CMSIS Hardware Abstraction Layer

Visual Studio Code

CubeMX Template Generation and Setup

Visual Studio Code for Embedded Programming

How to set up your IDE

SIEMENS

VSCode

INSALL VISUAL STUDIO CODE

Download Visual Studio Code - Code Editing. Redefined and install.

EXTENSIONS

Go to **View -> Extensions**

Install

1. C/C++
2. C/C++ Extension Pack
3. Cortex-Debug
4. STM-Helper

ST-LINK

Driver Installation

1. Install the driver **STM32 ST-Link Utility v4.1.0 setup.exe**
2. During the process you have to press confirm by "**Installieren**" that you want to install the driver
3. At the end of the installation there are two windows offered: Press the "**Weiter**" in the window "Welcome to the Device Driver Installation Wizard!"
4. Then press "**Fertig stellen**" in the window "Completing the Device Driver Installation Wizard"

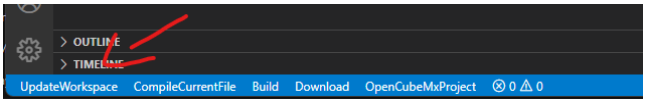
ARM-GNU-TOOLCHAIN

Save the Toolchain permanently in a folder and remember the location.

PROJECT SETUP

Open VSCode

1. Extract and open the Template Folder.
2. On the lower left corner in Visual Studio Code press **Update Workspace**



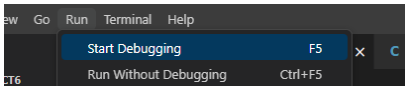
➔ UpdateWorkspace resets the **armToolchainPath**

3. Then open launch.**json**

4. Replace the **armToolchainPath** with the bin folder of your downloaded ToolChain

HOW TO USE

Run Program with Debugger:



Upper left: **Run -> Start Debugging** or press **F5**

STM32 CubeMX

CubeMX is a tool to generate templates enabling specific modules and configuration settings for modules. It also includes the HAL (Hardware Abstraction Layer) and LL (Low Level) drivers for the board.

GETTING STARTED

After installing the CubeMX program, click **Access to MCU selector**, in the middle of the screen.

A new window will open. In the upper left you will find a field to enter **the Commercial Part Number**. For our purposes the part number is **STM32F107VCT6**.

After searching for the right processor, near the middle select the right Version; **STM32F107VCT6**. And click in the upper right on **Start Project.**

PINOUT AND CONFIGURATION

On the pinout diagram you can select individual pins and enable them; setting them to be an input and or output, not default input configuration does not set it to be with a pull-down resistor. To set this go to the left panel and select **System Core -> GPIO**, there you can set the setting for every configured ping.

In the left panel you can also enable individual timers and other modules with their configuration.

CLOCK CONFIGURATION

In the upper tab select **Clock Configuration**, Cemex will try to auto solve all dependencies but you can also easily manually set clock speeds.

PROJECT MANAGER AND EXPORT

In order to Export Enter a **Project Name** and **Project Location.** Set the **Application Structure** to **Basic.** And the **Toolchain/IDE** to **Makefile.**

Go to the **Code Generator** tab on the left and enable **Generate peripheral initialization as pair of '.c/.h' files per peripheral**.

Under advanced settings you can select the drivers either HAL (Hardware Abstraction Layer) or LL (Low Level) for each enabled Module.

To export press, **Generate Code,** and that's it. Now you can open the folder in the export directory, with the Makefile, in VSCode just like you could open the template. Follow the instructions at the top of the guide for this.