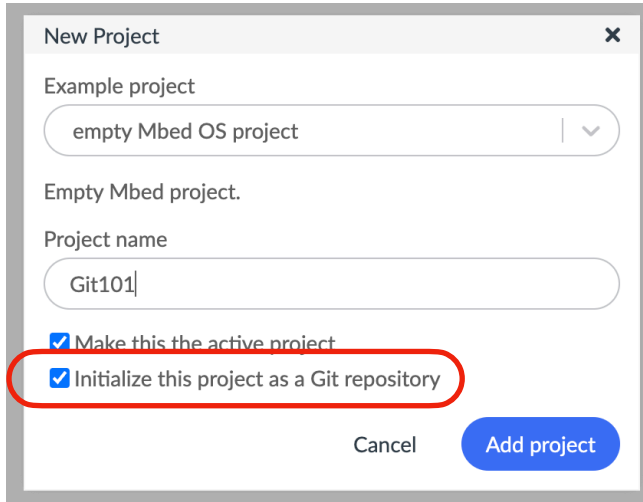# Git fundamentals with Keil Studio
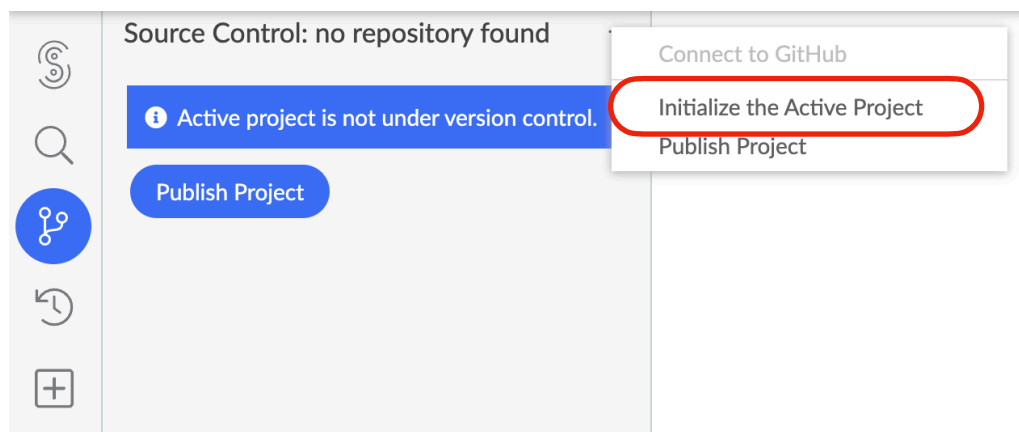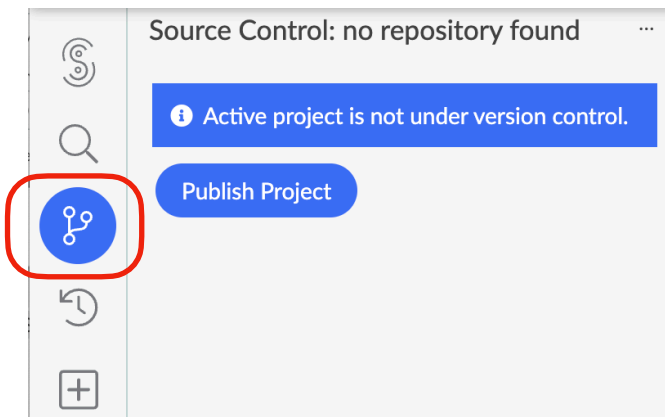
Setup

You can setup a project for source code control when you create it:

Or initalise an existing project:

Click the Source Control button and then click the tiny … menu top right to see the option
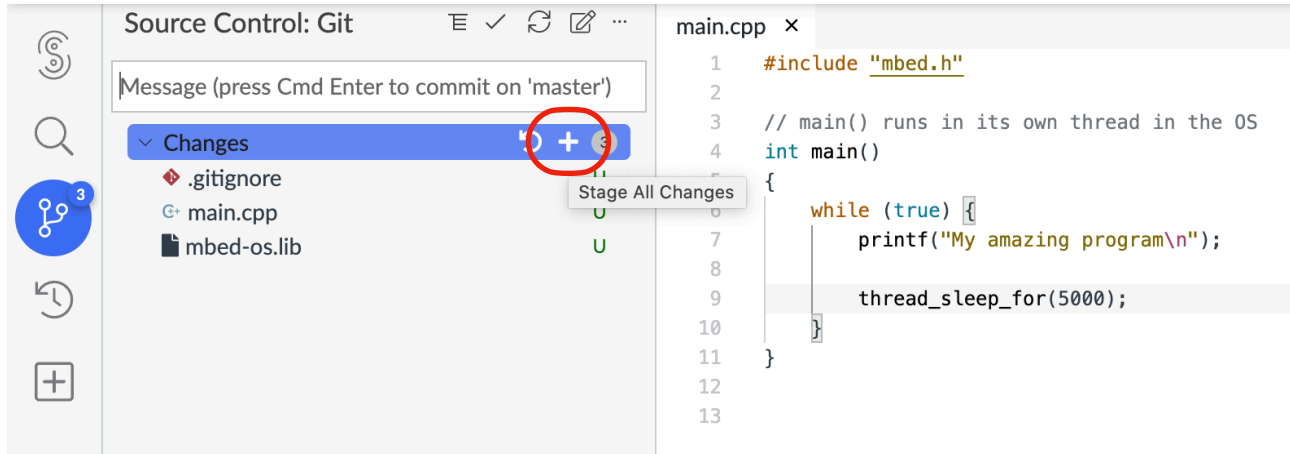
Your First Commit

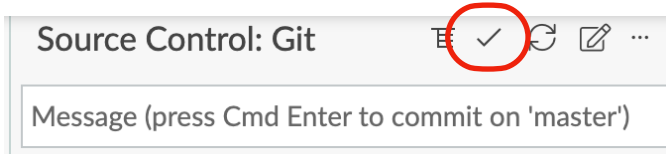This is putting your new files in to a virtual envelope for safe keeping.

Click the Source Control button.

Point at the Changes line and click the + button to add the files to the repository.
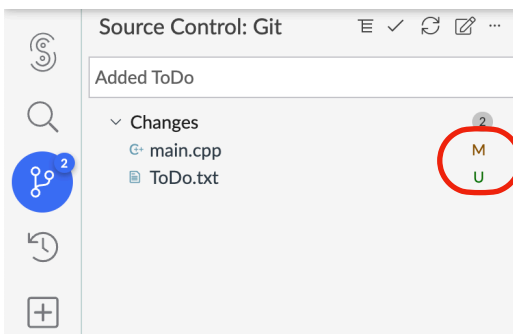

The file status will change from U, untracked, to A, added.

You then enter a message and click the tick to commit your files to a safe place.

Subsequent Commits

Make changes & add files and when you are ready to save a copy of your files go in to Source Control:

U means the file is untracked - a new file not yet added to the repository. M means the file is modified - a file that is already in the repository that you have made changes to since your last commit.

You can use the + button on the Changes line to update all files, or do them on a file by file basis.

Then add a message and click the tick to Commit. Normally you commit your work when you are happy it is in a good place - you've fixed an issue or added new functionality - and just possibly, tested it too!

Review the history

Click the History button:

Click the commit you want to look at in the left hand list & click the little eye icon on the right. You can then click a file name in the list to see the changes.

The icons in the far top right will allow you to open the files in the editor for you to copy & paste as you require.

Immediate step back in time

If you want to throw away the changes you just made, go in to Source Control.

You can either click the circle icon on the Changes line to revert all files, or just for an individual file, as you need.

Personal use only?

These are the core activities of using Git to keep track of the changes you make to a project and allow you to step back in time if you need to - usually when you make changes that don't go to plan.

Publishing your project on GitHub

In Source Control, select the … menu and choose Publish Project
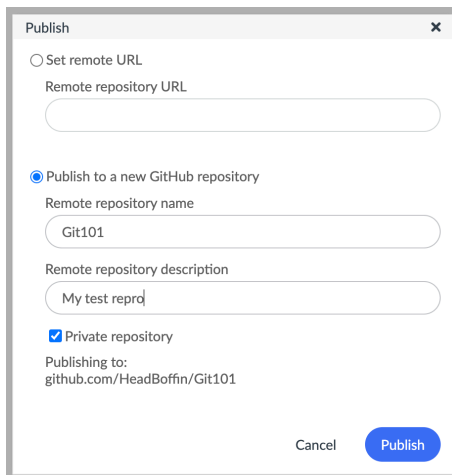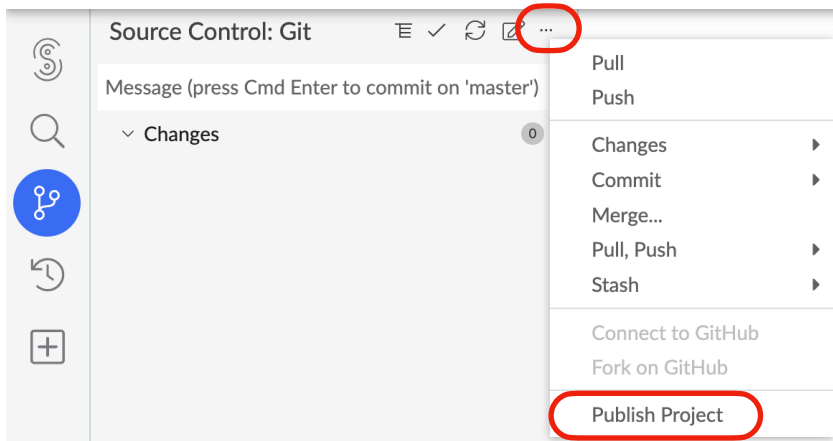
Choose the new option, fill in an optional description and if you want to keep your project private:

When you click publish the repro will be created, but your work will still need to be "pushed" up to GitHub.

Push your commits

You work on your project locally, when you are happy to share your efforts, you then push it to the main repro.
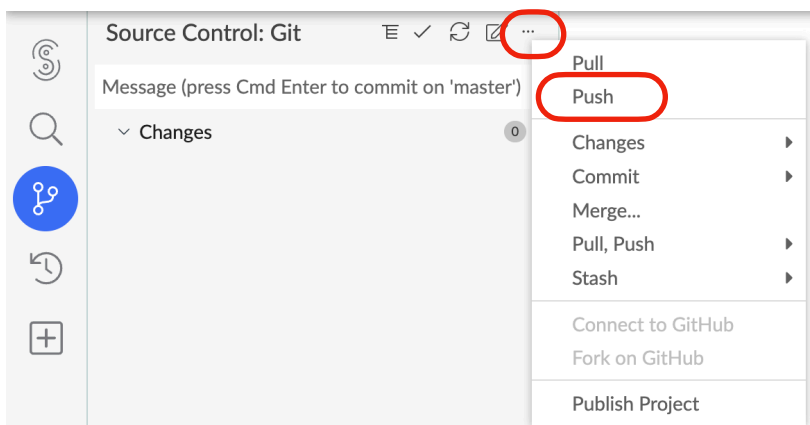
It is normal to split up your work in to small pieces that are standalone or a set of pieces that make up a larger piece of functionality. Each one would be a commit. So when you are have a coding session, you may create a few commits. These are then transferred in one go when you push your commits.