New EMAC OE SDK Project Using Eclipse

The purpose of this guide is to demonstrate how to create a new project in Eclipse using the EMAC Open Embedded SDK. This will include a step-by-step procedure which assumes that the SDK has already been imported into Eclipse.

Since this guide is specific to the EMAC OE SDK, it is not interchangeable with a general guide on creating a new Eclipse project.

Table 1. Conventions Used

`/path/to/sdk/EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects`	Indicates the Project Explorer directory where the SDK example projects can be found.
`/path/to/sdk/`	Refers to the full path name where the EMAC OE SDK was extracted during install.
`/path/to/workspace/`	Refers to the full path to the current Eclipse workspace, including the workspace directory name.
`new_project`	The new project being created.
`app_name`	Refers to the target name of the application being developed.
`/app_dir/`	Refers to the directory name where the source files for `app_name` are stored. Expected to be relative to the `/path/to/sdk/EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` directory.

Not all EMAC OE SDKs will be named `EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/`. The pattern is actually `EMAC-OE-target_architecture-SDK_XX.YY/`, where `target_architecture` represents the name of the toolchain compiled for that version of the SDK.

Tools

- GNU 'Make' (For Documentation, please see the GNU 'Make' official website. (http://www.gnu.org/software/make/))
- EMAC Eclipse 3.6
- EMAC OE SDK

New Project Setup

Before getting started creating a new project, the following setup must be performed. It is up to the application developer whether the new project will be placed in the `EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` project folder in the Eclipse Project Explorer View or within its own project folder. The latter does involve more complexity during setup, though it is a bit neater to work with since it does not require the developer to navigate the SDK directory tree to get to the new project.

Note that only one of the following subsections is necessary to move on to the General Setup Section.

Create a New SDK Project within EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects

Follow the steps in this guide to create a new folder within the `EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` directory. This project will compile using a Makefile as described in the Makefile Guide below.

1. Ensure that the C/C++ Perspective is selected and that the EMAC OE SDK project (EMAC-OE-arm-linux-gnueabi-SDK_XX.YY) is listed in the Project Explorer and Make Target views. Both views will be referenced in this guide.
2. Create a new folder in `EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` directory. The new folder must be created in this directory. Otherwise the relative path to global.properties will be incorrect and the new

project will not cross compile correctly.

1. Select *File → New → Folder*.
2. Choose *EMAC-OE-arm-linux-gnueabi-SDK_XX.YY → projects* in the New Folder window as shown in Figure 1. Write the name of the project in the Folder Name field and click Finish.

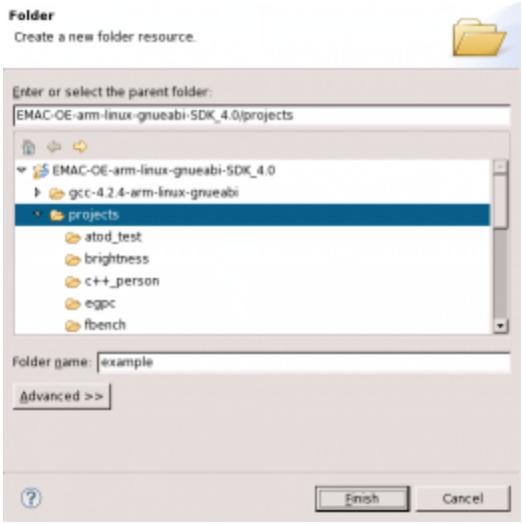

Figure 1. Create New Folder

3. If this project uses existing source files, copy them to this folder using a graphical file browser or shell commands. *Do not link them here* or make will put the targets in the directory containing the actual files!
4. Download and configure the SDK Makefile according to the SDK New Project Guide. Once finished, save the Makefile in the same directory as the other source files.

Create a New SDK Project Outside of EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects

The purpose of this section is to demonstrate the creation of a new EMAC OE SDK project *outside* of the `EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` directory as seen in the Eclipse Project Explorer View.

1. Ensure that the C/C++ Perspective is selected.
2. Create a new Empty Project. See Figure 2 for an example of how to set up in the *C Project* dialog window.
    1. From the workspace, select *File → New → C Project*
    2. In the *C Project* dialog window, enter "hello" in the *Project name:* field.
    3. Choose *Makefile Project → Empty Project* in the *Project type:* list.
    4. Ensure that *– Other Toolchain –* is selected in the *Toolchains:* list.
    5. Click the *Finish* button.

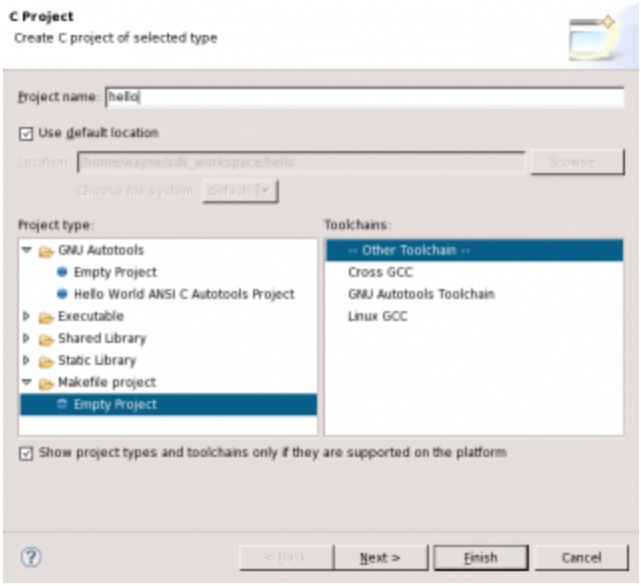

Figure 2. Create New Project Outside the SDK

3. Once these steps are complete, the workbench should look similar to Figure 3 below.

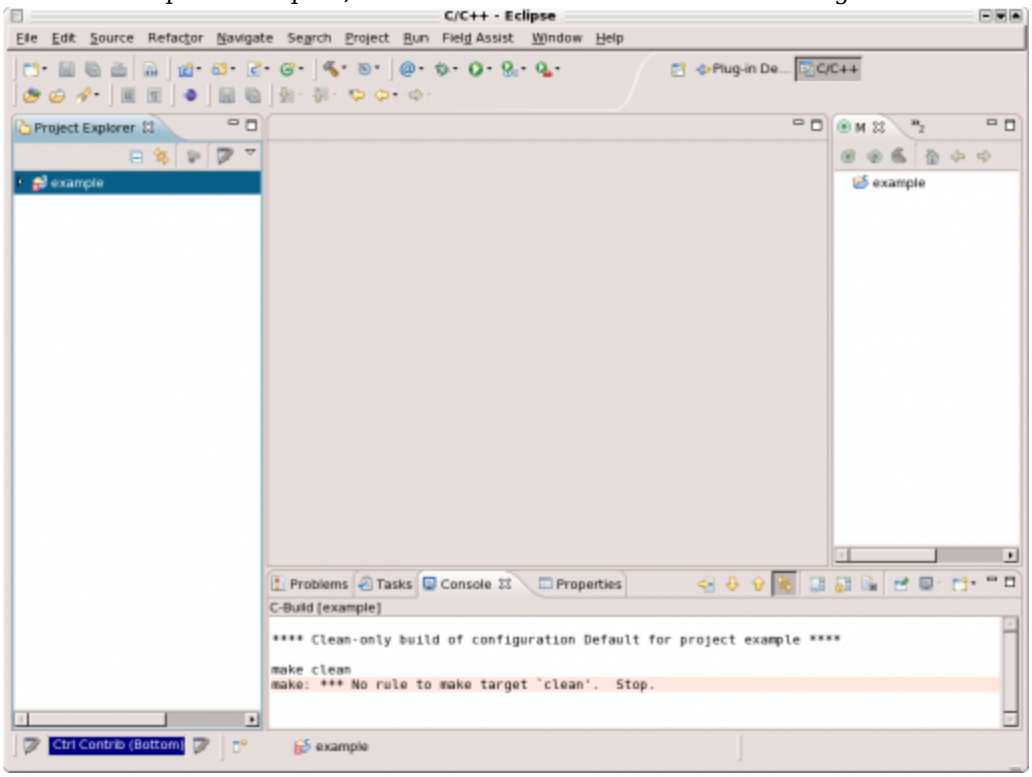

Figure 3. New Project Workbench

General Setup

This setup section applies to both of the above New Project Creation options. Once one of those setup procedures has been followed, the following procedure can be followed independent of what was chosen above unless otherwise noted.

Generate Eclipse Make Targets

This guide will demonstrate how to add `all`, `clean`, and `upload` to the list of make targets in the Make Target

view in Eclipse.

1. Bring up the Create Make Target dialog:
    1. Navigate to `example` under the `EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` directory in the Eclipse Make Target view.
    2. Right click on `example` to bring up the context menu.
    3. Select *New...*

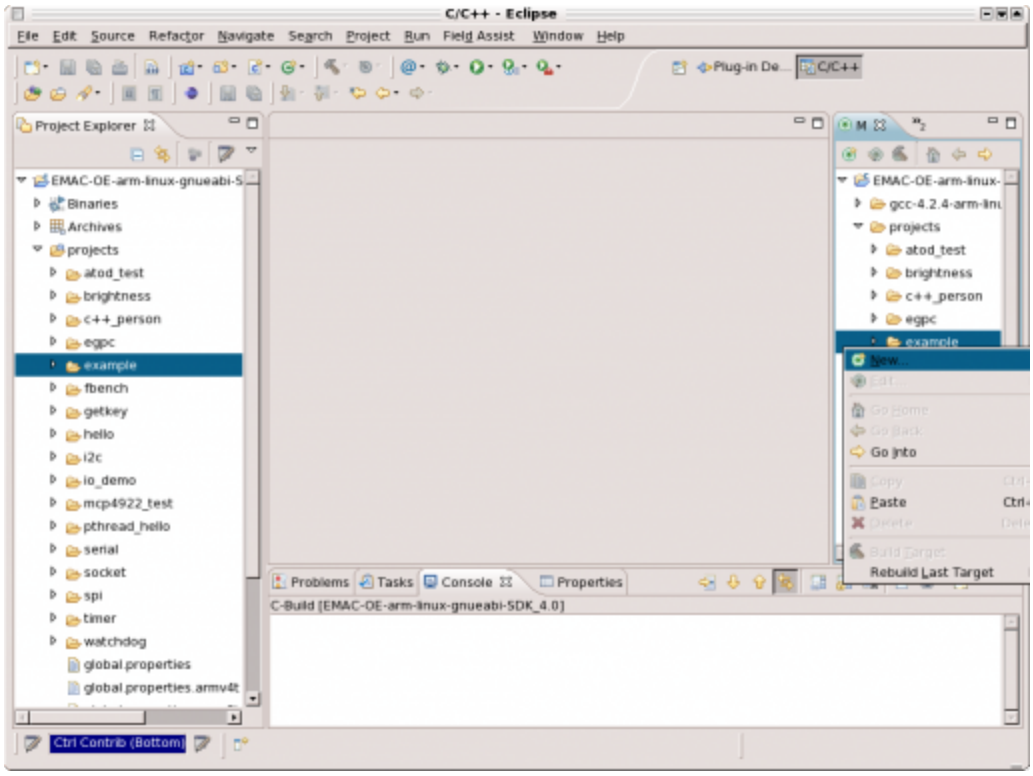

Figure 4. The Make Target context menu

2. Create the make targets `all`,`clean`, and `upload` as in Figures 5, 6, and 7 below.

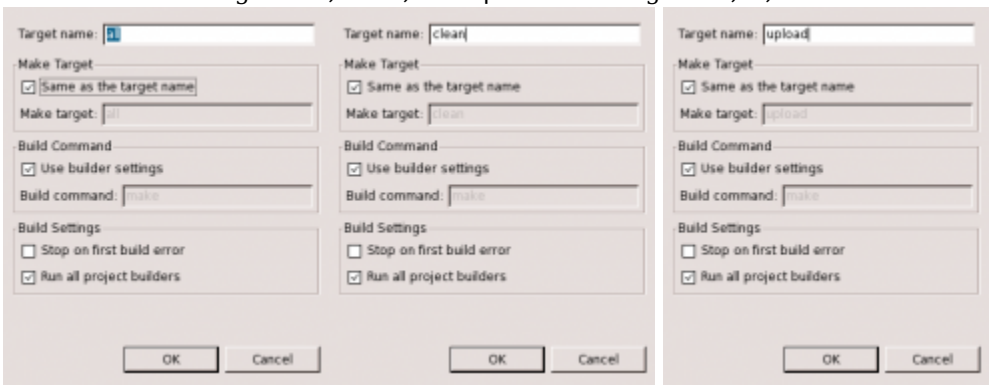

Figure 5, 6 and 7. Make `all`, `clean`, and `upload`

Modify the Makefile

Once the Make Targets have been created, a `Makefile` needs to be created that contains the same targets so that GNU make knows how to compile and link the source into a binary that will run on the target machine. See the Makefile modification section from the EMAC OE SDK New Project guide for an explanation of what to do. The major difference when using Eclipse will be the availability of the Project Explorer View for exploring the project folder to find the Makefile to be copied.

Additional Makefile Configuration for a New Project Outside of SDK/projects

This procedure only applies to the New Projects created outside of the SDK directory tree. To accommodate the placement of the new project outside of the `/path/to/sdk/EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects` directory, the developer must perform the following steps in order for the `Makefile` to have all the information that GNU make needs when running the EMAC OE SDK toolchain.

1. Add the following line to the `Makefile` in the new project.

```
-include ./global.properties
```

2. Copy the `global.properties` into the new project folder.

```
$ cp -L /path/to/sdk/EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects/global.properties /path/to/w
```

3. Change the value of `SDKBASE` in `global.properties` to reflect the change in location relative to the SDK installation. For example, `SDKBASE=../../` would be changed to `SDKBASE=/path/to/sdk/EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/`.

Using the New EMAC OE SDK Project

Write the C Code

Now it is time to write the `example.c` code. This step is analogous to the C code writing section of the EMAC OE SDK New Project Guide.

1. Open the C/C++ Perspective if it is not open already.
    1. From the Eclipse workbench, select *Window → Open Perspective → Other...*
    2. In the *Open Perspective* dialog window, choose *C/C++*.
2. Create a new C file using the Eclipse New Source Wizard.
    1. From the Eclipse workbench, select *File → New → Source File*.
    2. Change the *Source folder:* field to `/path/to/sdk/EMAC-OE-arm-linux-gnueabi-SDK_XX.YY/projects/example`
    3. Set the *Source file:* field to `example.c`
    4. Set the value of *Template:* to `<None>` to create a file with no text to start out.
    5. Click *Finish*.
3. Open the new C file using the Eclipse Editor.
    1. In the Eclipse Project Explorer View, expand the *EMAC-OE-arm-linux-gnueabi-SDK_XX.YY → projects → example* folder.
    2. Double-click the *example.c* file. This will open the new C file in the Eclipse Editor. See Figure 8 below for an example of how this should look.

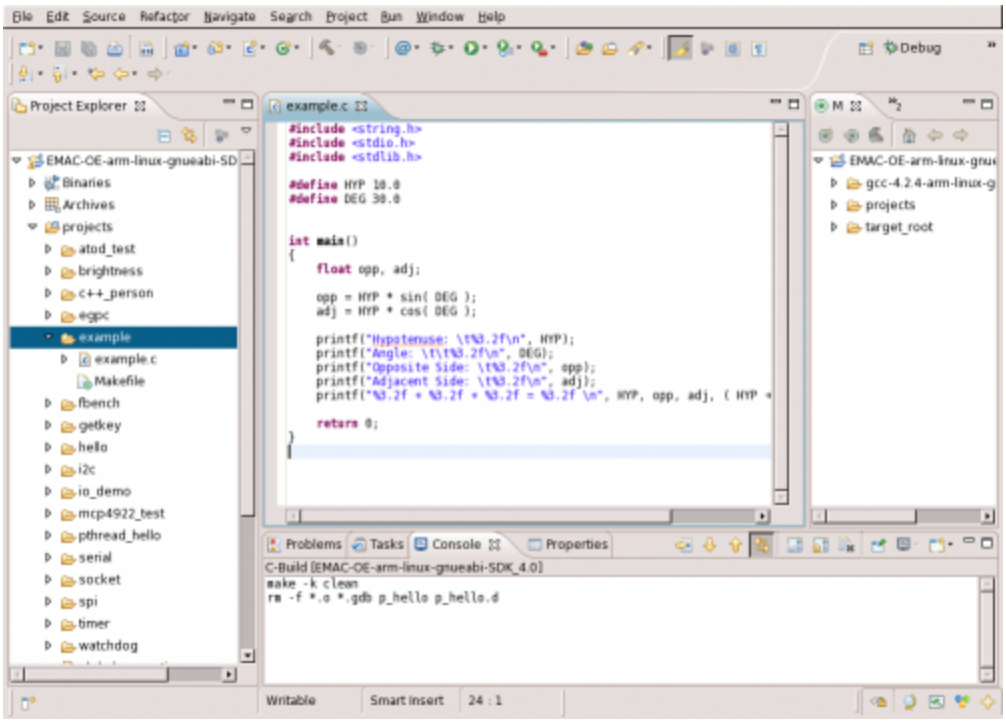

Figure 8. example.c in the Eclipse Editor

4. Copy the code from Listing 1 in the C code writing section of EMAC OE SDK New Project Guide into the open *example.c* file.
5. Save the file with the new code–from the Eclipse workbench, select *File → Save*.

Cross-Compile

Follow the Compile Procedure from the Eclipse Example Project Guide to compile this new project. There will be differences between the directory and folder names used in that guide and those used in this guide.

Next Steps

If a project compiles without trouble but has runtime bugs, you can remote debug the application using `gdbserver`. Read the Remote Debugging with Eclipse Guide to learn more.

See Also

- Eclipse IDE
  - Install
  - Development System Configuration
  - First Time Using Eclipse
  - Import EMAC OE SDK
  - Eclipse Terminal View
  - Using the EMAC OE SDK Examples Projects
  - Create New EMAC OE SDK Projects
  - Using the EMAC OE SDK Eclipse Plugin
  - Remote System Explorer Configuration
    - RSE Setup
    - RSE SFTP Setup
    - Remote Shell/Terminal Setup
  - Execute Remote Applications
  - Debug Remote Applications

» existing_build » eclipse » install » configure » firststeps » import » terminal » example » linux_start » newproject

---