Lecture 00.06   Exploring C—building a sandbox

When beginning with any programming language, it helps to have a sort of "sandbox" in which one can play. If one follows the instructions in Resource 7 to set up the same environment used in the lab to compile for the myRIO, one can develop C programs in Eclipse at home. Even without connecting to a myRIO, one can compile (Eclipse calls this *build*) a myRIO **build** program to check for syntax and other errors.

Taking it one step further, one can install another C compiler made for the development computer. This compiler may differ in certain aspects, but typically these differences are insignificant.

A good compiler to use for this purpose is the free GNU C compiler GCC. It is available on most platforms.

It is important to note that the compiler used in the class is provided by C/C++ Development Tools for NI Linux Real-Time. It is based on GCC, but may have different functionality.

00.06.1   Installing and using GCC on Windows

Download minGW from this link. Run the installer and include the GUI interface. Open the interface and, under the `Basic Setup` tab, check the boxes for `mingw32-base` and `mingw-gcc-g++`. Then select menu item `Installation` ⟩ `Update Catalog`.

Probably, there is no need to, but if the following step fails, try adding the minGW installation directory's `bin` to the system `PATH` environment variable.

Restart Eclipse. Select menu item `File` ⟩ `New` ⟩ `C Project`. Select from the left `Project Type` menu `Executable` ⟩ `Hello World ANSI C Project`. From the right `Toolchains` menu choose `minGW GCC`. Name the project (say) `my_project` and select `Finish`.

In the *C/C++ Perspective*, under the `Project Explorer`, select `my_project`. Build it by selecting menu item `Project` ⟩ `Build Project`. Run it by selecting menu item `Run` ⟩ `Run As` ⟩ `1 Local C/C++ Application`.

If everything is working, you should see the "hello world" message display in the console. You can now edit the project, build, and run at will!

Of course, device driver functions like `fgets_keypad` still won't work in this environment because our system isn't connected to this hardware. However, many analogous functions can be substituted, like `fgets`.

00.06.2   Installing and using GCC on macOS

For those using macOS, the following instructions will help installing and using GCC. The following instructions assume you are using a terminal (e.g. Terminal.app) to execute the commands.

First, install the package manager Homebrew. Homebrew is great for getting and maintaining other software, too! It will install GCC with the following command.

```
brew install gcc
```

Now, just write a C program! Let's say you have `hi.c` in the current directory with contents as follows.

```
#include "stdio.h"

int main()
{
  printf("Hello World\n");
}
```

Compile this with GCC using the following command.

```
gcc -o hi hi.c
```

This compiles `hi.c` to the output executable file `hi` in the working directory. Now, try it out!

```
./hi # => prints Hello World to terminal
```

Alternatively, the Eclipse IDE can be used to write and debug GCC-compiled programs. The configuration is analogous to that for a Windows machine, described above.