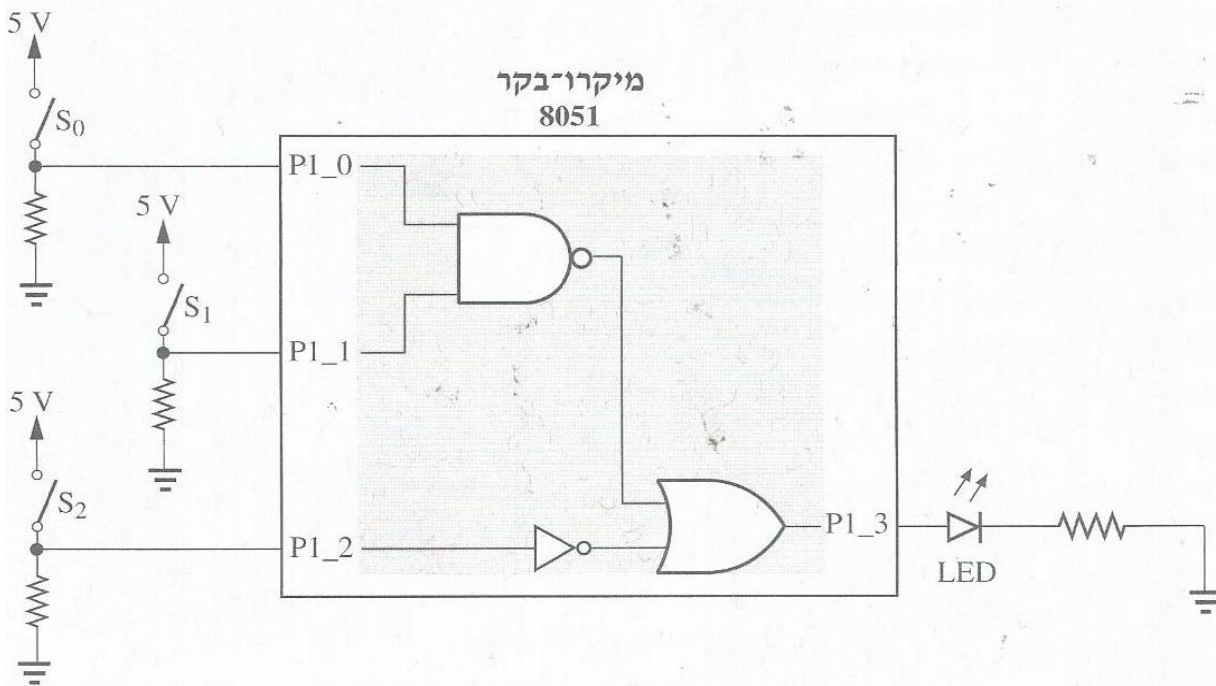
זמן תכנית בשפת C של המיקרו-בקר 8051, העושה שימוש ב-UART ובפסיקות חומרה חיצוניות:

```

1. #include <8051.H>
2. unsigned char CH,I;
3. void int0() interrupt0
4. {
5.     I++;
6.     if(I%3==0)
7.         CH = 'B';
8.     else
9.         CH = 'C';
10. }
11. void main(void)

```

באיור לשאלה ☐ נתון מיקרו-בקר 8051, מעוניינים לממש בתוכנה את המעגל הלוגי המתואר בתוכו.



☐ איור לשאלה

א. כתוב תכנית בשפת-הסף של המיקרו-בקר 8051 שתממש את המעגל הלוגי המתואר באיור. על התכנית:

1. לקלוט את הרמה הלוגית של המפסקים $S_0 \div S_2$ דרך שלושת ההדקים $P1_0 \div P1_2$, בהתאמה.
2. לחשב את הרמה הלוגית במוצא המעגל הלוגי, ולהציג אותה על-ידי נורית ה-LED במוצא ההדק $P1_3$.

שאלה 63 (2017)

להלן שתי תכניות בשפת C של המיקרו-בקר 8051, המפיקות גל מרובע בהדק P_{2.0} שלו. תדר הגביש של המיקרו-בקר הוא 12 MHz.

תכנית 2	תכנית 1
<pre> 1. #include <8051.h> 2. unsigned int d; 3. sbit P2_0 = 0xA0; 4. void main (void) 5. { 6. while (1) 7. { 8. P2_0=0; 9. for(d=0;d<400;d++); 10. P2_0=1; 11. for(d=0;d<400;d++); 12. } 13. }</pre>	<pre> 1. #include <8051.h> 2. sbit P2_0 = 0xA0; 3. void main (void) 4. { 5. TMOD=0x01; 6. TH0=0xF0; 7. TL0=0x00; 8. TR0=1; 9. P2_0=0; 10. while (1) 11. { 12. TF0=0; 13. P2_0=!P2_0; 14. TH0=0xF0; 15. TL0=0x00; 16. while (TF0==0); 17. } 18. }</pre>

- הסבר את ההוראות שבשורות 2, 10, 13 ו-16 בתכנית 1.
- מעוניינים להקטין פי 2 את זמן-המחזור של האות המופק בהדק P_{2.0}. רשום את הנדרש כדי לבצע זאת בתכנית 2.
- יש צורך להפיק אות, מדויק ככל שניתן, שתדרו יהיה 1 kHz ומחזור הפעולה שלו יהיה 50%.
 - קבע באיזו משתי התכניות נכון יותר להשתמש כדי להפיק תדר מדויק. נמק את תשובתך.
 - שנה את קוד התכנית שבחרת כדי להפיק אות בתדר הזה.

שאלה 64 (2017)

פלינדרום הוא מספר, מילה, משפט או כל רצף סמלים אחר, שניתן לקרוא משני הכיוונים, מימין לשמאל או משמאל לימין, ולקבל תוצאה זהה. למשל:

מספר פלינדרומי – 10100101, 01100110, 10000001

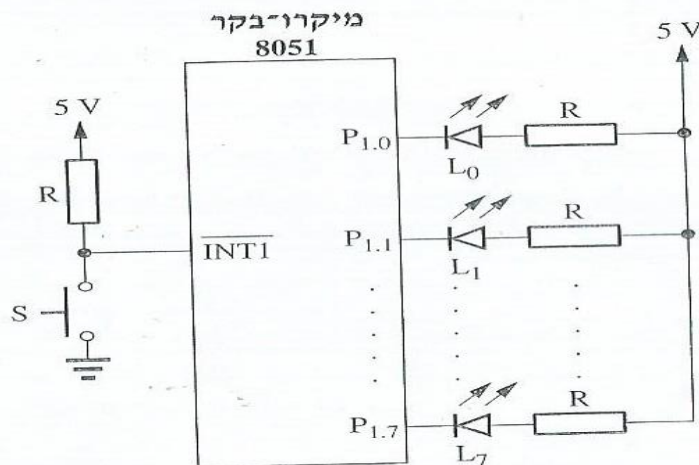
מילה פלינדרומית – שמש, תות, מילים

משפט פלינדרומי – רבע הלילה עבר

כתוב תת-שגרה בשפת C של המיקרו-בקר 8051, שתבצע את הפעולות שלהלן:

- תבדוק אם תוכנו של התא שכתובתו 40H בזיכרון הפנימי של המיקרו-בקר 8051 הוא פלינדרום.
- אם תוכנו של התא שכתובתו 40H הוא פלינדרום – היא תשאיר את תוכן התא ללא שינוי. אם לא – היא תאפס את תוכן התא.

באיור לשאלה □ נתון מיקרו־בקר 8051, שלהדק הפסיקה $\overline{INT1}$ שלו חובר לחצן S. נוריות ה־LED □ $L_0 \div L_7$, חוברים בהתאמה להדקים $P_{1.0} \div P_{1.7}$ של המיקרו־בקר.



□ איור לשאלה

להלן המבנה הכללי של תכנית בשפת C של המיקרו־בקר 8051, הכוללת פונקציה ראשית ופונקציית־פסיקה:

```
void main()
{
    .
    .
    .
}

void int1() interrupt 2
{
    .
    .
    .
}
```

א כתוב פונקציה ראשית main, שתאפשר פסיקה במבוא $\overline{INT1}$ בירידת השעון של המיקרו־בקר 8051, תדליק את הנורית L_0 בלבד, ותמתין לפסיקות נוספות.

ב כתוב פונקציית־פסיקה, שתדליק בכל פעם שהלחצן S נסגר נורית LED נוספת לאלו שכבר דולקות, החל מ־ L_1 ועד L_7 . הפסיקה תיחסם כאשר כל נוריות ה־LED, $L_0 \div L_7$, דולקות.

שאלה 66 (ממבחן תשע"ח - מרץ 2018)

להלן תוכנית המודדת את הרוחב של דופק חיובי המתקבל בהדק P2.0 של המיקרו-בקר 8051, ומאחסנת את הערך הנמדד במשתנה PulseWidth. תדר הגביש במיקרו-בקר הוא 12 MHz.

```
#include<8051.h>
① sbit P2_0 = P2 ^ 0;
unsigned int PulseWidth;
void main(void)
{
    ② TMOD = 0x01;
    while(1)
    {
        TH0 = 0;
        TL0 = 0;
        ③ while (P2_0 == 0);
        TR0 = 1;
        while(P2_0 == 1);
        TR0 = 0;
        ④ PulseWidth = (TH0 << 8) + TL0;
    }
}
```

א. הסבר את ההוראות המסומנות בספרות ①, ②, ③, ו-④.

ב. חשב את ערך המשתנה PulseWidth, כאשר רוחב הדופק החיובי הוא:

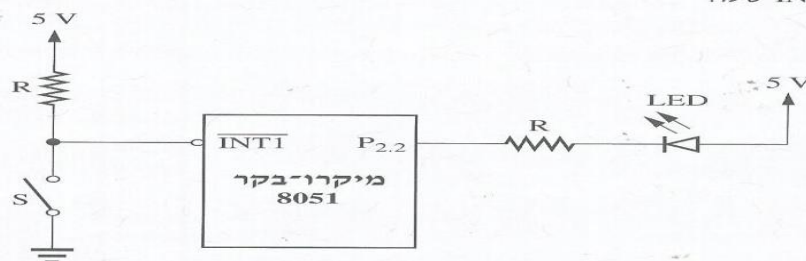
1. 100 μ sec
2. 12 msec
3. 100 msec

הערה: בחישוביך הזנח את זמני הפקודות בתוכנית.

ג. מהו רוחב הדופק המרבי שניתן למדוד באמצעות התוכנית הזו? נמק את תשובתך.

שאלה 67 (ממבחן תשע"ח - מרץ 2018)

באיור לשאלה 7 נתונים נורית LED המחוברת להדק P2.2 של המיקרו-בקר 8051 ומפסק המחובר להדק הפסיקה INTI שלו.



איור לשאלה 7

כתוב תוכנית הכוללת פונקציית main ותוכנית פסיקה, שהמבנה שלה יהיה:

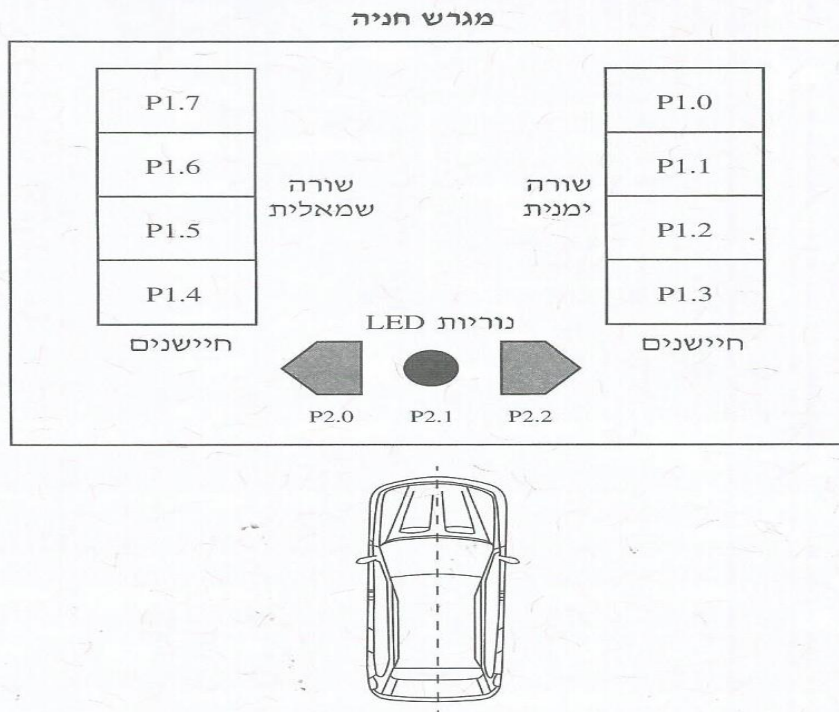
```
#include <8051.h>
void main (void)
{
}
void int1 interrupt 2
{
}
}
```

הדרישות מן התוכנית:

1. בפונקציית main נדרש לבצע את הפעולות האלה:
 - איפשור פסיקה במבוא INTI בירידת השעון
 - כיבוי נורית ה-LED
 2. תוכנית הפסיקה תגרום לנורית ה-LED להידלק למשך שנייה אחת בכל אחת מ-10 הפסיקות הראשונות. לאחר 10 פסיקות תיחסם תוכנית הפסיקה.
- הערה: לרשותך פונקציית השהיה המקבלת ערך במילי-שניות – delay (int msec).

שאלה 68 (ממבחן תשע"ח - מרץ 2018)

במגרש חניה יש שתי שורות של חניות, שבכל אחת מהן ארבעה מקומות, כמתואר באיור לשאלה ☐ מעל כל מקום חניה מותקן חיישן, המספק '1' כאשר מקום החניה תפוס ו-'0' כאשר הוא פנוי.



☐ **איור לשאלה**

ארבעת החיישנים בשורה הימנית מחוברים להדקים $P_{1.0} \div P_{1.3}$ של מיקרו-בקר 8051, וארבעת החיישנים בשורה השמאלית מחוברים להדקים $P_{1.4} \div P_{1.7}$ שלו.

בכניסה למגרש החניה ישנן שלוש נוריות LED, המחוברות להדקים $P_{2.0} \div P_{2.2}$ של המיקרו-בקר. הן משמשות כנוריות חיווי לרכב המבקש לחנות במגרש, ונדלקות כאשר המיקרו-בקר מספק להן '1':

- נורית ה-LED הירוקה השמאלית מחוברת להדק $P_{2.0}$, ונדלקת כאשר יש מקום חניה פנוי בשורה השמאלית.
- נורית ה-LED הירוקה הימנית מחוברת להדק $P_{2.2}$, ונדלקת כאשר יש מקום חניה פנוי בשורה הימנית.
- נורית ה-LED האדומה מחוברת להדק $P_{2.1}$, ודולקת כאשר אין מקום חניה פנוי במגרש.

כתוב תוכנית בשפת C של המיקרו-בקר 8051 למימוש מערכת החניה הזו.

שאלה 69 (תש"פ - מרץ 2020)

בשאלה זו יש לשנות את הכתובות המתחילות ב 2000H והלאה ל 200H והלאה בהתאמה.

כתוב תת-שגרה בשפת-הסף של המיקרו-בקר 8051 או תוכנית בשפת C שתבצע את הפעולות האלה:

1. תגדיר בלוק נתונים המתחיל בכתובת 2000H בזיכרון החיצוני של המיקרו-בקר וגודלו 100 בתיים.

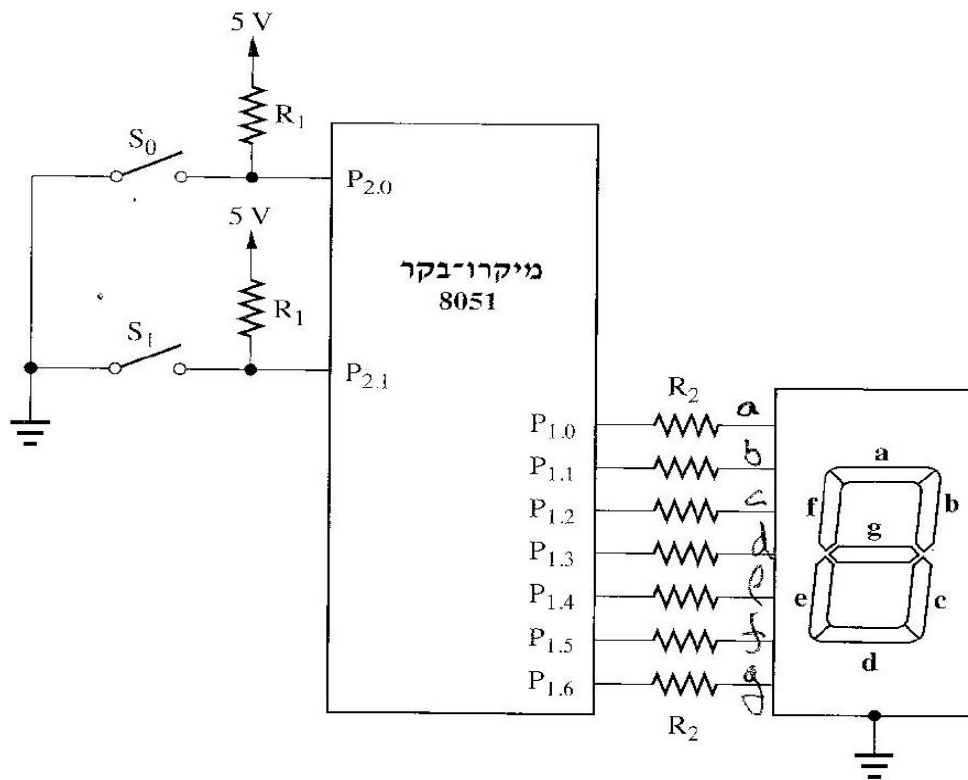
2. תציב בבלוק הנתונים:

– החל מהתא שכתובתו 2000H עד התא שכתובתו 2031H – מספרים זוגיים בסדר עולה מ-0 עד 98 (בבסיס 10).

– החל מהתא שכתובתו 2032H עד תא שכתובתו 2063H – מספרים אי-זוגיים בסדר יורד מ-99 עד 1 (בבסיס 10).

להלן דוגמה עבור בלוק הנתונים הזה:

תוכן התא	כתובת התא
0	2000H
2	2001H
4	2002H
6	2003H
.	.
.	.
.	.
94	202FH
96	2030H
98	2031H
99	2032H
97	2033H
95	2034H
.	.
.	.
.	.
5	2061H
3	2062H
1	2063H

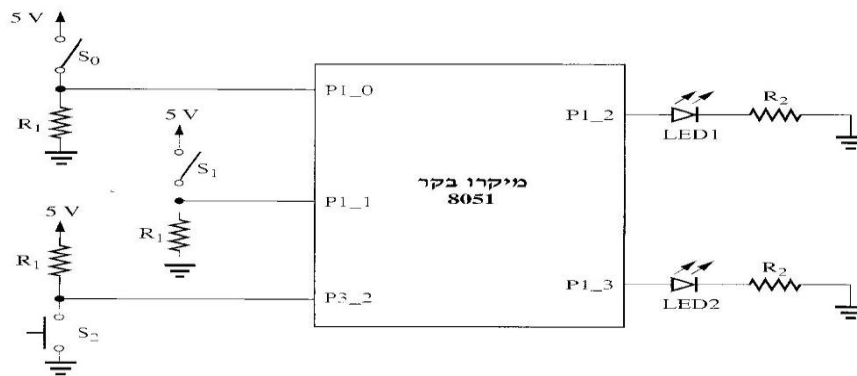


איור לשאלה 7

- המערכת מופעלת באמצעות מפסקים S_0 ו- S_1 המחוברים בהתאמה להדקים $P_{2,0}$ ו- $P_{2,1}$.
ההדקים $P_{1,6} \div P_{1,0}$ מחוברים לתצוגת שבעה מקטעים (7-seg) בחיבור CC (קתודה משותפת).
כתוב תוכנית בשפת C של המיקרו־בקר 8051 שתבצע את הפעולות האלה:
1. תקרא את מצב המפסקים המחוברים להדקים $P_{2,0}$ ו- $P_{2,1}$.
 2. תציג בתצוגת שבעה המקטעים (7-seg) את הערך המספרי (0, 1, 2 או 3) המתקבל בהדקים $P_{2,0}$ ו- $P_{2,1}$.

שאלה 71 (תש"פ – מרץ 2020)

באיור לשאלה 8 נתון מיקרו-בקר 8051 המכמש מעגל לוגי כלשהו.



איור לשאלה 8

התוכנית שלהלן נכתבה למימוש המעגל הלוגי. התוכנית כתובה בשפת C של המיקרו-בקר 8051.

```

1.  #include <8051.h>
2.  sbit P1_0 = 0x90;
3.  sbit P1_1 = 0x91;
4.  sbit P1_2 = 0x92;
5.  sbit P1_3 = 0x93;
6.  bit Q=0,NQ=1,t;
7.  void int0() interrupt0
8.  {
9.      if(P1_0==1 && P1_1==1)
10.     {
11.         t=Q;
12.         Q=NQ;
13.         NQ=t;
14.     }
15.     else if(P1_0==1 && P1_1==0)
16.     {
17.         Q=1;
18.         NQ=0;
19.     }
20.     else if(P1_0==0 && P1_1==1)
21.     {
22.         Q=0;
23.         NQ=1;
24.     }
25. }
26. void main (void)
27. {
28.     IT0 = 1;
29.     IE = 0x81;
30.     while(1)
31.     {
32.         P1_2=Q;
33.         P1_3=NQ;
34.     }
35. }
```

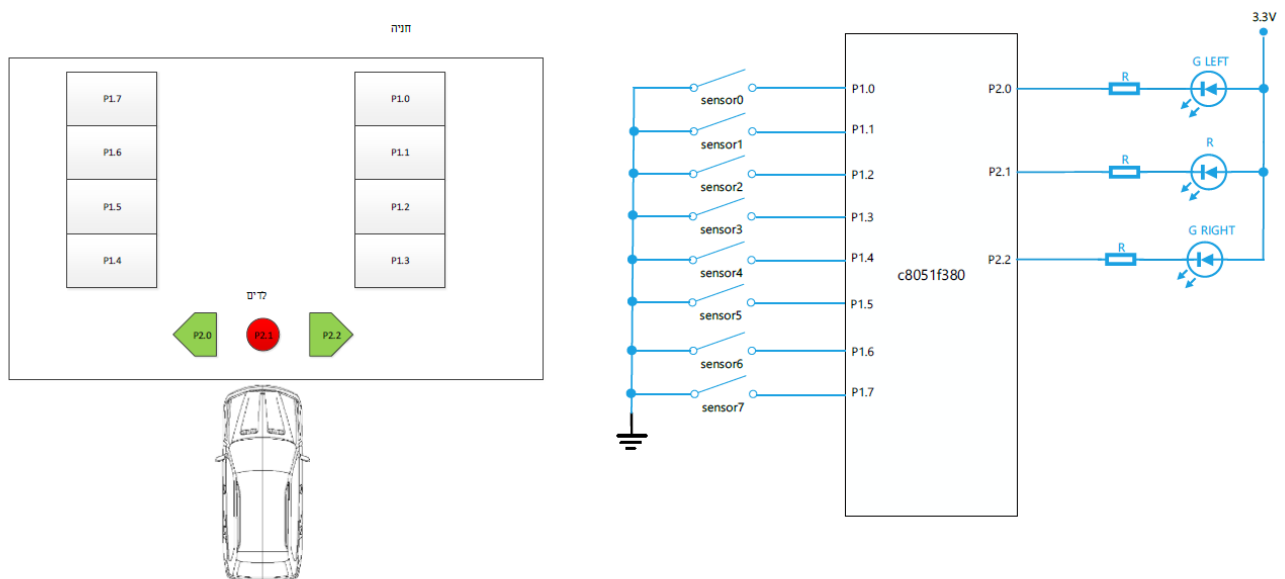
8. (נק') א. הסבר את ההוראות שבשורות 2, 6, 7 ו-9.
 ב. (נק') לפיך ארבעה היגדים המתייחסים לקוד התוכנית. בחר את ההיגד הנכון מבין הארבעה. העתק אותו למחברתך ונמק את בחירתך.

1. כאשר הרמה הלוגית של הדק P3_2 נמוכה, הפונקציית int0 תופעל באופן מחזורי.
2. בקוד התוכנית הראשית אין זימון לפונקציית int0, ולכן הפונקציית int0 לא תופעל.
3. ירידה ברמה הלוגית של הדק P3_2 (מגבוהה לנמוכה) תגרום לתוכנית לעבור לפונקציית int0.
4. בתחילה תופעל הפונקציית int0 על-ידי המיקרו-בקר, ובהמשך תופעל הפונקציית main(), על-פי סדר כתיבת הפונקציות בקוד התוכנית.

במגרש חניה ממוקמים שמונה חיישנים למיקום רכב. המגרש מחולק לארבע חניות בצידו הימני של המגרש וארבע חניות בצידו השמאלי של המגרש. ארבע החיישנים של הצד הימני מחוברים להדקים P1.0- P1.3 וארבע החיישנים של הצד השמאלי מחוברים להדקים P1.4 – P1.7. כאשר רכב נמצא בחנייה החיישן סגור. בנוסף במגרש החניה קיימים שלוש נורות LED: שתי נורות LED בצבע ירוק ונורה LED בצבע אדום. נורות ה-LED מחוברים להדקים P2.0-P2.2 המשמשים לחיווי לפי המצבים הבאים :

- נורת LED הימנית בצבע ירוק המחוברת להדק P2.2 תדלק כאשר יש מקום חניה פנוי בצד הימני של המגרש.
- נורת LED השמאלית בצבע ירוק המחוברת להדק P2.0 תדלק כאשר יש מקום חניה פנוי בצד השמאלי של המגרש.
- נורת LED בצבע אדום המחוברת להדק P2.1 מהבהב כל שניה כאשר אין מקום בחניה. לצורך יצירת השהיות בין הדלקות נתונה פונקציית השהייה במילי שניות :

```
delay_ms(int ms)
```



הנח שהתוכנית כוללת אתחול תדר השעון וביטול פעולת WDT .

- כתבו תוכנית המפעילה את נורות ה-LED בהתאם למצבים שהוגדרו.
- כתוב פונקציה בשם NumberParking() שתספור את מספר המקומות הפנויים ותחזיר ערך זה.

.א.

```

#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);
void delay_ms(int ms);

sbit G_LEFT=P2^0;
sbit R=P2^1;
sbit G_RIGHT=P2^2;

void main(){
    Init_Device();
    // LÉDS OFF
    G_LEFT=1;
    R=1;
    G_RIGHT=1;

    while(1){
        if((P1&0x0F) != 0)
            G_RIGHT=0;
        else
            G_RIGHT=1;

        if((P1&0xF0) != 0)
            G_LEFT=0;
        else
            G_LEFT=1;

        if(P1 == 0){
            R=!R;
            delay_ms(1000);
        }
        else
            R=1;
    }
}

```

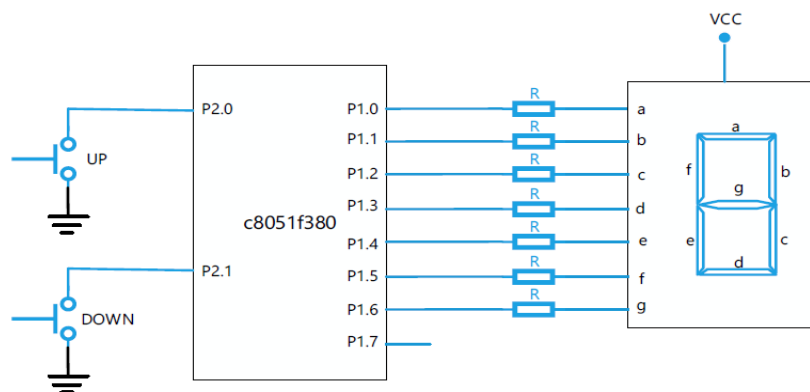
.ב.

```

U8 NumberParking(){
    U8 i,cnt=0;
    for(i=0;i<8 ;i++) {
        if(P1 & (1<<i) )
            cnt++;
    }
    return cnt;
}

```

נתונה המערכת הבאה הכוללת מיקרו-בקר C8051F380, תצוגה 7 Segment ושני לחצנים (ללא ריטוטים).



להלן קוד התוכנית

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
void Init_Device(void);
code unsigned char table[10]={ 0x3f, 0x06, 0x5b, 0x4f, 0x66,
                                0x6d, 0x7d, 0x07, 0x7f, 0x6f };

sbit UP=P2^0;
sbit DOWN=P2^1;
void main() {
    int cnt=0;
    Init_Device();
    P1=~table[0];
    while(1) {
        if(!UP){
            if (cnt<9) cnt++;
            P1=~table[cnt];
            while(!UP);
        }

        if(!DOWN){
            if (cnt>0) cnt--;
            P1=~table[cnt];
            while(!DOWN);
        }
    }
}
```

- א. הסבר מה מבצעת התוכנית.
- ב. מה יוצג בתצוגת 7segment בזמן אתחול הבקר.
- ג. הסבר את הפקודה : `P1=~table[cnt];`
- ד. למה נועדו הפקודות : `while(!UP);` ו- `while(!DOWN);`
- ה. שנה את התוכנית כך שלחיצה אחת על לחצן UP תקדם את המונה כל שנייה מהערך הקיים עד 9 ותעצור ולחיצה אחת על DOWN תמנה מטה כל שנייה עד לערך 0 ותעצור. לצורך יצירת השהיות בין הדלקות נתונה פונקציית השהייה במילי שניות :

```
delay_ms(int ms)
```

א. התוכנית משמשת כמונה בתחום 0-9. קידום ערך המונה באחד מתבצע באמצעות לחצן UP, חיסור אחד מערך המונה מתבצע באמצעות לחצן DOWN. תוצאת המניה תוצג על גבי תצוגה 7 Segment.

ב. בזמן אתחול הבקר והפקודה $P1=\sim table[0]$; תגרום להצגת הספרה 0 בתצוגה.

ג. הפקודה $P1=\sim table[cnt]$ תשלח ל-P1 ערך שיגרום להצגת הערך של cnt בתצוגה

הסבר: cnt מצביע על מיקום במערך מ-0 עד 9, הערך מהטבלה עובר היפוך ~ וגורם

לנורות ה-LED המקבלים '0' לוגי לדלוק (אנודה משותפת)

לדוגמה – עבור $cnt=1$ P1 יקבל את ההיפוך של 0x06 כלומר 11111001 לכן נורות LED b

ו- c יפעלו ונקבל את הערך 1 בתצוגה.

ד. הפקודות נועדו להמתנה לשחרור הלחצן, אחרת המונה היה מתקדם במהירות לפי זמן הלולאה.

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
void Init_Device(void);
void delay_ms(int ms);
code unsigned char table[10]= {0x3f, 0x06, 0x5b, 0x4f, 0x66,
                                0x6d, 0x7d, 0x07, 0x7f, 0x6f };

sbit UP=P2^0;
sbit DOWN=P2^1;
void main() {
    int cnt=0;
    Init_Device();
    P1=~table[0];
    while(1) {
        if(!UP) {
            while(cnt<9) {
                delay_ms(1000);
                cnt++;
                P1=~table[cnt];
            }
        }

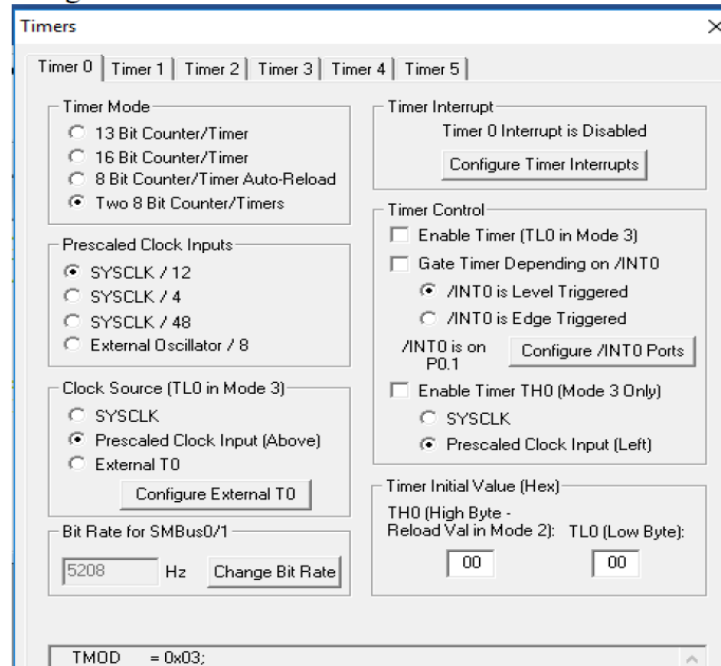
        if(!DOWN) {
            while(cnt>0) {
                delay_ms(1000);
                cnt--;
                P1=~table[cnt];
            }
        }
    }
}
```


נתונה התוכנית הבאה המייצרת גל ריבועי בהדק P2.0 בהתאם לערך המתקבל בפורט P1
תדר השעון SYSCLK הוא 12MHz:

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
void Init_Device(void);
sbit P2_0 = P2 ^ 0;
void main(void) {
    Init_Device();
    TMOD = 0x03;
    P2_0 = 0;

    while (1) {
        if (P1 == 0) {
            TR0 = 0;
            P2_0 = 0;
        }
        else {
            TR0 = 1;
            TF0 = 0;
            TL0 = P1;
            while (TF0 == 0);
            P2_0 = 1;
            TF0 = 0;
            TL0 = 256 - P1;
            while (TF0 == 0);
            P2_0 = 0;
        }
    }
}
```

להלן צילום מסך של אתחול Timer 0 באמצעות תוכנת 2 Configuration Wizard



- א. הסברו את הפקודה $TMOD = 0x03$; וציין באיזה תדר פועל הטיימר.
 ב. צייר עם ערכי הזמן את צורת הגל בפורט P2.0 עבור:

1. $P1=0$

2. $P1=0x80$

3. $P1=0x20$

4. $P1=255$

הערה: ניתן להזניח את זמני הפקודות.

- ג. הצע דרך להגדיל פי 2 את תדר גל המוצא בפורט P2.0

פתרון 74

- א. אוגר הבקרה $TMOD$ אחראי לפעולת $TIMER0/1$

זהו מבנה רגיסטר:

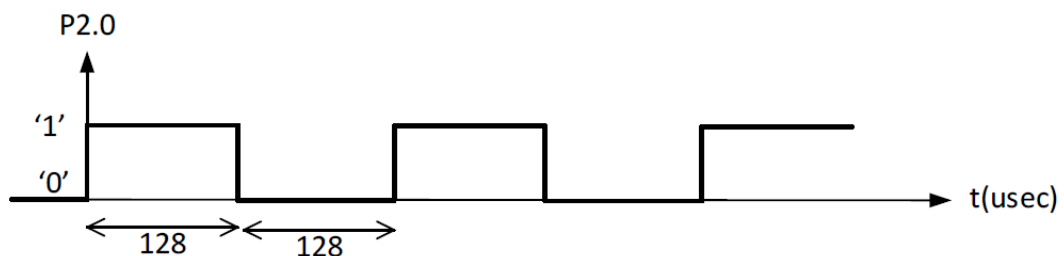
Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1]	T1M[0]	GATE0	C/T0	T0M[1]	T0M[0]

עבור ערך $0x03$ המונה 0 פועל במוד 3 בו פועלים 2 מונים של 8 סיביות והמונים פועלים כ-

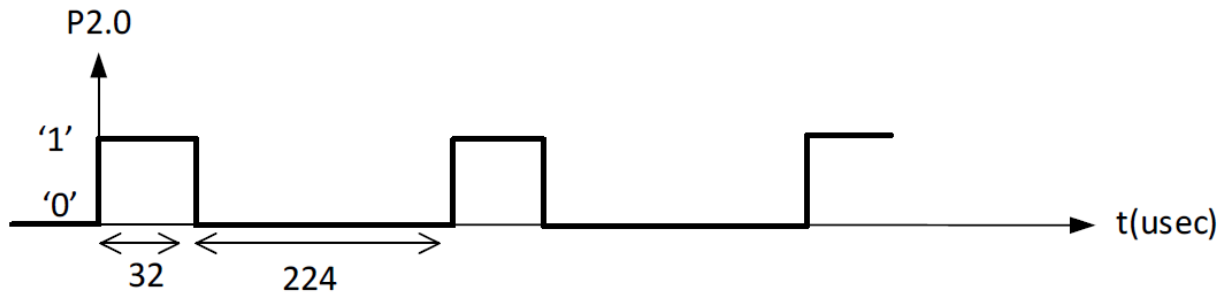
$TIMER$

- ב. 1. נקבל מתח 0 במוצא.

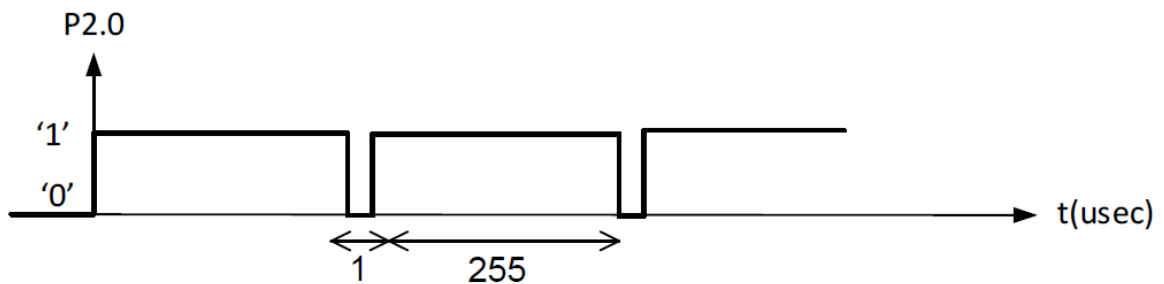
2. המוצא יהיה 0 בזמן שהטיימר ימנה מעלה מהערך $0x80=128$ עד שיגיע ל-0 (אחרי הערך 255 ואז הדגל $TF0$ עולה ל-1) כלומר 128 מחזורי שעון של $1\mu sec$ ($12MHz/12$), לאחר מכן המוצא יהיה '1' לוגי והמונה סופר מעלה מהערך $256-128=128$ עד 0. לכן נקבל גל מחזורי עם $128\mu sec$ במצב '0' ו-'1' לוגי.



3. '0' לוגי במוצא במשך $256 - 0 \times 20 = 224 \mu\text{sec}$ ו-'1' לוגי במשך $32 \mu\text{sec}$



4. עבור ערך 255, '0' לוגי במשך 1 מחזור ו-255 מחזורים עבור '1' לוגי



ג. אפשרות אחת היא להגדיל את תדר השעון מ-12MHz ל-24MHz

נתונה התוכנית הבאה המודדת רוחב פולס חיובי בהדק P2.0 ומאכסנת את הערך הנמדד במשתנה pulseWidth. המיקרו-בקר מאותחל לתדר SYSCLK של 12MHz ומחולק ב-12 עבור

TIMER0

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

sbit P2_0 = P2 ^ 0;

void main(void) {
    U16 pulseWidth;
    Init_Device();
    TMOD = 0x01;

    while (1) {
        TH0 = 0;
        TL0 = 0;
        while (P2_0 == 0);
        TR0 = 1;
        while (P2_0 == 1);
        TR0 = 0;
        PulseWidth = (TH0 << 8) + TL0;
    }
}
```

א. הסברו את ההוראה TMOD = 0x01;

ב. ציינו את ערך המשתנה pulseWidth, כאשר רוחב הפולס החיובי הוא:

1. 100 usec

2. 12 msec

3. 100 msec

הערה: ניתן להזניח את זמני הפקודות.

ג. מהו רוחב הפולס המינימלי והמכסימלי שניתן למדוד באמצעות תוכנית זו.

ד. הוסיפו/שנו את התוכנית כך שימדוד את זמן המחזור ביחידות מיקרו שניה במשתנה

(T) כאשר מסופק לכניסה (P2.0) גל מחזורי ריבועי.

א. אוגר הבקרה TMOD אחראי לפעולת TIMER0/1

זהו מבנה הרגיסטר:

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1]	T1M[0]	GATE0	C/T0	T0M[1]	T0M[0]

עבור ערך 0x01 המונה 0 פועל במוד 1 בו פועל מונה אחד של 16 סיביות ופועל כ- TIMER
 ב. המונה של 16 סיביות רץ לפי שעון של 1MHz (12MHz/12, זמן מחזור של 1usec) כאשר
 P2.0 עולה ל-1' לוגי ומפסיק כאשר P2.0 יורד ל-0' לוגי.

1. עבור רוחב פולס של 100usec, $PulseWidth = T0 = 100$
2. עבור רוחב פולס של 12msec, $PulseWidth = T0 = 12000$
3. עבור רוחב פולס של 100msec, המונה יכול לספור רק עד 2^{16} כלומר עד רוחב פולס של 65.536msec לכן המונה יגיע לערך זה ויספור שוב בשארית הזמן 100m-
 $65.536m = 34.464msec$ ונקבל $PulseWidth = 34464$.
- ג. רוחב פולס מכסימלי 65.536msec ומינימלי 1usec.

ד.

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

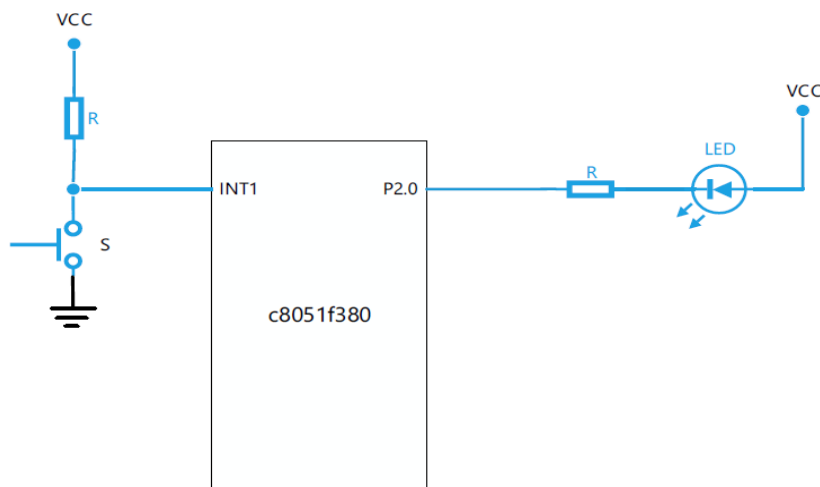
void Init_Device(void);
sbit P2_0 = P2 ^ 0;

void main(void) {
    unsigned int T;
    Init_Device();
    TMOD = 0x01;
    while (1) {
        TH0 = 0;
        TL0 = 0;
        while (P2_0 == 1);
        while (P2_0 == 0);
        TR0 = 1;
        while (P2_0 == 1);
        while (P2_0 == 0);
        TR0 = 0;
        T = (TH0 << 8) + TL0;
    }
}
```

התוכנית ממתינה לעלייה של 1' לוגי (P2_0 == 0) הפקודה לפני כן (P2_0 == 1) במקרה שהבדיקה מתחילה באמצע הפולס החיובי של הגל);
 המונה מתחיל לספור עד P2.0 יורד ל-0' (P2_0 == 1); ויעלה שוב ל-1' (P2_0 == 0);

76. פסיקות חומרה

נתונה המערכת הבאה הכוללת מיקרו-בקר C8051F380, נורת LED המחוברת להדק P2.0 ולחצן ללא ריטוטים המחובר להדק הפסיקה .



לפניכם חלק מפונקציית main ופונקציית הפסיקה :

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

sbit P2_0 = P2 ^ 0;

void main(void)
{
    Init_Device();
    P2_0 = 1;
    IE = 0x84;
    TCON = 0x04;

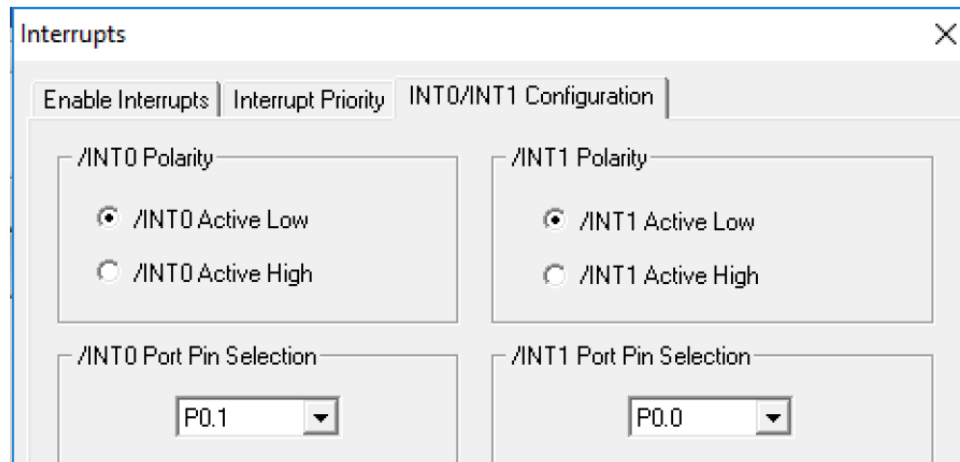
    while (1)
    {
    }
}

void int1() interrupt 2
{
}
```

א. בפונקציית main, הסבירו את שתי מילות הבקרה

```
IE = 0x84;
TCON = 0x04;
```

ב. נתון חלון הדקי הפסיקות :



כתבו לאיזה הדק חובר הלחצן, האם הפסיקה מתבצעת בזמן לחיצה או שחרור הלחצן.

ג. כתוב את תוכנית הפסיקה, אשר תגרום ללד לדלוק ולאחר שנייה להיכבות, עבור 5 הפסיקות הראשונות ועבור 5 הפסיקות הבאות לדלוק למשך 2 שניות. לאחר 10 פסיקות תיחסם הפסיקה.

לצורך יצירת השהיות בין הדלקות נתונה פונקציית השהייה במילי שניות :

`delay_ms(int ms)`

פתרון

א. כדי לאפשר את פסיקה חיצונית מספר 1 יש לכתוב לרגיסטר IE זהו מבנה הרגיסטר IE:

Bit	7	6	5	4	3	2	1	0
Name	EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0
Value	1	0	0	0	0	1	0	0

מהרגיסטר ניתן לראות שסיבית אפשר קבלת הפסיקות פעילה EA וסיבית אפשר פסיקה חיצונית 1 גם פעילה.

רגיסטר TCON מאפשר את אופן דרבון הפסיקה זהו מבנה הרגיסטר:

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
value	0	0	0	0	0	1	0	0

מהרגיסטר ניתן לראות שערך סיבית IT1 '1' לוגי כך שהפסיקה תהיה פעילה בדרבון קצה.

ב. הלחצן מחובר להדק P0.0 (int1) והפסיקה מתבצעת בזמן הלחיצה (Active Low) ניתן לראות

זאת באמצעות רגיסטר IT01CF.

זהו מבנה הרגיסטר:

Bit	7	6	5	4	3	2	1	0
Name	IN1PL	IN1SL[2:0]			IN0PL	IN0SL[2:0]		
value	0	0	0	1	0	0	0	0

.א

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);
void delay_ms(int ms);

sbit P2_0 = P2 ^ 0;

int cnt=0;

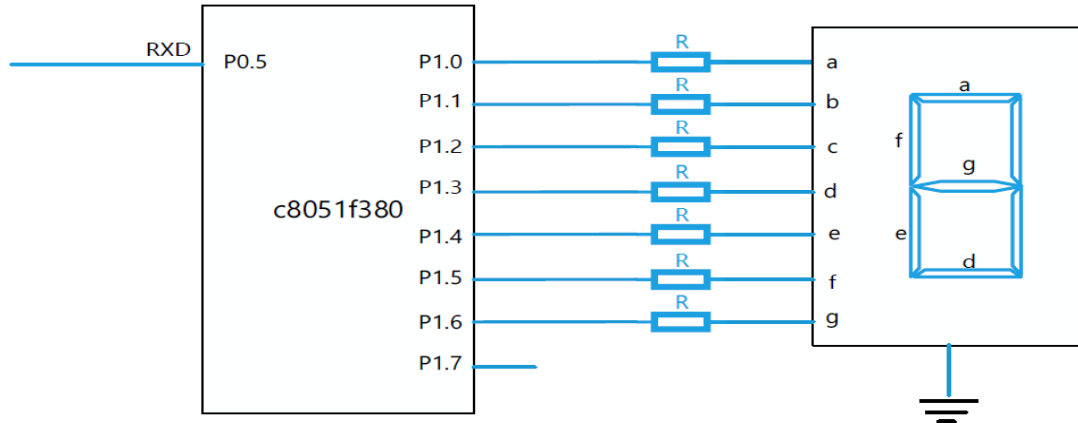
void main(void) {
    Init_Device();
    P2_0 = 1;
    IE = 0x84;
    TCON = 0x04;

    while (1);
}

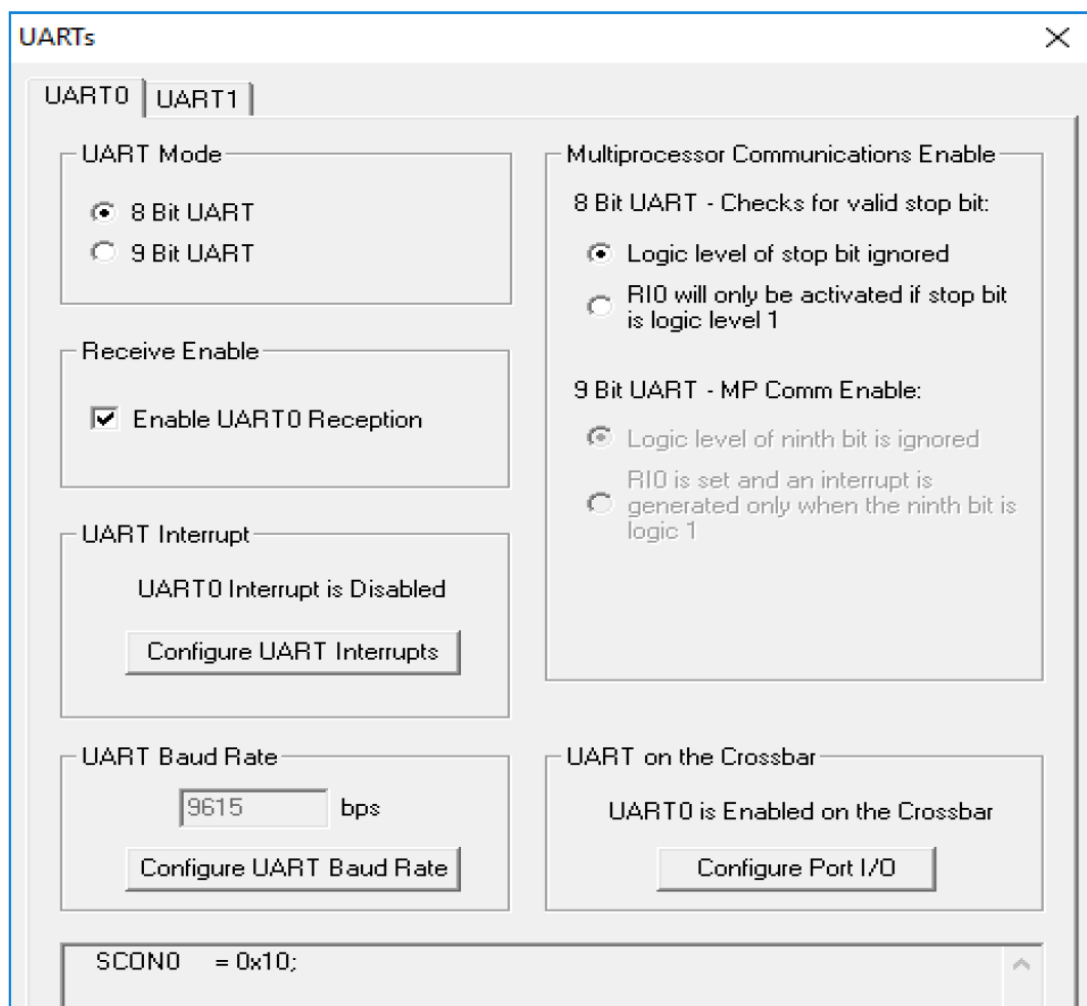
void int1() interrupt 2 {
    if(cnt<5)
    {
        cnt++;
        P2_0=0;
        delay_ms(1000);
        P2_0=1;
    }
    else if(cnt<10)
    {
        cnt++;
        P2_0=0;
        delay_ms(2000);
        P2_0=1;
    }
    else
        IE = 0x80;
}
```


UART .77

נתונה המערכת הבאה במשמשת למערכת לקליטת תו המייצג ספרה בתקשורת טורית. המערכת כוללת מיקרו-בקר C8051F380, תצוגת 7 Segment בחיבור קתודה משותפת המחוברת לפורט P1.



להלן תמונת חלון UART0 כפי שהוגדרו באמצעות תוכנה גרפית 2 Configuration Wizard.



להלן תמונת חלון TIMER1 כפי שהוגדרו באמצעות תוכנה גרפית 2 Configuration Wizard.

Timers

Timer 0 | Timer 1 | Timer 2 | Timer 3 | Timer 4 | Timer 5

Timer Mode

- ☐ 13 Bit Counter/Timer
- ☐ 16 Bit Counter/Timer
- ☒ 8 Bit Counter/Timer Auto-Reload
- ☐ Timer 1 Disabled

Timer Interrupt

Timer 1 Interrupt is Disabled

[Configure Timer Interrupts](#)

Timer Control

- ☒ Enable Timer
- ☐ Gate Timer Depending on /INT1
 - ☒ /INT1 is Level Triggered
 - ☐ /INT1 is Edge Triggered

/INT1 is on P0.0 [Configure /INT1 Ports](#)

Prescaled Clock Inputs

- ☒ SYSCLK / 12
- ☐ SYSCLK / 4
- ☐ SYSCLK / 48
- ☐ External Oscillator / 8

Clock Source

- ☐ SYSCLK
- ☒ Prescaled Clock Input (Above)
- ☐ External T1

[Configure External T1](#)

Timer Initial Value (Hex)

TH1 (High Byte - Reload Val in Mode 2): TL1 (Low Byte):

30 00

Bit Rate for SMBus0

6410 Hz [Change Bit Rate](#)

Baud Rate for UART

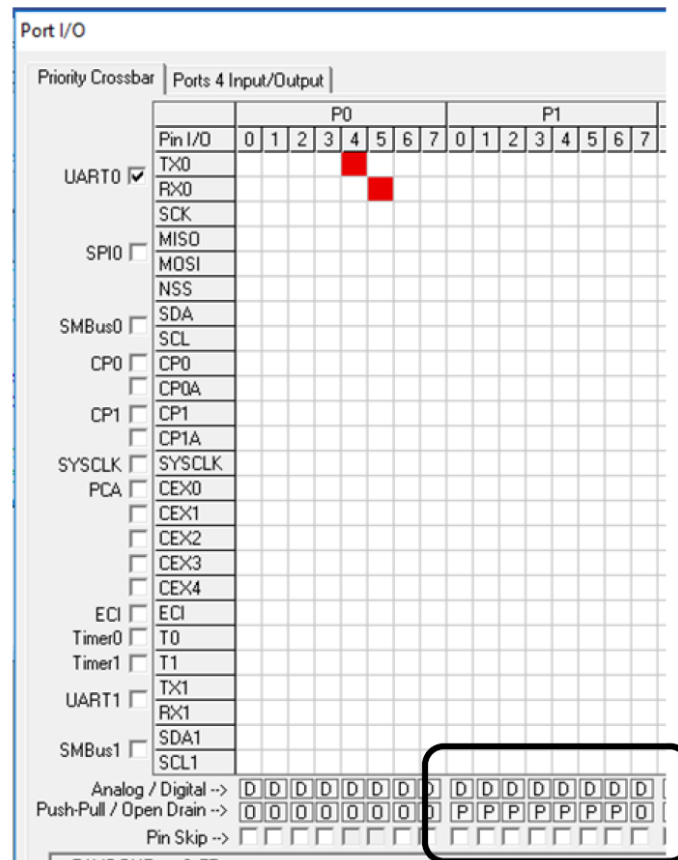
9615 bps [Change Baud Rate](#)

```
TCON = 0x40;
TMOD = 0x20;
TH1 = 0x30;
```

א. ציינו את הקשר בין מילת הבקרה SCON0 לאתחול ה-UART

ב. ציינו את הקשר בין מילות הבקרה לקביעת קצב התקשורת 9615Baud ואתחולו (תדר שעון הבקר הוא 48MHz)

ג. זהו צילום מסך של חלון PORTS כפי שהוגדר באמצעות תוכנה גרפית Configuration Wizard 2.



ציינו מדוע סומנו P1.0- P1.6 עם האות P.

ד. הוסיפו קוד לתוכנית הבאה, כך שהתוכנית תקלוט תו אחד בקוד ASCII, בודקת האם נקלט ספרה עשרונית במידה וכן יש להציג את הספרה העשרונית על גבי תצוגת 7segment. במידה ונקלט תו לא מספרי יוצג הסימן – בתצוגה. לפני קליטת התו בתקשורת הטורית יש לכבות את התצוגה.

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

code unsigned char table[11] = {0x3f, 0x06, 0x5b, 0x4f, 0x66,
                                0x6d, 0x7d, 0x07, 0x7f, 0x6f, 0x40};

void main(void) {
    Init_Device();
    while (1) {
        if (RI0)
        {
        }
    }
}
```

א. $SCON0 = 0x10$ (Serial Port 0 Control) $REN0=1$, מאפשר קליטה

Bit	7	6	5	4	3	2	1	0
Name	S0MODE	-	MCE0	REN0	TB80	RB80	TI0	RI0

ב. $TCON=0x40$ (Timer/Counter Control) $TR1=1$, מפעיל טיימר 1

Bit	7	6	5	4	3	2	1	0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

$TMOD=0x20$ (Timer Mode) $TIMER1$, פועל במוד 2, טיימר Auto-Reload 8 Bits

Bit	7	6	5	4	3	2	1	0
Name	GATE1	C/T1	T1M[1]	T1M[0]	GATE0	C/T0	T0M[1]	T0M[0]

$TH1=0x30$, קביעת קצב לפי הנוסחה:

$$TH1 = 256 - \frac{T1_CLK}{2 \cdot UartBaudRate} = 256 - \frac{SYSCLK / 12}{2 \cdot UartBaudRate} = 256 - \frac{48M / 12}{2 \cdot 9615} = 48 = 0x30$$

ג. האות P (PushPull) מסמן שה-PORT במצב output עם יכולת דחיפת זרם גבוהה במצב '1' לוגי. כיוון שמחובר ל- P1 תצוגת לדים הזקוקים לזרם של מספר mA, דרוש שה-PORT יוכל לספק. במצב רגיל ה-PORT יכול לספק זרם של עד $15\mu A$ ובמצב P יכול לספק זרם של עד $10mA$.

ד.

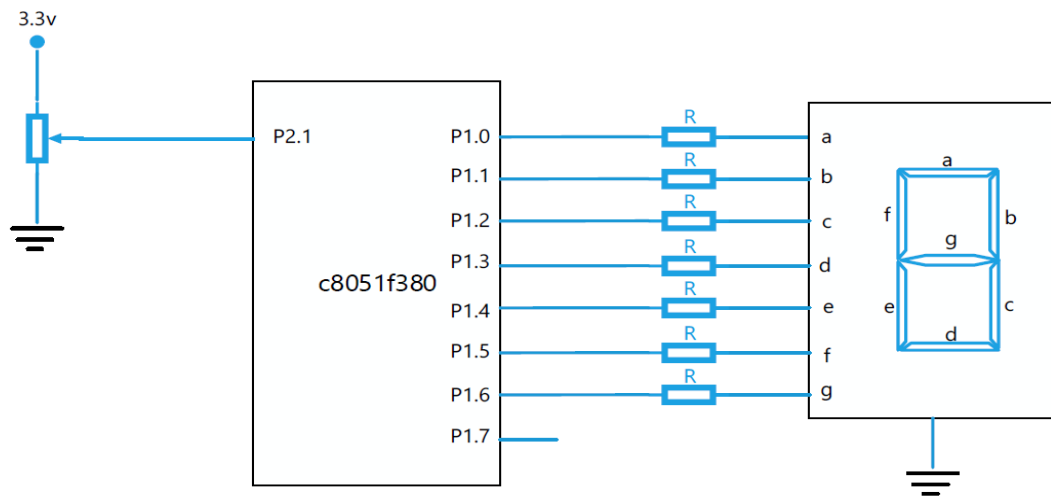
```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

code unsigned char table[10] = {0x3f, 0x06, 0x5b, 0x4f, 0x66,
                                0x6d, 0x7d, 0x07, 0x7f, 0x6f};

void main(void) {
    char tav;
    Init_Device();
    P1=0x00;
    while (1) {
        if (RI0)
        {
            RI0=0;
            tav=SBUF0;
            if ((tav>='0') && (tav<='9'))
                P1=table[tav-'0'];
            else
                P1=0x40;
        }
    }
}
```

נתונה המערכת הבאה כוללת מיקרו-בקר C8051F380, תצוגת 7 Segment בחיבור קתודה משותפת ופוטנציומטר המחובר להדק P2.1.



המערכת דוגמת (מודדת) את ערך הפוטנציומטר ומציגה נתונים על התצוגה בהתאם לטבלה הבאה:

$V_{P2.1}$	הצגה - 7Segment
$V_{P2.1} < 1V$	
$1V \leq V_{P2.1} \leq 2.5V$	תצוגה כבויה
$V_{P2.1} > 2.5V$	

המתח האנלוגי נמדד באמצעות ממיר ADC עם מתח יחוס של 3.3v

נתון חלק מהתוכנית הכוללת פונקציית קריאה מממיר ה-ADC

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
```

```
void Init_Device(void);
int ADC_IN();
```

```
void main(){
    Init_Device();
    while(1) {

    }
}
```

```
int ADC_IN() {
    AD0BUSY = 1;
    while (!AD0INT);
    AD0INT = 0;
    return (ADC0);
}
```

א. סמנו בחלון הבא את הדרוש לאתחול ממיר ה-ADC

ב. סמנו בחלון הבא את הדרוש לאתחול מתח הייחוס

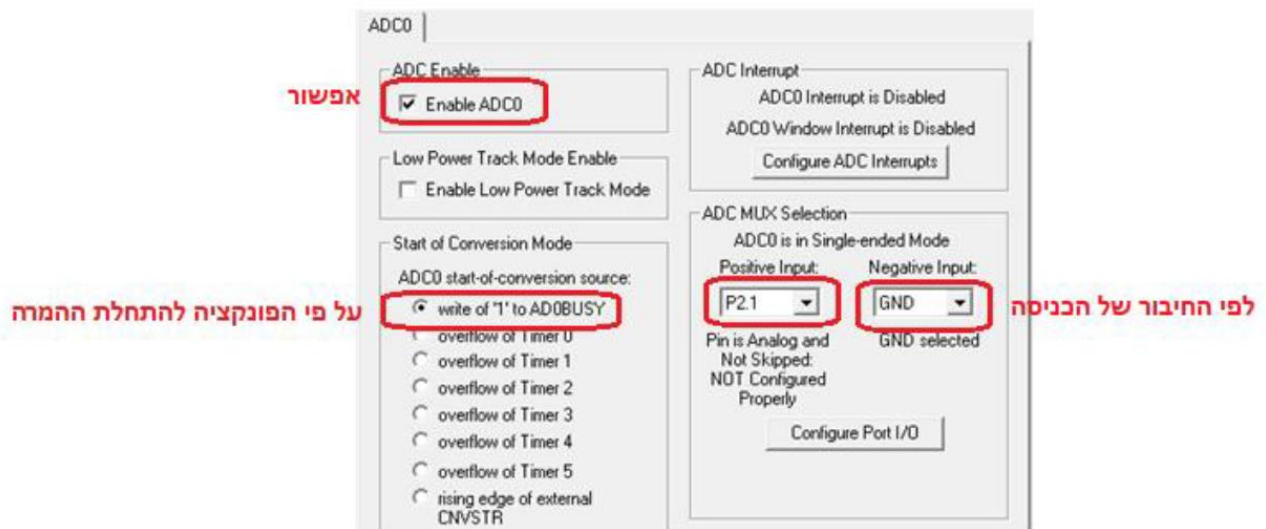


ג. הסברו את הסימונים A ו-P בפורטים ולמה נועדו במערכת

P0								P1								P2							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	A	D	D	D	D	D	D
0	0	0	0	0	0	0	0	P	P	P	P	P	P	P	0	0	0	0	0	0	0	0	0

ד. השלימו את התוכנית הראשית לקריאת הערך האנלוגי ולהצגה לפי הטבלה הנ"ל.

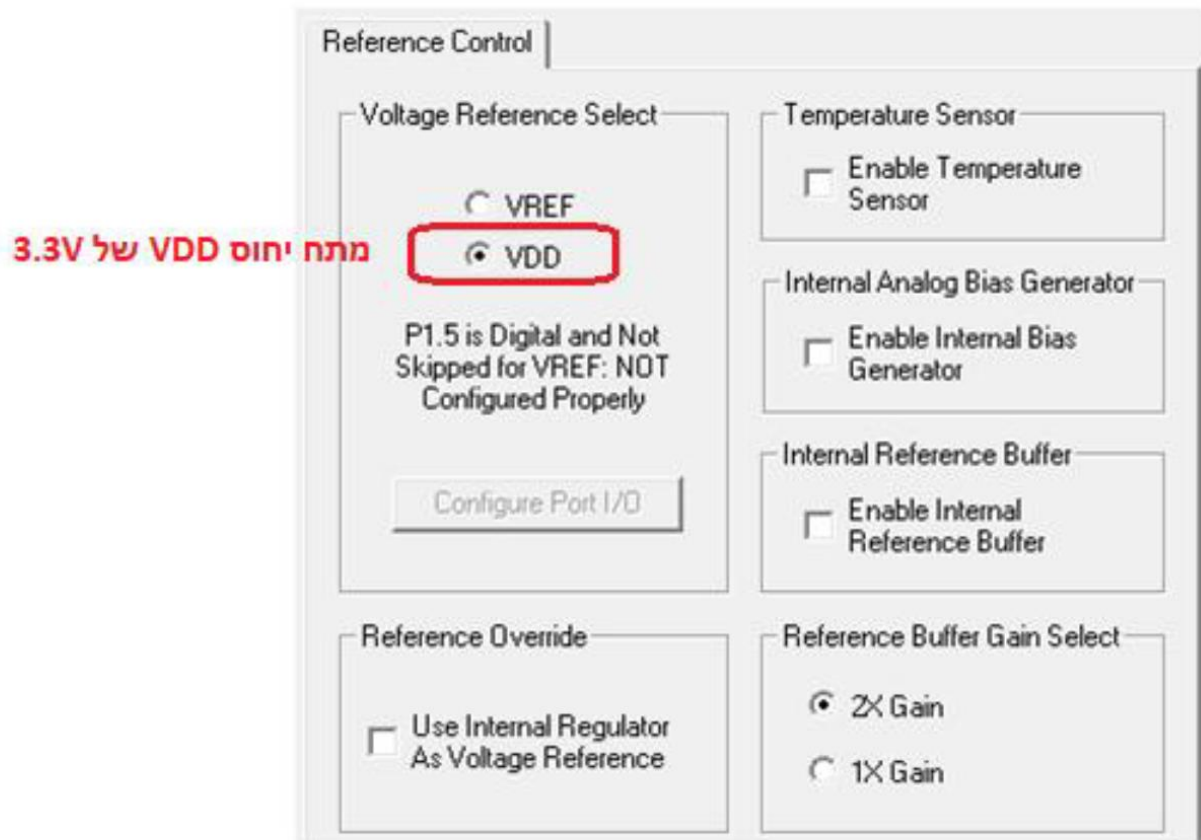
א. להלן תמונת מסך של חלון יחידת ADC לאחר האתחול:



להלן השלבים שיש לבצע:

1. אפשרנו עבודה עם יחידת ADC ע"י אפשרור סיבית ADC Enable.
2. בחרנו באפשרות Write of '1' to AD0BUSY
3. בחרנו בהדק P2.1 בהדק MUX0P (Positive Input)
4. בחרנו בהדק GND בהדק MUX0N (Negative input).

ב. להלן תמונת מסך של חלון יחידת Voltage Reference לאחר האתחול:



בחלון זה בחרנו שמתח הייחוס יחובר למתח פנימי של VDD (3.3V)

ג. האות P (PushPull) מסמן שה-PORט במצב output עם יכולת דחיפת זרם גבוהה במצב '1' לוגי. כיוון שמחובר ל-P1 תצוגת לדים הזקוקים לזרם של מספר mA, דרוש שה-PORט יוכל לספק. במצב רגיל ה-PORט יכול לספק זרם של עד 15μA ובמצב P יכול לספק זרם של עד 10mA. האות A מסמן שהכניסה אנלוגית.


```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);
int ADC_IN();

// AdcValue= Vin/(3.3/1024)
// Vin=1V    => AdcValue=310
// Vin=2.5V => AdcValue=775

void main(){
    int AdcValue;
    Init_Device();

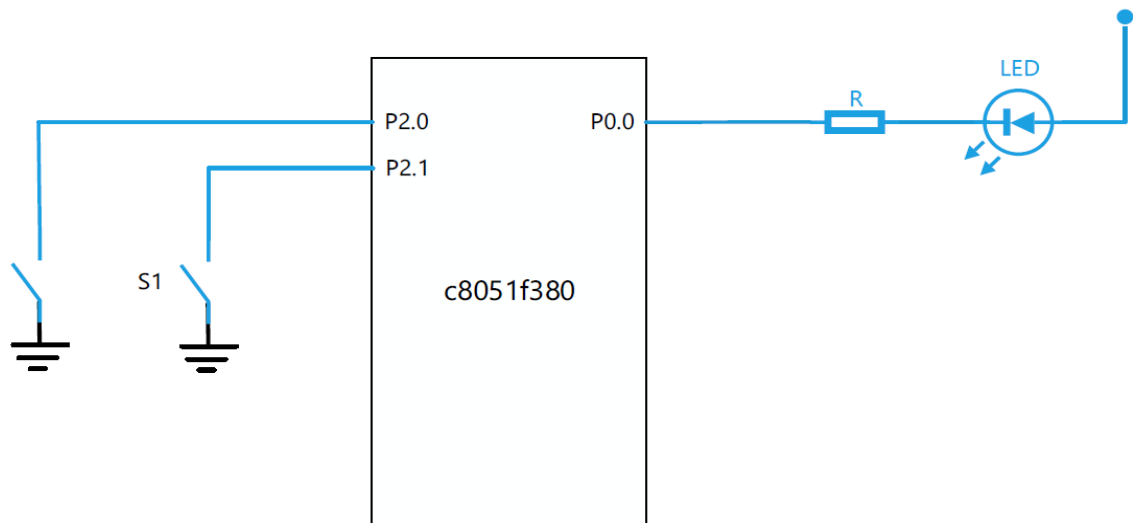
    while(1) {
        AdcValue=ADC_IN();
        if(AdcValue<310)
            P1=0X38;        //L 7Segment
        else if(AdcValue>775)
            P1=0X76;        //H 7Segment
        else
            P1=0X00; // 7Segment off
    }
}

int ADC_IN()
{
    AD0BUSY = 1;
    while (!AD0INT);
    AD0INT = 0;
    return (ADC0);
}
```



79. PCA למוד PWM

נתונה המערכת הבאה כוללת מיקרו-בקר C8051F380, שני לחצנים ללא ריטוטים המחוברים להדק P2.0 וללהדק P0.1 ונורת LED המחוברת להדק P0.0 במופעלת באפנון PWM.



עוצמת ההארה של ה-LED תלויה במצב המפסקים לפי הטבלה הבאה:

מפסק S1	מפסק S0	עוצמת ה-LED
סגור	סגור	עוצמה מינימלית
סגור	פתוח	1/3 מהעוצמה מקסימלית
פתוח	סגור	2/3 מהעוצמה מקסימלית
פתוח	פתוח	עוצמה מקסימלית

א. הסברו את הסימון בחלון הבא:

	Pin I/O	P0								
		0	1	2	3	4	5	6	7	
UART0 <input type="checkbox"/>	TX0									
	RX0									
	SCK									
SPI0 <input type="checkbox"/>	MISO									
	MOSI									
	NSS									
SMBus0 <input type="checkbox"/>	SDA									
	SCL									
CP0 <input type="checkbox"/>	CP0									
	CP0A									
CP1 <input type="checkbox"/>	CP1									
	CP1A									
SYSClk <input type="checkbox"/>	SYSClk									
PCA <input checked="" type="checkbox"/>	CEX0									
	CEX1									
	CEX2									

ב. סמנו בחלונות הבאים את הדרוש לאתחול יחידת - PCA ליצירת PWM בגודל 8 סיביות:

Programmable Counter Array

PCA0 | Module 0 | Module 1 | Module 2 | Module 3 | Module 4/WDT

PCA Counter/Timer Run

☐ Enable PCA0

PCA Counter/Timer Idle Control

☐ Suspend PCA operation while system controller is in Idle Mode

PCA Counter/Timer Initial Value (Hex)

PCA0H (High Byte): 00 PCA0L (Low Byte): 00

PCA Interrupt

☐ Enable PCA Counter/Timer Overflow Interrupt (occurs when CF is set)

PCA0 Interrupt is Disabled
Configure PCA Interrupts

PCA Timebase Input

☒ SYSCLK / 12
☐ SYSCLK / 4
☐ Timer 0 Overflow
☐ High to Low Transitions on ECI
☐ SYSCLK
☐ External Oscillator / 8

Programmable Counter Array

PCA0 | Module 0 | Module 1 | Module 2 | Module 3 | Module 4/WDT

Capture/Compare Modules

☒ None
☐ Capture on positive edge of CEX0
☐ Capture on negative edge of CEX0
☐ Capture on transition of CEX0
☐ Software Timer
☐ High Speed Output
☐ Frequency Output
☐ 8-Bit Pulse Width Modulator
☐ 16-Bit Pulse Width Modulator

PCA Capture/Compare Register (Hex)

PCA0CPH0 (High Byte): 00 PCA0CPL0 (Low Byte): 00

CEX0 on the Crossbar

CEX0 is Not Enabled on the Crossbar
Configure Port I/O

Capture/Compare Interrupt Enable

☐ Enable CCF Interrupt

ג. הסברו איזה אוגר קובע את ה-PWM עבור 8 סיביות.

ד. נתון בחלון הנ"ל $PCA\ Timebase = SYSCLK / 12$ ותדר שעון המעבד הוא 48MHz, מהו תדו אות ה-PWM.

ה. השלימו את התוכנית הבאה לפי דרישות המערכת.

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

sbit S0=P2^0;
sbit S1=P2^1;

void main() {
    Init_Device();

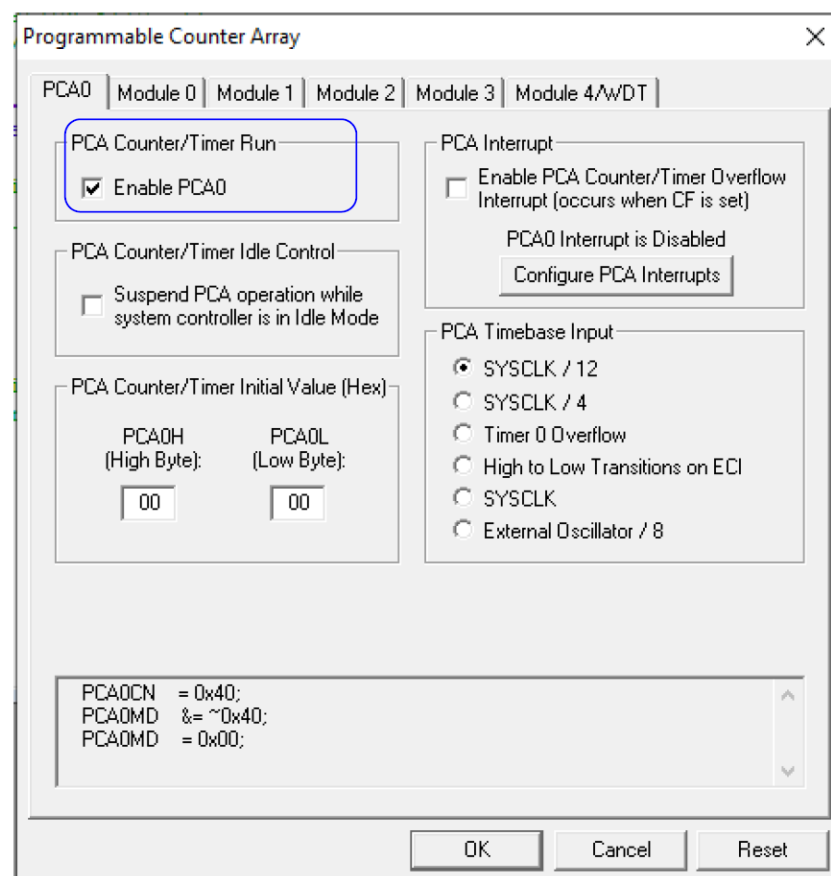
    while(1) {

    }
}
```

פתרון (PWM)

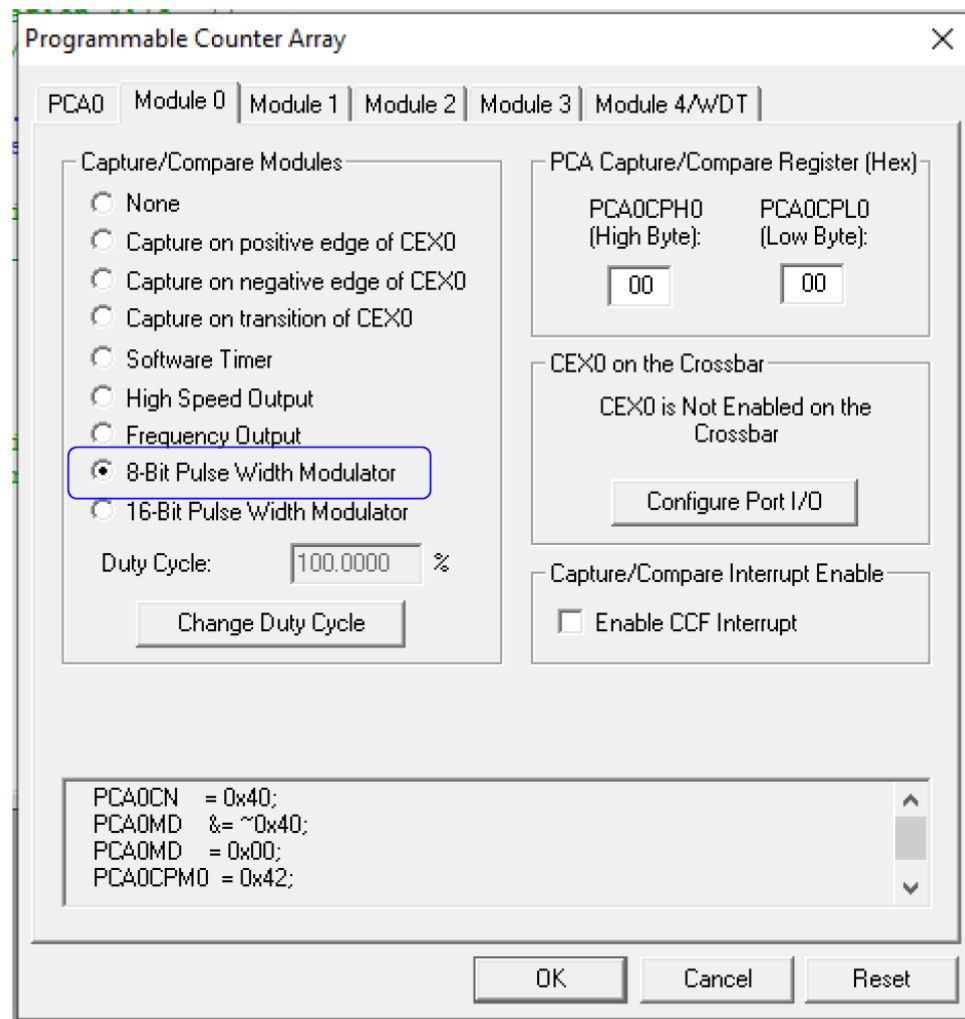
א. קביעת הדק P0.0 כהדק יציאת ה-PWM עבור PCA0

להלן תמונת מסך של חלון לשונית PCA0



ניתן לראות שאפשרנו את יחידת PCA ע"י אפשרור סיבית Enable PCA0

להלן תמונת מסך של חלון לשונית Module 0



ניתן לראות שאפשרנו את אפשרות של 8 Bit Pulse Width modulator.

ב. אוגר PCA0CPH0 קובע את ה- Duty Cycle של אות ה-PWM. האות פעיל בנמוך, כלומר ככל שהערך גדול יותר מצב ה-יסי' לוגי גדל.

ג. מחזור מניה של מונה ה-PCA הוא 256 לכן על כל 256 מחזורי שעון של PCA Timebase, נקבל מחזור אחד של גל המוצא.

$$f_{PWM} = \frac{SYSCLK/12}{2^8} = \frac{48M/12}{256} = 15625Hz$$

ד. ה- LED פעיל בנמוך וגם ה-PWM לכן ערך שניתן לאוגר PCA0CPH0 יקבע ביחס ישר את עוצמת הארת ה-LED

תרשים זמנים עבור ערכים שונים של PCA0CPH0 הקובע את ה-Duty Cycle
רזולוציית הזמן של ה- Duty Cycle היא לפי תדר $SYSCLK/12=4MHz$, כלומר 0.25usec

מפסק S1	מפסק S0	עוצמת ה- LED	PCA0CPH0	צורת הגל ב- P0.0
סגור	סגור	עוצמה מינימלית	0	
סגור	פתוח	1/3 מהעוצמה מקסימלית	85	
פתוח	סגור	2/3 מהעוצמה מקסימלית	170	
פתוח	פתוח	עוצמה מקסימלית	255	

תוכנית

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

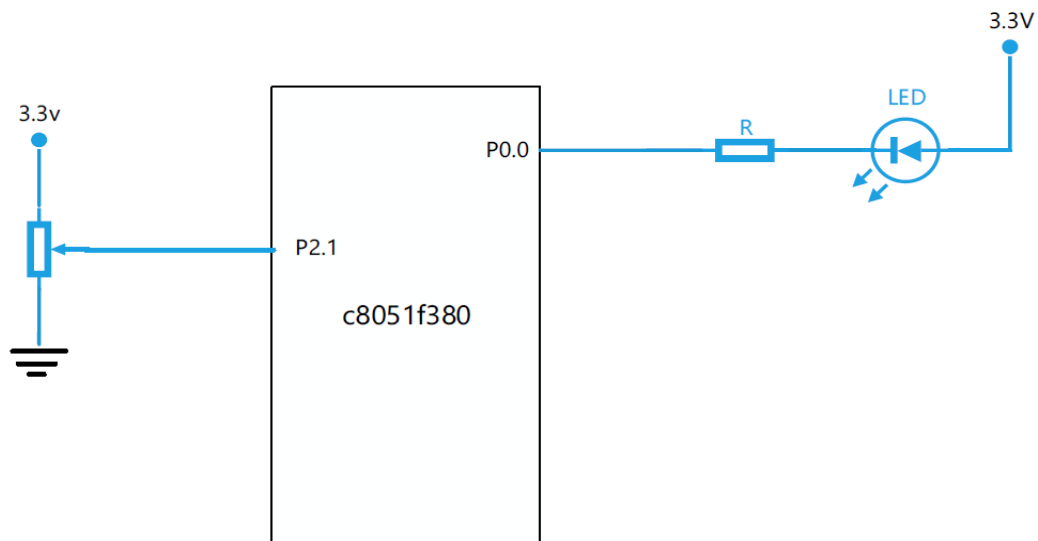
sbit S0=P2^0;
sbit S1=P2^1;

// minimum PCA0CPH0=0, maximum PCA0CPH0=255
// 1/3 PWM PCA0CPH0=85 2/3 PWM PCA0CPH0=170

void main(){
    Init_Device();

    while(1){
        if((S1==0) && (S0==0))
            PCA0CPH0 = 0;
        else if((S1==0) && (S0==1))
            PCA0CPH0 = 85;
        else if((S1==1) && (S0==0))
            PCA0CPH0 = 170;
        else
            PCA0CPH0 = 255;
    }
}
```

נתונה המערכת הבאה כוללת מיקרו-בקר C8051F380, פוטנציומטר המחובר להדק P2.1 ונורת LED המחוברת להדק P0.0 במופעלת באפנון PWM.



עוצמת ההארה של ה-LED תלויה במתח האנלוגי בהדק P2.1, במתח 0 עוצמה מינימלית ובמתח 3.3V עוצמה מקסימלית.

נתון שהבקר מאותחל, ממיר ה-ADC פועל לפי מתח יחוס של 3.3V, מאופשר ומחובר ל-P2.1, מעגל ה-PCA מאופשר ופועל לפי 8 סיביות.

א. הסברו את הדגלים AD0BUSY ו-AD0INT של ממיר ה-ADC.

ב. כתבו פונקציה לממיר ה-ADC המחזירה את ערך הממיר. השתמש בדגלים הנ"ל.

ג. הסברו איזה אוגר קובע את ה-PWM עבור 8 סיביות.

ד. כתבו תוכנית למימוש המערכת במבנה הבא:

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

void main() {
    Init_Device();

    while(1) {

    }
}
```

א. דגל AD0BUSY משמש להתחלת ההמרה של ממיר ה-ADC.
 דגל AD0INT משמש לבדיקת סיום המרה של ממיר ה-ADC.

ב. פונקציית הממיר

```
int ADC_IN()
{
    AD0BUSY = 1;           // התחל המרה
    while (!AD0INT);       // חכה לסיום המרה
    AD0INT = 0;            // אפס את הדגל להמרה הבאה
    return (ADC0);         // החזר את ערך הממיר
}
```

ג. אוגר PCA0CPH0 קובע את ה- Duty Cycle של אות ה-PWM.
 האות פעיל בנמוך, כלומר ככל שהערך גדול יותר זמן מצב ה-0' לוגי גדל.
 ד. להלן קוד התוכנית לאחר הוספה קוד בהתאם לדרישות:

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);
int ADC_IN();

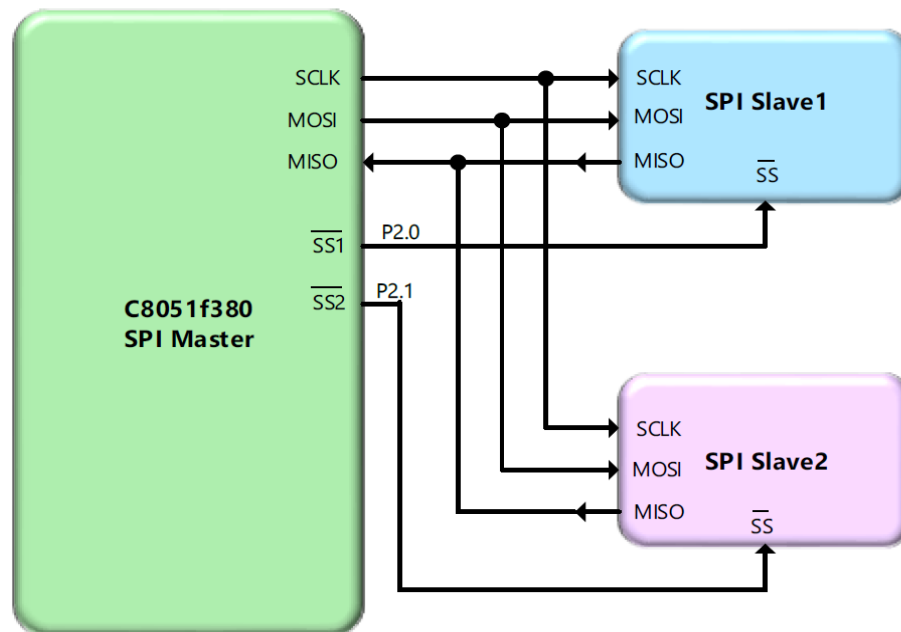
// AdcValue= Vin/(3.3/1024)

void main() {
    int AdcValue;
    Init_Device();

    while(1) {
        AdcValue=ADC_IN(); //max 1023
        PCA0CPH0=AdcValue/4; //maxPWM=1023/4=255
    }
}

int ADC_IN() {
    AD0BUSY = 1;           // התחל המרה
    while (!AD0INT);       // חכה לסיום המרה
    AD0INT = 0;            // אפס את הדגל להמרה הבאה
    return (ADC0);         // החזר את ערך הממיר
}
```


נתונה המערכת הבאה הכוללת את מיקור-בקר C8051F380 המשמש כרכיב Master ושני רכיבי Slave המתקשרים בפרוטוקול תקשורת טורית SPI



דרוש לקרוא מידע של 8 סיביות מרכיב Slave1 ולשדר אותו לרכיב Slave2 בלולאה אינסופית.

נתונה התוכנית הבאה הכוללת פונקציות קריאה וכתיבה בפרוטוקול SPI ואתחול המעבד.

```
#include"compiler_defs.h"
#include"C8051F380_defs.h"
void Init_Device(void) ;

void SpiWrite(unsignedchar Value) {
    SPIF = 0;
    SPI0DAT = Value;

    while(!SPIF);
    SPIF = 0;
}

unsignedchar SpiRead(){
    SPIF = 0;
    SPI0DAT = 0;
    while (!SPIF);
    SPIF = 0;
    return SPI0DAT;
}

void main(void) {
    Init_Device();
    while(1)
    {
    }
}
```

א. זהו צילום מסך של חלון של יחידת SPI0

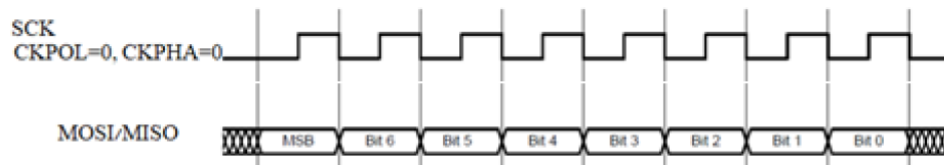
הסברו את הסימונים המוקפים 1-5.

- ב. הסברו את פונקציית SpiWrite ופונקציית SpiRead
- ג. הסברו תפקיד הדגל SPIF בשידור ובקליטה.
- ד. השלימו את קוד התוכנית הראשית לפי הדרישות הנ"ל.

פתרון

א. 1- הבקר פועל כ-MASTER

2+3 – מתאים לתיאור הבא : פועל בעלייה ראשונה של השעון וערכו הוא '0' במצב idle



- 4 – מצב עם 3 קווים (SCLK, MISO, MOSI) בלבד כאשר הדק אפשר SS יהיה בתוכנה.
- 5 – קביעת תדר השעון ל- 12MHz

- ב. פונקציית SpiWrite – רגיסטר SPI0DAT נטען בערך Value ומתחיל לשדר את המידע בצורה טורית. הפונקציה ממתינה לסיום על ידי דגל SPIF.
- פונקציית SpiRead – הפונקציה ממתינה לסיום קליטה על ידי בדיקת דגל SPIF ומזירה את הערך הנקלט.

ג. תפקיד דגל SPIF בשידור הוא לציין ששודר מידע של 8 סיביות ובקליטה לציין שנקלט מידע של 8 סיביות.

ד. להלן קוד התוכנית לפי הדרישות:

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"

void Init_Device(void);

sbit SS1= P2^0;
sbit SS2= P2^1;

void SpiWrite(unsignedchar Value) {
    SPIF = 0;
    SPI0DAT = Value;

    while(!SPIF);
    SPIF = 0;
}

unsignedchar SpiRead() {
    SPIF = 0;
    SPI0DAT = 0;
    while (!SPIF);
    SPIF = 0;
    return SPI0DAT;
}

void main(void) {
    unsignedchar SPI_data;
    Init_Device();
    SS1=1;
    SS2=1;

    while(1) {
        SS1=0;
        SPI_data= SpiRead();
        SS1=1;

        SS2=0;
        SpiWrite(SPI_data);
        SS2=1;
    }
}
```


ה. נתונות הפונקציות הבאות לתקשורת בפרוטוקול I²C:

- התחלת תקשורת

```
void startI2c()
```

- סיום תקשורת

```
void stopI2c()
```

- שליחת מידע בגודל 8 סיביות לרכיב I²C והחזרה של סיבית ack

```
bit writeByteI2c(unsigned char outchar)
```

- קריאה מרכיב I²C

```
unsigned char readByteI2c()
```

כתוב תוכנית הקוראת מרכיב I²C בכתובת 0x40 מידע של 8 סיביות ושולחת אותו לרכיב אחר בכתובת 0x50 בלולאה אינסופית.

פתרון

- א. יתרון – באמצעות 2 קווים ניתן לתקשר עם מספר רכיבים וב- UART רק רכיב אחד.
חסרון – תקשורת חד כיוונית - Half Duplex לעומת - UART Full Duplex
- ב. יתרון – באמצעות 2 קווים ניתן לתקשר עם מספר רכיבים לעומת SPI הדורש 3 קווים ועוד קווי אפשר לכל רכיב.
- חסרון – מהירות נמוכה עד 400Kbps לעומת SPI היכול להגיע לקצב של עשרות Mbps.
- ג. 1. לא, רק עם רכיב אחד בכל רגע.
2. שולח start, כתובת בגודל 7 סיביות, ביט לכתיבה = '0', בדיקת ביט אישור ack מה-SLAVE, שליחת מידע של 8 סיביות, שליחת end.
3. לכל רכיב Slave יש כתובת ייחודית לו.
4. קווי הכתובת מכילים 7 סיביות ולכן 128 רכיבים בעלי כתובת שונה (בהנחה שלא שמורים כתובות)

ד.

1. כתובת הרכיב $1110111 = 0x77$
2. לאחר הכתובת הערך הוא '0' לוגי, לכן w-כתיבה, הערך הוא $11110110 = 0xF6$
3. כן – אחרי ביט r/w ערך ה- '0' ack.

ה.

```
#include "compiler_defs.h"
#include "C8051F380_defs.h"
void Init_Device(void);
void startI2c(void);
void stopI2c(void);
bit writeByteI2c(U8 outchar);
U8 readByteI2c();

void main(void) {
    U8 I2C_data;
    Init_Device();

    while(1) {
        startI2c();
        if(!writeByteI2c((0x40<<1) | 1)) // 7bit Address+r/w=1
        {
            I2C_data=readByteI2c();
        }
        stopI2c();

        startI2c();
        if(!writeByteI2c(0x50<<1)) // 7bit Address+ r/w=0
        {
            writeByteI2c(I2C_data);
        }
        stopI2c();
    }
}
```

עבודה מהנה