gclib 450

C API for Galil controllers and PLCs

Galil Motion Control

Wed Sep 18 2019

Contents

Chapter 1

Getting Started

The Galil Communication Library (gclib) is a communication library for Galil motion controllers and PLCs. A number of programming languages, operating systems, and hardware platforms are supported.

The library consists of a basic set of function calls (gclib.h), and an open-source extension library (gclibo.h). A number of examples are provided to demonstrate how to use the library with various languages.

The gclib will import virtually anywhere a dll/so/dylib can be imported. See installation for details. Please contact `softwaresupport@galil.com` if the language or platform required is not listed.

Contents

- **`List of all functions`**
- Installation and supported operating systems
- Language Support
- Using gclib

Release Notes

See the update history of gclib in the `release notes`.

Galil maintains an `RSS` page to notify users of updates.

See the update history of gcaps in the `release notes`.

Technical Support

For help please email `softwaresupport@galil.com`, or call `Galil Applications`.

Chapter 2

Installation

gclib is available on the following operating systems.

- Microsoft Windows
  - 10 x64 ♢, x86
  - 8.1 x64 ♢, x86
  - 8 x64 ♢, x86
  - 7 x64 ♢, x86
- Ubuntu Linux
  - 18.04 LTS x64 ♢
  - 16.04 LTS x64 ♢ †
  - 14.04 LTS x64 †
  - 12.04 LTS x64 †
- Fedora Linux
  - fc30 x64 ♢
  - fc29 x64 ♢ †
  - fc28 x64 ♢ †
  - fc27 x64 ♢ †
  - fc26 x64 ♢ †
  - fc25 x64 ♢ †
  - fc24 x64 ♢ †
  - fc23 x64 ♢ †
  - fc22 x64 †
  - fc21 x64 †
- Red Hat and CentOS
  - Red Hat 7 & CentOS 7 Linux
    - RHEL 7 x64 ♢
    - CentOS 7 x64 ♢
  - Red Hat 6 & CentOS 6 Linux
    - RHEL 6 x64 †
    - CentOS 6 x64 †
- Raspbian, Raspberry Pi

  - Raspberry Pi 3 Model B+, Stretch 9.8
  - Raspberry Pi 3 Model B, Stretch 9.1 †
  - Raspberry Pi 2 Model B, Jessie 8.0 †
- Apple OS X
  - Yosemite 10.10 x64
  - Mavericks 10.9 x64
- Other builds, contact Galil Applications for more info
  - Nvidia Jetson TX1 running Ubuntu 16.04 arm64

♢ gcaps available on these operating systems.

† Although no longer built on these older operating systems, previous versions of gclib are available.

Don't see your OS? Please email softwaresupport@galil.com, or call Galil Applications.

2.1 Microsoft Windows

Tested versions

See the installation page for supported versions.

Installation

On Windows, gclib is distributed in the following formats.

- An executable installer which will install the library in the proper location to work with the included examples and documentation. PCI users can optionally install the PCI driver from within this installer.
- A zip file containing the same set of files as the executable but in a zip archive. PCI users can use the stand-alone PCI driver installer.
  - A stand-alone PCI driver installer for PCI users (DMC-1806, 1800, 1802, 1417).

Note

The PCI driver is compatible with GalilTools but is enhanced for gclib communications.

Download Installer

Recommended. All instructions and examples depend on the installation paths.

Download Zip

For custom deployment or non-default file locations. Downloads are available on the release notes page.

Required third-party DLLs

gclib is built using **MSVC 2015** and requires run-time components available in the Microsoft Visual C++ Redistributable Packages for Visual Studio 2015.

The gclib installer will automatically install these prerequisites for both 32 bit (x86) and 64 bit (x64) builds. The installer allows the user to opt out of this installation, if desired.

If using the zip installation, the binaries must be downloaded and installed manually.

Silent Installation

For developers wishing to bundle gclib within their own installers, execute the gclib installer with the /S switch to run silently with defaults. If the Galil security certificate is not already trusted on the deployment target, a digital signature dialog may be presented.

Uninstall gclib

- Run `uninstall.exe` in "C:\Program Files (x86)\Galil\gclib"

Installed Files

Installation from the executable installer looks like the following.

```
C:\Program Files (x86)\Galil\gclib>tree /a
Folder PATH listing for volume OS
Volume serial number is AE3F-6836
C:.
+---dll
|   +---x64
|   \---x86
+---doc
|   \---html
|       \---search
+---examples
|   +---cpp
|   +---cs
|   |   \---2013_12.0
|   |       \---gclib_example
|   |           \---gclib_example
|   |               \---Properties
|   +---gcc
|   +---mingw
|   +---msvc
|   |   \---2013_12.0
|   |       \---gclib_example
|   |           \---gclib_example
|   \---vb
|       \---2013_12.0
|           \---gclib_example
|               \---gclib_example
|                   \---My Project
+---include
+---lib
|   \---dynamic
|       +---x64
|       \---x86
\---source
    +---gclibo
    \---wrappers
        +---cs
        +---gcl
        \---vb
```

dll

The *dll* directory contains the binary *dynamic link libraries* (DLLs) for both x86 and x64 architectures. **Dynamically linked executables must have the correct dlls in their path at runtime**.

doc

The *doc* directory contains this documentation and a printable, pdf version.

examples

The *examples* directory contains example projects for various compilers. The *cpp* directory contains *x_examples.h* and the implementation of the example files documented in this manual.

Warning

> Before using the examples, copy the files to a user location such as *C:\Users\user\Documents*. Failing to do so may cause source files to be deleted upon gclib uninstallation.

include

The *include* directory contains header files needed for compiling code. The compiler will need to know where these files are at compile time.

See the compiler-specific directions for more information, e.g. `gclib using MinGW`.

lib

The *lib* directory contains linker files (*gclib.lib* and *gclibo.lib*) for both x86 and x64 architectures. The linker should include *gclib.lib* and *gclibo.lib*.

source

The *source* directory contains source files such as gclibo.c.

2.2 Ubuntu Linux

Tested versions

This version of Linux has **x64/AMD64 Support Only**. Contact Galil if another version is required for an application.

See the installation page for supported versions.

Installation

Create a temporary variable for OS version

```
$ os=$(lsb_release -r -s)
$ echo $os
18.04
```

The number printed must be in the installation list of supported operating systems.

Install Galil's public certificate

```
$ wget http://www.galil.com/sw/pub/ubuntu/$os/GALIL-PUB-KEY
# apt-key add GALIL-PUB-KEY
```

Get Galil's apt sources list

```
# wget http://www.galil.com/sw/pub/ubuntu/$os/galil.list -O /etc/apt/sources.list.d/galil.list
# apt-get update
```

Install Package

Install gclib

```
 # apt-get install gclib
```

Install gcaps on 16.04 and better (optional)

Following Linux's daemon naming conventions, gcaps is called *gcapsd* on Ubuntu. See the gcaps documentation for more information.

```
# apt-get install gcapsd
```

Verify that the systemd unit is running.

```
$ systemctl is-active gcapsd
active
```

Uninstall Package

If the packages need to be removed from the system, the following commands may be used.

Uninstall gclib

```
# apt-get remove gclib
```

Uninstall gcaps

```
# apt-get remove gcapsd
```

Serial Ports and USB

If access to the serial ports or USB (e.g. DMC-4103) is desired through gclib, the following will provide steps to join the correct access group. If using USB, be sure the controller is powered and the usb is plugged in before beginning.

Determine group with access

```
$ ls -l /dev/ttyUSB* /dev/ttyS*
crw-rw----. 1 root dialout   4, 64 Mar  3 16:39 /dev/ttyS0
crw-rw----. 1 root dialout   4, 65 Mar  3 16:39 /dev/ttyS1
crw-rw----. 1 root dialout   4, 66 Mar  3 16:39 /dev/ttyS2
crw-rw----. 1 root dialout   4, 67 Mar  3 16:39 /dev/ttyS3
crw-rw----. 1 root dialout 188,  0 Mar  6 11:08 /dev/ttyUSB0
```

In the above listing, **dialout** is the group that needs to be joined. **uucp** is another common group that may be listed.

Add the desired *username* to the group.

```
$ sudo gpasswd -a username dialout
[sudo] password for username:
Adding user username to group dialout
```

Log out and back in for change to take effect.

```
$ groups
username wheel dialout
```

gclib can now connect to serial and usb devices from user *username*.

PCI Controllers

If using a Galil PCI controller, the PCI driver must be installed.

Extract source and build driver

```
$ tar -xf /usr/share/doc/gclib/src/gclib_pci.tar.gz
$ make
```

Copy module and add to kernel

```
$ sudo cp galilpci.ko /lib/modules/$(uname -r)
$ sudo depmod
$ sudo modprobe galilpci
```

Add galil group for access to PCI

```
$ sudo groupadd -f -K GID_MIN=100 -K GID_MAX=499 galil
$ sudo cp 90-galilpci.rules /etc/udev/rules.d/
$ sudo udevadm control --reload-rules
$ sudo udevadm trigger
$ sudo usermod -a -G galil username  #exchange "username" with actual user’s name
```

Logout and back in. The PCI hardware is now available for access.

```
$ ls -l /dev/galil*
crw-rw---- 1 root galil 10, 56 Jun  9 11:07 /dev/galilpci0
$ echo -e "\x12\x16\r" > /dev/galilpci0
$ cat /dev/galilpci0
DMC1846 Rev 1.1a
:
```

Documentation

The documentation is left as a tarball to minimize disk usage. The latest release version of the user manual is available at the following link.

- http://www.galil.com/sw/pub/all/doc/gclib/html/

Offline html

The following allows viewing of the html docs from the installation.

```
$ tar -xzf /usr/share/doc/gclib/gclib_doc.tar.gz html
$ firefox html/index.html
```

2.3 Fedora Linux

Tested versions

This version of Linux has **x64/AMD64 Support Only**. Contact Galil if another version is required for an application.

See the installation page for supported versions.

Installation

On Fedora, gclib and gcaps are distributed in RPM repositories. The following steps can be performed to install.

Download Galil's repository information

Point a browser at http://www.galil.com/sw/pub/fedora/galilrpm-3-1.noarch.rpm and install the rpm. This installs Galil's RPM repositories and can be done from the terminal with the following.

```
$ wget http://www.galil.com/sw/pub/fedora/galilrpm-3-1.noarch.rpm
# rpm -Uvh galilrpm-3-1.noarch.rpm
```

Install Packages

Install gclib

```
# yum install gclib
```

Approve "Installed size" and "Importing GPG key", if prompted.

Install gcaps (optional)

Following Linux's daemon naming conventions, gcaps is called *gcapsd* on Fedora. See the gcaps documentation for more information.

```
# yum install gcapsd
```

Verify that the systemd unit is running.

```
$ systemctl is-active gcapsd
active
```

Uninstall Packages

If the packages need to be removed from the system, the following commands may be used.

Uninstall gclib

```
# yum remove gclib
```

Uninstall gcaps

```
# yum remove gcaps
```

Serial Ports and USB

If access to the serial ports or USB (e.g. DMC-4103) is desired through gclib, the following will provide steps to join the correct access group. If using USB, be sure the controller is powered and the usb is plugged in before beginning.

Determine group with access

```
$ ls -l /dev/ttyUSB* /dev/ttyS*
crw-rw----. 1 root dialout   4, 64 Mar  3 16:39 /dev/ttyS0
crw-rw----. 1 root dialout   4, 65 Mar  3 16:39 /dev/ttyS1
crw-rw----. 1 root dialout   4, 66 Mar  3 16:39 /dev/ttyS2
crw-rw----. 1 root dialout   4, 67 Mar  3 16:39 /dev/ttyS3
crw-rw----. 1 root dialout 188,  0 Mar  6 11:08 /dev/ttyUSB0
```

In the above listing, **dialout** is the group that needs to be joined. **uucp** is another common group that may be listed.

Add the desired *username* to the group.

```
$ sudo gpasswd -a username dialout
[sudo] password for username:
Adding user username to group dialout
```

Log out and back in for change to take effect.

```
$ groups
username wheel dialout
```

gclib can now connect to serial and usb devices from user *username*.

PCI Controllers

If using a Galil PCI controller, the PCI driver must be installed.

Install prerequisites

```
$ sudo yum install kernel-devel-$(uname -r)
$ sudo yum install kernel-headers-$(uname -r)
$ sudo yum install gcc
```

Extract source and build driver

```
$ tar -xf /usr/share/doc/gclib/src/gclib_pci.tar.gz
$ make
```

Copy module and add to kernel

```
$ sudo cp galilpci.ko /lib/modules/$(uname -r)
$ sudo depmod
$ sudo modprobe galilpci
```

Add galil group for access to PCI

```
$ sudo groupadd -f -K GID_MIN=100 -K GID_MAX=499 galil
$ sudo cp 90-galilpci.rules /etc/udev/rules.d/
$ sudo udevadm control --reload-rules
$ sudo udevadm trigger
$ sudo usermod -a -G galil username  #exchange "username" with actual user's name
```

Logout and back in. The PCI hardware is now available for access.

```
$ ls -l /dev/galil*
crw-rw---- 1 root galil 10, 56 Jun  9 11:07 /dev/galilpci0
$ echo -e "\x12\x16\r" > /dev/galilpci0
$ cat /dev/galilpci0
DMC1846 Rev 1.1a
:
```

Documentation

The documentation is left as a tarball to minimize disk usage. The latest release version of the user manual is available at the following link.

- http://www.galil.com/sw/pub/all/doc/gclib/html/

Offline html

The following allows viewing of the html docs from the installation.

```
$ tar -xzf /usr/share/doc/gclib/gclib_doc.tar.gz html
$ firefox html/index.html
```

2.4 Red Hat 7 & CentOS 7 Linux

Tested versions

This version of Linux has **x64/AMD64 Support Only**. Contact Galil if another version is required for an application.

See the installation page for supported versions.

Installation

On Red Hat, gclib and gcaps are distributed in RPM repositories. The following steps can be performed to install.

Download Galil's repository information

Point a browser at `http://www.galil.com/sw/pub/rhel/7/galilrpm-4-1.noarch.rpm` and install the rpm. This installs Galil's RPM repositories and can be done from the terminal with the following.

```
$ wget http://www.galil.com/sw/pub/rhel/7/galilrpm-4-1.noarch.rpm
# rpm -Uvh galilrpm-4-1.noarch.rpm
```

Install Packages

Install gclib

```
# yum install gclib
```

Approve "Installed size" and "Importing GPG key", if prompted.

Install gcaps (optional)

Following Linux's daemon naming conventions, gcaps is called *gcapsd* on Red Hat. See the gcaps documentation for more information.

```
# yum install gcapsd
```

Verify that the systemd unit is running.

```
$ systemctl is-active gcapsd
active
```

Uninstall Packages

If the packages need to be removed from the system, the following commands may be used.

Uninstall gclib

```
# yum remove gclib
```

Uninstall gcaps

```
# yum remove gcaps
```

Serial Ports and USB

If access to the serial ports or USB (e.g. DMC-4103) is desired through gclib, the following will provide steps to join the correct access group. If using USB, be sure the controller is powered and the usb is plugged in before beginning.

Determine group with access

```
$ ls -l /dev/ttyUSB* /dev/ttyS*
crw-rw----. 1 root dialout   4, 64 Mar  3 16:39 /dev/ttyS0
crw-rw----. 1 root dialout   4, 65 Mar  3 16:39 /dev/ttyS1
crw-rw----. 1 root dialout   4, 66 Mar  3 16:39 /dev/ttyS2
crw-rw----. 1 root dialout   4, 67 Mar  3 16:39 /dev/ttyS3
crw-rw----. 1 root dialout 188,  0 Mar  6 11:08 /dev/ttyUSB0
```

In the above listing, **dialout** is the group that needs to be joined. **uucp** is another common group that may be listed.

Add the desired *username* to the group.

```
$ sudo gpasswd -a username dialout
[sudo] password for username:
Adding user username to group dialout
```

Log out and back in for change to take effect.

```
$ groups
username wheel dialout
```

gclib can now connect to serial and usb devices from user *username*.

PCI Controllers

If using a Galil PCI controller, the PCI driver must be installed.

Install prerequisites

```
# yum update kernel
```

Reboot

```
# yum install kernel-devel-$(uname -r)
# yum install kernel-headers-$(uname -r)
# yum install gcc
```

Extract source and build driver

```
$ tar -xf /usr/share/doc/gclib/src/gclib_pci.tar.gz
$ make
```

Copy module and add to kernel

```
# cp galilpci.ko /lib/modules/$(uname -r)
# depmod
# modprobe galilpci
```

Add galil group for access to PCI

```
# groupadd -f -K GID_MIN=100 -K GID_MAX=499 galil
# cp 90-galilpci.rules /etc/udev/rules.d/
# udevadm control --reload-rules
# udevadm trigger
# usermod -a -G galil username  #exchange "username" with actual user’s name
```

Logout and back in. The PCI hardware is now available for access.

```
$ ls -l /dev/galil*
crw-rw---- 1 root galil 10, 56 Jun  9 11:07 /dev/galilpci0
$ echo -e "\x12\x16\r" > /dev/galilpci0
$ cat /dev/galilpci0
DMC1846 Rev 1.1a
:
```

Documentation

The documentation is left as a tarball to minimize disk usage. The latest release version of the user manual is available at the following link.

- http://www.galil.com/sw/pub/all/doc/gclib/html/

Offline html

The following allows viewing of the html docs from the installation.

```
$ tar -xzf /usr/share/doc/gclib/gclib_doc.tar.gz html
$ firefox html/index.html
```

2.5 Red Hat 6 & CentOS 6 Linux

Tested versions

This version of Linux has **x64/AMD64 Support Only**. Contact Galil if another version is required for an application.

See the installation page for supported versions.

Installation

On Red Hat, gclib is distributed in an RPM repository. The following steps can be performed to install gclib.

Download Galil's repository information

This step installs Galil's RPM repositories and only needs to be done once.

Point a browser at http://www.galil.com/sw/pub/rhel/6/galilrpm-2-1.noarch.rpm and install the rpm.

Install Package

Install gclib package, approve "Installed size" and "Importing GPG key", if prompted.

```
# yum install gclib
```

Uninstall Package

To uninstall gclib.

```
# yum remove gclib
```

Serial Ports and USB

If access to the serial ports or USB (e.g. DMC-4103) is desired through gclib, the following will provide steps to join the correct access group. If using USB, be sure the controller is powered and the usb is plugged in before beginning.

Determine group with access

```
$ ls -l /dev/ttyUSB* /dev/ttyS*
crw-rw----. 1 root dialout   4, 64 Mar  3 16:39 /dev/ttyS0
crw-rw----. 1 root dialout   4, 65 Mar  3 16:39 /dev/ttyS1
crw-rw----. 1 root dialout   4, 66 Mar  3 16:39 /dev/ttyS2
crw-rw----. 1 root dialout   4, 67 Mar  3 16:39 /dev/ttyS3
crw-rw----. 1 root dialout 188,  0 Mar  6 11:08 /dev/ttyUSB0
```

In the above listing, **dialout** is the group that needs to be joined. **uucp** is another common group that may be listed.

Add the desired *username* to the group.

```
$ sudo gpasswd -a username dialout
[sudo] password for username:
Adding user username to group dialout
```

Log out and back in for change to take effect.

```
$ groups
username wheel dialout
```

gclib can now connect to serial and usb devices from user *username*.

PCI Controllers

If using a Galil PCI controller, the PCI driver must be installed.

Install prerequisites

```
# yum update kernel
```

Reboot

```
# yum install kernel-devel-$(uname -r)
# yum install kernel-headers-$(uname -r)
# yum install gcc
```

Extract source and build driver

```
$ tar -xf /usr/share/doc/gclib/src/gclib_pci.tar.gz
$ make
```

Copy module and add to kernel

```
# cp galilpci.ko /lib/modules/$(uname -r)
# depmod
# modprobe galilpci
```

Add galil group for access to PCI

```
# groupadd -f -K GID_MIN=100 -K GID_MAX=499 galil
# cp 90-galilpci.rules /etc/udev/rules.d/
# udevadm control --reload-rules
# udevadm trigger
# usermod -a -G galil username  #exchange "username" with actual user's name
```

Logout and back in. The PCI hardware is now available for access.

```
$ ls -l /dev/galil*
crw-rw---- 1 root galil 10, 56 Jun  9 11:07 /dev/galilpci0
$ echo -e "\x12\x16\r" > /dev/galilpci0
$ cat /dev/galilpci0
DMC1846 Rev 1.1a
:
```

Documentation

The documentation is left as a tarball to minimize disk usage. The latest release version of the user manual is available at the following link.

- http://www.galil.com/sw/pub/all/doc/gclib/html/

Offline html

The following allows viewing of the html docs from the installation.

```
$ tar -xzf /usr/share/doc/gclib/gclib_doc.tar.gz html
$ firefox html/index.html
```

2.6 Raspberry Pi

Tested versions

See the installation page for supported versions.

Installation

Create a temporary variable for OS version

```
$ os=$(lsb_release -r -s)
$ echo $os
9.8
```

The number printed must be in the installation list of supported operating systems.

Install Galil's public certificate

```
$ wget http://www.galil.com/sw/pub/raspbian/$os/GALIL-PUB-KEY
$ sudo apt-key add GALIL-PUB-KEY
```

Get Galil's apt sources list

```
$sudo wget http://www.galil.com/sw/pub/raspbian/$os/galil.list -O /etc/apt/sources.list.d/galil.list
$sudo apt-get update
```

Install Package

```
 $sudo apt-get install gclib
```

Uninstall Package

To uninstall gclib.

```
$sudo apt-get remove gclib
```

Serial Ports and USB

If access to the serial ports or USB (e.g. DMC-4103) is desired through gclib, the following will provide steps to join the correct access group. If using USB, be sure the controller is powered and the usb is plugged in before beginning.

Determine group with access

```
$ ls -l /dev/ttyUSB* /dev/ttyS*
crw-rw----. 1 root dialout   4, 64 Mar  3 16:39 /dev/ttyS0
crw-rw----. 1 root dialout   4, 65 Mar  3 16:39 /dev/ttyS1
crw-rw----. 1 root dialout   4, 66 Mar  3 16:39 /dev/ttyS2
crw-rw----. 1 root dialout   4, 67 Mar  3 16:39 /dev/ttyS3
crw-rw----. 1 root dialout 188,  0 Mar  6 11:08 /dev/ttyUSB0
```

In the above listing, **dialout** is the group that needs to be joined. **uucp** is another common group that may be listed.

Check the user's group

The default *pi* username is already a member of dialout.

```
$ groups
pi adm dialout cdrom sudo audio video plugdev games users input netdev gpio i2c spi
```

If needed, add the desired *username* to the group.

```
$ sudo gpasswd -a username dialout
[sudo] password for username:
Adding user username to group dialout
```

Log out and back in for change to take effect.

```
$ groups
username wheel dialout
```

gclib can now connect to serial and usb devices from user *username*.

Documentation

The documentation is left as a tarball to minimize disk usage. The latest release version of the user manual is available at the following link.

- http://www.galil.com/sw/pub/all/doc/gclib/html/

Offline html

The following allows viewing of the html docs from the installation, in the GUI mode.

```
$ tar -xzf /usr/share/doc/gclib/gclib_doc.tar.gz html
$ epiphany html/index.html
```

2.7 Apple OS X

Tested versions

See the installation page for supported versions.

Installation

On OS X, gclib is distributed in a dmg image. The following steps can be performed to install gclib.

Download the gclib dmg

- Open the dmg file and drag the gclib directory to the Applications alias or another installation location.

Create Environment Variable (Optional)

- To provide maximum functionality, e.g. usage of the Python wrapper, add to the *DYLD_LIBRARY_PATH* by typing the following at a Terminal prompt.

```
$ echo "export DYLD_LIBRARY_PATH=/Applications/gclib/dylib/:\$DYLD_LIBRARY_PATH" >> ~/.profile
```

- Log Out and back in to set the environment variable.

Make links for usb devices

If using the DMC4103 or another Galil USB product, symbolic links may be created so GAddresses() can list the controllers.

Make a link from the Terminal.

```
user-mac:~ user$ #plug in DMC4103 usb cable
user-mac:~ user$ ls /dev/tty.usb*
/dev/tty.usbserial-A402L6KG
user-mac:~ user$ #make a symbolic link so gclib can list it
user-mac:~ user$ sudo ln -s /dev/tty.usbserial-A402L6KG /dev/tty.usbserial0
user-mac:~ user$ #gclib searches start at 0
user-mac:~ user$ #GAddresses() will now list this device
```

Demonstrating with Python.

```
user-mac:~ user$ python
Python 2.7.10 (default, Jul 14 2015, 19:46:27)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.39)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import gclib
>>> g = gclib.py()
>>> g.GAddresses()
{'/dev/tty.usbserial0': ''}
>>> g.GOpen("/dev/tty.usbserial0 -d")
>>> print(g.GInfo())
/dev/tty.usbserial0, DMC4143 Rev 1.2b, 9998
>>> g.GClose()
>>> exit()
user-mac:~ user$
```

Installed files

- The gclib shared object files
  - /Applications/gclib/dylib/gclib.0.dylib
  - /Applications/gclib/dylib/gclibo.0.dylib
- The gclib header files
  - /Applications/gclib/include/gclib_errors.h
  - /Applications/gclib/include/gclibo.h
  - /Applications/gclib/include/gclib.h
  - /Applications/gclib/include/gclib_record.h
- gclib documentation tarball
  - /Applications/gclib/doc/gclib_doc.tar.gz
- Example source tarball
  - /Applications/gclib/examples/gclib_examples.tar.gz
- Source files to modify/rebuild libgclibo.so
  - /Applications/gclib/source/gclibo_229_src.tar.gz
- GalilTools Communication Library (gcl) wrapper
  - /Applications/gclib/source/gclib_gcl.tar.gz

Documentation

The documentation is left as a tarball to minimize disk usage. The latest release version of the user manual is available at the following link.

- http://www.galil.com/sw/pub/all/doc/gclib/html/

Offline pdf

The following allows viewing of the pdf docs from the installation.

- Browse in the Finder to Applications/gclib/doc.
- Double-click the tar.gz file to extact it.
- Open the resultant directory.
- Open the pdf.

Offline html

The following allows viewing of the html docs from the installation.

- Browse in the Finder to Applications/gclib/doc.
- Double-click the tar.gz file to extact it.
- Open the resultant directory.
- Open the html directory.
- Double-click index.html to open the help.

Chapter 3

Language Support

Below are a number of examples demonstrating how to use the library with various languages and on various platforms.

- C/C++
- Python
- .Net
- Java

Can't find what you need? Please email `softwaresupport@galil.com`, or call `Galil Applications`.

3.1 C/C++

- Microsoft Visual Studio 2015 (14.0)
- Microsoft Visual Studio 2013 (12.0)
- MinGW
- Borland C++
- gcc (Linux)
- clang (OS X)

3.1.1 Microsoft Visual Studio 2015 (14.0)

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

x_simple.c from *VS2015 x64 Native Tools Command Prompt*

Open *VS2015 x64 Native Tools Command Prompt*.

Copy files

Navigate to a convenient, empty, writable location.

Set an environment variable for the base path.

```
>set base=C:\Program Files (x86)\Galil\gclib
```

Copy simple example

```
>copy "%base%\examples\cpp\x_simple.c" .
```

Edit GOpen() call as necessary

In a text editor, open *x_simple.c*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

Compile

```
>cl x_simple.c "%base%\lib\dynamic\x64\*.lib" -I "%base%\include"
```

Set Path to DLL

```
>set PATH=%base%\dll\x64\;%PATH%
```

Execute

```
>x_simple.exe
version: 211.211.211 1.0.0.128
info: 192.168.0.42, DMCC640 Rev 1.0g, 9999
response:  3757802.0000
:
```

Using the pre-configured MSVC project (x_examples.cpp)

The directory *gclib\examples\msvc* has fully functional MSVC examples. These instructions detail how to use the 2013 version.

- Copy *gclib\examples\msvc\2015_14.0\gclib_example* to a convenient, writable location.
- Run *gclib_example\gclib_example\copy_source.bat* to copy the files.
- Open *gclib_example\gclib_example.sln* in Visual Studio 2015.
- In the *Solution Explorer*, expand the *gclib_example* and expand *Source Files* to show a listing of source.
- Open *x_examples.cpp*.
- Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

Create Project with MSVC 2015 (x_examples.cpp)

The instructions below allow building a project from scratch.

The following instructions were performed on *Visual Studio Professional 2015* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x64**.

- Launch *Visual Studio 2015*.
- Choose *File->New->Project*.
- In the *New Project* dialog, choose *Visual C++->Empty Project*.
- Choose a Name, e.g. **gclib_example**.
- Choose a Location, e.g. *C:\Users\user\Desktop*.
- Check *Create directory for solution*.
- Click *OK*.
- In the *Solution Explorer*, right-click on *Source Files* and choose *Add->Existing Item*.
  - Navigate to the gclib installation directory, then to *examples\cpp* in the installation directory.
  - In *File Name* type **x_*.cpp** and click *Add*, this will filter out the files needed
  - Select all files in the file chooser and click *Add*.
- In the *Solution Explorer* right-click on *gclib_example*, choose *Properties*.
  - Click the *Configuration Manager...* button.
    - Under *Active solution platform:* choose *x64*.
    - Click *Close*.
  - Highlight *Configuration Properties* in the side bar, and set the following project properties.
    - At the top of the window, change *Configuration:* to *All Configurations* and ensure *Platform* lists *Active(x64)*.
    - *Configuration Properties -> C/C++ -> Additional Include Directories* add **C:\Program Files (x86)\Galil\gclib\include**
    - *Configuration Properties -> Linker -> General -> Additional Library Directories* add **C:\Program Files (x86)\Galil\gclib\lib\dynamic\x64**
    - *Configuration Properties -> Linker -> Input -> Additional Dependencies* add **gclib.lib;gclibo.↩ lib;**`{rest of text}` where `{rest of text}` is the original string that was in the cell. Note the semicolons between library files.
    - *Configuration Properties -> Debugging -> Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x64;%PATH%**
    - Click *OK*.
- In the *Solution Explorer* open *x_examples.cpp*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.
- Find the `#if 0` preprocessor blocks enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

3.1.2 Microsoft Visual Studio 2013 (12.0)

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

x_simple.c from *VS2013 x64 Native Tools Command Prompt*

Open *VS2013 x64 Native Tools Command Prompt*.

Copy files

Navigate to a convenient, empty, writable location, e.g. *C:\temp*.

Set an environment variable for the base path.

```
C:\temp>set base=C:\Program Files (x86)\Galil\gclib
```

Copy simple example

```
C:\temp>copy "%base%\examples\cpp\x_simple.c" .
```

Edit GOpen() call as necessary

In a text editor, open *x_simple.c*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

Compile

```
C:\temp>cl x_simple.c "%base%\lib\dynamic\x64\*.lib" -I "%base%\include"
```

Set Path to DLL

```
C:\temp>set PATH=%base%\dll\x64\;%PATH%
```

Execute

```
C:\temp>x_simple.exe
rc: 0
version: 85.60.138
rc: 0
rc: 0
info: 10.1.3.17, DMC4020 Rev 1.2b, 291
rc: 0
response:  357247808.0000
:
```

Using the pre-configured MSVC project (x_examples.cpp)

The directory *gclib\examples\msvc* has fully functional MSVC examples. These instructions detail how to use the 2013 version.

- Copy *gclib\examples\msvc\2013_12.0\gclib_example* to a convenient, writable location, e.g. *C:\temp*.
- Run *gclib_example\gclib_example\copy_source.bat* to copy the files.
- Open *gclib_example\gclib_example.sln* in Visual Studio 2013.
- In the *Solution Explorer*, expand the *gclib_example* and expand *Source Files* to show a listing of source.
- Open *x_examples.cpp*
- Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

Create Project with MSVC 2013 (x_examples.cpp)

The instructions below allow building a project from scratch.

The following instructions were performed on *Visual Studio Professional 2013* and can be extended to other Visual Studio versions. For brevity, the instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib** and a build type of **x86 (win32)**.

- Launch *Visual Studio 2013*
- Choose *File->New->Project*
- In the *New Project* dialog, choose *Visual C++->Empty Project*
- Choose a Name, e.g. **gclib_example**
- Choose a Location, e.g. *C:\Users\user\Desktop*
- Check *Create directory for solution*
- Click *OK*
- In the *Solution Explorer*, right-click on *Source Files* and choose *Add->Existing Item*
  - Navigate to the gclib installation directory, then to *examples\cpp* in the installation directory
  - In *File Name* type **x_∗.cpp** and click *Add*, this will filter out the files needed
  - Select all files in the file chooser and click *Add*
- In the *Solution Explorer* right-click on *gclib_example*, choose *Properties*, highlight *Configuration Properties*, and set the following project properties
  - At the top of the window, change *Configuration:* to *All Configurations* and ensure *Platform* lists *Active(↩ Win32)*
  - *Configuration Properties -> C/C++ -> Additional Include Directories* add **C:\Program Files (x86)\Galil\gclib\include**
  - *Configuration Properties -> Linker -> General -> Additional Library Directories* add **C:\Program Files (x86)\Galil\gclib\lib\dynamic\x86**
  - *Configuration Properties -> Linker -> Input -> Additional Dependencies* add **gclib.lib;gclibo.↩ lib;**`{rest of text}` where `{rest of text}` is the original string that was in the cell. Note the semicolons between library files.
  - *Configuration Properties -> Debugging -> Environment* add **PATH=C:\Program Files (x86)\Galil\gclib\dll\x86;%P↩ ATH%**
- In the *Solution Explorer* open *x_examples.cpp*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.
- Hit *F5* to build and run the example.

3.1.3 MinGW

The following instructions were performed with x86 Minimalist GNU for Windows (MinGW) installed from http://mingw-w64.sourceforge.net/download.php#mingw-builds

For brevity, these instructions assume the default installation location of "C:\Program Files (x86)\Galil\gclib".

Copy Files

Copy "gclib\examples\mingw" to a convenient, writable location, e.g. "C:\temp". Run `C:\temp\mingw\copy_source.bat` to copy all files.

x_simple.c

Edit GOpen() call as necessary

In a text editor, open *x_simple.c*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

Compile

- Launch the MinGW terminal, e.g. *Start* -> *All Programs* -> *MinGW-W64 project* -> *i686-4.9.1-posix-dwarf-rt_v3-rev3* -> *Run Terminal*.
- Navigate to the directory with the files above.
- Compile the code.

```
C:\temp\mingw\>gcc x_simple.c -L. -lgclibo -lgclib -o simple.exe
```

Execute

```
C:\temp\mingw>simple.exe
rc: 0
version: 85.60.138
rc: 0
rc: 0
info: 10.1.3.17, DMC4020 Rev 1.2b, 291
rc: 0
response:  1584328.0000
:
```

x_examples.cpp

Review and Modify source

- In a text editor, open *x_examples.cpp*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

Compile

- Launch the MinGW terminal, e.g. *Start* -> *All Programs* -> *MinGW-W64 project* -> *i686-4.9.1-posix-dwarf-rt_v3-rev3* -> *Run Terminal*.
- Navigate to the directory with the files above.
- Compile the code.

```
C:\temp\mingw>g++ *.cpp -L. -lgclibo -lgclib -o examples.exe
```

Execute

```
C:\temp\mingw>examples.exe
Library version: 41.35.34

192.168.0.43, DMC4020 Rev 1.2b, 291


************************************************************************
Example GRead() and GWrite() usage
************************************************************************

Read 155 QR bytes.

************************************************************************
Example GCommand() usage
************************************************************************
Revision report, ^R^V
DMC4020 Rev 1.2b
:

Command Values
val is 10
val is 11
val is 3.1415
val is 9.869

Command Trimming
> 95653016.0000
:<
> 95653016.0000<
>95653016.0000<

Receiving Binary Data
QR read 155 bytes

Error handling
QD correctly trapped, not allowed, try GArrayDownload()
DL correctly trapped, not allowed, try GProgramDownload()

Modifying timeout
Burning program...OK

************************************************************************
Example GProgramDownload() and GProgramUpload() usage
************************************************************************
GProgramDownload() correctly errored. Can’t fit with level 3 compression
Program Downloaded with compression level 4
Uploading program:
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN

Program executed as expected
************************************************************************
Example GArrayDownload() and GArrayUpload() usage
************************************************************************
2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.000
0000
```

```
2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.000
0000

3.0000, 5.0000, 10.0000

************************************************************************
Example GRecord() usage
************************************************************************

QR-based data record
38564
393216000

DR-based data record
38670
38772
38874
38976
39078
39180
39282
39384
39486
39588
39690

QR-based data record with offsets
39692
39692

************************************************************************
Example GMessage() usage
************************************************************************
 0.0000
 1.0000
 2.0000
 3.0000
 4.0000
 5.0000
 6.0000
 7.0000
 8.0000
 9.0000

************************************************************************
Example GInterrupt() usage
************************************************************************
"UI 8" executed.

************************************************************************
Example GMotionComplete() usage
************************************************************************

Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0

Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
Position: 6000, 0


examples.cpp executed OK
main() is finished. Press Enter to exit:
```

3.1.4 Borland C++

The following instructions were performed on:

```
Embarcadero C++ 7.10 for Win32 Copyright (c) 1993-2015 Embarcadero Technologies, Inc.
```

For brevity, these instructions assume the default installation location of "C:\Program Files (x86)\Galil\gclib".

Copy Files

Copy "gclib\examples\borland" to a convenient, writable location, e.g. "C:\temp". Run `C:\temp\borland\copy_source.bat` to copy all files.

```
C:\temp>cd borland

C:\temp\borland>copy_source.bat
\Program Files (x86)\Galil\gclib\examples\cpp\x_arrays.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_examples.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_examples.h
\Program Files (x86)\Galil\gclib\examples\cpp\x_gcommand.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_ginterrupt.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_gmessage.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_gmotioncomplete.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_gread_gwrite.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_grecord.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_nonblocking.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_programs.cpp
\Program Files (x86)\Galil\gclib\examples\cpp\x_simple.c
       12 file(s) copied.
\Program Files (x86)\Galil\gclib\include\gclib.h
\Program Files (x86)\Galil\gclib\include\gclibo.h
\Program Files (x86)\Galil\gclib\include\gclib_errors.h
\Program Files (x86)\Galil\gclib\include\gclib_record.h
        4 file(s) copied.
\Program Files (x86)\Galil\gclib\lib\dynamic\x86\gclib.lib
\Program Files (x86)\Galil\gclib\lib\dynamic\x86\gclibo.lib
        2 file(s) copied.
\Program Files (x86)\Galil\gclib\dll\x86\gclib.dll
\Program Files (x86)\Galil\gclib\dll\x86\gclibo.dll
        2 file(s) copied.

C:\temp\borland>
```

Modify Path

- Add Borland's compiler to the `PATH` variable.

```
C:\temp\borland>set PATH=c:\Program Files (x86)\Embarcadero\Studio\17.0\bin;%PATH%
```

Convert lib files

```
C:\temp\borland>move gclib.lib _gclib.lib
    1 file(s) moved.

C:\temp\borland>move gclibo.lib _gclibo.lib
    1 file(s) moved.

C:\temp\borland>coff2omf.exe _gclib.lib gclib.lib
COFF to OMF Converter Version 1.2.0 Copyright (c) 1999-2009 Embarcadero Technologies, Inc.
All rights reserved.

C:\temp\borland>coff2omf.exe _gclibo.lib gclibo.lib
COFF to OMF Converter Version 1.2.0 Copyright (c) 1999-2009 Embarcadero Technologies, Inc.
All rights reserved.
```

x_simple.c

Edit GOpen() call as necessary

In a text editor, open *x_simple.c*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

Compile

```
C:\temp\borland>bcc32 gclib.lib gclibo.lib x_simple.c
Embarcadero C++ 7.10 for Win32 Copyright (c) 1993-2015 Embarcadero Technologies, Inc.
x_simple.c:
Turbo Incremental Link 6.72 Copyright (c) 1997-2015 Embarcadero Technologies, Inc.
```

Execute

```
C:\temp\borland>x_simple.exe
version: 130.115.279
info: 192.168.0.43, DMC4143 Rev 1.2b, 9998
response:  61016.0000
:
```

x_examples.cpp

Review and Modify source

- In a text editor, open *x_examples.cpp*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

Compile

```
C:\temp\borland>bcc32 -c  *.cpp
```

Link

```
C:\temp\borland>bcc32 -o examples.exe *.obj gclib.lib gclibo.lib
```

Execute

```
C:\temp\borland>examples.exe
Library version: 130.115.279

192.168.0.43, DMC4020 Rev 1.2b, 291


************************************************************************
Example GRead() and GWrite() usage
************************************************************************

Read 155 QR bytes.

************************************************************************
Example GCommand() usage
************************************************************************
Revision report, ^R^V
DMC4020 Rev 1.2b
:
```

```
Command Values
val is 10
val is 11
val is 3.1415
val is 9.869

Command Trimming
> 95653016.0000
:<
> 95653016.0000<
>95653016.0000<

Receiving Binary Data
QR read 155 bytes

Error handling
QD correctly trapped, not allowed, try GArrayDownload()
DL correctly trapped, not allowed, try GProgramDownload()

Modifying timeout
Burning program...OK

************************************************************************
Example GProgramDownload() and GProgramUpload() usage
************************************************************************
GProgramDownload() correctly errored. Can't fit with level 3 compression
Program Downloaded with compression level 4
Uploading program:
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN

Program executed as expected
************************************************************************
Example GArrayDownload() and GArrayUpload() usage
************************************************************************
2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.000
0000

2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.000
0000

3.0000, 5.0000, 10.0000

************************************************************************
Example GRecord() usage
************************************************************************

QR-based data record
38564
393216000

DR-based data record
38670
38772
38874
38976
39078
39180
39282
39384
39486
39588
39690

QR-based data record with offsets
39692
39692

************************************************************************
Example GMessage() usage
************************************************************************
 0.0000
 1.0000
 2.0000
 3.0000
 4.0000
 5.0000
 6.0000
 7.0000
 8.0000
 9.0000

************************************************************************
Example GInterrupt() usage
************************************************************************
"UI 8" executed.

************************************************************************
Example GMotionComplete() usage
************************************************************************

Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0

Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
Position: 6000, 0


examples.cpp executed OK
main() is finished. Press Enter to exit:
```

3.1.5 gcc (Linux)

The following instructions were performed on

```
$ uname -a
Linux localhost.localdomain 3.17.4-301.fc21.x86_64 #1 SMP Thu Nov 27 19:09:10 UTC 2014 x86_64 x86_64 x86_64 GN
$ g++ --version
g++ (GCC) 4.9.2 20150212 (Red Hat 4.9.2-6)
```

Copy Files

```
$ mkdir test
$ cd test
$ tar -xzf /usr/share/doc/gclib/src/gclib_examples.tar.gz
$ ls
x_arrays.cpp    x_gcommand.cpp    x_gmotioncomplete.cpp  x_programs.cpp
x_examples.cpp  x_ginterrupt.cpp  x_gread_gwrite.cpp     x_simple.c
x_examples.h    x_gmessage.cpp    x_grecord.cpp
```

x_simple.c

- In a text editor, open *x_simple.c*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

Compile

```
$ gcc -Wall -Werror x_simple.c -lgclib -lgclibo -o simple
```

Run

```
$ ./simple
rc: 0
version: 85.60.131
```

```
rc: 0
rc: 0
info: 10.1.3.17, DMC4020 Rev 1.2b, 291
rc: 0
response:  179340166.0000
:
```

x_examples.cpp

- In a text editor, open *x_examples.cpp*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options. Don't forget `-s ALL` if data records, interrupts, and messages are to be tested.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

Compile

```
$ g++ x_*.cpp -lgclib -lgclibo -o example
```

Run

$./example Library version: 85.60.131

10.1.3.17, DMC4020 Rev 1.2b, 291

Example GRead() and GWrite() usage

Read 155 QR bytes.

Example GCommand() usage

Revision report, ^R^V DMC4020 Rev 1.2b :

Command Values val is 10 val is 11 val is 3.1415 val is 9.869

Command Trimming > 179798738.0000 :< > 179798738.0000< >179798738.0000<

Receiving Binary Data QR read 155 bytes

Error handling QD correctly trapped, not allowed, try GArrayDownload() DL correctly trapped, not allowed, try GProgramDownload()

Modifying timeout Burning program...OK

Example GProgramDownload() and GProgramUpload() usage

GProgramDownload() correctly errored. Can't fit with level 3 compression Program Downloaded with compression level 4 Uploading program: #A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1 i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN

Program executed as expected

Example GArrayDownload(), GArrayUploadFile() GArrayDownloadFile(), and GArrayUpload usage

2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

3.0000, 5.0000, 10.0000 2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

Example GRecord() usage

QR-based data record 36100 6000

DR-based data record 36204 36306 36408 36510 36612 36714 36816 36918 37020 37122 37224

QR-based data record with offsets 37224 37224

Example GMessage() usage

0.0000 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 7.0000 8.0000 9.0000

Example GInterrupt() usage

"UI 8" executed.

Example GMotionComplete() usage

Position: 0, 0 Beginning independent motion... Motion Complete on A Position: 8000, 0

Position: 0, 0 Beginning vector motion... Motion Complete on vector plane S Position: 6000, 0

examples.cpp executed OK main() is finished. Press Enter to exit:

3.1.6 clang (OS X)

The following instructions were performed on

```
$ sw_vers
ProductName:  Mac OS X
ProductVersion: 10.10.5
BuildVersion: 14F27
$ gcc --version
Configured with: --prefix=/Library/Developer/CommandLineTools/usr --with-gxx-include-dir=/usr/include/c++/4.2.
Apple LLVM version 6.1.0 (clang-602.0.53) (based on LLVM 3.6.0svn)
Target: x86_64-apple-darwin14.5.0
Thread model: posix
```

Copy Files

```
$ cd ~
$ mkdir test
$ cd test
$ tar -xzf /Applications/gclib/examples/gclib_examples.tar.gz
$ cp /Applications/gclib/include/* .
$ cp /Applications/gclib/dylib/* .
$ ls
gclib.0.dylib   x_arrays.cpp    x_gmotioncomplete.cpp
gclib.h     x_examples.cpp    x_gread_gwrite.cpp
gclib_errors.h    x_examples.h    x_grecord.cpp
gclib_record.h    x_gcommand.cpp    x_nonblocking.cpp
gclibo.0.dylib    x_ginterrupt.cpp  x_programs.cpp
gclibo.h    x_gmessage.cpp    x_simple.c
```

x_simple.c

- In a text editor, open *x_simple.c*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options.

Compile

```
$ gcc -Wall -Werror x_simple.c gclib.0.dylib gclibo.0.dylib -o simple
```

Run

```
$ ./simple
rc: 0
version: 126.108.229
rc: 0
```

```
rc: 0
info: 10.1.3.142, DMC4020 Rev 1.2a-BH, 291
rc: 0
response:  206676.0000
:
```

x_examples.cpp

- In a text editor, open *x_examples.cpp*. Find the GOpen() call and update the address to match the desired hardware. See the documentation for GOpen() for address formatting options. Don't forget `-s ALL` if data records, interrupts, and messages are to be tested.
- Find the `#if 0` preprocessor block enclosing the example calls. Change to `#if 1` to run the examples. Comment out the function calls to be avoided. Note some calls attempt to move motors and not all functions are compatible with all Galil products.

Compile

```
$ g++ x_*.cpp gclib.0.dylib gclibo.0.dylib -o example
```

Run

```
$ ./example
Library version: 126.108.229

10.1.3.142, DMC4020 Rev 1.2a-BH, 291

************************************************************************
Example GRead() and GWrite() usage
************************************************************************

Read 1 byte(s)
:
Program test OK.

************************************************************************
Example GCommand() usage
************************************************************************
Revision report, ^R^V
DMC4020 Rev 1.2a-BH
:

Command Values
val is 10
val is 11
val is 3.1415
val is 9.869

Command Trimming
> 408978.0000
:<
> 408978.0000<
>408978.0000<

Receiving Binary Data
QR read 155 bytes

Error handling
QD correctly trapped, not allowed, try GArrayDownload()
DL correctly trapped, not allowed, try GProgramDownload()

Modifying timeout
Burning program...OK

************************************************************************
Example GProgramDownload() and GProgramUpload() usage
************************************************************************
```

```
GProgramDownload() correctly errored. Can’t fit with level 3 compression
Program Downloaded with compression level 4
Uploading program:
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN

Program executed as expected
************************************************************************
Example GArrayDownload(), GArrayUploadFile()
GArrayDownloadFile(), and GArrayUpload usage
************************************************************************
2.0000, 4.0000, 6.0000, 8.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000

3.0000, 5.0000, 10.0000
2.0000, 1.0000, 3.0000, 5.0000, 10.0000, 12.0000, 14.0000, 16.0000, 18.0000, 20.0000


************************************************************************
Example GRecord() usage
************************************************************************

QR-based data record
18358
0

DR-based data record
18462
18564
18666
18768
18870
18972
19074
19176
19278
19380
19482

QR-based data record with offsets
19482
19482

************************************************************************
Example GMessage() usage
************************************************************************
 0.0000
 1.0000
 2.0000
 3.0000
 4.0000
 5.0000
 6.0000
 7.0000
 8.0000
 9.0000

************************************************************************
Example GInterrupt() usage
************************************************************************
"UI 8" executed.

************************************************************************
Example GMotionComplete() usage
************************************************************************

Position: 0, 0
Beginning independent motion... Motion Complete on A
Position: 8000, 0

Position: 0, 0
Beginning vector motion... Motion Complete on vector plane S
```

```
Position: 6000, 0

************************************************************************
Example GMessage non-blocking usage
************************************************************************
 422902.0000


************************************************************************
Example GInterrupt non-blocking usage
************************************************************************
 F1

************************************************************************
Example GRecord non-blocking usage
************************************************************************
 33786


examples.cpp executed OK
main() is finished. Press Enter to exit:
```

3.2 Python

Install gclib

The gclib Python wrapper assumes the default gclib installation location.

Install Python

- See `https://www.python.org/` if Python is not already installed on the system. The gclib Python wrapper supports Python versions 2 and 3.
- On Windows, choose to add Python to the environment variable during installation. This allows Python to be invoked from the command line.

Install the gclib Python module

Windows

- Type the following commands into a command prompt.

```
>cd %temp%
>mkdir py
>cd py
>copy "c:\Program Files (x86)\Galil\gclib\source\wrappers\python\*" .
c:\Program Files (x86)\Galil\gclib\source\wrappers\python\gclib.py
c:\Program Files (x86)\Galil\gclib\source\wrappers\python\setup.py
    2 file(s) copied.
>copy "c:\Program Files (x86)\Galil\gclib\examples\python\*" .
c:\Program Files (x86)\Galil\gclib\examples\python\example.py
    1 file(s) copied.
>python setup.py install
running install
running build
running build_py
creating build
creating build\lib
copying gclib.py -> build\lib
running install_lib
running install_egg_info
Removing C:\Users\user\AppData\Local\Programs\Python\Python37-32\Lib\site-packages\gclib-1.0-py3.7.egg-info
Writing C:\Users\user\AppData\Local\Programs\Python\Python37-32\Lib\site-packages\gclib-1.0-py3.7.egg-info
```

- The gclib Python wrapper is now installed. Go to the next section, **Using gclib from the Python Interpreter**.

Linux

- Type the following commands into a terminal prompt.

```
$ mkdir ~/py
$ cd ~/py
$ tar -xvf /usr/share/doc/gclib/src/gclib_python.tar.gz
gclib.py
setup.py
$ tar -xvf /usr/share/doc/gclib/src/gclib_python_examples.tar.gz
example.py
$ sudo python setup.py install
[sudo] password for user:
running install
running build
running build_py
creating build
creating build/lib
copying gclib.py -> build/lib
running install_lib
copying build/lib/gclib.py -> /usr/lib/python2.7/site-packages
byte-compiling /usr/lib/python2.7/site-packages/gclib.py to gclib.pyc
running install_egg_info
Writing /usr/lib/python2.7/site-packages/gclib-1.0-py2.7.egg-info
```

- The gclib Python wrapper is now installed. Go to the next section, **Using gclib from the Python Interpreter**.

OS X

- Be sure that the *Create Environment Variable* step has been followed in the OS X installation instructions.
- Type the following commands into a Terminal prompt.

```
$ mkdir ~/python_temp
$ cd ~/python_temp/
$ tar -xvf /Applications/gclib/source/gclib_python.tar.gz
x gclib.py
x setup.py
$ tar -xvf /Applications/gclib/examples/gclib_python_examples.tar.gz
x example.py
$ sudo python setup.py install
running install
running build
running build_py
creating build
creating build/lib
copying gclib.py -> build/lib
running install_lib
copying build/lib/gclib.py -> /Library/Python/2.7/site-packages
byte-compiling /Library/Python/2.7/site-packages/gclib.py to gclib.pyc
running install_egg_info
Writing /Library/Python/2.7/site-packages/gclib-1.0-py2.7.egg-info
```

- The gclib Python wrapper is now installed. Go to the next section, **Using gclib from the Python Interpreter**.

Using gclib from the Python Interpreter

- Invoke the `Python Interpreter`.
- Type the following into the Python prompt.

```
>>> import gclib
>>> g = gclib.py()
>>> g.GOpen('192.168.0.42 --direct')
>>> print(g.GInfo())
192.168.0.42, DMC4080 Rev 1.2c, 783
```

Running Python scripts

- Navigate the terminal to the location from **Install the gclib Python module** where example.py was copied.
- Open *example.py* in a text editor.
- Set the address in the g.GOpen() call to match an avaiable connection.
- Execute the following command at the Terminal.

```
$ python example.py
gclib version: py.127.110.250
192.168.0.42, DMC4080 Rev 1.2c, 783
```

- Experiment with the example by uncommenting sections, between the triple quotes, "'.

```
$ python example.py
gclib version: py.127.110.250
192.168.0.42, DMC4080 Rev 1.2c, 783
GProgramDownload() correctly errored. Can't fit with level 3 compression
Uploaded program:
#A;i=0;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1
i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;i=i+1;EN
Downloaded program verified
Array element verified
 187942.0000

Starting move...
done.
```

Getting help

>>> help(g.GOpen) Help on method GOpen in module gclib:

GOpen(address) method of gclib.py instance Opens a connection a galil controller. See the gclib docs for address string formatting. See Link GOpen() http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h_aef4aec8a85630eed029b7a46aea7db54.html#aef4aec8a85630eed029b7a46aea7db54

>>> help(g.GCommand) Help on method GCommand in module gclib:

GCommand(command) method of gclib.py instance Performs a command-and-response transaction on the connection. Trims the response. See Link GCommand() http://www.galil.com/sw/pub/all/doc/gclib/html/gclib_8h_a5ac031e76efc965affdd73a1bec084a8.html#a5ac031e76efc965affdd73a1bec084a8

>>> 'for a full listing, try help(g)'

3.3 .Net

- VB.NET
- C#.NET

3.3.1 VB.NET

gclib ships with *gclib.vb*, a Visual Basic class which exposes the functionality of the gclib. In addition, a VB forms example is included which demonstrates how to use *gclib.vb*. The following instructions were performed on Visual Studio Professional 2013 and can be extended to other Visual Studio versions.

Running the included Visual Basic Example

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

Copy files

- Navigate to a convenient, empty, writable location, e.g. *C:\temp*.
- Copy the contents of *C:\Program Files (x86)\Galil\gclib\examples\vb\2013_12.0\gclib_example* to this location.

Open in Microsoft Visual Studio 2013

- Open *gclib_example.sln* in Visual Studio. This demo was tested on MSVS 2013.

Add existing item, *gclib.vb*

- In the *Solution Explorer*, right-click on *gclib_example* and choose *Add->Existing Item...*
- Choose *C:\Program Files (x86)\Galil\gclib\source\wrappers\vb\gclib.vb*

Run Demo

- Type *F5* to run the program.
- Type a valid GOpen() address in the text box and click Go.

Create Project from scratch with MSVC 2013

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

Configure Project

- Launch Visual Studio 2013
- Choose File->New->Project
- In the *New Project* dialog, choose Visual Basic -> Windows Forms Application
- Type *gclib_example* for the Name
- Choose a Location, e.g. C:\Users\user\Desktop
- Check *Create directory for solution*
- Click OK, the project will configure itself
- In the *Solution Explorer*, right click on *Solution 'gclib_example' (1 project)* and choose *Configuration Manager...*
  - In the *gclib_example* project row, click in the *Platform* column and choose <New...>
    - Choose *x86* from *Type or select the new platform:*
    - Choose *Any CPU* from *Copy settings from:*
    - Check *Create new solutions platform*
    - Click OK.
  - If x64 support is also desired, repeat the <New...> procedure for *x64*

  - In the *Active solution platform* combobox at the top of the *Configuration Manager* dialog, choose <Edit...>
    - Select *Any CPU* and click the *Remove* button
    - Click *Close*
  - Close the *Configuration Manager* dialog

- In the *Solution Explorer*, right-click on *gclib_example* and choose Add->Existing Item
  - Navigate to the installation location C:\Program Files (x86)\Galil\gclib\source\wrappers\vb
  - Choose *gclib.vb*

- In the *Solution Explorer* double-click on *gclib.vb*
  - Note that there is a preprocessor definition starting with `#if PLATFORM = "x86" Then` and `#ElseIf PLATFORM = "x64" Then`
  - Note that these sections of code enable/disable with the choice of the *Solution Platform* x86/x64, usually found in the Visual Studio toolbar
  - If a non-default gclib installation location is used, the paths in these sections of code must be updated to reflect the dll locations

Add some simple code

- In the *Solution Explorer* right-click on *Form1.vb* and choose *View Code*

- Replace the text in *Form1.vb* with the following code

```
Public Class Form1
    Dim gclib As New Gclib()
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Me.Text = "gclib simple example"
        Dim tb As New TextBox
        With tb
            .Multiline = True
            .Dock = DockStyle.Fill
            .Parent = Me
            Try
                'calls to gclib should be in a try-catch
                .AppendText("GVersion: " & gclib.GVersion() & vbCrLf)
                gclib.GOpen("192.168.0.42 -d") 'Set an appropriate IP address here
                .AppendText("GInfo: " & gclib.GInfo() & vbCrLf)
                .AppendText("GCommand: " & gclib.GCommand("MG TIME") & vbCrLf)
            Catch ex As Exception
                .AppendText("ERROR: " & ex.Message)
      Finally
        gclib.GClose() ' Don't forget to close!
            End Try

        End With
    End Sub
End Class
```

- Hit *F5* to run the project

3.3.2 C#.NET

gclib ships with *gclib.cs*, a C# class which exposes the functionality of the gclib. In addition, a C# forms example is included which demonstrates how to use *gclib.cs*.

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

Running the C# Example

Copy files

- Navigate to a convenient, empty, writable location, e.g. *C:\temp*.
- Copy the contents of *C:\Program Files (x86)\Galil\gclib\examples\cs\2013_12.0\gclib_example* to this location.

Open in Microsoft Visual Studio 2013

- Open *gclib_example.sln* in Visual Studio. This demo was tested on MSVS 2013.

Add existing item, *gclib.cs*

- In the *Solution Explorer*, right-click on *gclib_example* and choose *Add->Existing Item...*
- Choose *C:\Program Files (x86)\Galil\gclib\source\wrappers\cs\gclib.cs*

Run Demo

- Type *F5* to run the program.
- Type a valid GOpen() address in the text box and click Go.

Create Project from scratch with MSVC 2013

For brevity, these instructions assume the default installation location of **C:\Program Files (x86)\Galil\gclib**.

Configure Project

- Launch Visual Studio 2013
- Choose File->New->Project
- In the *New Project* dialog, choose Visual C# -> Windows Forms Application
- Type *gclib_example* for the Name
- Choose a Location, e.g. C:\Users\user\Desktop
- Check *Create directory for solution*
- Click OK, the project will configure itself
- In the *Solution Explorer*, right click on *Solution 'gclib_example' (1 project)* and choose *Configuration Manager...*
  - In the *gclib_example* project row, click in the *Platform* column and choose <New...>
    - Choose *x86* from *Type or select the new platform:*
    - Choose *Any CPU* from *Copy settings from:*
    - Check *Create new solutions platform*
    - Click OK.
  - If x64 support is also desired, repeat the <New...> procedure for *x64*
  - In the *Active solution platform* combobox at the top of the *Configuration Manager* dialog, choose <Edit...>
    - Select *Any CPU* and click the *Remove* button

        - Click *Close*
    - Close the *Configuration Manager* dialog
- In the *Solution Explorer*, right-click on *gclib_example* and choose *Properties*
    - Choose the *Build* item on the left
        - In the *Configuration:* combobox, choose *All Configurations*
        - Choose *x86* from the *Platform* combobox
        - In *Conditional compilation symbols* type *x86*
    - If x64 is to be used also, add an *x64* token as well to the *x64 Platform*
    - Save and close the *Properties* window
- In the *Solution Explorer*, right-click on *gclib_example* and choose Add->Existing Item
    - Navigate to the installation location C:\Program Files (x86)\Galil\gclib\source\wrappers\cs
    - Choose *gclib.cs*
- In the *Solution Explorer* double-click on *gclib.cs*
    - Note that there is a preprocessor definition starting with `#if x86` and `#elif x64`
    - Note that these sections of code enable/disable with the choice of the *Solution Platform* x86/x64, usually found in the Visual Studio toolbar
    - If a non-default gclib installation location is used, the paths in these sections of code must be updated to reflect the dll locations

Add some simple code

- In the *Solution Explorer* right-click on *Form1.cs* and choose *View Code*
- Replace the text in *Form1.vb* with the following code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace gclib_example
{
    public partial class Form1 : Form
    {
        gclib gclib = new gclib();
        public Form1()
        {

            InitializeComponent();
            this.Text = "gclib simple example";
            TextBox tb = new TextBox();
            tb.Multiline = true;
            tb.Dock = DockStyle.Fill;
            tb.Parent = this;
            try
            {
                //calls to gclib should be in a try-catch
                tb.AppendText("GVersion: " + gclib.GVersion() + "\n");
                gclib.GOpen("192.168.0.42 -d"); //Set an appropriate IP address here
                tb.AppendText("GInfo: " + gclib.GInfo() + "\n");
                tb.AppendText("GCommand: " + gclib.GCommand("MG TIME") + "\n");

            }
            catch(Exception ex)
            {
                tb.AppendText("ERROR: " + ex.Message);

            }
            finally
            {
                gclib.GClose(); //Don't forget to close!
            }

        }
    }
}
```

- Hit *F5* to run the project

3.4 Java

gclib uses the venerable

Java Native Access (JNA) library to simplify integration into the Java Native Interface (JNI).

Attention

> This is the initial version of the the gclib Java wrapper. As such, GclibJava ships as source files, not the compiled jar files. All functions are subject to change in future releases of gclib. Java hackers with recommendations on how to make this library better are encouraged to email softwaresupport@galil.com. Somebody has to teach those Galil Java noobs what's what.

Windows

The following instructions were performed with 64 bit Windows 7 on Oracle NetBeans IDE 8.2 and Java 1.8.0_131.

For brevity, these instructions assume the default gclib installation location of "C:\Program Files (x86)\Galil\gclib".

Step-by-Step

1. Install gclib with 64 bit binaries (default install).
2. Install 64 bit NetBeans and Java, jdk-8u131-nb-8_2-windows-x64.exe.
3. Launch NetBeans.
4. Create a new application.
   (a) File | New Project...
   (b) Under *Categories*, select *Java*.
   (c) Under *Projects*, select *Java Application*.
   (d) Click *Next*.
   (e) Type `GclibTest` for the *Project Name*.
   (f) Note the location of the *Project Folder*.
   (g) Uncheck *Create Main Class*
   (h) Click *Finish*
5. Open the *Project Folder* as noted above.
6. Open the *src* directory in the *Project Folder* location.
7. Copy the whole directory *C:\Program Files (x86)\Galil\gclib\examples\java\gclibtest* to this directory.

8. Copy the whole directory *C:\Program Files (x86)\Galil\gclib\source\wrappers\java\gclibjava* to this directory.
9. Create a directory at `c:\jna\`.
   - Another directory may be chosen. The purpose of this directory is to hold jna's *jar* binary for the Java classpath.
10. Download a copy of *jna.jar* to the new directory.
    - `https://github.com/java-native-access/jna#download`
    - This example uses *jna-4.4.0.jar*.
11. In the NetBeans *Projects* tab, expand *GclibTest*.
12. Right-click on *Libraries* and choose *Add JAR/Folder...*.
13. Navigate to the *jna.jar* saved above. Click *Open* to add *jna.jar* to the classpath.
14. In the NetBeans *Projects* tab, right-click on *GclibTest* and choose *Properties*.
15. Choose the *Run* item out of the *Categories* options tree.
16. In the *Main Class* text box, type `gclibtest.GclibTest`. Click *OK*.
17. In the NetBeans *Projects* tab, expand *GclibTest* | *Source Packages* | *gclibtest*.
18. Double click *GclibTest.java*, and find the line containing `gclib.GOpen`.
19. Update the address for the desired hardware.
20. Choose *Run* | *Run Project (GclibTest)* or hit the `F6` key to run the application.
21. The appplication output will print in the NetBeans *Output* window.

Documentation

The GclibJava class has helpful documentation for developing a Java application. Use the following instructions to create the Javadoc.

1. In the NetBeans *Projects* tab, right-click *GclibTest*.
2. Choose *Generate Javadoc* to create the documentation and open it in the system's default browser.