

CÔNG TY CỔ PHẦN CÔNG NGHỆ OKAI



TÀI LIỆU HƯỚNG DẪN PHÁT TRIỂN

N76E003AT20

LỊCH SỬ UPDATE

Phiên bản	Ngày Update	Người Update	Ghi chú
V1.0	18/10/2025	Lê Xuân Hiếu	Update lần đầu

WWW.OKAI.VN

Mục lục

1. Giới thiệu về Nuvoton N76E003AT20.....	5
2. Quy trình phát triển chip N76E003AT20.....	6
3. Hướng dẫn cài đặt môi trường phát triển.....	7
3.1. Phát triển trên Visual Studio Code (VS code).....	7
3.1.1 Cài đặt.....	7
3.1.2. Tạo dự án.....	12
3.2 Phát triển dự án trên Nu Eclipse.....	20
3.2.1 Cài đặt.....	20
3.2.2 Mở một dự án mẫu.....	22
3.2.3 Thêm thư viện.....	28
3.2.4 Tạo thư viện.....	30
3.2.5 Debug.....	32
4. Một số lỗi thường gặp.....	33

Lời nói đầu

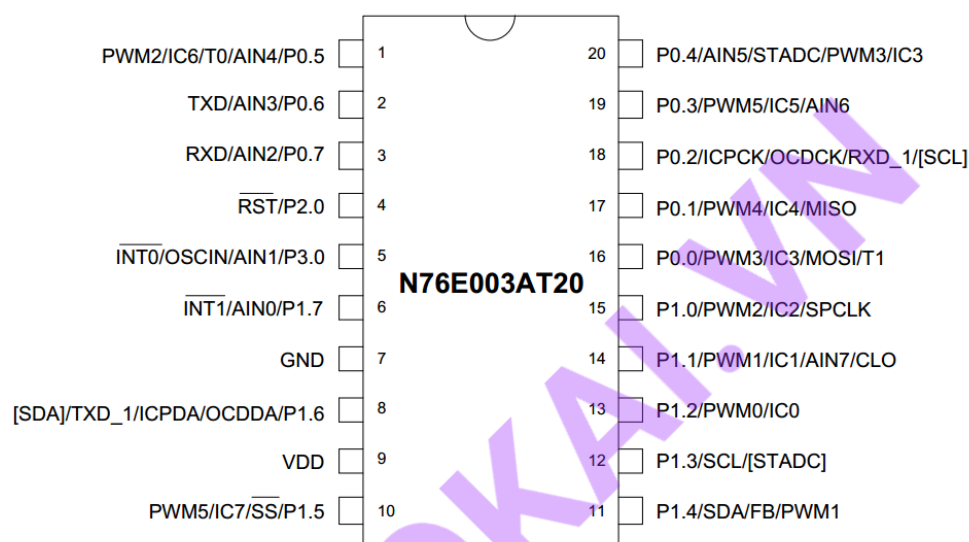
Tài liệu này được biên soạn bởi Công ty Cổ phần Công nghệ OKAI nhằm hướng dẫn quy trình phát triển và nạp chương trình cho vi điều khiển N76E003AT20 của hãng Nuvoton. Nội dung tài liệu tập trung vào các bước thực hành cụ thể, giúp kỹ sư và kỹ thuật viên có thể thiết lập môi trường phát triển trên VS Code và Nu Eclipse, tạo và biên dịch dự án sử dụng bộ biên dịch SDCC, cũng như nạp chương trình (flash) vào chip N76E003 thông qua mạch nạp NuLink.

Khác với các tài liệu giảng dạy lập trình hay đào tạo tư duy phần mềm, tài liệu này chỉ tập trung vào hướng dẫn thao tác kỹ thuật thực tế, giúp người dùng nhanh chóng làm chủ quá trình phát triển phần cứng và phần mềm cơ bản cho vi điều khiển Nuvoton. Ngoài các bước cài đặt và sử dụng công cụ, tài liệu còn tổng hợp một số lỗi thường gặp trong quá trình phát triển cùng các phương án khắc phục, được đúc kết từ kinh nghiệm triển khai nội bộ tại OKAI.

Với mục tiêu chuẩn hóa quy trình phát triển vi điều khiển N76E003 trong các dự án của công ty, tài liệu này hy vọng sẽ trở thành nguồn tham khảo nội bộ tin cậy, giúp rút ngắn thời gian làm quen, đảm bảo tính thống nhất trong công tác phát triển và bảo trì sản phẩm.

1. Giới thiệu về Nuvoton N76E003AT20

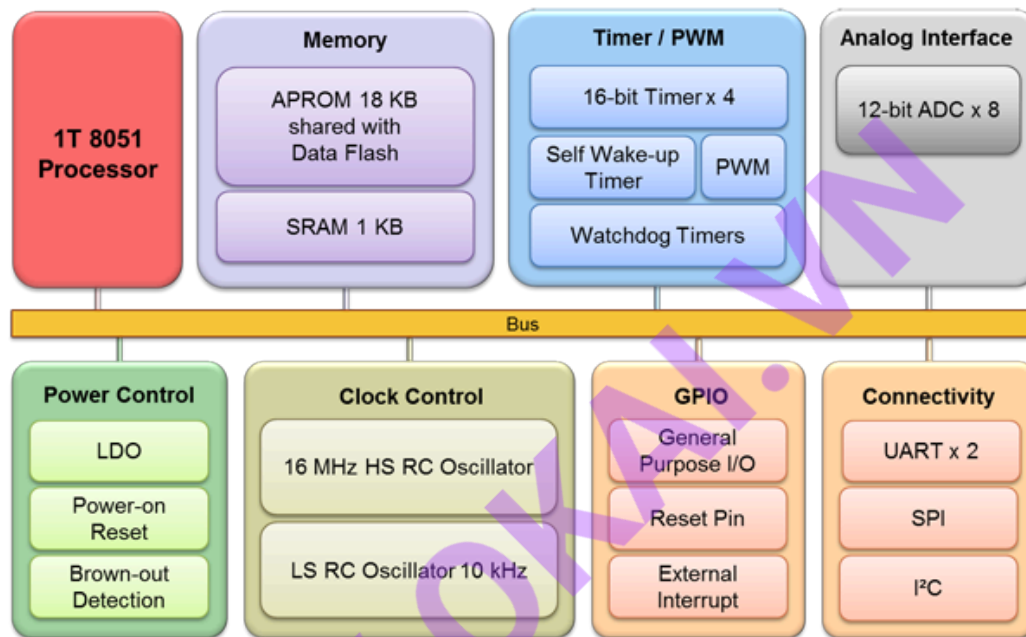
N76E003AT20 là một vi điều khiển 8 bit (MCU) thuộc dòng N76E003 của Nuvoton, sử dụng kiến trúc 8051, tối ưu cho các ứng dụng nhúng với yêu cầu chi phí thấp và hiệu năng vừa phải.



Một số đặc điểm nổi bật:

- Bộ nhớ Flash chính (APROM) lên tới ~ 18 kB, hỗ trợ chức năng In-Application Programming (IAP) để cập nhật firmware khi thiết bị đang hoạt động
- Điện áp hoạt động rộng: từ 2,4 V đến 5,5 V
- Tần số hoạt động: 16 MHz
- Hỗ trợ các giao tiếp tiêu chuẩn như UART, SPI, I²C; có ADC, PWM, Watchdog Timer, và các chế độ tiết kiệm năng lượng.
- Phạm vi nhiệt độ hoạt động công nghiệp: -40 °C đến +105 °C
- Số chân: gói 20-TSSOP (20 chân bề mặt)

- Dòng vi điều khiển N76E003 nói chung được thiết kế hướng tới các ứng dụng điều khiển nhúng, thiết bị gia dụng, cảm biến, hệ thống điều khiển nhỏ, hoặc các hệ thống nhúng có yêu cầu xử lý đơn giản nhưng tiết kiệm chi phí và năng lượng.



Datasheet chính thức của dòng vi điều khiển N76E003 tại trang của Nuvoton:

<https://www.nuvoton.com/products/microcontrollers/8bit-8051-mcus/low-pin-count-8051-series/n76e003/>

2. Quy trình phát triển chip N76E003AT20

Quy trình phát triển chip N76E003AT20 có thể được chia thành 5 bước như sau:

Bước 1. Trình soạn thảo/môi trường phát triển tích hợp: Sử dụng các trình soạn thảo như Notepad++, Vscodex,... .Hoặc các IDE như KEILC, IAR, Nu eclipse(IDE hãng) để viết chương trình cho N76E003AT20.

Bước 2. Trình biên dịch: Hiện tại có 3 trình biên dịch phổ biến cho dòng 8051 là Keil, IAR, SDCC. Keil và IAR là các trình biên dịch thương mại còn SDCC là trình biên dịch mã nguồn mở hoàn toàn miễn phí.

Bước 3. Tạo file hex: Các công cụ trình biên dịch sẽ tạo ra file này, Nếu dùng trình biên dịch (không cài công cụ mở rộng) phải gõ Terminal để tạo ra file hex hoặc tạo ra một makefile hoặc dùng python để viết lệnh tự động hóa cho quá trình này.

Bước 4. Nạp chip: ISP và ICP là 2 cách nạp, ISP cần bootloader sau đó có thể dùng mạch uart để nạp thông qua chân TX và RX, ICP thì cần mạch nạp chuyên dụng như Nu-link.

Bước 5. Chạy chip/Debug: sau khi nạp chip xong cần test xem mạch đã chạy đúng với yêu cầu đặt ra chưa nếu có lỗi cần debug chương trình.

3. Hướng dẫn cài đặt môi trường phát triển

KEILC, IAR là những môi trường phát triển miễn phí có giới hạn, nếu muốn dùng toàn bộ tính năng cần có licence, ta sẽ dùng SDCC là trình biên dịch mã nguồn mở hoàn toàn miễn phí kết hợp với một trình soạn thảo.

3.1. Phát triển trên Visual Studio Code (VS code)

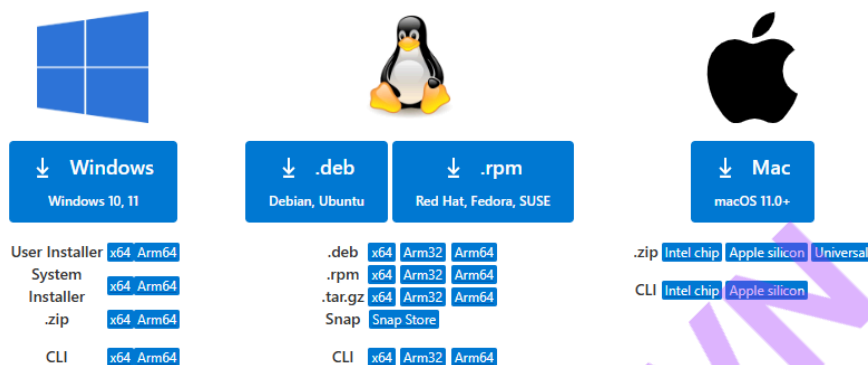
3.1.1 Cài đặt

Bước 1. Cài đặt VS code

Tải VS code tại đường dẫn này: [download](#). Hãy lựa chọn đúng hệ điều hành bạn đang sử dụng trên máy tính. Hướng dẫn này sử dụng hệ điều hành Windows

Download Visual Studio Code

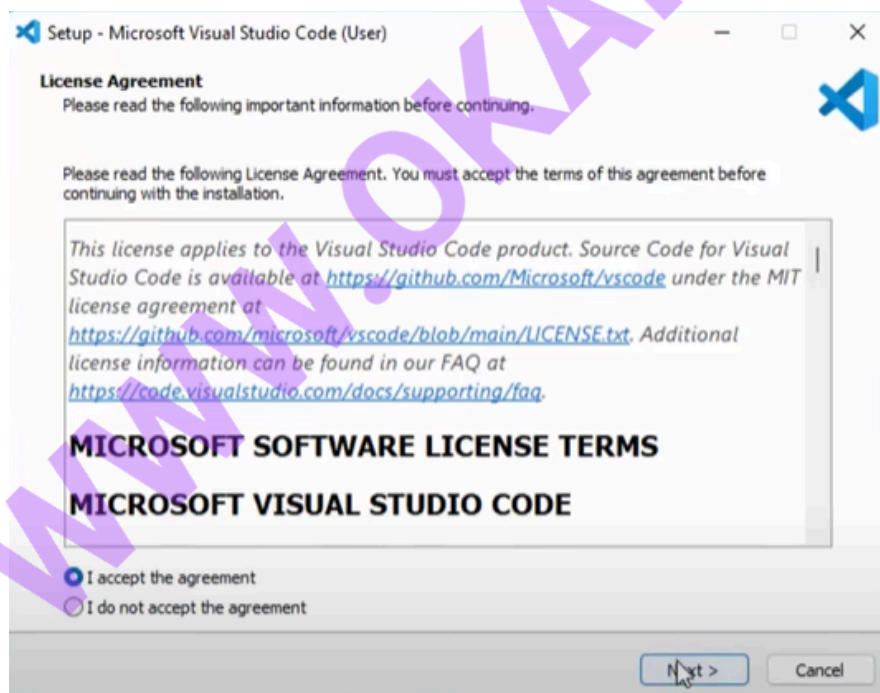
Free and built on open source. Integrated Git, debugging and extensions.



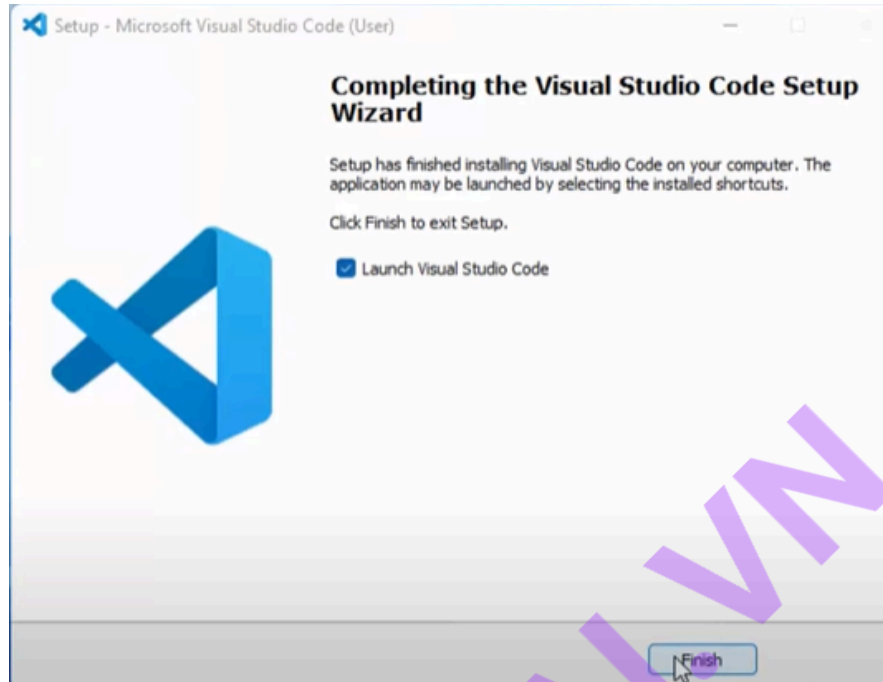
The download page features three main sections: Windows, Linux, and Mac. Each section has a download button and a list of available architectures.

Platform	Download Button	Available Architectures
Windows	Windows 10, 11	User Installer: x64, Arm64; System Installer: x64, Arm64; .zip: x64, Arm64; CLI: x64, Arm64
Linux	.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 11.0+	.zip: Intel chip, Apple silicon, Universal; CLI: Intel chip, Apple silicon

Mở ứng dụng vừa download về, chọn “I accept the agreement”

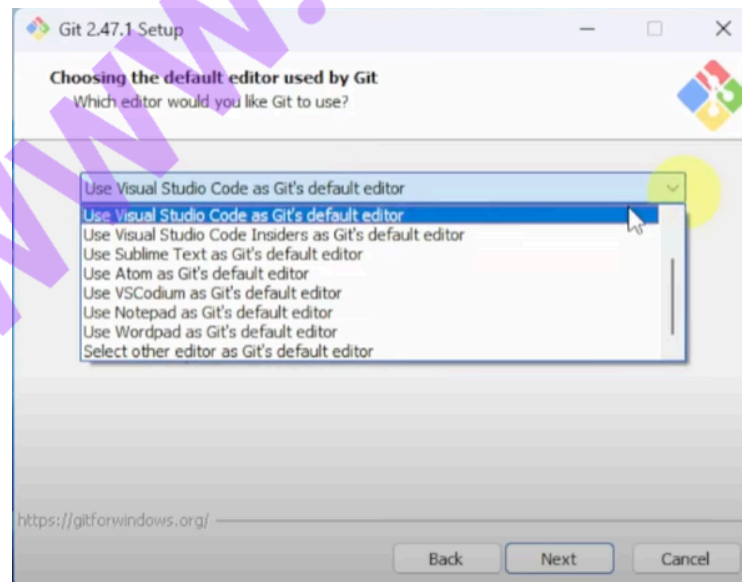


Ấn “Next” cho đến khi hiện chữ hiện chữ “Finish”

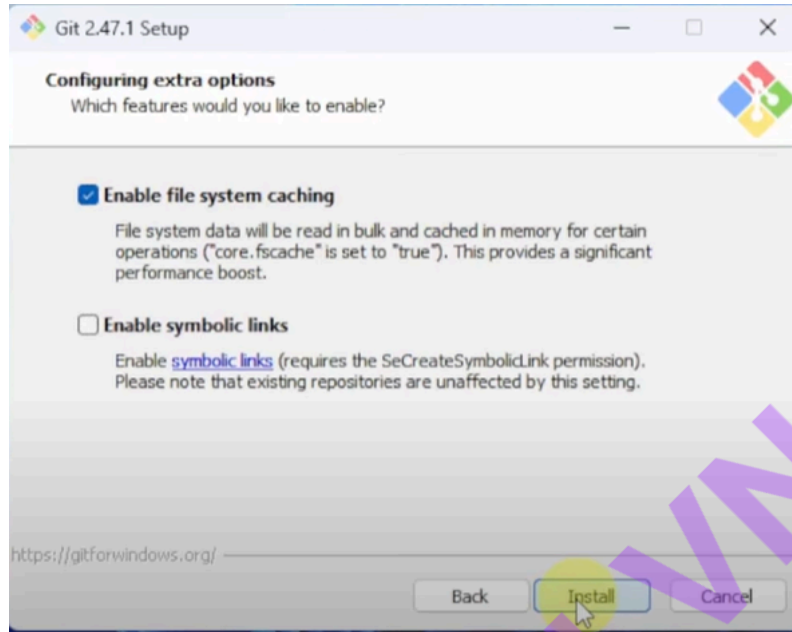


Bước 2. Cài đặt Git

Cài đặt git [tại đây](#). Click mở phần mềm ấn “Next” cho đến khi cửa sổ như hình bên dưới hiện lên chọn “Use Visual Studio Code as Git’s default editor”



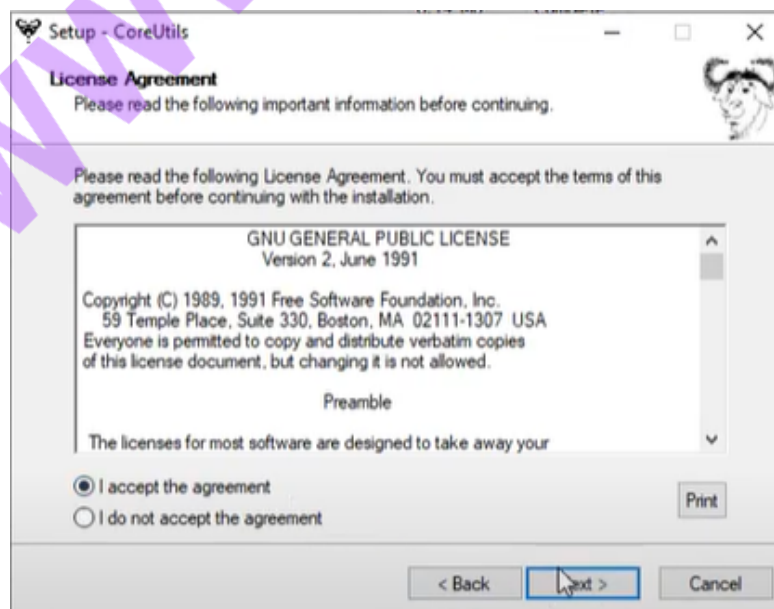
Ấn “Next” cho đến khi hiện chữ “Install”



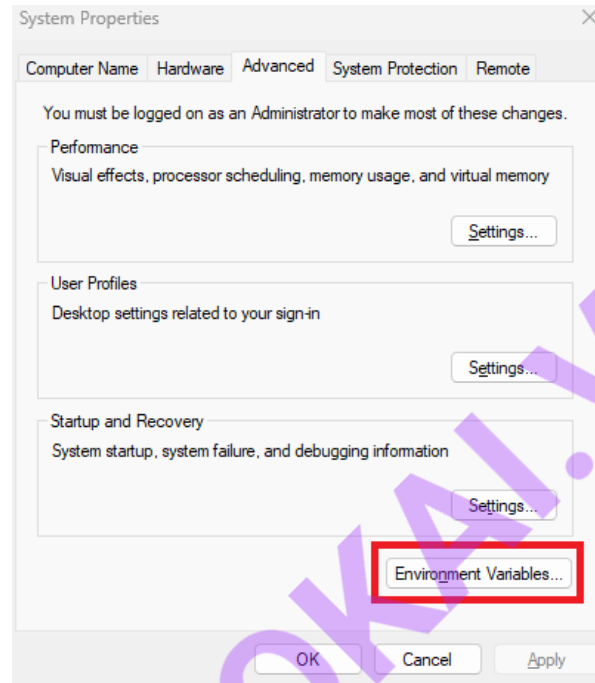
Ấn “Install” để cài đặt. Mở CMD nhập “git --version” để kiểm tra phiên bản.

Bước 3. Cài đặt Make, CoreUtils và SDCC

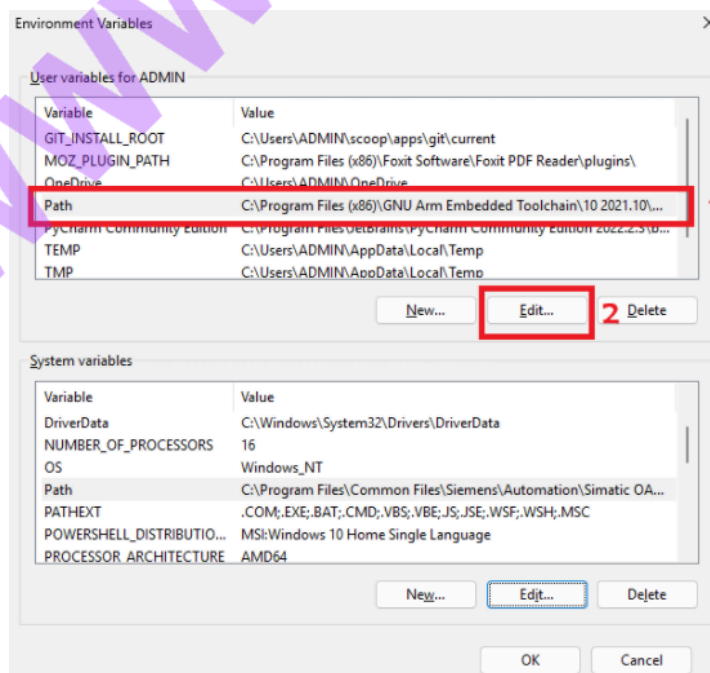
Cài đặt tại đây: [CoreUtils](#), [Make](#), [SDCC](#). Tương tự với những ứng dụng ở trên chọn “i accept the agreement” nếu có và ấn “Next” liên tục để cài đặt



Sau khi cài hết các ứng dụng trên ấn biểu tượng Window tìm từ khóa “edit the system environment variables” click chọn vào “Environment Variables...”



Tại cửa sổ “User variables for ADMIN” chọn “Path”->”Edit”



Ấn “New” sau đó điền đường dẫn “C:\Program Files (x86)\GnuWin32\bin” (điền đúng đường dẫn trên thiết bị của bạn). Để chắc chắn bạn có thể làm tương tự với mục System variable

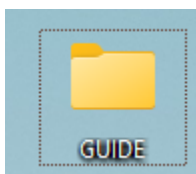
Bước 4. Cài đặt NuMicro ICP Programing Tool

Tải tại đây: [NuMicro ICP Programing Tool](#)

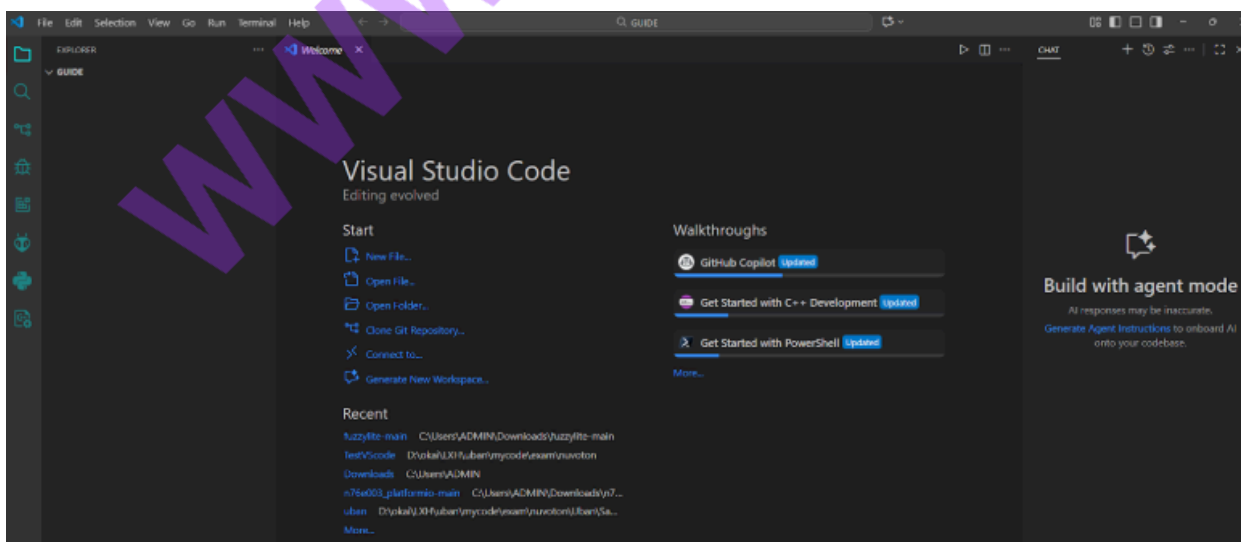
Cài đặt tương tự với các phần mềm ở trên.

3.1.2. Tạo dự án

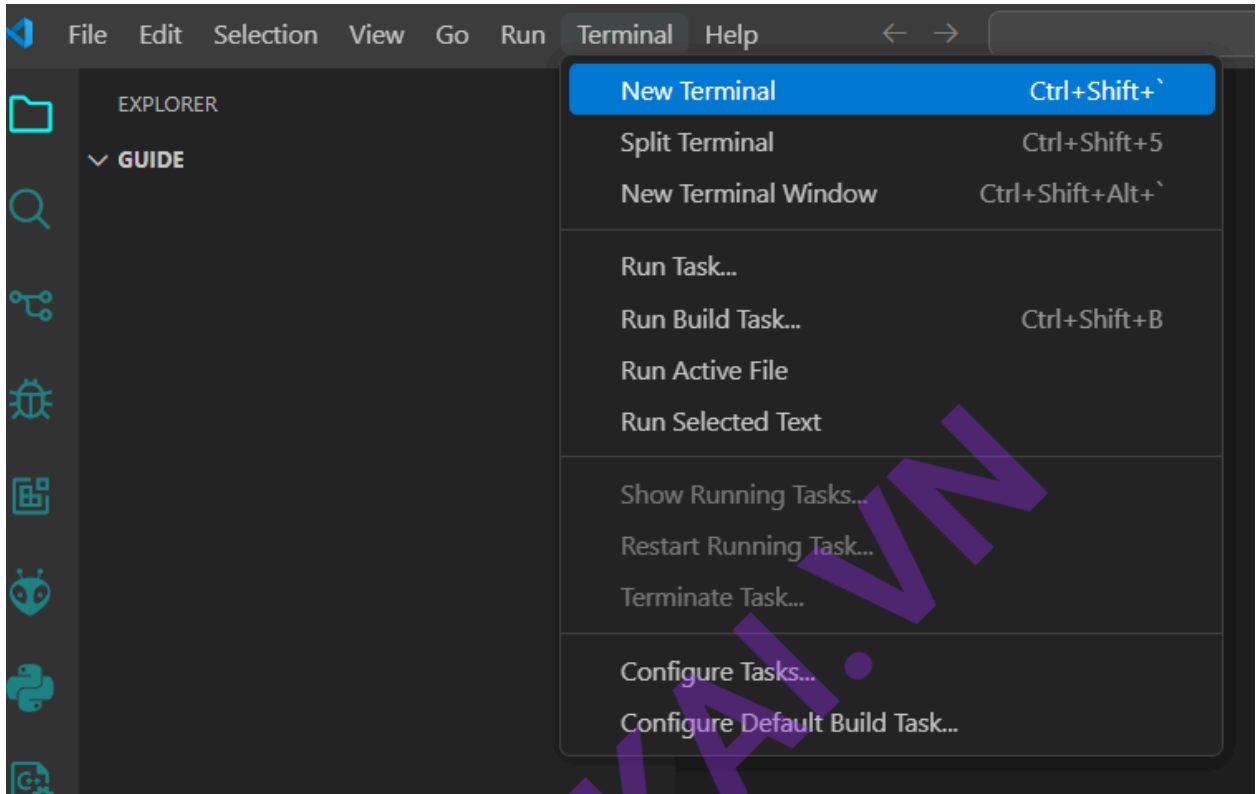
Tạo một thư mục trống ở đây tác giả đặt tên thư mục là “GUIDE”



Mở VS code, trên thanh công cụ ấn “File” -> “Open folder” -> tìm folder “GUIDE” vừa tạo -> “Select folder”



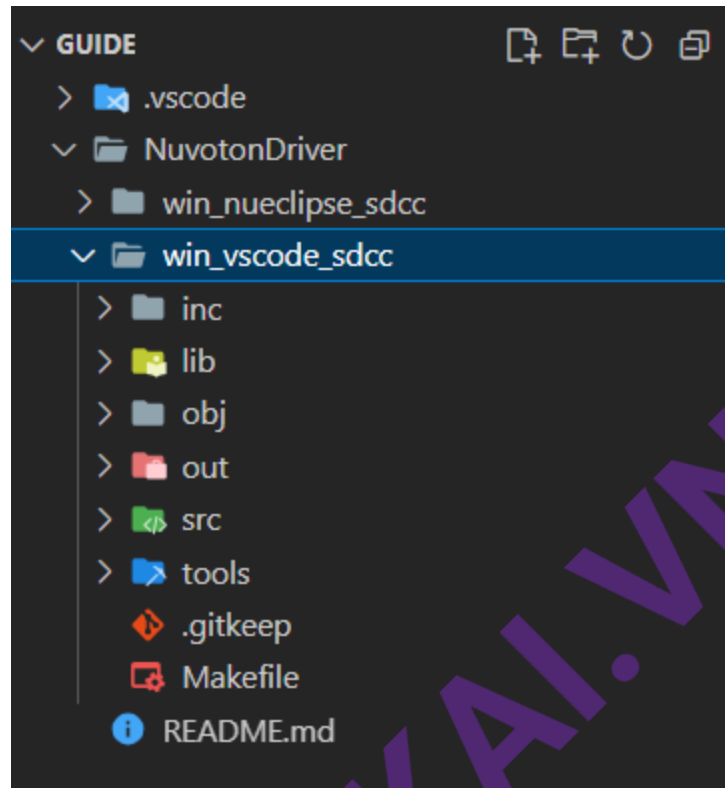
Trên thanh công cụ chọn “Terminal” -> “New Terminal”



Nhập “git clone <https://github.com/nguyennguyenokai/NuvotonDriver.git>” để tải thư viện mẫu.

```
PS C:\Users\ADMIN\Desktop\GUIDE> git clone https://github.com/nguyennguyenokai/NuvotonDriver.git
Cloning into 'NuvotonDriver'...
remote: Enumerating objects: 68, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 68 (delta 16), reused 25 (delta 2), pack-reused 0 (from 0)
Receiving objects: 100% (68/68), 31.68 KiB | 1.76 MiB/s, done.
Resolving deltas: 100% (16/16), done.
```

Sau khi tải xong góc trái màn hình sẽ hiển thị cấu trúc dự án như sau



Cần quan tâm đến các mục sau:

-> win_vscode_sdcc

Inc: mục chứa các file header của thư viện

Lib: thư mục chứa file c của thư viện

Obj: thư mục sau khi build thành sẽ chứa các asm, map, mem,...

Out: mục chứa mã hex và bin để nạp vào chip

Src: thư mục chứa chương trình chính main.c

-> win_nueclipse_sdcc (sử dụng nếu phát triển trên Nu eclipse, phần hướng dẫn ở vscode không dùng đến folder này)

Trong main.c đã có một ví dụ như sau:

```
#include "N76E003.h"

#include "SFR_Macro.h"

#include "Function_define.h"

#include "Common.h"

#include "Delay.h"

int putchar (int c) {

    while (!TI);

    TI = 0;

    SBUF = c;

    return c;

}

void main (void)

{

    InitialUART0_Timer3(115200);

    TI = 1;

    while(1)

    {

        printf_small("\n Hello world");

        Timer0_Delay1ms(300);

    }

}
```

Ví dụ này sẽ in ra serial chữ “Hello world” sau mỗi 300ms.

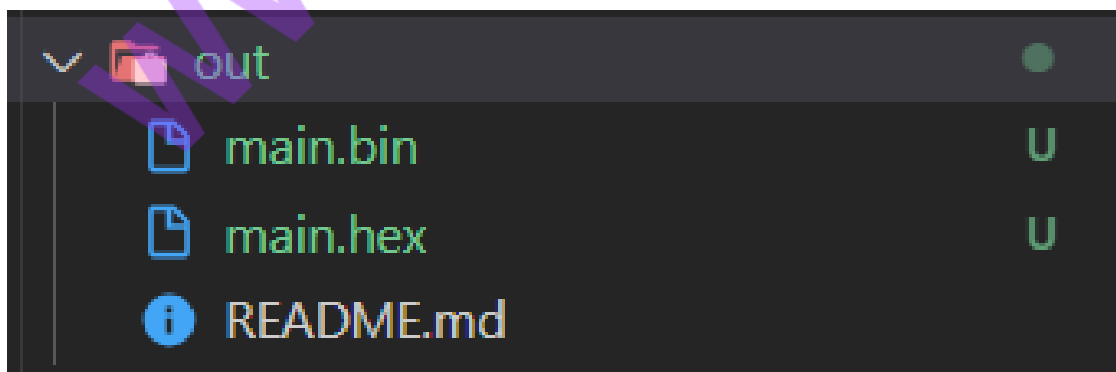
Để build ra file bin và hex, trên thanh công cụ chọn “Terminal”->”New Terminal” nhập lần lượt “cd NuvotonDriver\win_vscode_sdcc” -> “Enter” -> “make clean”->”Enter”->”make”

```

PS C:\Users\ADMIN\Desktop\GUIDE> cd .\NuvotonDriver\
PS C:\Users\ADMIN\Desktop\GUIDE\NuvotonDriver> make clean
rm -rf ./obj/*
rm -rf ./out/main.hex
rm -rf ./out/main.bin
rm -rf ./out/main.asm
sdcc -o obj/main.c.rel -I./inc -mmcs51 --model-small --out-fmt-ihx --no-xinit-opt -Dsdcc
cc -o obj/main.c.rel -I./inc -mmcs51 --model-small --out-fmt-ihx --no-xinit-opt -Dsdcc
-o obj/main.c.rel -I./inc -mmcs51 --model-small --out-fmt-ihx --no-xinit-opt -Dsdccsdcc
sdcc -o obj/main.c.rel -I./inc -mmcs51 --model-small --out-fmt-ihx --no-xinit-opt -DFOSC_160000 --peep-file tools/peep.def -c src/main.c
sdcc -o obj/Common.c.rel -I./inc -mmcs51 --model-small --out-fmt-ihx --no-xinit-opt -DFOSC_160000 --peep-file tools/peep.def -c lib/Common.c
lib/Common.c:136: warning 294: __sdcc_external_startup function definition - probable deprecated old-style variant of __sdcc_external_startup
sdcc -o obj/Delay.c.rel -I./inc -mmcs51 --model-small --out-fmt-ihx --no-xinit-opt -DFOSC_160000 --peep-file tools/peep.def -c lib/Delay.c
sdcc -o obj/main.ihx -mmcs51 --model-small --code-size 18432 --iram-size 256 --xram-size 768 --out-fmt-ihx -DFOSC_160000 obj/main.c.rel obj/Common.c.rel obj/Delay.c.rel
makebin -p obj/main.ihx out/main.bin
packihx obj/main.ihx > out/main.hex
packihx: read 67 lines, wrote 117: OK.
rm obj/Common.c.rel obj/main.c.rel obj/main.ihx obj/Delay.c.rel

```

Hiện thị “OK” là thành công. Ở mục out sẽ xuất hiện file .hex và .bin



Cắm mạch nạp vào vào MCU theo bảng tín hiệu bên dưới

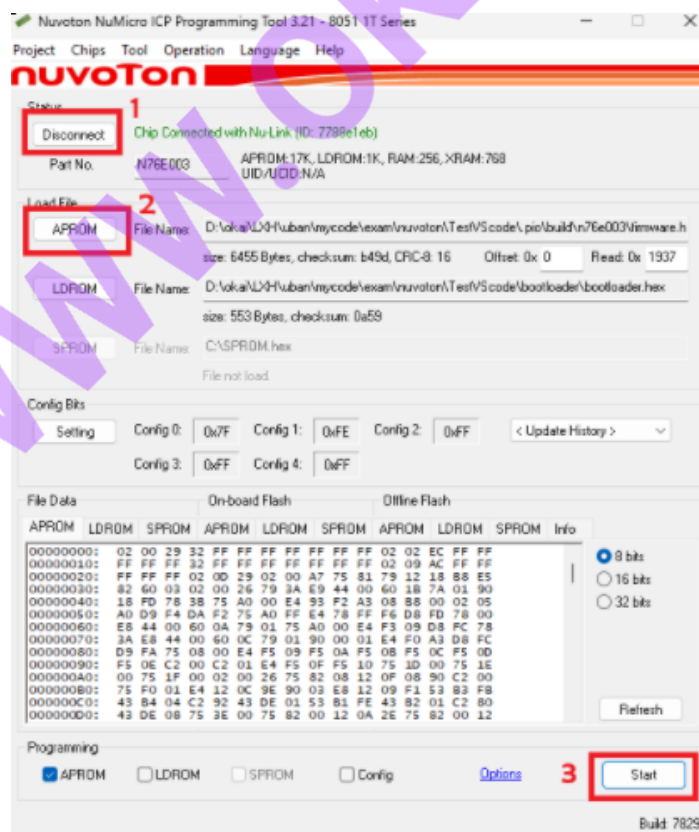


Mạch nạp Nu-link 1	N76E003AT20
3.3V	VCC
Dat	P1.6
CLK	P0.2
RST	P2.0
GND	GND

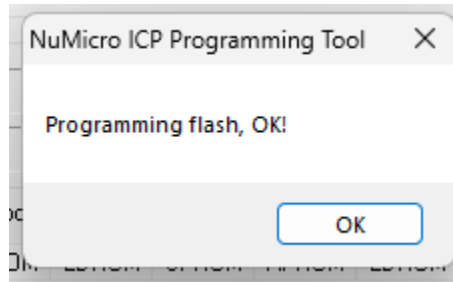
Sau đó mở phần mềm “NuMicro ICP Programming Tool” tiến hành nạp chương trình.



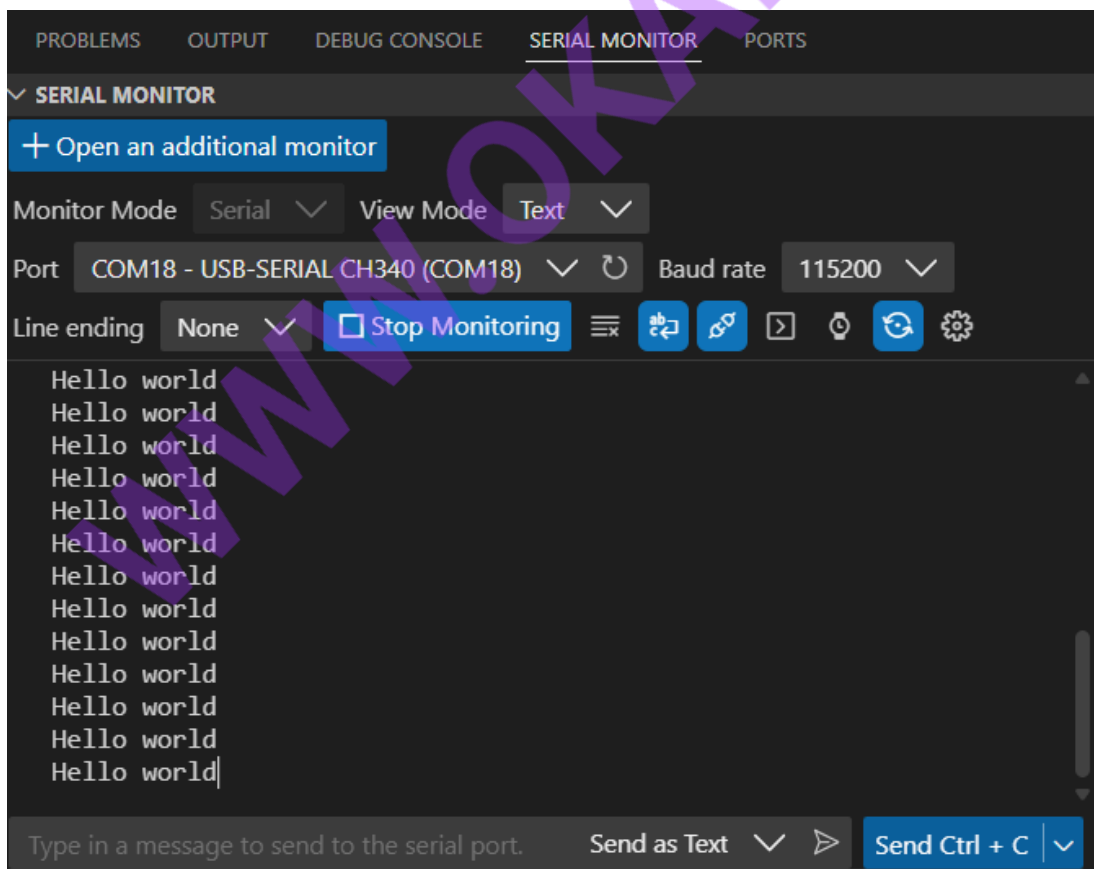
Chọn đúng MCU là “8051 1T Series” sau đó ấn “Continue”



“Connect” với MCU sau đó chọn “APROM” điền đường dẫn đến file hex vừa build ra sau đó ấn “start” nếu cửa sổ có hỏi gì ấn “No”.



Nếu cửa sổ “Programing flash, OK!” hiện ra, chip đã được nạp thành công. Bạn có thể dùng mạch [USB to TTL](#) để đọc kết quả. Đảm bảo đấu đúng sơ đồ như sau. P0.6 [TX] -> TTL[RX] , P0.7 [RX] <- TTL[TX]



Trên đây là một trong “N” cách để phát triển N76E003 trên vscode. Còn rất nhiều cách khác hay và thuận tiện cho lập trình viên phát triển MCU này, các tài liệu này sẽ được cập nhật sau.

3.2 Phát triển dự án trên Nu Eclipse

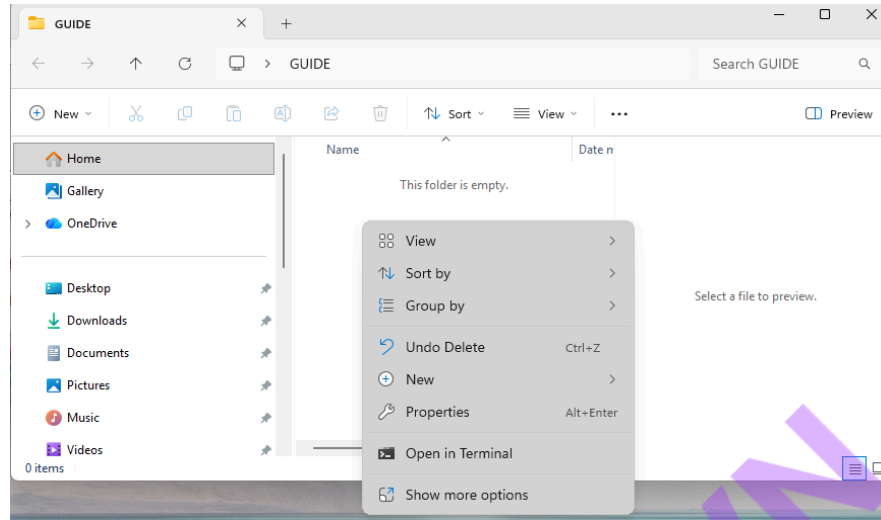
Nu eclipse là IDE do Nuvoton phát triển, nó dựa trên phần mềm eclipse được tích hợp một số tính năng để phát triển các MCU của hãng trong đó có N76E003. Nhược điểm của phần mềm này là kém thân thiện với người dùng so với vs code nhưng có thể debug khi sử dụng kit Nu-link1.

3.2.1 Cài đặt

Tải Nu eclipse [tại đây](#). Bạn có thể cài đặt nó dễ dàng tương tự với các ứng dụng khác

Lưu ý: Nu eclipse đã có SDCC và các công cụ liên quan để phát triển N76E003 nên bạn không cần cài đặt các bước như ở Visual Studio Code. Bạn chỉ cần tải [git](#) nếu chưa có.

Tương tự tạo một dự án với tên mà bạn mong muốn sau đó mở Terminal của thư mục đó lên.



Nhập “git clone <https://github.com/nguyennguyenokai/NuvotonDriver.git>”

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

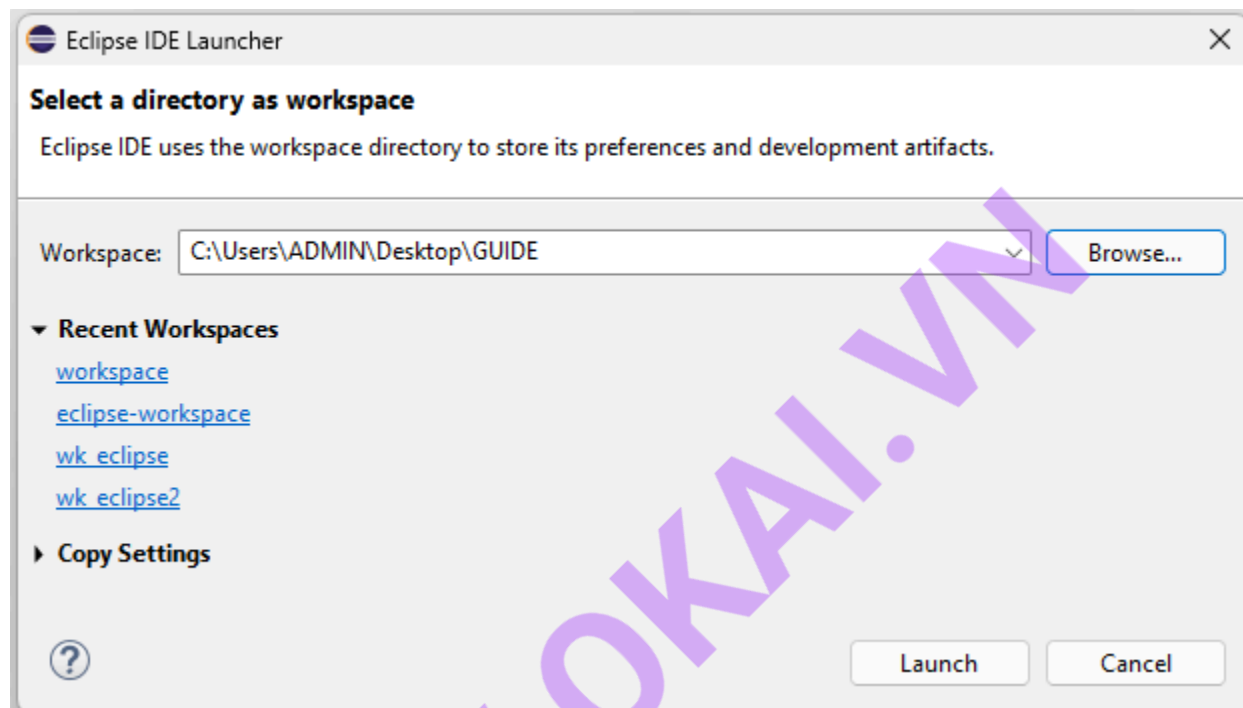
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\ADMIN\Desktop\GUIDE> git clone https://github.com/nguyennguyenokai/NuvotonDriver.git
Cloning into 'NuvotonDriver'...
remote: Enumerating objects: 135, done.
remote: Counting objects: 100% (135/135), done.
remote: Compressing objects: 100% (114/114), done.
remote: Total 135 (delta 23), reused 90 (delta 6), pack-reused 0 (from 0)
Receiving objects: 100% (135/135), 84.26 KiB | 401.00 KiB/s, done.
Resolving deltas: 100% (23/23), done.
PS C:\Users\ADMIN\Desktop\GUIDE> |
```

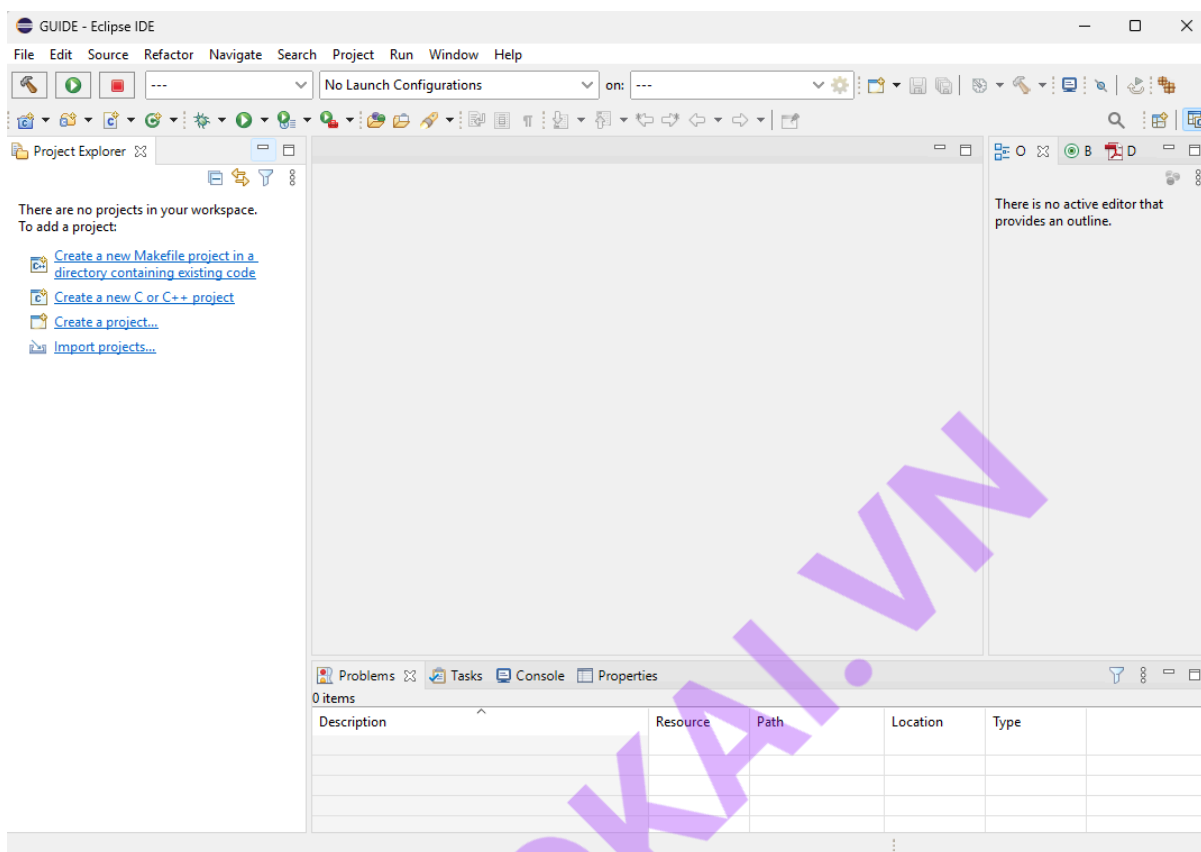
Tương tự với hướng dẫn trên vs code. Thư viện sẽ được tải về thư mục. Trong hướng dẫn này ta quan tâm đến folder “win_nueclipse_sdcc” trong này chứa cấu trúc của một dự án mẫu.

3.2.2 Mở một dự án mẫu

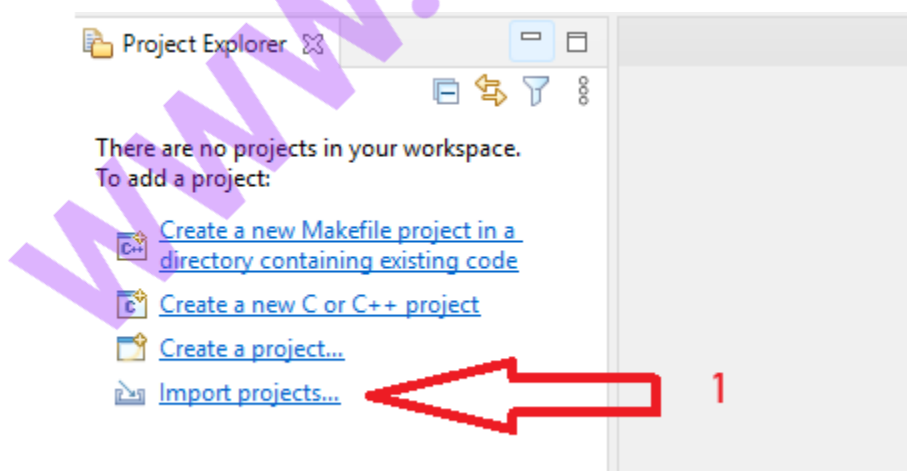
Trước khi làm việc với Nu eclipse cần phải tạo một “workspace” -> ấn “lunch” để mở

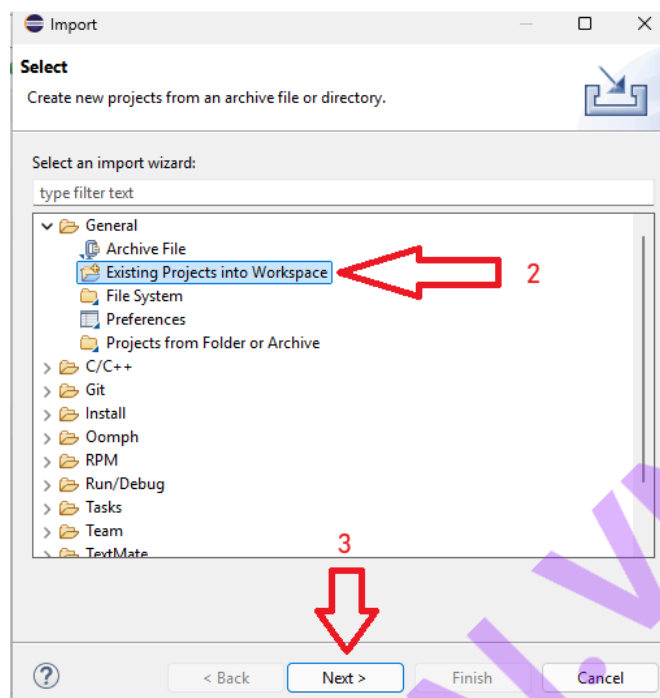


Nếu có cửa sổ “welcome” ấn “hide” ở góc phải. Giao diện của Nu eclipse sẽ hiện ra như này

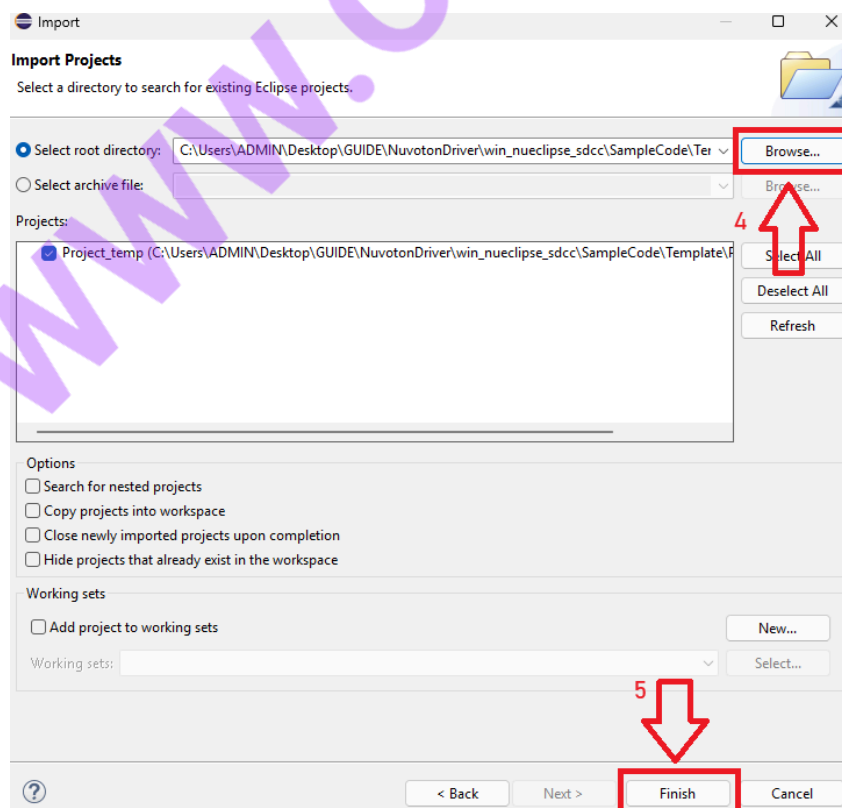


Làm lần lượt theo các bước dưới đây

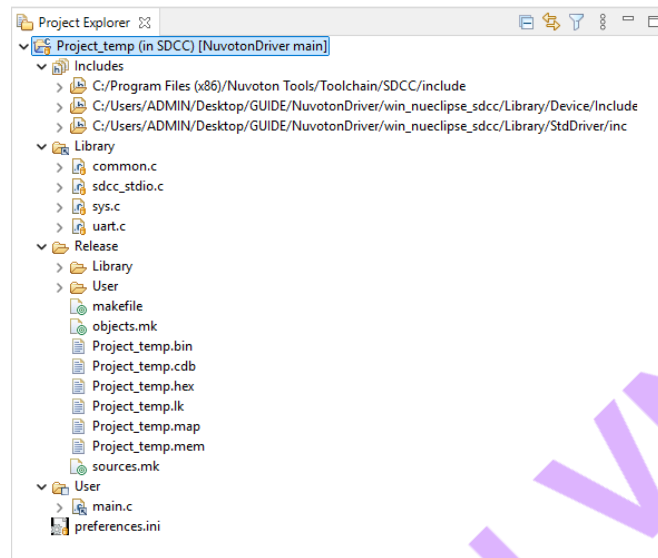




Hãy điền đúng đường dẫn đến folder SDCC của dự án ở đây đường dẫn của



Đây là cấu trúc dự án khi được mở đúng trên Nu eclipse



Includes: chứa các mục header của dự án

Library: chứa thư viện của dự án

Release: chứa các file được tạo ra trong quá trình build

User: chứa file main.c hoặc các thư viện do người dùng tạo

Trong main.c đã có sẵn một ví dụ in ra serial chữ "Hello world!"

```
#include "numicro_8051.h"

void main (void){

    MODIFY_HIRC(HIRC_166);

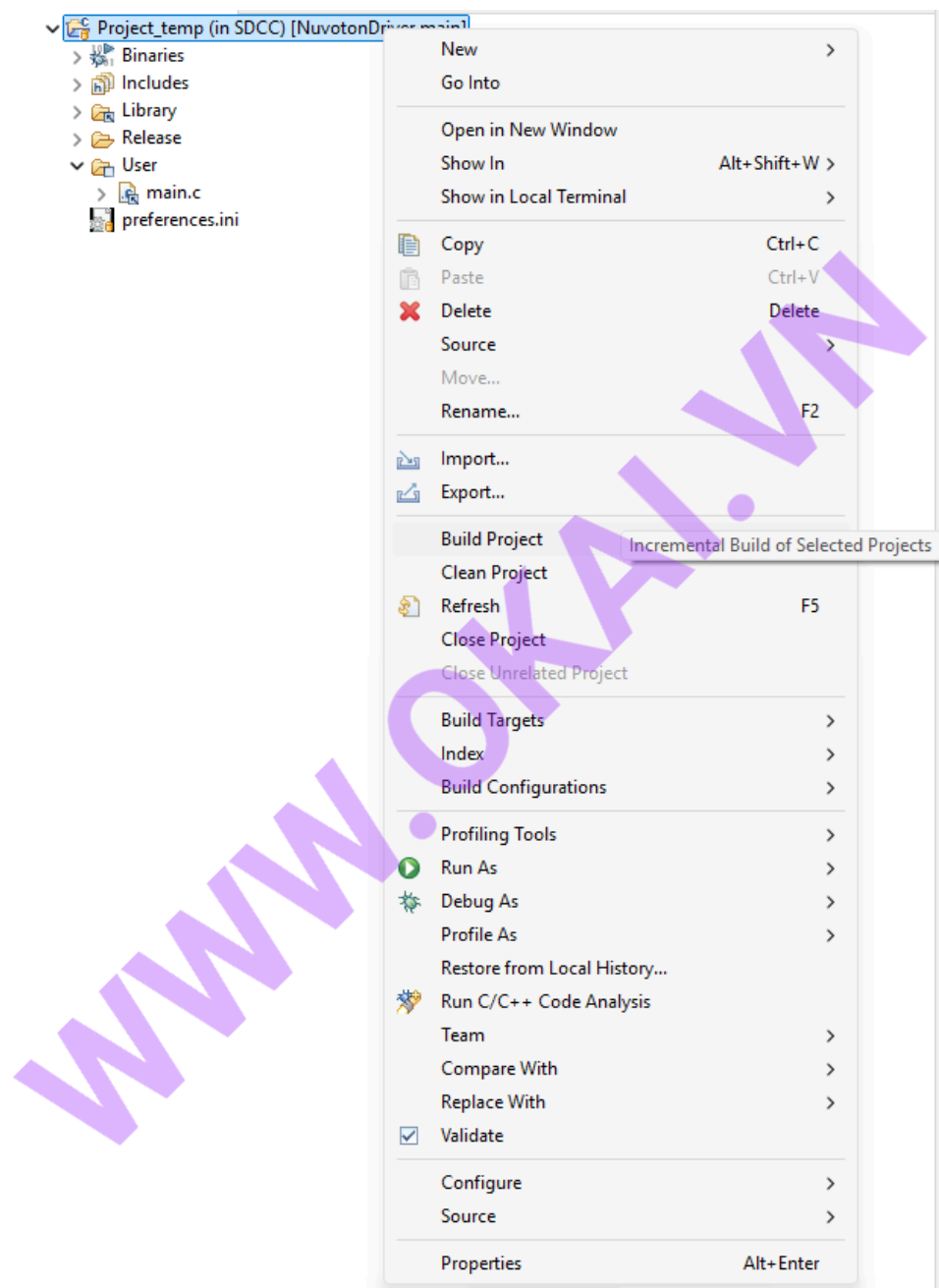
    Enable_UART0_VCOM_printf_166M_115200();

    printf("\n\r Hello world!");

    while(1);

}
```

Tiến hành build code bằng cách nhấp chuột phải vào thư mục gốc của dự án và chọn “Build project”



Mở cửa sổ console để theo dõi quá trình build

```

CDT Build Console [Project_temp]
sdcc -c --debug --out-fmt-elf -D __SDCC__ -I"C:/Users/ADMIN/Desktop/GUIDE/NuvotonDriver/win_nueclipse_sdcc/SampleCode/Template/Project_temp/SDCC/../../../../Library/Device/Include" -I
Finished building: C:/Users/ADMIN/Desktop/GUIDE/NuvotonDriver/win_nueclipse_sdcc/Library/StdDriver/src/sys.c

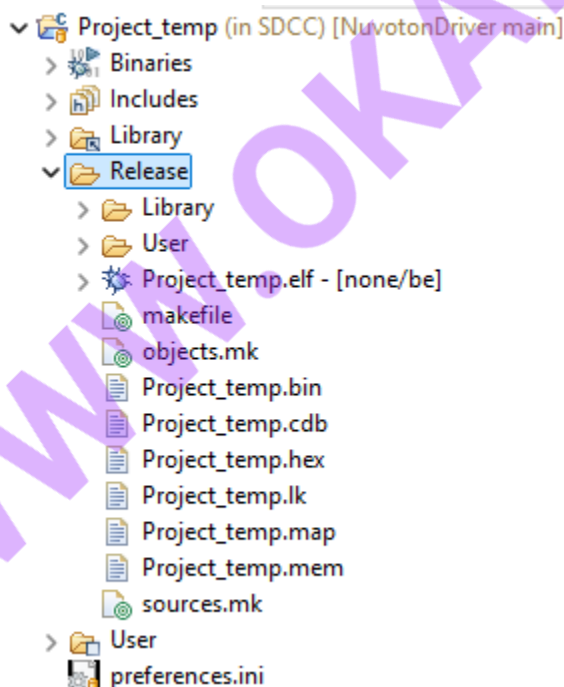
Building file: C:/Users/ADMIN/Desktop/GUIDE/NuvotonDriver/win_nueclipse_sdcc/Library/StdDriver/src/uart.c
Invoking: SDCC Compiler
sdcc -c --debug --out-fmt-elf -D __SDCC__ -I"C:/Users/ADMIN/Desktop/GUIDE/NuvotonDriver/win_nueclipse_sdcc/SampleCode/Template/Project_temp/SDCC/../../../../Library/Device/Include" -I"C:/
echo -n Library/uart.d Library/ > Library/uart.d && \
sdcc -c --debug --out-fmt-elf -D __SDCC__ -I"C:/Users/ADMIN/Desktop/GUIDE/NuvotonDriver/win_nueclipse_sdcc/SampleCode/Template/Project_temp/SDCC/../../../../Library/Device/Include" -I
Finished building: C:/Users/ADMIN/Desktop/GUIDE/NuvotonDriver/win_nueclipse_sdcc/Library/StdDriver/src/uart.c

Building target: Project_temp.elf
Invoking: SDCC Linker
sdcc --debug --out-fmt-elf -L"C:/Program Files (x86)/Nuvoton Tools/OpenOCD/bin/../../../../Toolchain/SDCC/lib/large" -o "Project_temp.elf" ./User/main.rel ./Library/common.rel ./Library/sdcc
Finished building target: Project_temp.elf

make --no-print-directory post-build
arm-none-eabi-objcopy -R REG_BANK_0 -R REG_BANK_1 -R REG_BANK_2 -R REG_BANK_3 -R BSEG -R DSEG -R ISEG -R OSEG -R SSEG -R XSEG -R XISEG -R INITIALIZER -O ihex Project_temp.elf Project_tem
>>> Project_temp.bin Checksum = 0xFF1F <<<
>>> Project_temp.bin CRC-8 Checksum = 0x68 <<<
    text      data      bss      total filename
    4584      131        0      4715 Project_temp.elf

15:27:27 Build Finished. 0 errors, 0 warnings. (took 2s.785ms)
  
```

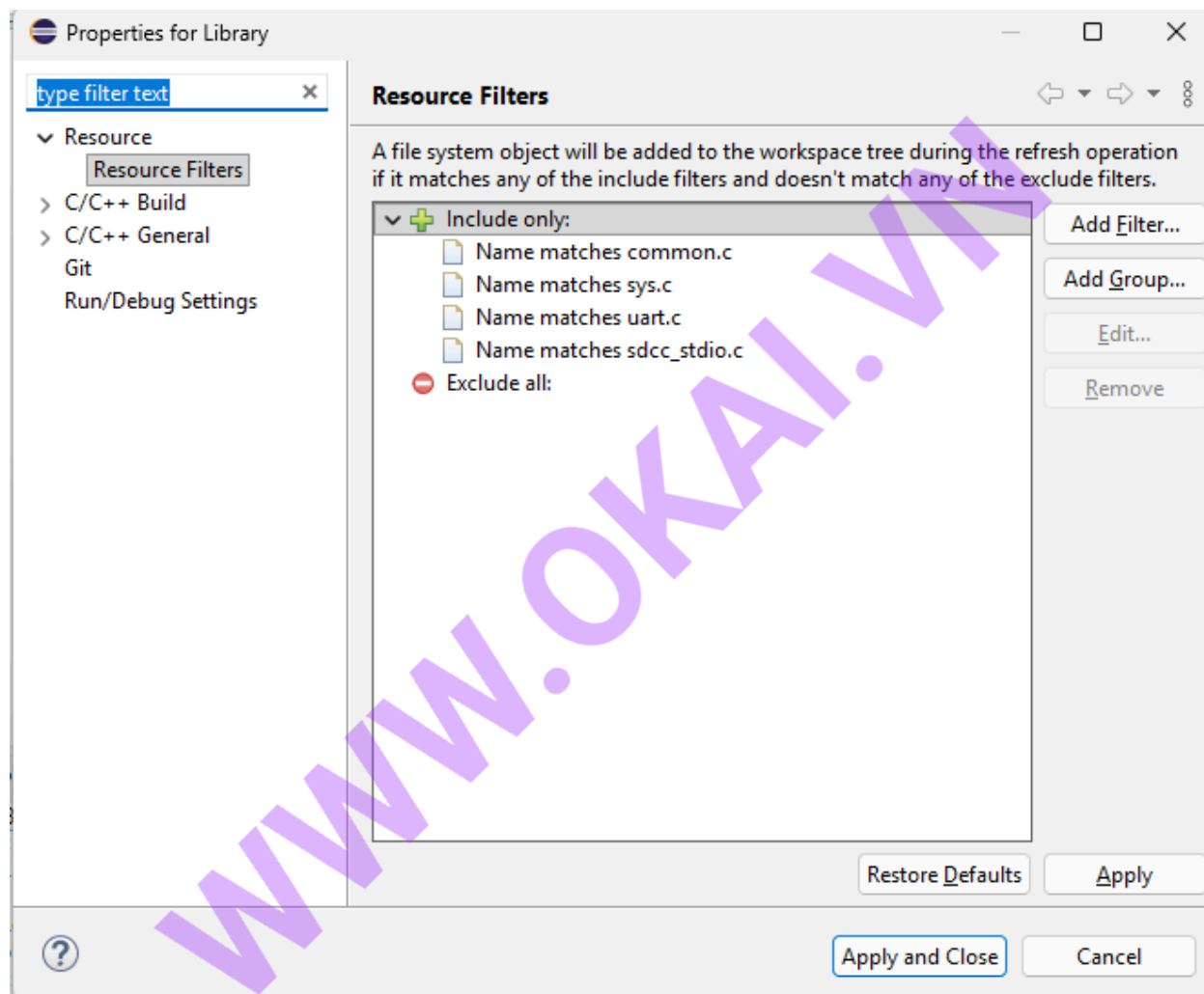
Như đã nói ở trên “Release” là nơi chứa các file trong quá trình build khi biên dịch thành công file hex và bin sẽ nằm ở thư mục này



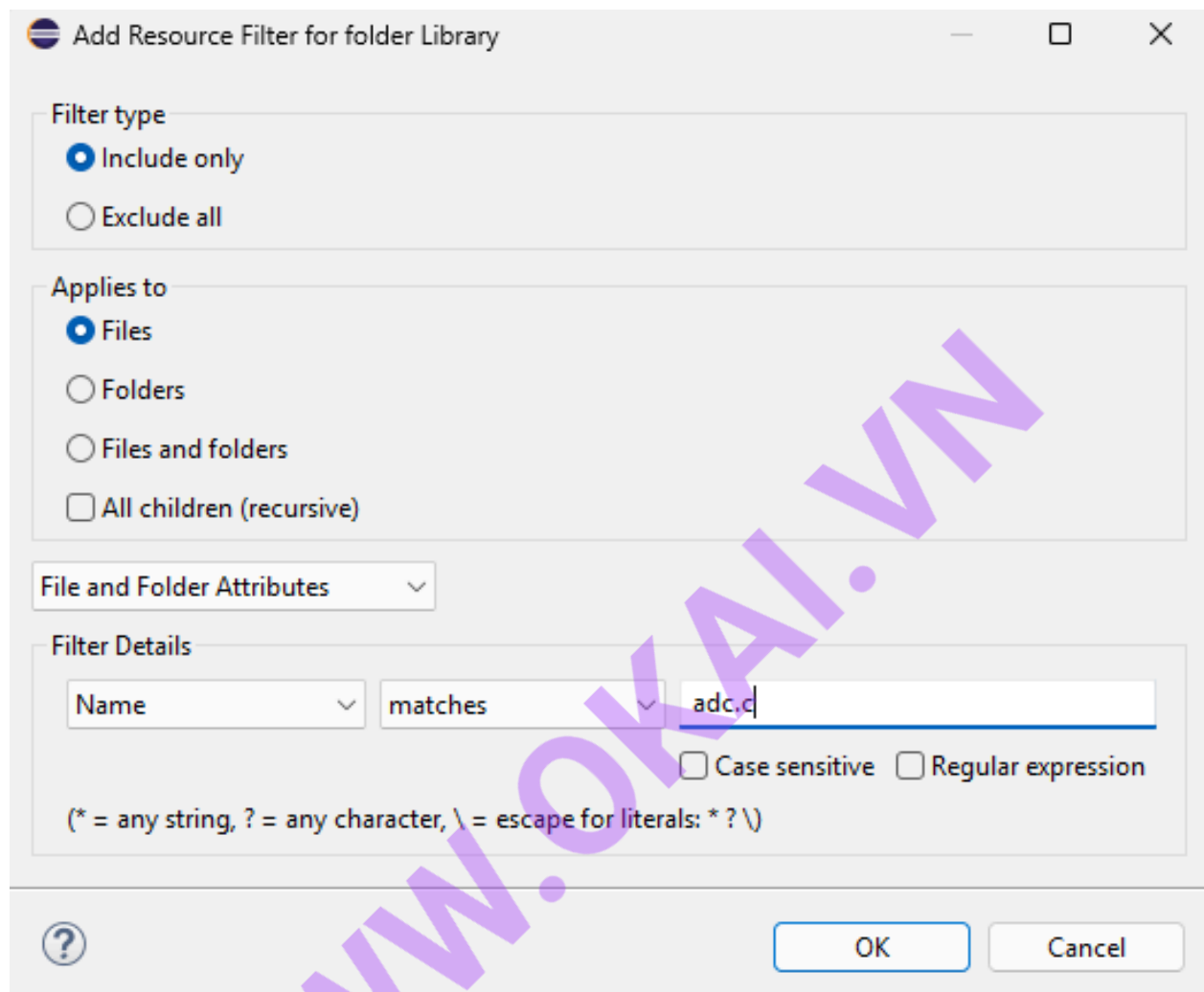
Bạn có thể xem cách nạp code ở mục [3.1.2](#). Bạn có thể lấy đây là một dự án gốc nhân bản ra để dùng cho dự án cá nhân của mình.

3.2.3 Thêm thư viện

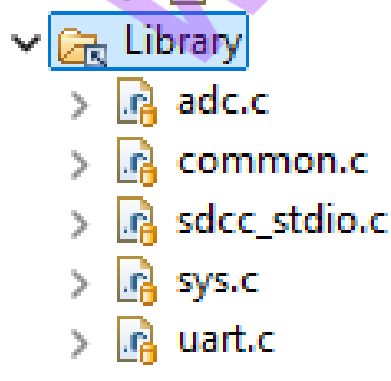
Click chuột vào folder của library ấn phím tắt “Alt + Enter” để vào “properties for library” tại mục “Resource filter” chứa các thư viện sẽ được khai báo khi biên dịch



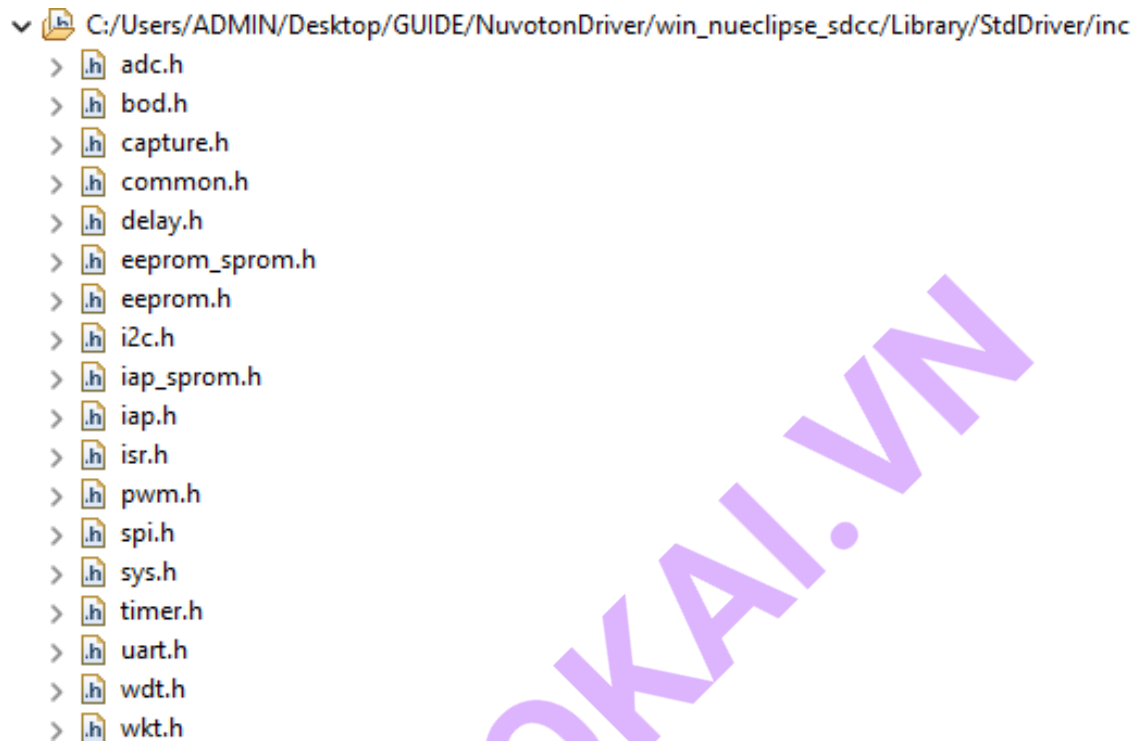
Click vào “Add Filter...” ví dụ ở đây ta muốn thêm thư viện “adc.c”



Thư viện “adc.c” sẽ được thêm vào “library”

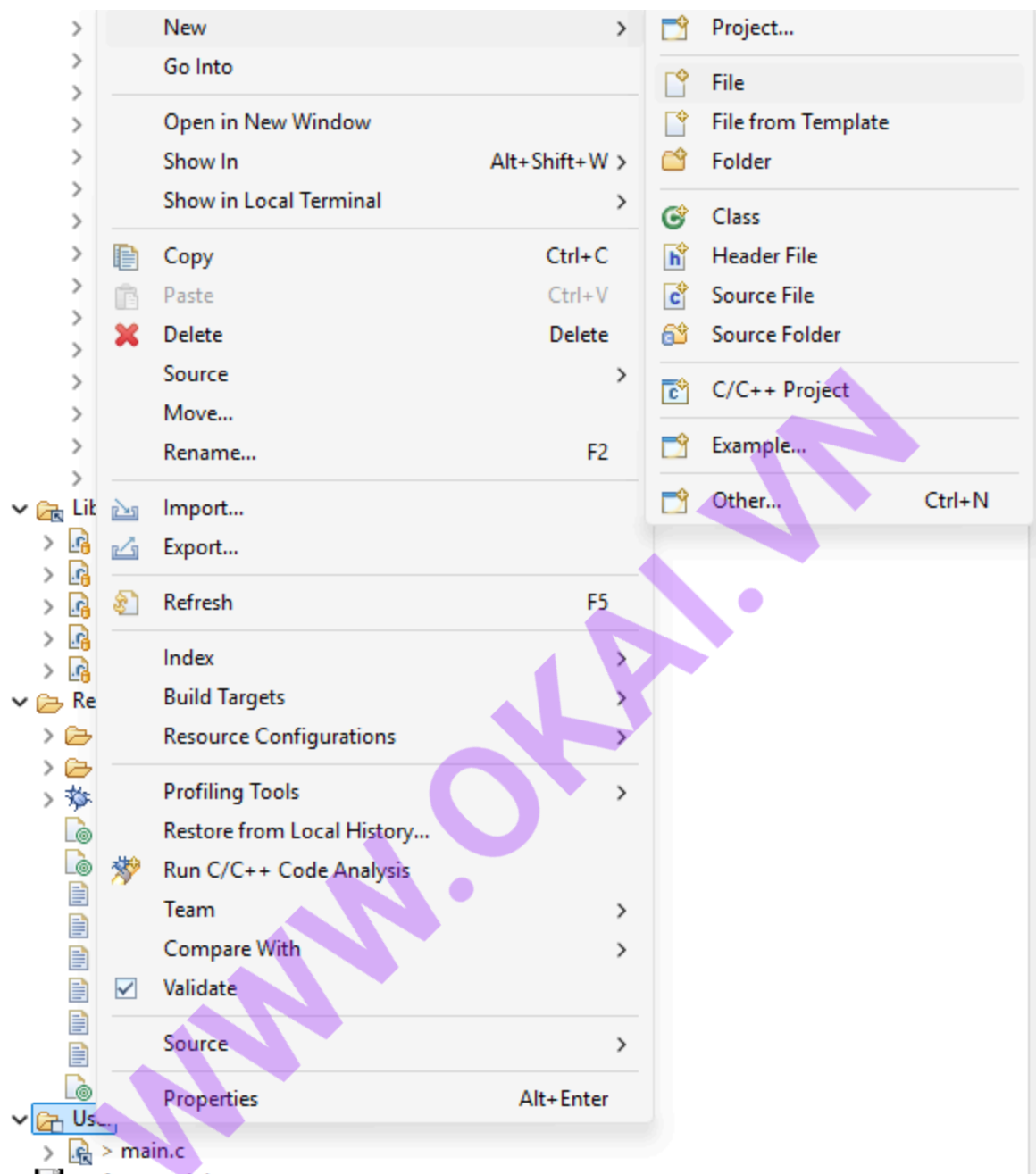


Bạn có thể dựa vào mục header của dự án để thêm các thư viện bạn cần
dưới đây là tất cả thư viện mẫu bạn có thể add vào dự án của mình

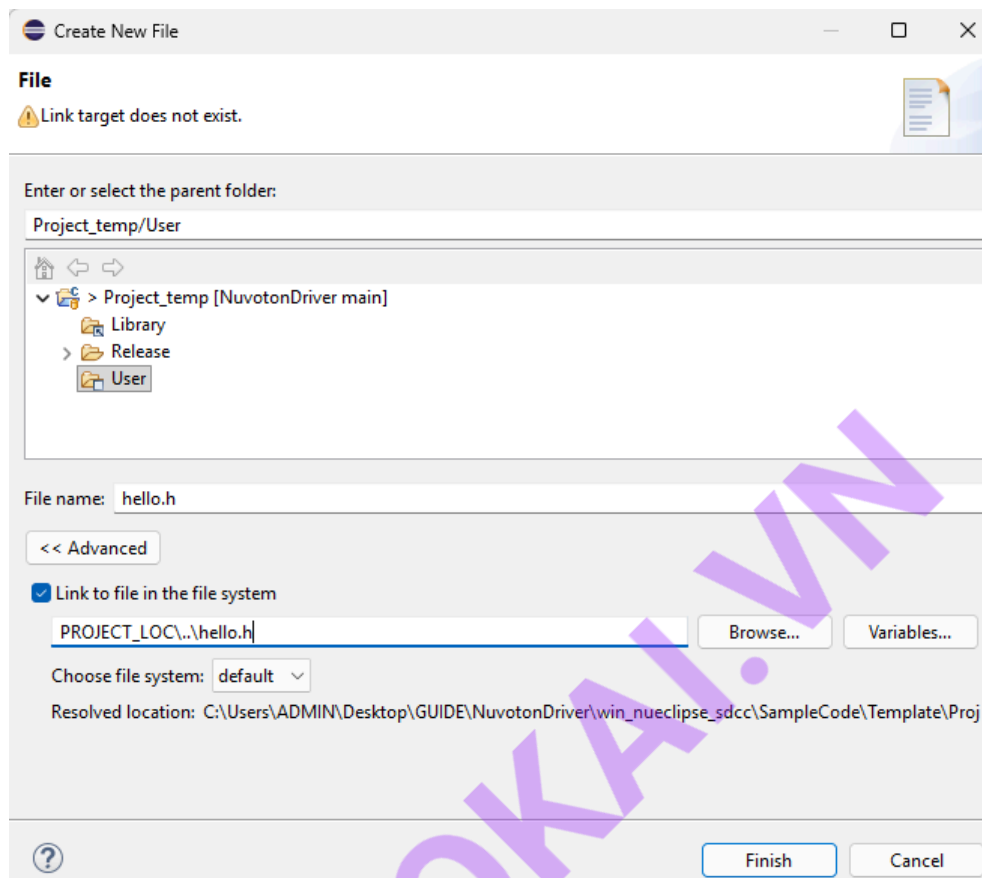


3.2.4 Tạo thư viện

Tại mục “Project Explorer” chuột phải vào folder “User” chọn
“New”->“File”



Click vào “Advanced” tick chọn “Link to file in the file system”. Ví dụ tạo file hello.h bạn viết như sau “PROJECT_LOC\..\hello.h” như hình dưới đây



Sau đó ấn “Fish” nếu có cửa sổ nào hiện ra ấn “Yes”



File header hello.h đã được thêm vào folder “User”.

3.2.5 Debug

Tính năng này hiện chưa được hỗ trợ tốt cho chip 8051 của hãng Nuvoton nên tác giả chỉ giới thiệu có tính năng này trên Nu eclipse nếu bạn quan tâm có thể liên hệ với tác giả.

4. Một số lỗi thường gặp

1. Vấn đề hiển thị bug trên vscode

Hiện tượng: build code thành công nhưng vẫn hiển thị bug

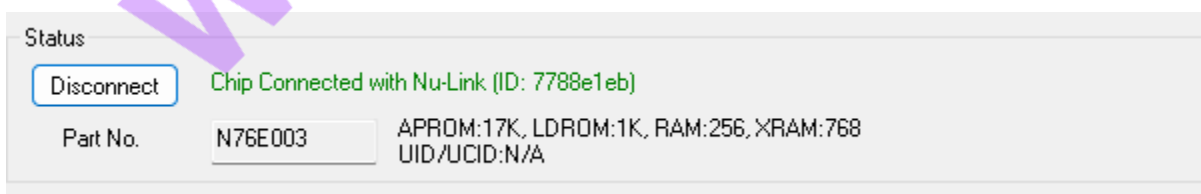
```
int putchar (int c) {  
    while (!TI);  
    TI = 0;  
    SBUF = c;  
    return c;  
}
```

Giải pháp: tìm đến thư mục ".vscode\settings.json" trong dự án disabled thuộc tính C_Cpp.errorSquiggles

```
{  
    "C_Cpp.errorSquiggles": "disabled"  
}
```

2. MCU tự reset

Hiện tượng: chương trình chạy nhưng sau khoảng 0.5s lại reset



Giải pháp: Lỗi này có thể do chưa tắt connect ở NuMicro ICP Programming Tool. Trước khi chạy đảm bảo rút mạch nạp hoặc ngắt kết nối với mạch