Batronix Prog-Studio

Prog-Studio is a modern, comprehensive, and very productive micro-controller development environment for industry standard 8051 (MCS-51) microcontroller architecture chips.

1 user license, only for non commercial projects . **Attention:** Delivery contents just a license key, no data medium or carton!

- Modern and advanced development environment
- All 8051's are supported, no specific assumptions
- Contains fantastic functions to develop µC based applications
- Source Code Debugger, Edit & Continue, Monitor, Watchlists, ...
- Powerful and reliable project manager
- Excellent ease of use, develop quickly without hassles
- technical support by our development engineers
- Fair price, best value for money

Description

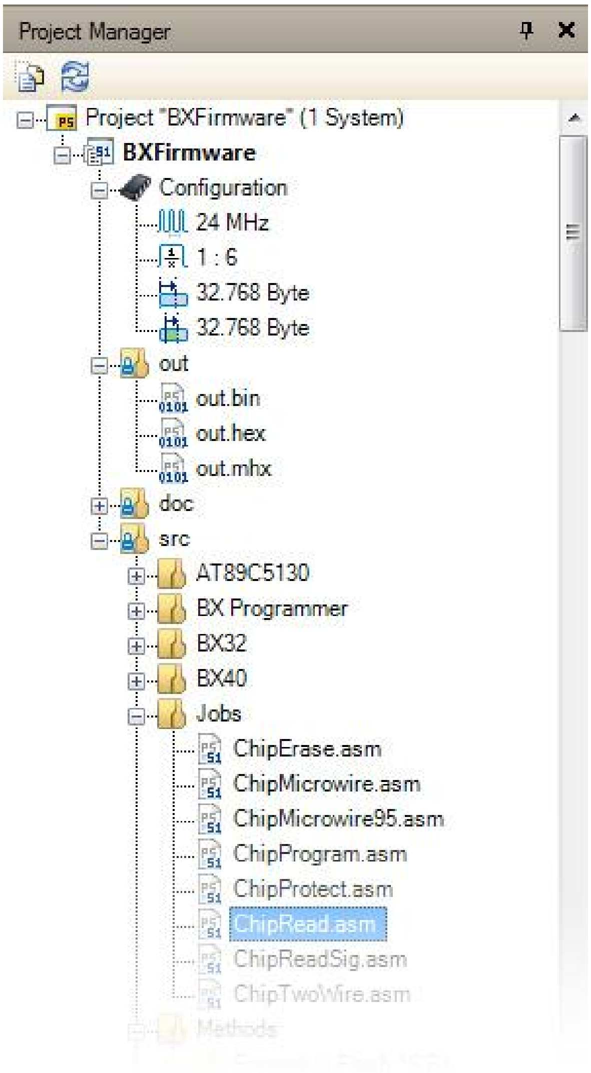
Prog-Studio is a modern, comprehensive, and very productive micro-controller development environment for industry standard 8051 (MCS-51) microcontroller architecture chips.

Working programs could be created with just a few steps. Subsequently, the program could be debugged. A hardware microcontroller is not nessasary, because Prog-Studio includes a source level debugger and a MCS-51 Emulator. Programs can be transmitted to the microcontroller by just one click (ISP In-System-Programming).

Prog-Studio provides a high level of comfort for the MCS-51 developer thanks to functions like Project Management, Dockable Windows, Code-Folding, and Edit & Continue. This type of functionality is typically only available for expensive IDE's offering high-level PC languages. Beginners prefer Prog-Studio because they like the intuitive usability of this IDE and also the good examples which help to understand the assembler language.

Features

Project Manager



Goal

The goal of the Project Manager is to simplify the management of your project and reduce as much complexity as possible. We would like to enable you to concentrate on the essentials: Creating source code and documentation.

Easy

A new project is created with just a few mouse clicks. Prog-Studio supports the structured development of programs by the consistent implementation of project orientated development.

Source code, documentation, published data and project settings are managed centrally by a project manager: Adding new files and folders, copying, moving, or erasing are accomplished by the project manager in the usual way (shortcuts, context menus and drag & drop) - an external file manager is not required.

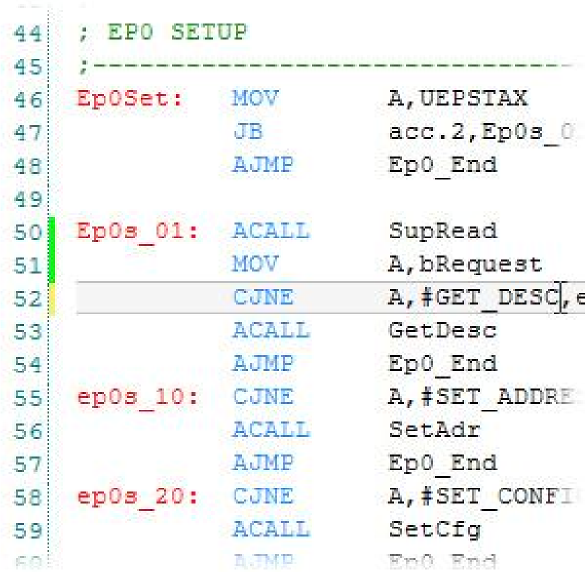
Multi Controller Projects

In some cases, several micro-controllers are required during the implementation of a project.

This situation is supported by the project manager, which automatically groups all files and settings for each micro-controller into systems.

Any number of these systems can be contained in a project and allow for even highly complex projects with several micro-controllers to be developed in Prog-Studio.

Source-Editor



```
; EP0 SETUP
;------------------------------
Ep0Set:   MOV      A,UEPSTAX
          JB       acc.2,Ep0s_0
          AJMP     Ep0_End

Ep0s_01:  ACALL    SupRead
          MOV      A,bRequest
          CJNE     A,#GET_DESC,e
          ACALL    GetDesc
          AJMP     Ep0_End
ep0s_10:  CJNE     A,#SET_ADDRE
          ACALL    SetAdr
          AJMP     Ep0_End
ep0s_20:  CJNE     A,#SET_CONFI
          ACALL    SetCfg
          AJMP     Ep0_End
```

Comfortable, Fast and Reliable

The source code editor was developed and adjusted in order to allow the comfortable, simple and sufficiently clear coding of programs attained by current technologies such as code-folding, line-marking and code-highlighting.

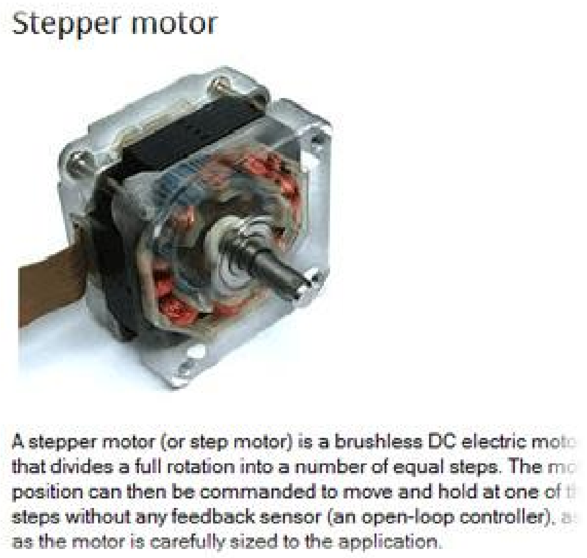
Code-folding allows for the maximizing and minimizing of code segments. That way you can increase the legibility of your program by hiding non-relevant sections.

Clearly arranged and Customizable

View the lastest modifications on your fingertips, as all recent changes are marked with a green indicator at the beginning of each line. Unsaved changes and modifications are marked with yellow indication.

Many features are customizable in the software options.

Documentation-Editor



Stepper motor

A stepper motor (or step motor) is a brushless DC electric mot
that divides a full rotation into a number of equal steps. The mo
position can then be commanded to move and hold at one of t
steps without any feedback sensor (an open-loop controller), a
as the motor is carefully sized to the application.

Efficient

In most cases, documentation is created before and during the development process. In order to avoid time-consuming re-work, a documentation editor has been integrated into the Prog-Studio software allowing a promt documentation of information and important data content.

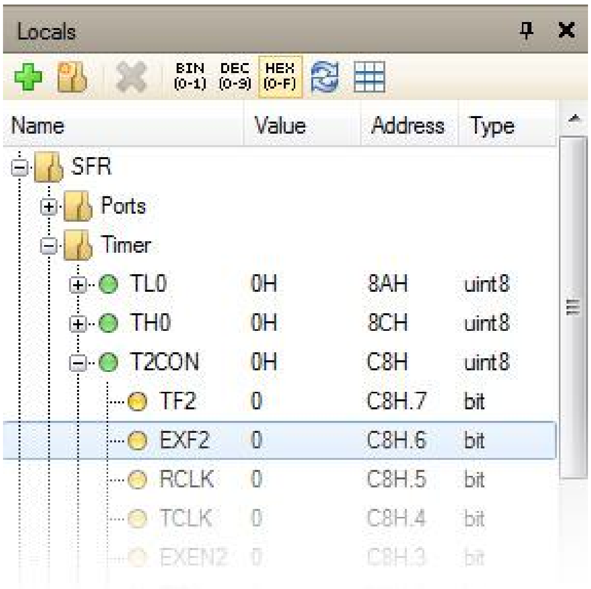
Number lists can be created and text can be formatted into different fonts supporting normal, bold, cursive, cross out, underlined formatting, as well as coloring and colored backgrounds.

The need for notes or text editing software running in parallel belongs to the past just as much as staying up late to write the documentation.

Interoperable

This editor uses the rich-text file format. This is a standardized format, which can therefore be read and stored by many programs offering a wide range of formatting options.

Watchlists



Name	Value	Address	Type
SFR			
Ports			
Timer			
TL0	0H	8AH	uint8
TH0	0H	8CH	uint8
T2CON	0H	C8H	uint8
TF2	0	C8H.7	bit
EXF2	0	C8H.6	bit
RCLK	0	C8H.5	bit
TCLK	0	C8H.4	bit
EXEN2	0	C8H.3	bit

Run time variables up-to-date

The current values of runtime variables are always shown. You can view all local variables, and also specify targeted watch lists. If this is not enough, you can view all of the emulated RAM storage directly.

Modified values are highlighted to make sure you don't miss any important changes. Running values can be changed at will in the lists. All visualized data can be viewed in both, hexadecimal and decimal formats.

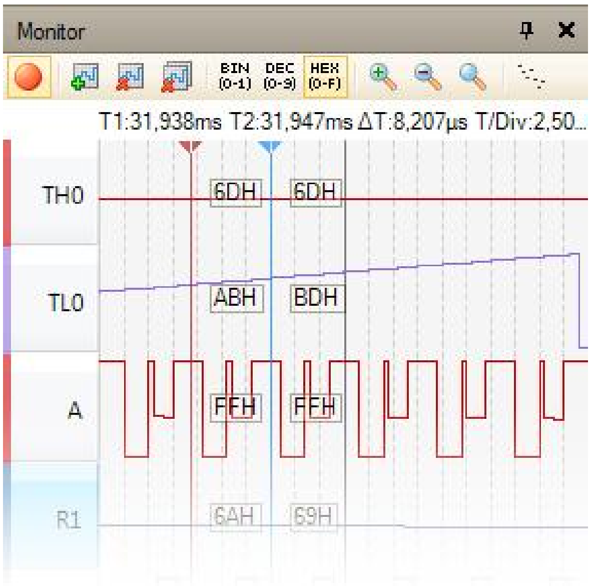
Edit & Continue

Nothing gets in the way of a fluid workflow more than having to interrupt a debug session to correct mistakes. This interruption is not necessary when using Prog-Studio. You can make changes directly to the source code during a debug session. The IDE saves the current state of the emulator in the background and recompiles the program. After assembly, program execution continues from the current program position. The time required to fix detected errors is reduced drastically.

Makro-Assembler

Repetitive source code functions can be defined as macros. This increases the legibility of the source code and makes the program a lot easier to maintain. During a debug session, the directions of the macro calls are marked during execution. This way the programming logic remains comprehensible and traceable.

Monitor

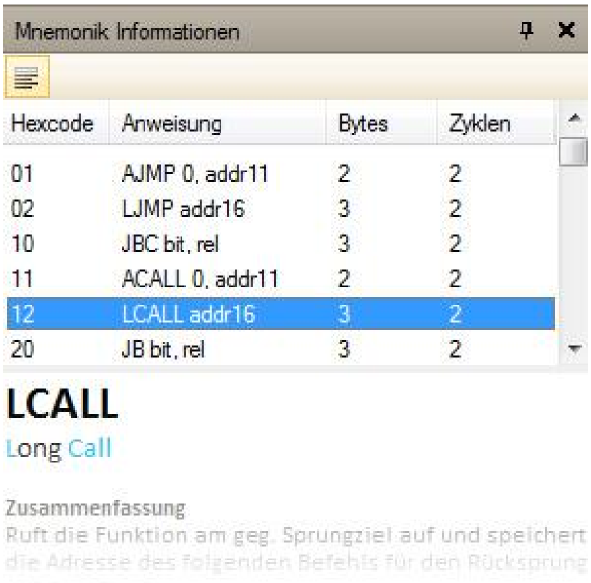


T1:31,938ms T2:31,947ms ΔT:8,207µs T/Div:2,50...

Time-Dependant Analysis

For a time-dependant analysis, you can display desired ports or variables in a monitor similar to an oscilloscope. With two tracing bars you can take time and value measurements of all channels. Unlike a real oscilloscope, there is no limit on the number of channels.

Additional Information



Hexcode	Anweisung	Bytes	Zyklen
01	AJMP 0, addr11	2	2
02	LJMP addr16	3	2
10	JBC bit, rel	3	2
11	ACALL 0, addr11	2	2
12	LCALL addr16	3	2
20	JB bit, rel	3	2

LCALL

Long Call

Zusammenfassung

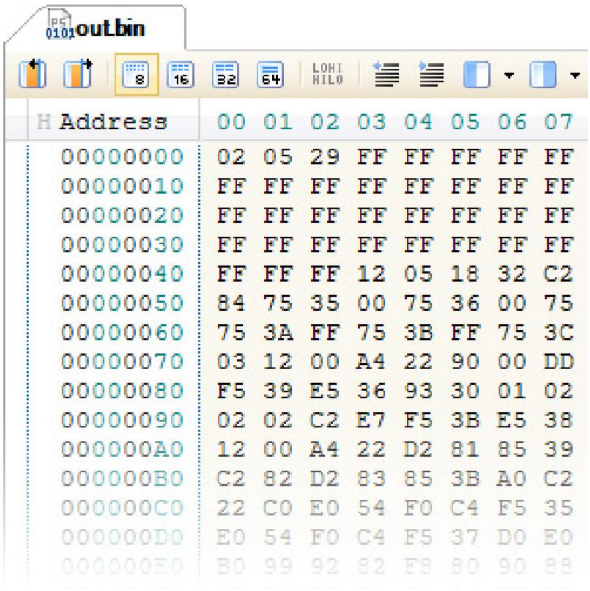
Ruft die Funktion am geg. Sprungziel auf und speichert die Adresse des folgenden Befehls für den Rücksprung

Quick Assembler Reference

Who hasn't had this happen to them? You look for a specific assembly instruction and it's allowed parameters. Looking it up in a book becomes unavoidable at the latest when you need to know how many cycles this instruction needs. This interruption to your work is not necessary with Prog-Studio. All assembly instructions for the MCS51 micro-controllers are clearly laid out in the assembler mnemonics list. If desired, the assembly instructions can also be displayed grouped by category.

Questions, such as which jump instruction takes the least cycles or what variations of the CJNE instruction are available, are answered quickly.

Hex-Editor



```
H Address   00 01 02 03 04 05 06 07
  00000000  02 05 29 FF FF FF FF FF
  00000010  FF FF FF FF FF FF FF FF
  00000020  FF FF FF FF FF FF FF FF
  00000030  FF FF FF FF FF FF FF FF
  00000040  FF FF FF 12 05 18 32 C2
  00000050  84 75 35 00 75 36 00 75
  00000060  75 3A FF 75 3B FF 75 3C
  00000070  03 12 00 A4 22 90 00 DD
  00000080  F5 39 E5 36 93 30 01 02
  00000090  02 02 C2 E7 F5 3B E5 38
  000000A0  12 00 A4 22 D2 81 85 39
  000000B0  C2 82 D2 83 85 3B A0 C2
  000000C0  22 C0 E0 54 F0 C4 F5 35
  000000D0  E0 54 F0 C4 F5 37 D0 E0
  000000E0  B0 99 92 82 F8 80 90 88
```

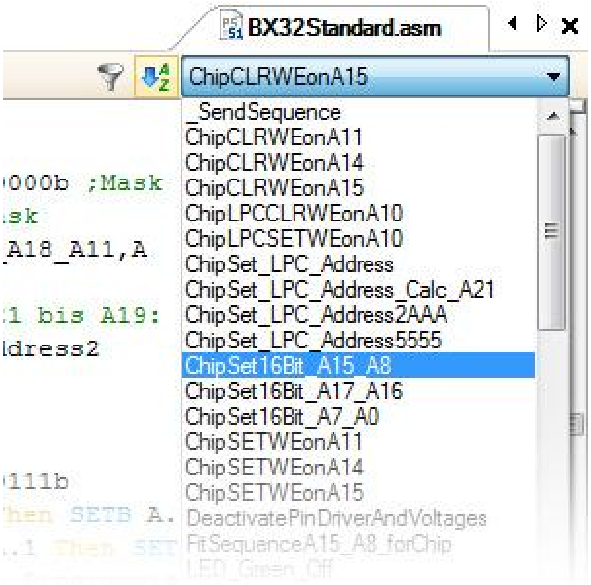
Inspect binary data

The integrated hex editor not only allows for comfortable binary data manipulation, but also supports pasting entire data sets. The hex-editor reads and writes Intel-Hex and Motorola-S-Record data files in addition to binary data files.

You can adjust the display of data to meet your requirements. For example, you can pre-set the amount of bytes to be displayed per row or whether you want to see the data as individual bytes or in word, double-word, or quad-word format.

Checksum calculation as well as various automatic data manipulations are supported. Incrementing or decrementing entire ranges, filling a range with random or sequential values, or switching nibbles or bytes does not pose a problem.

Label-List



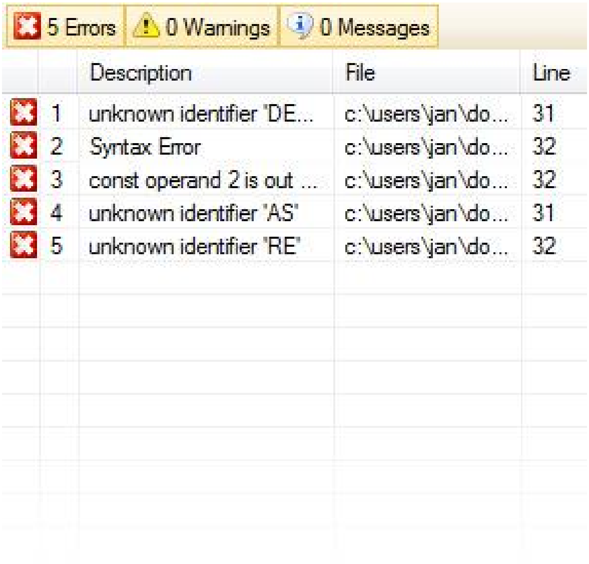
Jump Label

Finding a specific label is made possible not just by the comprehensive search functions of the IDE. All labels for the file currently displayed are quickly selectable through a selection list. For qucik access this list is located immediately on top of the data entry screen.

Filter, Sort

In addition, this list can be sorted and filtered to remove less important labels.

Errorlist



☒ 5 Errors | ⚠ 0 Warnings | ℹ 0 Messages

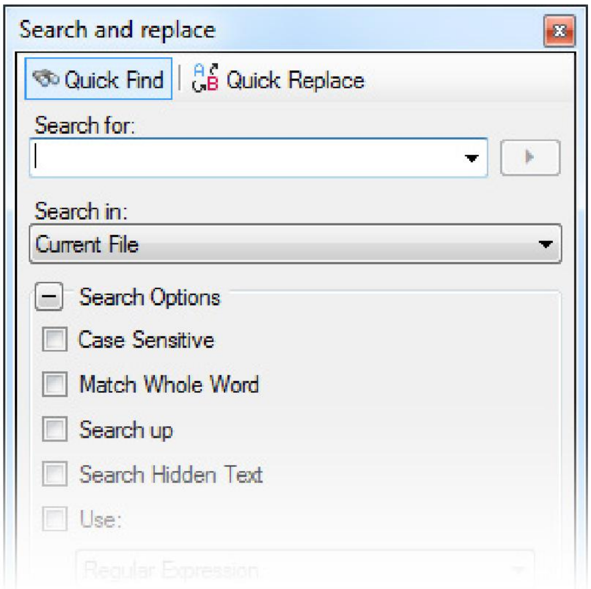
		Description	File	Line
☒	1	unknown identifier 'DE...	c:\users\jan\do...	31
☒	2	Syntax Error	c:\users\jan\do...	32
☒	3	const operand 2 is out ...	c:\users\jan\do...	32
☒	4	unknown identifier 'AS'	c:\users\jan\do...	31
☒	5	unknown identifier 'RE'	c:\users\jan\do...	32

View errors

Everybody makes programming errors! In order to get rid of them quickly, Prog-Studio offers detailed error descriptions including file path, line number and position within the line.

Clicking on an error in the error list is all it takes to open the corresponding file and set the cursor to the error location.

Find & Replace



Search for:

Search in: Current File

⊟ Search Options

□ Case Sensitive

□ Match Whole Word

□ Search up

□ Search Hidden Text

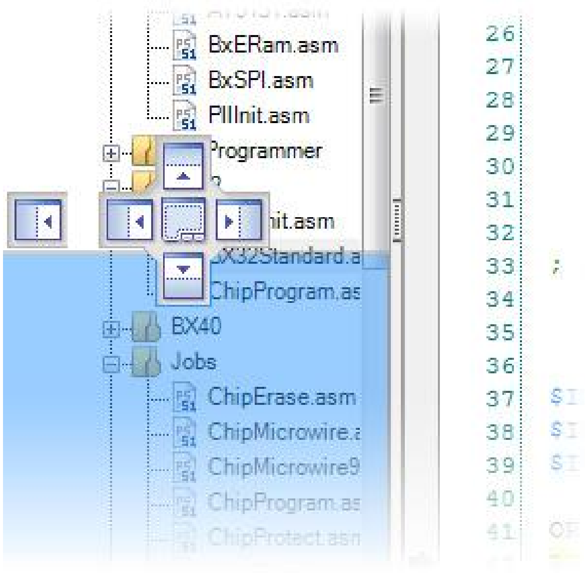
□ Use: Regular Expression

Quick and Easy Finding

Prog-Studio offers a very easy to use search and replacement interface. Of course, searching is not limited to one file but can be expanded to include the entire project. Matching case, selecting the search direction and limiting the search to whole words can be done quickly and comfortably using easy to reach options.

In the options section, you can also change the type of search. The classic full-text search, searching with wildcards (known from the DOS era) and searching with regular expressions are all available. When searching with regular expressions, replacing the source can also be done with regular expressions.

Dockable User Interface



Flexible

Whether you want to move a window, have it floating, or automatically hide and integrate specific windows is up to you. You can configure your “Workplace” Prog-Studio completely according to your personal preferences.