



# Topic: Managing NDS Projects

## Git and GitLab Repositories

Systèmes Embarqués Microprogrammés

- Introduction to Git
  - Version control systems
  - Why using Git?
  - How to manage git repositories
- Introduction to GitLab
  - How to use it
  - How to create and use a git repository
- Live example

- Git is a version control system
  - It allows us to keep track of changes in our software projects
- Key features
  - History of projects developments
  - Efficient multi-users working environment
  - Traceability



- Changes are efficiently tracked
  - You don't need to rename several files to trace different (updated) versions
- Multiple developers can work in parallel
  - Each one on his local environment
  - Users see which files have been modified by other colleagues
  - Only desired files from different users can be merged together
- Repository on cloud
  - Local versions for current updates are stored locally
  - The main repository is on the cloud
  - No risk of loosing your work

**Mandatory use for final NDS projects in EE-310 course!**

- Install git - Native Ubuntu OS
  - Install with:
    - **sudo apt-get update**
    - **sudo apt-get install git**
  - Reference – other distributions  
[Git \(git-scm.com\)](http://git-scm.com)
- Check installation success with
  - **git –version**
  - (should return the number of the installed git version)

**Note: the Ubuntu version of the virtual machine is obsolete! These commands will not work there**

- Install git - Mac OS
  - Already installed if you have Xcode package
  - Install with:
    - **brew install git**
  - Reference
    - [Git - Downloading Package \(git-scm.com\)](#)
- Check installation success with
  - **git --version**
  - (should return the number of the installed git version)

- Install git - Windows PowerShell
  - Power Shell environment for Git: ***posh-git*** module
  - Check all required packages are already installed
  - Install reference commands  
[Git - Git in PowerShell \(git-scm.com\)](#)
- Download installer  
[Git - Downloading Package \(git-scm.com\)](#)
- Check installation success with
  - **git –version**
  - (should return the number of the installed git version)

- Main git commands

- **git clone**
- **git status**
- **git fetch**
- **git pull**
- **git push**
- **git add /my\_path/my\_file\_to\_add**
- **git remove /my\_path/my\_file\_to\_add**
- **git commit –m “short description of commit goal”**

- **git clone**

- Clone (i.e., download) a repository in your local machine

- **git status**

- Check the status of current git environment
    - Are you on the master branch? (we'll not consider branches here)
    - Is your local version different with respect to the one on the cloud?

- **git fetch**

- Get latest information about the repository

- **git pull**

- Changes in the repository on the cloud are applied in your local version

- **git push**

- Changes in your local version are applied in the repository on the cloud

- **git add /my\_path/my\_file\_to\_add**
  - Add specific files to be updated in your next commit
- **git remove /my\_path/my\_file\_to\_add**
  - Remove specific files to be updated in your next commit
- **git commit –m “short description of commit goal”**
  - Commit your current version
  - It actually includes the added/removed files in your new version

Complete git reference

[\*\*Git - Reference \(git-scm.com\)\*\*](http://git-scm.com)

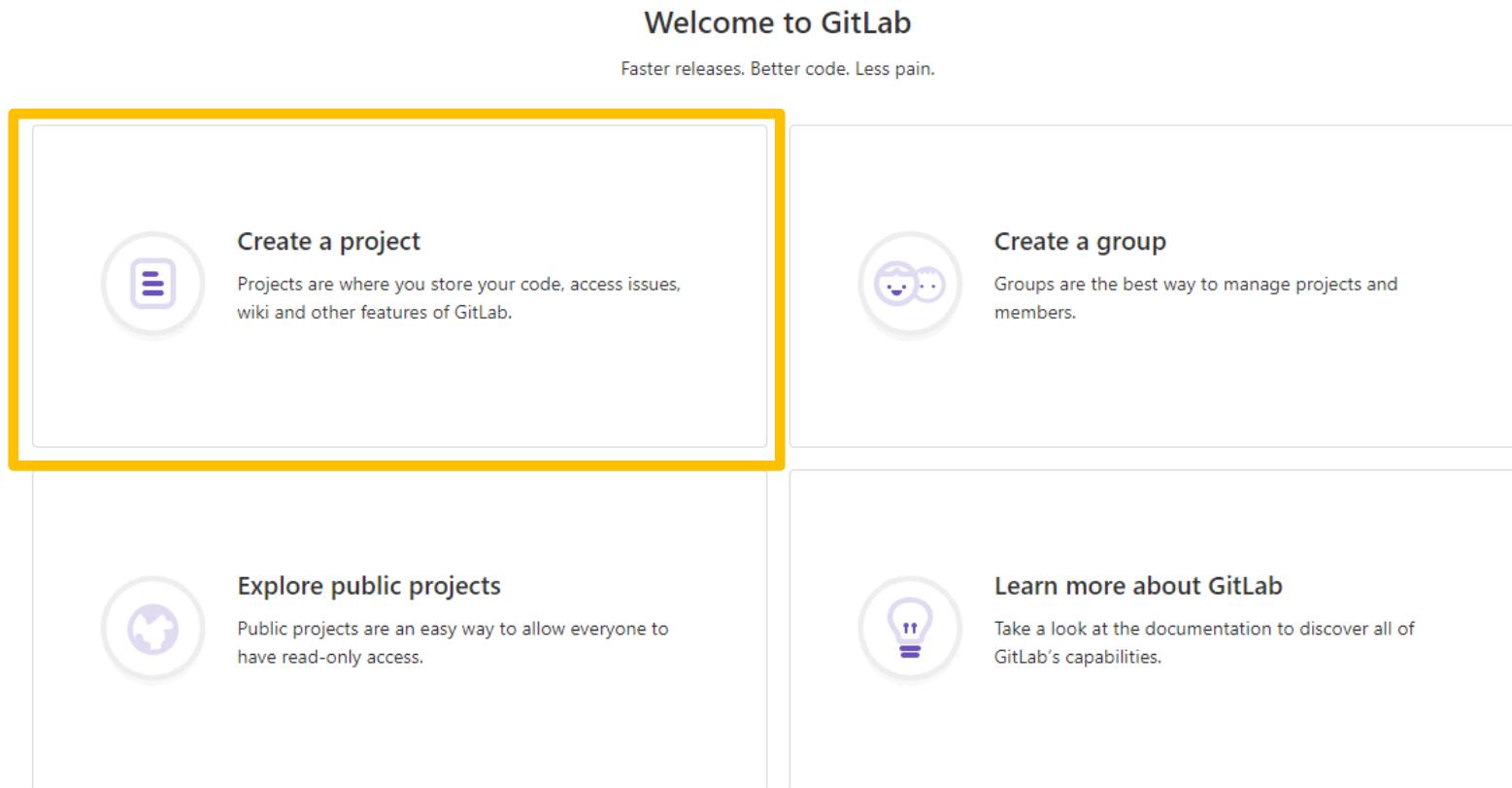
Brief reference page of key commands  
on Moodle page

- What is GitLab?
  - EPFL platform for code-sharing
  - The One DevOps Platform for software innovation
  - Started in 2011, open source project
- Link to complete GitLab description
  - [Why GitLab?](#)
  - It uses version control
  - Safe cloud storage of projects for academia and public entities in Switzerland



- Login with EPFL credentials
- New project/repository
  1. Create blank project
  2. Select a project name (mandatory) and a project description (optional)
  3. Set visibility to 'Private'
  4. Click on 'Create project'
- Create a SSH public/private key pairs on your PC
- Upload the public key on GitLab
- Clone the repository on your PC

- Login into GitLab
- Click on “Create a project”



Welcome to GitLab

Faster releases. Better code. Less pain.

**Create a project**

Projects are where you store your code, access issues, wiki and other features of GitLab.

**Create a group**

Groups are the best way to manage projects and members.

**Explore public projects**

Public projects are an easy way to allow everyone to have read-only access.

**Learn more about GitLab**

Take a look at the documentation to discover all of GitLab's capabilities.

A yellow box highlights the 'Create a project' section on the left.

- Insert the name of your project
- Include your username in “Project URL”
- Keep the project private

New project > Create blank project

---

Project name

Project URL

Project slug

Want to organize several dependent projects under the same namespace? [Create a group](#).

Visibility Level [?](#)

 Private  
Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the group.

 Internal  
The project can be accessed by any logged in user except external users.

 Public  
The project can be accessed without any authentication.

Project Configuration

Initialize repository with a README  
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Enable Static Application Security Testing (SAST)  
Analyze your source code for known security vulnerabilities. [Learn more](#).

[Create project](#) [Cancel](#)

# Example: Template Project – pt. 3

- A warning message reminds you to upload the “SSH key”
- Need of SSH public/private key pairs to clone locally

The screenshot shows a GitHub repository page for 'ExampleNDSProject'. A yellow box highlights a warning message: 'You can't push or pull repositories using SSH until you add an SSH key to your profile.' with buttons 'Add SSH key' and 'Don't show again'. The repository details include a project ID of 13796, 1 commit, 1 branch, 0 tags, and 51 KB of project storage. The commit history shows an 'Initial commit' by Flavio Ponzina. There are buttons for 'Find file', 'Web IDE', 'Clone', and a download icon. A 'Configure Integrations' button is also present. A table at the bottom shows the file 'README.md' with a 'Last commit' of 'Initial commit' and a 'Last update' of 'just now'.

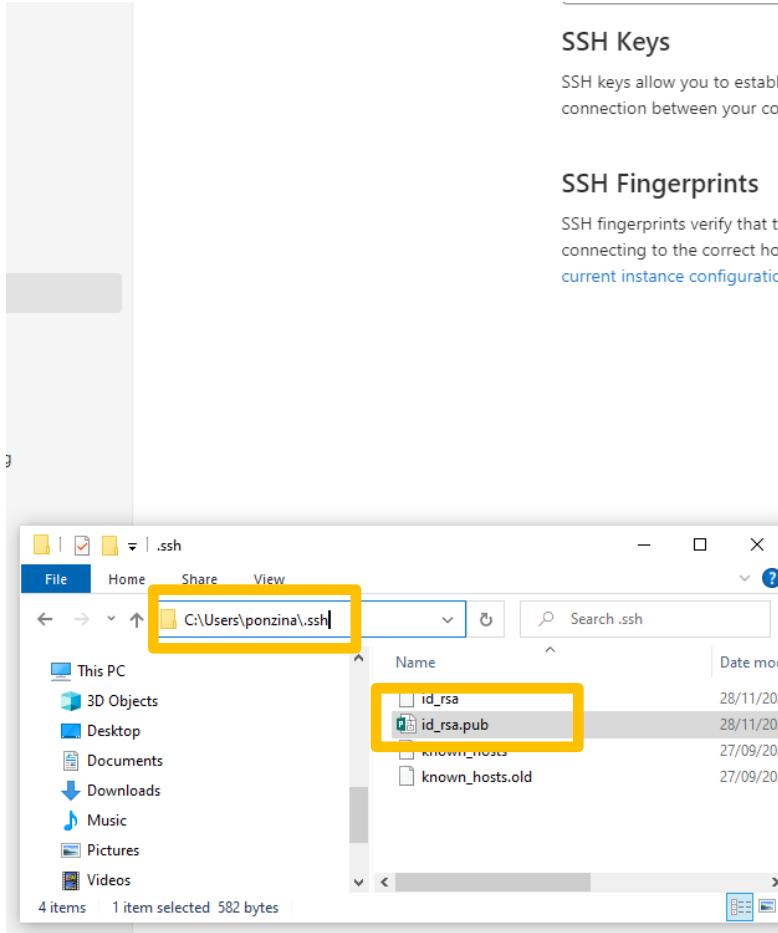
Name	Last commit	Last update
README.md	Initial commit	just now

- *ssh-keygen* command generates private/public key pair
- No need for password here
- Keys stored in two files in *.ssh* folder

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ponzina>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\ponzina/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\ponzina/.ssh/id_rsa.
Your public key has been saved in C:\Users\ponzina/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tsAETPZ7jobe5Em1wtCEYQLsR0Mn/aXJ0xJbcUsqEn4 intranet\ponzina@eslb208104
The key's randomart image is:
+---[RSA 3072]---+
|o.o=B.
|. B+* . +
|. o + * 0 .
|. + E 0 .
|. + X S
| * X o
| . 0 =
| . * o
| .
+---[SHA256]---+
C:\Users\ponzina>
```

- GitLab >> User Settings >> SSH Keys
- Copy the content of the public key (*id\_rsa.pub*) in GitLab



## Add an SSH key

Add an SSH key for secure access to GitLab. [Learn more](#).

### Key

`ssh-rsa`

```
AAAAB3NzaC1yc2EAAAQABAAAQgQDNsD0H8TjzvBfKSAK1tKFKD+VpTqJ9HByXH2rjw36YWJ
U+wSuwZc08fMnOKjfBwngZObMQA/TB4ccL3idkUraG5PeFSGmTdCA+MLZt8LufcoCrWPPrvvZIK
mbBXlvEvTu1axlPO09/v2jdM7xVkJLH76HbHnwxdVVjmSew8501xoxMh2zNvpgAugOaNvve9459
3xNJaRNy/GydKs+SQKq+2vmpNBE2eKYIWkwyrcDDDW1M5fEc0OWC3HRmSJ/5lt3M2IN61dvG
deWjF098EPpjgrjLqZog6f5YtLzmwmdCOZBxHlIqExAR7AzwuQtxPjSA8V7YIBKQEpVPFy1kZAZPHY2
EpVPjvhwlxdwKTMDkSQtDVStz/X+/yRjd9iPb3LZTkwd7NcaSjooEnDwzRi4lEgfsBZeBjW6d2uDRNG
WeuiF98Gaizkxkhah5bg4tpzMRC0tSx8b6KRgLGuuX/j7nR1e3cc9w3PUCJi4JdhmcuzSJB5xwQs
3rM= intranet\ponzina@eslb208104
```

begins with `ssh-rsa`, `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp521`, `ecdsa-sha2-nistp511`, `ssh-ed25519`, 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

### Title

Key titles are publicly visible.

### Expiration date

Optional but recommended. If set, key becomes invalid on the specified date.

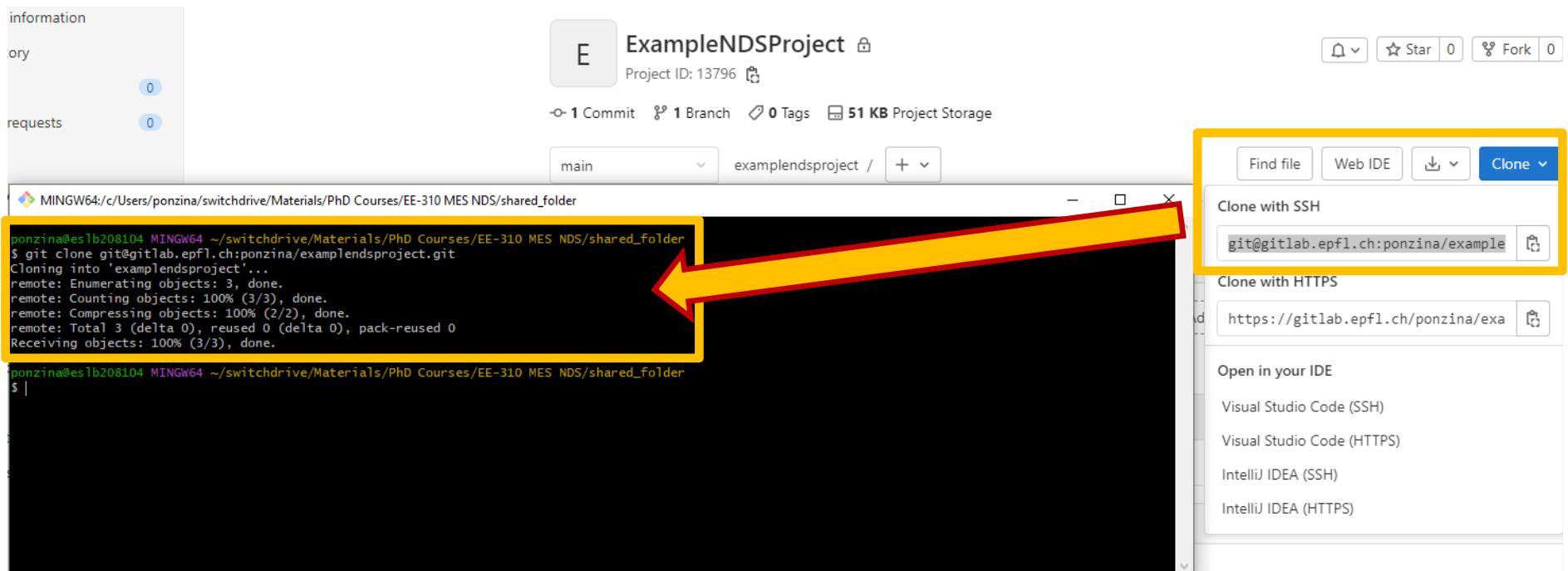
[Add key](#)

### Your SSH keys (0)

There are no SSH keys with access to your account.

## Example: Template Project – pt. 6

- “Clone” button to copy the SSH cloning path
- In your shared folder, open git and run *git clone <SSH path>*
  - This way, the repository will be visible in the virtual machine!



- Enter inside the directory of your cloned repository
- *git status*

```
ponzina@eslb208104 MINGW64 ~/switchdrive/Materials/PhD Courses/EE-310 MES NDS/shared_folder/exemplendsproject (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

- Copy here the “Template” project from Moodle
- *git status*

```
ponzina@eslb208104 MINGW64 ~/switchdrive/Materials/PhD Courses/EE-310 MES NDS/shared_folder/exemplendsproject (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Template/

nothing added to commit but untracked files present (use "git add" to track)
```

- Template project is now in the (local) repository

- *git add* to add the folder to current branch (main)

```
ponzina@es1b208104 MINGW64 ~/switchdrive/Materials/PhD Courses/EE-310 MES NDS/shared_folder/exmplendsproject (main)
$ git add Template/
warning: in the working copy of 'Template/.cdtproject', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Template/.cproject', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Template/.project', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Template/.settings/org.eclipse.cdt.coreprefs', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Template/Makefile', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Template/source/main.c', LF will be replaced by CRLF the next time Git touches it
```

- *git commit* to save changes

```
ponzina@es1b208104 MINGW64 ~/switchdrive/Materials/PhD Courses/EE-310 MES NDS/shared_folder/exmplendsproject (main)
$ git commit -m "creating the template project on git"
[main 22c92c6] creating the template project on git
  Committer: Ponzina Flavio <flavio.ponzina@epfl.ch>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

  git config --global --edit

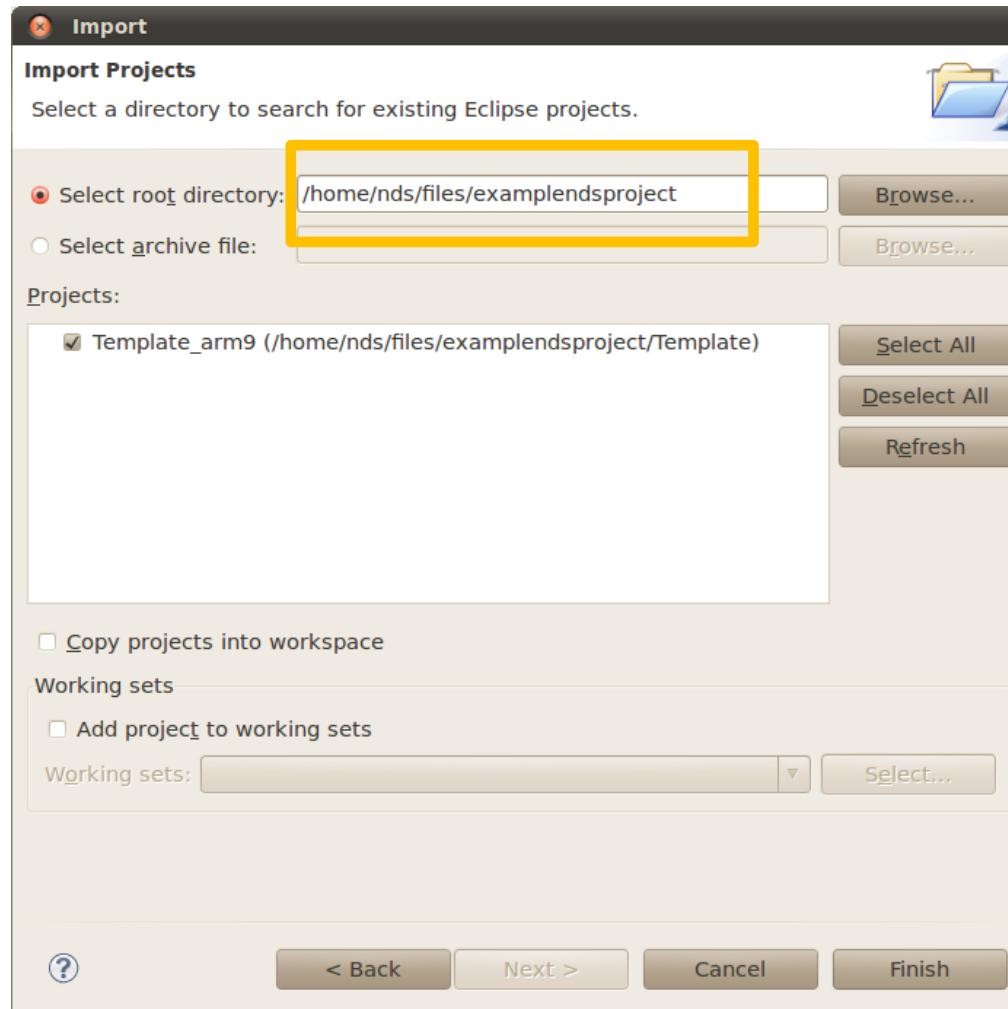
After doing this, you may fix the identity used for this commit with:

  git commit --amend --reset-author

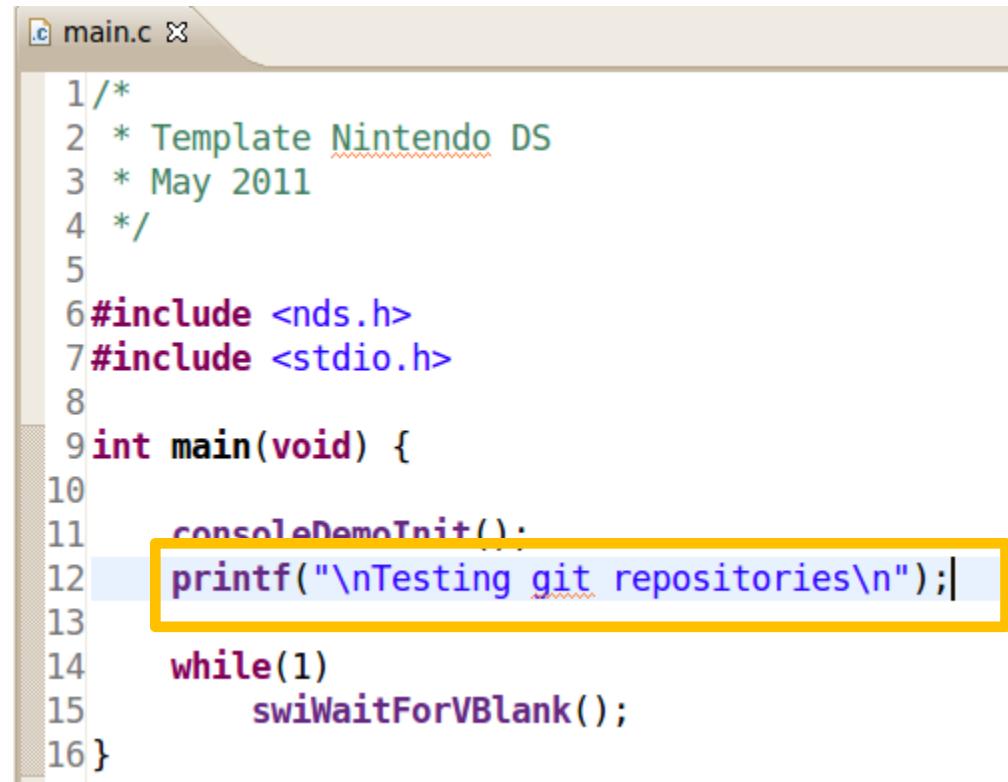
6 files changed, 1048 insertions(+)
create mode 100644 Template/.cdtproject
create mode 100644 Template/.cproject
create mode 100644 Template/.project
create mode 100644 Template/.settings/org.eclipse.cdt.coreprefs
create mode 100644 Template/Makefile
create mode 100644 Template/source/main.c
```

# Example: Template Project – pt. 9

- In the virtual machine, import the project in Eclipse



- Make some changes to the main.c file and recompile



```
1 /*
2  * Template Nintendo DS
3  * May 2011
4  */
5
6 #include <nds.h>
7 #include <stdio.h>
8
9 int main(void) {
10
11     consoleDemoInit();
12     printf("\nTesting git repositories\n");
13
14     while(1)
15         swiWaitForVBlank();
16 }
```

- *git status* to check current state of the repository

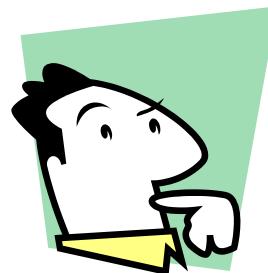
```
ponzina@es1b208104 MINGW64 ~/switchdrive/Materials/PhD Courses/EE-310 MES NDS/shared_folder/exmplendsproject (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   Template/source/main.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Template/Template.arm9
    Template/Template.elf
    Template/Template.nds
    Template/build/
no changes added to commit (use "git add" and/or "git commit -a")
```

- *git add* + *git commit* + *git push* to save changes and update the repository on GitLab
- Note: it is a good practice to add/upload only source files

# Questions?



**Let's use GitLab to develop  
NDS applications!**