

מהלך העבודה עם כרטיס המיקרו בקר AT89C5131 –

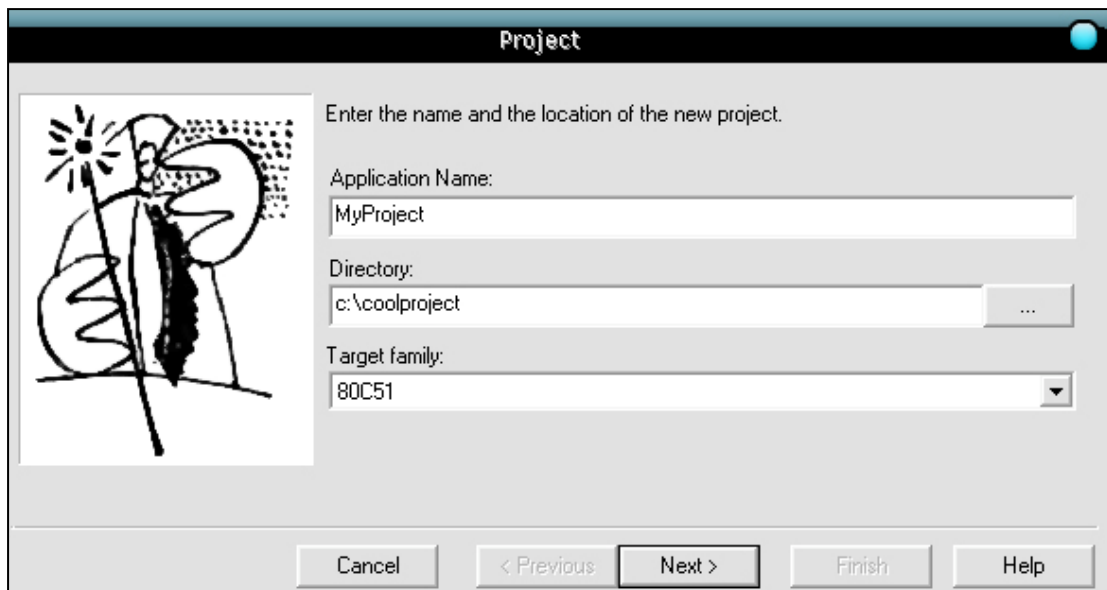
כדי להכיר את התוכנה והחומרה של המיקרו בקר, אנו נבצע כמה ניסויים שיעזרו לנו להכיר את סביבת העבודה. בניסויים נשתמש בכרטיס המיקרו בקר ונתרגל שימוש בשפת בשפת תכנות, כתיבת תוכניות והרצתן. התוכניות נכתבות בשפת C51 שהיא שפת C ייעודית למיקרו בקר.

כמו כן, כדי לכתוב את התוכניות ולהעבירן למיקרו בקר השתמשנו בתוכנות הבאות:
Raisonance Kit 6.1(RIDE) – המהדר לכתובת התוכניות ותרגומן לשפת המכונה של המעבד. תוכנה זו קיימת בדיסק התקנה תחת התיקיה compiler
FLIP 2.4.4 – להרצת התוכניות וצריבתן למיקרו בקר. תוכנה זו קיימת בדיסק התקנה תחת התיקיה flip-2_4_4

הערה: תוכנות אילו קיימות בדיסק התקנה שקבילתן בעת רכישת כרטיס המיקרו-בקר AT89C5131

א. הפעלת המהדר ויצירת פרויקט חדש:

1. יצירת תיקייה חדשה לפרויקט בכונן C.
2. הפעלת תוכנת המהדר RIDE באמצעות:
Start>All Programs> Raisonance Kit 6.1> RIDE IDE
3. יצירת פרויקט חדש ע"י: Project>New
4. בחלון שנפתח מקלידים את שם הפרויקט ומיקום התיקיה.
5. בשורה Target Family להשאיר את הערך ההתחלתי, 80C51, ולאחר מכן Next.



6. בחלון שנפתח בוחרים בתיקיה Atmel ואז את הרכיב AT89C5131 ולוחצים finish

ב. כתיבת התוכנית:

1. File>New ולאחר מכן "C Files".
2. השורה הראשונה תהיה:

```
#include<at89c5131.h>
```

שורה זו היא הוראה למהדר להכליל בתוכנית, בזמן ביצוע ההידור, הגדרות מיוחדות למעבד שאיתו אנו עובדים לדוגמא: עבודה עם הפרטים שלו.

3. כתיבת תוכנית הגורמת לנוריות ה-LED המחוברת ל-P1 להבהב:

```
#include<at89c5131.h>
void main()
{
    unsigned int i; //ההשהיה לולאת לצורך
    while(1) //לולאה אינסופית
    {
        P1=0xff; //1 בפורט
        for(i=0;i<=50000;i++); // לולאת השהיה
        P1=0x00; // 1 בפורט
        for(i=0;i<=50000;i++); // לולאת השהיה
    }
}
```

4. שמירת התוכנית בשם באמצעות: File>Save. יש לשמור את הקובץ עם סיומת c לדוגמא: main.c רצוי לבחור שם שונה מזה של שם הפרויקט.
5. בשלב זה יש לצרף את התוכנית לפרויקט:
- א. בתפריט Project בוחרים באפשרות: "Add node Source/Application".
- ב. בחלון שנפתח בוחרים את הקובץ שאותו שמרנו.
6. בשלב זה מתבצע ההידור:

בתפריט Project בוחרים ב-"Build all". בסיום תהליך התרגום המהדר ידווח על שגיאות(אם יש כאלה) בחלון שבתחתית המסך. כל עוד קיימות שגיאות יש לתקן, לשמור ולחזור על פעולות ההידור, עד שמתקבלים V ירוקים בתחתית החלון.

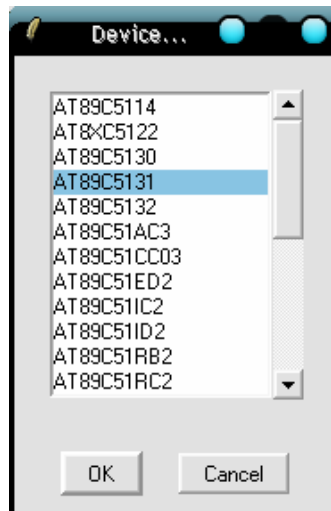
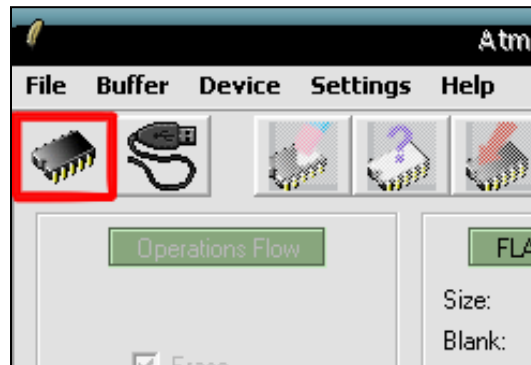
ג. צריבת התוכנית בזיכרון המעבד:

אופן עבודה עם תוכנת FLIP:

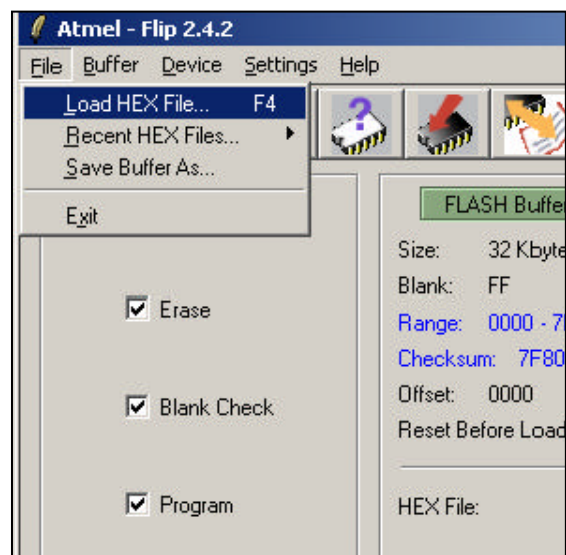
1. פותחים את תוכנית הצריבה באמצעות:

Start>All Programs>Atmel>Flip 2.4.4>flip.exe

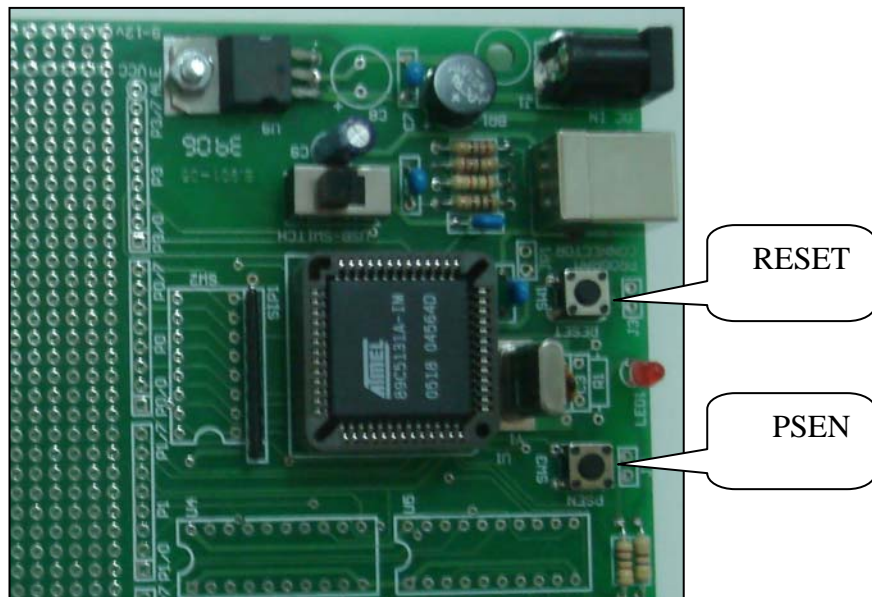
2. לוחצים על הצלמית השמאלית ובחרים במעבד AT89C5131.



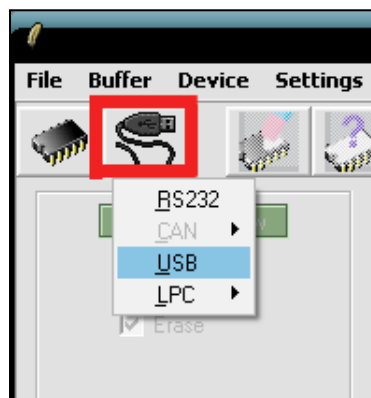
3. בוחרים את הקובץ לצריבה (עם סיומת HEX) באמצעות File>Load HEX file או הקשה על F4.



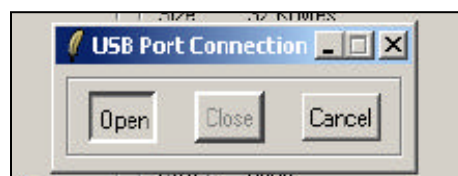
4. לוחצים על מקש RESET ועל מקש PSEN במיקרובקר. משחררים את המקש RESET, ולאחר מכן את המקש PSEN.



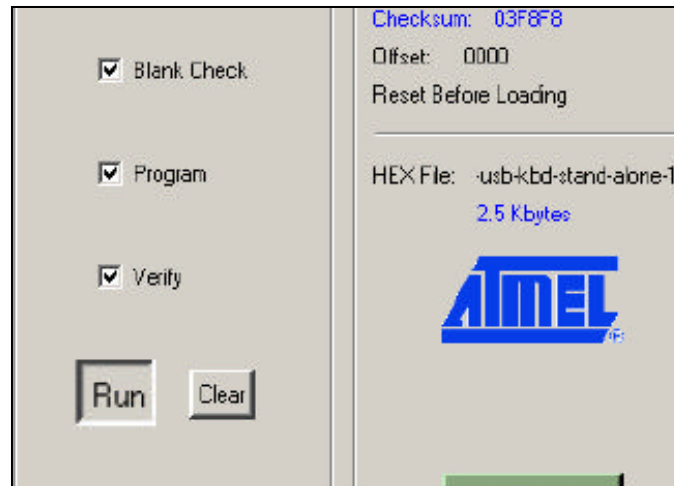
5. מחברים את כרטיס המיקרובקר למחשב באמצעות קבל USB.
6. לוחצים על צלמית הכבל בתוכנת הצריבה, ובוחרים בחלון שנפתח ב-USB.



7. בחלון שנפתח מקישים Open ולאחר מכן Cancel



8. בשלב זה המערכת מוכנה לצריבה. לוחצים על מקש Run בפינה השמאלית התחתונה של המסך.

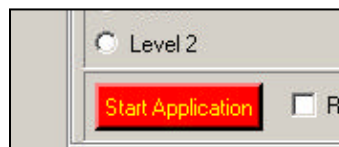


**יש להקפיד שהמשבצת BLJZ במסך אינה מסומנת.



9. לאחר מכן מסתיימת הציבה, בפינה השמאלית התחתונה של המסך מופיעה ההודעה "Memory verify pass".

10. כדי להריץ אץ התוכנית לוחצים על המקש "Start Application", או על מקש RESET שבכרטיס המיקרובקר. אם הכול תקין הנוריות שבפיתחה 1 צריכות להבהב.



דוגמאות נוספות להרצת תוכניות בסביבת עבודה זו:

משנים את התכנית שבקובץ המקור, כך שרק ארבע נוריות תהבהבנה בכל פעם, וקצב ההבהוב יהיה איטי.

1. פותחים את תוכנת המהדר "RIDE".
התוכנה תיפתח ותציג את התוכנית שנכתבה.

2. משנים את התוכנית בהתאם לכתוב:

```
#include<at89c5131.h>
void main()
{
  unsigned int i; //הגדרת משתנה לצורך לולאת ההשהיה//
  while(1) //לולאה אינסופית//
  {
    P1=0x0f; //הדלקת 4 הלדים בפורט 1//
    for(i=0;i<=50000;i++); // לולאת השהיה //
    P1=0x00; // כיבוי הלדים בפורט 1 //
    for(i=0;i<=50000;i++); // לולאת השהיה //
  }
}
```

כתיבת והרצת תוכניות קלט/פלט בסיסיות:

בתרגילים הבאים נתנסה בכתיבת תוכניות הקשורות בהפעלת נוריות הדפ"א המחוברות לפורט 1 ו-2 אשר פועלות כפלט. הנוריות בפיתוחה 1 הן מסוג "קתודה משותפת" (Common Cathode) והדלקתן תיעשה באמצעות שידור של '1' מהמיקרו בקר למוצא הפורט, ואילו הנוריות בפיתוחה 2 הן מסוג "אנודה משותפת" (Common Anode) והדלקתן תיעשה באמצעות שידור של '0'. כמו כן, נתנסה בכתיבת תוכניות בהן מצב הנוריות(הפלט) תלוי במצב המפסקים(הקלט) אשר מחוברים ל-P0. המפסקים מחוברים בצד אחד באופן קבוע להארקה, ובצד השני לאחד ממבואות פיתחת הקלט של המיקרו בקר ולנגד בעל התנגדות של 10K אום.

*הדלקת לד רץ בפורט 1:

```
1) #include<at89c5131.h>
void main()
{
  int i,t,x=0x01; //הגדרתו X והגדרתו
  while(1)//לולאה אינסופית. כך הרצת הלדים תהיה אינסופית
  {
    for(i=0;i<8;i++)
    {
      P1=x; // הצבת הערך שבמשתנה X בפורט 1 //
      for(t=0;t<50000;t++); // לולאת השהיה //
      x=x*2; // הגדלת הערך שבמשתנה X פי שניים //
    }
    x=0x01; // השמת ערך התחלתי בX מחדש //
  }
}
//end main
```

*הדלקת לד רץ בפורט 2:

פורט 2 הינו פורט מסוג "אנודה משותפת" (Common Anode), לכן על מנת להדליק לד מסוים יש להציב '0' לוגי.

```
2) #include<at89c5131.h>
void main()
{
    int i,t,x=0xfe,y=1;// והגדרתם Y,X בשמת ערך התחלתי
    while(1) // לולאה אינסופית. כך הרצת הלדים תהיה אינסופית
    {
        for(i=0;i<8;i++)
        {
            P1=x; // הצבת הערך שבמשתנה X בפורט 1
            for(t=0;t<50000;t++); // לולאת השהייה
            x=x-y; } // יצירת שורש ריבועי על מנת להדליק בכל פעם לד 1 בלבד
            y=y*2; }
        x=0xfe; } // השמת ערך התחלתי בY,X מחדש
        y=1; }
    }
} //main
```

*הדלקת הלדים על פי מצב המפסקים

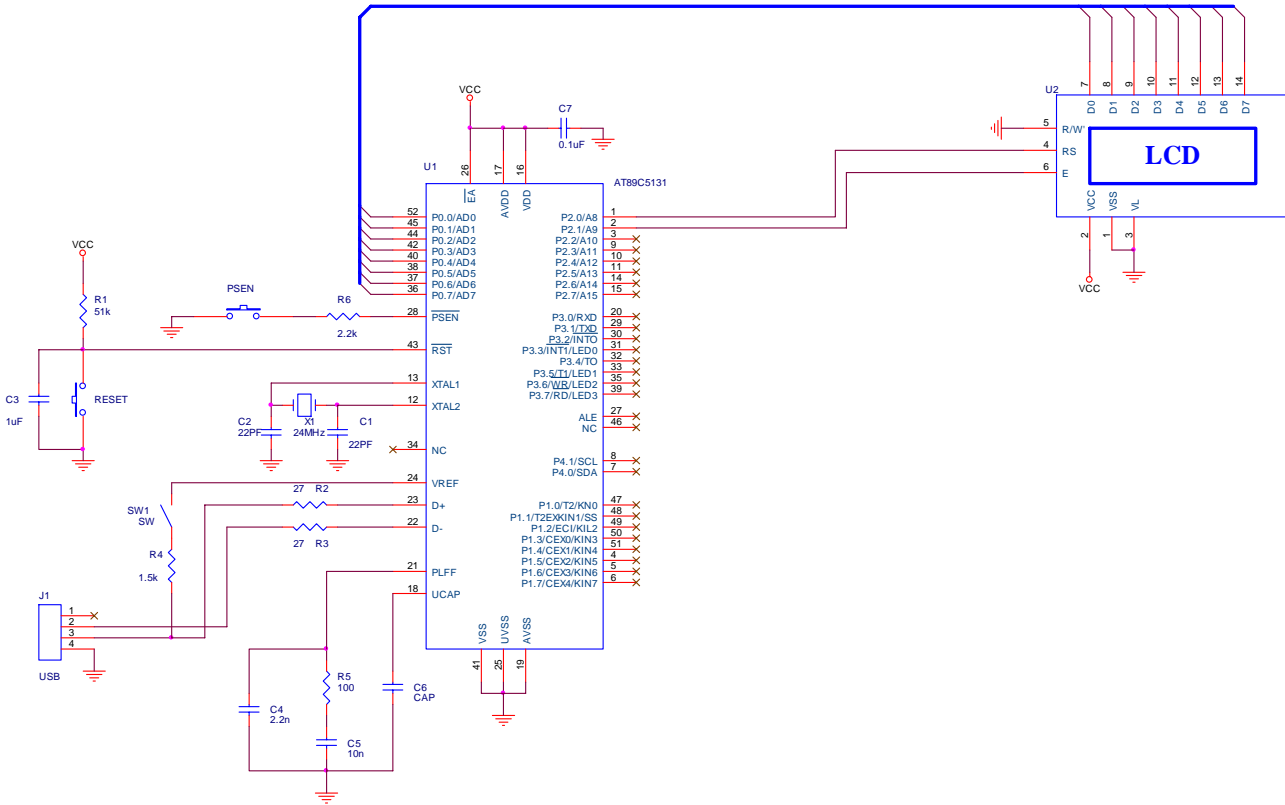
```
3) #include<at89c5131.h>
Void main ()
{
    while(1) // לולאה אינסופית. כך הפעולה תהיה אינסופית
    P1=P0; // הדלקת הלדים בפורט 1 לפי המפסקים בפורט 0
}
}
```

*הדלקת הלדים על פי מצב של פורט P0.0

```
4) #include<at89c5131.h>
void main()
{
  int i,t,x=0x01; // השמת ערך התחלתי בX והגדרתו
  while(1) // לולאה אינסופית. כך הפעולה תהיה אינסופית
  {
    if(P0^0==1) // אם P0.0 שווה ל'1' הרצת הלדים תהיה שמאלה
    {
      for(i=128;i>0;i=i/2)
      {
        P1=i; // הצבת הערך שבמשתנה i בפורט 1
        for(t=0;t<50000;t++); // לולאת השהייה
      }
      i=128; // השמת ערך התחלתי ב i מחדש
    }
    else if(P0^0==0) // אחרת, אם P0.0 שווה ל'1' הרמת הלדים תהיה ימינה
    {
      for(i=0;i<8;i++)
      {
        P1=x; // הצבת הערך שבמשתנה X בפורט 1
        x=x*2; // הגדלת הערך שבמשתנה X פי שניים על מנת להדליק בכל פעם לד אחד בלבד
        for(t=0;t<50000;t++); // לולאת השהייה
      }
      x=0x01; // השמת ערך התחלתי בX מחדש
    }
  }
} //main
```


לאחר שהתנסינו בכתובת תוכניות להתקני קלט/פלט פשוטים כגון : לדים ומפסקים, נכתוב תוכניות מורכבות יותר המיועדות לחומרה המחוברת לכרטיס הפרויקט שלנו :

בקובץ זה אני מביא דוגמא להצגת הודעות על-גבי תצוגת LCD המחוברת למיקרו-בקר לפי השרטוט הבא :



בשרטוט הנ"ל ה-DATA BUS של התצוגה מחובר ל-P0 של המיקרו-בקר, רגל RS של התצוגה מחוברת ל-P2.0 ורגל E של התצוגה מחוברת להדק P2.1. יש לשים לב לחומרה המחוברת בכרטיס על-מנת שהתוכנית תתאים.

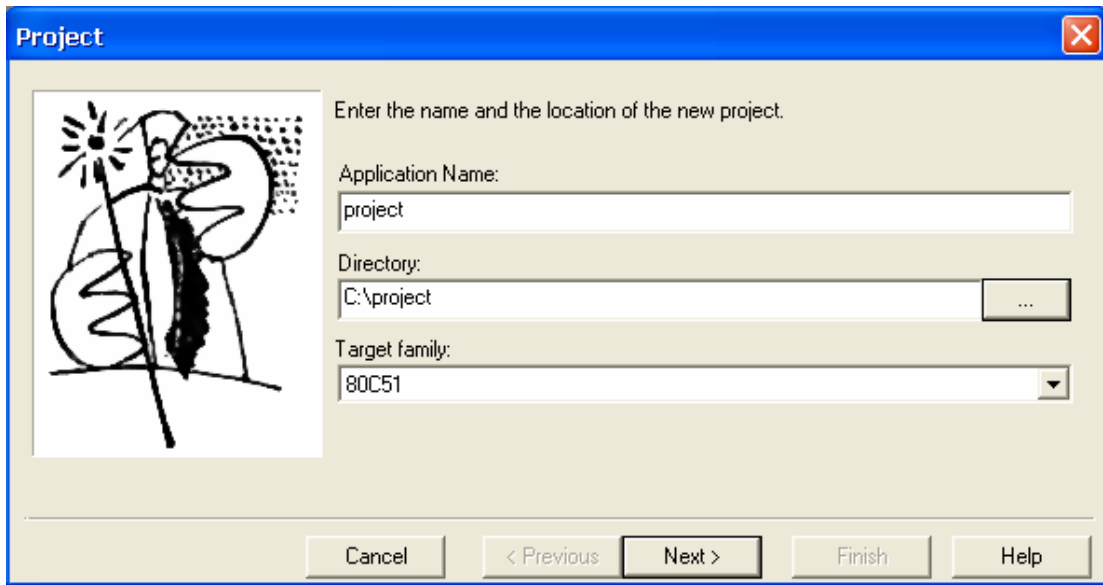
הרעיון בכתובה בשפת C היא לכתוב בצורה מודולרית, תוך שימוש בפונקציות קיימות של התוכנה ושל משתמשים אחרים. על-מנת להקל עליכם בכתובת תוכניות, תקבלו מאיתנו קטעי תוכניות (פונקציות עבור רכיבי חומרה היקפיים) וכל שעליכם לעשות הוא לכתוב את התוכנית הראשית ולשלב את הפונקציות.

ההכרזה על הפונקציות, רשומה בקובץ עם סיומת h והפונקציות עצמן נמצאות בקובץ עם סיומת c

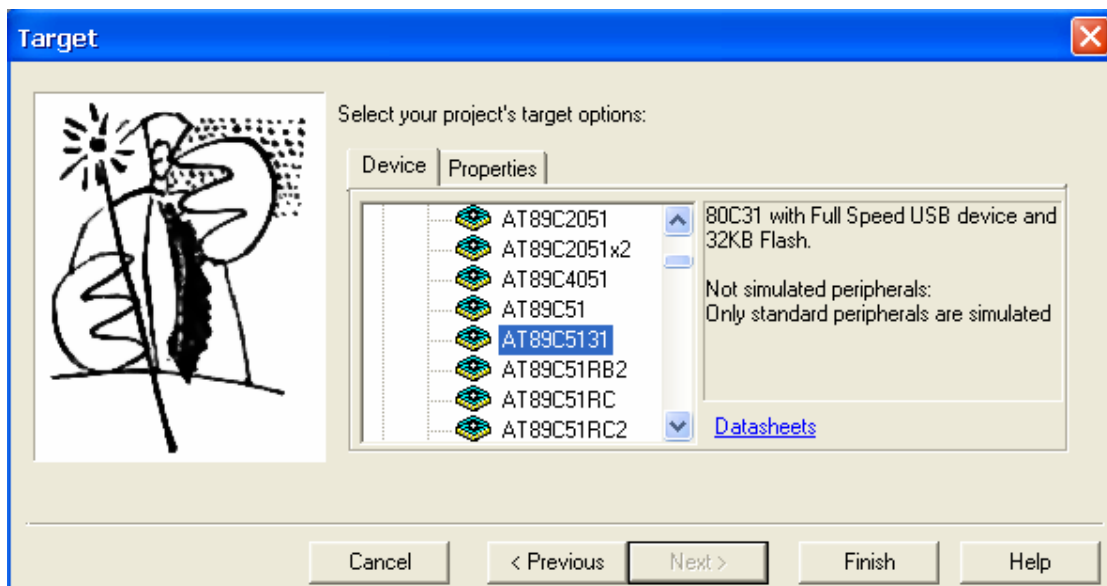
דוגמא לכתובת תוכנית המציגה בשורה 1 במיקום 0 של תצוגת ה-LCD את המילה test, ובשורה השנייה במיקום 5 את המשפט yossi&avi

1. פתח ספרייה חדשה בכונן C של הפרויקט לדוגמא: project ושמו לזכור את קבצי c וקבצי h המכילים את הפונקציות. בדוגמא שלנו אם הפרויקט מכיל רכיב LCD, שמור את הקבצים: lcd_p.c ו-lcd_p.h
2. הפעלת תוכנת המהדר RIDE באמצעות:
Start>All Programs> Raisonance Kit 6.1> RIDE IDE
3. יצירת פרויקט חדש ע"י: Project>New
4. בחלון שנפתח מקלידים את שם הפרויקט ומיקום התיקייה, נבחר בשם project (לא חייב באותו שם).

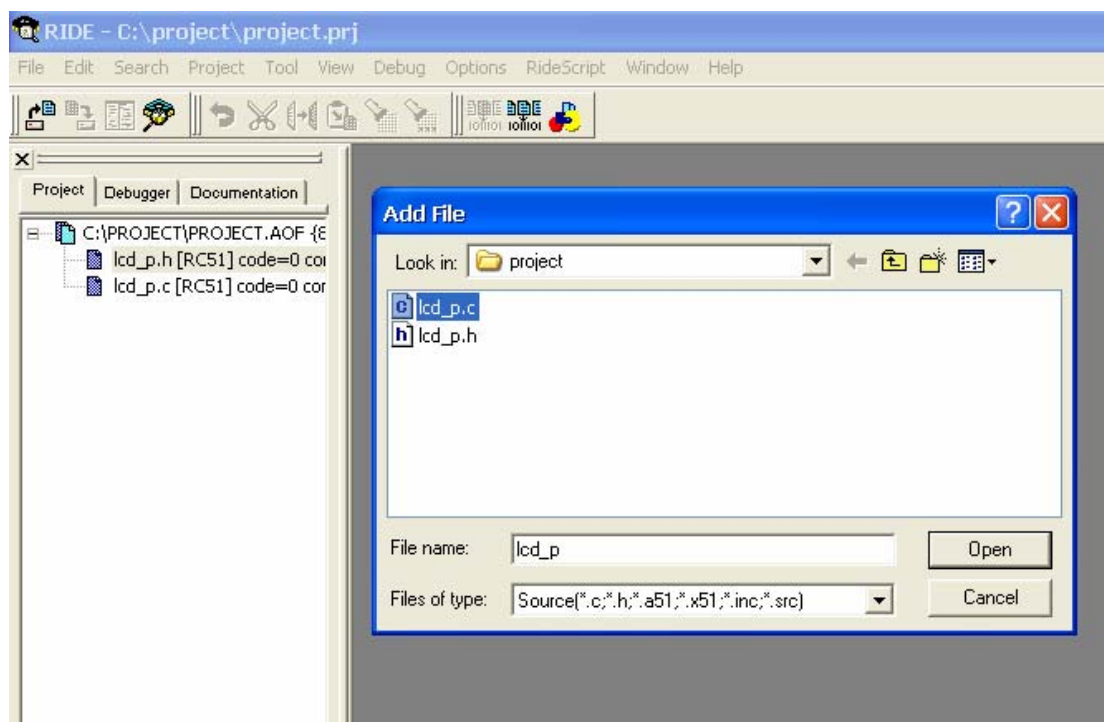
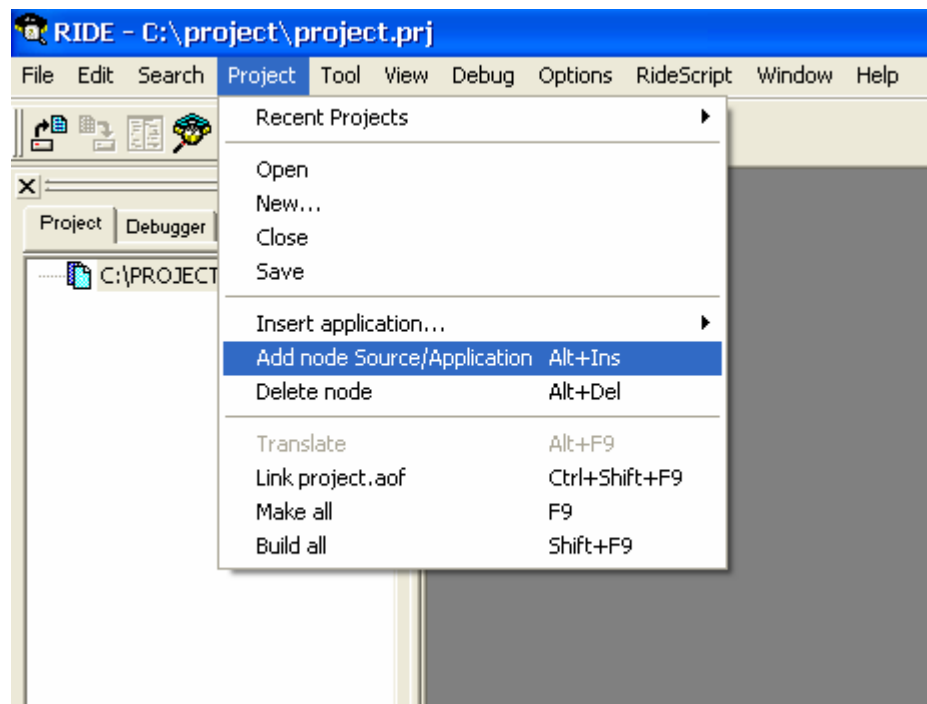
5. בשורה Target Family להשאיר את הערך ההתחלתי, 80C51 ולאחר מכן Next.



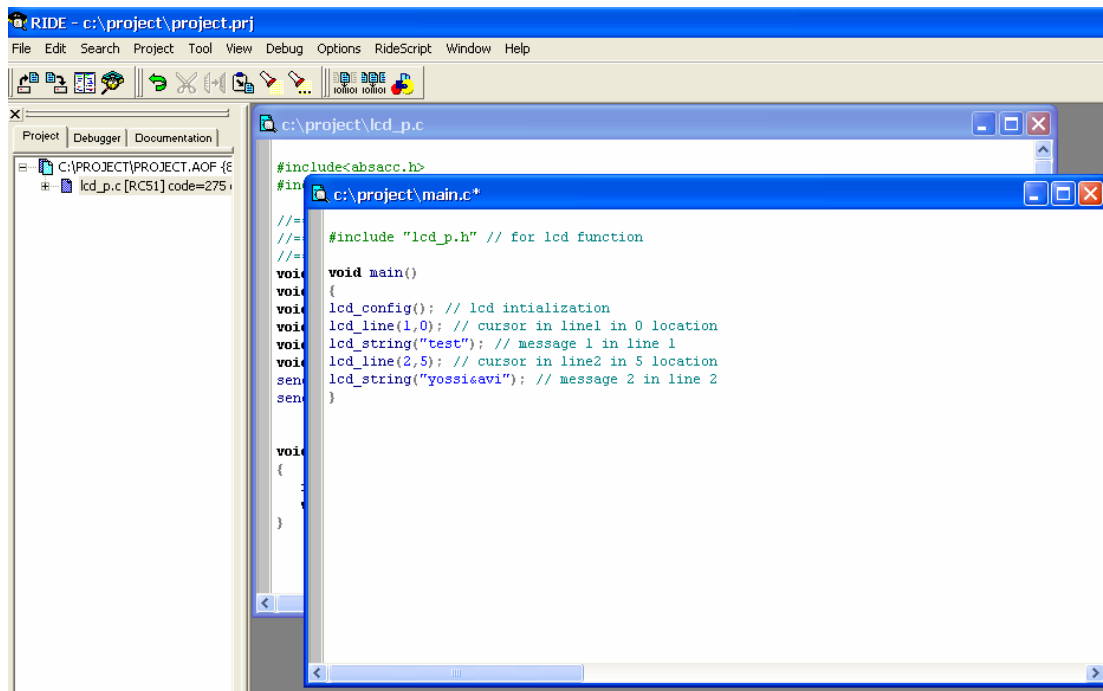
6. בחלון שנפתח בוחרים בתיקייה Atmel ואז את הרכיב AT89C5131 ולוחצים finish



7. הוסף לפרויקט את קבצי C של רכיבי הפרויקט כמו: lcd_p.c וכיון ← בתפריט Project בוחרים באפשרות: "Add node Source/Application"



8. פתח קובץ C חדש `File>New` ולאחר מכן "C Files" וכתוב בה את התוכנית הראשית (main). בהצהרות שלפני ה-main רשום למשל: עבור רכיב ה-LCD `#include "lcd.h"` הצהרה זו באה להכליל את הפונקציה שאתה נשתמש בתוכנית הראשית.
9. שמירת התוכנית בשם באמצעות: `File>Save`. יש לשמור את הקובץ עם סיומת C לדוגמא: `main.c`



10. לאחר ששמרנו יש לבדוק את טעויות תחביר של התוכנית -קומפילציה, בתפריט `project → build all`. בסיום תהליך התרגום המהדר ידווח על שגיאות(אם יש כאלה) בחלון שבתחתית המסך. כל עוד קיימות שגיאות יש לתקן, לשמור ולחזור על פעולות ההידור, עד שמתקבלים V ירוקים בתחתית החלון.

10. שינוי חיבורי חומרה של הרכיבים : פורטים, כתובות וכו' יתבצעו בקבצי h של אותם רכיב, בדוגמא שלנו עלינו לשנות את הדקי RS ו-E בקובץ `lcd_p.h` בהתאם לחיבורים שלנו.

כעת יש לצרוב את הקובץ עם הסיומת `hex` שנוצר לנו בתיקיית הפרויקט לכרטיס ולבדוק את החומרה במקרה זה יש לבדוק אם התקבלו הודעות בתצוגת ה-LCD.

- דרך נוספת לכתיבת תוכנית לתצוגה היא ע"י הגדרת מערך חד ממדי ושימוש בפונקציה `: sprintf`

```

#include<stdio.h> // for sprintf function
#include "lcd_p.h" // for lcd function
char lcd_array[16]; // array for lcd

void main()
{
    int i;
    float a;

    i=100/4;
    a=25/3;
    lcd_config(); // lcd initialization
    lcd_line(1,0); // cursor in line1 in 0 location
    sprintf(lcd_array,"The res are:%d",i); // The res are:25
    lcd_string (lcd_array); //send the array to lcd
    lcd_line(2,0); // cursor in line2 in 0 location
    sprintf(lcd_array,"The res are:%f",a); // The res are:8.33
    lcd_string (lcd_array); //send the array to lcd
}

```