MDK-6 Workflow with VS-Code

By Trevor Martin

Introduction

The MDK-6 extensions to VS-Code provide a fully featured IDE that allows a developer to write, build and debug code for Cortex-M based microcontrollers. In Addition, there are many features in the VS-Code IDE that enable more complex workflows and strong integration with local and cloud-based version control systems. This Tech tip will provide an overview of these features as a guide to your future project development.

Continuous Integration

Improved workflow really comes into its own with the support for Version Control Systems (VCS). While most VCS can work with VS-Code it is especially easy to use with Git and the cloud repository GitHub.

Once you have an account on GitHub (free or paid) you can enter your credentials into VS-Code through the 'accounts' icon at the bottom of the sidebar.

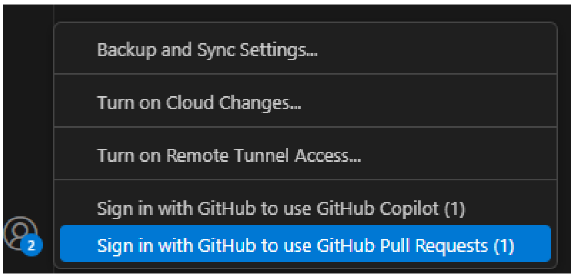

Figure 1. Sign in the external services such as GitHub through the accounts Icon

Now when a new project is created there is an option to automatically initialise a GitHub repository for your project.

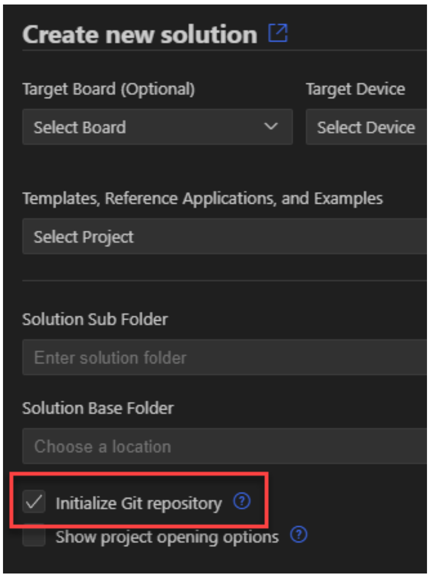

Figure 2. Sign in the external services such as GitHub through the accounts Icon

Now as you develop code it is possible to perform operations such as clone, commit and pull request directly from VS-Code using the source code control on the sidebar.

While you can use GitHub as a simple repository it is also possible to install additional extensions that allows you to develop and run GitHub Actions scripts and also create shared cloud based 'Codespaces' which are shared online development environments.

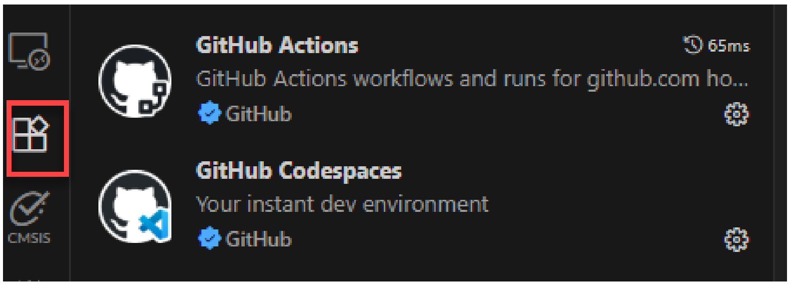

Figure 3. VS-Code extensions for GitHub Actions and GitHub CodeSpaces

GitHub Actions are stored within the repository in a github/workflows directory.

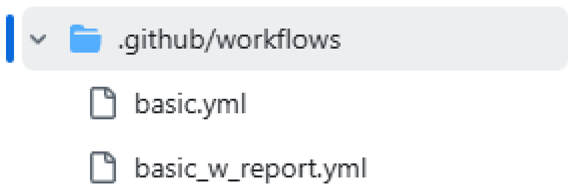

Figure 4. The GitHub Actions scripts are stored in the GitHub/Workflows directory

The GitHub actions scripts use the 'Yet Another Markup Language' (YAML) scripting language and are triggered by user actions defined within the script. While there are a large number of standard scripts available Arm provide a set of custom scripts for use with a Cortex-M project. One key use of the GitHub actions scripts is to perform a set of continuous integration tests when new code is committed, an example template project is available here.

https://github.com/Arm-Examples/AVH_CI_Template

To try this template out, first fork the repository to your own account, next clone a local copy. The code is structured with a set of ASSERT tests using the unity testing framework. The pass-fail results are written to a file.

Open the project and make a small change. After, a successful build a commit to the repository will cause the script to run. The new image is then rebuilt using the CLI of CMSIS-Toolbox and then run on a matching Cortex-M core using the Arm Virtual Hardware (AVH) Simulators.

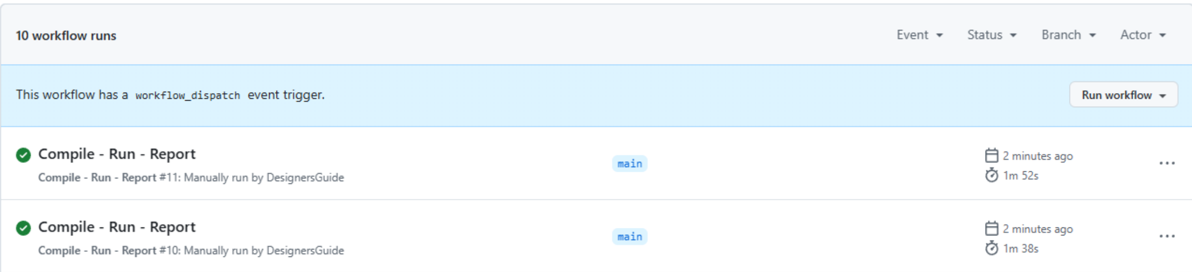

Figure 5. The Actions menu item in your repository reports the status of your

The test results are examined by the GitHub Actions script. If any test fails, the script will report this is the GitHub Actions dialogue and also send a notification email to the developer who started the GitHub Action.

Online collaboration

GitHub also provides a virtual development environment called Codespaces. For a given repository you can create a Codespace from within the "<> Code" button on your repository.

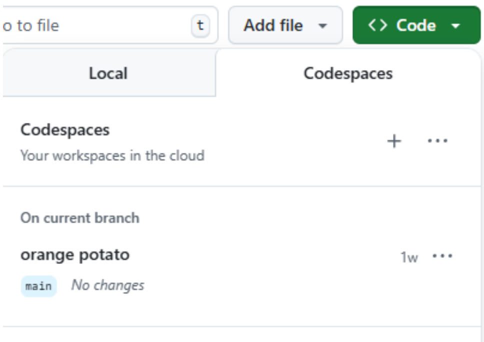

Figure 6. Codespace instances are created through the <> Code button

A Codespace is a container running within a Linux image that can be configured with your development tools. Once created the container has a browser-based version of VS-Code which when launched will install the MDK-6 extensions. This recreates your development environment in the cloud based Codespace. You can also connect to the Codespace through a remote connection from desktop VS-Code installation.

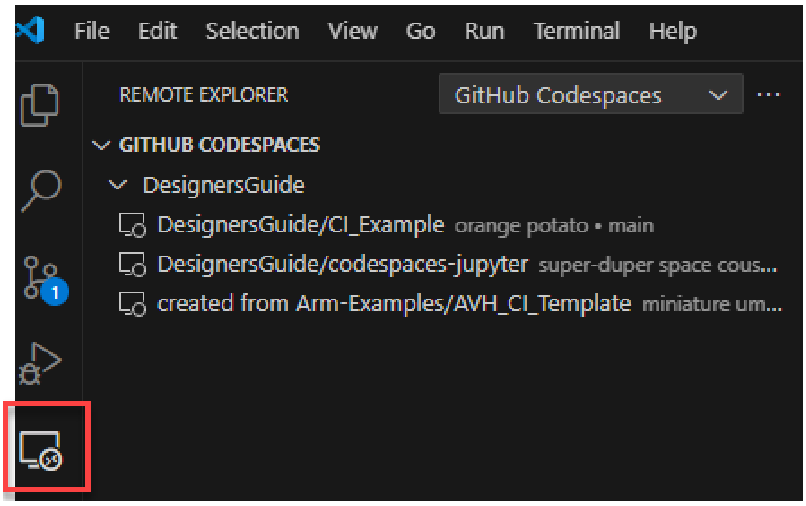

Figure 7. VS-Code can connect and work with a Codespaces remotely

This allows multiple developers to work on the same codebase using remote connections. This approach is very useful for collaboration when working as part of a team of developers. Particular use cases are code review, pair programming and integrating several developers code. Also if you have a large codebase that takes a long time to build and test you can scale the virtual machine to speed up the whole process.

Once you have created a Codespace the container may be exported as a docker container that can be deployed on a local machine. This can be very useful to setup a new workspace with a known configuration either manually or through a configuration tool such as Ansible.

In Summary

With the introduction of MDK-6 VS-Code is much more than a replacement editor for Microvision. It brings in a new era of modern code development for MCU based projects by supporting modern workflows and online collaboration.

Further Information

For more information visit our website: www.hitex.co.uk or get in touch: info@hitex.co.uk. You can also connect with us: LinkedIn

Sign up to receive Tech Tips directly to your inbox. Never miss a trick!