

Developing Applications for MIOS

Developing custom applications for MIOS-driven Pic-Chips is very easy and straightforward:



This page covers the **APPLICATION** part as seen above.

When coding your app in C, the compiler will automatically include when necessary, the MIOS-enhanced SDCC-Libraries (<http://svnmios.midibox.org/trunk/lib/>), but this will increase the code size(!).

If you stick to the provided functions coming with MIOS, you just have to learn the C-Syntax which is very easy. If you ever programmed in any other (also scripting-) language, you might be able to learn the syntax in just a few hours and start programming like a pro!

As you can also program in Assembler and don't know which language to choose: Technically spoken, you are free to choose whatever language you like, C, ASM or [mix it](#) only when necessary. But be aware: the learning curve for ASM is very steep, and if you don't need to squeeze out the very last mililisecond or spare another byte, it is recommended to code in C.

Hint for beginners: You will soon find out, that it's no problem to add and subtract; it's slightly more difficult to multiply, but actually very complex to divide. One possibility to avoid multiplications and divisions (beside the use of ASM code snippets inside the C-Code) is the intelligent use of bitshifting techniques. So buy yourself a good C book or find a webpage and read carefully about bitshifting operations. Also see [C Tips and Tricks for PIC Programming](#).

[MIDI Specification](#) also contains some useful informations.

Setup & Install

Windows

Quick Start Guide Most users just want to download the source, configure the code for their hardware, and compile it. If that's you, and you're familiar with Windows, the [Quick Start Guide](#) will get you up and running, quick and easy. If you have any problems, or want to know all the gory step-by-step details, you may prefer the Detailed Setup Guides. If you follow these guides exactly, you can't go wrong :)

- Detailed Setup Guides
 - **Part 1** - [Setup the core software toolchain for MIOS application development on Windows](#) - These are the minimal Windows applications you need, to write a MIOS Application in C and/or ASM for your MIDibox. Not sure? Do this part.
 - **Part 2** - [Extend your toolchain with MIOS Studio](#) - MIOS Studio is an extremely useful tool for debugging on the MBHP platform, and is required for uploading apps. Not sure? Do this part.
 - **Part 3** - IDE installation. You only need to do one of these, but there's nothing stopping you from doing them all if you feel like wasting disk space ;)
 - **Part 3a** - [Extend your toolchain with NotePad ++](#) - NotePad++ is an advanced text editor with useful features like code highlighting, code completion, and very basic debugging. It has the advantage of simplicity, and that it is suitable for both C and ASM apps, and those that mix both languages. Not sure? Do this part.
 - **Part 3b** - [Extend your toolchain with Code::Blocks and GDB](#) - Code::Blocks is an IDE which has several helpful features when developing applications, such as code highlighting, code completion, and debugging. in conjunction with MinGW (installed in Part 1) and GDB, you can use the AC-Sim DOS console C application simulator for debugging and development. Code::Blocks provides a very comfortable environment for C development, but is less useful for ASM or mixed language apps. Not sure? Skip this part.
 - **Part 4** - [Extend your toolchain with TortoiseSVN](#) - TortoiseSVN is used to interface with the MIOS SVN repository. Advanced coders and those creating their own apps will need this. Not sure? Skip this part.

Macintosh

- [Installing GPUTILS and SDCC on OSX](#) - step-by-step tutorial & download links
- [How to use Xcode2 as IDE on a Mac](#) - step-by-step tutorial
- [\[Leopard OS X 10.5\]How to setup XCode3 to access the SVN server](#) - step-by-step tutorial

Linux

- [Installing GPUTILS and SDCC on Gentoo Linux](#) - step-by-step tutorial
- [Developing MIOS Apps with Eclipse](#) <incomplete - v3.3 missing> - step-by-step tutorial &

download links

- [Installing GPUTILS and SDCC on Fedora Core 5](#) - step-by-step tutorial

Platform independent

- [ACSim](#) <working, always in development/improvement phase> - Debug and simulate your C-Application in a console application before uploading it to the Core!
- [Using the AutoTools application skeleton](#) - step-by-step tutorial & download links
- [Developing MIOS Apps with Eclipse](#) <incomplete - v3.3 missing> - step-by-step tutorial & download links

References, Tipps & Snippets

<box 46% left bluelight|General (C & ASM)> **MBHP**

- [MBHP acronyms](#) ← Abbreviations overview for a better understanding of the Function Reference List
- [MIOS I/O Pin usage in wiki format](#) or [Official MIOS I/O pin usage text file](#) ^{uCApps}
- [MIOS RAM Handling](#) ^{uCApps}

PICs

- [Using PIC18F4620](#) - How to use the newer PIC18F4620 with a lot more cool features together with MIOS v1.9
- [Using PIC18F4685](#) - How to use the newer PIC18F4685

LCDs

- [MIOS LCD Offsets](#) - How MIOS LCD Offsets work with various display types
- [Character LCDS](#) - find there the link to the famous special character creator of Cpt. Hastings
- [Graphical LCDS](#) - find there the link to the famous GLCD character creator of Cpt. Hastings

Encoders

- [Activate encoders in C](#) - How to activate your encoders in an C-Application

</box>

<box 46% right blue|Coding in C>

- [Function Reference List](#) ^{uCApps}
- [C Interface & Examples](#) ^{uCApps}

- **C Tips and Tricks for PIC Programming**
- **How to mix C and ASM** - optimizing code: using plain ASM in a C-Based project

</box>

<box 46% right bluedark|Coding in ASM>

- **ASM Function Reference List** ^{uCApps}

MPASM/MPLab are no longer used for MIOS. The new tool is **GPASM** </box>

From:

<https://midibox.org/dokuwiki/> - **MIDibox**

Permanent link:

https://midibox.org/dokuwiki/doku.php?id=application_development&rev=1216991446

Last update: **2008/07/25 13:36**

