

mikroBasic for dsPIC



Creating First Project in mikroBasic for dsPIC

In this simple tutorial, we will create a new project, write some code and compile it in mikroBasic for dsPIC and test the results. Our project will make LED diodes blink, so it can be easily tested on dsPIC microcontrollers.

Software and Hardware
solutions for Embedded World



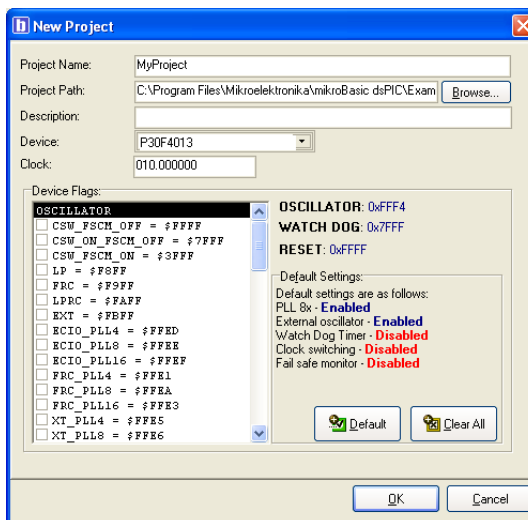
Creating first project in mikroBasic for dsPIC

Step 4: Set up your project

Project Wizard dialog will appear - fill the dialog with appropriate settings for your project:

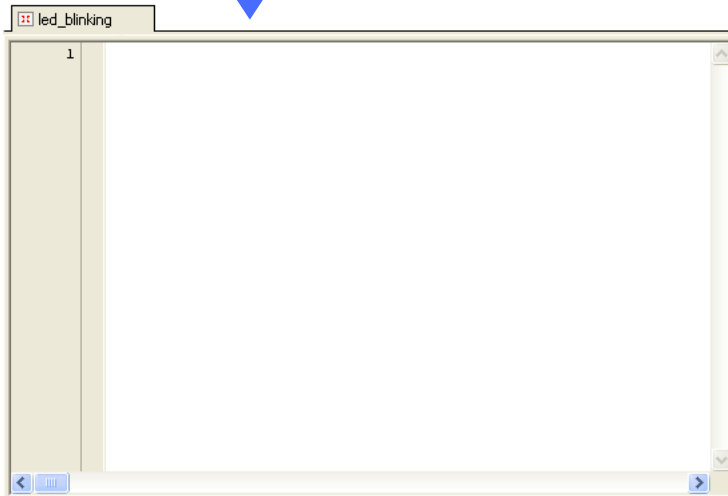
- Enter a name for your new project,
- Choose project path,
- Enter short text which describes your project (this is optional),
- Choose the microcontroller from drop-down menu (we have chosen the one that is used in schematic for selected microcontroller, you may select any other),
- Set the device clock by entering the value in edit box (we have set the same value as drawn in schematic),
- Set configuration bits (Device Flags) by clicking Default,
- After you have set up your project, click OK to continue.

New Project window for mikroBasic for dsPIC



Step 5: Write the code

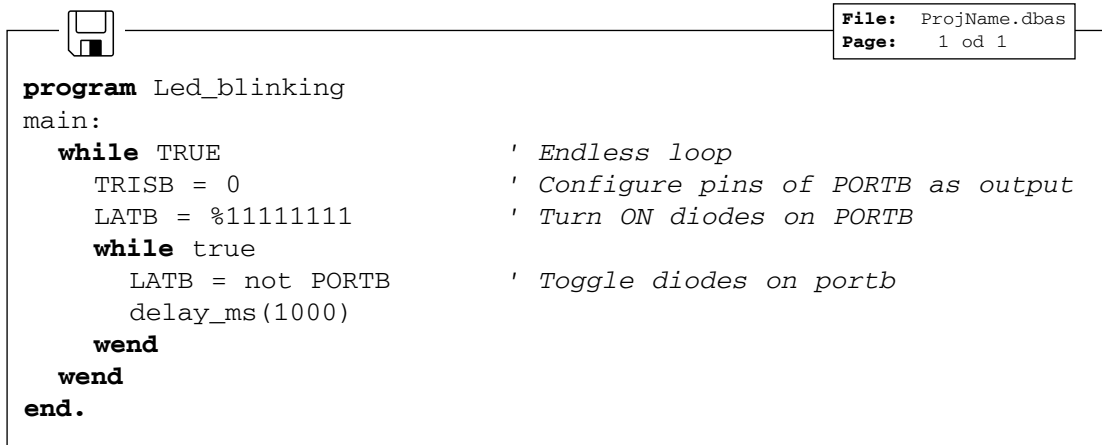
Compiler will create the project file and an accompanying source file, named same as your project. This source file will be automatically opened in the Code Editor, so we can write the source code. This is the Code Editor:



Creating first project in mikroBasic for dsPIC

Led blinking for mikroBasic for dsPIC:

Here is the code that will make LED's on PORTB blink every second. Type it in the Code Editor (if there is any default code you can over write it):



```
program Led_blinking
main:
  while TRUE           ' Endless loop
    TRISB = 0          ' Configure pins of PORTB as output
    LATB = %11111111   ' Turn ON diodes on PORTB
    while true
      LATB = not PORTB ' Toggle diodes on portb
      delay_ms(1000)
    wend
  wend
end.
```

File: ProjName.dbas
Page: 1 od 1

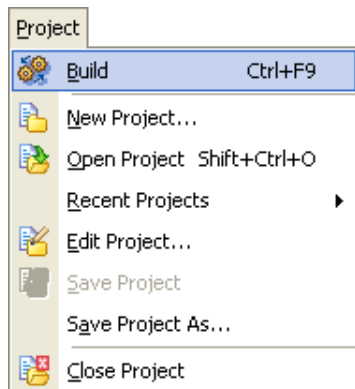
Note: If you want to try another port, just replace each instance of LATB and TRISB in code with your port, e.g. LATC/TRISC.

Step 6: Build!



Build.

Now it's time to build our project. First, save your file by clicking on the Save Icon, or click Ctrl+S. Select **Project > Build** from the drop-down menu, or click the Build Icon. You can also use the shortcut Ctrl+F9.



Compiler will generate output files in the project folder. There should be 4 new files: ProjectName.asm, ProjectName.lst, ProjectName.mcl, and ProjectName.hex. The last one will be used to program the microcontroller directly.

Step 7: Real World test

To "burn" the dsPIC you can use the compiler's integrated (dsPICprog) programmer (default shortcut is F11) or any other programmer for selected microcontroller. As a result, LED's connected to PORTB of programmed microcontroller should blink indefinitely.