



DesignNews

C++ Primer for Embedded Systems

Day 2:

Tooling Up for Embedded C++ : VS Code

Sponsored by

DigiKey



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

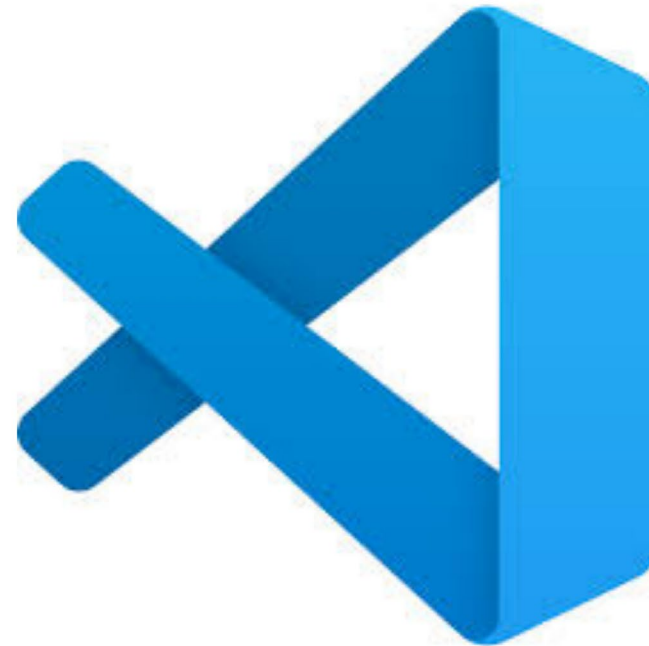


Fred Eady

Visit 'Lecturer Profile' in your console for more details.

AGENDA

- **Install Visual Studio Code**
- **Install the STM32 VS Code Extension**
- **Install the STM32 VS Code Prerequisites**
- **Create an STM32 C++ Project Using VS Code**



Download and Install VS Code

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.

<https://code.visualstudio.com/download#>



↓ Windows

Windows 10, 11

User Installer	x64	Arm64
System Installer	x64	Arm64
.zip	x64	Arm64
CLI	x64	Arm64



↓ .deb

Debian, Ubuntu

↓ .rpm

Red Hat, Fedora, SUSE

→ .deb	x64	Arm32	Arm64
.rpm	x64	Arm32	Arm64
.tar.gz	x64	Arm32	Arm64
Snap	Snap Store		
CLI	x64	Arm32	Arm64

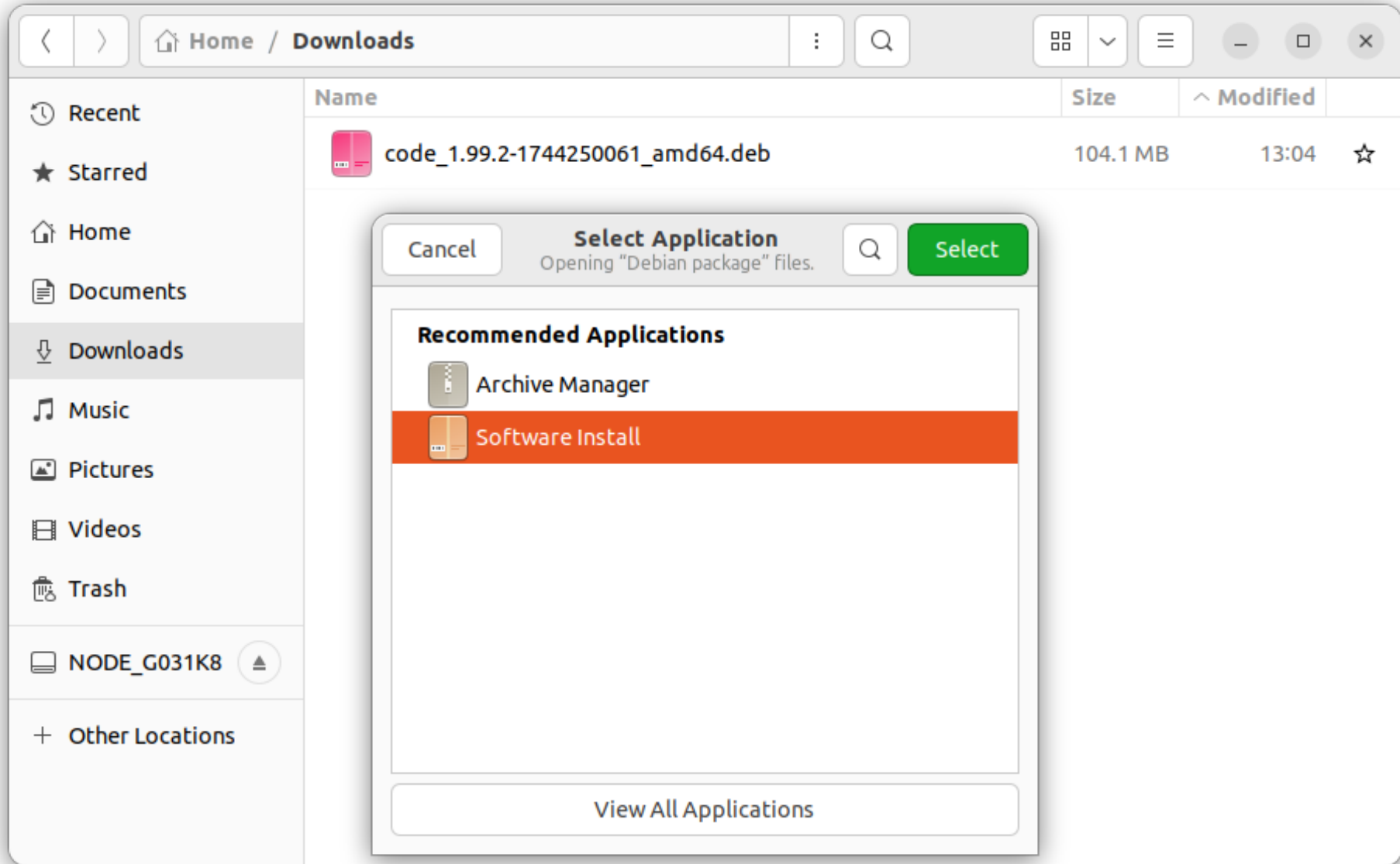


↓ Mac

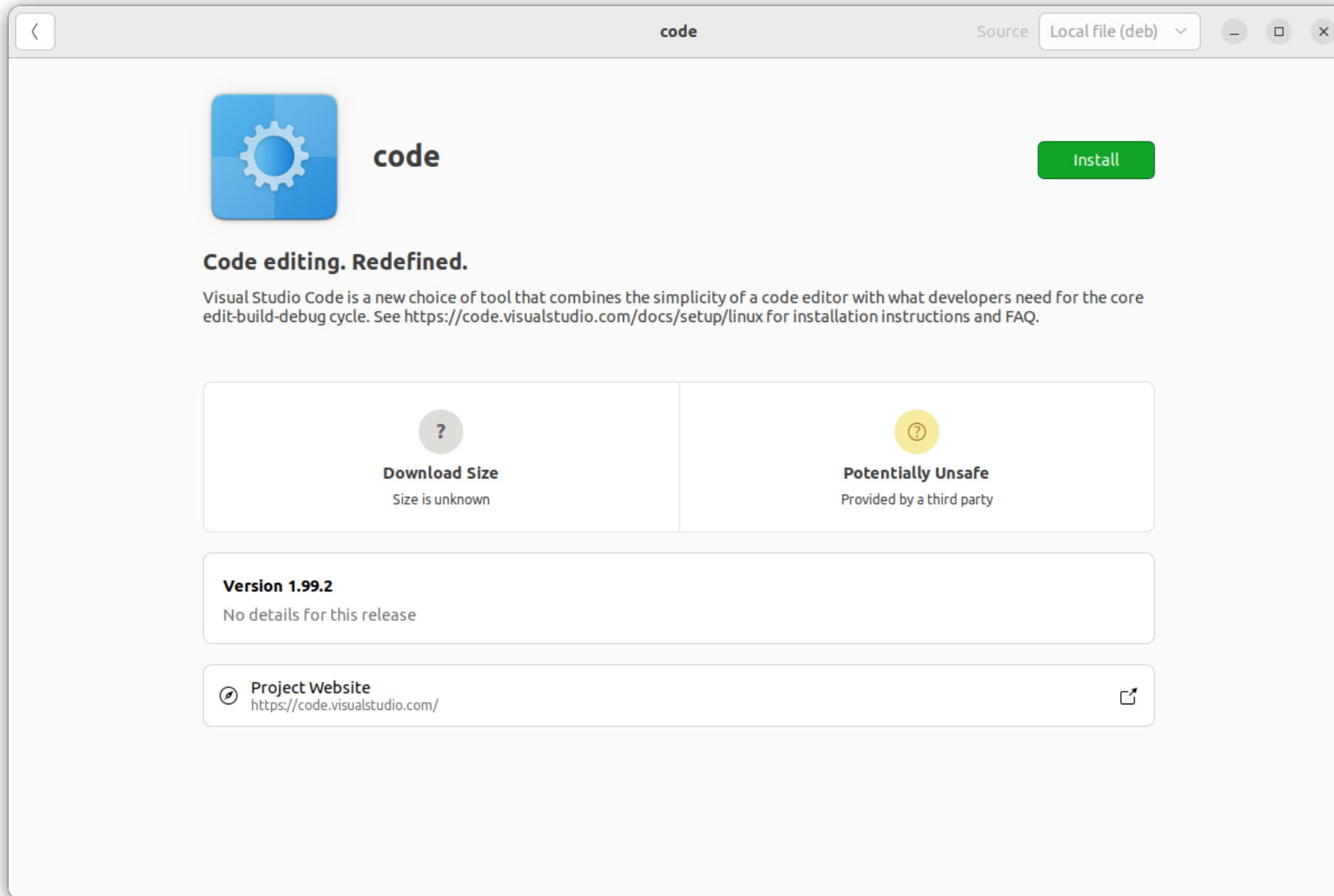
macOS 10.15+

.zip	Intel chip	Apple silicon	Universal
CLI	Intel chip	Apple silicon	

Download and Install VS Code



Download and Install VS Code



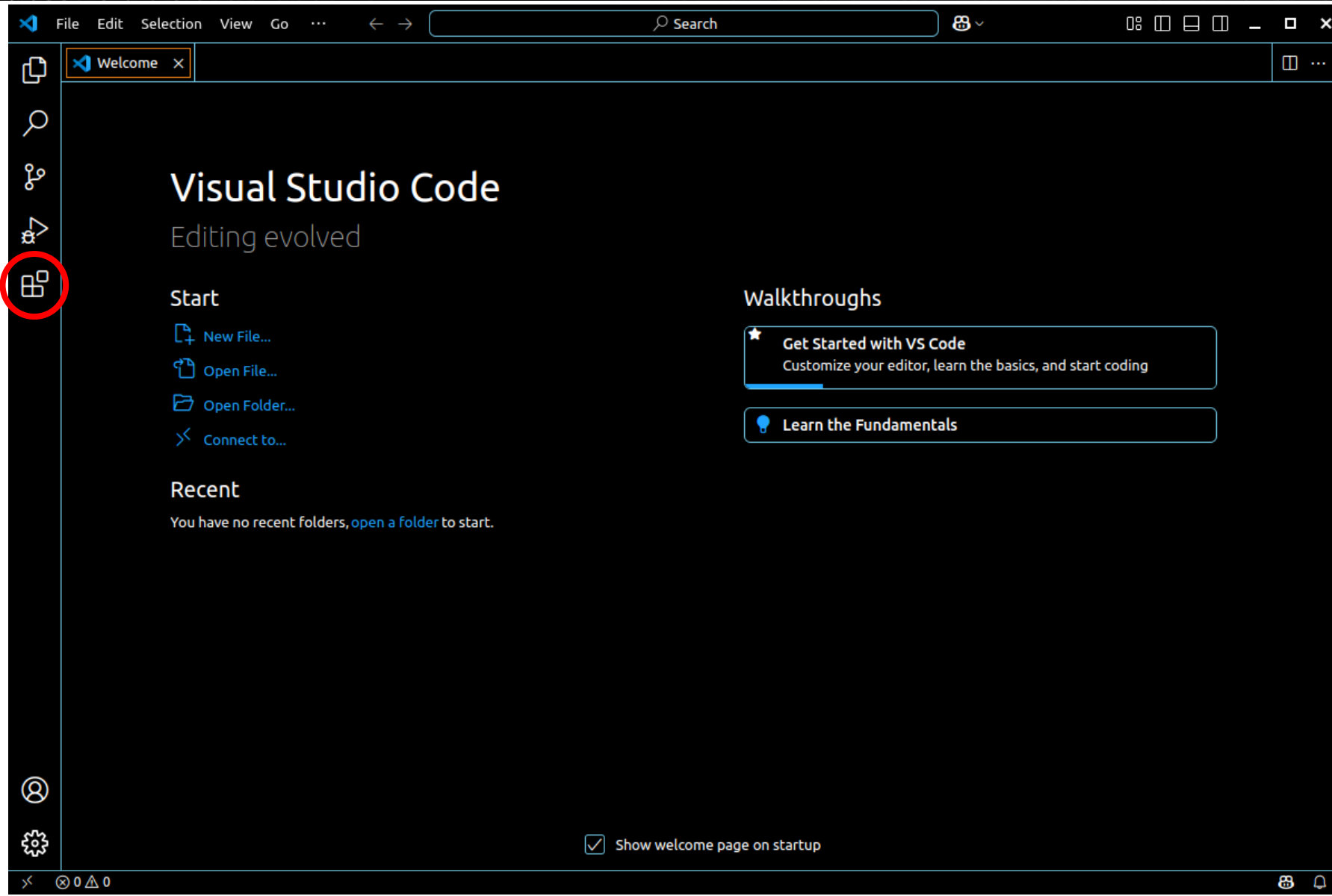
The screenshot shows a window titled "code" with a "Source" dropdown set to "Local file (deb)". The main content area features the VS Code logo (a blue square with a white gear) and the word "code" in a bold, sans-serif font. To the right of the logo is a green "Install" button. Below the logo and name, the text reads "Code editing. Redefined." followed by a paragraph: "Visual Studio Code is a new choice of tool that combines the simplicity of a code editor with what developers need for the core edit-build-debug cycle. See <https://code.visualstudio.com/docs/setup/linux> for installation instructions and FAQ."

Below the text are two informational boxes:

- Download Size**: Size is unknown (indicated by a question mark icon).
- Potentially Unsafe**: Provided by a third party (indicated by a question mark icon).

Further down, there is a section for **Version 1.99.2** with the text "No details for this release". At the bottom, there is a **Project Website** link to <https://code.visualstudio.com/> with an external link icon.

Install the STM32 VS Code Extension



Install the STM32 VS Code Extension

The screenshot shows the Visual Studio Code interface with the Extensions Marketplace open. The left sidebar displays a list of extensions related to STM32. The main panel shows the details for the 'STM32 VS Code Extension' by STMicroelectronics. The extension is currently installed, as indicated by the 'Disable' and 'Uninstall' buttons. The description states that it provides STM32 embedded development support added to Visual Studio Code. The right sidebar shows the Marketplace metadata, including the identifier, version (2.1.1), published date (2023-03-09), and last released date (2024-09-24). The 'Evolution and Breaking Changes' section highlights that version 2.1.0 added support for TrustZone, Dual-core, and Boot flash devices. The 'Prerequisites' section lists the required packages for different device types and configurations.

EXTENSIONS: MARKETPLACE

stm32

- STM32 VS Code Extension**
STM32 embedded development support added to Visual Studio Code.
STMicroelectronics
- stm32-for-vscode**
An extension for: setting up, compiling, uploading and debugging ST...
Bureau Moeliljke Dingen
- STM32 Project Assistant**
STM32工程助手, 用于生成vscode的代码浏览、编辑、编译、下载、调...
jswyll
- stm32-collir-one**
This extension allows users to customize their rockets using the STM3...
Litai Tech
- STM32 Build Analyzer**
STM32 Build Analyzer for VSCode
Aleksel Perevozchikov
- PlatformIO IDE**
Your Gateway to Embedded Software Development Excellence: CMSI...
PlatformIO
- VSCode STM32Cube Configurator**
Extension for generating file c_cpp_properties.json from Makefile ge...
Samuel Travnicek
- Cortex-Debug: Device Support Pack - STM32F1**
This package contains additional support functionality (SVD definitio...
marus25
- Cortex-Debug: Device Support Pack - STM32F4**
This package contains additional support functionality (SVD definitio...
marus25
- Cortex-Debug: Device Support Pack - STM32L4**
This package contains additional support functionality (SVD definitio...
marus25
- Cortex-Debug: Device Support Pack - STM32H7**
This package contains additional support functionality (SVD definitio...
jeandudey
- stm32plugin**
test
allankueng
- Keil Assistant**
An assistant for Keil uVision
cl

STM32 VS Code Extension
STMicroelectronics | st.com | 90,328 | ★★★★★ (29)
STM32 embedded development support added to Visual Studio Code.

Disable Uninstall Auto Update

DETAILS FEATURES CHANGELOG DEPENDENCIES

STM32 VS Code Extension

STM32 embedded development support: added to **Visual Studio Code**. A new view container for the **STM32VS Code extension** is included in the activity bar, which provides a list of features that can be accessed with a single click.

Evolution and Breaking Changes

Version 2.1.0 of the **STM32Cube VS Code extension** added the support of:

- TrustZone devices
- Dual-core devices
- Boot flash devices

Working with **simple / single-core** devices, requires **STM32CubeMX v6.11** and **STM32CubeCLT v1.15**.
Working with **TrustZone, Dual-core, or Boot flash** devices, requires using **STM32CubeMX v6.12** and **STM32CubeCLT v1.16**.

Prerequisites

- STM32CubeCLT v1.15.0 or later**: **STM32CubeCLT** is a package containing toolchain and STM32 device related data required for project creation, build, and debug functionality.

To have all features available, you also need to install the following:

- STM32CubeMX v6.11.0 or later**: **STM32CubeMX** simplifies the configuration of STM32 microcontrollers and generates the corresponding initialization C code. Starting from **v6.11.0**, **STM32CubeMX** can generate VSCode-compatible **CMake** projects, eliminating the need for **.cproject/.project** conversion in CubeIDE.
- ST-MCU-FINDER**: **ST-MCU-FINDER-PC** connects to and explores the full range of **STM32** and **STM8** microcontrollers, processors, dev boards, and examples.
- Libncurses**: **Ncurses** is a tool that helps programmers create text-based user interfaces that work on different types of computers. It is **mandatory** to install this requirement when using Linux.

Marketplace

Identifier	stmicroelectronics.stm32-vscode-extension
Version	2.1.1
Published	2023-03-09, 09:40:58
Last Released	2024-09-24, 10:35:53

Categories

Other

Resources

Marketplace License
STMicroelectronics

Install the STM32 VS Code Extension Prerequisites

Prerequisites

- **STM32CubeCLT v1.15.0 or later**: STM32CubeCLT is a package containing toolchain and STM32 device related data required for project creation, build, and debug functionality.

To have all features available, you also need to install the following:

- **STM32CubeMX v6.11.0 or later**: STM32CubeMX simplifies the configuration of STM32 microcontrollers and generates the corresponding initialization C code.
Starting from v6.11.0, STM32CubeMX can generate VSCode-compatible CMake projects, eliminating the need for .cproject/.project conversion in CubeIDE.
- **ST-MCU-FINDER**: ST-MCU-FINDER-PC connects to and explores the full range of STM32 and STM8 microcontrollers, processors, dev boards, and examples.
- **Libncurses**: Ncurses is a tool that helps programmers create text-based user interfaces that work on different types of computers.
It is **mandatory** to install this requirement when using Linux.



Download the STM32 VS Code Prerequisites

The screenshot shows a file manager window with the following content:

- Navigation pane: Recent, Starred, Home, Documents, Downloads (selected), Music, Pictures, Videos, Trash, NODE_G031K8, Other Locations.
- File list table:

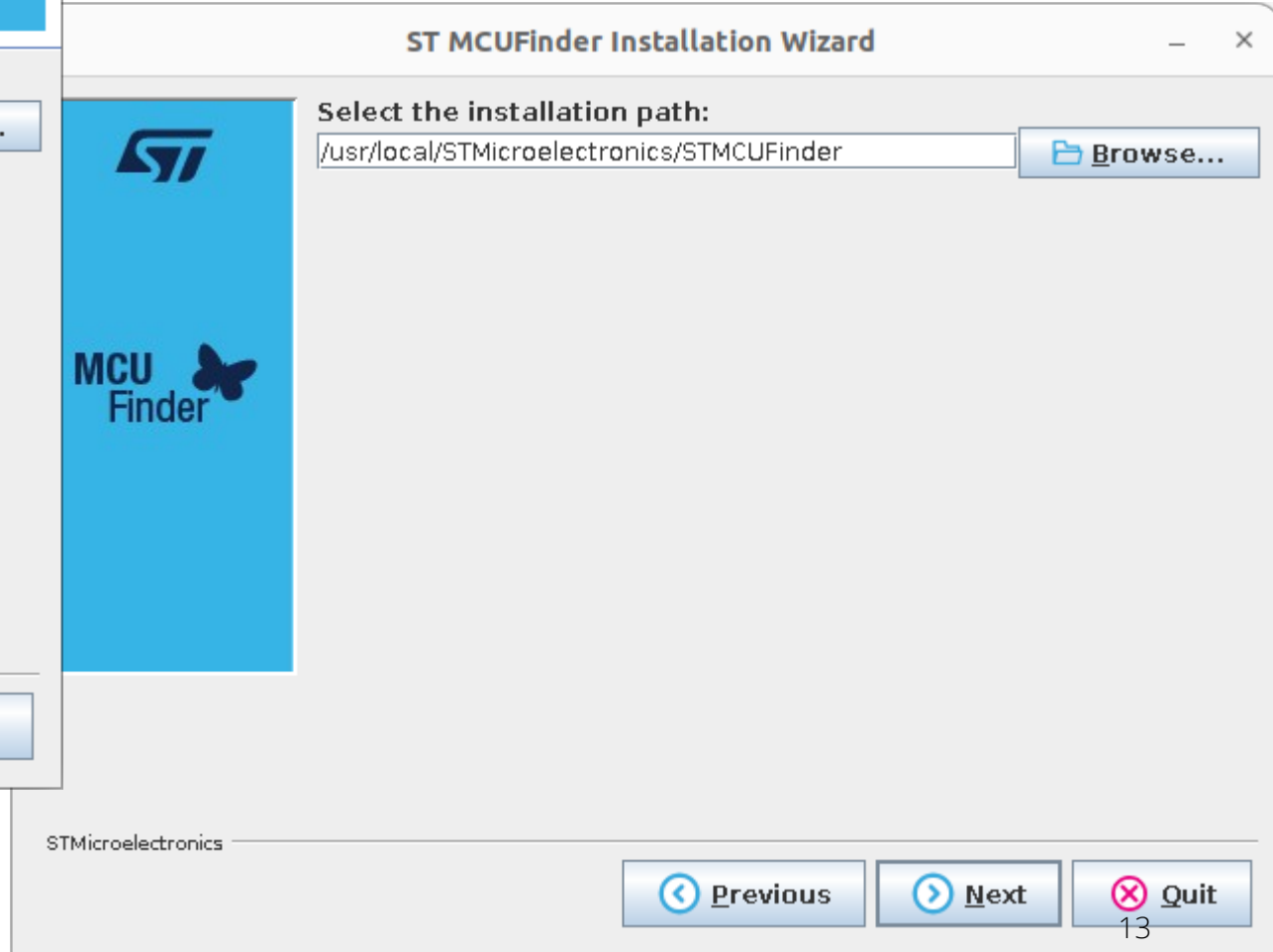
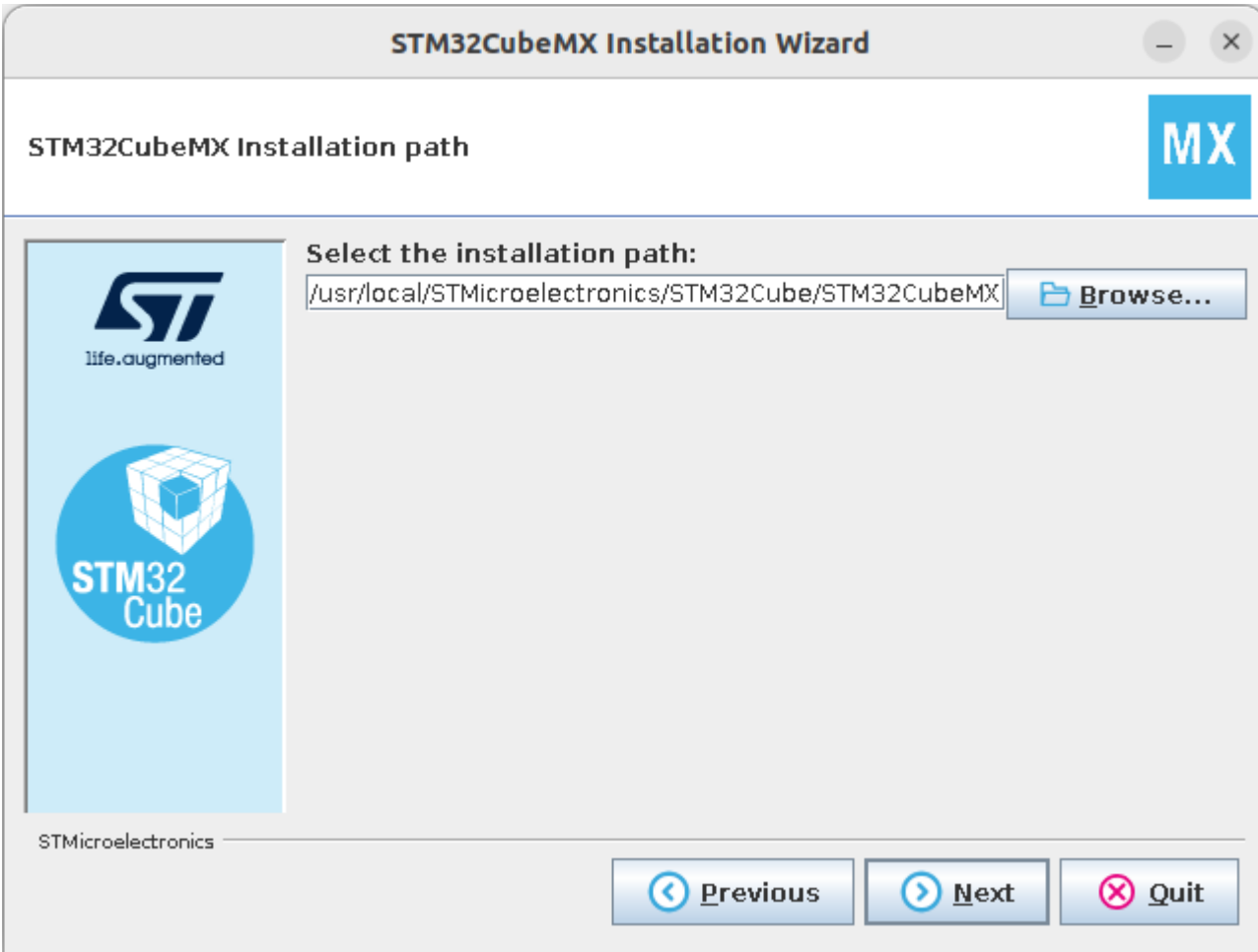
Name	Size	Mo
en.st-mcu-finderlin-v6-1-0.zip	327.9 MB	
en.stm32cubemx-lin-v6-14-1.zip	703.5 MB	
en.st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh.zip	486.1 MB	
st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh		
- Dialog box: "Select Application" with title "Opening 'Zip archive' files." It lists "Recommended Applications":
 - Archive Manager (highlighted)
 - FilesButtons: Cancel, Select, View All Applications.

Install the Prerequisites – STM32CubeCLT

```
fred@shop-ubuntu-1770: ~/Downloads
fred@shop-ubuntu-1770:~$ cd Downloads/
fred@shop-ubuntu-1770:~/Downloads$ ls -l
total 1956564
-rw-rw-r-- 1 fred fred 703471499 Apr 16 13:45 en.stm32cubemx-lin-v6-14-1.zip
-rw-rw-r-- 1 fred fred 327926737 Apr 16 13:46 en.st-mcu-finderlin-v6-1-0.zip
-rw-rw-r-- 1 fred fred 486054626 Apr 16 13:42 en.st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh.zip
-rw-rw-r-- 1 fred fred 486054368 Feb 26 12:38 st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh
fred@shop-ubuntu-1770:~/Downloads$ chmod +x st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh
fred@shop-ubuntu-1770:~/Downloads$ ls -l
total 1956564
-rw-rw-r-- 1 fred fred 703471499 Apr 16 13:45 en.stm32cubemx-lin-v6-14-1.zip
-rw-rw-r-- 1 fred fred 327926737 Apr 16 13:46 en.st-mcu-finderlin-v6-1-0.zip
-rw-rw-r-- 1 fred fred 486054626 Apr 16 13:42 en.st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh.zip
-rwxrwxr-x 1 fred fred 486054368 Feb 26 12:38 st-stm32cubeclt_1.18.0_24403_20250225_1636_amd64.deb_bundle.sh
fred@shop-ubuntu-1770:~/Downloads$
```

STM32CubeCLT (Command Line Tools)

Install the Prerequisites – STM32CubeMX/STMCUFinder



Install the Prerequisites – STM32CubeMX/STMCUFinder

@ext:STMicroelectronics.stm32-vscode-extension

User

Extensions (3)

STM32 VS Code E... (3)

STM32CubeMX (1)

STMCUFinder (1)

STM32CubeCLT (1)

STM32 VSCode Extension › Project Creator: Executable Path

Absolute path to the STM32CubeMX executable

`/usr/local/STMicroelectronics/STM32Cube/STM32CubeMX/STM32...`

STM32 VSCode Extension › Product Finder: Executable Path

Absolute path to the STMCUFinder executable

`/usr/local/STMicroelectronics/STMCUFinder/STMCUFinder`

STM32 VSCode Extension › Cube CLT: Path

Absolute path to the STM32CubeCLT folder

`/opt/st/stm32cubeclt_1.18.0`

Launch STMCUFinder

STM32 VS CODE EXTENSION

PROJECT MANAGER

- Import CMake project
- Launch STM32CubeMX
- Create empty project
- Launch STMCUFinder
- Upgrade ST Link firmware

STMCUFinder

MCU/MPU Selector | Board Selector | Example Selector | Cross Selector

Board Filters

Commercial Part Number: NUCLEO-G031K8

PRODUCT INFO

- Type
- Supplier
- MCU / MPU Series
- Marketing Status
- Price

MEMORY

- Ext. Flash = 0 (MBit)
- Ext. EEPROM = 0 (kBytes)
- Ext. RAM = 0 (MBit)

FEATURES

- Embedded Sensor
- User Button
- Camera

Features

Large Picture | Docs & Resources | Datasheet | Buy | STM32CubeMX

STM32G0 Series

NUCLEO-G031K8 **STM32 Nucleo-32 development board with STM32G031K8 MCU, supports Arduino nano connectivity**


ACTIVE
Product is in mass production

Part Number : NUCLEO-G031K8
Commercial Part Number : NUCLEO-G031K8

Unit Price (US\$) : **10.32**
Mounted Device : [STM32G031K8T6](#)

The STM32 Nucleo-32 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller. The Arduino™ Nano V3 connectivity support allows the easy expansion of the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields. The STM32 Nucleo-32 board does not require any separate probe as it integrates the ST-LINK debugger/programmer. The STM32 Nucleo-32 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Boards List: 1 item

	Overview	Commercial Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		NUCLEO-G031K8	Nucleo-32	Active	10.32	STM32G031K8T6

Launch STM32CubeMX

The screenshot displays the STM32CubeMX software interface. The main window shows the 'Pinout & Configuration' tab for the STM32G031K8Tx NUCLEO-G031K8. The central area features a pinout diagram of the STM32G031K8Tx LQFP32 microcontroller. Various pins are highlighted in green, indicating they are configured or used. These include PA13, PA12, PA11, PA10, PA9, PA8, PA7, PA6, PA5, PA4, PA3, PA2, PA1, PC14, PC15, VSS, PF2, PC6, and PA14. Labels for these pins include T_VCP_TX, T_VCP_RX, T_VTCK, T_JTMS, LD3 [Green], T_NRST, RCC_OSC32_IN, and RCC_OSC32_OUT. The interface also shows a 'Software Packs' section and a 'Pinout' section. At the bottom, the status bar indicates 'Unused GPIOs: 22 / 30'.

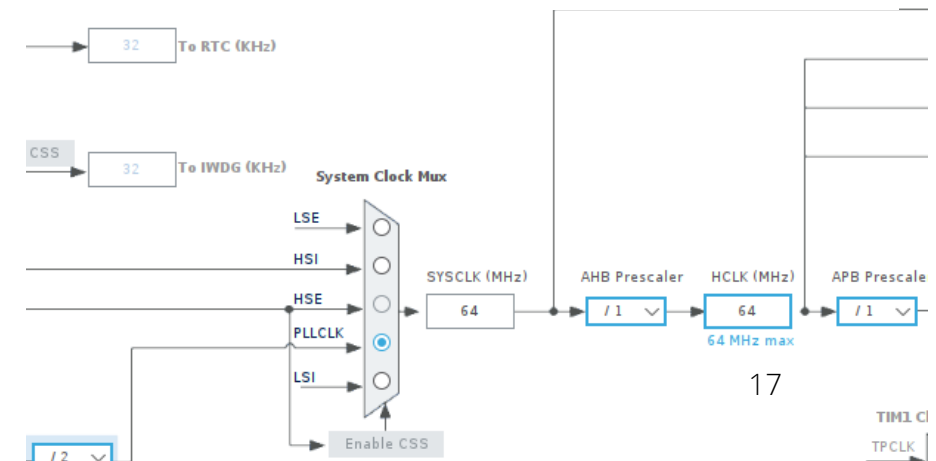
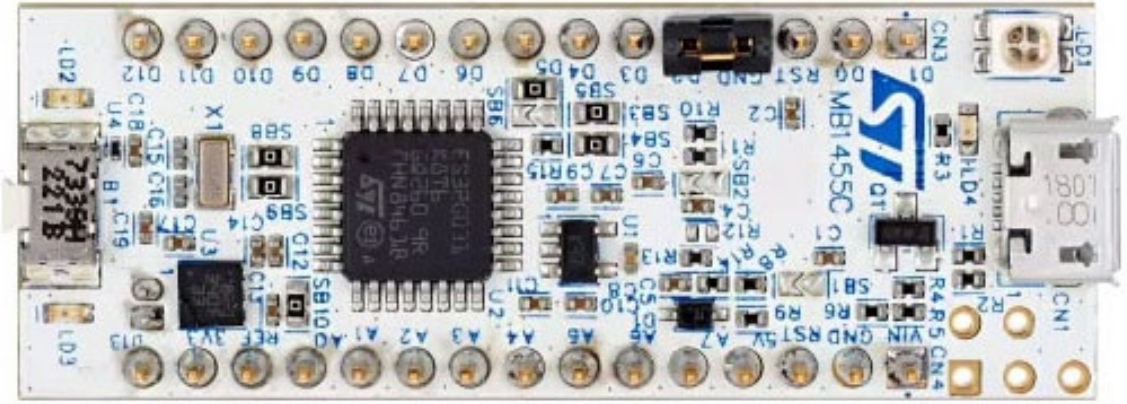
Configure the C++ Project

STM32 CubeMX

File Window Help

Home > STM32G031K8Tx - NUCLEO-G031K8 > Untitled - Project Manager >

Pinout & Configuration	Clock Configuration	Project Manager
Project	Project Settings	
	Project Name	blinkit_vsc
	Project Location	/home/fred/workspace_vsc Browse
	Application Structure	Advanced <input type="checkbox"/> Do not generate the main()
Code Generator	Toolchain Folder Location	/home/fred/workspace_vsc/blinkit_VSC/
	Toolchain / IDE	CMake <input type="checkbox"/> Generate Under Root



Import the C++ Project

The screenshot shows the VS Code interface with the STM32 VS Code Extension installed. The Project Manager on the left lists several actions, with a red arrow pointing to the 'Import project' option. A dialog box titled 'Confirm final settings | Import project (3/3)' is open, displaying the following settings:

Setting	Value	Section
Project directory	/home/fred/workspace_vsc/blinkit_VSC	Summary
Project type	CMake	
Hardware	Board > NUCLEO-G031K8 (detected)	
Toolchain	GCC	Advanced
Import project	/home/fred/workspace_vsc/blinkit_VSC	Actions

Below the dialog, a 'Code Generation' window displays a success message:

The Code is successfully generated under :
/home/fred/workspace_vsc/blinkit_VSC
Project language : C

Buttons: [Open Folder](#) [Close](#)

Configure the C++ Project For Debugging

The screenshot shows the Visual Studio Code Explorer sidebar with the project 'BLINKIT_VSC' expanded. A context menu is open over the project, listing various configuration options. A red arrow points to the 'Debug' option, which is highlighted. The menu also includes 'RelWithDebInfo', 'Release', 'MinSizeRel', and 'Add a New Preset...'. The Explorer sidebar lists files such as '.vscode', 'cmake', 'CMakeFiles', 'Core', 'Drivers', '.mxproject', 'blinkit_VSC.ioc', 'build.ninja', 'cmake_install.cmake', 'CMakeCache.txt', 'CMakeLists.txt', 'CMakePresets.json', 'compile_commands.json', 'startup_stm32g031xx.s', and 'STM32G031XX_FLASH.ld'. The background shows the Visual Studio Code interface with the 'Welcome' tab and the text 'Visual Studio Editing evolved Start New File... Open File... Open Folder'.

EXPLORER

BLINKIT_VSC

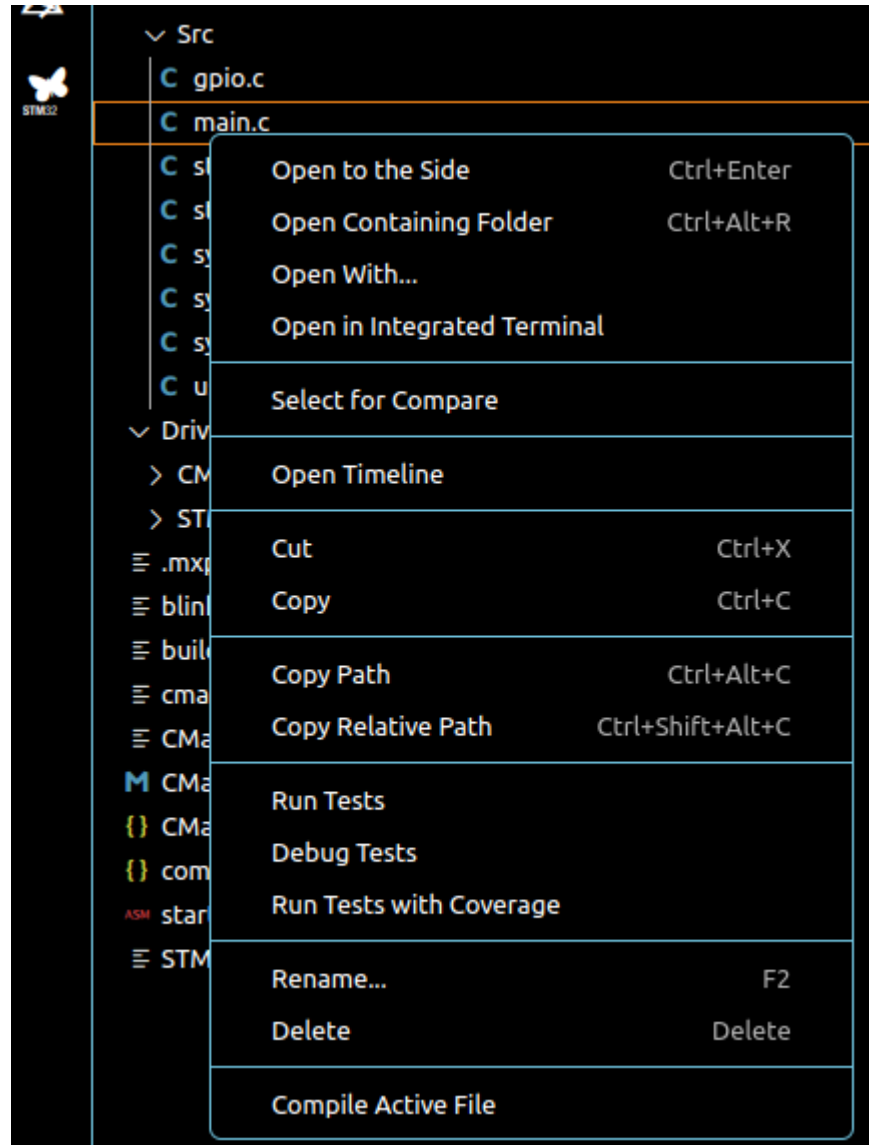
- > .vscode
- > cmake
- > CMakeFiles
- > Core
- > Drivers
- ≡ .mxproject
- ≡ blinkit_VSC.ioc
- ≡ build.ninja
- ≡ cmake_install.cmake
- ≡ CMakeCache.txt
- M CMakeLists.txt
- { } CMakePresets.json
- { } compile_commands.json
- ASM startup_stm32g031xx.s
- ≡ STM32G031XX_FLASH.ld

Select a configure preset for blinkit_VSC

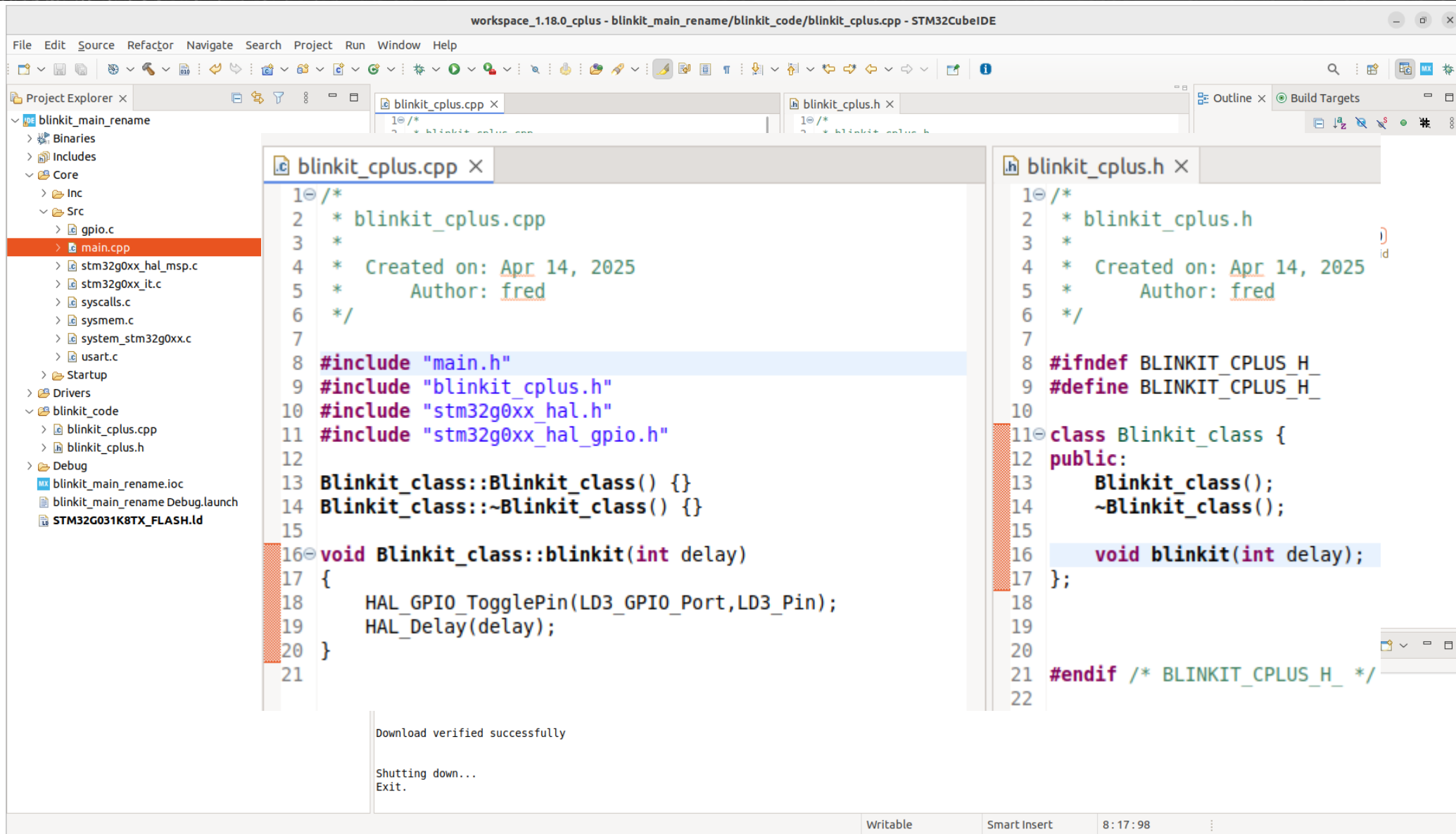
- Debug
- RelWithDebInfo
- Release
- MinSizeRel
- Add a New Preset...

Visual Studio
Editing evolved
Start
New File...
Open File...
Open Folder

Configure the C++ Project – Create main.cpp



Add Files to the C++ Project



The screenshot displays the Visual Studio Code interface for a C++ project named 'workspace_1.18.0_cplus - blinkit_main_rename/blinkit_code/blinkit_cplus.cpp - STM32CubeIDE'. The Project Explorer on the left shows the file structure, with 'main.cpp' highlighted under the 'Core' directory. The main editor shows the contents of 'blinkit_cplus.cpp' and 'blinkit_cplus.h'.

```
1 /*
2  * blinkit_cplus.cpp
3  *
4  * Created on: Apr 14, 2025
5  * Author: fred
6  */
7
8 #include "main.h"
9 #include "blinkit_cplus.h"
10 #include "stm32g0xx_hal.h"
11 #include "stm32g0xx_hal_gpio.h"
12
13 Blinkit_class::Blinkit_class() {}
14 Blinkit_class::~Blinkit_class() {}
15
16 void Blinkit_class::blinkit(int delay)
17 {
18     HAL_GPIO_TogglePin(LD3_GPIO_Port,LD3_Pin);
19     HAL_Delay(delay);
20 }
21
```

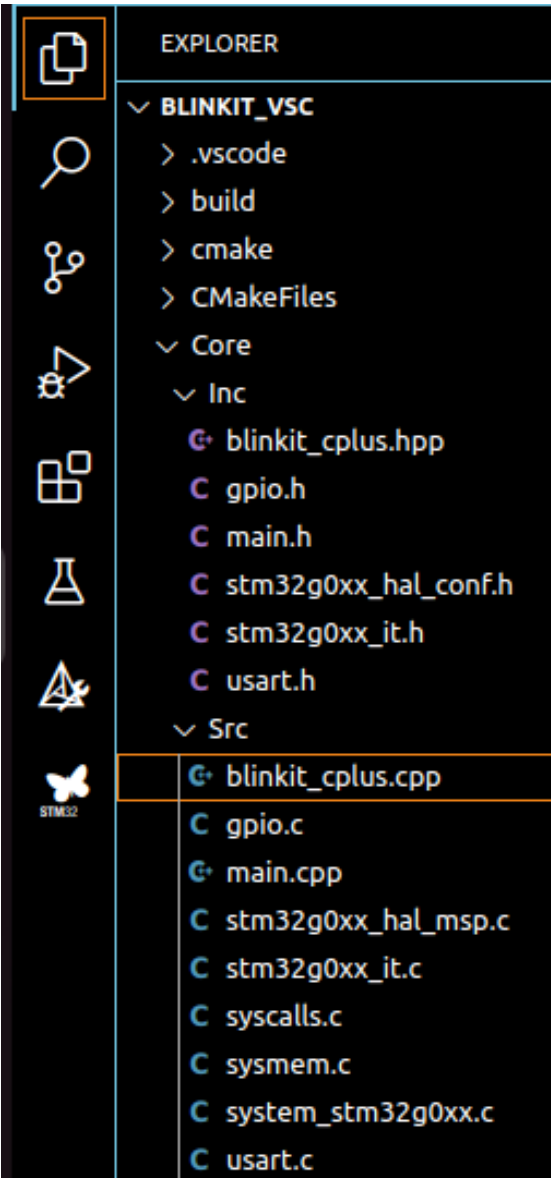
```
1 /*
2  * blinkit_cplus.h
3  *
4  * Created on: Apr 14, 2025
5  * Author: fred
6  */
7
8 #ifndef BLINKIT_CPLUS_H_
9 #define BLINKIT_CPLUS_H_
10
11 class Blinkit_class {
12 public:
13     Blinkit_class();
14     ~Blinkit_class();
15
16     void blinkit(int delay);
17 };
18
19
20
21 #endif /* BLINKIT_CPLUS_H_ */
22
```

Download verified successfully

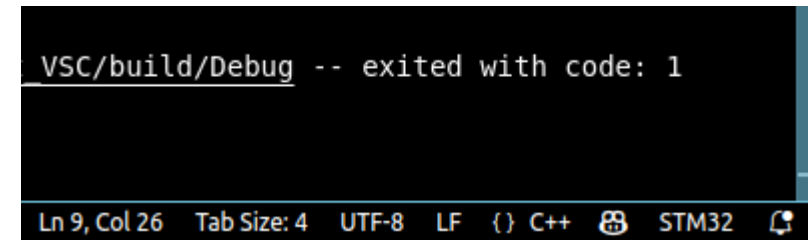
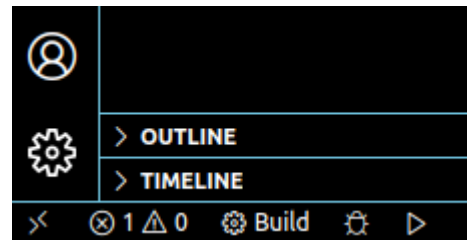
Shutting down...
Exit.

Writable Smart Insert 8:17:98

C++ Project Initial Build



```
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS MEMORY XRTOS Filter
[cmake] -- Configuring done (0.0s)
[cmake] CMake Error at cmake/stm32cubemx/CMakeLists.txt:77 (target_sources):
[cmake]   Cannot find source file:
[cmake]
[cmake]     /home/fred/workspace_vsc/blinkit_VSC/Core/Src/main.c
[cmake]
[cmake] -- Generating done (0.0s)
[cmake] CMake Generate step failed.  Build files cannot be regenerated correctly.
[proc] The command: /opt/st/stm32cubeclt_1.18.0/CMake/bin/cmake -DCMAKE_BUILD
cmake/gcc-arm-none-eabi.cmake -S/home/fred/workspace_vsc/blinkit_VSC -B/home/
```



Add Required Files to cmake/stm32cubemx/CMakeLists.txt

cplusplus_vs_code/CMakeLists.txt

```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS MEMORY XRTOS Filter
[cmake] -- Configuring done (0.0s)
[cmake] CMake Error at cmake/stm32cubemx/CMakeLists.txt:77 (target_sources):
[cmake] Cannot find source file:
[cmake]
77 # Add STM32CubeMX generated application sources to the project
78 target_sources(${CMAKE_PROJECT_NAME} PRIVATE ${MX_Application_Src})
[cmake]
[cmake] -- Generating done (0.0s)

```

```

11 # STM32CubeMX generated include paths
12 set(MX_Include_Dirs
13     ${CMAKE_SOURCE_DIR}/Core/Inc
14     ${CMAKE_SOURCE_DIR}/Drivers/STM32G0xx_HAL_Driver/Inc
15     ${CMAKE_SOURCE_DIR}/Drivers/STM32G0xx_HAL_Driver/Inc/Legacy
16     ${CMAKE_SOURCE_DIR}/Drivers/CMSIS/Device/ST/STM32G0xx/Include
17     ${CMAKE_SOURCE_DIR}/Drivers/CMSIS/Include
18 )
19
20 # STM32CubeMX generated application sources
21 set(MX_Application_Src
22     ${CMAKE_SOURCE_DIR}/Core/Src/main.c
23     ${CMAKE_SOURCE_DIR}/Core/Src/gpio.c
24     ${CMAKE_SOURCE_DIR}/Core/Src/usart.c
25     ${CMAKE_SOURCE_DIR}/Core/Src/stm32g0xx_it.c
26     ${CMAKE_SOURCE_DIR}/Core/Src/stm32g0xx_hal_msp.c
27     ${CMAKE_SOURCE_DIR}/Core/Src/systemem.c
28     ${CMAKE_SOURCE_DIR}/Core/Src/syscalls.c
29     ${CMAKE_SOURCE_DIR}/startup_stm32g031xx.s
30 )

```

```

d. | 11 # STM32CubeMX generated include paths
cube| 12 set(MX_Include_Dirs
home| 13     ${CMAKE_SOURCE_DIR}/Core/Inc
14     ${CMAKE_SOURCE_DIR}/Drivers/STM32G0xx_HAL_Driver/Inc
15     ${CMAKE_SOURCE_DIR}/Drivers/STM32G0xx_HAL_Driver/Inc/Legacy
16     ${CMAKE_SOURCE_DIR}/Drivers/CMSIS/Device/ST/STM32G0xx/Include
17     ${CMAKE_SOURCE_DIR}/Drivers/CMSIS/Include
18 )
19
20 # STM32CubeMX generated application sources
21 set(MX_Application_Src
22     ${CMAKE_SOURCE_DIR}/Core/Src/main.cpp
23     ${CMAKE_SOURCE_DIR}/Core/Src/blinkit_cplus.cpp }
24     ${CMAKE_SOURCE_DIR}/Core/Src/gpio.c
25     ${CMAKE_SOURCE_DIR}/Core/Src/usart.c
26     ${CMAKE_SOURCE_DIR}/Core/Src/stm32g0xx_it.c
27     ${CMAKE_SOURCE_DIR}/Core/Src/stm32g0xx_hal_msp.c
28     ${CMAKE_SOURCE_DIR}/Core/Src/systemem.c
29     ${CMAKE_SOURCE_DIR}/Core/Src/syscalls.c
30     ${CMAKE_SOURCE_DIR}/startup_stm32g031xx.s

```

Debug the C++ Project

The screenshot displays the Visual Studio Code (VS Code) interface during a debug session. The top bar shows the 'RUN AND DEBUG' mode with the configuration 'Build & Debug Microcontroller - ST-Link'. The active file is 'main.cpp'.

VARIABLES and WATCH: The left sidebar shows the 'VARIABLES' and 'WATCH' panels. Under 'Registers', a register with value '0x' is visible. The 'WATCH' panel is currently empty.

main.cpp Source Code: The main editor shows the following code:

```
65 int main(void)
74 /* Reset of all peripherals. Initializes the Flash interface and the Systick */
75 HAL_Init();
76
77 /* USER CODE BEGIN 2 */
78
79 /* USER CODE END 2 */
80
81 /* Configure the SystemClock_Config
82
83 /* USER CODE BEGIN 3 */
84
85 /* USER CODE END 3 */
86
87 /* Initialize all user application peripherals (without HAL_Init)
88 MX_GPIO_Init();
89 MX_USART2_UART_Init();
90
91 /* USER CODE BEGIN 2 */
92 Blinkit_class blinkled;
93 /* USER CODE END 2 */
94
95 /* Infinite loop */
96 /* USER CODE BEGIN WHILE */
97 while (1)
98 {
99     blinkled.blinkit(100);
100
101     /* USER CODE END WHILE */
102
103     /* USER CODE BEGIN 3 */
104 }
105 /* USER CODE END 3 */
106 }
```

Debug the C++ Project



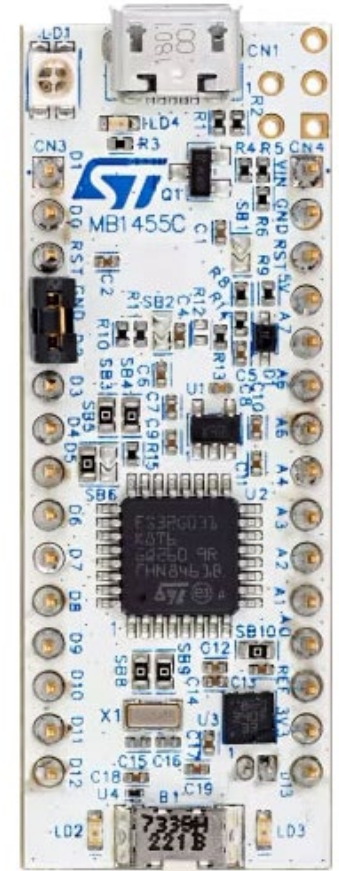
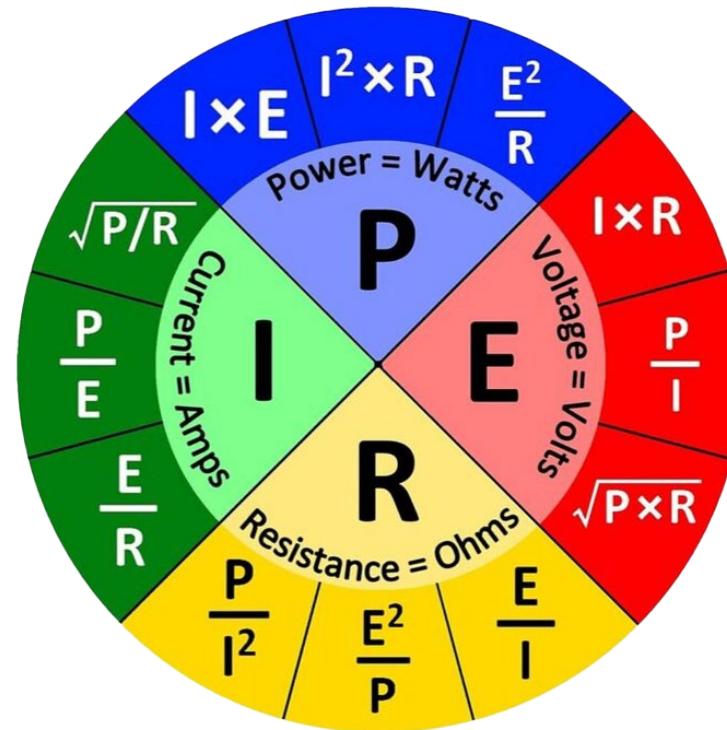
Next Time...

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- [Today's Download Package](#)





DesignNews

Thank You

Sponsored by

DigiKey

