

# How to setup GP2X development environment on Windows and use Code::Blocks as IDE

Written by Delrina Chang ([gmail.com@delrinachang](mailto:gmail.com@delrinachang)) version 1.0

January 9<sup>th</sup> 2007

## 1. Introduction

The GP2X is an open-source, Linux-based handheld video game console and media player created and sold by GamePark Holdings of South Korea. Released on [November 10, 2005](#) in South Korea, the GP2X is designed to play video and music, view photos, and play games. It has an open architecture, allowing anybody to develop and run software. Also, there is the possibility for additional features (such as support for new media formats) to be added in the future due to the upgradeable [firmware](#).

A popular use of the GP2X is for running [emulators](#), which allow one to use software from another system on the GP2X.

GP2X and its development kits (breakout board) can be purchased at <http://www.gp2x.com>

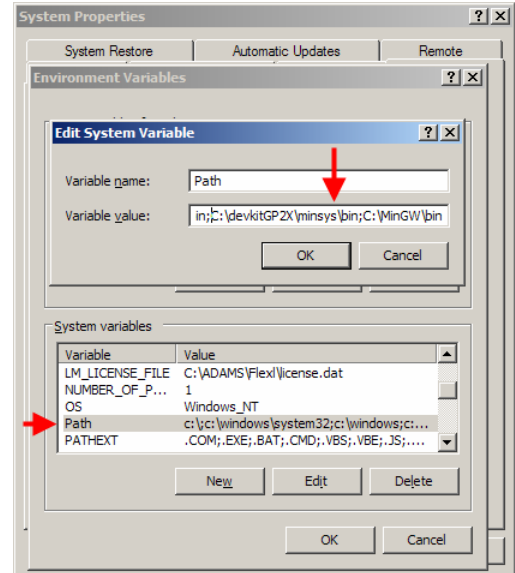
To develop graphics applications (games, multimedia apps) on GP2X, we need to use an IDE such as Visual Studio, Dev C+ or Code::Blocks to write code in host PC. In this case, we use Code:Blocks IDE integrated with MingW32 because it is free. The OS in host PC is Windows XP because it is most popular OS today. Using Code::Blocks, MingW32, we will compile our application into two target operating systems:

- Windows platform for debugging and simulation, \*.exe file.
- Linux platform on GP2x, \*.gpe file.



## 2. Set up GP2X development kit

- 1- Download: gp2x devpack compiled by *Octoate* at here:  
<http://archive.gp2x.de/cgi-bin/cfiles.cgi?0,0,0,0,14,1362> . Unzip the devkitGP2X directory into your root directory. If you chose a development directory other than **C:/devkitGP2X**, you will need to correct the paths in bin/arm-linux-sdl-config and demo/Makefile.
- 2- Upgrade the SDL libraries in C:/devkitGP2X by download this zip file *sdl-libs-211006.zip* compiled by *Guyfawkes* at <http://archive.gp2x.de/cgi-bin/cfiles.cgi?0,0,0,0,19,1875>. Extract and copy all directories bin, etc, include, info, lib, man and share to C:/devkitGP2X. Overwrite if necessary.
- 3- Add the bin directories to your path. Click on Start->Settings->Control Panel, open System->Advanced->Environment Variables, select PATH and add **C:\devkitGP2X\bin;C:\devkitGP2X\minsys\bin;** to your path.
- 4- Download a zip file **SDL Test Program Thingy** version 1.51 made by *Guyfawkes*. Unzip it to a directory for example, example SDLTest. Run command line and type “make” to compile this demo program. If compilation is successful, copy sdltest.gpe and sdltest folder to GP2X to test drive. If compilation fails, then check step 1, 2, 3 or go to <http://www.gp32x.com/board/index.php> to ask for help.

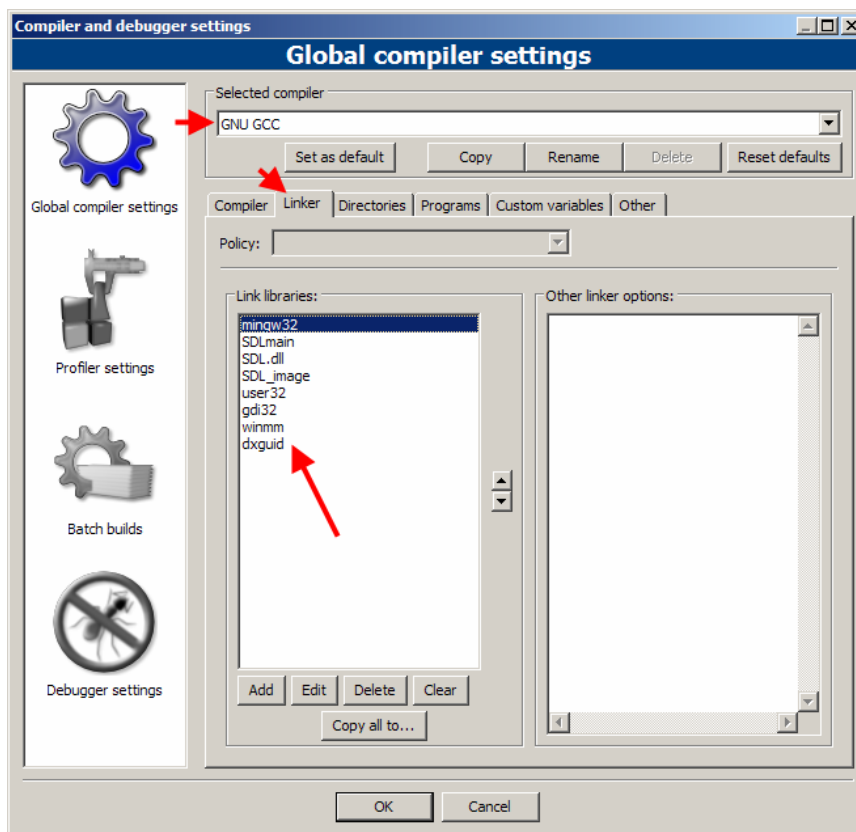


## 3. Set up Code::Blocks IDE

In the previous section, I mentioned how to setup GP2x development kit, which is used to create \*.gpe file for Linux OS in GP2X. In this section, we go further to install Code::Blocks, MingW. The MingW is integrated with Code::Blocks to create \*.exe file.

- 1- Down load nightly built Code::Blocks at <http://forums.codeblocks.org/index.php?board=20.0> , unzip to a folder C:\CodeBlock.
- 2- Down load MinGW Installer 5.0.2 at <http://prdownloads.sourceforge.net/mingw/MinGW-5.0.2.exe>. This MingGW is compiler to produce executable file for Windows OS. Run installer, we just need gcc and g++ compiler therefore unselect g77, ada and java components. Suppose MingW is installed at C:\MinGW
- 3- Go here <http://www.mingw.org/download.shtml> to download latest mingw32-gcc.exe, mingw32-g++.exe, ar.exe, gdb.exe, windres.exe, mingw32-make.exe files to update to C:\MinGW\bin folder.
- 4- Click on Start->Settings->Control Panel, open System->Advanced->Environment Variables, select PATH and add **C:\MinGW\bin**. Otherwise, you cannot start up Code::Blocks because it requires a DLL file *mingwm10.dll*

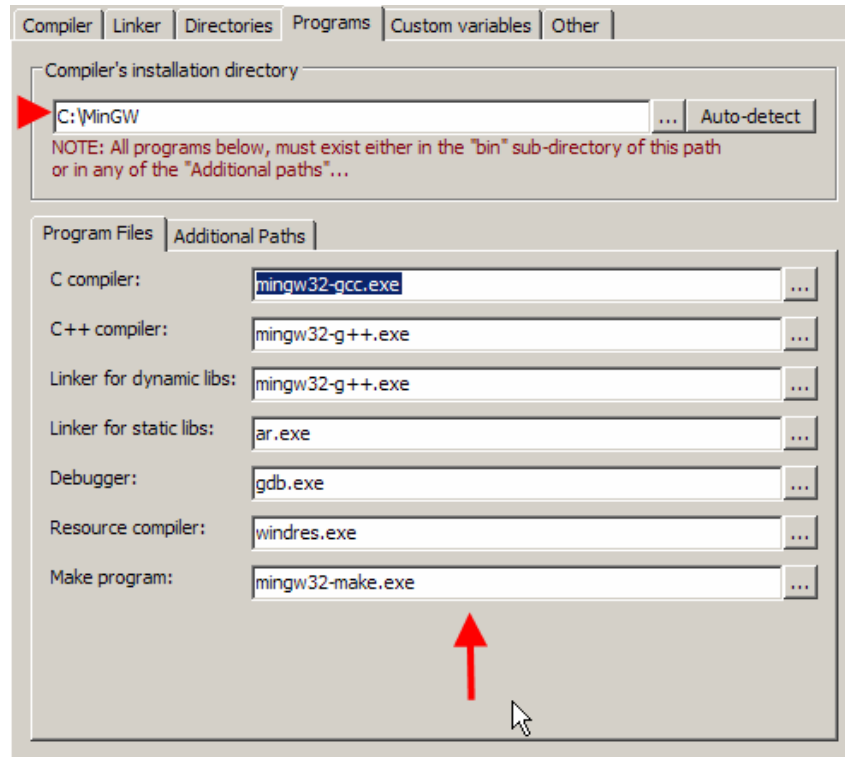
- 5- Next, we need install SDL library for MinGW, go here <http://www.libsdl.org/download-1.2.php> and download <http://www.libsdl.org/release/SDL-devel-1.2.11-mingw32.tar.gz> unzip to folder C:\SDLMingW
- 6- In this step, we configure Code::Blocks global compiler settings. Notes, in Code::Blocks, the project compiler settings augment to the global compiler settings. First we configure GCC compiler settings for compiling Win32 \*.exe file.
- Start Code::Blocks and go to menu Settings > Compiler and Debugger.
  - In the combo box labeled Selected Compiler, select GNU GCC, you may copy this compiler profile as new one such as “GCC Win32 SDL”. As for me, I make changes directly to “GNU GCC”.
  - Go to tab Compiler, then tab #defines, add WIN32. It is equivalent to #define WIN32 in our source code. It will tell the compiler to compile only Win32 code not GP2X code when creating \*.exe file.
  - Go to tab Linker, add following libraries: mingw32, SDLmain, SDL.dll, SDL\_image, user32, gdi32, winmm, dxguid. (Notes this is very basic libraries for a simple SDL application. For specific project, you may include additional library to project's linker settings).



- Go to tab Directories, to add location storing header files and libraries files. First, select tab Compiler, add:
  - C:\SDLMingW\include, location of header files of SDL libraries.
  - C:\MinGW\include, location of standard header files of MinGW such as string.h, stdio.h...

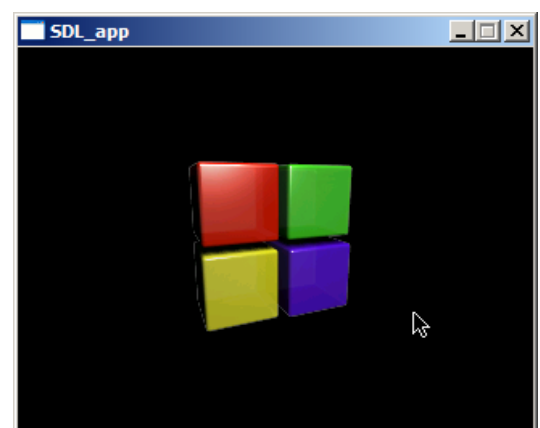
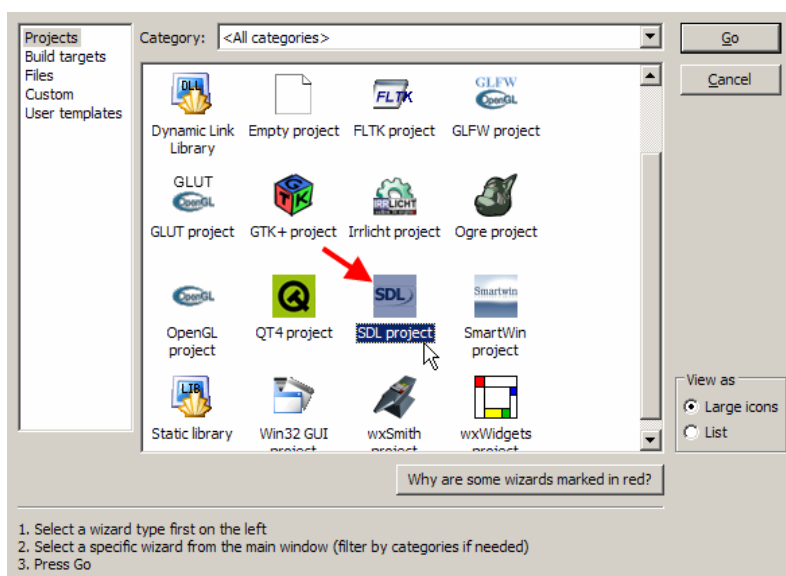
Second, select tab Linker, add C:\SDLMingW\lib and C:\MinGW\lib

f. Go to tab Program, and configure as bellow:



g. To test GNU GCC compiler profiles, lets create a SDL project, in menu, select File > New > Project then choose SDL Project icon. Follow wizard:

- i. Enter C:\SDLMingW as SDL's location
- ii. Select GNU GCC as compiler and click button Finish.
- iii. Compile and you should see a simple demo SDL application "SDL\_app".
- iv. Save this project for later modification.

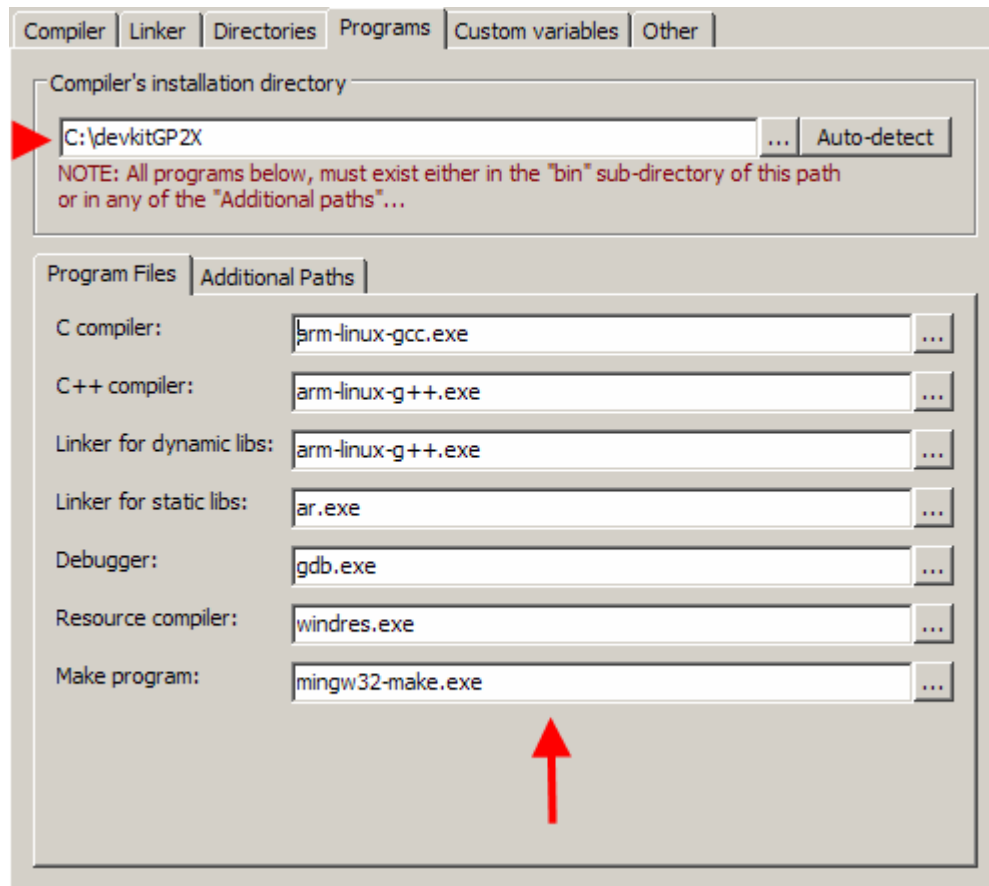


We can optimize a little for our Win32 SDL application by settings compiler flags in tab Compiler > Compiler Flags...

Since we use gdb.exe as debugger, we can debug our Win32 SDL as easily as in Microsoft Visual Studio. More important, you don't have to pay a penny !

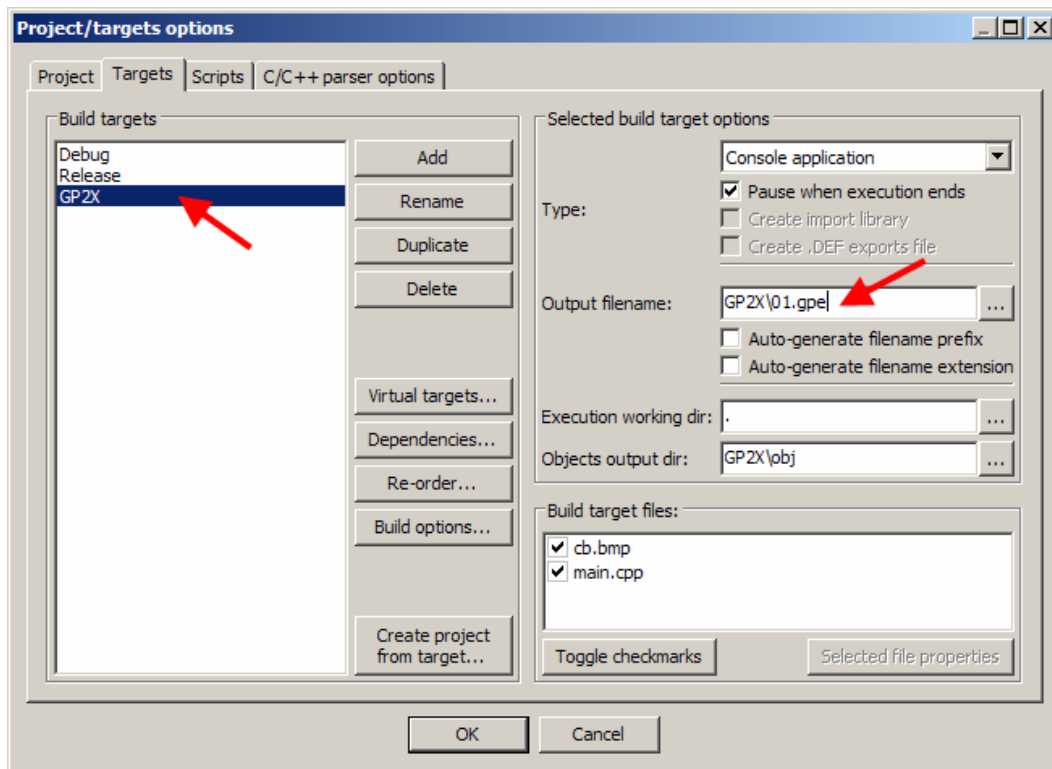
7- Next, we configure compiler for GP2X. Make a copy of GNU GCC compiler profile settings and rename it to “DevKitGP2X”

- a. Go to tab Compiler > #define GP2X
- b. Go to tab Directories > tab Compiler, add path to location of header files  
C:\devkitGP2X\include, C:\devkitGP2X\lib\gcc\arm-linux\4.0.2\include,  
C:\devkitGP2X\include\SDL
- c. Go to tab Directories > tab Linker add path to location of library files  
C:\devkitGP2X\lib
- d. Go to tab Programs and configure as follow:



8- After creating compiler settings for GP2XdevKit, we can modify our Code::Blocks project so it can compile to \*.gpe file.

- a. Go to menu Project > Properties. Select tab Targets and click button “Add” to create new target.
- b. Modify extension of output file name from \*.exe to \*.gpe.
- c. Turn on the check boxes, Auto-generate file name prefix and Auto-generate file name extension. Then click button Ok to save settings.



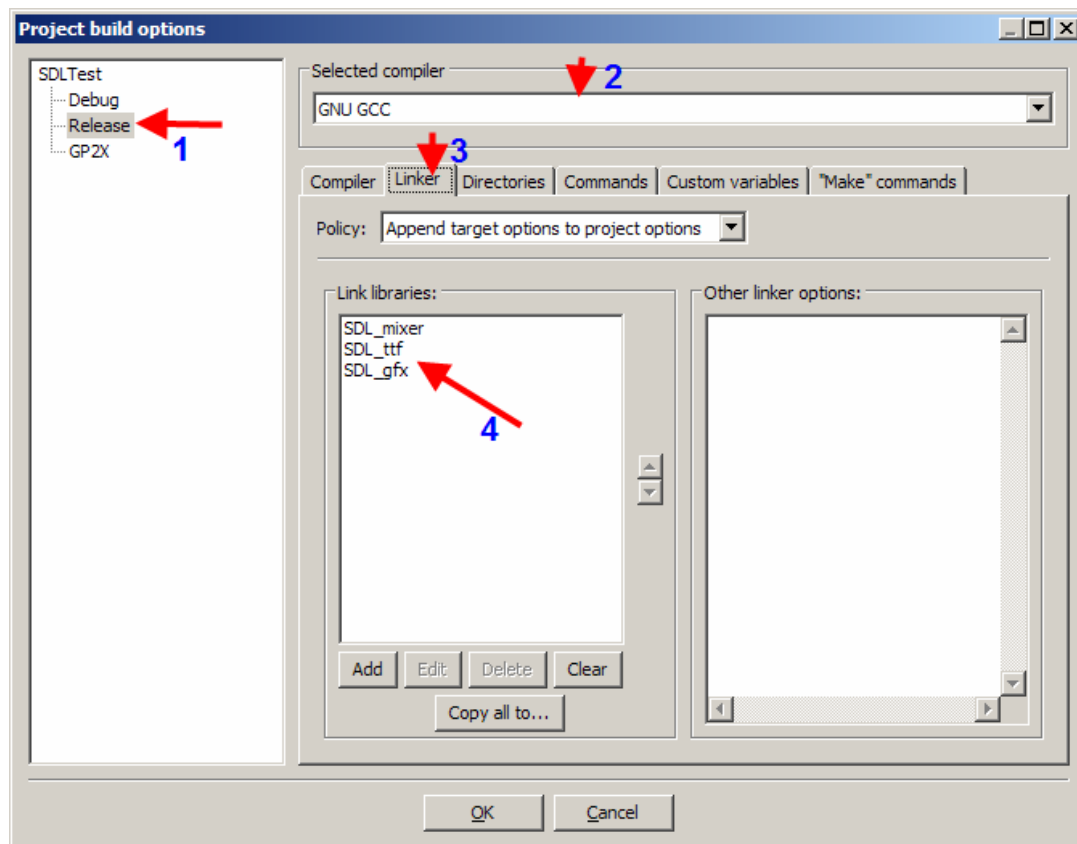
- d. Go to menu Project > Build Options. Set GNU GCC as compiler for Debug and Release targets. Set DevKitGP2X as compiler for GP2X.
- e. In dialog “Project build options”, switch to tab Linker, paste following line to Other linker options: `-static -lSDL_gfx -lSDL_image -ljpeg -lz -lSDL_mixer -lgcc -lm -lc -lexpat -lpthread -lmad -dgp2x` . We will use static link to create \*.gpe file. The result \*.gpe file will be big. But we cannot create \*.gpe file that dynamically links to SDL libraries since DevKitGP2X use SDL and gcc which is newer than those supplied by Games Park Holdings.
- f. Menu Build > Select Target > GP2X and then Build. Transfer output \*.gpe with cb.bmp file to GP2x and test. (*Be warned, when you run this program, you cannot exit because, there is no appropriate handling code to terminate this sample program in GP2X environment*).

## 4. Compile additional libraries for SDL

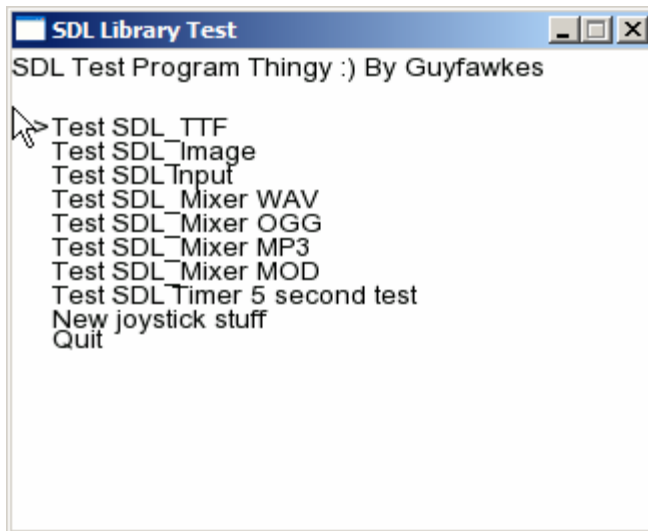
While devkitGP2X supplies many additional libraries such as SDL\_image, SDL\_ttf, SDL\_mixer, so you can easily build \*.gpe file for GP2X. We better to make similar environment in host PC, Windows XP so we can create and simulate our game conveniently in Windows. As we satisfy with Win32 game version, we switch target to GP2X and make \*.gpe file. In this section, I explain briefly how to install (or compile if necessary) SDL\_image, SDL\_mixer, SDL\_gfx and SDL\_ttf. You can visit here for more detail [http://lazyfoo.net/SDL\\_tutorials/lesson03/windows/codeblocks/index.php](http://lazyfoo.net/SDL_tutorials/lesson03/windows/codeblocks/index.php)

- 1- SDL\_Image: Go to [http://www.libsdl.org/projects/SDL\\_image/](http://www.libsdl.org/projects/SDL_image/) download SDL\_image-devel-1.2.5-VC6.zip file and copy \*.h, \*.lib and \*.dll file to appropriate folder.
- 2- SDL\_mixer: Go to [http://www.libsdl.org/projects/SDL\\_mixer/](http://www.libsdl.org/projects/SDL_mixer/) and download SDL\_mixer-devel-1.2.7-VC6.zip

- 3- SDL\_gfx: Go to <http://www.ferzkopp.net/joomla/content/view/19/14/> and download this file SDL\_gfx-2.0.15.tar.gz.
  - a. Go to Other builds unzip “Makefile” to source folder.
  - b. Replace “c:/dev/local” to the path to SDL library folder “C:/SDLMingW”
  - c. Replace “c:/dev/local/include/SDL” by “C:/SDLMingW/include”
  - d. Use command line, go to source folder and type “mingw32-make -B” to compile. The output files are SDL\_gfx.dll, libSDL\_gfx.a, libSDL\_gfx.dll.a
- 4- SDL\_ttf: Go to [http://www.libsdl.org/projects/SDL\\_ttf/](http://www.libsdl.org/projects/SDL_ttf/) and download this file SDL\_ttf-devel-2.0.8-VC6.zip. SDL\_ttf requires FreeType 2 library. Go here to download FreeType 2. <http://gnuwin32.sourceforge.net/packages/freetype.htm>
- 5- To compile SDL Test Program Thingy version 1.51,
  - a. Go to menu Project > Build options
  - b. Select Release build target (GNU GCC compiler)
  - c. Go to tab Linker, in the list box Link Libraries, add SDL\_mixer, SDL\_ttf, SDL\_gfx
  - d. Menu Build > Build and Run







SDL Test Program Thingy is running on Windows XP OS.

*Notes: Test SDL\_Input, you need to use keyboard to simulate GP2X joystick and buttons.*

## 5. Some good SDL programming tutorials

Firstly, I recommend the demo of **SDL Test Program Thingy** version 1.51 made by *Guyfawkes*. This demo works well in GP2X and Windows.

Some good SDL tutorials:

[http://lazyfoo.net/SDL\\_tutorials/index.php](http://lazyfoo.net/SDL_tutorials/index.php) Lazy Foo's Productions

<http://sol.gfxile.net/gp/> Sol Tutorials

More tutor please see here <http://www.libsdl.org/tutorials.php>

Book about games programming in SDL: **Data Structures for Game Programmers** by Ron Penton, 2002.

Special thanks to all members in <http://archive.gp2x.de>

Chang,