

GCC-CyberGL

Sebastian Huebner <cyco@gmx.de>

COLLABORATORS			
	TITLE : GCC-CyberGL		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Sebastian Huebner <cyco@gmx.de>	May 28, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1	GCC-CyberGL	1
1.1	cybergl39_12gcc - GCC-CyberGL.guide	1
1.2	Idea & Copyright	1
1.3	Installation	2
1.4	How to use	3
1.5	Troubleshooting	4
1.6	Changes & Bugfixes	5
1.7	Sources of Information	5
1.8	Greetinx & Contact	6
1.9	Index	6

Chapter 1

GCC-CyberGL

1.1 cybergl39_12gcc - GCC-CyberGL.guide

cybergl39_12gcc

GCC-CyberGL.guide V0.4 13-Apr-1998

This package is based on cybergl39_12dev.lha, freely available at <http://www.vgr.com/cybergl/>. It is slightly modified, bugfixed and extended to let you develop programs with GCC, the GNU C compiler, that is a part of the Geek Gadgets project.

Idea & Copyright
Installation
How to use
Troubleshooting
Changes & Bugfixes
Sources of Information
Greetinx & Contact

INDEX

1.2 Idea & Copyright

IDEA:

Phase 5 Digital Products have developed an implementation of OpenGL for the AMIGA, that is called CyberGL. To write your own programs you can get the package cybergl39_12dev.lha from <http://www.vgr.com/cybergl/>. Unfortunately this package only supports the SAS C compiler and it is not possible to use the freely available GCC compiler, for example included in the Geek Gadgets environment.

cybergl_lib.fd says:

"We provide a pragma file for SAS/C since SAS/C supports fpu registers in pragma files. GNU C or other compilers probably need special stub libraries, we can't provide yet."

But with the help of this SAS pragma file and the brilliant FD2Pragma by Dirk Stroeker (Hi!) – after a few bugfixes – I was able to create the needed inlines for gcc. Additionally a stub library is included (libCyberGL.a), which gives you the possibility to use CyberGL in an GL_APICOMPATIBLE way.

COPYRIGHT:

The only included (and modified) file from the original package is cybergl_protos.h. It says:

```
/*
** $VER: cybergl_protos.h 39.12 (12.03.1998)
**
** C prototypes. For use with 32 bit integers only.
**
** Copyright © 1996–1998 by phase5 digital products
** All Rights reserved.
**
**
*/
```

You can find it in cybergl39_12gcc/orig/. Just do a "diff" to see, which lines differ from the original file, or see changes & bugfixes.

So my files are definetly Freeware (feel free to email me), cybergl_protos.h is © Phase 5 Digital Products.

1.3 Installation

ATTENTION:

YOU NEED A 68040 OR 68060 CPU TO USE CYBERGL WITH GCC !!

INSTALLATION:

First, you need cybergl39_12usr.lha, that contains the standard shared library. You can get it from Aminet (/biz/p5/cybergl39_12usr.lha) or from ftp.phase5.de (/pub/phase5/cgx3/cybergl39_12usr.lha). It contains 3 versions of cybergl.library (030/040/060). Copy the right one to LIBS:cybergl.library.

You should get cybergl39_12dev.lha from <http://www.vgr.com/cybergl/>, too, because it includes some useful examples to start with cybergl and we need the two files C/Include/cybergl/{cybergl.h|display.h}.

Now unarchive cybergl39_12gcc.lha (Aminet: /dev/gcc/cybergl39_12gcc.lha) to e.g. RAM:.

Normally, gcc searches for the system include files in GG:os-include. If it isn't there, you should create it. All other amiga system files should live here, too. Or you use an independent directory, then copy the cybergl includes there. I will suppose here that you are using GG:os-include.

If you are using an old version of cybergl-dev (last was 39.8), delete

GG:os-include/libraries/cybergl.h and GG:os-include/libraries/cybergl_display.h. Since 39.12, they live in an own directory "cybergl" and are called cybergl.h and display.h! The SAS/C pragma file in pragmas/cybergl_pragmas.h is useless for us, but it will be replaced soon.

Install the includes:

Just copy all files in cybergl39_12gcc/include to GG:os-include, or to a place you would like them to live.

```
Copy RAM:cybergl39_12gcc/include/ GG:os-include/ ALL
```

Copy the two files from cybergl39_12dev/C/Include/cybergl to the same directory.

```
MakeDir GG:os-include/cybergl
```

```
Copy RAM:cybergl39_12dev/C/Include/cybergl/ GG:os-include/ ALL
```

Install the link library:

Normally gcc searches for link libraries in GG:lib, so you should copy libCyberGL.a there.

```
Copy RAM:cybergl39_12gcc/lib/libCyberGL.a GG:lib/
```

Done!

1.4 How to use

HOW TO USE:

After installing the includes and the link library, we're ready to go. There are only a few point to think of, when porting the SAS/C examples to gcc.

- Since gcc has no library-auto-open feature you always have to open and check for cybergl.library yourself. You need a global variable to store the pointer to the library base that IS NOT EXTERN! SAS/C uses an extern base pointer for its l-a-o feature.

```
/* global */
struct Library *CyberGLBase;

/* in main() */
if(CyberGLBase=OpenLibrary("cybergl.library",0))
{
    ... /* your program */

    CloseLibrary(CyberGLBase);
}
else
    puts("Unable to open cybergl.library!");
```

- You always have to define SHARED, because the supplied link library only includes stubs for some special functions.

```
#define SHARED
```

- You always have to include `inline/cybergl.h`, because the supplied... (see above ;-). This include file and the whole `cybergl.library` use fpu registers to specify parameters, so you have to compile with the `-m68040` flag (see troubleshooting). Yes, this implies that your program is only usable with a 68040 or 68060 cpu, but - `cybergl` needs speed, so it is unusable on anything below 040, anyhow.

```
#include <inline/cybergl>
```

```
gcc -m68040 example.c -o example
```

- If you would like to use the `GL_APICOMPATIBLE` way of calling `glLookAt()`, `glOrtho()`, etc. you need to link with the supplied link library `libCyberGL.a`. Attention: you need to specify the link library behind the source/object files that use the stub functions (see troubleshooting)!

```
#define GL_APICOMPATIBLE
```

```
gcc -m68040 example.c -lCyberGL -o example
```

1.5 Troubleshooting

TROUBLESHOOTING:

- `"XXX.o(.text+0xYYYY): undefined reference to 'CyberGLBase'"`
 ==> You must not define `CyberGLBase` as `extern`.
 ==> You have to open `cybergl.library` with `OpenLibrary()`.
- `"XXX.o(.text+0xYYYY): undefined reference to 'glZZZ_stub'"`
 ==> When using `CyberGL GL_APICOMPATIBLE`, you have to link with `"-lCyberGL"`.
 ==> `"-lCyberGL"` has to be specified behind the object that uses functions in `libCyberGL.a`.
- `"XXX.c:YY: register number for '_ZZ' isn't suitable for the data type"`
 ==> Compile with `"-m68040"` (FPU!) Note: Yes, you need a 040!
- `"XXX.c:YY: warning: initialization makes integer from pointer without a cast"`
 ==> You used a `"Tags"`-function with a string: Use `(ULONG)"String"`, because all tags are mapped to `ULONGS` internally by `inline/cybergl.h`.

```
- "XXX.c:YY: libraries/cybergl.h: No such file or directory"
  "XXX.c:YY: libraries/cybergl_display.h: No such file or directory"

==> cybergl.h & display.h now live in INCLUDE:cybergl/ .
```

See How to use for details!

1.6 Changes & Bugfixes

BUGFIXES:

pragmas/cybergl_pragmas.h

- (1) glPixelTransferf is #pragma flibcall (not #pragma libcall)
- (2) glProject and glUnProject aren't GL_APICOMPATIBLE (says cyberg_protos.h), so they have only one parameter in a0
- (3) glTexImage1D has only 8 parameters (not 9)

clib/cybergl_protos.h

- (4) There's no type "struct TagList *", it is called "struct TagItem *"
- (5) glFrustum, glProject and glUnProject aren't GL_APICOMPATIBLE, so they need glFrustum_stub, glProject_stub and glUnProject_stub

CHANGES:

pragmas/cybergl_pragmas.h

- glPixelTransferf is flibcall (1)
- glProject/glUnProject only 1 parameter (a0); check for GL_APICOMPATIBLE (2)
- glTexImage1D has 8 parameter (3)

clib/cybergl_protos.h

- openGLWindowTagList, attachGLWindowTagList, attachGLWndToRPtagList: "struct TagList *" replaced by "struct TagItem *" (4)
- #define glProject glProject_stub
- #define glUnProject glUnProject_stub
- #define glFrustum glFrustum_stub (5)

1.7 Sources of Information

SOURCES OF INFORMATION:

CyberGL by Phase 5
<http://www.vgr.com/cybergl/>
http://www.vgr.com/cybergl/cybergl39_12dev.lha (examples)

Architectural Review Board (ARB): OpenGL
<http://www.opengl.org/>
<http://www.opengl.org/Info/Info.html> (FAQs, ...)
<http://www.opengl.org/Developers/developers.html> (specs, ...)

OpenGL Specification V1.2 Mar-1998
<ftp://sgigate.sgi.com/pub/opengl/doc/opengl1.2/opengl1.2.ps> (PostScript)

CyberGL Workshop by German "AMIGA Magazin" (language: german)
<http://www.magnamedia.de/amiga/magazin/a04-97/cyber.html>
<http://www.magnamedia.de/amiga/magazin/a05-97/cyber.html>
<http://www.magnamedia.de/amiga/magazin/a06-97/cyber.html>

1.8 Greetinx & Contact

Greetinx to: Dirk Stoecker (author of FD2Pragma)
Kamil Iskra (author of FD2Inline)
Andreas "Aenda makes the difference" Retzlaff
Alexander "ALeX" Kazik
Andreas "Andreas" Mayer-Guerr

Contact me: Sebastian Huebner <cyco@gmx.de>

1.9 Index

No Index yet. Sorry.