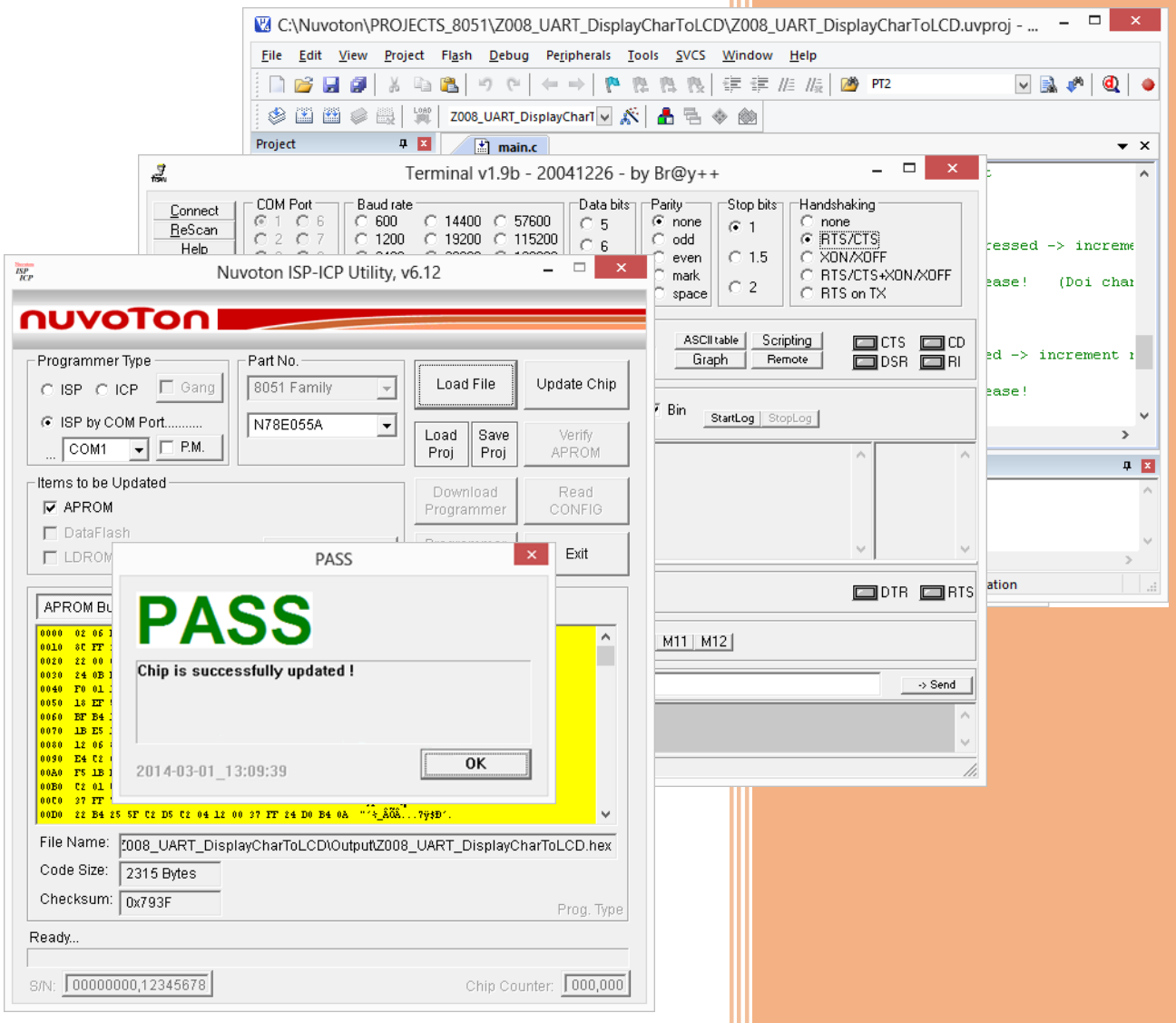
TULA Solution

The Distributor of Quality

2014

nuvoTon

Hướng dẫn phát triển N78E055A



tranthangTMT

TULA Solution

20/03/2014

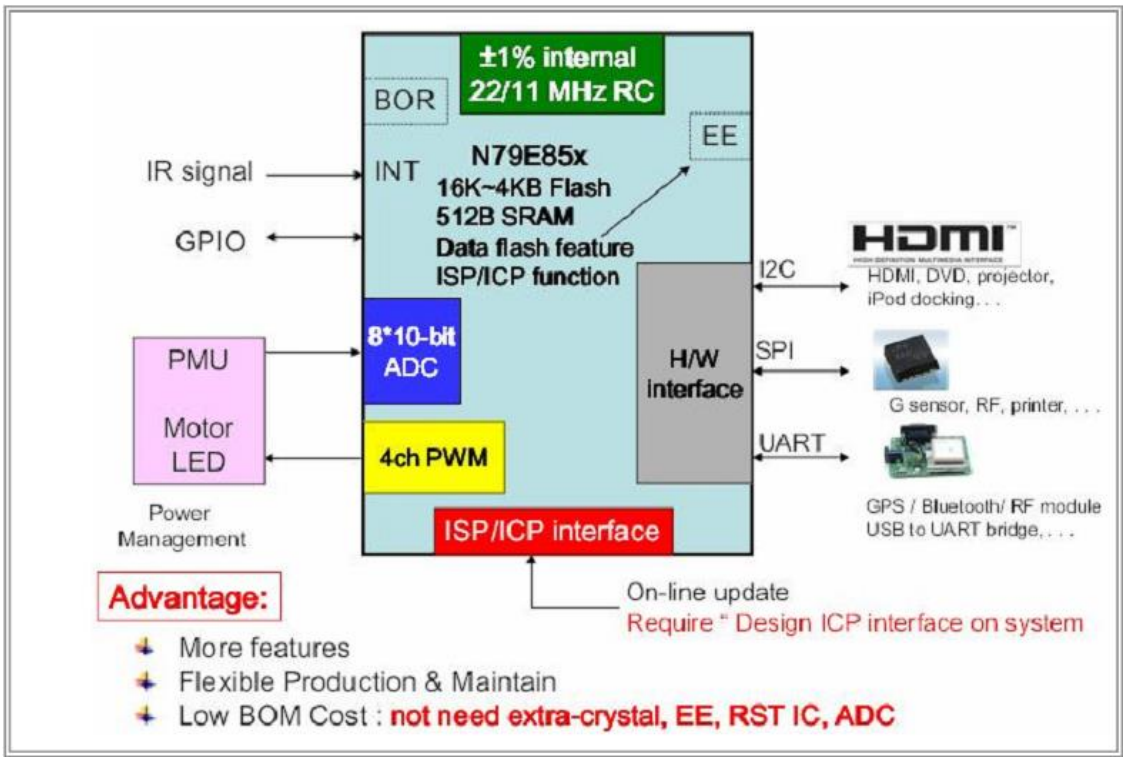
MỤC LỤC

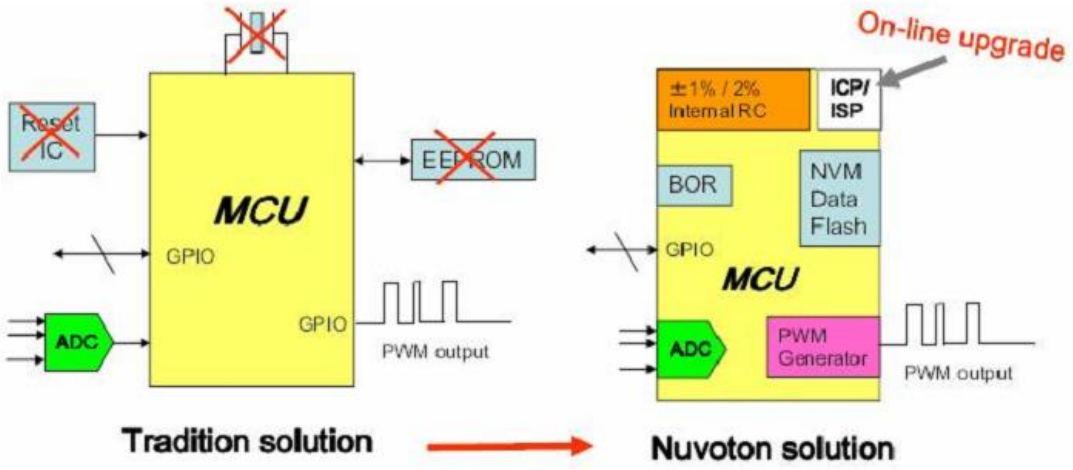
HƯỚNG DẪN PHÁT TRIỂN N78E055A

I. GIỚI THIỆU N78E055A

1. Tổng quan

- N78E055A là dòng vi điều khiển của hãng Nuvoton sử dụng lõi 80C51.
- N78E055A có thể thay thế cho MCU 8051 của các hãng khác như ATMEL, NXP,..
- So với các MCU nền 8051 thì N78E055A mang những tính năng kỹ thuật hiện đại và nổi trội hơn như mật độ tích hợp, có thạch anh nội tốc độ cao, hỗ trợ nhiều kiểu nạp code, kháng nhiễu, chống ồn, giá rẻ. Ta có thể xem ở phần tiếp theo về tính năng của N78E055A:
- Lợi thế của dòng MCU-8bit lõi 8051 của Nuvoton so với truyền thống:
    - Tổng BOM thấp
    - Dễ dàng phát triển: có đầy đủ tài nguyên phát triển.
    - Bảo trì và sản suất linh hoạt.

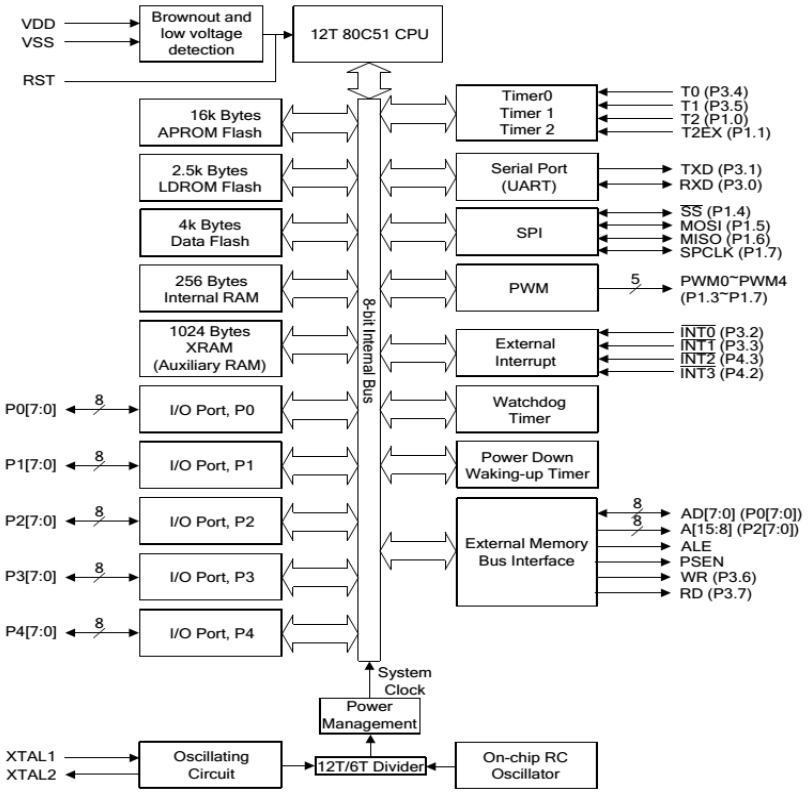




2. Tính năng nổi bật

- Dòng vi điều khiển 8bit chế tạo trên công nghệ CMOS với thiết kế chống nhiễu, tĩnh điện ESD/EFT.
- Lõi 80C51 12T/6T (chạy với 12 hoặc 6 xung dao động cho chu kỳ máy).
- Dải áp hoạt động rộng: 2.4 ~ 5.5V và tần số chạy rộng từ 4 ~ 40MHz.
- Tích hợp bộ dao động nội RC 22.1184MHz/ 11.0592MHz với sai số ±1% ở nhiệt độ 25°C và ±3% cho toàn dải nhiệt độ -40°C ~ +85°C.
- Về bộ nhớ Flash: 16KB APROM, 2.5KB ISPROM, 4KB DataFlash, với khả năng ghi xóa 10.000 lần và dữ liệu còn nguyên vẹn trên 10 năm tại nhiệt độ dưới +85°C
- Về RAM có 255B, bổ sung thêm 1KB XRAM (dùng các lệnh MOVX để truy cập)
- Ngoài ra, còn hỗ trợ giao diện Bus bộ nhớ ngoài (Programe/Data) quản lý tới 64KB.
- Hỗ trợ nạp ISP được với dải điện áp rộng 3.0 ~ 5.0V.
- Hỗ trợ bảo mật dữ liệu.
- Ngoại vi có:
    - 3 x Timer-16bit
    - 5 x 3 Bộ đếm / Bộ định thời 16 bit.
    - 1 x UART song công
    - 1 x SPI
    - 5 kênh ra PWM
    - 11 nguồn ngắt, 4 mức ưu tiên cho mỗi nguồn
    - 4 x Ngắt ngoài INT (dạng chân DIP40 chỉ có INT1 và INT2)
    - WDT, POR, Hỗ trợ Reset mềm, 4 mức BOD
    - Có bộ quản lý nguồn cho chế độ Idle và Power-Down tiết kiệm điện.
- Dạng đóng gói: DIP-40, PLCC4-4, PQFP-44, LQFP-48

3. Sơ đồ chối chức năng



Hình: Sơ đồ khối chức năng của N78E055A

II. TÀI NGUYÊN PHÁT TRIỂN

1. Tài liệu

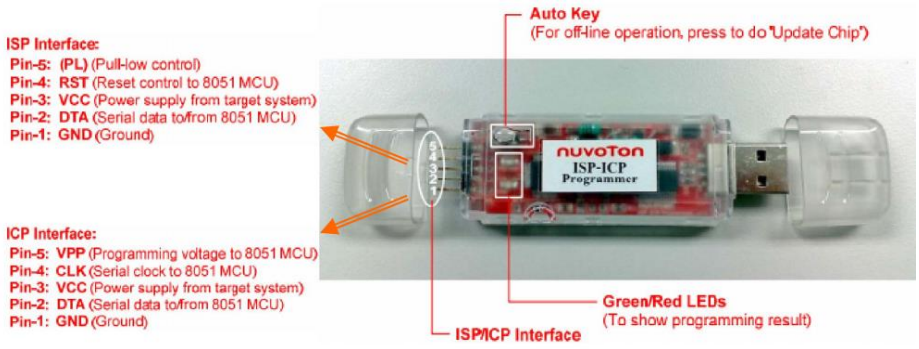
- Toàn bộ tài nguyên phát triển của N78E055A có thể tải về từ website của hãng, gồm có: Công cụ phát triển: "Driver để thêm thư viện chip của Nuvoton cho Keil-uVision " + "Hướng dẫn sử dụng cho mạch nạp chip MCU 8051 của Nuvoton" + "Phần mềm điều kiển nạp ISP-ICP cho chip MCU 8051 của Nuvoton".
  http://www.nuvoton.com/NuvotonMOSS/Community/ProductInfo.aspx?tp_GUID=670aaf31-5d5c-45d3-8a9e-040e148d55cf
- Datasheet của N78E055A:
  http://www.keil.com/dd/docs/datashts/nuvoton/n78e055a.pdf
- Mã nguồn Demo: cho các dòng chip MCU 8051 của Nuvoton và mã trình ISP.
  http://www.nuvoton.com/NuvotonMOSS/Community/ProductInfo.aspx?tp_GUID=515275c4-f6ae-4915-89ce-366fd8097efc
- Các lưu ý ứng dụng (Application Note):
  http://www.nuvoton.com/NuvotonMOSS/Community/ProductInfo.aspx?tp_GUID=cb37b14c-6b06-4c6d-878f-0572f9befd73
- Bảng thay thế tương thích chip của Nuvoton thay thế cho các MCU LPC của NXP.
  http://www.nuvoton.com/NuvotonMOSS/Community/ProductInfo.aspx?tp_GUID=2d53cbaa-f858-4dfd-b6cb-597c6f467055
- Xem các tài liệu hướng dẫn tại trang web của các nhà phân phối của hãng Nuvoton.

2. Phần mềm

- N78E055A của Nuvoton tương tích với tập lệnh 8051 chuẩn (MCS-51) cho nên mọi môi trường phát triển, trình biên dịch, phần mềm mô phỏng dùng được cho chip MCU 8051 chuẩn thì đều dùng được cho chip dòng này; Ví dụ các phần mềm Keil C (uVision), IAR, Hi-Tech, SPKT-C (SPKT-8051),…
- Với thiết kế mạch (PCB) thì hoàn toàn tương tự như với dòng 8051 của các hãng khác. Các phần mềm thiết kế PCB thông dụng: Altium Designer, Orcad, Eagle, Proteus,…

3. Phần cứng

- Công cụ nạp (mạch nạp): N78E055A được hỗ trợ giải thuật nạp bởi hầu hết các máy nạp rom đa năng của các hãng sản xuất bộ nạp ROM nổi tiếng trên thế giới như Xeltek, Elnec, Hilosystems, Leap Electronics…
- Ngoài ra hãng Nuvoton cũng chế tạo riêng các mạch nạp có kích thước nhỏ gọn, rẻ tiền mà hỗ trợ khả năng nạp được toàn bộ các chip MCU lõi 8051 của hãng, giúp tiện dùng cả trong quá trình phát triển lẫn sản xuất. Mạch nạp Gang của Nuvoton cho sản xuất hàng loạt. Mạch nạp Stand-alone kiểu ISP/ICP:



- Lưu ý: N78E055A hỗ trợ nạp ISP qua cổng COM nên rất tiện lợi, mạch nạp chi phí thấp, dễ dàng cho tiếp cận và phát triển ứng dụng.

4. Hướng dẫn phát triển nhanh

Bước 1: Download Keil C bản chuẩn từ trang chủ của hãng

https://www.keil.com/c51/demo/eval/c51.htm

Bước 2 : Tải các dữ liệu từ trang Công cụ phát triển của hãng Nuvoton:

- Driver của Nuvoton N78E055A để thêm thư viện cho Keil C.
- Mã nguồn và chương trình mẫu.
- Phần mềm nạp cho N78E055A của Nuvoton ISP-ICP Programmer (phần mềm hỗ trợ nạp ICP, ISP bằng mạch nạp của hãng hoặc nạp qua cổng COM).

Bước 3 : Lập trình firmware và thiết kế mạch cho chip N78E055A hệt như với dòng 8051 của hãng khác (Atmel, NXP, STC…).

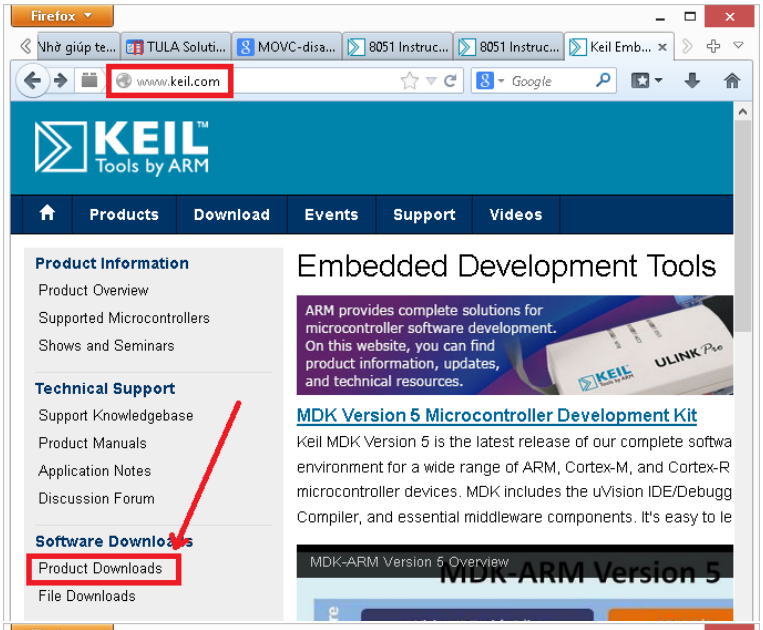
Bước 4 : Nạp dữ liệu (firmware & data) cho chip N78E055A của Nuvoton:

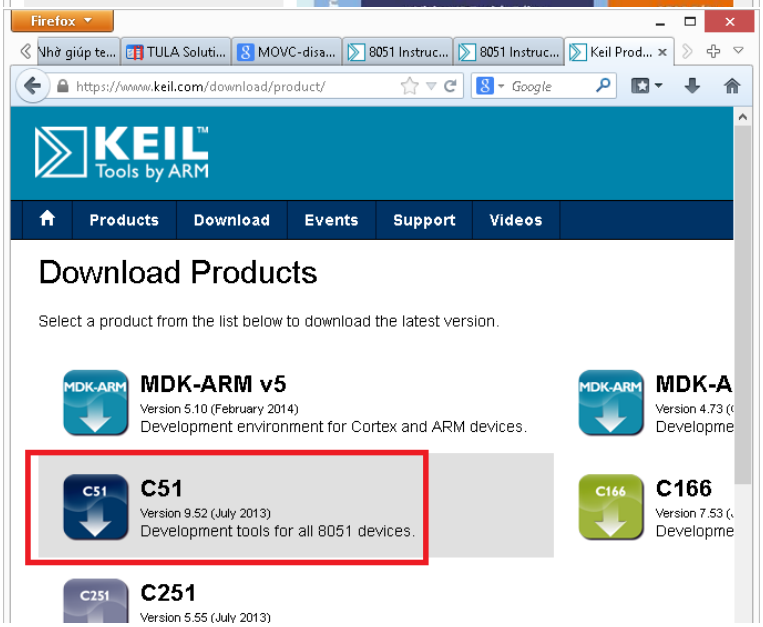
- Có thể nạp bằng mạch nạp của chính hãng Nuvoton hoặc mạch nạp đa năng, mạch nạp chuyên dùng của các hãng khác (Xeltek, Elnec,…).
- Cách nạp ICP, ISP (on-board) bằng mạch nạp của Nuvoton: thực hiện kết nối tương ứng đúng như tài liệu hướng dẫn của hãng Nuvoton (). Thực thi phần mềm nạp của Nuvoton và chọn cấu hình kiểu nạp (ICP hay ISP), chọn mã chip và các tham số tương ứng để nạp cho chip.
- Cách nạp ISP qua cổng UART của N78E055A với cổng COM của PC: Thực hiện kết nối kiểu nạp ISP cổng COM giống hệt như nạp cho với các dòng 8051 của các hãng khác (89S52 của ATMEL chẳng hạn), khi này dùng các chân Pin 3.1 và Pin 3.0 của MCU giao tiếp RS-232 với PC. Thực thi phần mềm nạp của Nuvoton và chọn cấu hình kiểu nạp ISP cổng COM, chọn mã chip và các tham số tương ứng để nạp cho chip. Lưu ý: nếu máy tính không có cổng COM thì có thể sửdụng cáp chuyểnđổi “USB to COM” để nạp. Khi đó cổng COM ảo sẽ được tạo ra trên máy tính để giao tiếp được qua cổng USB như cổng COM thông thường).

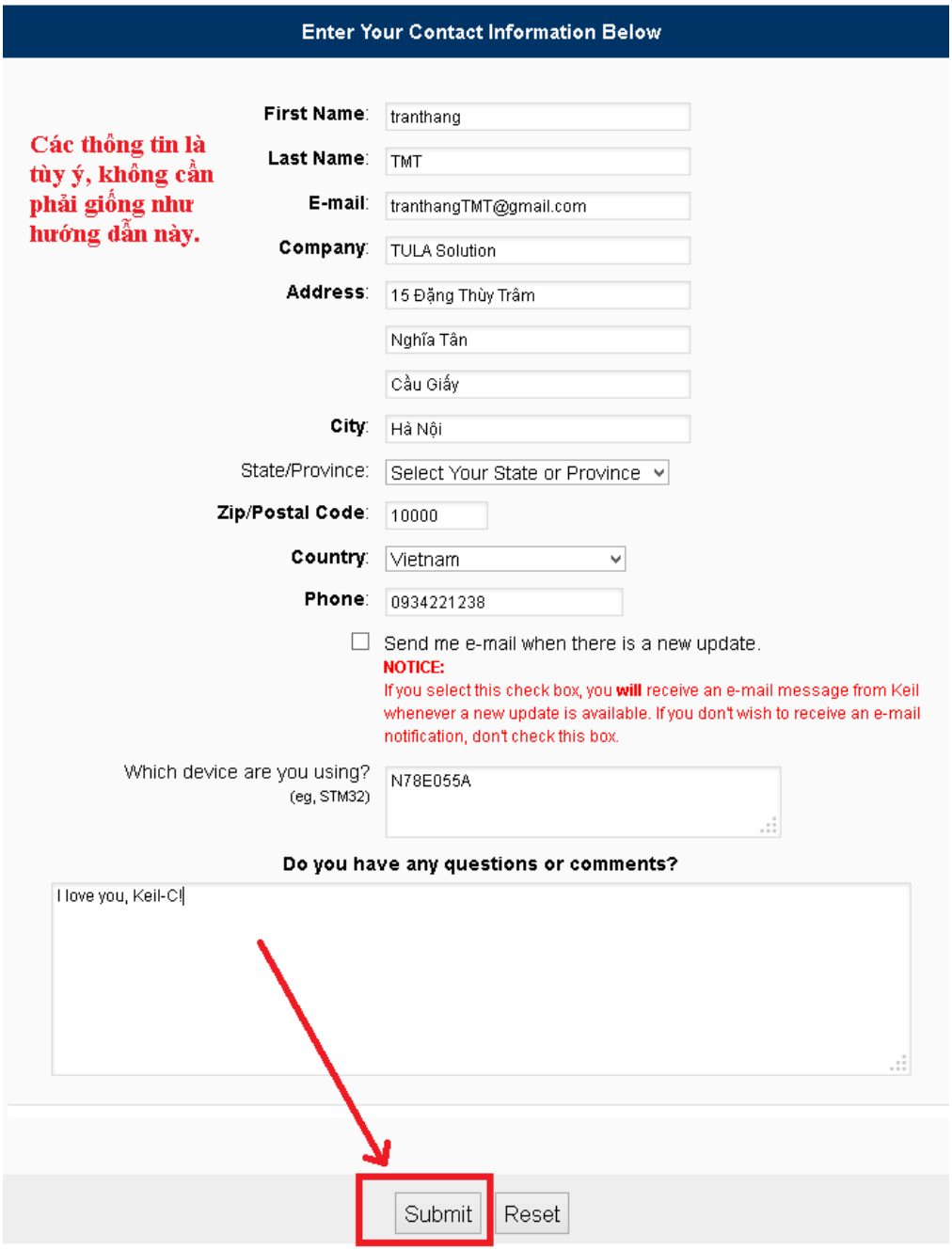
III. SỬ DỤNG KEIL-µVISION PHÁT TRIỂN N78E055A

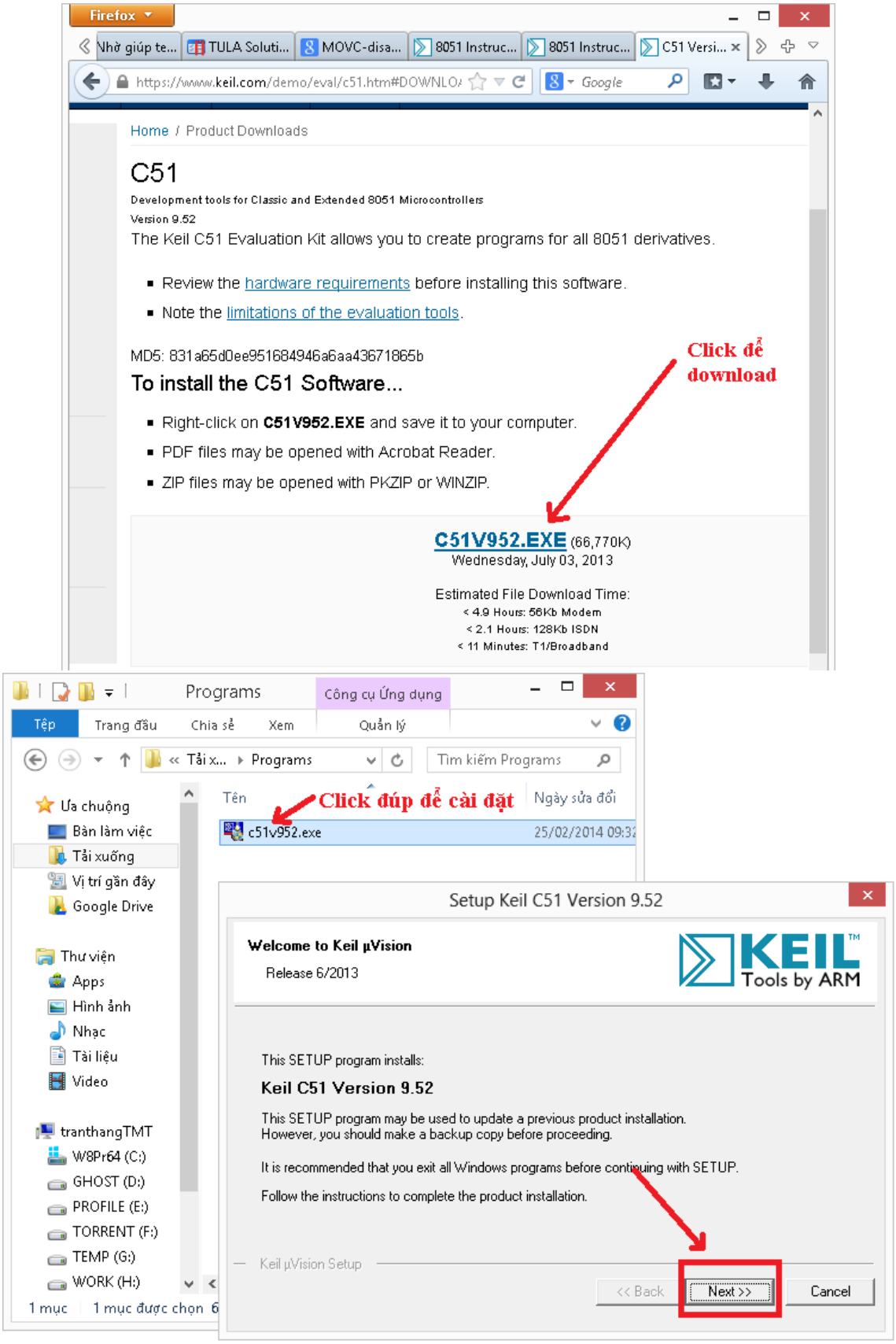
1. Cài Keil-uVision

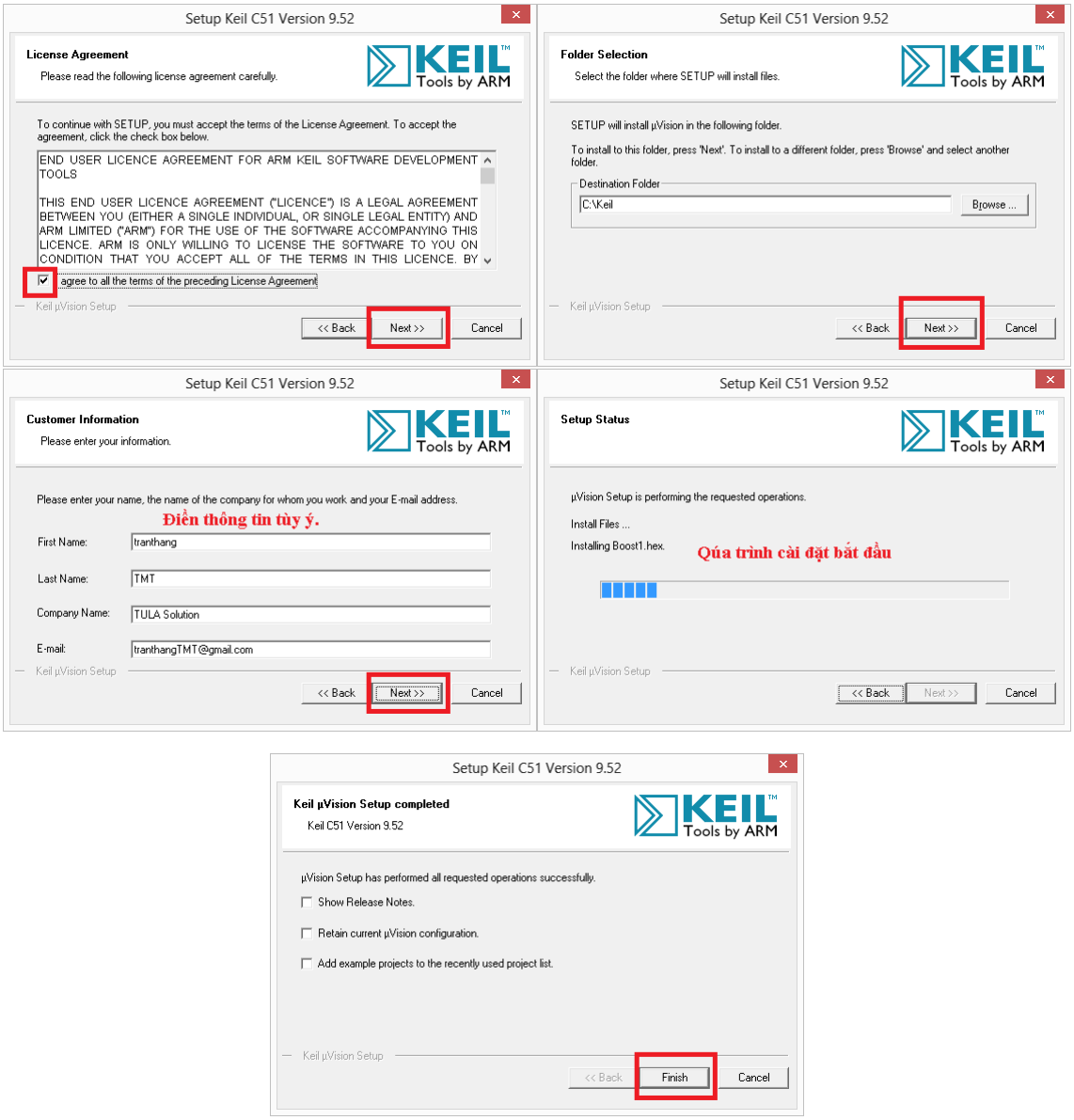
- Ta có thể download từ trên các diễn đàn điện tử hoặc từ trang chủ:
- Download Phần mềm … từ trang chủ (www.keil.com) hoặc từ những trang chia sẽ phần mềm trên Internet (sử dụng google để tìm kiếm). Ở đây, hướng dẫn download từ trang chủ:



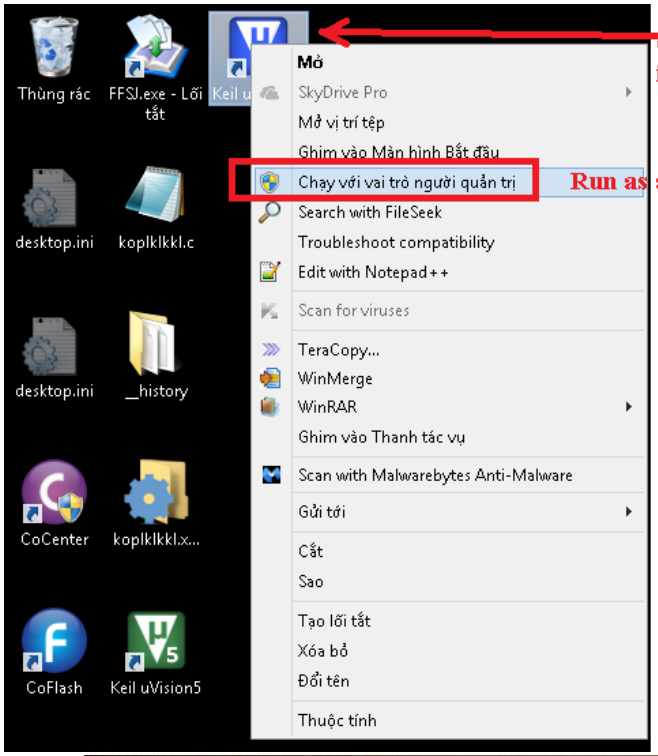



Enter Your Contact Information Below
Các thông tin là tùy ý, không cần phải giống như hướng dẫn này.
First Name: tranthang
Last Name: TMT
E-mail: tranthangTMT@gmail.com
Company: TULA Solution
Address: 15 Đặng Thùy Trâm
Nghĩa Tân
Cầu Giấy
City: Hà Nội
State/Province: Select Your State or Province
Zip/Postal Code: 10000
Country: Vietnam
Phone: 0934221238
Send me e-mail when there is a new update.
NOTICE:
If you select this check box, you will receive an e-mail message from Keil whenever a new update is available. If you don't wish to receive an e-mail notification, don't check this box.
Which device are you using? (eg, STM32)
N78E055A
Do you have any questions or comments?
I love you, Keil-C!
Submit
Reset


Firefox
https://www.keil.com/demo/eval/c51.htm#DOWNLO
Home / Product Downloads
C51
Development tools for Classic and Extended 8051 Microcontrollers
Version 9.52
The Keil C51 Evaluation Kit allows you to create programs for all 8051 derivatives.
Review the hardware requirements before installing this software.
Note the limitations of the evaluation tools.
MD5: 831a65d0ee951684946a6aa43671865b
To install the C51 Software...
Right-click on C51V952.EXE and save it to your computer.
PDF files may be opened with Acrobat Reader.
ZIP files may be opened with PKZIP or WINZIP.
Click để download
C51V952.EXE (66,770K)
Wednesday, July 03, 2013
Estimated File Download Time:
< 4.9 Hours: 56Kb Modem
< 2.1 Hours: 128Kb ISDN
< 11 Minutes: T1/Broadband
Programs
Công cụ Ứng dụng
Tệp
Trang đầu
Chia sẻ
Xem
Quản lý
Tải x... ▸ Programs
Tìm kiếm Programs
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
GHOST (D:)
PROFILE (E:)
TORRENT (F:)
TEMP (G:)
WORK (H:)
1 mục
1 mục được chọn
Tên
Ngày sửa đổi
Click đúp để cài đặt
c51v952.exe
25/02/2014 09:3
Setup Keil C51 Version 9.52
Welcome to Keil µVision
Release 6/2013
KEIL™ Tools by ARM
This SETUP program installs:
Keil C51 Version 9.52
This SETUP program may be used to update a previous product installation.
However, you should make a backup copy before proceeding.
It is recommended that you exit all Windows programs before continuing with SETUP.
Follow the instructions to complete the product installation.
Keil µVision Setup
<< Back
Next >>
Cancel

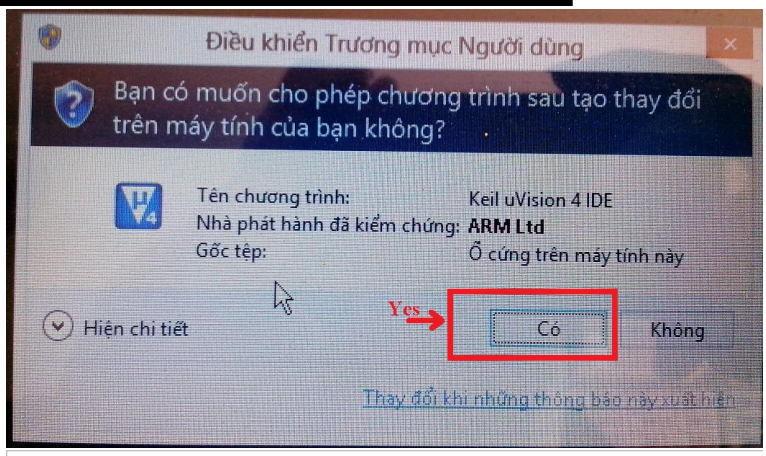


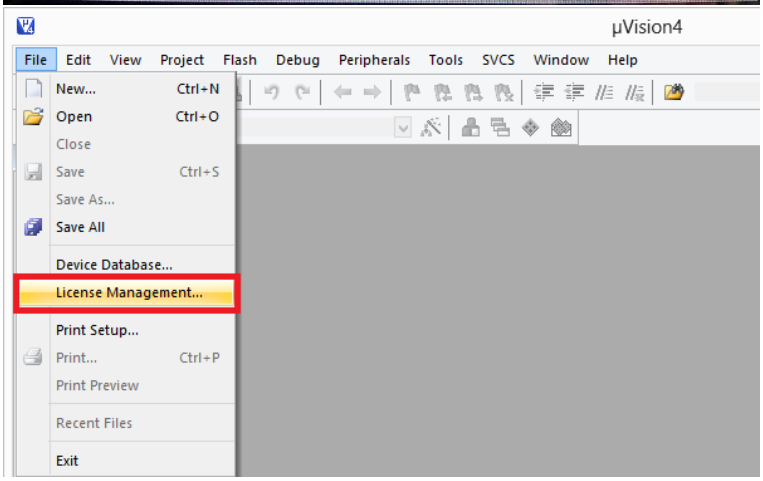
- Add licese cho Keil MDK v4: Với bản được download từ trang chủ của Keil khi ta cài đặt xong thì sẽ ở dạng bản MDK-ARM Lite Edition với những giới hạn:
  + Chương trình trên 32Kbytes (code và data) sẽ không biên dịch, ghép nối,..
  + Việc gỡ rối cũng chỉ giới hạn trong 32Kbytes trở xuống.
  + Trình biên dịch không tạo ra một danh sách tháo gỡ của các mã máy tạo ra.
  + Và một số giới hạn khác có thể xem thông tin chi tiết tại:
    http://www.keil.com/demo/limits.asp
- Với việc nghiên cứu, học tập về N78E055A (chỉ có 16KB APROM) thì bản MDK-ARM Lite Edition có thể đáp ứng nhu cầu của người học.
- Để có những tính năng cao cấp hơn ta có thể mua giấy phép bản quyền của Keil theo hướng dẫn cụ thể tại:
  http://www.keil.com/support/man/docs/license/license_management.htm
- Trực quan hơn, theo hướng dẫn này:

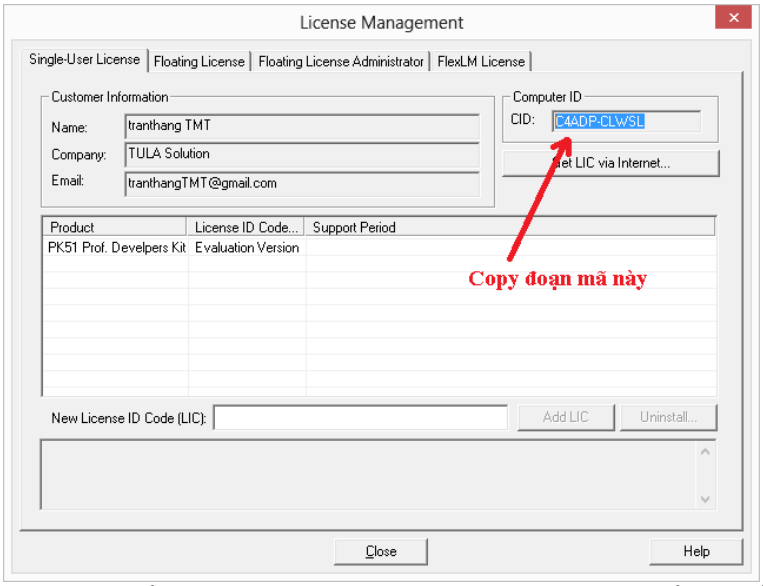


Click chuột phải vào icon của chương trình.

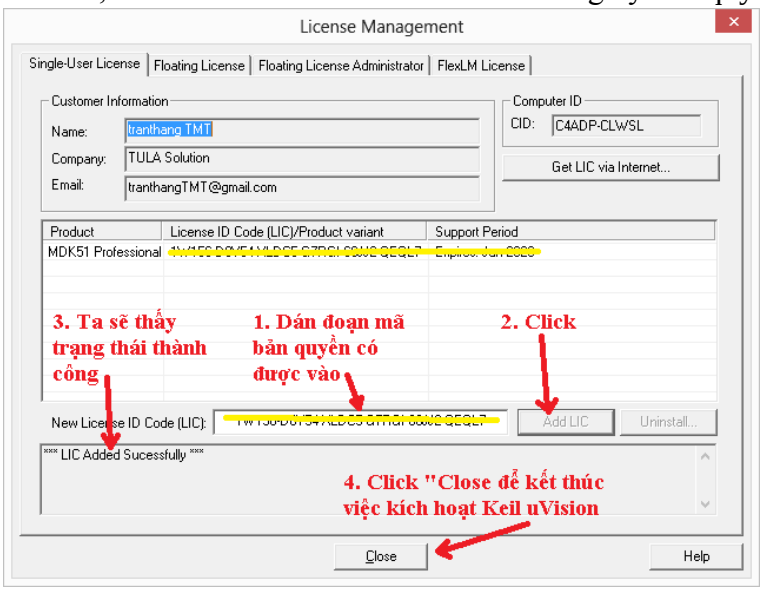
Run as administrator





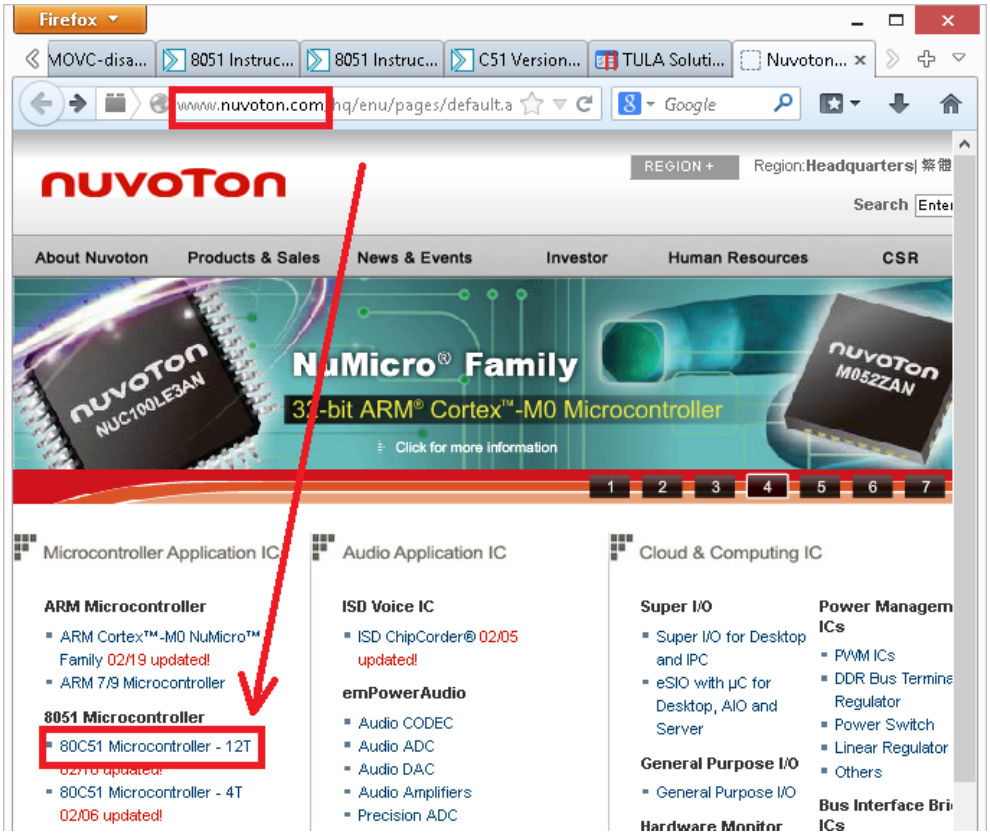


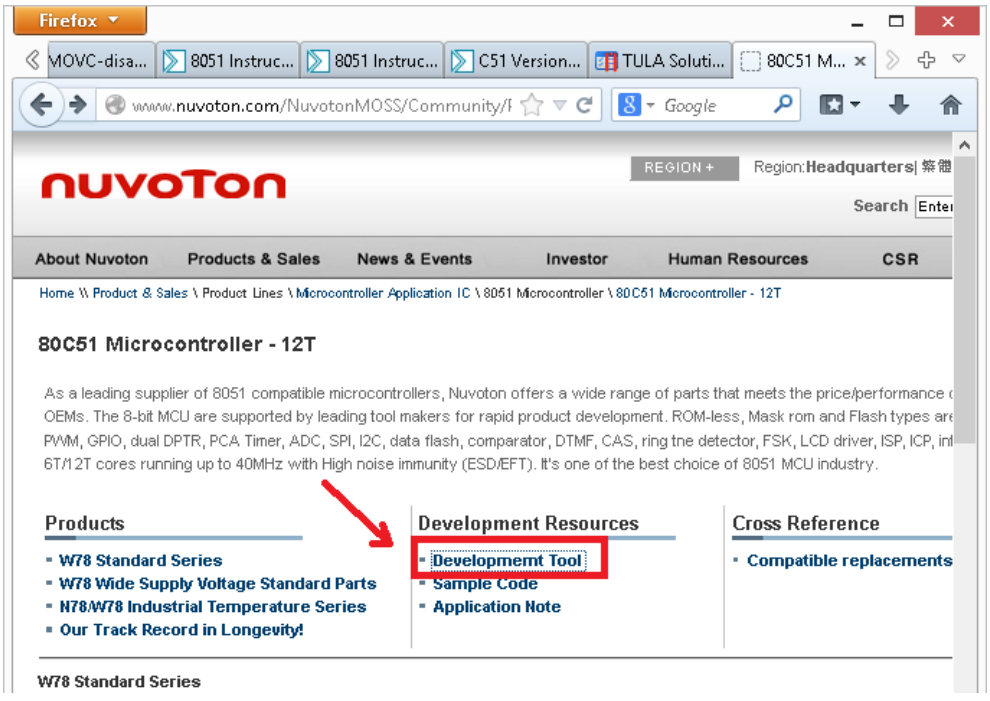
- Sau đó, bạn dùng mã này để liên hệ với bộ phận bán hàng của keil để có giấy phép: https://www.keil.com/company/contact/
- Hoặc để dễ hiểu hơn, có thể lên internet tìm kiếm cách đăng ký bản quyền.

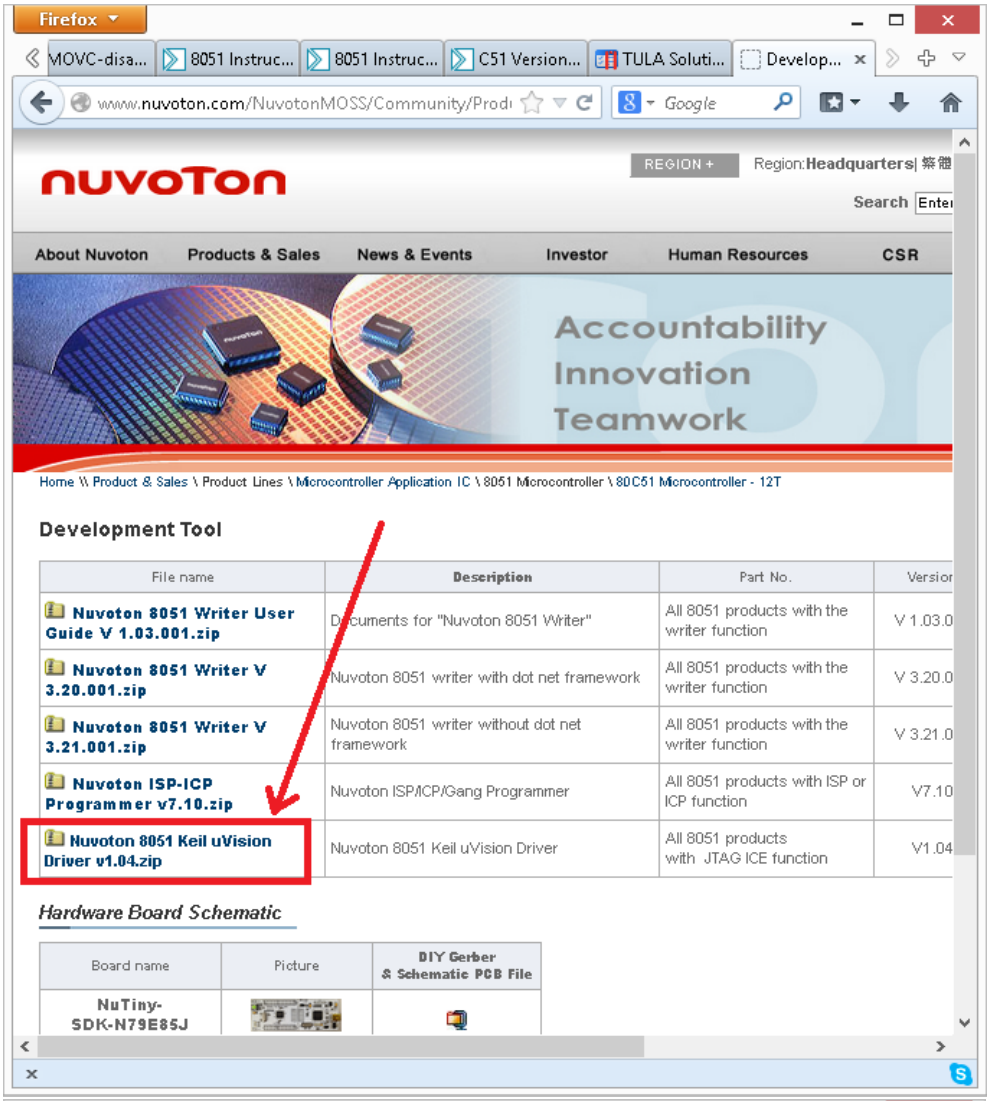


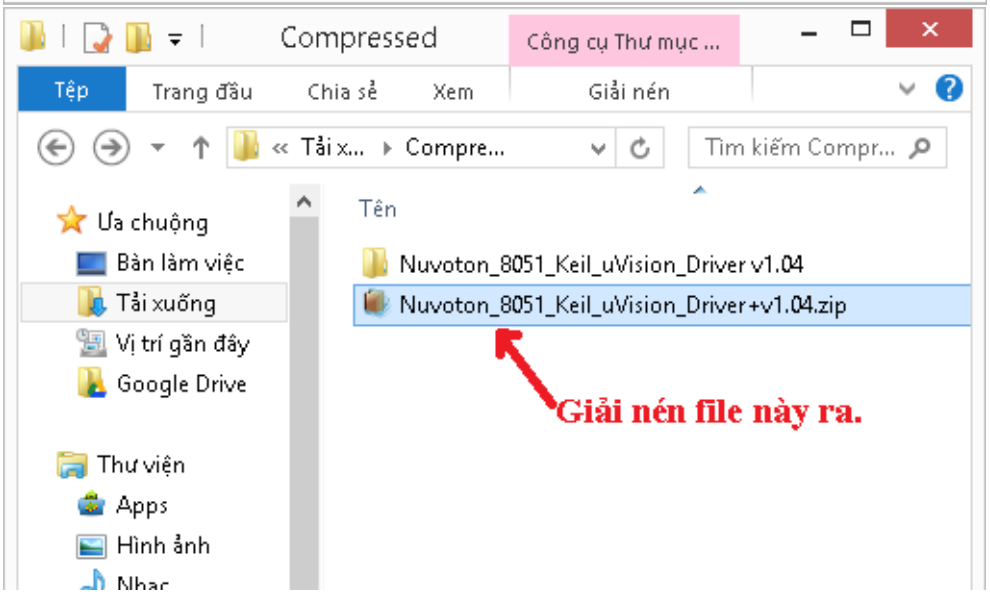
2. Cài thư viện phát triển N78E055A

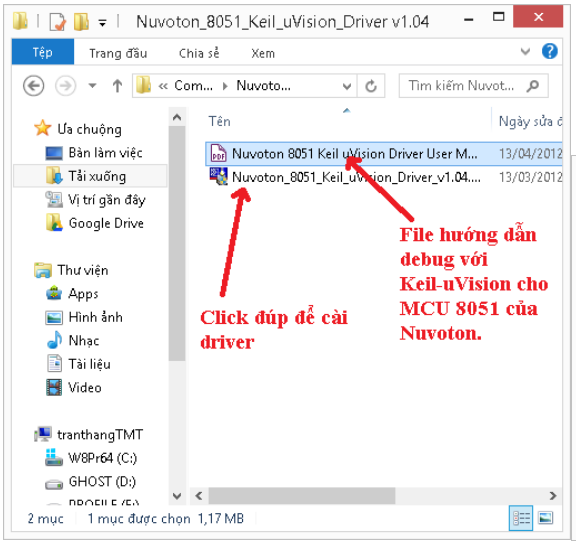
- Download Driver và thư viện phát triển N78E055A

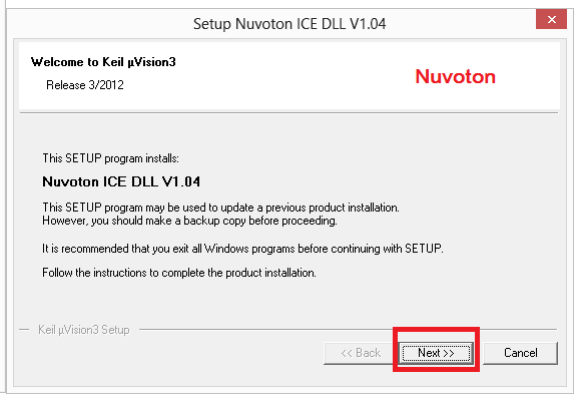


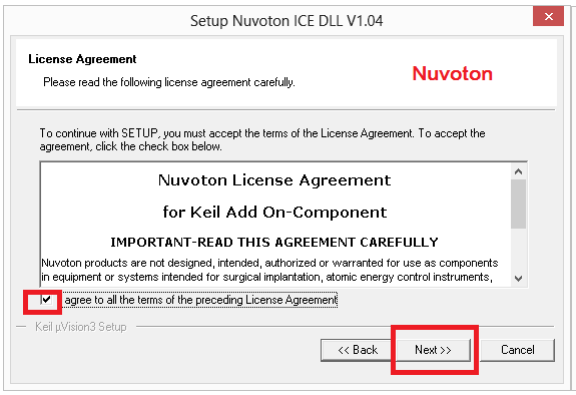


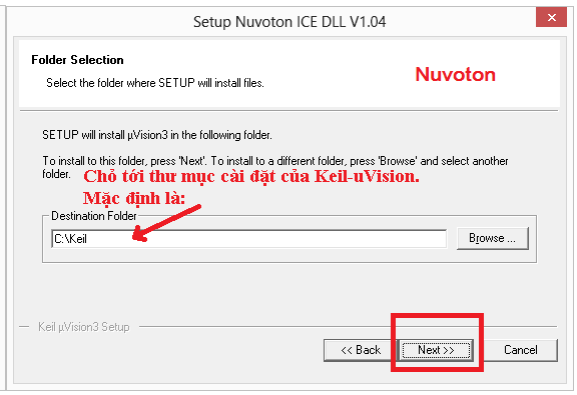

Firefox
www.nuvoton.com/NuvotonMOSS/Community/Prodi
nuvoTon
About Nuvoton
Products & Sales
News & Events
Investor
Human Resources
CSR
Accountability
Innovation
Teamwork
Home \ Product & Sales \ Product Lines \ Microcontroller Application IC \ 8051 Microcontroller \ 80C51 Microcontroller - 12T
Development Tool
File name
Description
Part No.
Nuvoton 8051 Writer User Guide V 1.03.001.zip
Documents for "Nuvoton 8051 Writer"
All 8051 products with the writer function
Nuvoton 8051 Writer V 3.20.001.zip
Nuvoton 8051 writer with dot net framework
Nuvoton 8051 Writer V 3.21.001.zip
Nuvoton 8051 writer without dot net framework
Nuvoton ISP-ICP Programmer v7.10.zip
Nuvoton ISP/ICP/Gang Programmer
All 8051 products with ISP or ICP function
Nuvoton 8051 Keil uVision Driver v1.04.zip
Nuvoton 8051 Keil uVision Driver
All 8051 products with JTAG ICE function
Hardware Board Schematic
Board name
Picture
DIY Gerber & Schematic PCB File
NuTiny-SDK-N79E85J

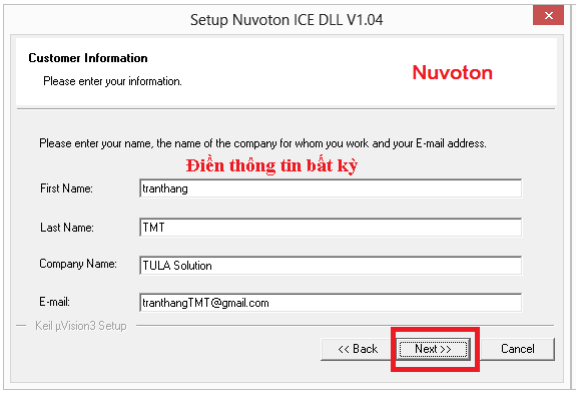

Compressed
Công cụ Thư mục ...
Tệp
Trang đầu
Chia sẻ
Xem
Giải nén
Tải x...
Compre...
Tìm kiếm Compr...
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Tên
Nuvoton_8051_Keil_uVision_Driver v1.04
Nuvoton_8051_Keil_uVision_Driver+v1.04.zip
Giải nén file này ra.

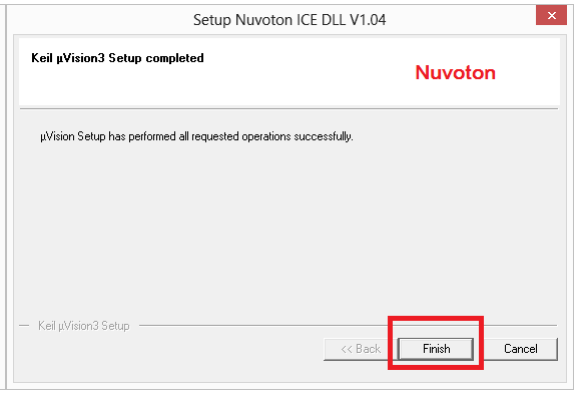

Nuvoton_8051_Keil_uVision_Driver v1.04
Tệp
Trang đầu
Chia sẻ
Xem
Tên
Ngày sửa đ
Nuvoton 8051 Keil uVision Driver User M... 13/04/2012
Nuvoton_8051_Keil_uVision_Driver_v1.04.... 13/03/2012
File hướng dẫn debug với Keil-uVision cho MCU 8051 của Nuvoton.
Click đúp để cài driver
2 mục
1 mục được chọn 1,17 MB


Setup Nuvoton ICE DLL V1.04
Welcome to Keil µVision3
Release 3/2012
Nuvoton
This SETUP program installs:
Nuvoton ICE DLL V1.04
This SETUP program may be used to update a previous product installation.
However, you should make a backup copy before proceeding.
It is recommended that you exit all Windows programs before continuing with SETUP.
Follow the instructions to complete the product installation.
Keil µVision3 Setup
<< Back
Next >>
Cancel

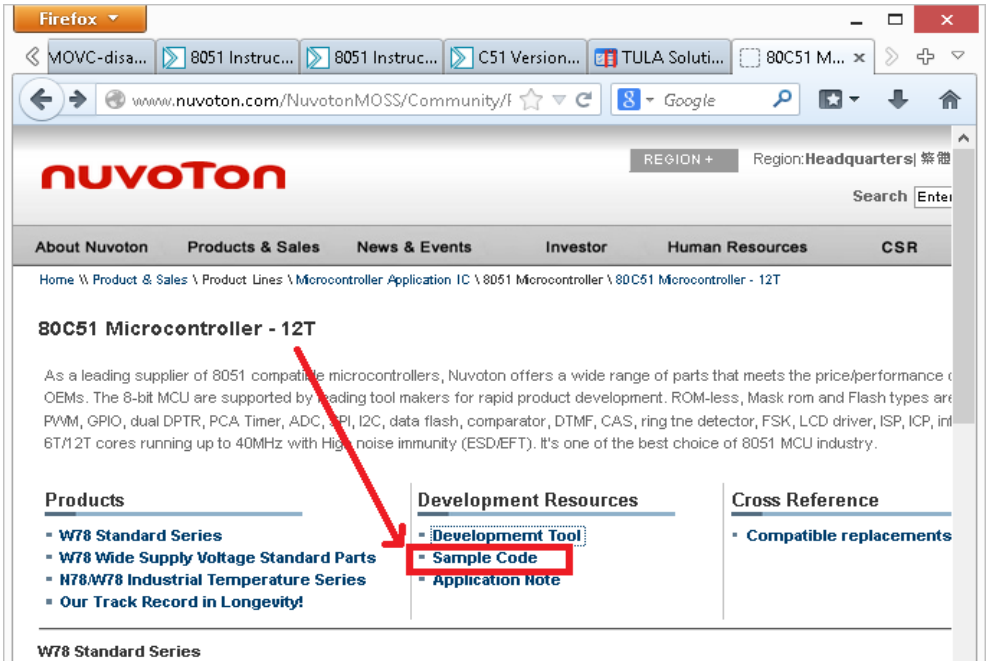

Setup Nuvoton ICE DLL V1.04
License Agreement
Please read the following license agreement carefully.
Nuvoton
To continue with SETUP, you must accept the terms of the License Agreement. To accept the agreement, click the check box below.
Nuvoton License Agreement
for Keil Add On-Component
IMPORTANT-READ THIS AGREEMENT CAREFULLY
Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments,
I agree to all the terms of the preceding License Agreement
Keil µVision3 Setup
<< Back
Next >>
Cancel

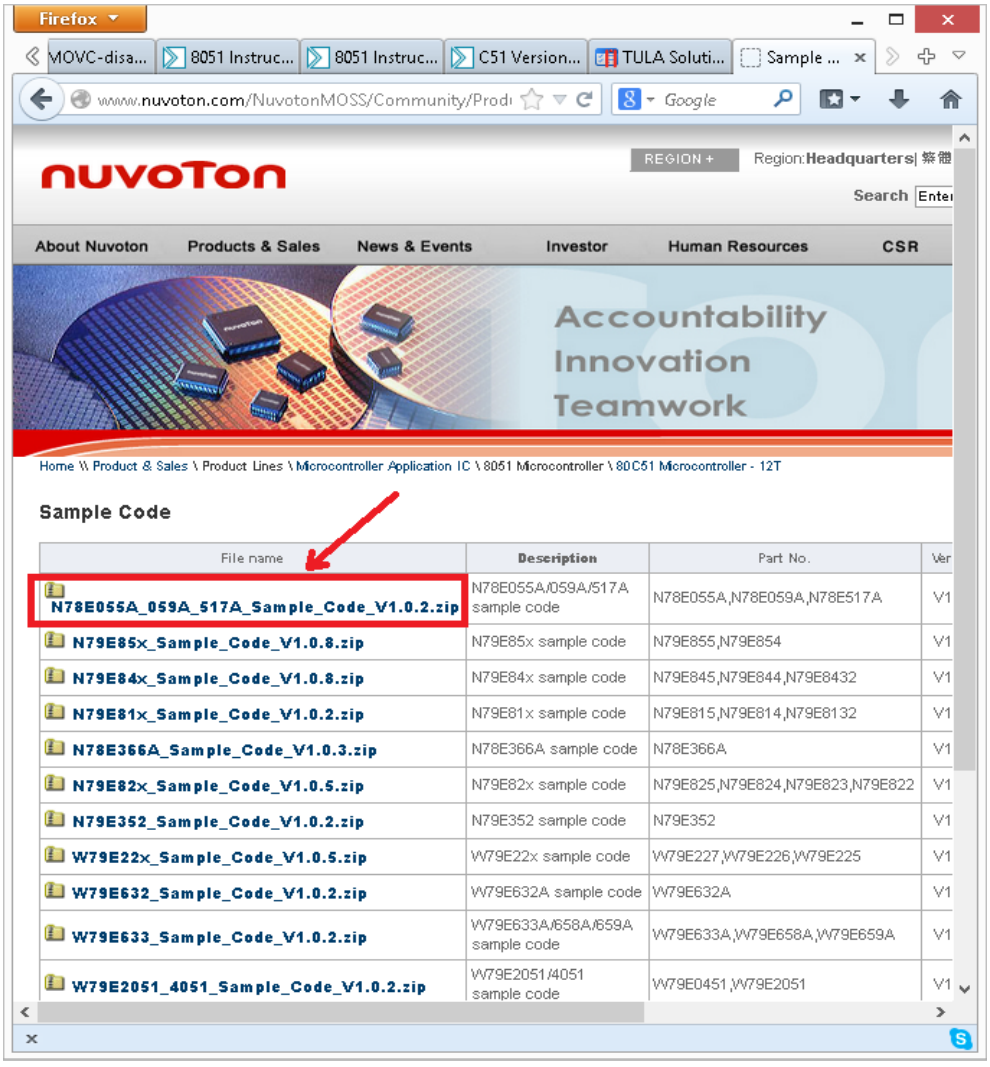

Setup Nuvoton ICE DLL V1.04
Folder Selection
Select the folder where SETUP will install files.
Nuvoton
SETUP will install µVision3 in the following folder.
To install to this folder, press 'Next'. To install to a different folder, press 'Browse' and select another folder.
Chỗ tới thư mục cài đặt của Keil-uVision.
Mặc định là:
Destination Folder
C:\Keil
Browse ...
Keil µVision3 Setup
<< Back
Next >>
Cancel

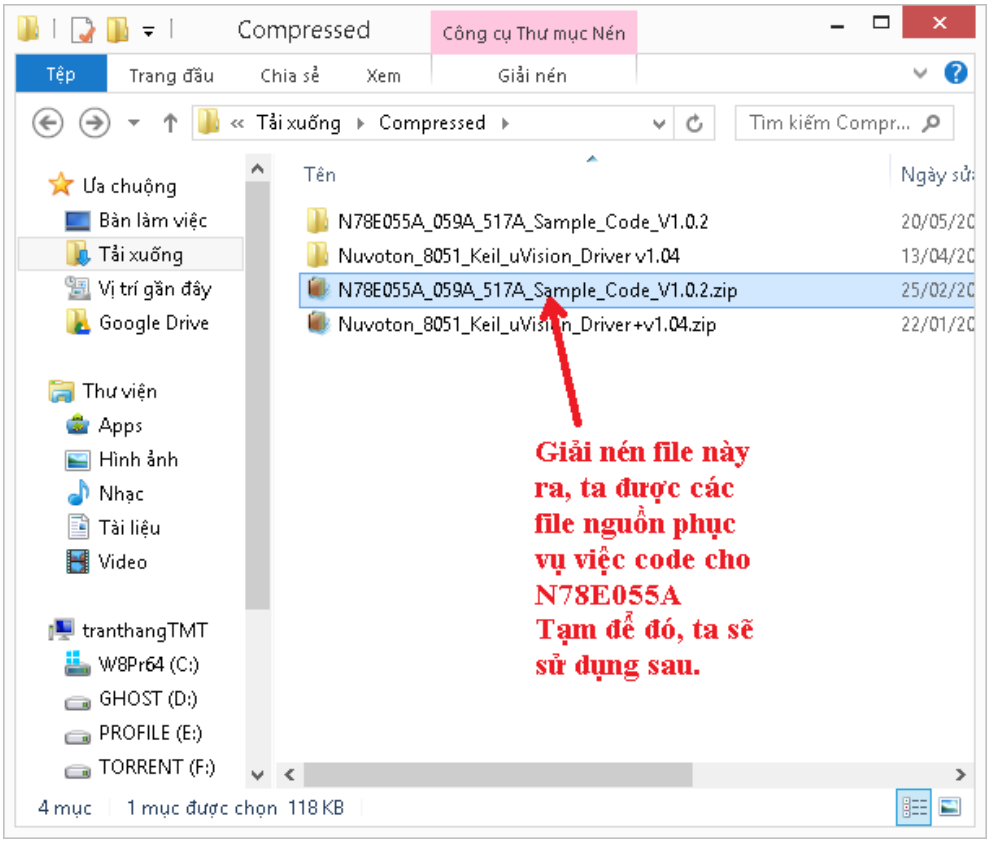

Setup Nuvoton ICE DLL V1.04
Customer Information
Please enter your information.
Nuvoton
Please enter your name, the name of the company for whom you work and your E-mail address.
Điền thông tin bất kỳ
First Name:
tranthang
Last Name:
TMT
Company Name:
TULA Solution
E-mail:
tranthangTMT@gmail.com
Keil µVision3 Setup
<< Back
Next >>
Cancel

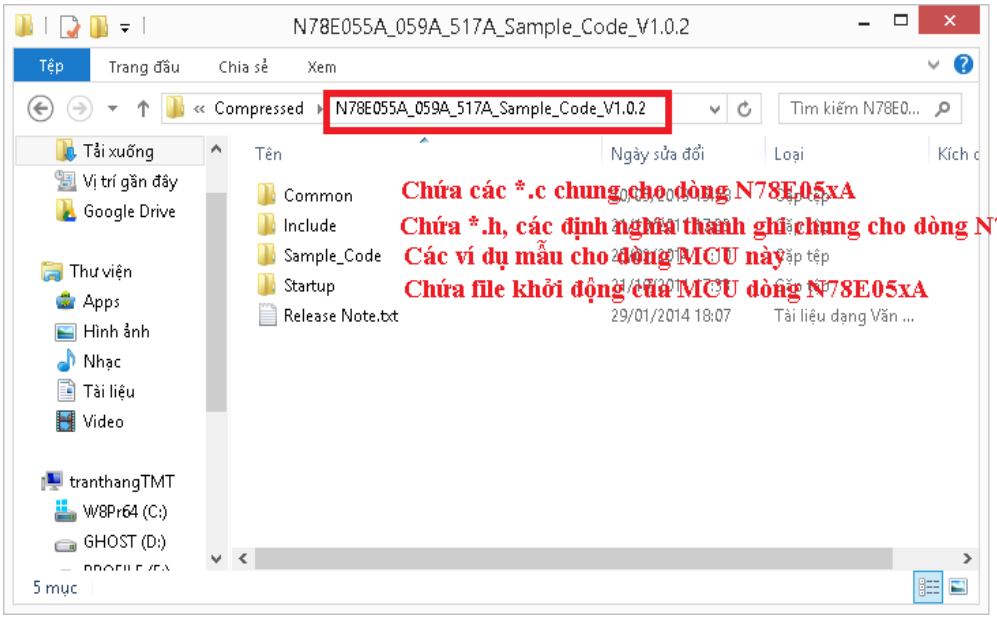

Setup Nuvoton ICE DLL V1.04
Keil µVision3 Setup completed
Nuvoton
µVision Setup has performed all requested operations successfully.
Keil µVision3 Setup
<< Back
Finish
Cancel

- Download các code mẫu cho N78E055A:





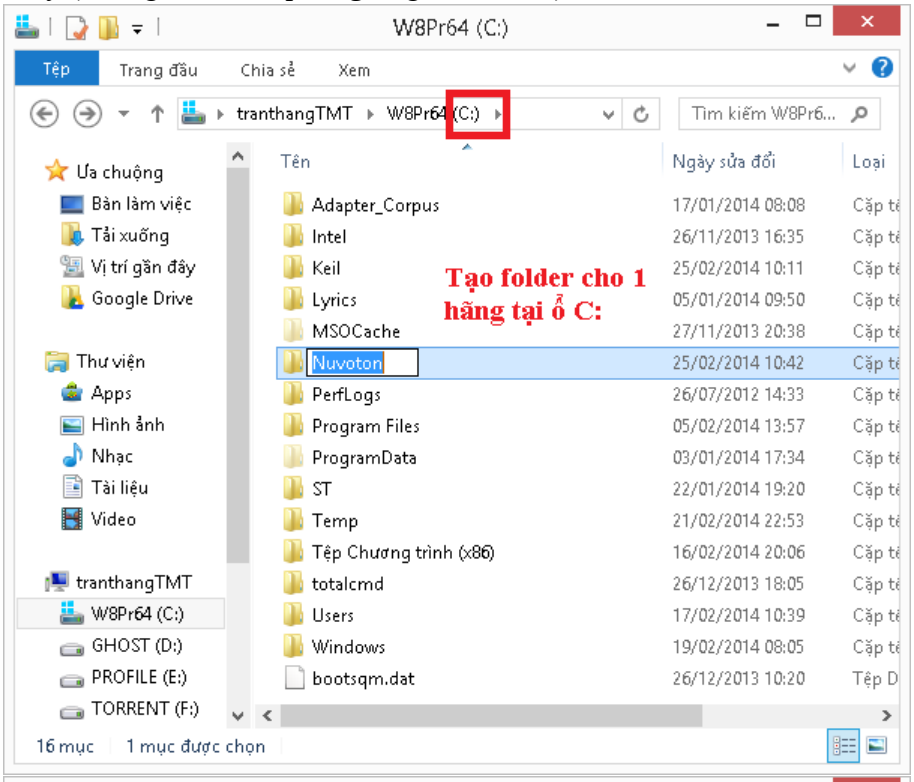

Compressed
Công cụ Thư mục Nén
Tệp
Trang đầu
Chia sẻ
Xem
Giải nén
« Tải xuống ▸ Compressed ▸
Tìm kiếm Compr...
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
GHOST (D:)
PROFILE (E:)
TORRENT (F:)
Tên
Ngày sử
N78E055A_059A_517A_Sample_Code_V1.0.2
20/05/20
Nuvoton_8051_Keil_uVision_Driver v1.04
13/04/20
N78E055A_059A_517A_Sample_Code_V1.0.2.zip
25/02/20
Nuvoton_8051_Keil_uVision_Driver+v1.04.zip
22/01/20
Giải nén file này ra, ta được các file nguồn phục vụ việc code cho N78E055A
Tạm để đó, ta sẽ sử dụng sau.
4 mục
1 mục được chọn 118 KB

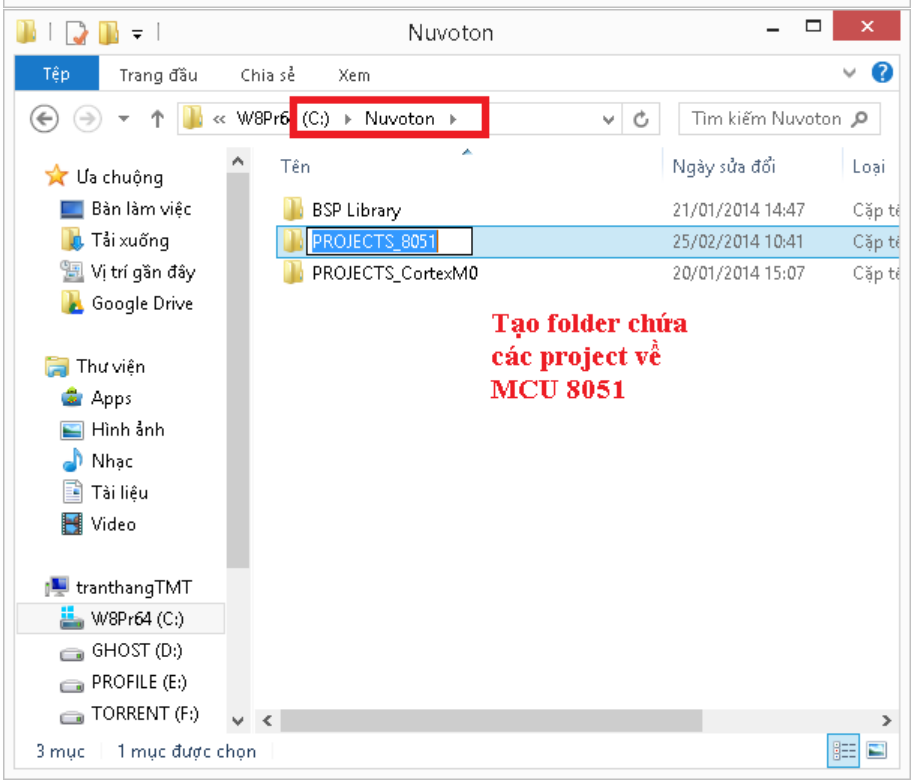

N78E055A_059A_517A_Sample_Code_V1.0.2
Tệp
Trang đầu
Chia sẻ
Xem
« Compressed ▸ N78E055A_059A_517A_Sample_Code_V1.0.2
Tìm kiếm N78E0...
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
GHOST (D:)
Tên
Ngày sửa đổi
Loại
Common
Include
Sample_Code
Startup
Release Note.txt
29/01/2014 18:07
Tài liệu dạng Văn ...
Chứa các *.c chung cho dòng N78E05xA
Chứa *.h, các định nghĩa thanh ghi chung cho dòng N78E05xA
Các ví dụ mẫu cho dòng MCU này
Chứa file khởi động của MCU dòng N78E05xA
5 mục

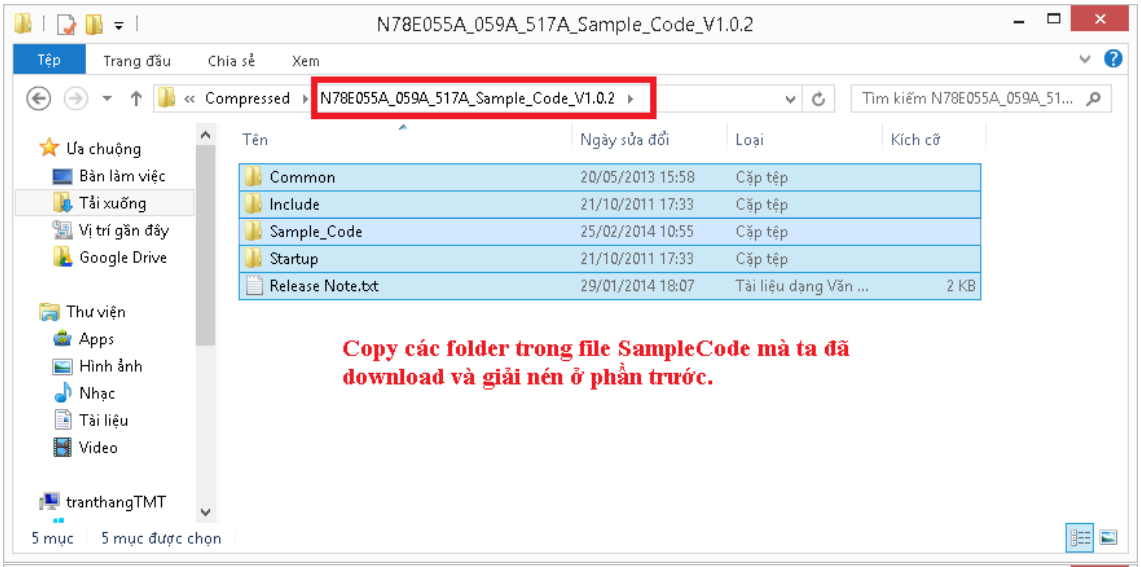
3. Build một project N78E055A với Keil-uVision 4

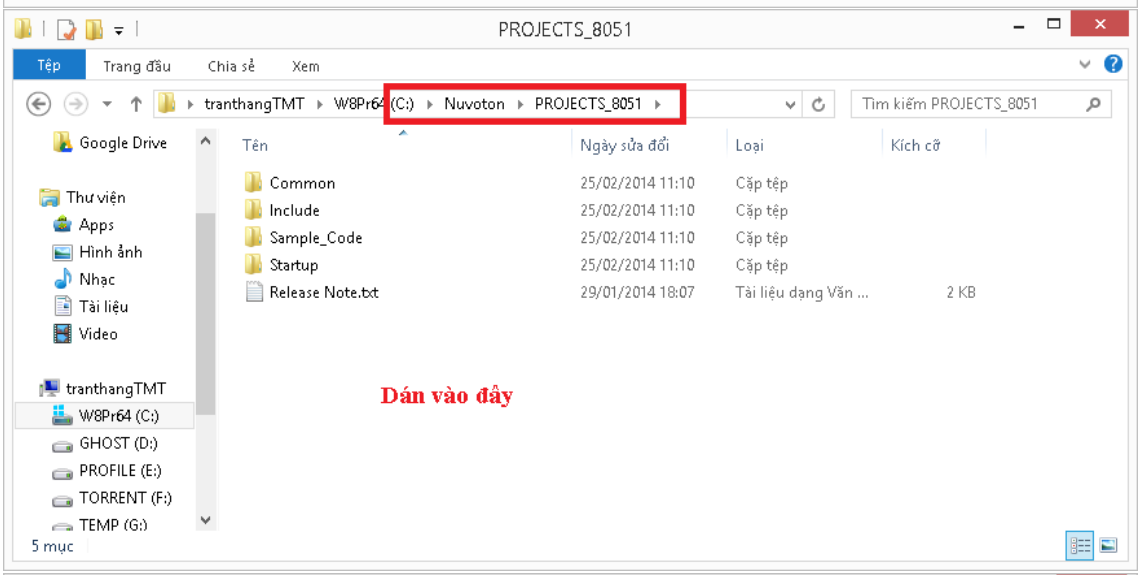
a. Tạo thư mục cho Project mới

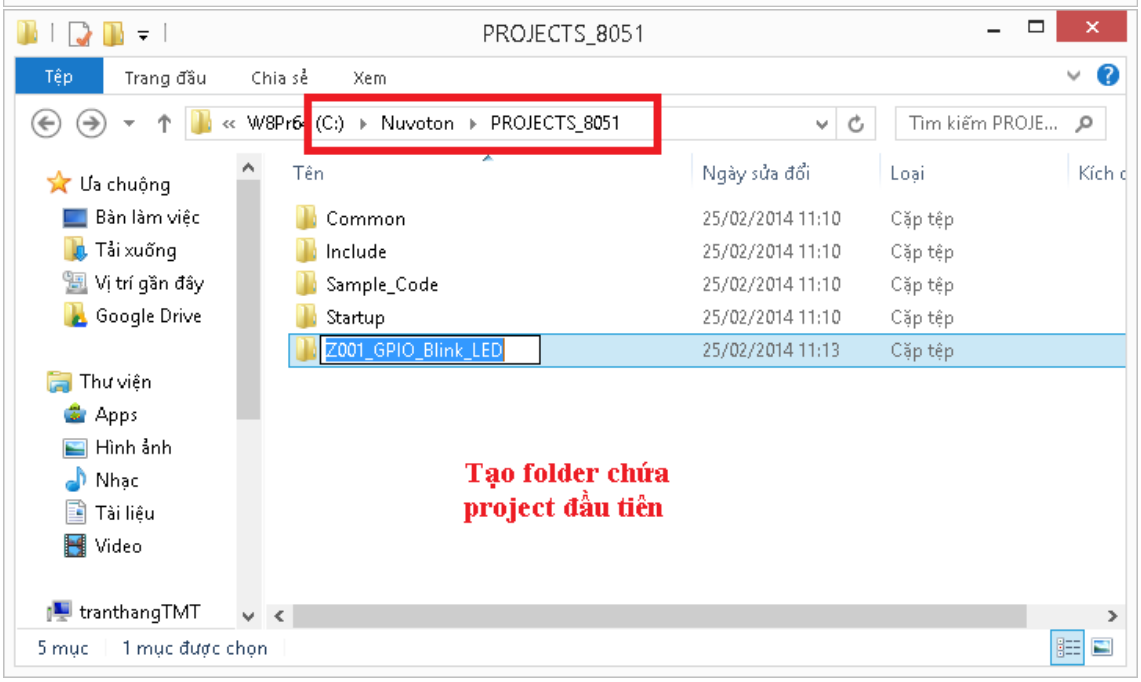
- Để thuận tiện cho việc phát triển N78E055A, ta tạo các folder như trong các hình dưới đây (không bắt buộc phải giống hoàn toàn):

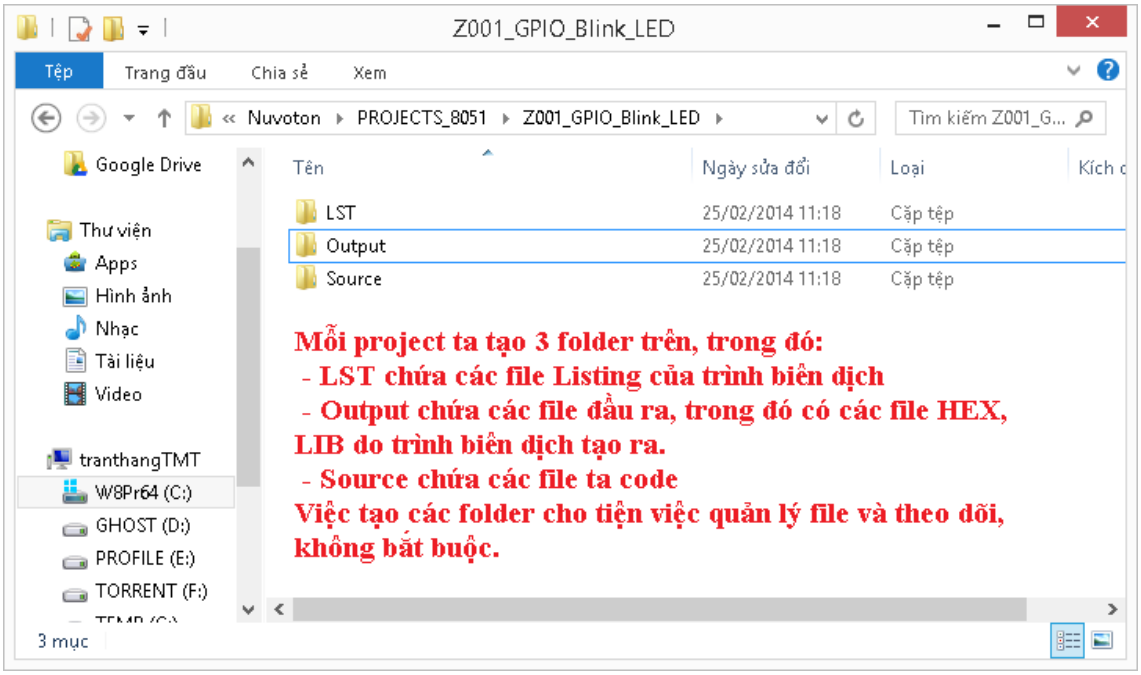





N78E055A_059A_517A_Sample_Code_V1.0.2
Tệp
Trang đầu
Chia sẻ
Xem
« Compressed ▸ N78E055A_059A_517A_Sample_Code_V1.0.2 ▸
Tìm kiếm N78E055A_059A_51...
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
Tên
Ngày sửa đổi
Loại
Kích cỡ
Common 20/05/2013 15:58 Cặp tệp
Include 21/10/2011 17:33 Cặp tệp
Sample_Code 25/02/2014 10:55 Cặp tệp
Startup 21/10/2011 17:33 Cặp tệp
Release Note.txt 29/01/2014 18:07 Tài liệu dạng Văn ... 2 KB
5 mục
5 mục được chọn
Copy các folder trong file SampleCode mà ta đã download và giải nén ở phần trước.

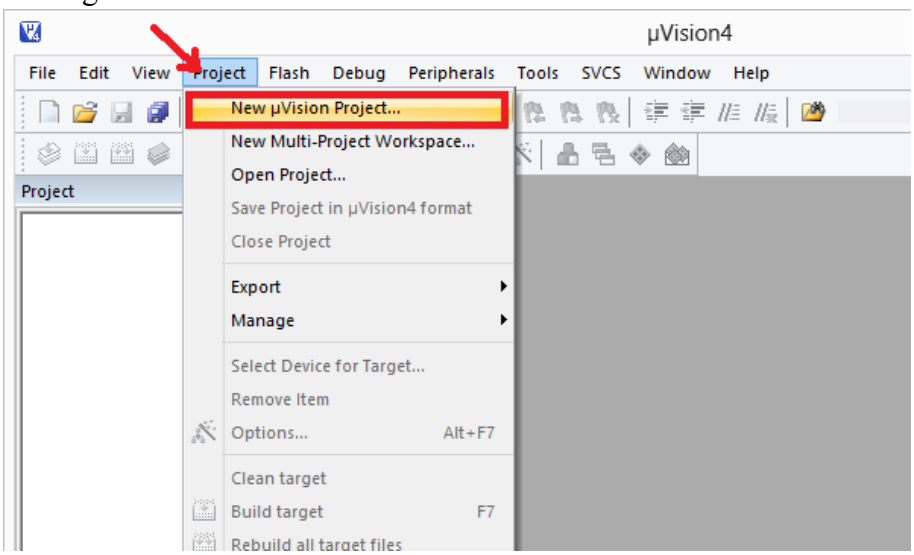

PROJECTS_8051
Tệp
Trang đầu
Chia sẻ
Xem
tranthangTMT ▸ W8Pr64 (C:) ▸ Nuvoton ▸ PROJECTS_8051 ▸
Tìm kiếm PROJECTS_8051
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
GHOST (D:)
PROFILE (E:)
TORRENT (F:)
TEMP (G:)
Tên
Ngày sửa đổi
Loại
Kích cỡ
Common 25/02/2014 11:10 Cặp tệp
Include 25/02/2014 11:10 Cặp tệp
Sample_Code 25/02/2014 11:10 Cặp tệp
Startup 25/02/2014 11:10 Cặp tệp
Release Note.txt 29/01/2014 18:07 Tài liệu dạng Văn ... 2 KB
5 mục
Dán vào đây

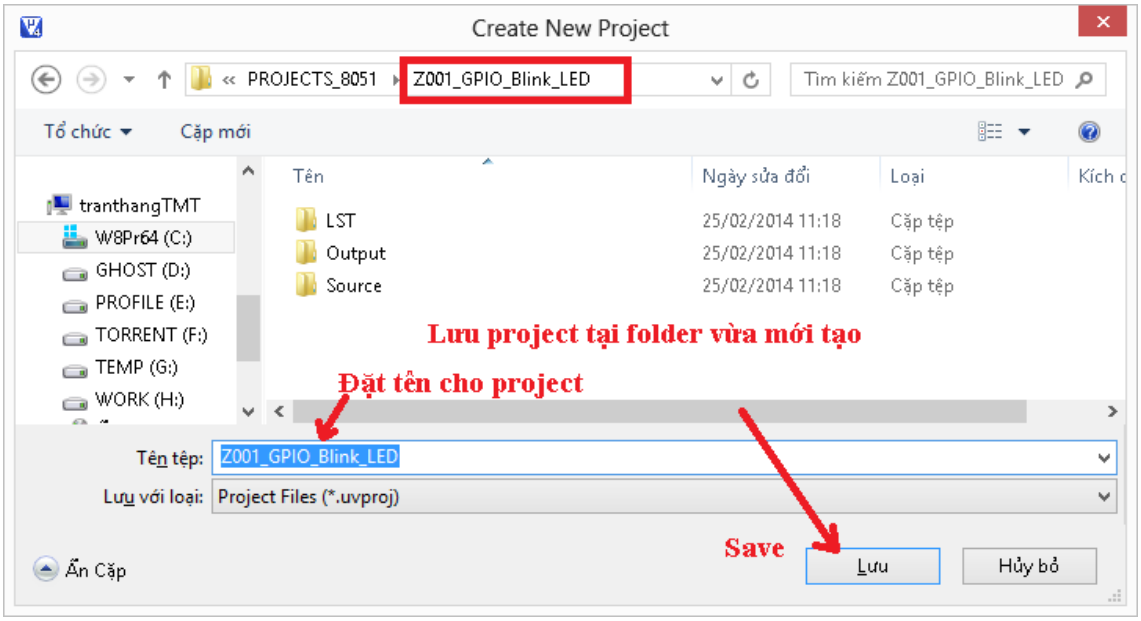

PROJECTS_8051
Tệp
Trang đầu
Chia sẻ
Xem
« W8Pr64 (C:) ▸ Nuvoton ▸ PROJECTS_8051
Tìm kiếm PROJE...
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
Tên
Ngày sửa đổi
Loại
Common 25/02/2014 11:10 Cặp tệp
Include 25/02/2014 11:10 Cặp tệp
Sample_Code 25/02/2014 11:10 Cặp tệp
Startup 25/02/2014 11:10 Cặp tệp
Z001_GPIO_Blink_LED 25/02/2014 11:13 Cặp tệp
5 mục
1 mục được chọn
Tạo folder chứa project đầu tiên

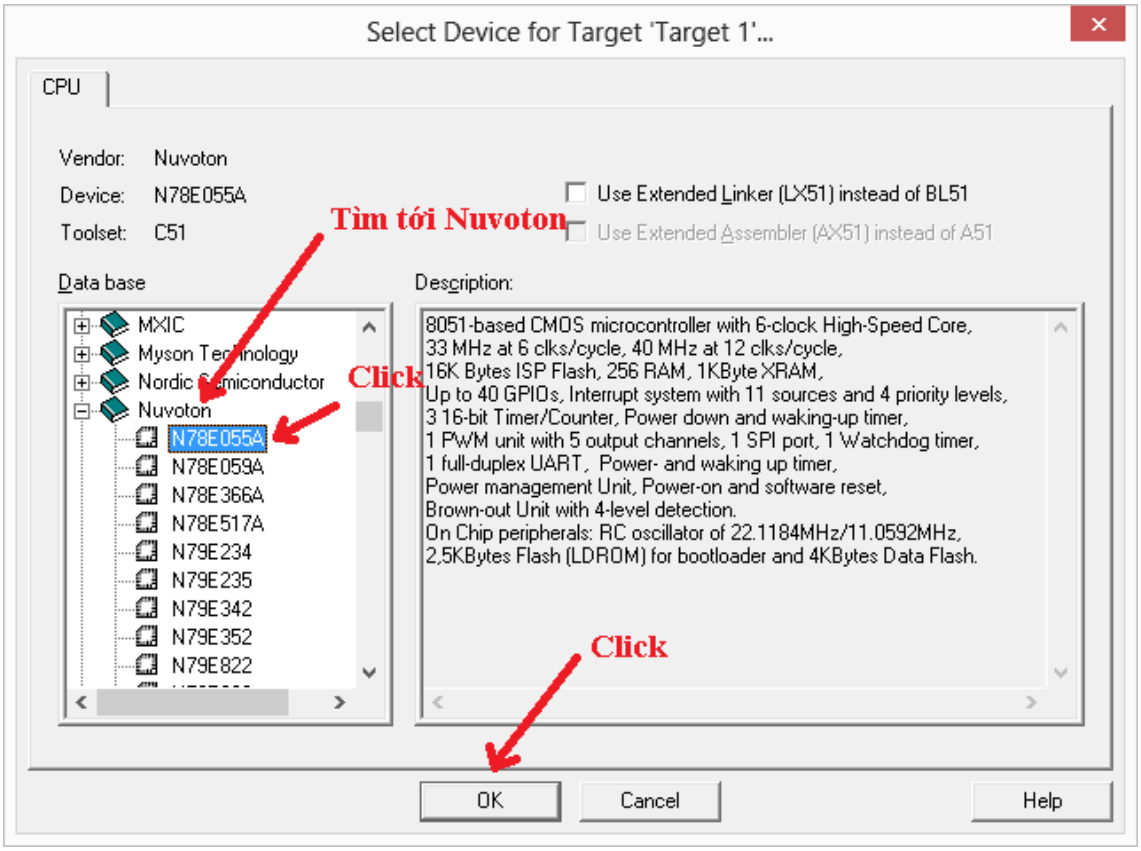

Z001_GPIO_Blink_LED
Tệp
Trang đầu
Chia sẻ
Xem
Nuvoton ▸ PROJECTS_8051 ▸ Z001_GPIO_Blink_LED ▸
Tìm kiếm Z001_G...
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
GHOST (D:)
PROFILE (E:)
TORRENT (F:)
Tên
Ngày sửa đổi
Loại
LST 25/02/2014 11:18 Cặp tệp
Output 25/02/2014 11:18 Cặp tệp
Source 25/02/2014 11:18 Cặp tệp
Mỗi project ta tạo 3 folder trên, trong đó:
- LST chứa các file Listing của trình biên dịch
- Output chứa các file đầu ra, trong đó có các file HEX, LIB do trình biên dịch tạo ra.
- Source chứa các file ta code
Việc tạo các folder cho tiện việc quản lý file và theo dõi, không bắt buộc.
3 mục

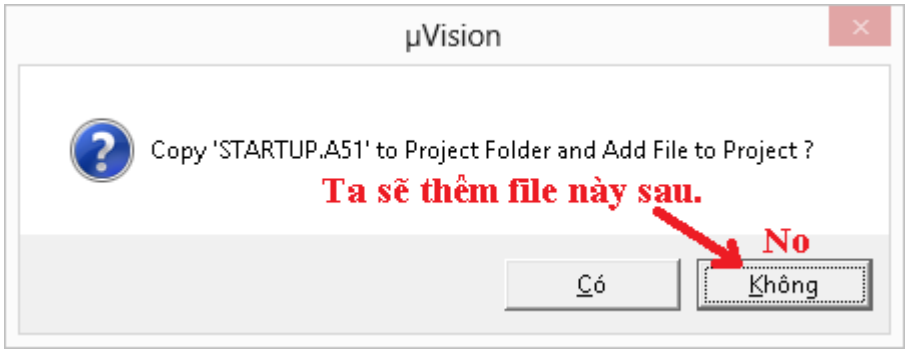
b. Tạo khung project mới

- Mở chương trình Keil-uVision 4 lên 

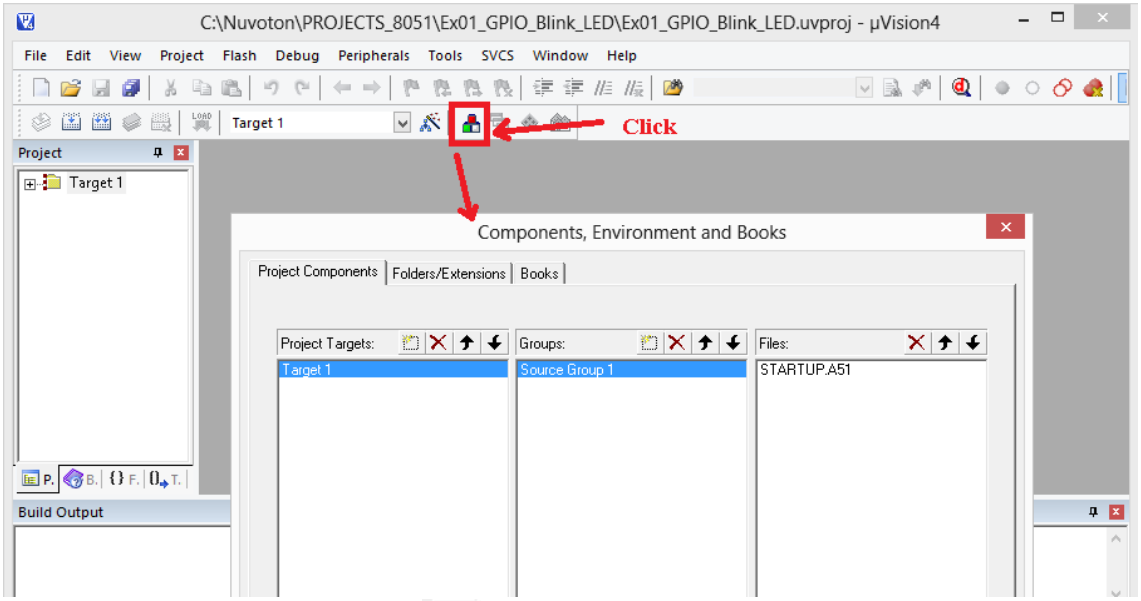




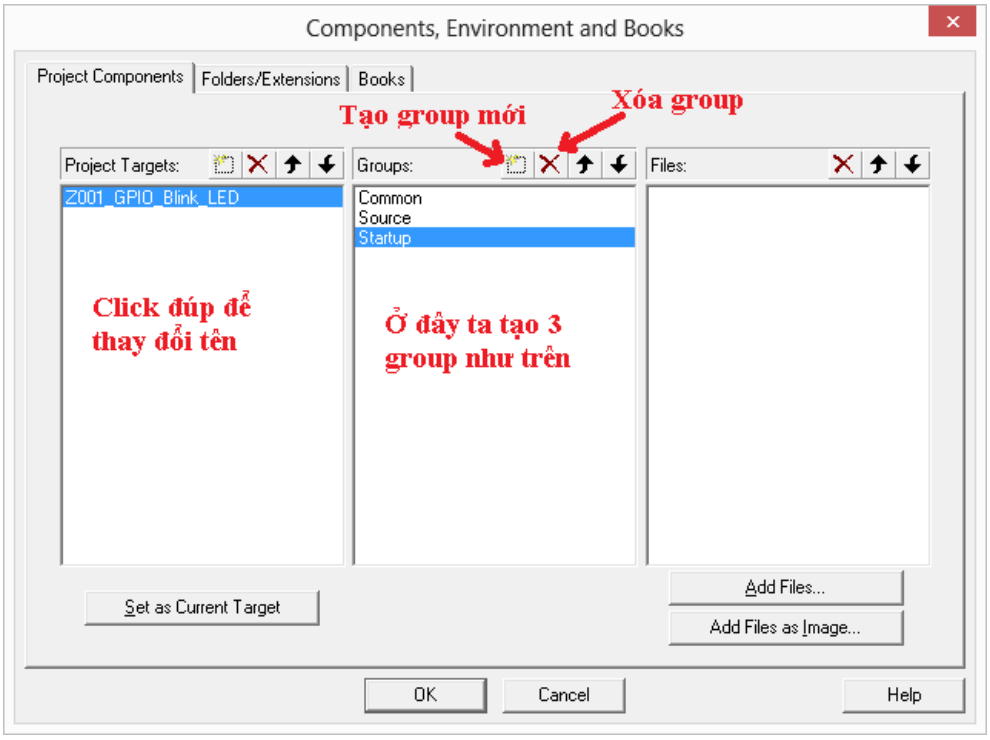

Select Device for Target 'Target 1'...
CPU
Vendor: Nuvoton
Device: N78E055A
Toolset: C51
Use Extended Linker (LX51) instead of BL51
Use Extended Assembler (AX51) instead of A51
Tìm tới Nuvoton
Data base
Description:
MXIC
Myson Technology
Nordic Semiconductor
Nuvoton
N78E055A
N78E059A
N78E366A
N78E517A
N79E234
N79E235
N79E342
N79E352
N79E822
Click
8051-based CMOS microcontroller with 6-clock High-Speed Core,
33 MHz at 6 clks/cycle, 40 MHz at 12 clks/cycle,
16K Bytes ISP Flash, 256 RAM, 1KByte XRAM,
Up to 40 GPIOs, Interrupt system with 11 sources and 4 priority levels,
3 16-bit Timer/Counter, Power down and waking-up timer,
1 PWM unit with 5 output channels, 1 SPI port, 1 Watchdog timer,
1 full-duplex UART, Power- and waking up timer,
Power management Unit, Power-on and software reset,
Brown-out Unit with 4-level detection.
On Chip peripherals: RC oscillator of 22.1184MHz/11.0592MHz,
2,5KBytes Flash (LDROM) for bootloader and 4KBytes Data Flash.
Click
OK
Cancel
Help


µVision
Copy 'STARTUP.A51' to Project Folder and Add File to Project ?
Ta sẽ thêm file này sau.
No
Có
Không

- Thêm các file dùng cho project phục vụ cho Editor:

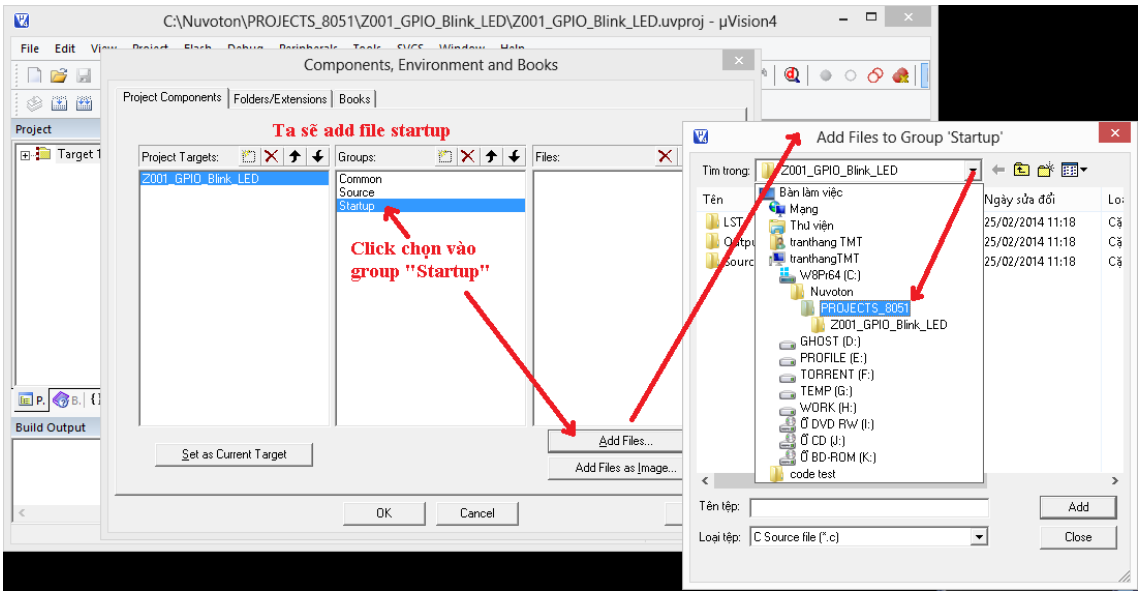


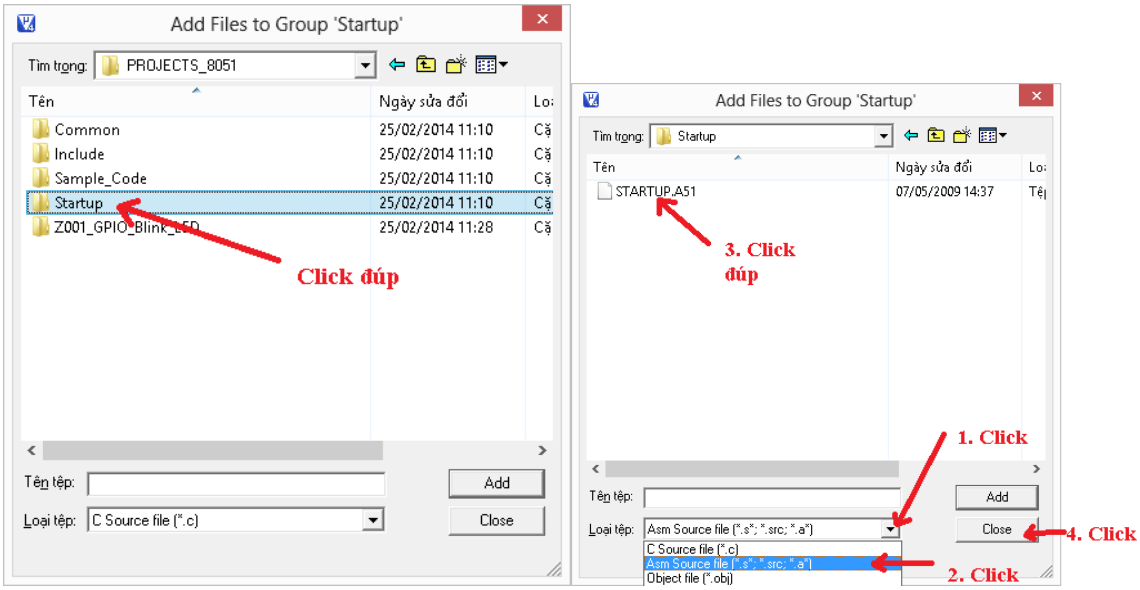
- Tạo Target/Group mới bằng  , xóa bỏ Target/Group bằng 
- Việc tạo các Targer/Group để thuận tiện cho việc quản lý, theo dõi, truy cập các thư viện, file mà ta dùng cho project được thuận tiện, không bắt buộc phải tạo các group như trong hướng dẫn. Và các tên của Target, Group là tự do, không bắt buộc phải theo đúng tên như hướng dẫn.
- Ở đây ta tạo 3 group Common (chứa các file thư viện chung như Delay,…), Startup (chứa file khởi động cho MCU – các thiết lập, .. ban đầu khi MCU reset), Sources (chứa các file của project do người dùng code)
- ***Chú ý: việc thêm các file mà ta không dùng đến cũng không sao, tuy nhiên sẽ tốn thời gian khi ta biên dịch toàn bộ project. Vì vậy, trong hướng dẫn dưới đây chỉ là hướng dẫn chung chứ không phải là bắt buộc cho tất cả project. Những file nào cần dùng cho project thì thêm, còn không thì thôi.***

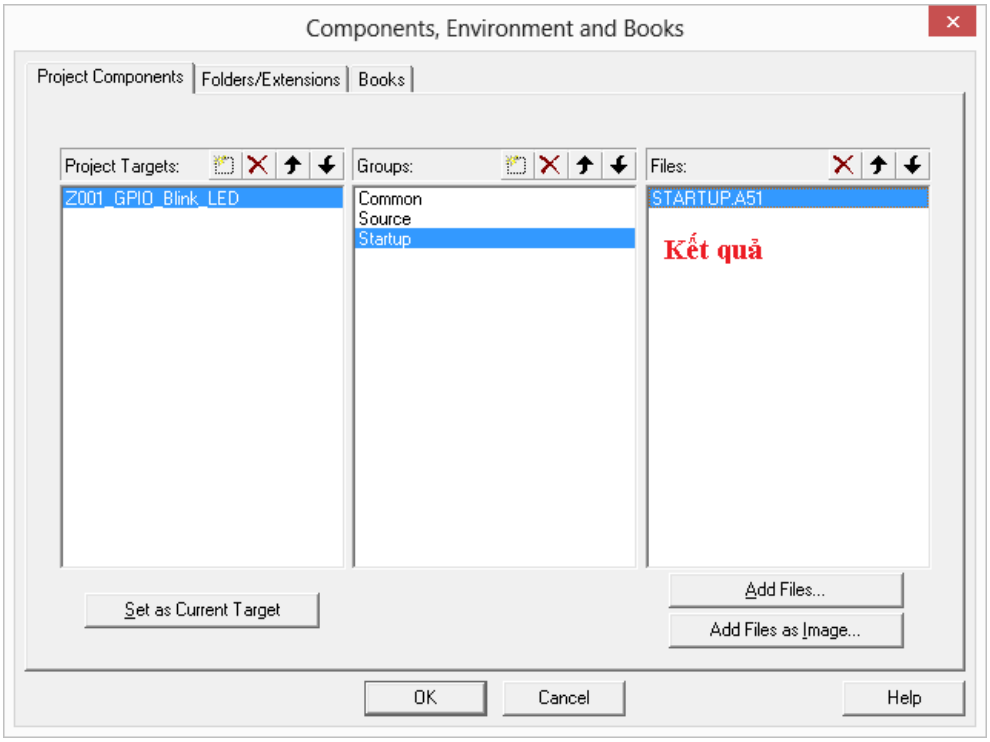


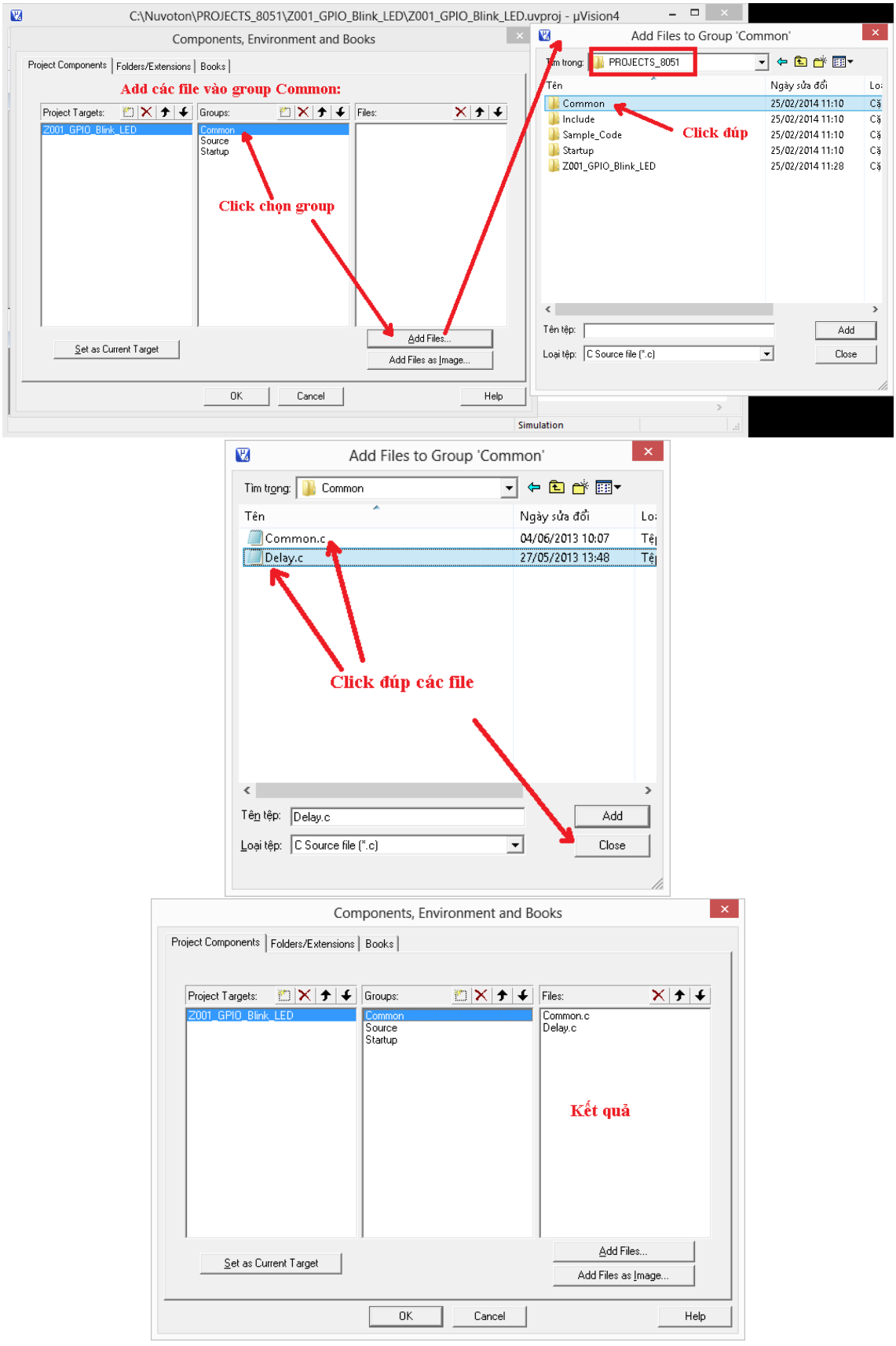
- Bây giờ ta sẽ add các file sẽ dùng cho Example này. Ta truy cập vào folder chứa thư viện của dòng N78E055A theo đường dẫn, mặc định:

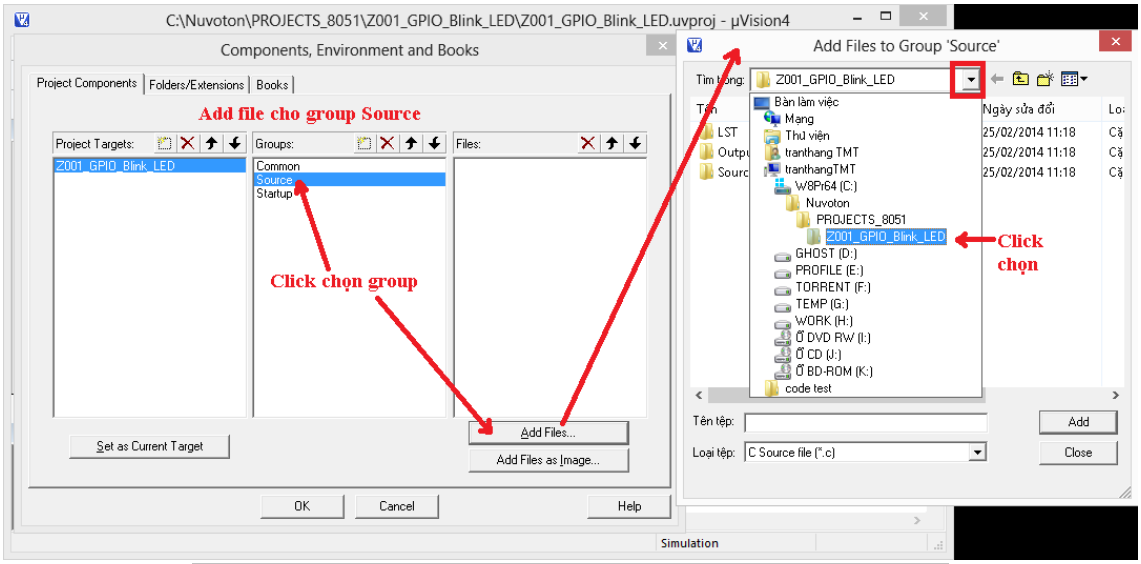
"C:\Nuvoton\PROJECTS_8051"

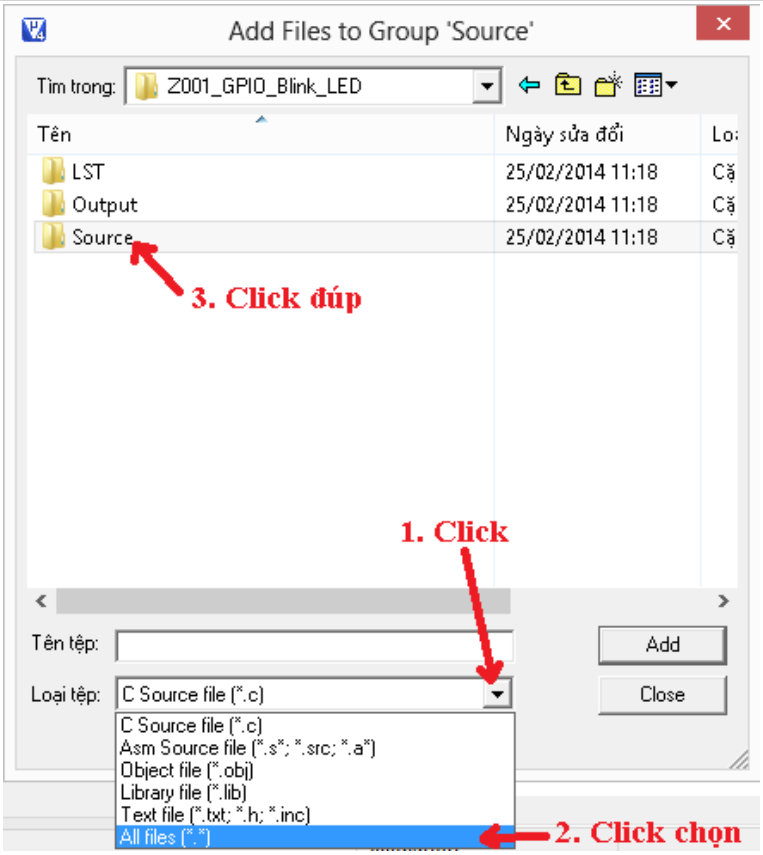


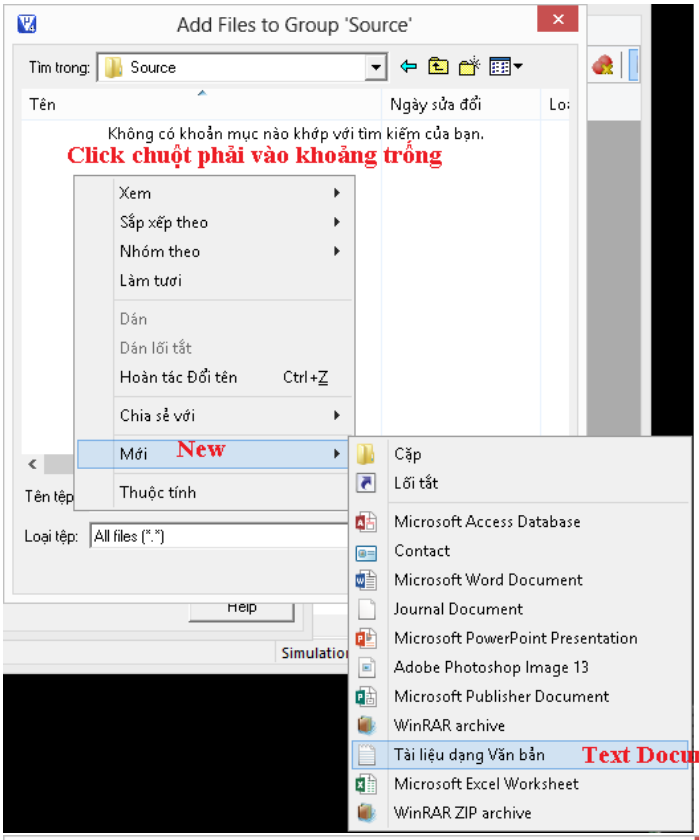

Add Files to Group 'Startup'
Tìm trong: PROJECTS_8051
Tên
Ngày sửa đổi
Common 25/02/2014 11:10
Include 25/02/2014 11:10
Sample_Code 25/02/2014 11:10
Startup 25/02/2014 11:10
Z001_GPIO_Blink_LED 25/02/2014 11:28
Click đúp
Tên tệp:
Loại tệp: C Source file (*.c)
Add
Close
Add Files to Group 'Startup'
Tìm trong: Startup
STARTUP.A51 07/05/2009 14:37
3. Click đúp
1. Click
Asm Source file (*.s*; *.src; *.a*)
C Source file (*.c)
Object file (*.obj)
2. Click
4. Click

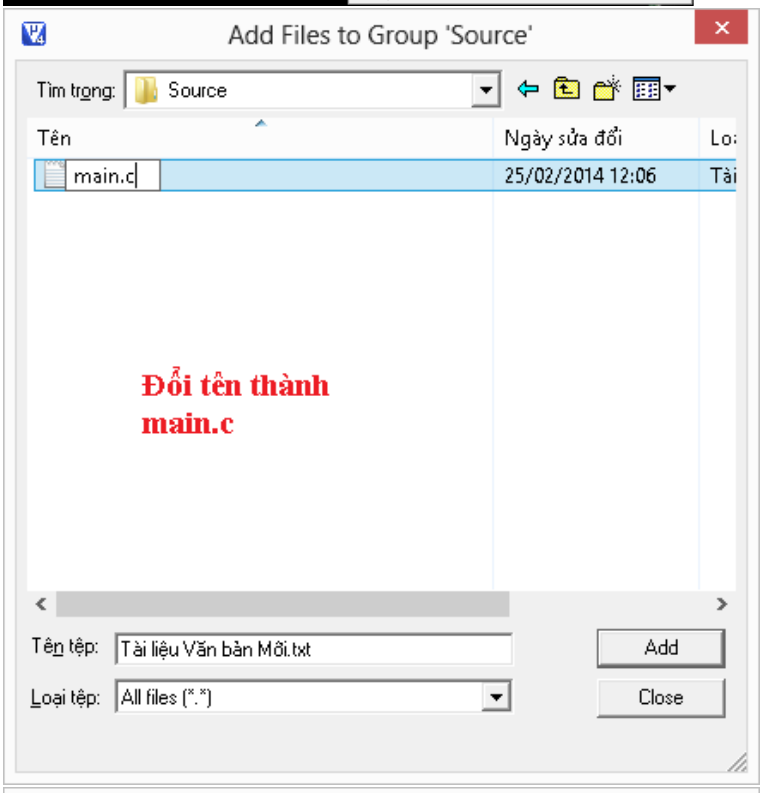

Components, Environment and Books
Project Components
Folders/Extensions
Books
Project Targets:
Groups:
Files:
Z001_GPIO_Blink_LED
Common
Source
Startup
STARTUP.A51
Kết quả
Set as Current Target
Add Files...
Add Files as Image...
OK
Cancel
Help

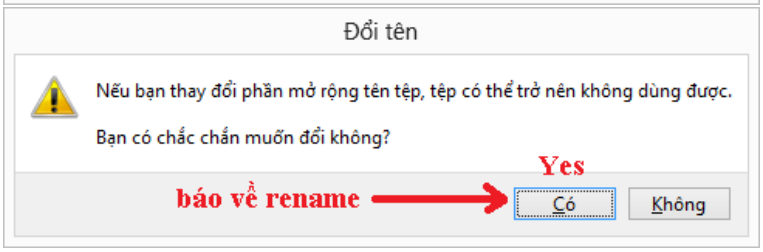

C:\Nuvoton\PROJECTS_8051\Z001_GPIO_Blink_LED\Z001_GPIO_Blink_LED.uvproj - µVision4
Components, Environment and Books
Project Components
Folders/Extensions
Books
Add các file vào group Common:
Project Targets:
Groups:
Files:
Z001_GPIO_Blink_LED
Common
Source
Startup
Click chọn group
Set as Current Target
Add Files...
Add Files as Image...
OK
Cancel
Help
Simulation
Add Files to Group 'Common'
Tìm trong:
PROJECTS_8051
Tên
Ngày sửa đổi
Common
Include
Sample_Code
Startup
Z001_GPIO_Blink_LED
25/02/2014 11:10
25/02/2014 11:28
Click đúp
Tên tệp:
Loại tệp:
C Source file (*.c)
Add
Close
Add Files to Group 'Common'
Tìm trong:
Common
Common.c
Delay.c
04/06/2013 10:07
27/05/2013 13:48
Click đúp các file
Tên tệp:
Delay.c
Loại tệp:
C Source file (*.c)
Add
Close
Components, Environment and Books
Project Components
Folders/Extensions
Books
Project Targets:
Groups:
Files:
Z001_GPIO_Blink_LED
Common
Source
Startup
Common.c
Delay.c
Kết quả
Set as Current Target
Add Files...
Add Files as Image...
OK
Cancel
Help

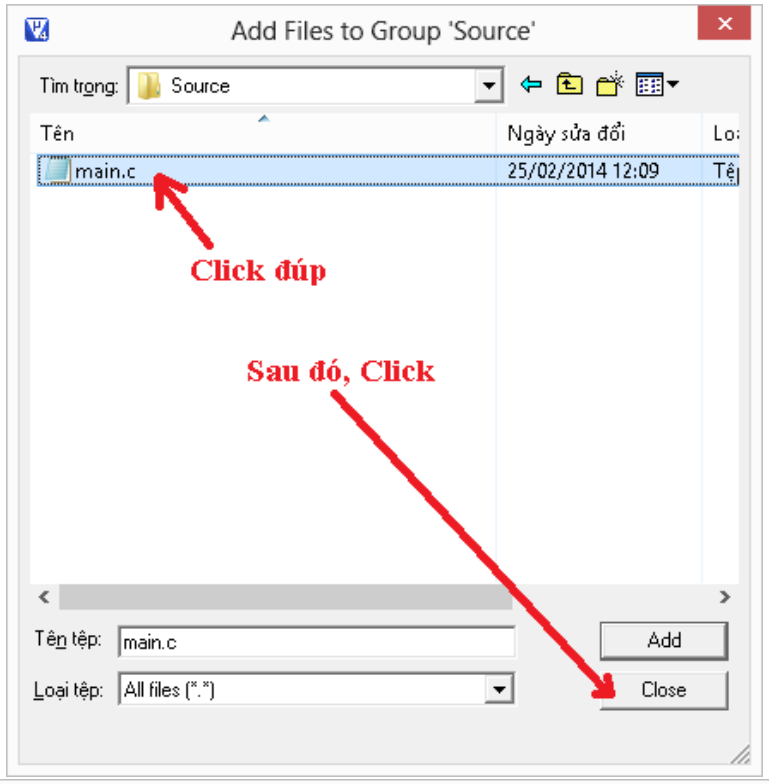

C:\Nuvoton\PROJECTS_8051\Z001_GPIO_Blink_LED\Z001_GPIO_Blink_LED.uvproj - µVision4
Components, Environment and Books
Project Components
Folders/Extensions
Books
Add file cho group Source
Project Targets:
Groups:
Files:
Z001_GPIO_Blink_LED
Common
Source
Startup
Click chọn group
Add Files...
Add Files as Image...
Set as Current Target
OK
Cancel
Help
Simulation
Add Files to Group 'Source'
Tìm trong:
Z001_GPIO_Blink_LED
Bàn làm việc
Mạng
Thư viện
tranthang TMT
tranthangTMT
W8Pr64 (C:)
Nuvoton
PROJECTS_8051
Z001_GPIO_Blink_LED
Click chọn
GHOST (D:)
PROFILE (E:)
TORRENT (F:)
TEMP (G:)
WORK (H:)
Ổ DVD RW (I:)
Ổ CD (J:)
Ổ BD-ROM (K:)
code test
Ngày sửa đổi
25/02/2014 11:18
Tên tệp:
Loại tệp:
C Source file (*.c)
Add
Close

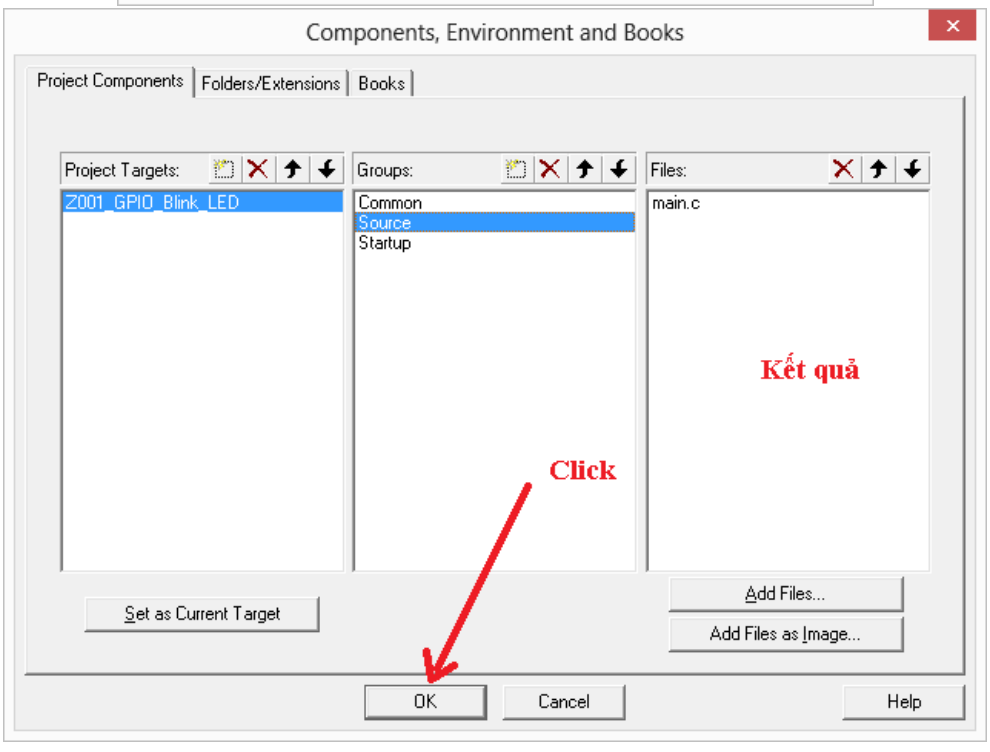

Add Files to Group 'Source'
Tìm trong:
Z001_GPIO_Blink_LED
Tên
Ngày sửa đổi
LST
25/02/2014 11:18
Output
25/02/2014 11:18
Source
25/02/2014 11:18
3. Click đúp
1. Click
Tên tệp:
Loại tệp:
C Source file (*.c)
Add
Close
C Source file (*.c)
Asm Source file (*.s*; *.src; *.a*)
Object file (*.obj)
Library file (*.lib)
Text file (*.txt; *.h; *.inc)
All files (*.*)
2. Click chọn

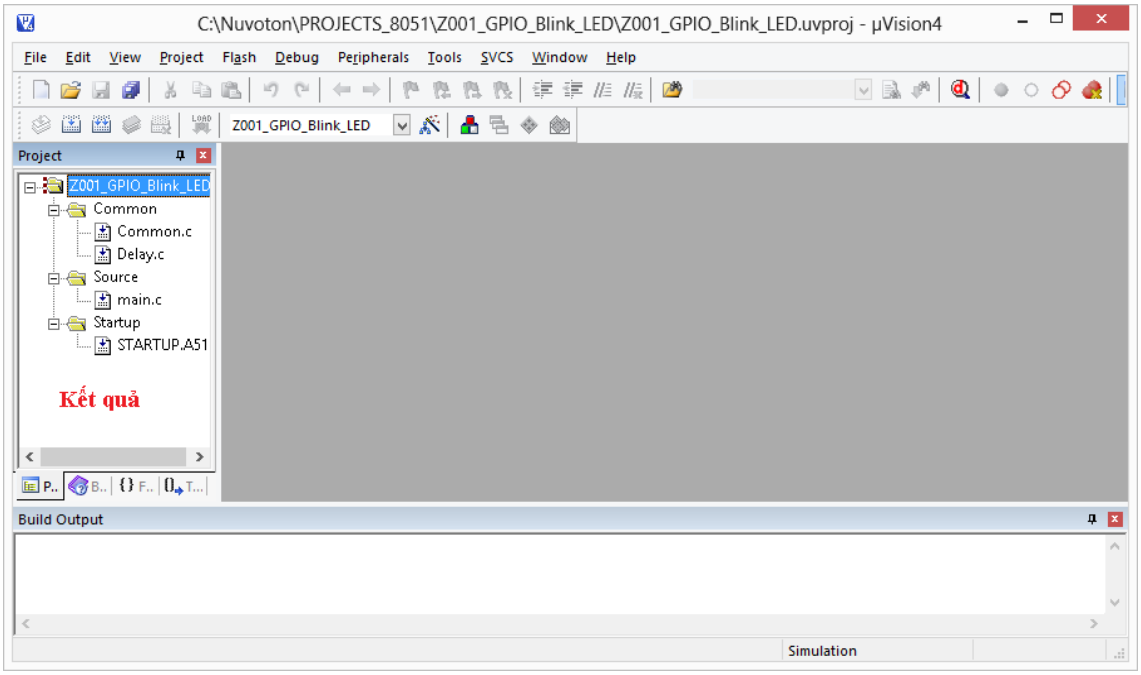

Add Files to Group 'Source'
Tìm trong: Source
Tên
Ngày sửa đổi
Không có khoản mục nào khớp với tìm kiếm của bạn.
Click chuột phải vào khoảng trống
Xem
Sắp xếp theo
Nhóm theo
Làm tươi
Dán
Dán lối tắt
Hoàn tác Đổi tên Ctrl+Z
Chia sẻ với
Mới
New
Thuộc tính
Tên tệp:
Loại tệp: All files (*.*)
Cặp
Lối tắt
Microsoft Access Database
Contact
Microsoft Word Document
Journal Document
Microsoft PowerPoint Presentation
Adobe Photoshop Image 13
Microsoft Publisher Document
WinRAR archive
Tài liệu dạng Văn bản
Text Document
Microsoft Excel Worksheet
WinRAR ZIP archive

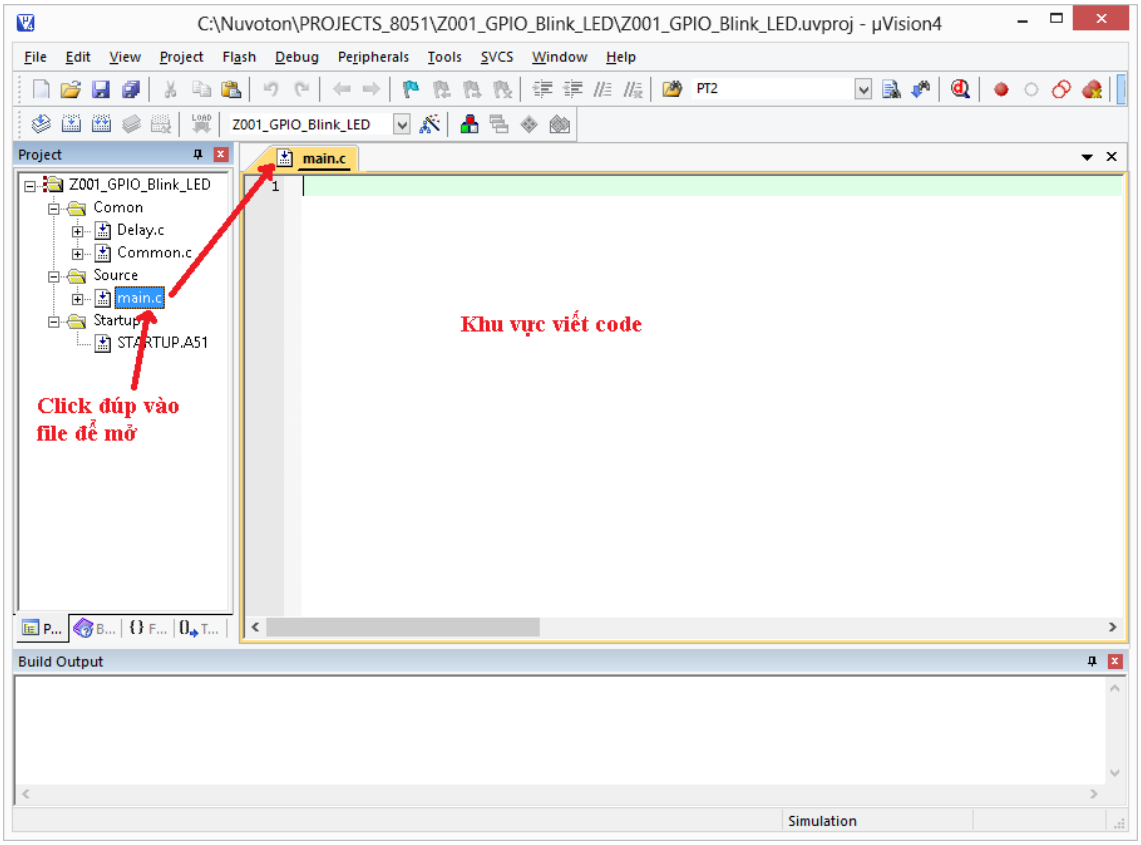

Add Files to Group 'Source'
Tìm trong: Source
Tên
Ngày sửa đổi
main.c
25/02/2014 12:06
Đổi tên thành
main.c
Tên tệp: Tài liệu Văn bản Mới.txt
Loại tệp: All files (*.*)
Add
Close


Đổi tên
Nếu bạn thay đổi phần mở rộng tên tệp, tệp có thể trở nên không dùng được.
Bạn có chắc chắn muốn đổi không?
Yes
báo về rename
Có
Không

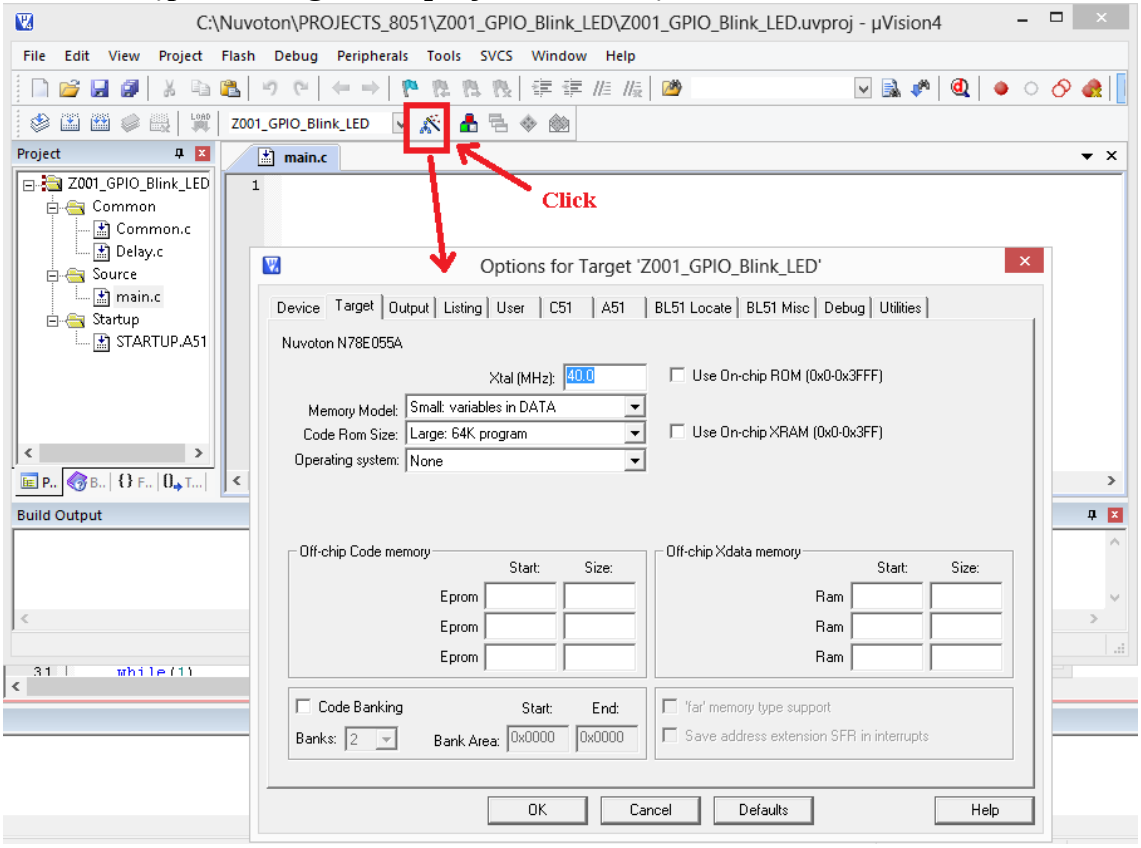

Add Files to Group 'Source'
Tìm trong:
Source
Tên
Ngày sửa đổi
main.c
25/02/2014 12:09
Click đúp
Sau đó, Click
Tên tệp:
main.c
Loại tệp:
All files (*.*)
Add
Close

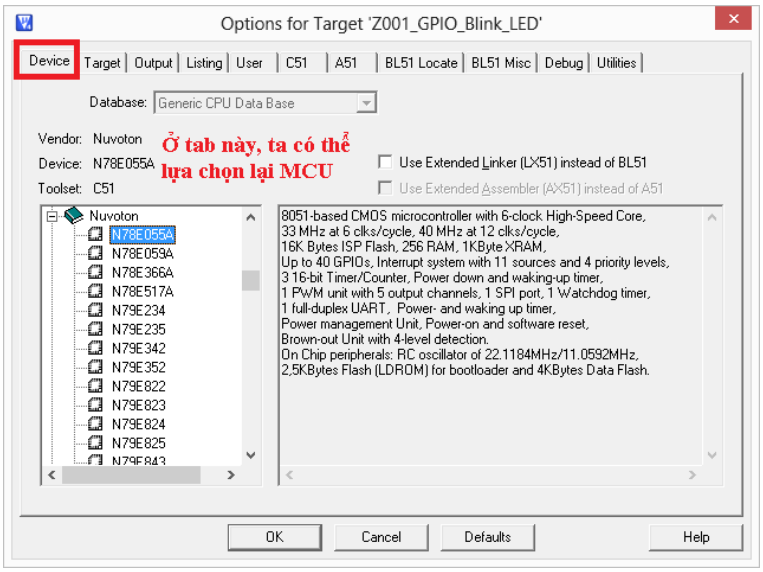

Components, Environment and Books
Project Components
Folders/Extensions
Books
Project Targets:
Z001_GPIO_Blink_LED
Groups:
Common
Source
Startup
Files:
main.c
Kết quả
Click
Set as Current Target
Add Files...
Add Files as Image...
OK
Cancel
Help

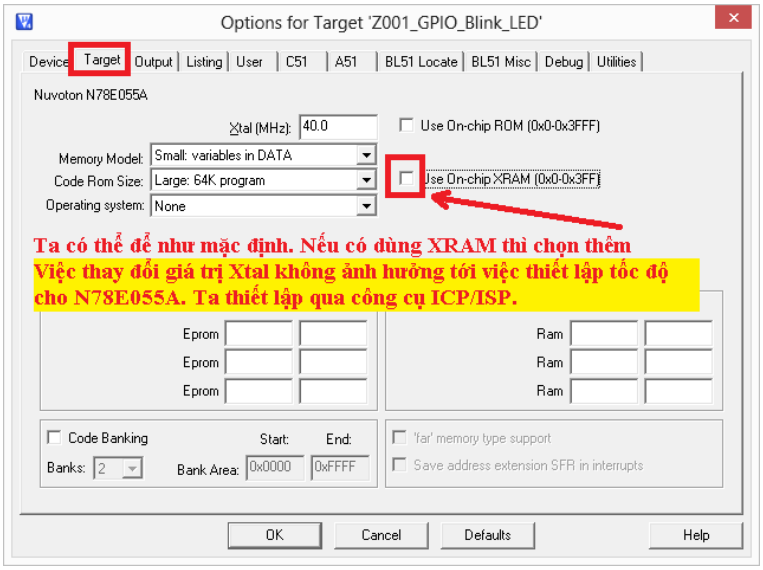

C:\Nuvoton\PROJECTS_8051\Z001_GPIO_Blink_LED\Z001_GPIO_Blink_LED.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Z001_GPIO_Blink_LED
Project
Z001_GPIO_Blink_LED
Common
Common.c
Delay.c
Source
main.c
Startup
STARTUP.A51
Kết quả
Build Output
Simulation

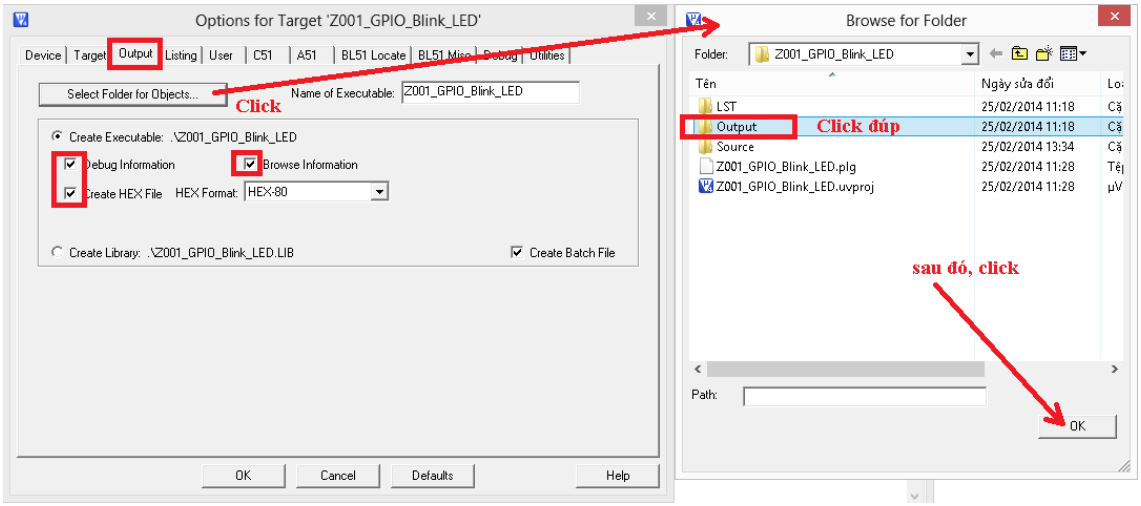

C:\Nuvoton\PROJECTS_8051\Z001_GPIO_Blink_LED\Z001_GPIO_Blink_LED.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
PT2
Z001_GPIO_Blink_LED
Project
main.c
Z001_GPIO_Blink_LED
Comon
Delay.c
Common.c
Source
main.c
Startup
STARTUP.A51
Click đúp vào file để mở
Khu vực viết code
Build Output
Simulation

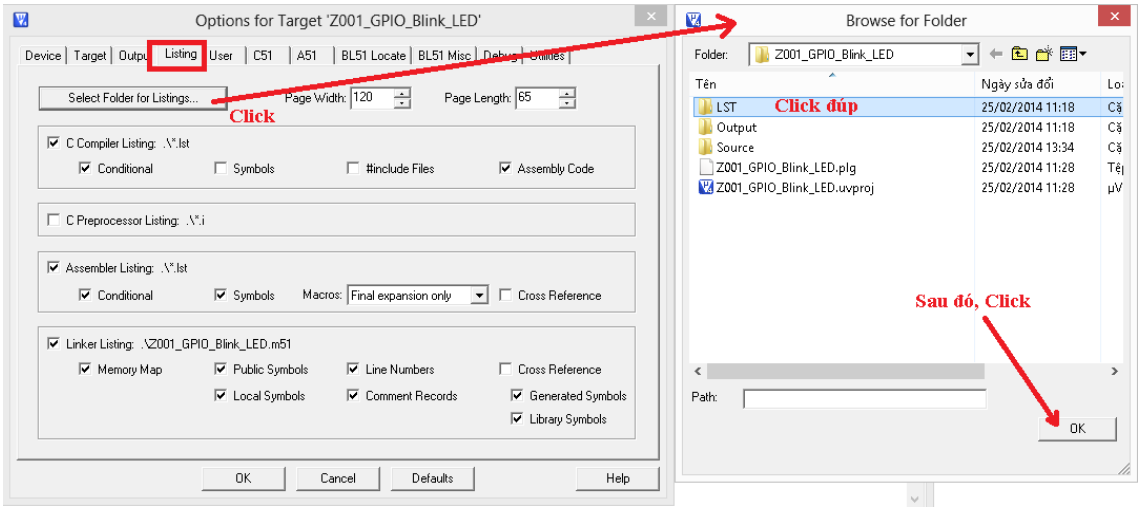
c. Thiết lập các thông số cho project để biên dịch

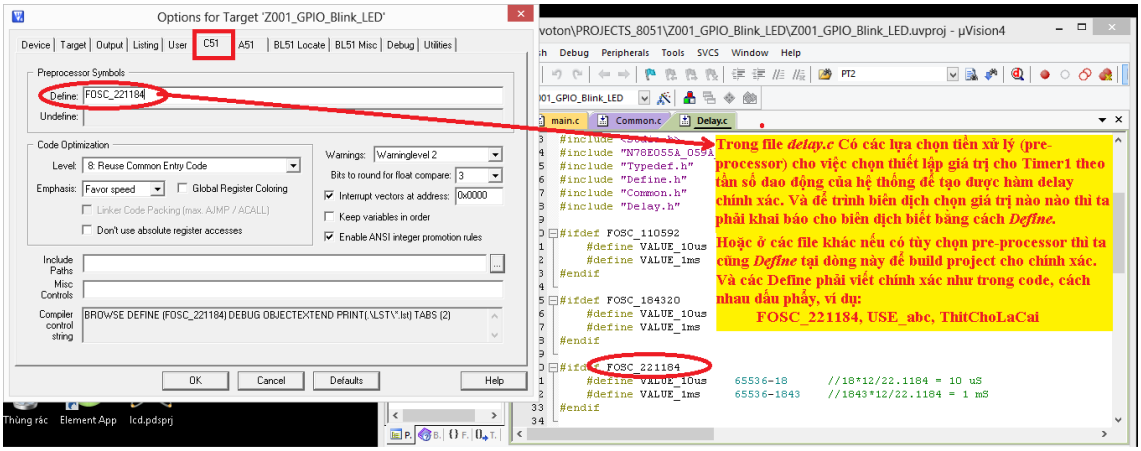




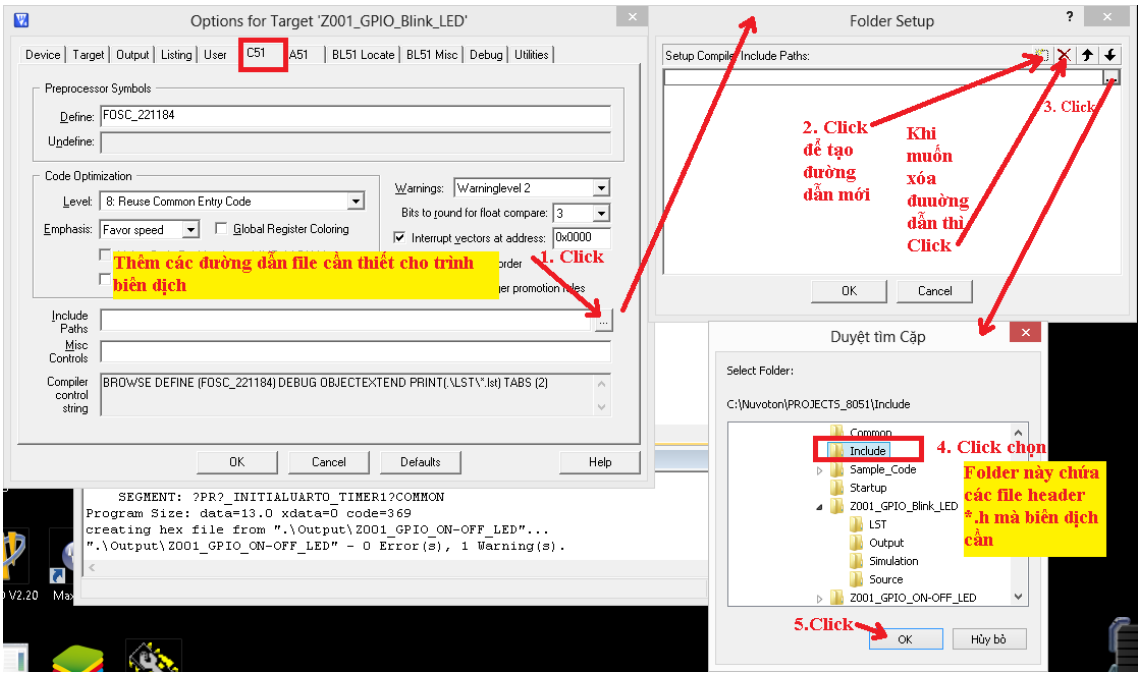

Options for Target 'Z001_GPIO_Blink_LED'
Device
Target
Output
Listing
User
C51
A51
BL51 Locate
BL51 Misc
Debug
Utilities
Nuvoton N78E055A
Xtal (MHz): 40.0
Use On-chip ROM (0x0-0x3FFF)
Memory Model: Small: variables in DATA
Code Rom Size: Large: 64K program
Operating system: None
Use On-chip XRAM (0x0-0x3FF)
Ta có thể để như mặc định. Nếu có dùng XRAM thì chọn thêm
Việc thay đổi giá trị Xtal không ảnh hưởng tới việc thiết lập tốc độ
cho N78E055A. Ta thiết lập qua công cụ ICP/ISP.
Eprom
Ram
Code Banking
Start:
End:
Banks:
Bank Area:
0x0000
0xFFFF
'far' memory type support
Save address extension SFR in interrupts
OK
Cancel
Defaults
Help

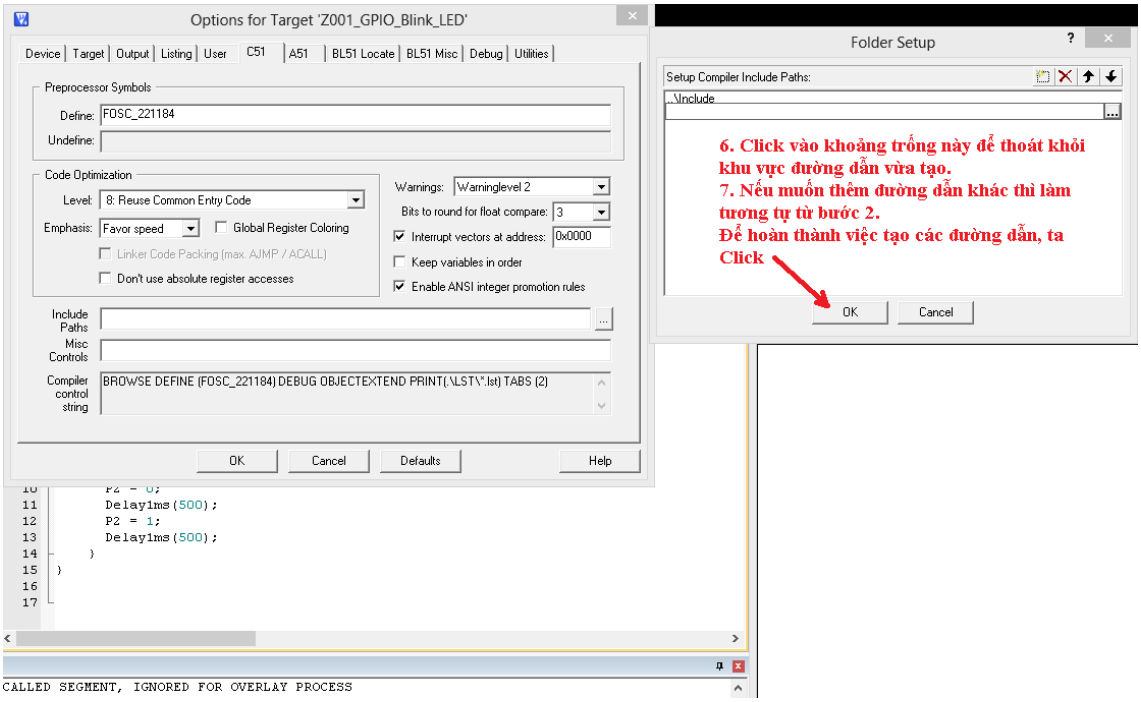

Options for Target 'Z001_GPIO_Blink_LED'
Select Folder for Objects...
Click
Name of Executable: Z001_GPIO_Blink_LED
Create Executable: .\Z001_GPIO_Blink_LED
Debug Information
Browse Information
Create HEX File
HEX Format: HEX-80
Create Library: .\Z001_GPIO_Blink_LED.LIB
Create Batch File
Browse for Folder
Folder: Z001_GPIO_Blink_LED
Tên
Ngày sửa đổi
LST
Output
Source
Z001_GPIO_Blink_LED.plg
Z001_GPIO_Blink_LED.uvproj
25/02/2014 11:18
25/02/2014 13:34
25/02/2014 11:28
Click đúp
sau đó, click
Path:
OK

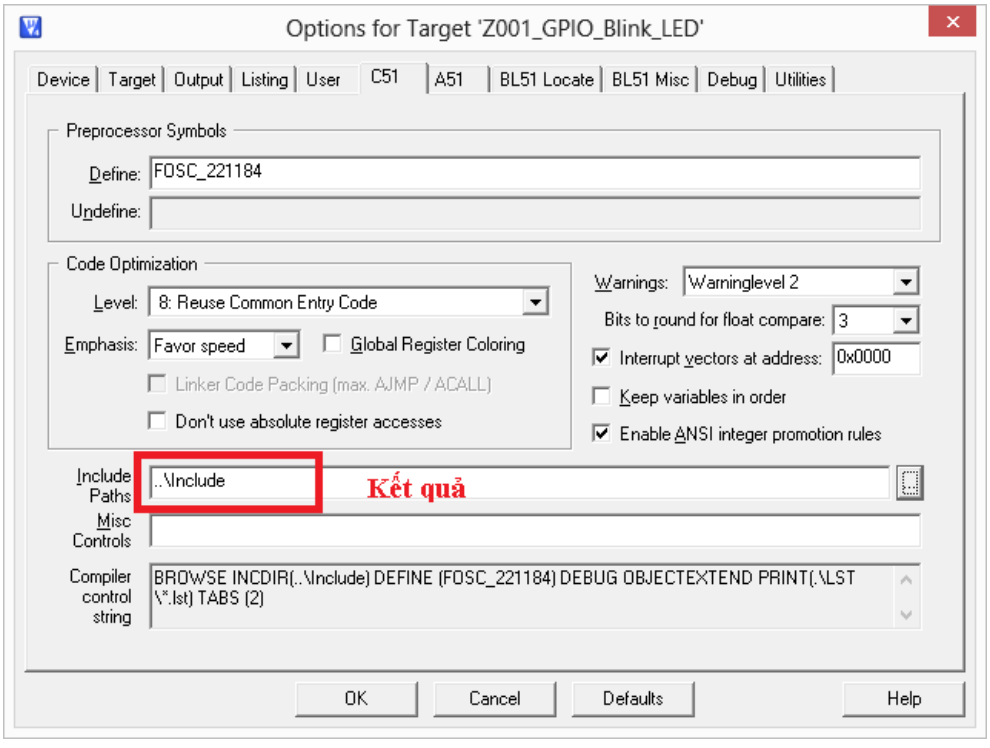

Options for Target 'Z001_GPIO_Blink_LED'
Select Folder for Listings...
Click
Page Width: 120
Page Length: 65
C Compiler Listing: .*.lst
Conditional
Symbols
#include Files
Assembly Code
C Preprocessor Listing: .*.i
Assembler Listing: .*.lst
Macros: Final expansion only
Cross Reference
Linker Listing: .\Z001_GPIO_Blink_LED.m51
Memory Map
Public Symbols
Line Numbers
Local Symbols
Comment Records
Generated Symbols
Library Symbols
Browse for Folder
Folder: Z001_GPIO_Blink_LED
Tên
Ngày sửa đổi
LST
Output
Source
Z001_GPIO_Blink_LED.plg
Z001_GPIO_Blink_LED.uvproj
25/02/2014 11:18
25/02/2014 13:34
25/02/2014 11:28
Click đúp
Sau đó, Click
Path:
OK

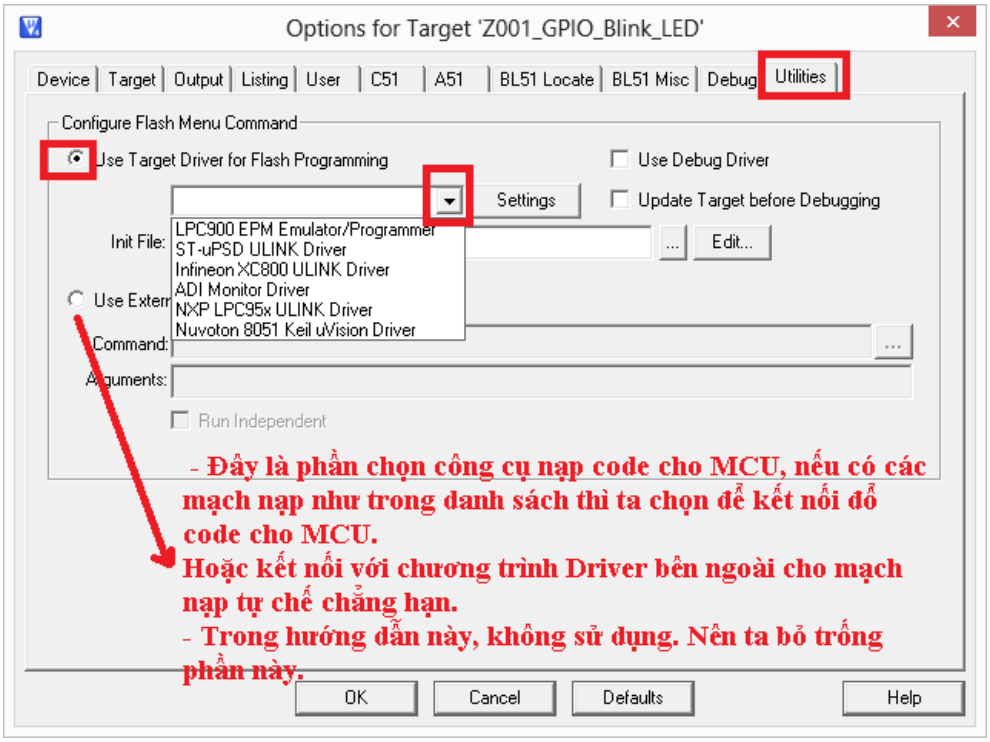


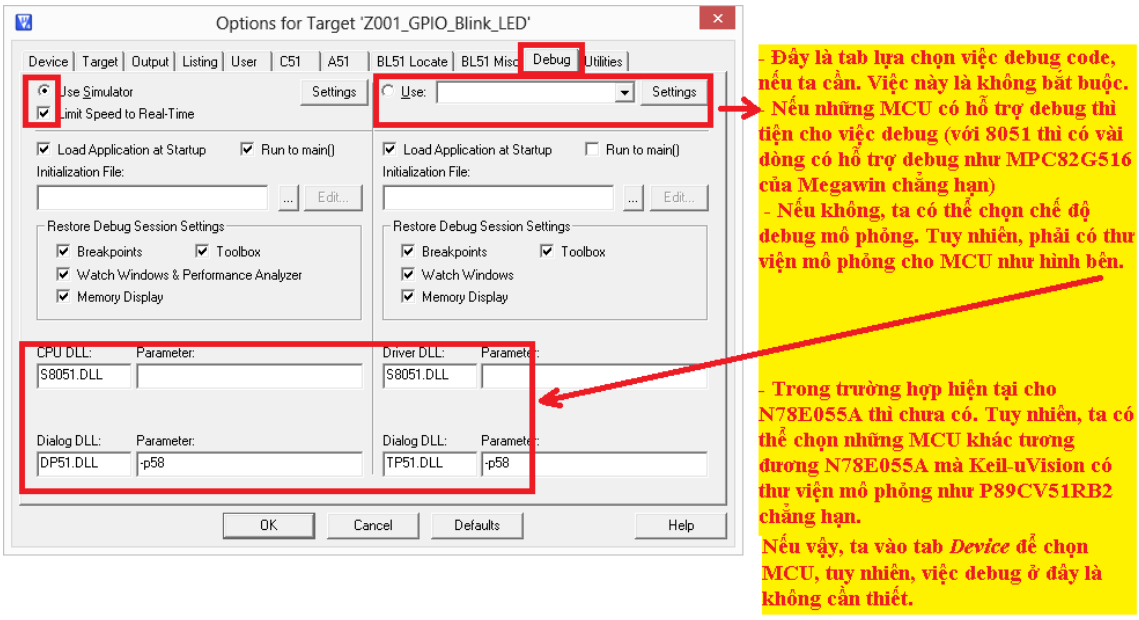
Lưu ý nhỏ:
- **Với N78E055A thì chương trình trong APROM mà ta sẽ code không có tác động được vào việc thiết lập tốc độ chip (việc này thông qua ICP/ISP) như một số dòng chip khác.**
- **Trong hướng dẫn này, sẽ dùng IRC 22.1184MHz vì vậy sẽ *Define* như ảnh bên để tạo hàm delay chính xác cho Example này. Và khi đổ code cho chip sẽ thiết lập tốc độ này qua ISP.**




Options for Target 'Z001_GPIO_Blink_LED'
Device | Target | Output | Listing | User | C51 | A51 | BL51 Locate | BL51 Misc | Debug | Utilities
Preprocessor Symbols
Define: FOSC_221184
Undefine:
Code Optimization
Level: 8: Reuse Common Entry Code
Emphasis: Favor speed
Global Register Coloring
Linker Code Packing (max. AJMP / ACALL)
Don't use absolute register accesses
Warnings: Warninglevel 2
Bits to round for float compare: 3
Interrupt vectors at address: 0x0000
Keep variables in order
Enable ANSI integer promotion rules
Include Paths
Misc Controls
Compiler control string
BROWSE DEFINE (FOSC_221184) DEBUG OBJECTEXTEND PRINT(.\LST*.lst) TABS (2)
OK
Cancel
Defaults
Help
Folder Setup
Setup Compiler Include Paths:
..\Include
6. Click vào khoảng trống này để thoát khỏi khu vực đường dẫn vừa tạo.
7. Nếu muốn thêm đường dẫn khác thì làm tương tự từ bước 2.
Để hoàn thành việc tạo các đường dẫn, ta Click
OK
Cancel
Delay1ms(500);
P2 = 1;
Delay1ms(500);
CALLED SEGMENT, IGNORED FOR OVERLAY PROCESS


Options for Target 'Z001_GPIO_Blink_LED'
Device | Target | Output | Listing | User | C51 | A51 | BL51 Locate | BL51 Misc | Debug | Utilities
Preprocessor Symbols
Define: FOSC_221184
Undefine:
Code Optimization
Level: 8: Reuse Common Entry Code
Emphasis: Favor speed
Global Register Coloring
Linker Code Packing (max. AJMP / ACALL)
Don't use absolute register accesses
Warnings: Warninglevel 2
Bits to round for float compare: 3
Interrupt vectors at address: 0x0000
Keep variables in order
Enable ANSI integer promotion rules
Include Paths
..\Include
Kết quả
Misc Controls
Compiler control string
BROWSE INCDIR(..\Include) DEFINE (FOSC_221184) DEBUG OBJECTEXTEND PRINT(.\LST*.lst) TABS (2)
OK
Cancel
Defaults
Help

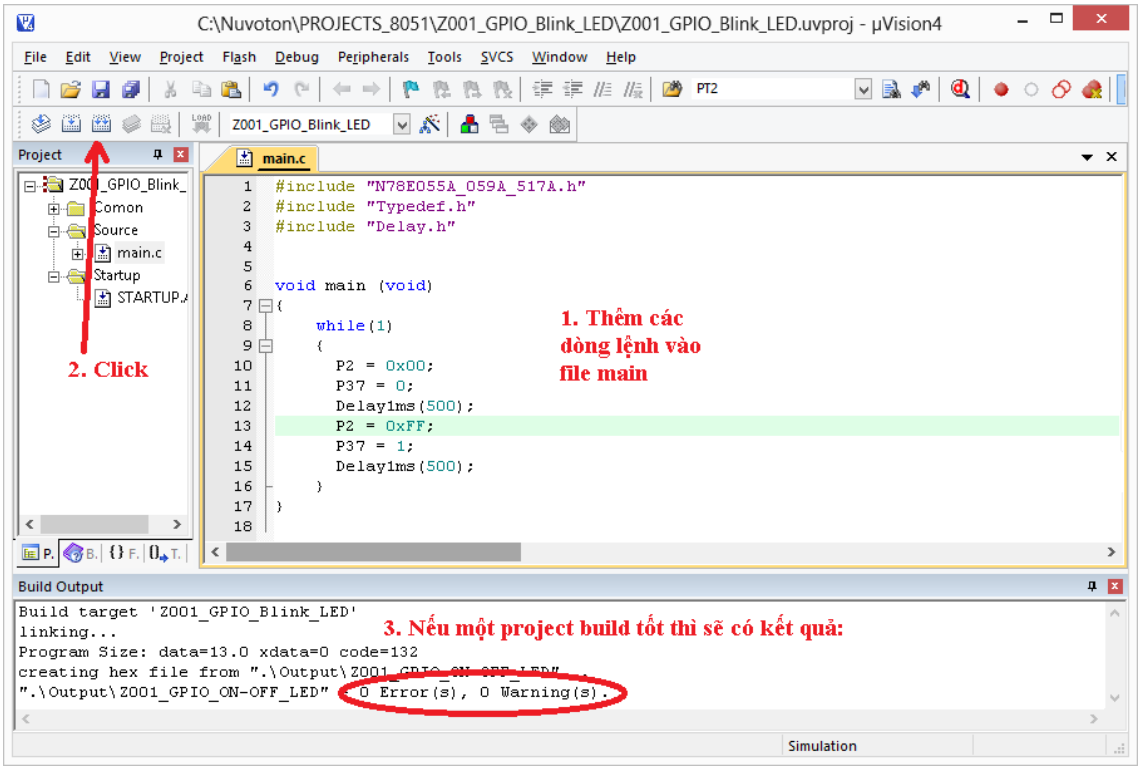

Options for Target 'Z001_GPIO_Blink_LED'
Device
Target
Output
Listing
User
C51
A51
BL51 Locate
BL51 Misc
Debug
Utilities
Configure Flash Menu Command
Use Target Driver for Flash Programming
Use Debug Driver
Settings
Update Target before Debugging
Init File:
Edit...
LPC900 EPM Emulator/Programmer
ST-uPSD ULINK Driver
Infineon XC800 ULINK Driver
ADI Monitor Driver
NXP LPC95x ULINK Driver
Nuvoton 8051 Keil uVision Driver
Use Extern
Command:
Arguments:
Run Independent
- Đây là phần chọn công cụ nạp code cho MCU, nếu có các mạch nạp như trong danh sách thì ta chọn để kết nối đổ code cho MCU.
Hoặc kết nối với chương trình Driver bên ngoài cho mạch nạp tự chế chẳng hạn.
- Trong hướng dẫn này, không sử dụng. Nên ta bỏ trống phần này.
OK
Cancel
Defaults
Help

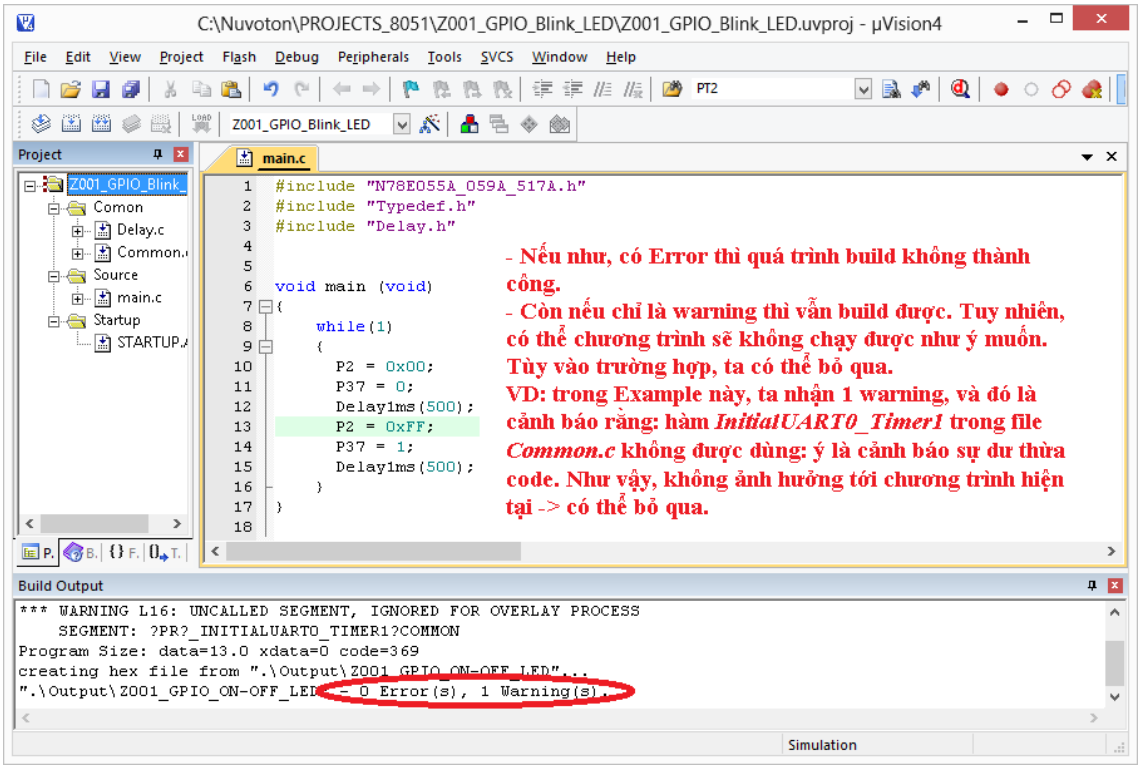

Options for Target 'Z001_GPIO_Blink_LED'
Device
Target
Output
Listing
User
C51
A51
BL51 Locate
BL51 Misc
Debug
Utilities
Use Simulator
Limit Speed to Real-Time
Settings
Use:
Load Application at Startup
Run to main()
Initialization File:
Edit...
Restore Debug Session Settings
Breakpoints
Toolbox
Watch Windows & Performance Analyzer
Watch Windows
Memory Display
CPU DLL:
Parameter:
S8051.DLL
Driver DLL:
Dialog DLL:
DP51.DLL
-p58
TP51.DLL
OK
Cancel
Defaults
Help
- Đây là tab lựa chọn việc debug code, nếu ta cần. Việc này là không bắt buộc.
- Nếu những MCU có hỗ trợ debug thì tiện cho việc debug (với 8051 thì có vài dòng có hỗ trợ debug như MPC82G516 của Megawin chẳng hạn)
- Nếu không, ta có thể chọn chế độ debug mô phỏng. Tuy nhiên, phải có thư viện mô phỏng cho MCU như hình bên.
- Trong trường hợp hiện tại cho N78E055A thì chưa có. Tuy nhiên, ta có thể chọn những MCU khác tương đương N78E055A mà Keil-uVision có thư viện mô phỏng như P89CV51RB2 chẳng hạn.
Nếu vậy, ta vào tab *Device* để chọn MCU, tuy nhiên, việc debug ở đây là không cần thiết.

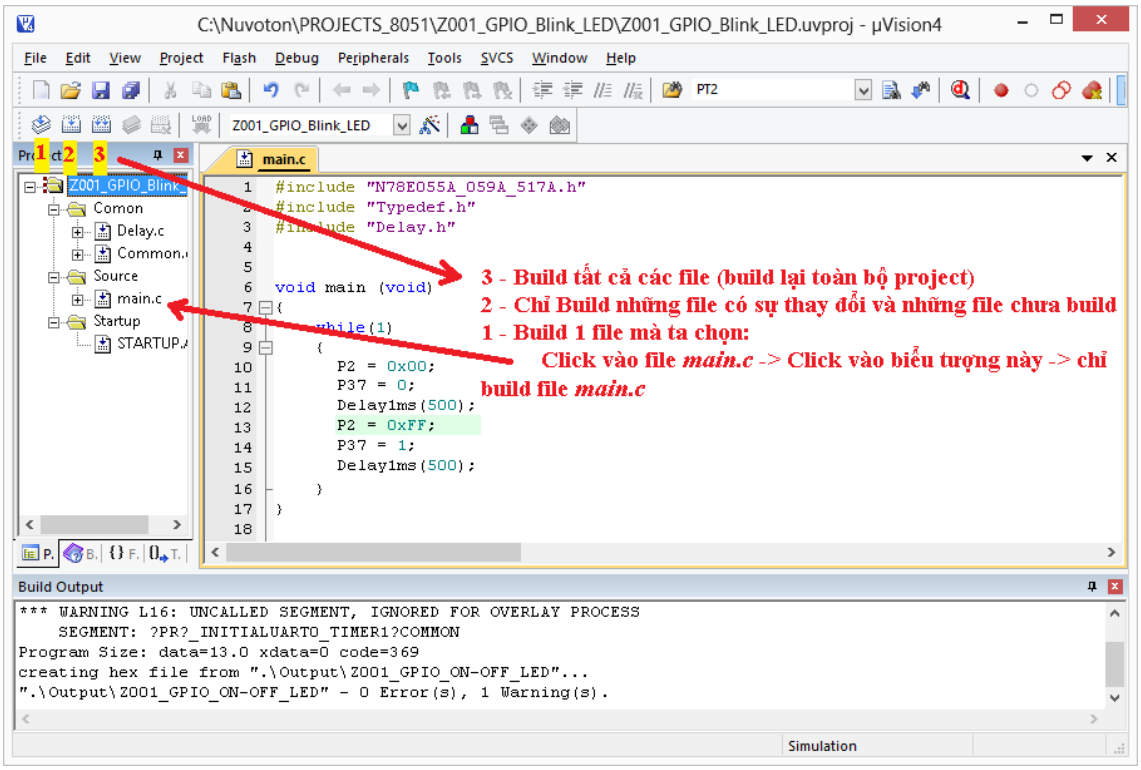
d. Build thử một project

- Bây giờ, ta sẽ code vài dòng để build thử một project:
  Thêm những dòng lệnh dưới vào file ***main.c***

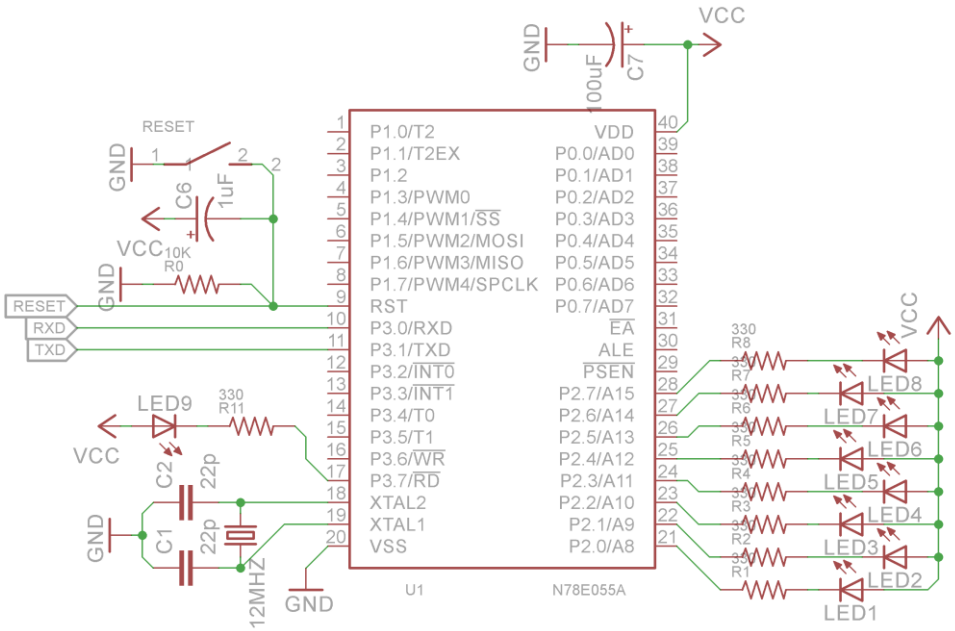
```
#include "N78E055A_059A_517A.h"   //Define register of MCU N78E055A/059A/517A
#include "Typedef.h"          //define type for Delay function in Delay.c
#include "Delay.h"            //header of Delay.c


void main (void)
{
   while(1)
   {
            P2 = 0x00;              //PORT 2 is low level
            P37 = 0;                //Pin 7 of Port 3 is low level
            Delay1ms(500);//Delay 500 mini second
            P2 = 0xFF;              //PORT 2 is high level
            P37 = 1;                //Pin 7 of Port 3 is high level
            Delay1ms(500);//Delay 500 mini second
   }
}
```




C:\Nuvoton\PROJECTS_8051\Z001_GPIO_Blink_LED\Z001_GPIO_Blink_LED.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Z001_GPIO_Blink_LED
Project
Z001_GPIO_Blink
Comon
Delay.c
Common.
Source
main.c
Startup
STARTUP.
main.c
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Delay.h"
void main (void)
{
while(1)
{
P2 = 0x00;
P37 = 0;
Delay1ms(500);
P2 = 0xFF;
P37 = 1;
Delay1ms(500);
}
}
- Nếu như, có Error thì quá trình build không thành công.
- Còn nếu chỉ là warning thì vẫn build được. Tuy nhiên, có thể chương trình sẽ không chạy được như ý muốn. Tùy vào trường hợp, ta có thể bỏ qua.
VD: trong Example này, ta nhận 1 warning, và đó là cảnh báo rằng: hàm InitialUART0_Timer1 trong file Common.c không được dùng: ý là cảnh báo sự dư thừa code. Như vậy, không ảnh hưởng tới chương trình hiện tại -> có thể bỏ qua.
Build Output
*** WARNING L16: UNCALLED SEGMENT, IGNORED FOR OVERLAY PROCESS
SEGMENT: ?PR?_INITIALUART0_TIMER1?COMMON
Program Size: data=13.0 xdata=0 code=369
creating hex file from ".\Output\Z001_GPIO_ON-OFF_LED"...
".\Output\Z001_GPIO_ON-OFF_LED" - 0 Error(s), 1 Warning(s).
Simulation


C:\Nuvoton\PROJECTS_8051\Z001_GPIO_Blink_LED\Z001_GPIO_Blink_LED.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Z001_GPIO_Blink_LED
1 2 3
Z001_GPIO_Blink
Comon
Delay.c
Common.
Source
main.c
Startup
STARTUP.
main.c
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Delay.h"
void main (void)
{
while(1)
{
P2 = 0x00;
P37 = 0;
Delay1ms(500);
P2 = 0xFF;
P37 = 1;
Delay1ms(500);
}
}
3 - Build tất cả các file (build lại toàn bộ project)
2 - Chỉ Build những file có sự thay đổi và những file chưa build
1 - Build 1 file mà ta chọn:
Click vào file main.c -> Click vào biểu tượng này -> chỉ build file main.c
Build Output
*** WARNING L16: UNCALLED SEGMENT, IGNORED FOR OVERLAY PROCESS
SEGMENT: ?PR?_INITIALUART0_TIMER1?COMMON
Program Size: data=13.0 xdata=0 code=369
creating hex file from ".\Output\Z001_GPIO_ON-OFF_LED"...
".\Output\Z001_GPIO_ON-OFF_LED" - 0 Error(s), 1 Warning(s).
Simulation

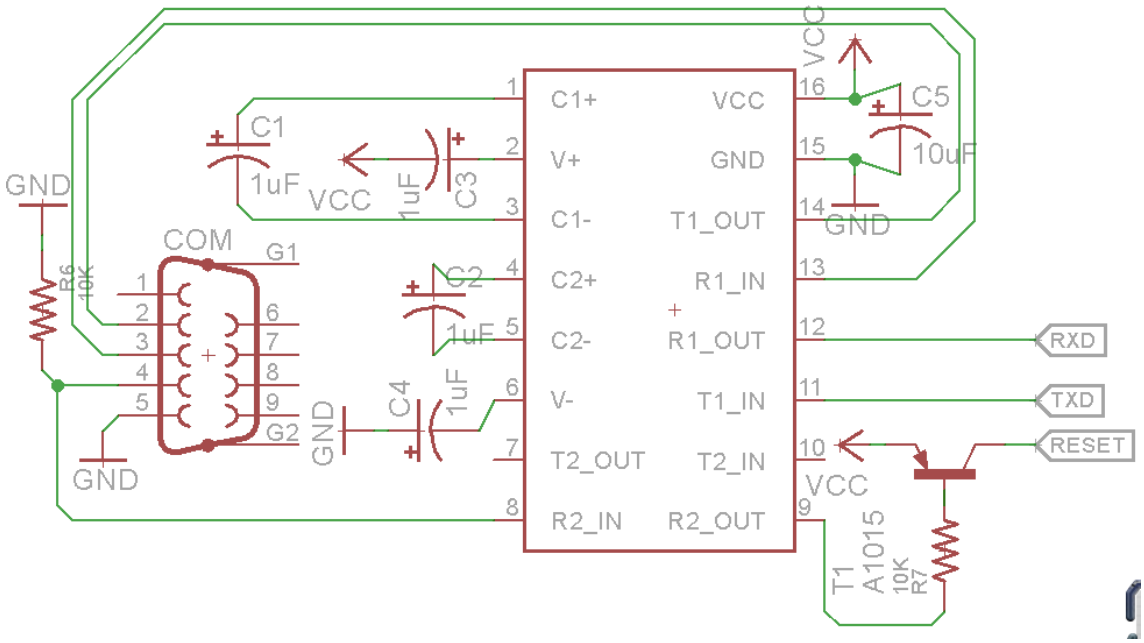
e. Ráp mạch điện để test



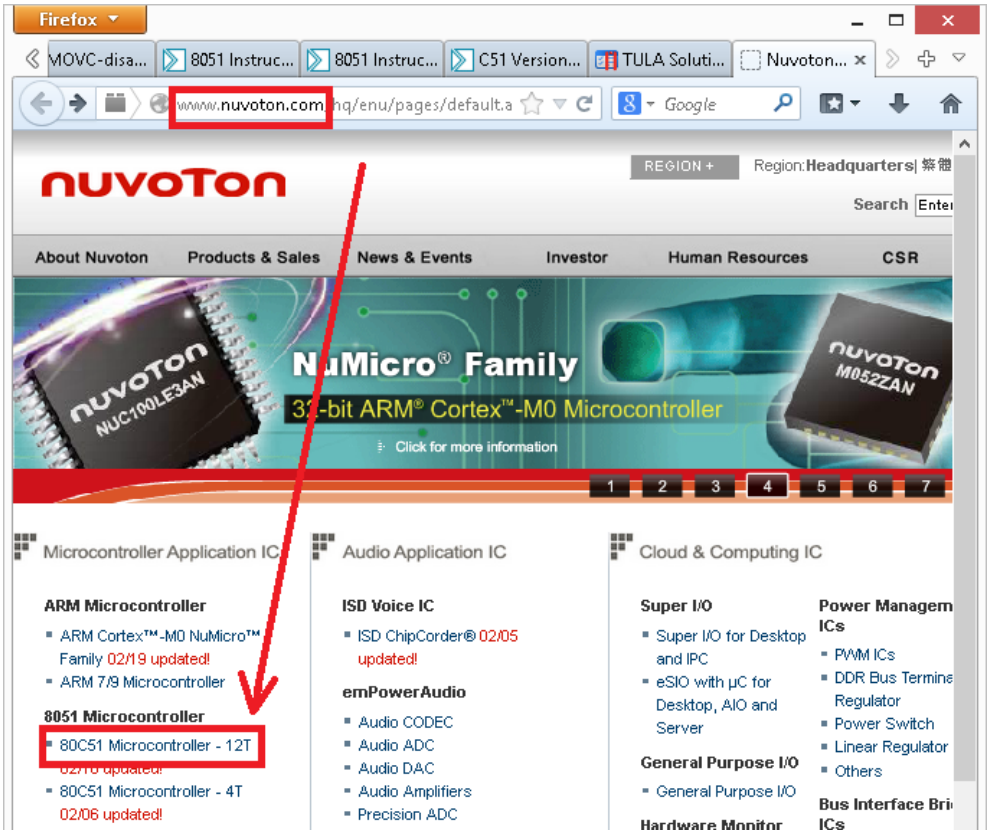
- Ở lần đầu, ta cần có thạch anh để nạp. Khi ta thiết lập được việc dùng thạch anh nội thì từ lần sau không cần tới thạch anh nữa.
- Phần mạch Reset có thể cần hoặc không cần. Bên trong MCU đã có sẵn điện trở kéo xuống (pull-down) .

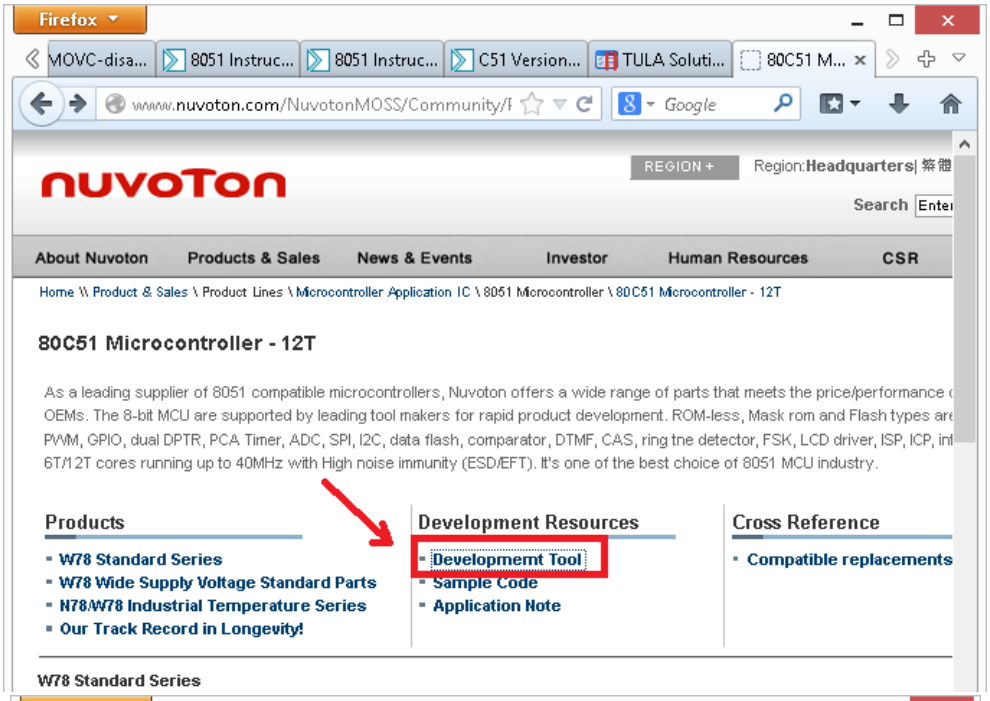
f. Nạp code cho N78E055A theo phương thức ISP qua cổng COM

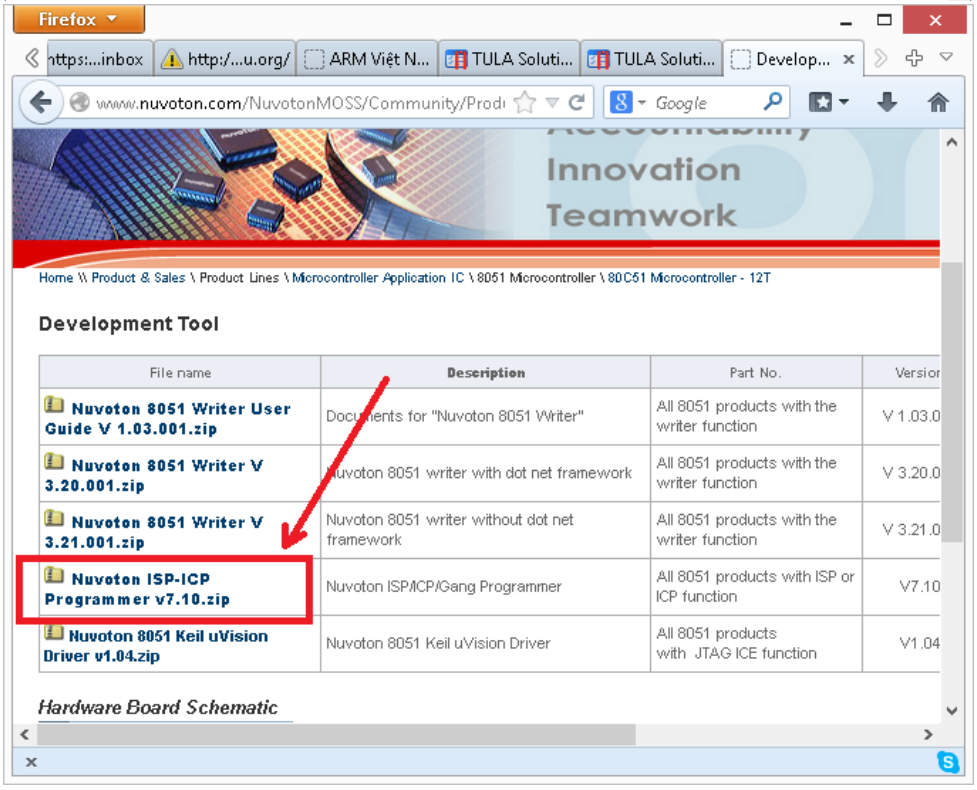
- N78E055A hỗ trợ phương thức nạp ICP, ISP. Ta co thể dùng mạch nạp ISP-ICP Programmer của Nuvoton hoặc NU-Gang, những mạch nạp, Writer của các hãng khác cho dòng 8051 đều được.
- Ở hướng dẫn này, sẽ dùng phương thức nạp ISP qua cổng COM để giảm thiểu chi phí cho người dùng nghiên cứu phát triển, học tập dòng MCU 8051 này.
- N78E055A hỗ trợ nạp ISP qua cổng COM (RS-232), chỉ với Max232 cùng một số linh kiện khác là ta đã có một mạch nạp giá rẻ và tiện lợi.
- Ta có sơ đồ nguyên lý:

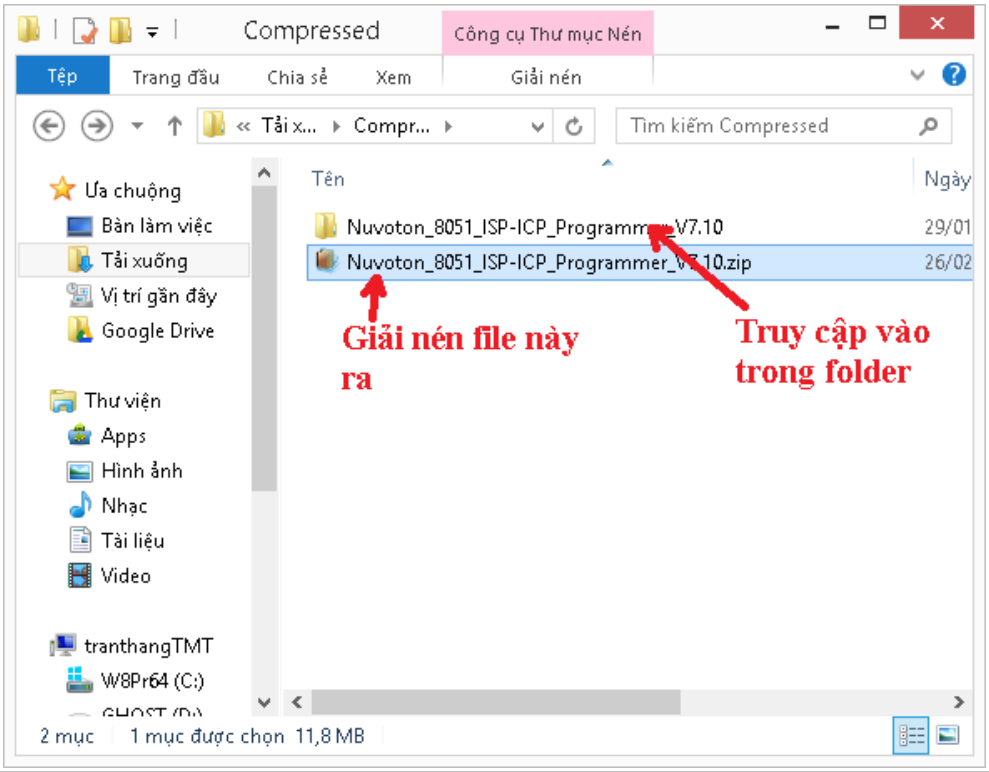


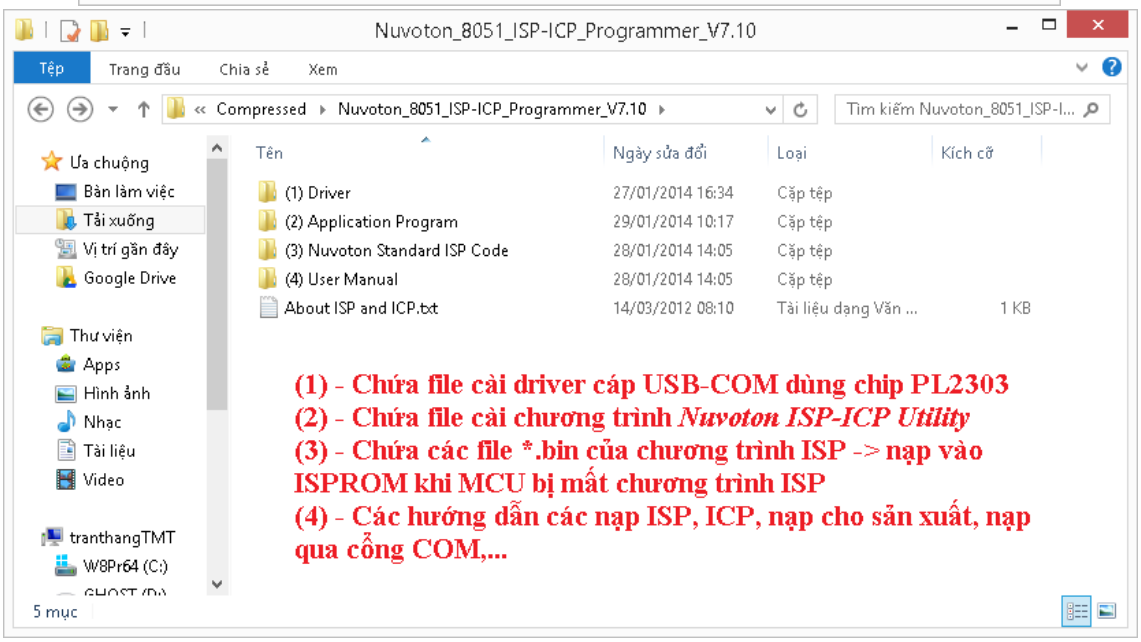
- Ta downoad phần phềm hỗ trợ nạp ICP-ISP của nuvoton về :

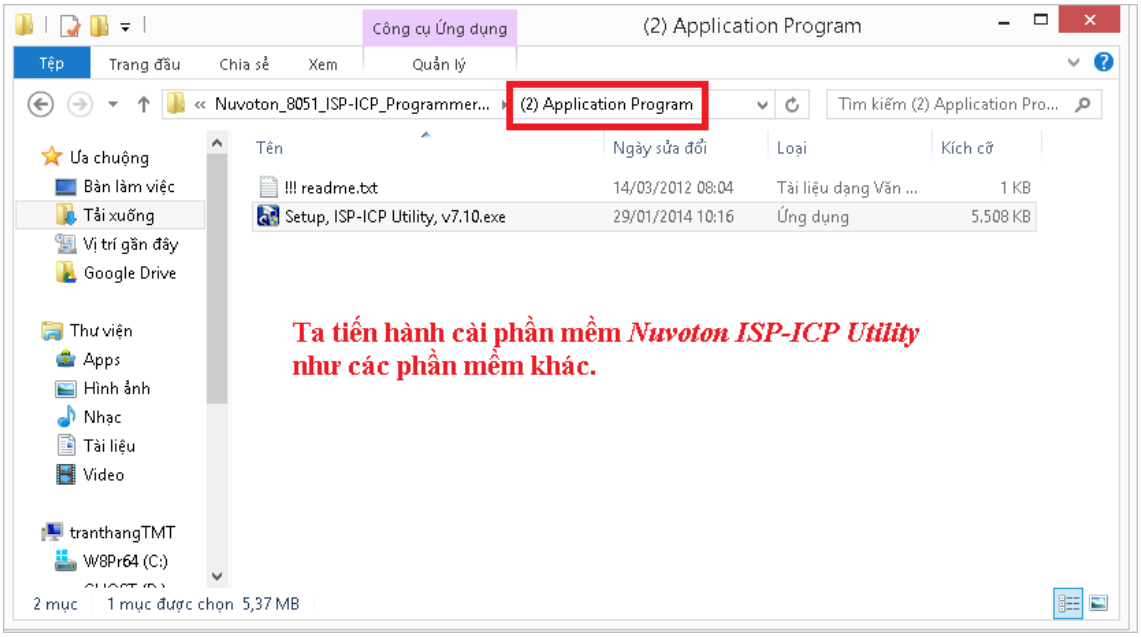



Firefox
www.nuvoton.com/NuvotonMOSS/Community/I
nuvoTon
REGION +
Region:Headquarters|
Search
About Nuvoton
Products & Sales
News & Events
Investor
Human Resources
CSR
Home \\ Product & Sales \ Product Lines \ Microcontroller Application IC \ 8051 Microcontroller \ 80C51 Microcontroller - 12T
80C51 Microcontroller - 12T
As a leading supplier of 8051 compatible microcontrollers, Nuvoton offers a wide range of parts that meets the price/performance
OEMs. The 8-bit MCU are supported by leading tool makers for rapid product development. ROM-less, Mask rom and Flash types ar
PWM, GPIO, dual DPTR, PCA Timer, ADC, SPI, I2C, data flash, comparator, DTMF, CAS, ring tne detector, FSK, LCD driver, ISP, ICP, in
6T/12T cores running up to 40MHz with High noise immunity (ESD/EFT). It's one of the best choice of 8051 MCU industry.
Products
W78 Standard Series
W78 Wide Supply Voltage Standard Parts
N78/W78 Industrial Temperature Series
Our Track Record in Longevity!
Development Resources
Development Tool
Sample Code
Application Note
Cross Reference
Compatible replacements
W78 Standard Series


Firefox
www.nuvoton.com/NuvotonMOSS/Community/Prod
Innovation
Teamwork
Home \\ Product & Sales \ Product Lines \ Microcontroller Application IC \ 8051 Microcontroller \ 80C51 Microcontroller - 12T
Development Tool
File name
Description
Part No.
Nuvoton 8051 Writer User Guide V 1.03.001.zip
Documents for "Nuvoton 8051 Writer"
All 8051 products with the writer function
Nuvoton 8051 Writer V 3.20.001.zip
Nuvoton 8051 writer with dot net framework
All 8051 products with the writer function
Nuvoton 8051 Writer V 3.21.001.zip
Nuvoton 8051 writer without dot net framework
All 8051 products with the writer function
Nuvoton ISP-ICP Programmer v7.10.zip
Nuvoton ISP/ICP/Gang Programmer
All 8051 products with ISP or ICP function
V7.10
Nuvoton 8051 Keil uVision Driver v1.04.zip
Nuvoton 8051 Keil uVision Driver
All 8051 products with JTAG ICE function
V1.04
Hardware Board Schematic

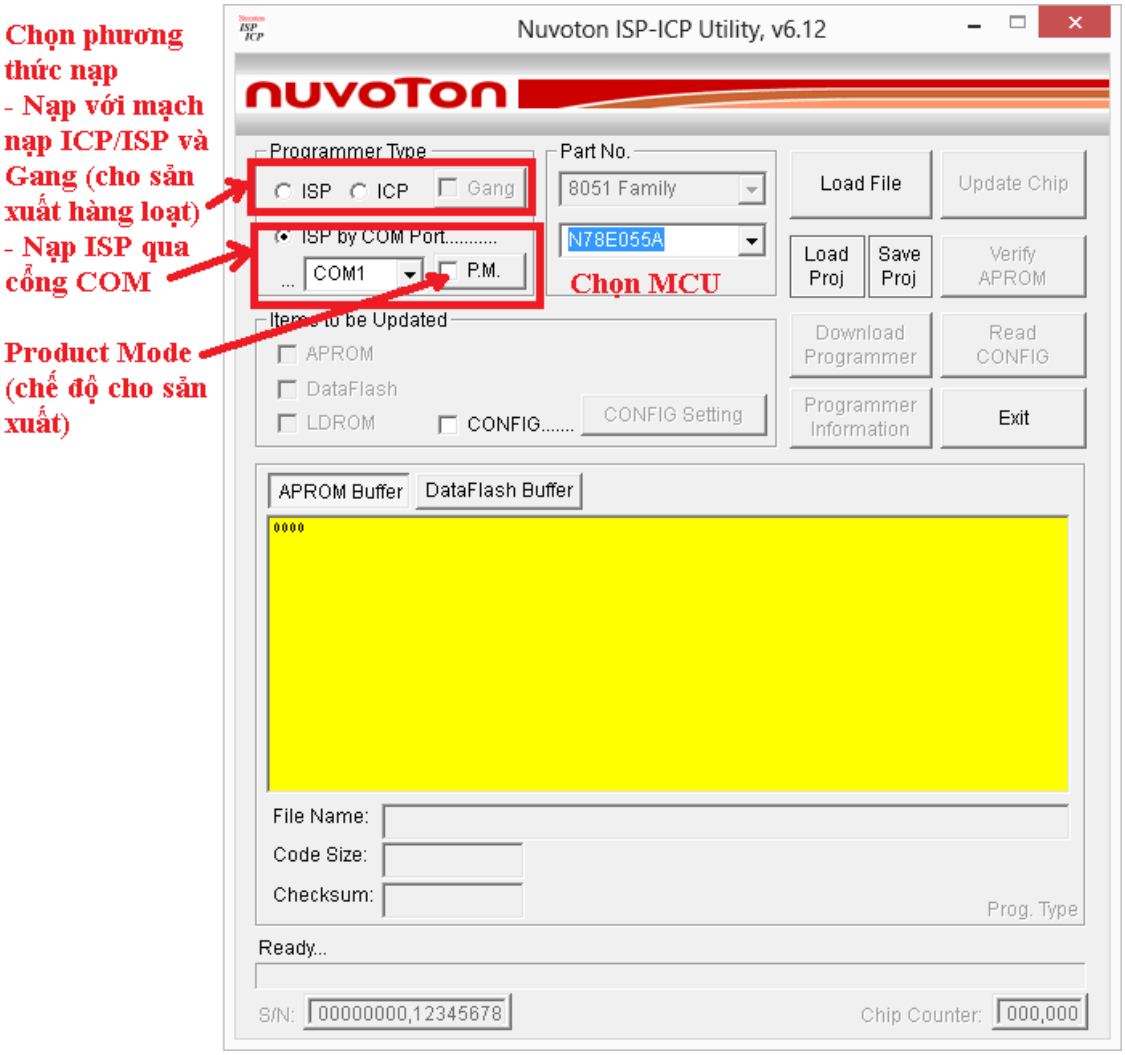

Compressed
Công cụ Thư mục Nén
Tệp
Trang đầu
Chia sẻ
Xem
Giải nén
« Tải x... ▸ Compr... ▸
Tìm kiếm Compressed
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
Tên
Ngày
Nuvoton_8051_ISP-ICP_Programmer_V7.10
29/01
Nuvoton_8051_ISP-ICP_Programmer_V7.10.zip
26/02
Giải nén file này ra
Truy cập vào trong folder
2 mục
1 mục được chọn 11,8 MB

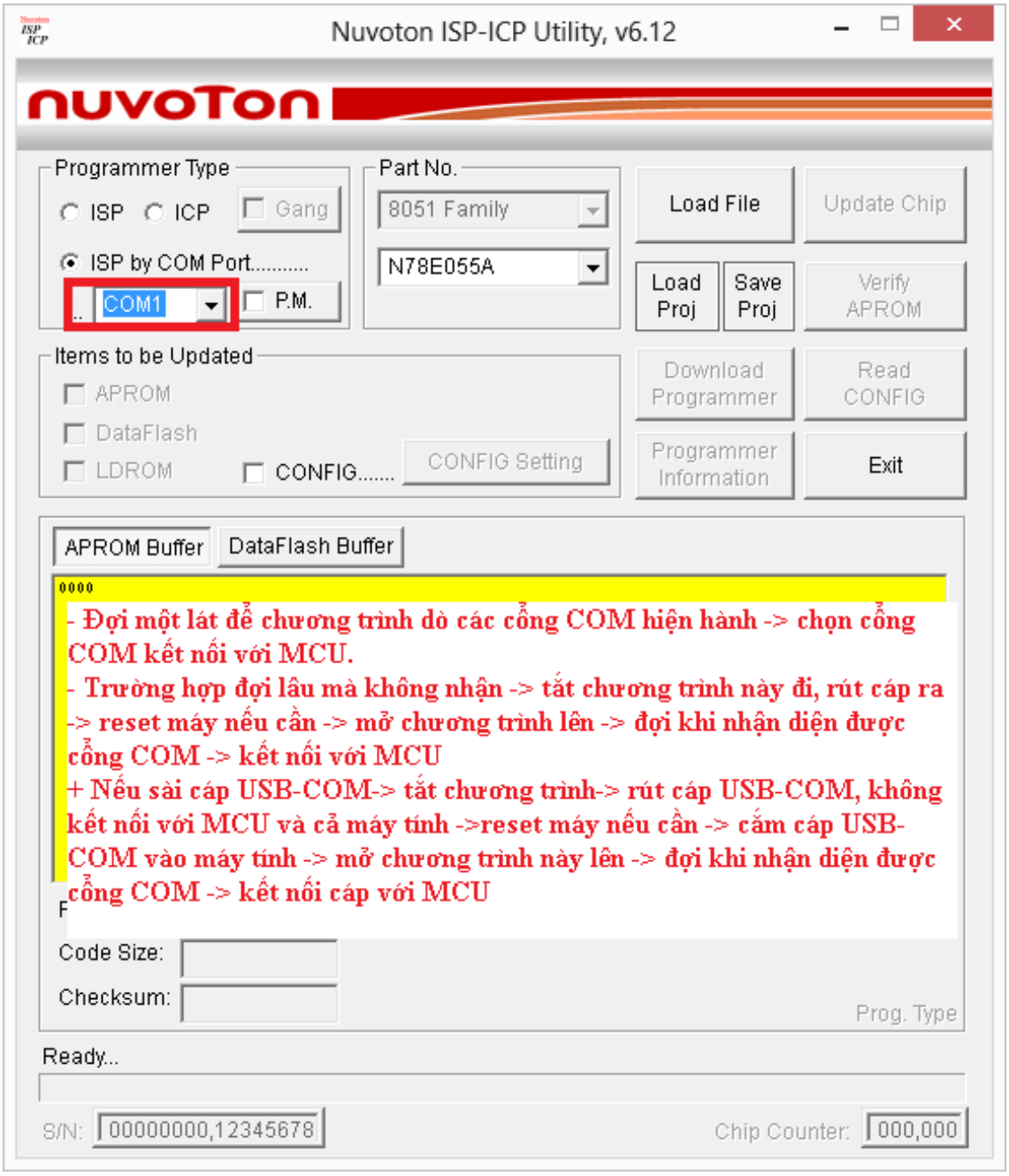

Nuvoton_8051_ISP-ICP_Programmer_V7.10
Tệp
Trang đầu
Chia sẻ
Xem
« Compressed ▸ Nuvoton_8051_ISP-ICP_Programmer_V7.10 ▸
Tìm kiếm Nuvoton_8051_ISP-I...
Ưa chuộng
Bàn làm việc
Tải xuống
Vị trí gần đây
Google Drive
Thư viện
Apps
Hình ảnh
Nhạc
Tài liệu
Video
tranthangTMT
W8Pr64 (C:)
Tên
Ngày sửa đổi
Loại
Kích cỡ
(1) Driver
27/01/2014 16:34
Cặp tệp
(2) Application Program
29/01/2014 10:17
Cặp tệp
(3) Nuvoton Standard ISP Code
28/01/2014 14:05
Cặp tệp
(4) User Manual
28/01/2014 14:05
Cặp tệp
About ISP and ICP.txt
14/03/2012 08:10
Tài liệu dạng Văn ...
1 KB
(1) - Chứa file cài driver cáp USB-COM dùng chip PL2303
(2) - Chứa file cài chương trình Nuvoton ISP-ICP Utility
(3) - Chứa các file *.bin của chương trình ISP -> nạp vào ISPROM khi MCU bị mất chương trình ISP
(4) - Các hướng dẫn các nạp ISP, ICP, nạp cho sản xuất, nạp qua cổng COM,...
5 mục

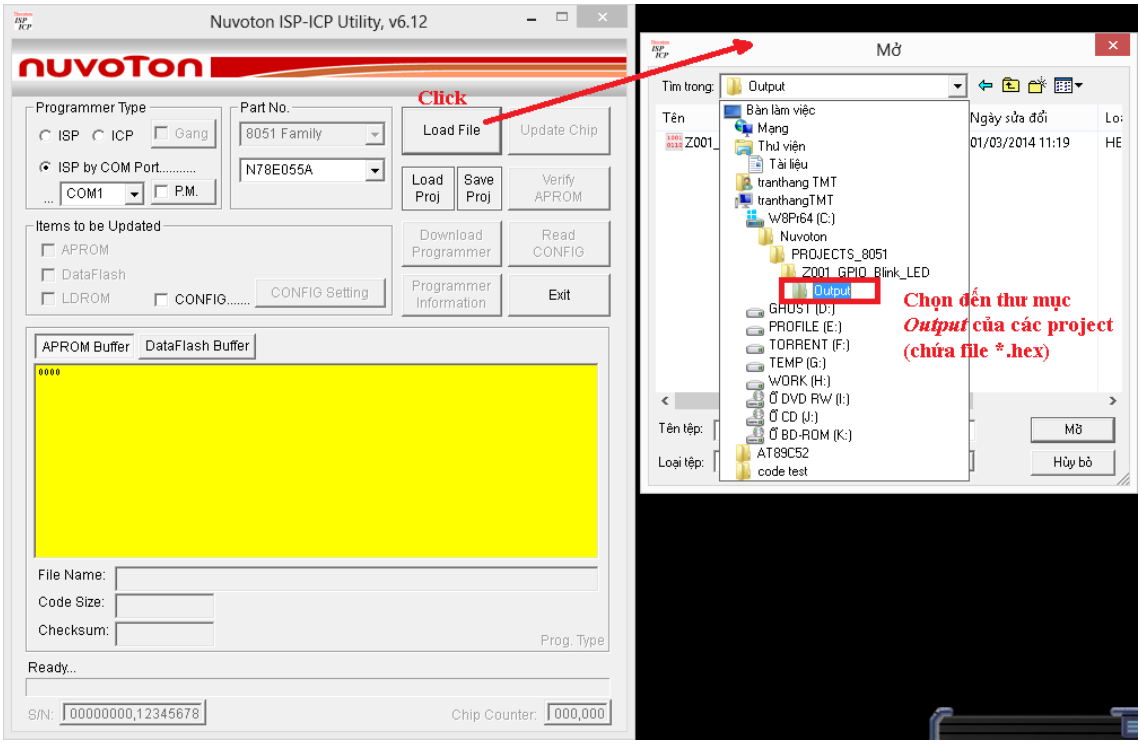


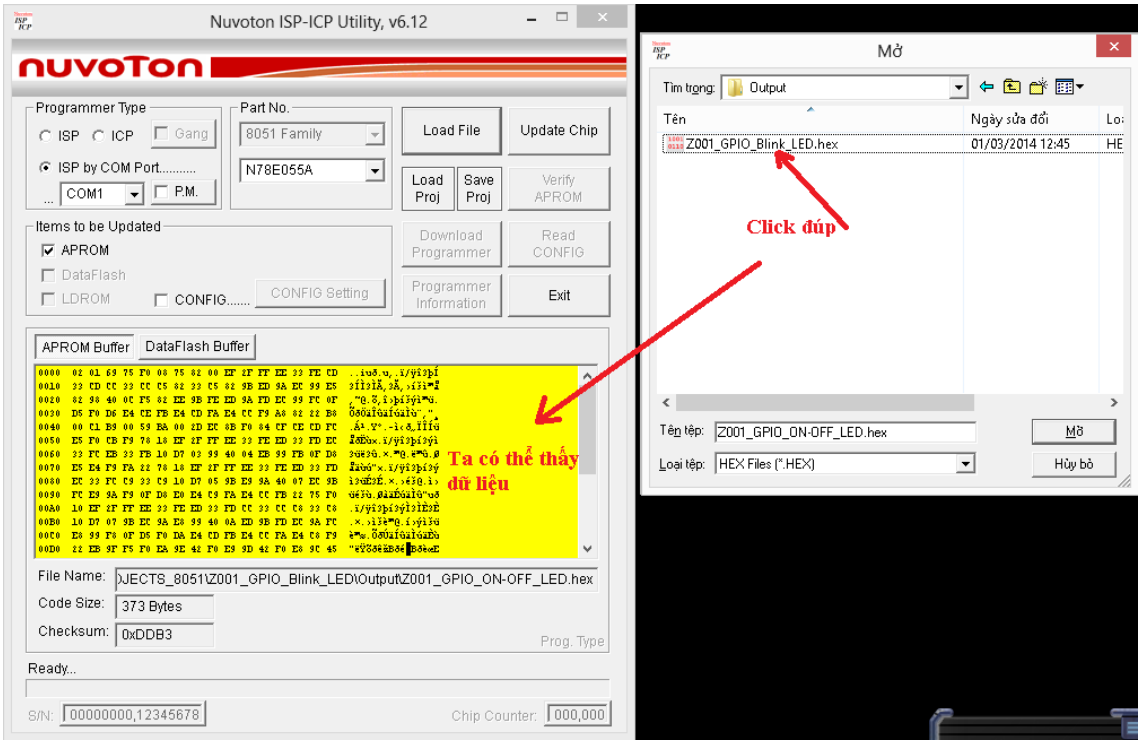
- Chú ý : ban đầu, khi chip mới từ nhà máy ra đã có sẵn chương trình trong ISPROM (gọi là ‘’Nuvoton Standard ISP Code’’) cùng với các thiết lập sẵn như: hoạt động với thạch anh ngoài, bảo mật code, chạy lõi 8051 @ 12T,… Giả sử bị mất ISP Code, ta có thể nạp lại. Vào folder *(4) User Manual* để xem hướng dẫn cách nạp lại chương trình ISP (LDROM).
- Nếu dùng cáp chuyển đổi USB-COM thì cắm cáp trước, đợi một lát cho hệ điều hành nhận diện, sau đó mới mở chương trình *Nuvoton ISP-ICP Utility*.
- Click đúp vào icon của chương trình 

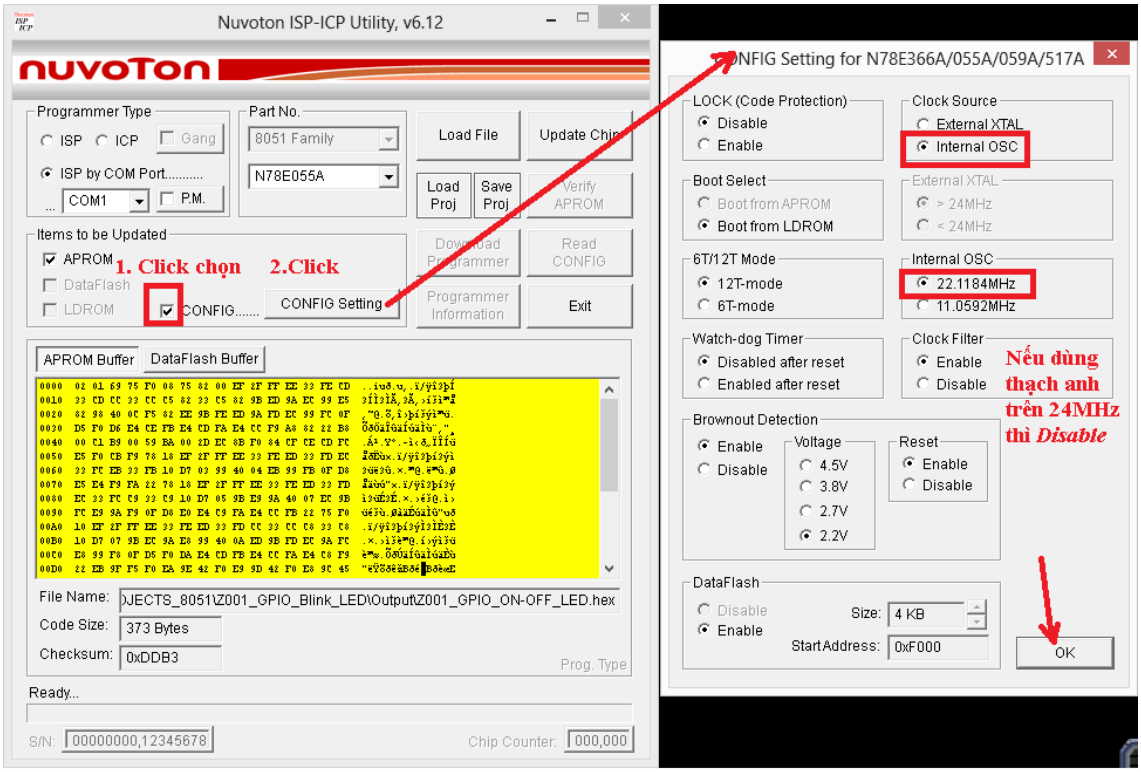
. Nếu hệ điều hành là Windows 7/8/8.1 thì mở bằng quyền administrator :

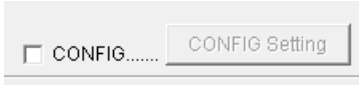

Chọn phương thức nạp
- Nạp với mạch nạp ICP/ISP và Gang (cho sản xuất hàng loạt)
- Nạp ISP qua cổng COM
Product Mode (chế độ cho sản xuất)
Nuvoton ISP-ICP Utility, v6.12
nuvoTon
Programmer Type
ISP
ICP
Gang
ISP by COM Port.....
COM1
P.M.
Part No.
8051 Family
N78E055A
Chọn MCU
Load File
Update Chip
Load Proj
Save Proj
Verify APROM
Download Programmer
Read CONFIG
Programmer Information
Exit
Items to be Updated
APROM
DataFlash
LDROM
CONFIG.....
CONFIG Setting
APROM Buffer
DataFlash Buffer
0000
File Name:
Code Size:
Checksum:
Prog. Type
Ready...
S/N:
00000000,12345678
Chip Counter:
000,000

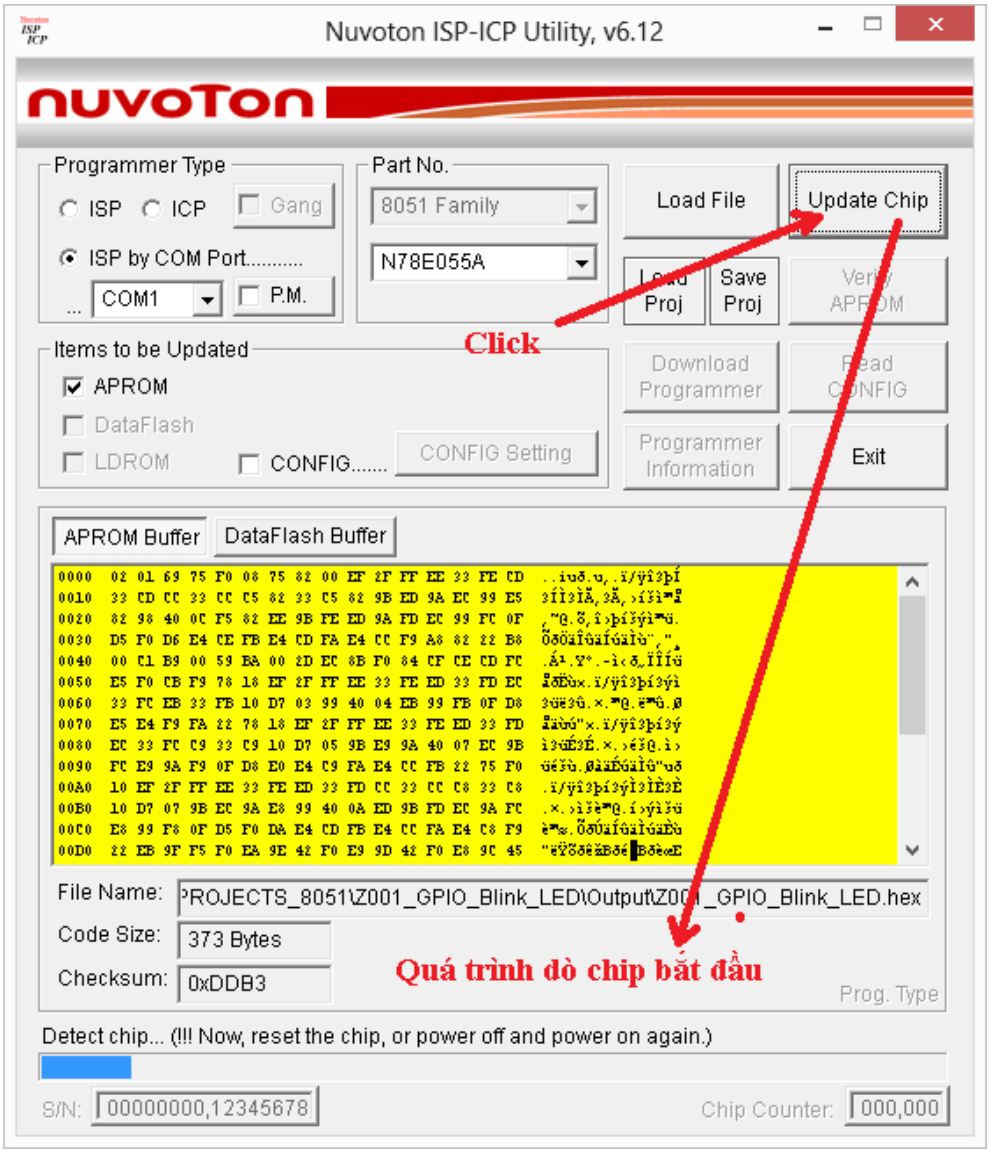

Nuvoton ISP-ICP Utility, v6.12
nuvoTon
Programmer Type
ISP
ICP
Gang
ISP by COM Port.....
COM1
P.M.
Part No.
8051 Family
N78E055A
Load File
Update Chip
Load Proj
Save Proj
Verify APROM
Items to be Updated
APROM
DataFlash
LDROM
CONFIG.....
CONFIG Setting
Download Programmer
Read CONFIG
Programmer Information
Exit
APROM Buffer
DataFlash Buffer
0000
- Đợi một lát để chương trình dò các cổng COM hiện hành -> chọn cổng COM kết nối với MCU.
- Trường hợp đợi lâu mà không nhận -> tắt chương trình này đi, rút cáp ra -> reset máy nếu cần -> mở chương trình lên -> đợi khi nhận diện được cổng COM -> kết nối với MCU
+ Nếu sài cáp USB-COM-> tắt chương trình-> rút cáp USB-COM, không kết nối với MCU và cả máy tính ->reset máy nếu cần -> cắm cáp USB-COM vào máy tính -> mở chương trình này lên -> đợi khi nhận diện được cổng COM -> kết nối cáp với MCU
Code Size:
Checksum:
Prog. Type
Ready...
S/N:
00000000,12345678
Chip Counter:
000,000


Nuvoton ISP-ICP Utility, v6.12
Click
Load File
Update Chip
Load Proj
Save Proj
Verify APROM
Download Programmer
Read CONFIG
Programmer Information
Exit
CONFIG Setting
APROM Buffer
DataFlash Buffer
File Name:
Code Size:
Checksum:
Prog. Type
Ready...
S/N:
00000000,12345678
Chip Counter:
000,000
Mở
Tìm trong:
Output
Tên
Ngày sửa đổi
01/03/2014 11:19
Bàn làm việc
Mạng
Thư viện
Tài liệu
tranthang TMT
W8Pr64 (C:)
Nuvoton
PROJECTS_8051
Z001_GPIO_Blink_LED
Output
GHOST (D:)
PROFILE (E:)
TORRENT (F:)
TEMP (G:)
WORK (H:)
DVD RW (I:)
CD (J:)
BD-ROM (K:)
AT89C52
code test
Tên tệp:
Loại tệp:
Hủy bỏ
Chọn đến thư mục Output của các project (chứa file *.hex)


Nuvoton ISP-ICP Utility, v6.12
APROM
Load File
Update Chip
Load Proj
Save Proj
Verify APROM
Download Programmer
Read CONFIG
Programmer Information
Exit
CONFIG Setting
APROM Buffer
DataFlash Buffer
File Name: JECTS_8051\Z001_GPIO_Blink_LED\Output\Z001_GPIO_ON-OFF_LED.hex
Code Size: 373 Bytes
Checksum: 0xDDB3
Prog. Type
Ready...
S/N:
00000000,12345678
Chip Counter:
000,000
Mở
Tìm trong:
Output
Tên
Ngày sửa đổi
Z001_GPIO_Blink_LED.hex
01/03/2014 12:45
Click đúp
Tên tệp:
Z001_GPIO_ON-OFF_LED.hex
Loại tệp:
HEX Files (*.HEX)
Hủy bỏ
Ta có thể thấy dữ liệu



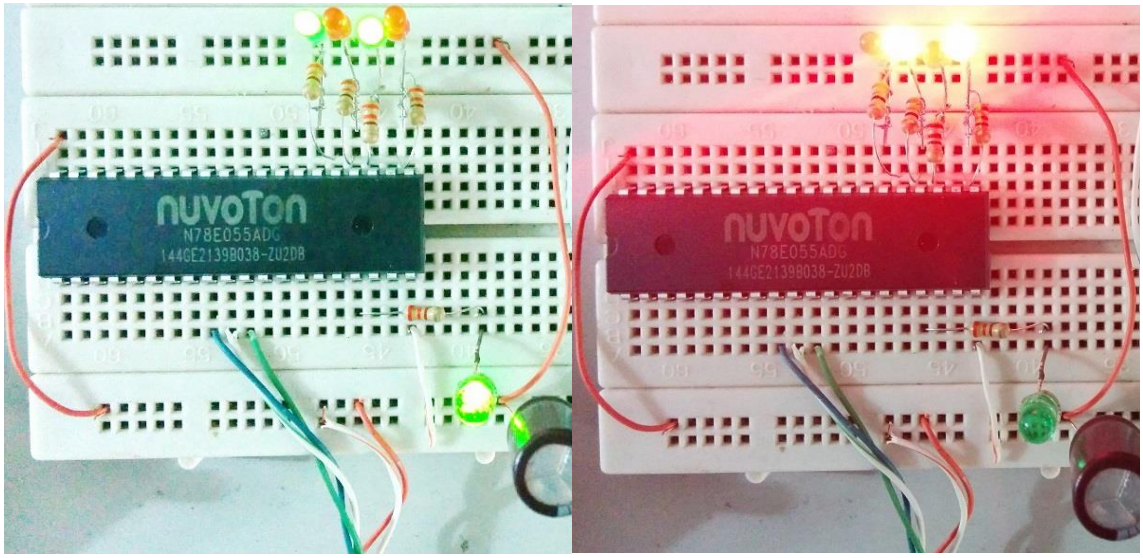
- Nếu không dùng IRC, dùng thạch anh ngoài -> chọn *External XTAL* và phải kết nối thạch anh cho MCU để nạp và chạy.
- ***Chú ý : Ở các ví dụ dưới dây sẽ sử dụng bộ dao động nội IRC của N78E055A, vì vậy, với lần nạp đầu tiên thì ta vẫn cần có thạch anh ngoài để có thể đổ code và thiết lập dùng IRC, từ lần sau ta có thể tháo thạch anh ra và chạy bình thường. Từ lần thứ 2 trở đi, nếu không có thiết lập gì cho các thông số trên thì ta nên bỏ chọn việc ghi thiết lập*** 


Nuvoton ISP-ICP Utility, v6.12
nuvoTon
Programmer Type
ISP
ICP
Gang
ISP by COM Port.....
COM1
P.M.
Part No.
8051 Family
N78E055A
Load File
Update Chip
Load Proj
Save Proj
Verify APROM
Items to be Updated
APROM
DataFlash
LDROM
CONFIG.....
CONFIG Setting
Download Programmer
Read CONFIG
Programmer Information
Exit
Click
APROM Buffer
DataFlash Buffer
File Name:
PROJECTS_8051\Z001_GPIO_Blink_LED\Output\Z001_GPIO_Blink_LED.hex
Code Size:
373 Bytes
Checksum:
0xDDB3
Quá trình dò chip bắt đầu
Prog. Type
Detect chip... (!!! Now, reset the chip, or power off and power on again.)
S/N:
00000000,12345678
Chip Counter:
000,000


Nuvoton ISP-ICP Utility, v6.12
nuvoTon
Programmer Type
ISP
ICP
Gang
ISP by COM Port.....
COM1
P.M.
Part No.
8051 Family
N78E055A
Load File
Update Chip
Load Proj
Save Proj
Verify APROM
Items to be Updated
APROM
DataFlash
LDROM
CONFIG.....
CONFIG Setting
Download Programmer
Read CONFIG
Programmer Information
Exit
APROM Buffer
DataFlash Buffer
File Name: PROJECTS_8051\Z001_GPIO_Blink_LED\Output\Z001_GPIO_Blink_LED.hex
Code Size: 373 Bytes
Checksum: 0xDDB3
Nếu dò được thì quá trình xóa và ghi bắt đầu
Prog. Type
Erase APROM...
S/N: 00000000,12345678
Chip Counter: 000,000


PASS
PASS
Chip is successfully updated !
Quá trình đổ code thành công
OK
2014-03-01_13:09:39

- Kết quả mà ta sẽ thấy được khi nạp thành công: các LED nhấp nháy :



g. Một số lưu ý về thiết lập phần cứng MCU cho W78E055A/N78E059A:

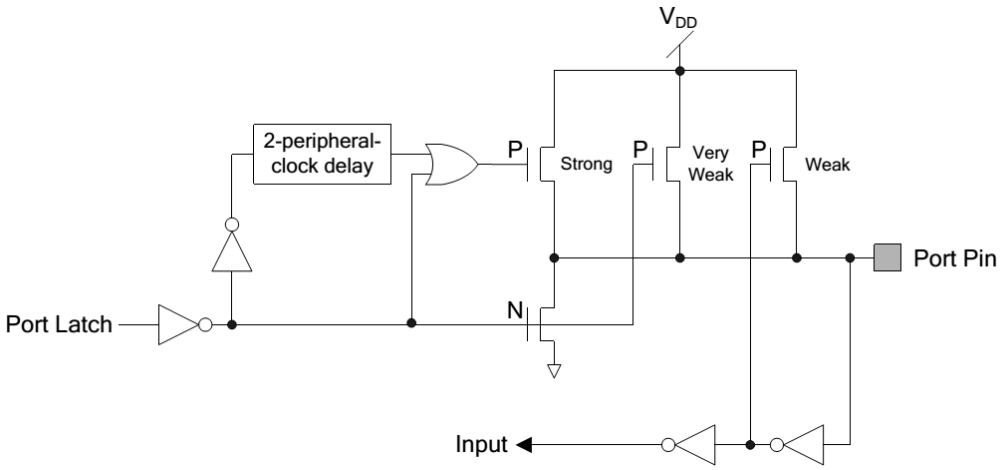
- Trong N78E055A/N78E059A có 3 bytes thiết lập phần cứng, gọi là CONFIG bytes (như xung hệ thống, chế độ 12T/6T, bảo mật code,..). Những thiết lập có thể thay đổi được thông qua Programmer/Writer hoặc chế độ ISP. Như vậy, ta không thể tác động tới các thiết lập này trong chương trình (APROM) như một số dòng chip khác. Chính vậy, cần chú ý tới tốc độ hệ thống khi giao tiếp UART, Timer, hàm Delay,… và khi đổ code cho MCU bằng phần mềm ‘***’Nuvoton ISP-ICP Utility’’*** cùng với việc thiết lập cho chính xác.
- Mặc định ngay từ khi xuất xưởng, MCU được thiết lập :
    - Chạy 12T (1 lệnh thực thi cần 12 xung dao động).
    - ***Chạy với bộ giao động ngoài.*** Chính vậy, trong hướng dẫn này sẽ đổ code lần đầu tiên với thạch anh ngoài rồi thiết lập chạy với IRC để từ lần sau không cần thạch anh nữa.
    - Vô hiệu hóa Watch-dog Timer
    - Hạn chế lệnh MOVC (lệnh này có thể giúp chương trình bên ngoài MCU đọc dữ liệu trong APROM, LDROM của MCU thông qua giao tiếp mở rộng bộ nhớ của kiến trúc 8051 – hiểu nôm na là : vô hiệu hóa MOVC thì dữ liệu trong APROM và LDROM không bị “đọc trộm” -> bảo mật dữ liệu).
    - Mở chế độ lọc xung dao động (Clock filter). Tăng khả năng chống nhiễu, chống ồn cho MCU. Có thể do cảm ứng điện từ hoặc từ board mạch dễ hấp thu dao động ngoài (nhiễu) làm sai lệnh xung hệ thống -> MCU hoạt động không ổn định, chập chờn, chạy lỗi,…Và bộ lọc này sẽ lọc những dao động ngoài mong muốn, thường là các xung gai tần số cao trên 24MHz. Chính vậy, ***khi sử dụng thạch anh hoặc bộ dao động ngoài từ 24MHz trở lên thì phải vô hiệu hóa bộ lọc này đi.***
- Cụ thể hơn, ta có thể xem chi tiết trong phần *24. CONFIG BYTES* của datasheet N78E059A/N78E055A.

IV. NGOẠI VI CỦA N78E055A

1. GPIO

a. Giới thiệu

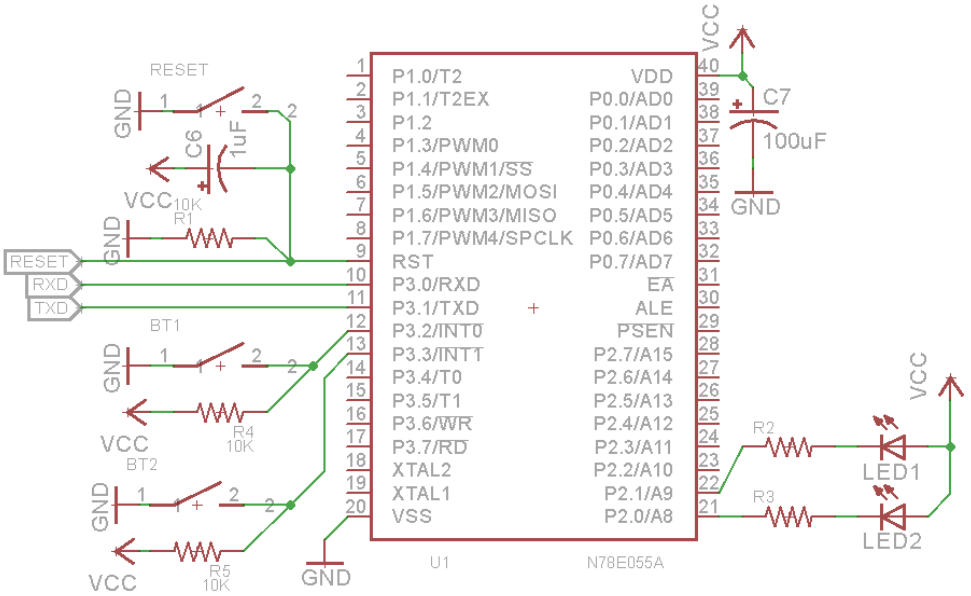
- N78E059A/N78E055A có tối đa 5 port-8 bit (mỗi port có 8 chân ra), tùy vào kiểu đóng gói mà đưa chân ra có đầy đủ các port hay không. Được đánh dấu P0 ~ P4, có thể thiết lập cho từng chân đóng vai trò là ngõ ra hoặc ngõ vào hoặc dạng 2 chiều. Với kiểu ngõ ra: với mức logic high/ low, high cho dòng ra (suorce), low cho dòng vào (sink). Và cũng như hầu hết MCU, N78E055A cho dòng sink lớn hơn là dòng source. Vì vậy, trong các ứng dụng điều khiển, ta vẫn hay cho MCU đóng vai trò kích xuống mass, dùng điện trở ngoài kéo lên (pull-up).



Hình: cấu trúc I/O của N78E055A

b. Ví dụ 1 – ON/OFF LED (*Z002_GPIO_ON-OFF_LED*)

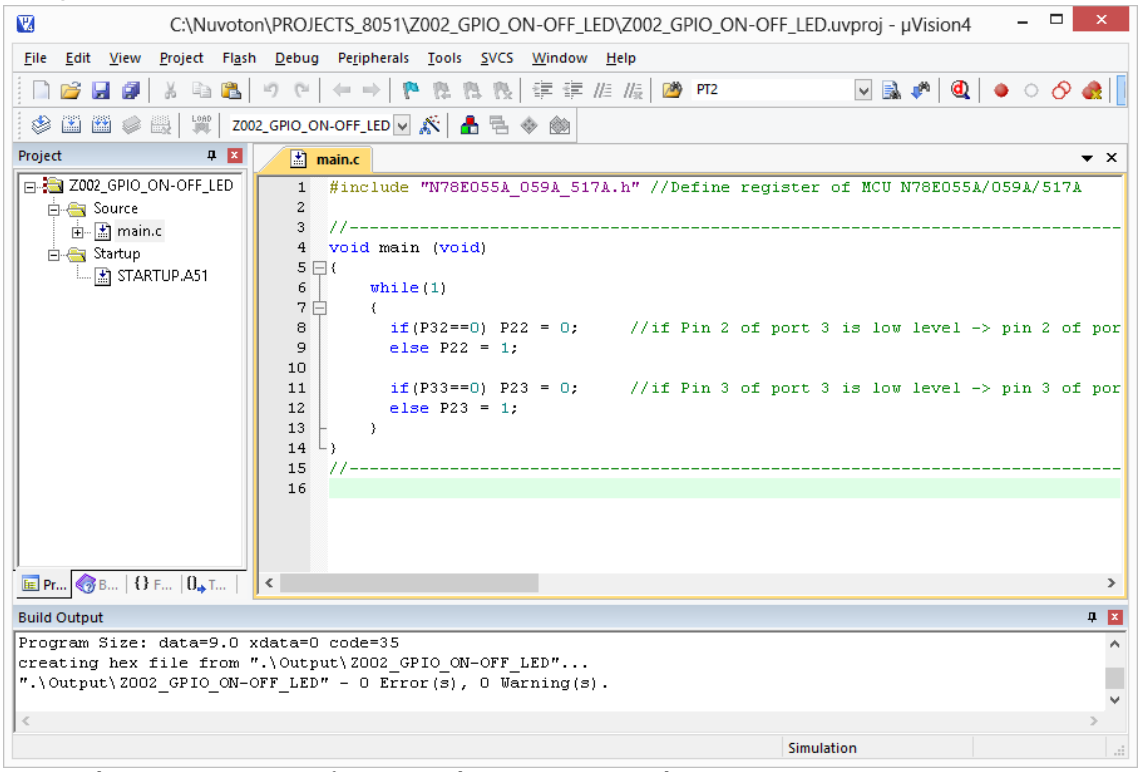
- **Ý tưởng:** xây dựng mạch điện dùng N78E055A, có 2 nút nhấn (BT1, BT2) và 2 LED (D1, D2). Nhấn giữ BT1 thì D1 sáng, nhấn giữ BT1 thì D2 sáng. Đấu nối mạch nguyên lý. Nhả bút nhấn thì LED tắt.
- **Xây dựng mạch:**



- **Code:** Ta tạo project *Z002_GPIO_ON-OFF_LED* tương tự như project đã được hướng dẫn phần trên. Ở ví dụ này, chỉ dùng file *main.c, STARTUP.A51*

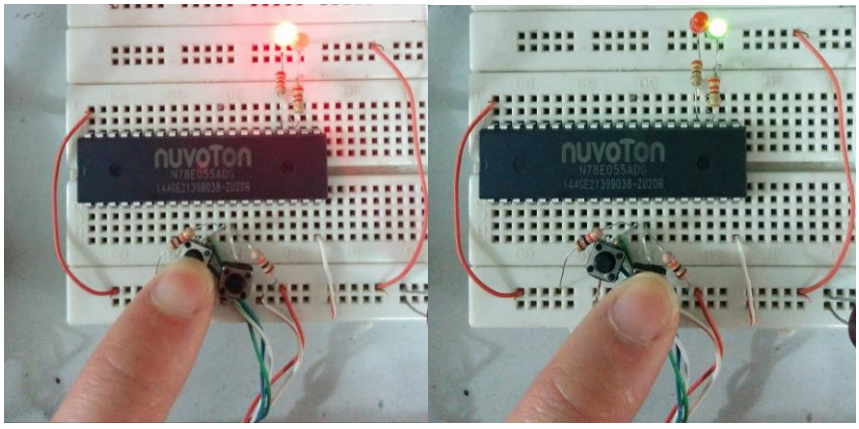
Thêm các dòng code dưới đây vào file ***main.c*** :

```
#include "N78E055A_059A_517A.h"          //Define register of MCU N78E055A/059A/517A
void main (void)
{
   while(1)
   {
            if(P32==0) P22 = 0;     //if Pin 2 of port 3 is low level -> pin 2 of port 2 is low level (LED ON), else it high level (LED off).
            else P22 = 1;

            if(P33==0) P23 = 0;     //if Pin 3 of port 3 is low level -> pin 3 of port 2 is low level (LED ON), else it high level (LED off).
            else P23 = 1;
   }
}
```



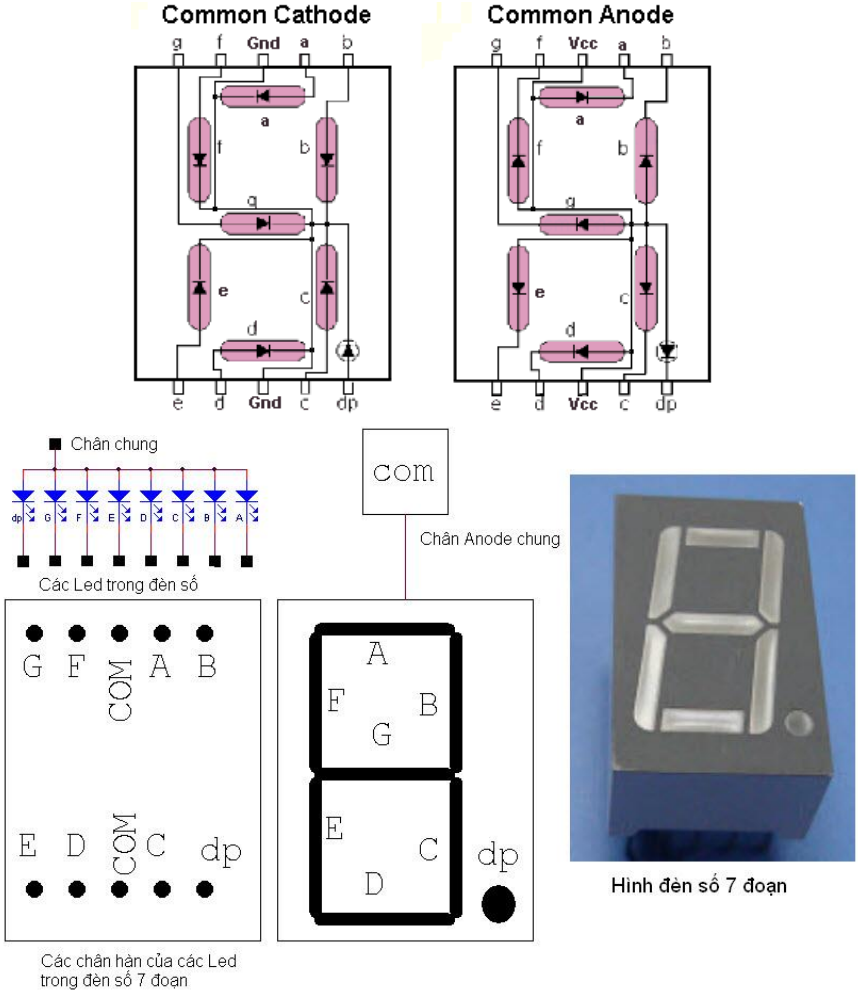
Tiến hành build và đổ code xuống MCU và nhấn các nút.

- **Kết quả:**

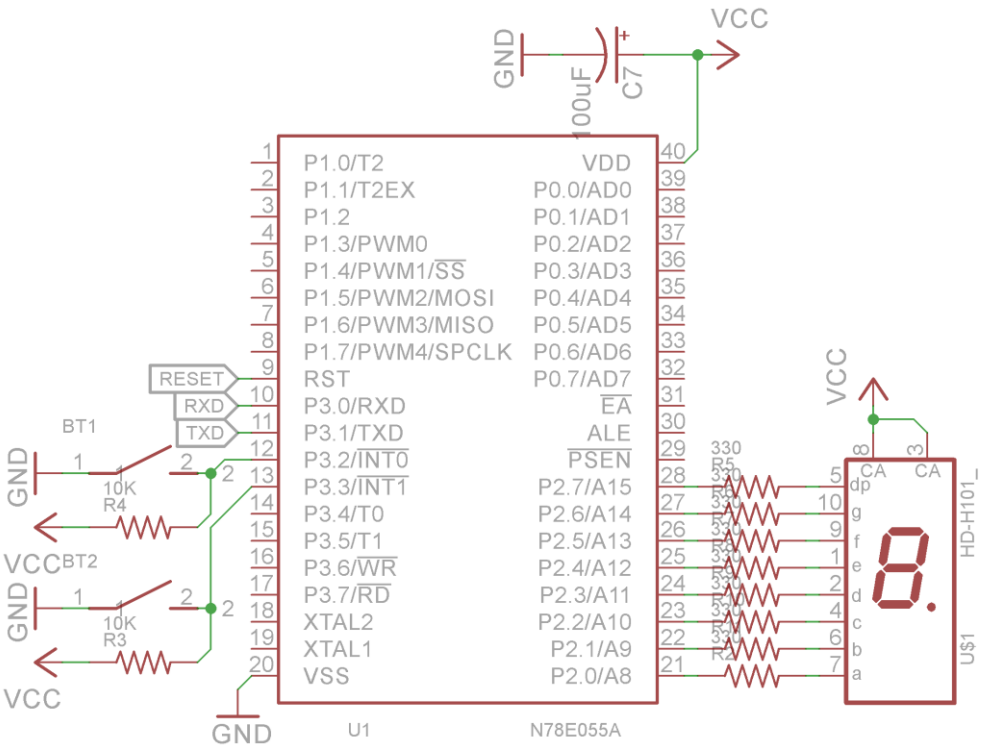


c. **Ví dụ 2 – LED 7 thanh (*Z003_GPIO_7SEG)***

- **Ý tưởng:** Xây dựng mạch điện dùng N78E055A, có 2 nút nhấn BT1 và BT2, 1 LED 7 đoạn hiện thị giá trị đếm theo hệ số 16 (từ 0 -> F). Nhấn BT1 thì tăng số đếm, cao nhất là F và dừng (không tăng nữa nếu nhấn tiếp). Nhấn BT2 thì giảm số đếm, thấp nhất là 0 và dừng (không giảm nữa nếu nhấn tiếp).

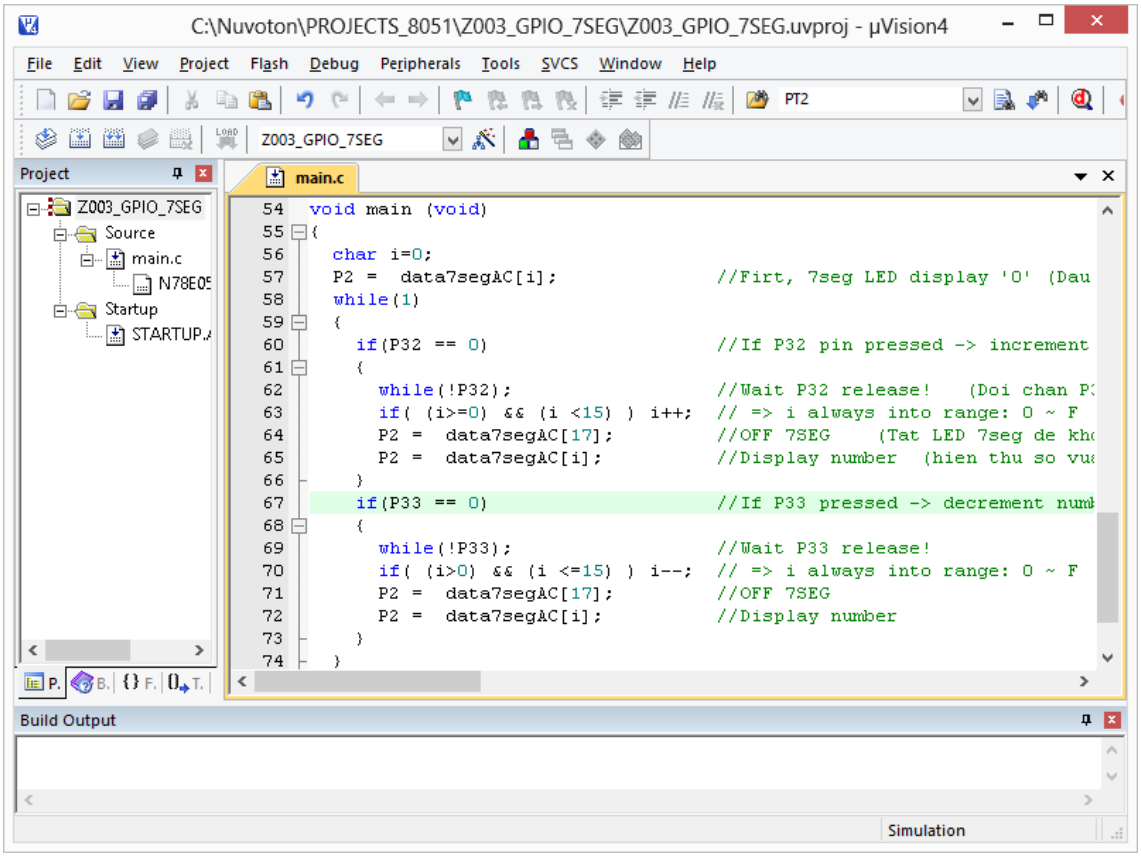


- **Xây dựng mạch:** BT1 kết nối pin P32, BT2 kết nối pin P33, LED 7 đoạn kết nối port 2 qua điện trở 330Ω hoặc 220Ω đều được. Dùng LED 7 đoạn chung Anode (cực dương).

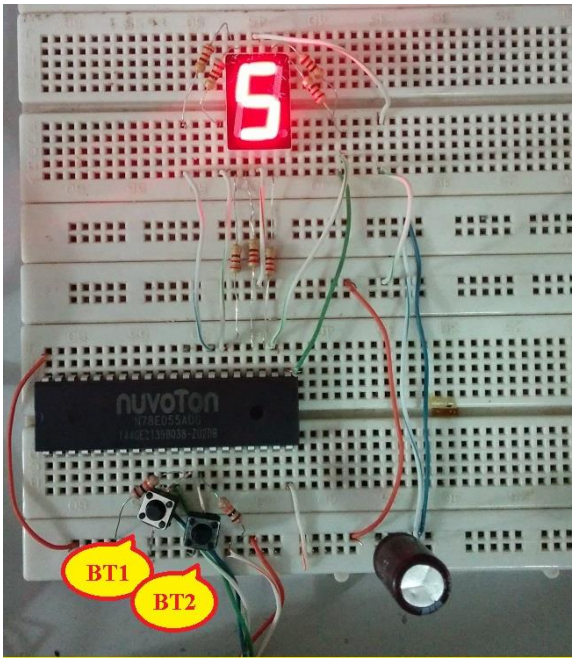


- **Code:** Thêm các dòng code sau vào file ***main.c***

```
#include "N78E055A_059A_517A.h"
//code display for 7SEG Cathode Common
unsigned char data7segCC[17] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F, 0x77, 0x7C, 0x39, 0x5E, 0x79, 0x71, 0x00      //OFF 7SEG};
//code display for 7SEG Anode Common
unsigned char data7segAC[17] = {0xC0,0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80,
0x90, 0x88, 0x83, 0xC6, 0xA1, 0x86, 0x8E, 0xFF //OFF 7SEG};
//-------------------------------------------------------------
void main (void)
{
    char i=0;
    P2 =  data7segAC[i];    //Firt, 7seg LED display '0'
 {
    if(P32 == 0)            //If P32 pin pressed -> increment number
    {
            while(!P32);    //Wait P32 release!
            if( (i>=0) && (i <15) ) i++;    // => i always into range: 0 ~ F
            P2 =  data7segAC[17];           //OFF 7SEG
            P2 =  data7segAC[i];            //Display number                    }
    if(P33 == 0)            //If P33 pressed -> decrement number
     {
            while(!P33);                    //Wait P33 release!
            if( (i>0) && (i <=15) ) i--;    // => i always into range: 0 ~ F
            P2 =  data7segAC[17];           //OFF 7SEG
            P2 =  data7segAC[i];            //Display number
    }
 }
}
//-------------------------------------------------------------
```

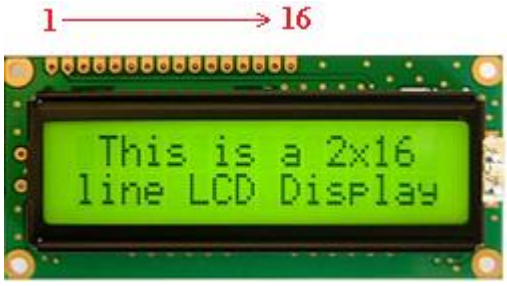

C:\Nuvoton\PROJECTS_8051\Z003_GPIO_7SEG\Z003_GPIO_7SEG.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
PT2
Z003_GPIO_7SEG
Project
Z003_GPIO_7SEG
Source
main.c
Startup
main.c
54 void main (void)
55 {
56 char i=0;
57 P2 = data7segAC[i]; //Firt, 7seg LED display '0' (Dau
58 while(1)
59 {
60 if(P32 == 0) //If P32 pin pressed -> increment
61 {
62 while(!P32); //Wait P32 release!
63 if((i>=0) && (i <15)) i++; // => i always into range: 0 ~ F
64 P2 = data7segAC[17]; //OFF 7SEG
65 P2 = data7segAC[i]; //Display number
66 }
67 if(P33 == 0) //If P33 pressed -> decrement
68 {
69 while(!P33); //Wait P33 release!
70 if((i>0) && (i <=15)) i--; // => i always into range: 0 ~ F
71 P2 = data7segAC[17]; //OFF 7SEG
72 P2 = data7segAC[i]; //Display number
73 }
74 }
Build Output
Simulation

- **Kết quả:**


BT1
BT2

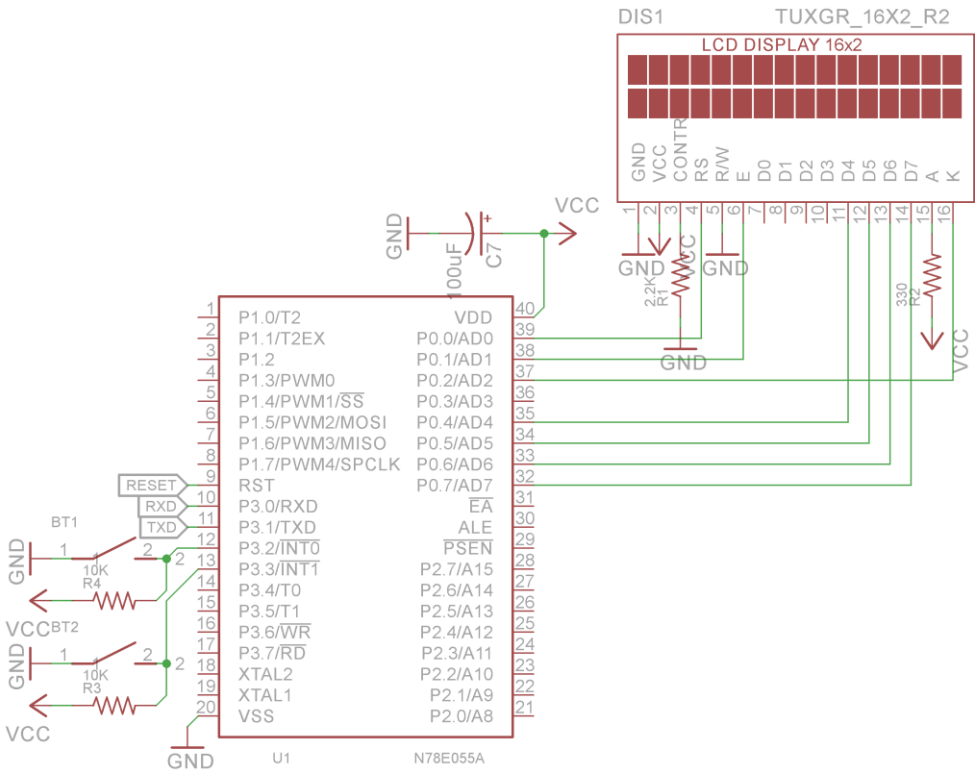
d. Ví dụ 3 – LCD Text 16x2 (*Z006 + Z007*)

- **Ý tưởng:** xây dựng mạch điện dùng N78E055A, có LCD Text (16x2 chẳng hạn), có nút nhấn BT1 và BT2. Nhấn BT1 thì tăng số đếm, nhấn BT2 thì giảm số đếm. Số đếm hiển thị lên LCD Text.

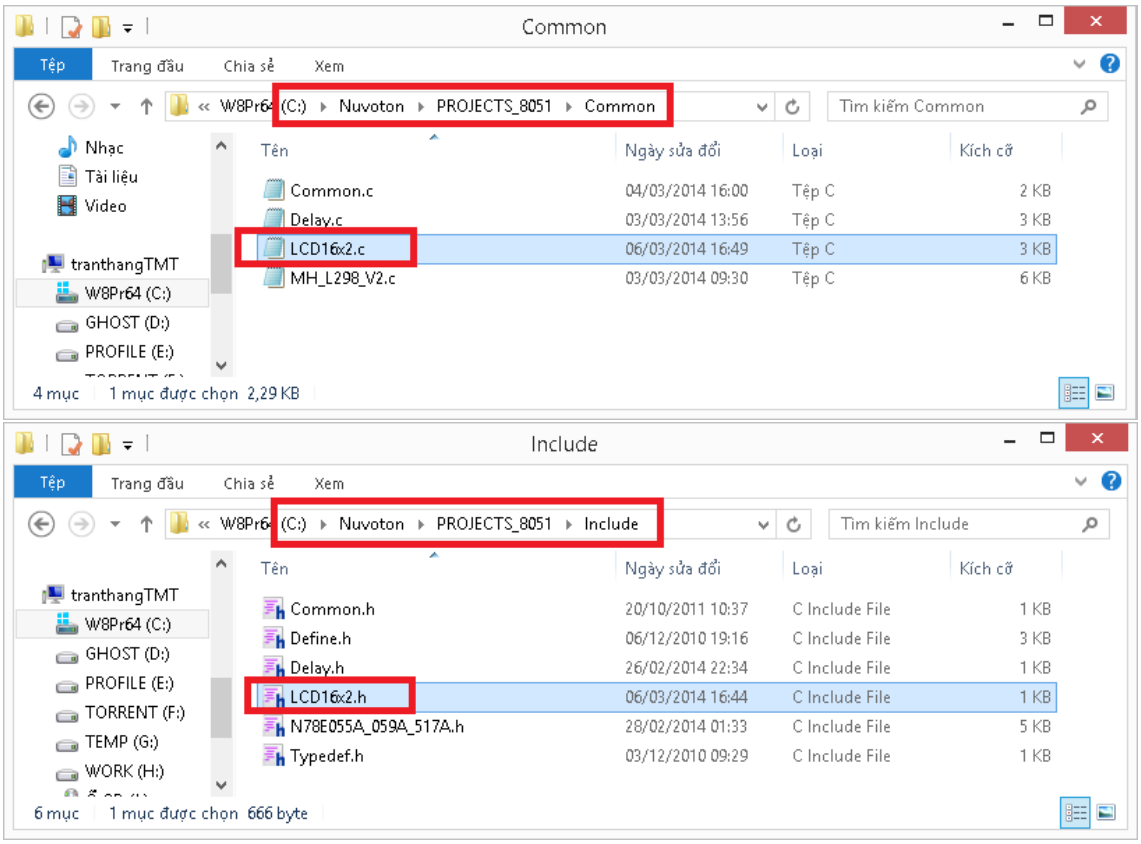


Chân	Ký hiệu	Trạng thái logic	Mô tả
1	VSS	-	0V
2	VDD	-	+5V
3	VE	-	0 - VDD
4	RS	0 1	D0~D7: lệnh D0~D7: dữ liệu
5	RW	0 1	Ghi (MCU-> LCD) Đọc (LCD-> MCU)
6	E	0 1 1 xuống 0	Vô hiệu hóa đọc/ ghi LCD LCD hoạt động Bắt đầu đọc/ghi LCD
7	D0	0/1	Bit 0 LSB
8	D1	0/1	Bit 1
9	D2	0/1	Bit 2
10	D3	0/1	Bit 3
11	D4	0/1	Bit 4
12	D5	0/1	Bit 5
13	D6	0/1	Bit 6
14	D7	0/1	Bit 7 MSB
15	A	-	+2V
16	K	-	0V

- **Xây dựng mạch:** LCD Text (16x2) kết nối N78E055A qua port 0, chế độ truyền dữ liệu 4-bit. Với LCD, ta kết nối như hình dưới, có thêm chân điều khiển đèn nền LCD. BT1 kế nối với P32, BT2 kết nối với P33.



- **Code:** Ta sẽ tạo thêm thư viện LCD Text này để dùng cho cả các ví dụ về sau. Tạo file *LCD16x2.h* cho vào folder *Include*, file *LCD16x2.c* cho vào folder *Common.*



- Chép các dòng lệnh này vao file ***LCD16x2.h***

//Khai bao cac chac hoc LCD. Neu thay doi thi khai bao lai tai day.
#define LCD_PIN_RS P00
#define LCD_PIN_E P01

```
#define LCD_PIN_LIGHT  P02
#define LCD_PIN_D4     P04
#define LCD_PIN_D5     P05
#define LCD_PIN_D6     P06
#define LCD_PIN_D7     P07

//Khai bao cac ham cho LCD
void LCD_Export_4bit(char _4bit);
void LCD_Cmd(char _cmd);
void LCD_Data(char _data);
void LCD_Init(void);
void LCD_Local(unsigned char line, unsigned char row);
void LCD_String(char* _char, unsigned char line, unsigned char row);

//Dinh nghia Macro
#define LCD_Trigger() {LCD_PIN_E=1; Delay10us(1); LCD_PIN_E=0; Delay10us(4);}
#define LCD_Clear_Display() {LCD_Cmd(0x01); Delay1ms(16);}
```

- Chép các dòng lệnh này vao file ***LCD16x2.c***

```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Delay.h"
#include "LCD16x2.h"
//-----------------------------------
void LCD_Export_4bit(char _4bit){
    LCD_PIN_D4 = _4bit & 0x01;_4bit >>=1;
    LCD_PIN_D5 = _4bit & 0x01;_4bit >>=1;
    LCD_PIN_D6 = _4bit & 0x01;_4bit >>=1;
    LCD_PIN_D7 = _4bit & 0x01;
}
void LCD_Cmd(char _cmd){
    LCD_PIN_RS=0;                 //Chon che do ghi lenh
    LCD_PIN_E=0;                  //Cho chan E muc thap
    LCD_Export_4bit(_cmd >>4);    //Xuat 4 bit cao ra D4~D7
    LCD_Trigger();                //Day du lieu vao LCD
    LCD_Export_4bit(_cmd);        //Xuat 4 bit thap ra D4~D7
    LCD_Trigger();                //day du lieu vao LCD
}
void LCD_Data(char _data){
    LCD_PIN_RS=1;                 //Chon che do ghi data
    LCD_PIN_E=0;                  //Cho chan E muc thap
    LCD_Export_4bit(_data >>4);   //Xuat 4 bit cao ra D4~D7
    LCD_Trigger();                //Day du lieu vao LCD
    LCD_Export_4bit(_data);       //Xuat 4 bit thap ra D4~D7
    LCD_Trigger();                //day du lieu vao LCD
}
void LCD_Init(void){
    P0OR = 0x01;                  //Enable internal pull-up resistors
    Delay1ms(20);                 //Delay 15ms for supply of LCD stabilize
    LCD_PIN_RS=0;
    LCD_PIN_E = 0;
    LCD_Export_4bit(0x03);
    LCD_Trigger();
    Delay1ms(5);
```

```
    LCD_Export_4bit(0x03);
    LCD_Trigger();
    Delay10us(10);
    LCD_Export_4bit(0x03);
    LCD_Trigger();
    Delay10us(5);
    LCD_Export_4bit(0x02);
    LCD_Trigger();
    Delay10us(1);

    LCD_Cmd(0x28);              //Data 4bit, 2 lines, font 5x7
    LCD_Cmd(0x0f);              //Display ON, Cursor ON and Blink
    LCD_Cmd(0x06);              //address increment, cursor shift
    LCD_Clear_Display();        //Clear Display
}
void LCD_Local(unsigned char line, unsigned char row){
     if(line==1)                //if chose line 1
            LCD_Cmd(0x80+row);
     else
            LCD_Cmd(0xC0+row);
    Delay10us(5);
}
void LCD_String(char* _char, unsigned char line, unsigned char row){
    char i=0;
    LCD_Local(line,row);
    while((*_char) != '\0')
    {
            LCD_Data(*_char);
            _char ++;
    }
}
```

- Thêm các dòng lệnh này vào file *Delay.h* ở folder *Include:*

```
void Delay10us(UINT16 u16CNT){
   TMOD |= 0x01;                //Timer0 is 16-bit mode
   TR0 = 1;                     //Trigger Timer0
   while (u16CNT != 0)
   {
      TL0 = LOBYTE(VALUE_10us);
      TH0 = HIBYTE(VALUE_10us);
      while (TF0 != 1);         //Check Timer0 Time-Out Flag
      TF0 = 0;
      u16CNT --;
   }
   TR0 = 0;                     //Stop the Timer0
}
```

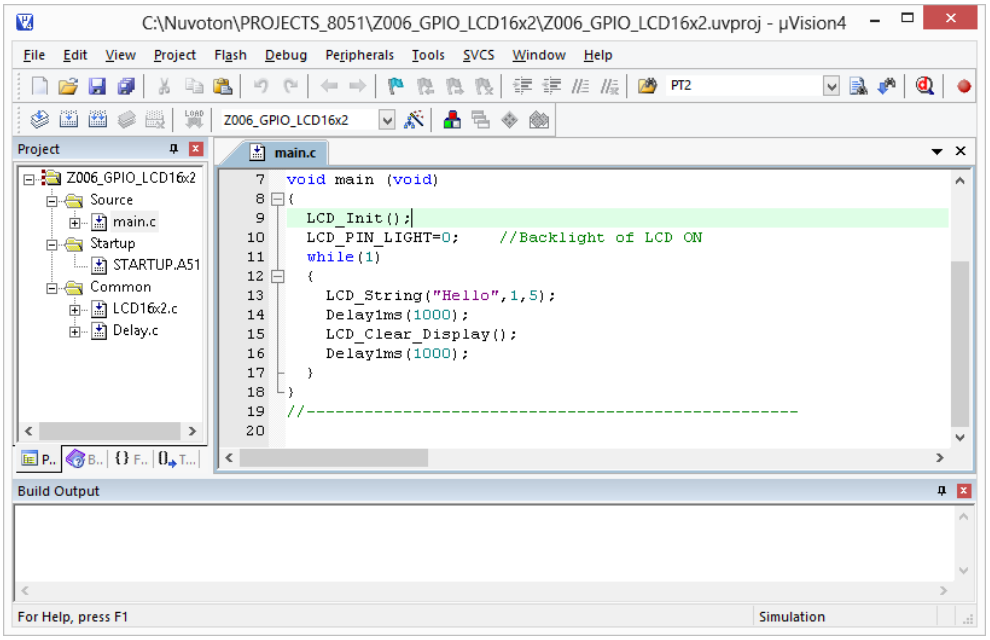
- Chép các dòng lệnh này vao file ***main.c*** cho ví dụ *Z006_GPIO_LCD16x2:*

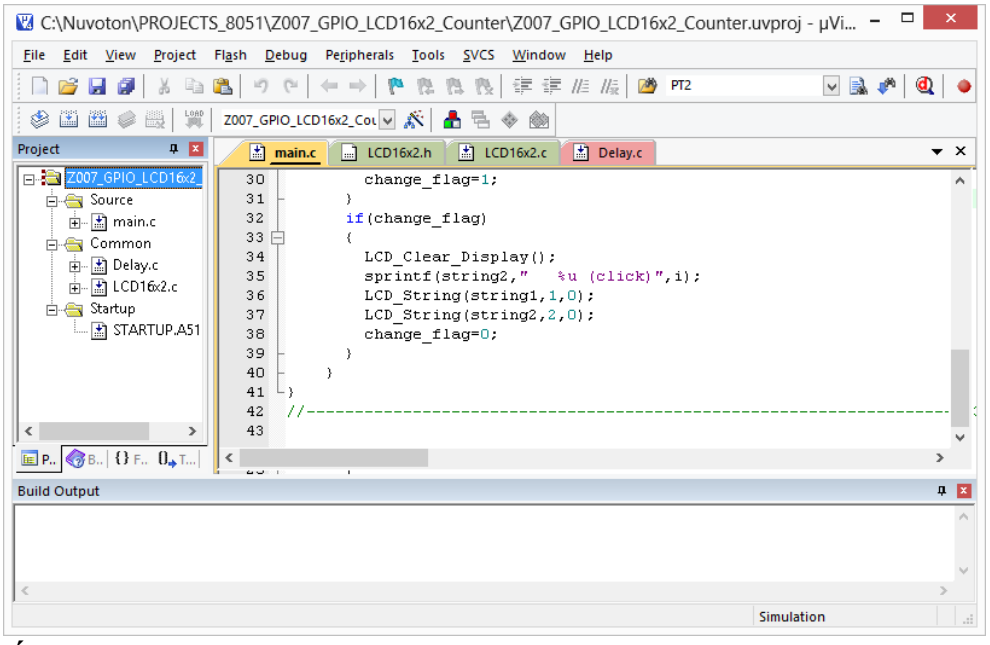
```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Delay.h"
#include "LCD16x2.h"
//-------------------------------------------------

void main (void)
```

```
{
    LCD_Init();
    LCD_PIN_LIGHT=0;              //Backlight of LCD ON
    while(1)
    {
            LCD_String("Hello",1,5);
            Delay1ms(1000);
            LCD_Clear_Display();
            Delay1ms(1000);
    }
}
//---------------------------------------------------
```

- Chép các dòng lệnh này vao file ***main.c*** cho ví dụ *Z007_GPIO_LCD16x2_Counter:*

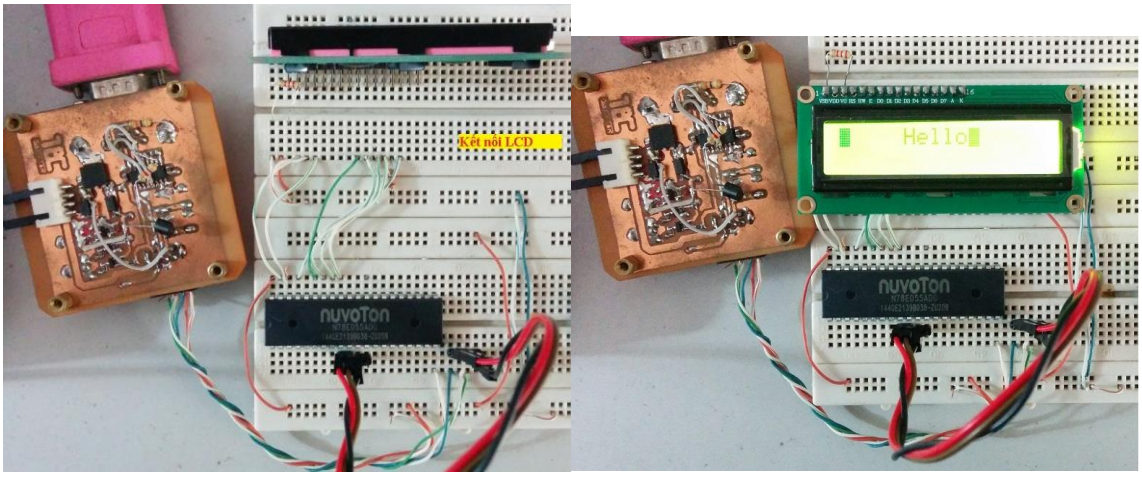
```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Delay.h"
#include "LCD16x2.h"
//---------------------------------------------------

void main (void)
{
    LCD_Init();
    LCD_PIN_LIGHT=0;   //Backlight of LCD ON
     while(1)
    {
            LCD_String("Hello",1,5);
            Delay1ms(1000);
            LCD_Clear_Display();
            Delay1ms(1000);
    }
}
//---------------------------------------------------
```



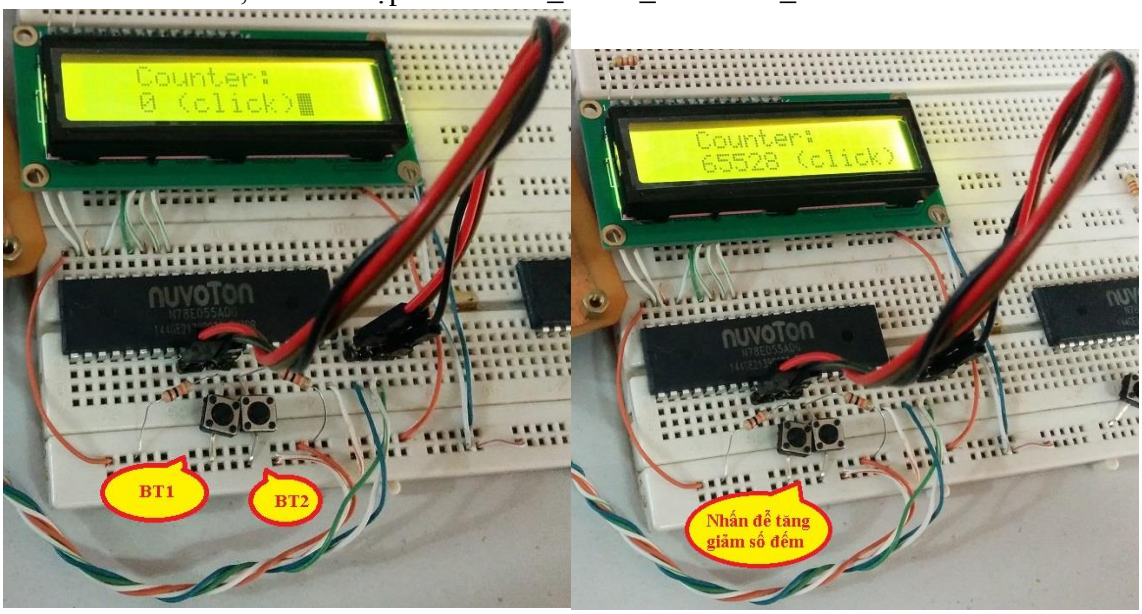


- **Kết quả:**

  Test LCD với *Z006_GPIO_LCD16x2*



  - Thêm BT1, BT2 và nạp code *Z007_GPIO_LCD16x2_Counter*



2. Ngắt (Interrupt)

a. Giới thiệu

- Trong N78E055A có 11 nguồn ngắt và có 4 mức ưu tiên. 11 nguồn ngắt từ các ngoại vi: Timer1/2/3/, ngắt ngoài INT0/1/2/3, ngắt UART, SPI, Brown-out, Waking-up Timer (thức dậy từ chế độ Power Down). Mỗi nguồn ngắt đều có các thanh ghi thiết lập cho phép ngắt, mức ưu tiên, cờ báo ngắt, địa chỉ vector ngắt. N78E055A cũng như các MCU 8051 có 4 mức ưu tiên ngắt (level 0 -> level 3), giá trị cao hơn thì ưu tiên cao hơn Việc thiết lập Ta có thể xem địa chỉ vec tor ngắt tại chương *17. INTERRUPT SYSTEM* ở datasheet. Cũng như có mô tả tại các chương về ngoại vi của N78E055A.

Interrupt Priority Control Bits		Interrupt Priority Level
IPH / EIPH	IP / EIP / XICON[7,3]	
0	0	Level 0 (lowest)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest)

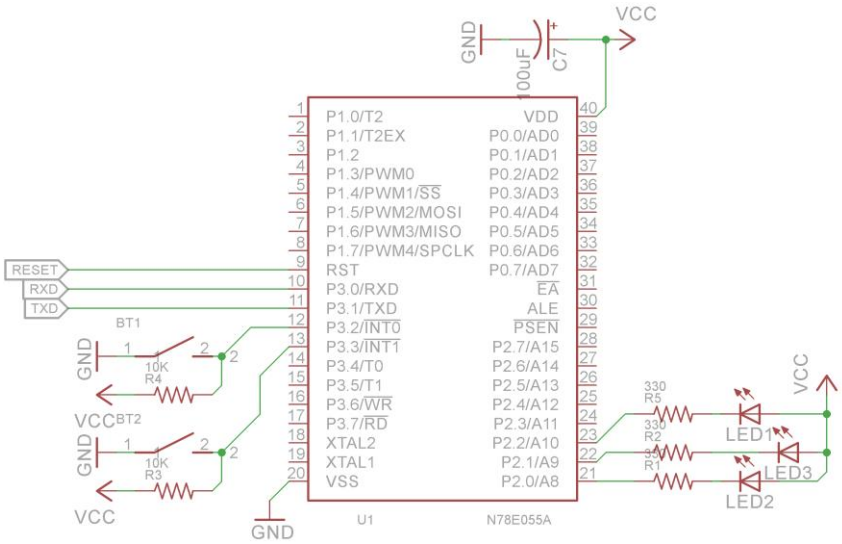
- Địa chỉ vector ngắt của mỗi nguồn ngắt ta xem trong datasheet để MCU truy cập đúng vị trí. Khi ngắt xảy ra, MCU sẽ nhảy tới địa chỉ này để thực thi chương trình trong ngắt.

Source	Vector Address	Vector Number	Source	Vector Address	Vector Number
External Interrupt 0	0003H	0	Timer 0 Overflow	000BH	1
External Interrupt 1	0013H	2	Timer 1 Overflow	001BH	3
Serial Port Interrupt	0023H	4	Timer 2 Overflow/Capture/Reload	002BH	5
External Interrupt 2	0033H	6	External Interrupt 3	003BH	7
SPI Interrupt	0043H	8	Power Down Waking-up Timer Interrupt	004BH	9
Brown-out Detection Interrupt	0053H	10			

- Các thanh ghi liên quan tới việc ngắt:
    - IE, EIE và XICON- Cho phép nguồn nào được ngắt, EIE là thanh ghi mở rộng của IE. XICON thiết lập cho INT2 và INT3.
    - IP và IPH – Thiết lập mức ưu tiên cho nguồn ngắt. Mỗi nguồn ngắt có 2 bit thiết lập mức ưu tiên (giá trị: 00 ~ 11). IP chứa bit thấp, IPH chứa bit cao của 2 bit thiếp lập này.
    - EIP và EIPH – mở rộng cho IP và IPH.
    - TCON – đây là thanh ghi điều khiển Timer0/1 nhưng có 4 bit thiết lập kiểu ngắt, flag của INT1, INT0.
    - Ngoài ra, mỗi ngoại vi có những thanh ghi riêng cho mình. Vì dụ: Timer có thiết lập kiểu ngắt mức/ cạnh,…
- Tùy vào thiết lập và nguồn ngắt mà các cờ ngắt tự xóa bằng phần cứng hoặc ta phải xóa trong chương trình.

b. Ví dụ - Điều khiển LED với ngắt ngoài INT0 và INT1 (*Z004_IN_Blink_LED*)

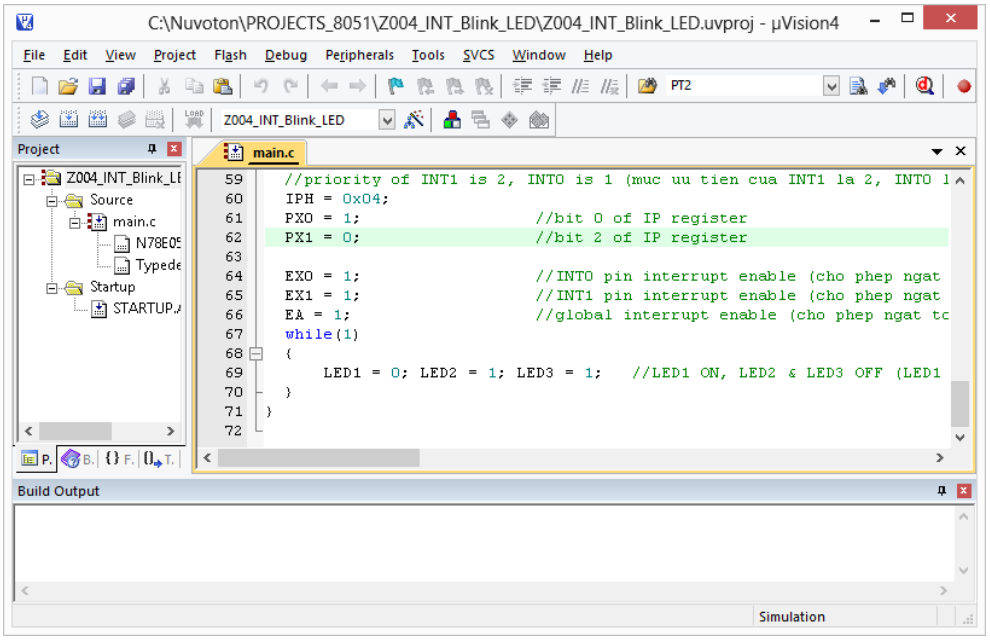
- ***Ý tưởng:*** Ta sẽ thử ngắt ngoài INT0 và INT1 của N78E055A. Thiết lập ngắt INT0 là dạng ngắt mức thấp, ngắt INT1 dạng ngắt sườn (cạnh) xuống. Ngắt INT1 ưu tiên cao hơn INT0. Chương trình ngắt INT0: sáng LED2 và tắt các LED khác. Chương trình ngắt INT1: sáng nhấp nháy 5 lần LED3 và tắt các LED khác. Chương trình main: sáng LED1 và tắt các LED khác.
- ***Xây dựng mạch:*** Dùng nút nhấn BT1 nối chân P32 (INT0), BT2 nối P33 (INT1). LED1 nối P22, LED2 nối P21, LED3 nối P20.



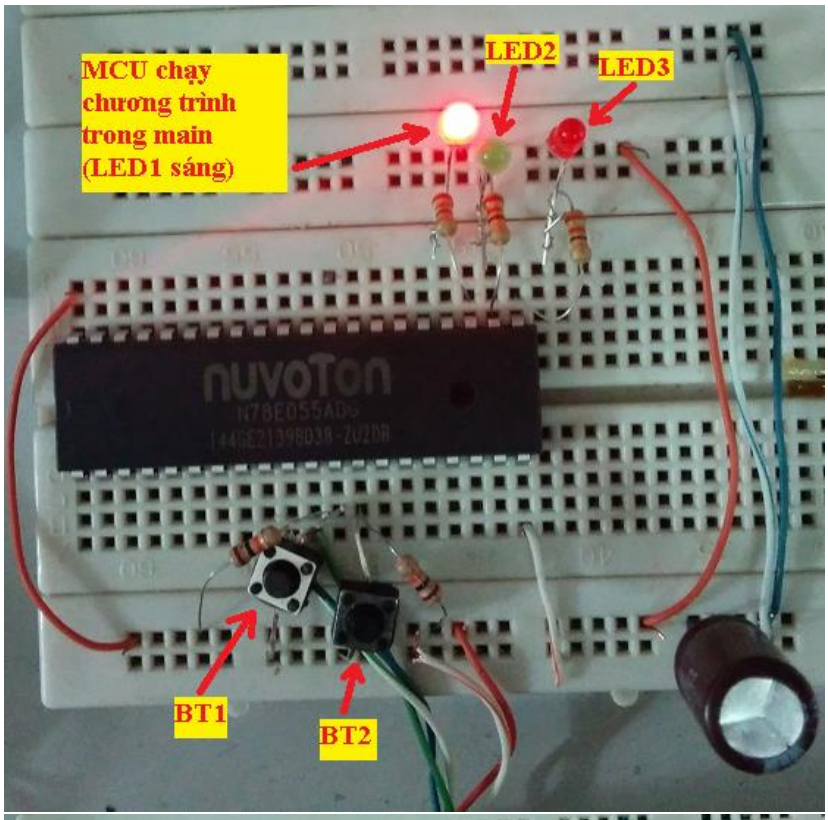
- ***Code:***

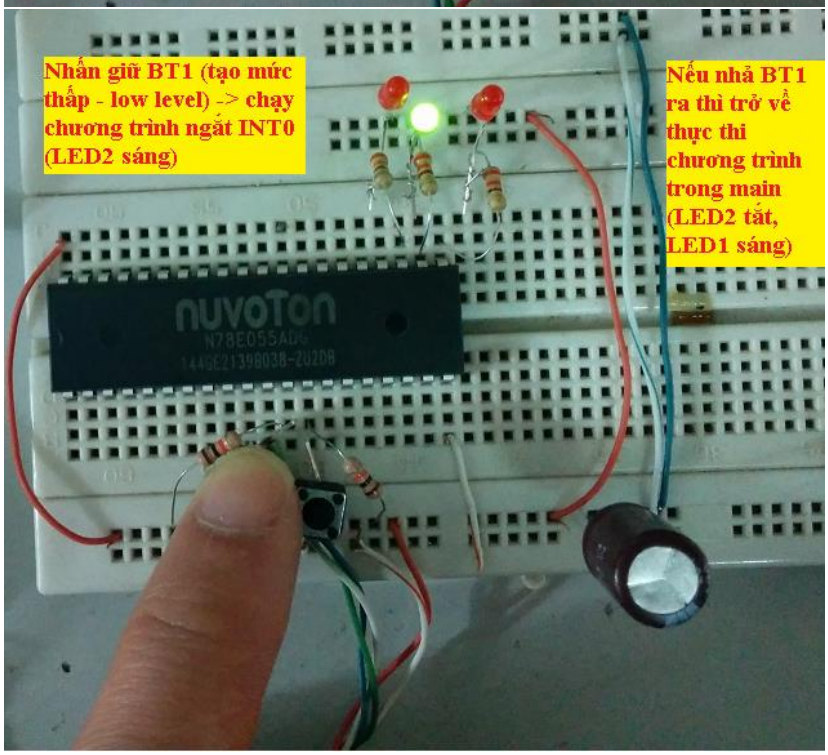
```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"

//Khai bao cac chan LED
#define LED1        P22
#define LED2        P21
#define LED3        P20
/*////////////////////////////////////////////////////////////////////////////
Tao chuong trinh delay don gian. Khong yeu cau cao ve do chinh xac.
*/
void Delay_ms(UINT32 _time)
{
    UINT32 _timeus;
    while(_time !=0)
    {
            _timeus = 70;
    while(_timeus!=0){_timeus --;}
            _time--;
    }
}
////////////////////////////////////////////////////////////////////////////INT0 interrupt subroutine
void INT0_ISR (void) interrupt 0    //interrupt address is 0x0003
{
    EX0 = 0;
    LED1 = 1; LED2 = 0; LED3 = 1;
    Delay_ms(100);
    EX0 = 1;
}
////////////////////////////////////////////////////////////////////////////INT1 interrupt subroutine
void INT1_ISR (void) interrupt 2
{
    char i=5;
    EX1 = 0;
    LED1 = 1; LED2 = 1;
    while(i>0)
```

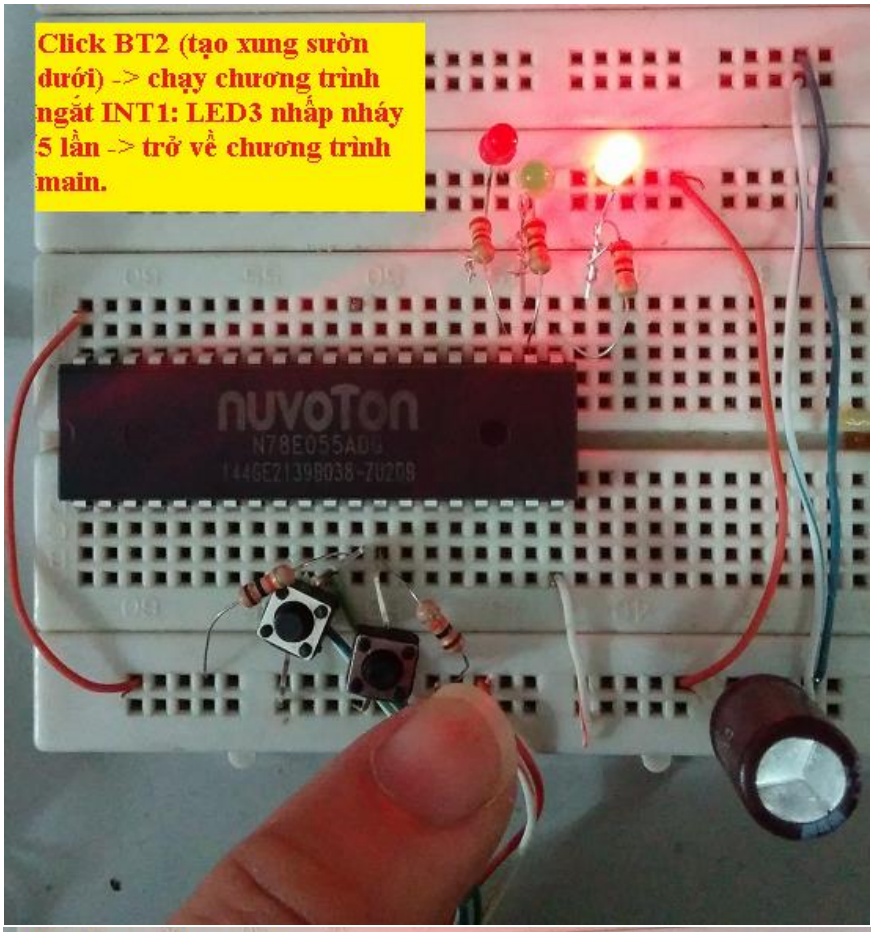
```
    {
            LED3 = 0; Delay_ms(100);
            LED3 = 1; Delay_ms(100);
            i--;
    }
    EX1 = 1;
}
void main (void)
{
     LED1=1; LED2=1; LED3=1;  //Off all LED
    IT0 = 0;                  //Low level triggered
     IT1 = 1;                 //Falling edge triggered
    //priority of INT1 is 2, INT0 is 1:
    IPH = 0x04;
    PX0 = 1;                  //bit 0 of IP register
    PX1 = 0;                  //bit 2 of IP register
    EX0 = 1;                 //INT0 pin interrupt enable (cho phep ngat INT0)
    EX1 = 1;                 //INT1 pin interrupt enable (cho phep ngat INT1)
    EA = 1;                  //global interrupt enable (cho phep ngat toan cuc)
    while(1)
    {
        LED1 = 0; LED2 = 1; LED3 = 1;   //LED1 ON, LED2 & LED3 OFF
    }
}
```

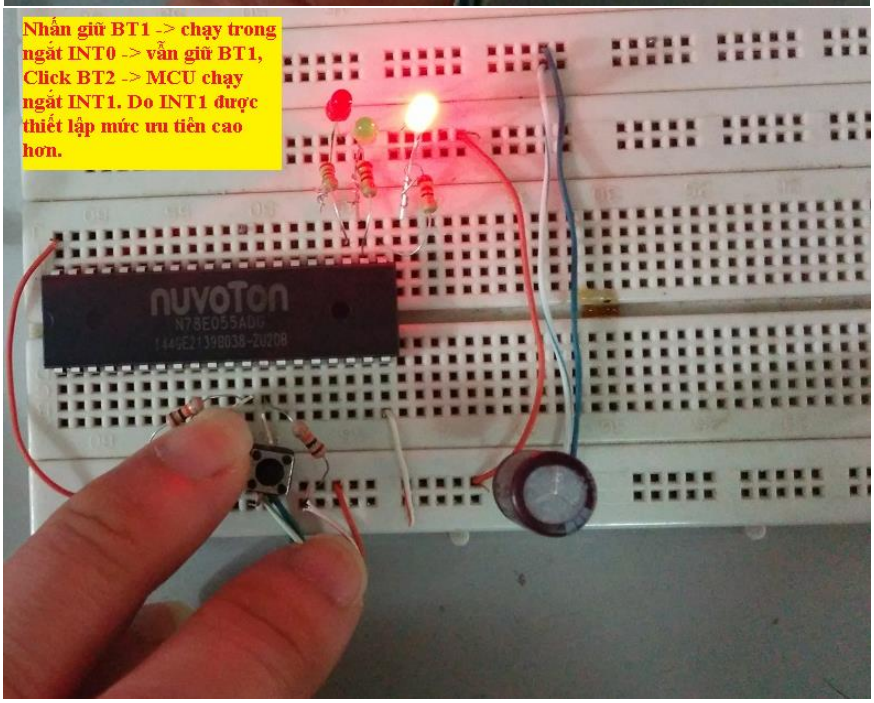


- ***Kết quả:***


MCU chạy chương trình trong main (LED1 sáng)
LED2
LED3
BT1
BT2


Nhấn giữ BT1 (tạo mức thấp - low level) -> chạy chương trình ngắt INT0 (LED2 sáng)
Nếu nhả BT1 ra thì trở về thực thi chương trình trong main (LED2 tắt, LED1 sáng)


Click BT2 (tạo xung sườn dưới) -> chạy chương trình ngắt INT1: LED3 nhấp nháy 5 lần -> trở về chương trình main.
nuvoTon
N78E055AD


Nhấn giữ BT1 -> chạy trong ngắt INT0 -> vẫn giữ BT1, Click BT2 -> MCU chạy ngắt INT1. Do INT1 được thiết lập mức ưu tiên cao hơn.
nuvoTon
N78E055AD

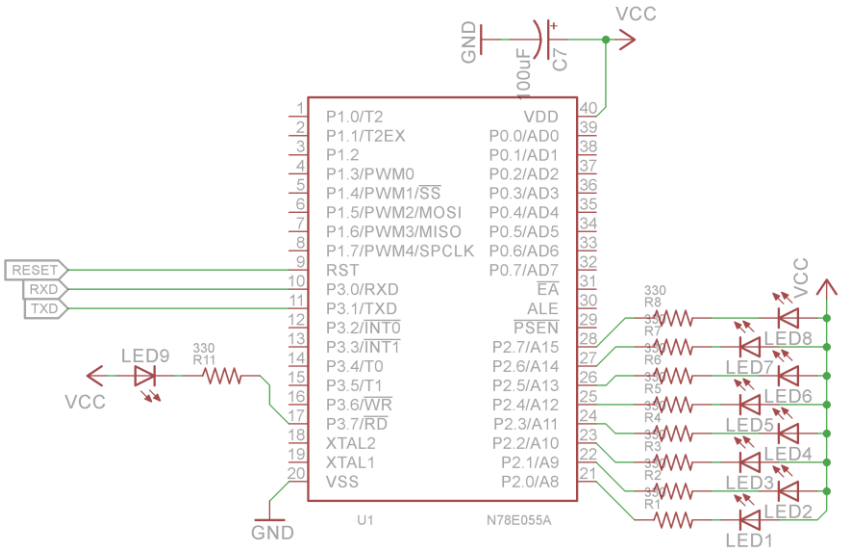
3. Timer/Counter

a. Giới thiệu

- N78E055A có 3 Timer/Counter 16 bit (xin gọi tắt là TC), mỗi timer có 2 thanh ghi 8bit cho việc đếm: trong đó THx là 8bit cao, TLx là 8 bit thấp, x= 0/1/2.
- TCON, TMOD là các thanh ghi thiết lập cho TC 0 và 1.
- T2CON, T2MOD, RCAP2L, RCAP2H là các thanh ghi thiết lập cho TC 2.
- Cụ thể chức năng từng bit trong các thanh ghi của TC, ta xem trong *10. Timer/Counters* của datasheet.
- TC 0 và 1 có 4 chế độ: 13bit timer (Mod 0), 16bit timer (Mod 1), 8 bit timer tự động nạp lại (Mod 2), 2 timer-8bit song hành (Mod3).
- TC2 có các chế độ: Capture mode, Timer 16bit tự nạp lại, chế độ bộ phát baud rate cho UART, xuất xung dao động ra ngoài.

b. Ví dụ 1 – Bink LED with Timer (*Z001_GPIO_Blink_LED)*

- ***Ý tưởng:*** *xây dựng mạch dùng N78E055A, dùng Timer tạo hàm delay tương đối chính xác cho việc nhấp nháy LED.*
- ***Xây dựng mạch:*** *Đây chính là mạch ta dùng để test ở phần hướng dẫn build project với Keil uVision.*

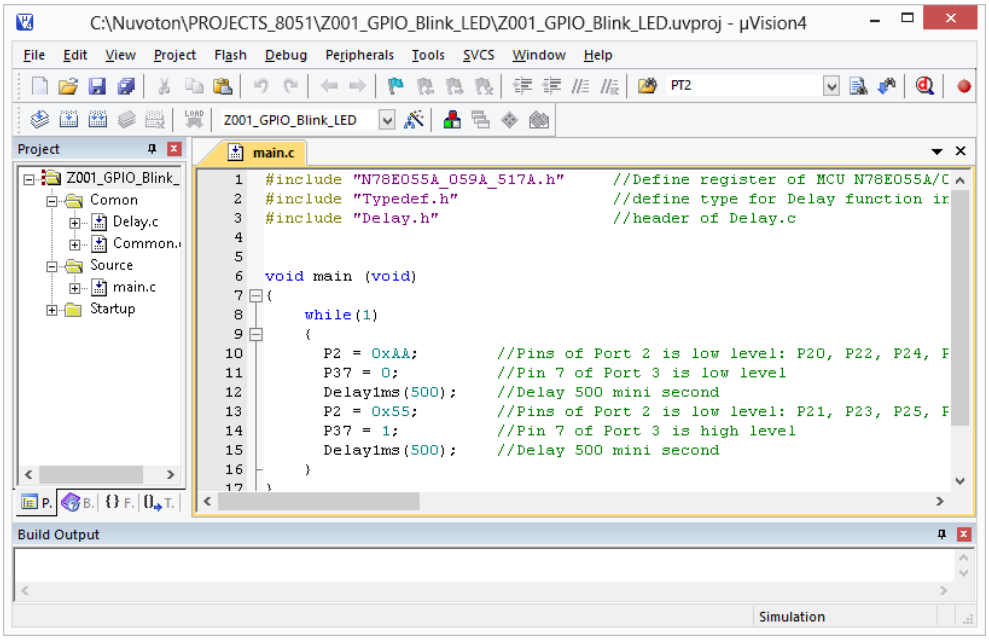


- ***Code:***

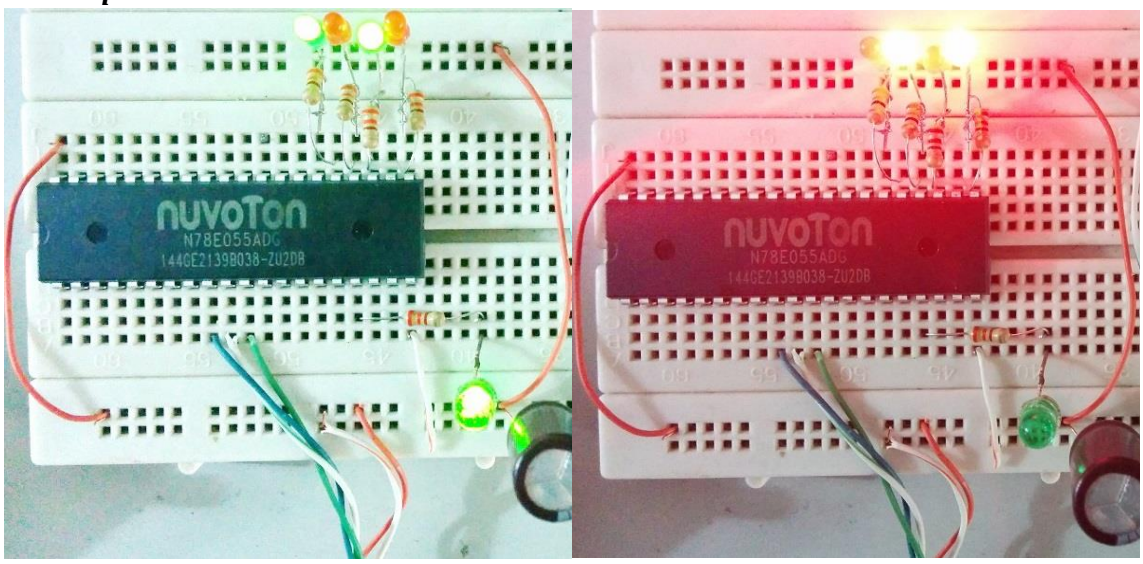
Thêm những dòng lệnh dưới vào file ***main.c***

```
#include "N78E055A_059A_517A.h"   //Define register of MCU N78E055A/059A/517A
#include "Typedef.h"   //define type for Delay function in Delay.c
#include "Delay.h"              //header of Delay.c


void main (void)
{
   while(1)
   {
        P2 = 0x00;              //PORT 2 is low level
        P37 = 0;                //Pin 7 of Port 3 is low level
        Delay1ms(500);//Delay 500 mini second
        P2 = 0xFF;              //PORT 2 is high level
        P37 = 1;                //Pin 7 of Port 3 is high level
```

```
        Delay1ms(500);//Delay 500 mini second
    }
}
```



- ***Kết quả:***



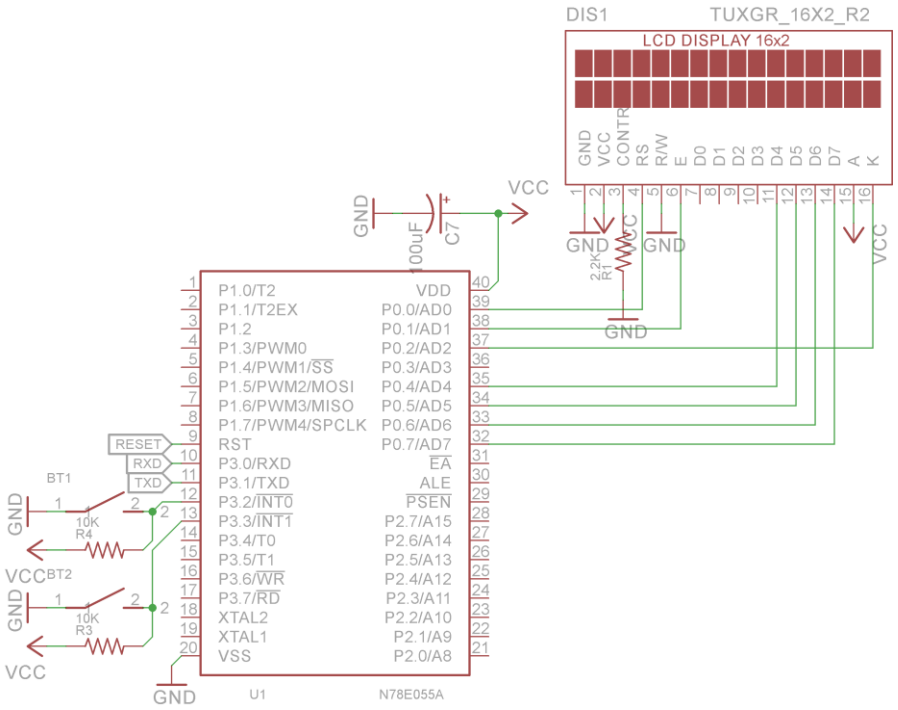
4. Serial Port (UART)

a. Giới thiệu

- N78E059A/N78E055A có một cổng nối tiếp song công hỗ trợ 3 chế độ cho UART.
- Các thanh ghi liên quan:
  - SCON: thiết lập chế độ cho serial port
  - PCON: với bit SMOD (PCON.7) thiết lập double baud rate.
  - SBUF: bộ đệm dữ liệu.

b. Ví dụ - *(Z008_UART_DisplayCharToLCD)*

- ***Ý tưởng:*** xây dựng mạch dùng N78E055A, có LCD Text 16x2, kết nối với máy tính qua UART, sử dụng phần mềm Terminal hoặc Hercules hoặc một số phần mềm khác có chức năng tương tác UART (COM) để tương tác, có nút nhấn BT1, BT2. Nhấn BT1, BT2 -> hiển thị báo "BT1 pressed!", "BT2 pressed!". Truyền text từ Terminal xuống N78E055A và hiển thị text đó ra LCD.
- ***Xây dựng mạch:*** BT1 kết nối P32, BT2 kết nối P33, LCD kết nối port 0, truyền 4-bit. Về kết nối UART thì ta sử dụng ngay kết nối lúc truyền ISP (nạp code cho N78E055A) như vậy ta vừa nạp code vừa giao tiếp N78E055A theo UART.



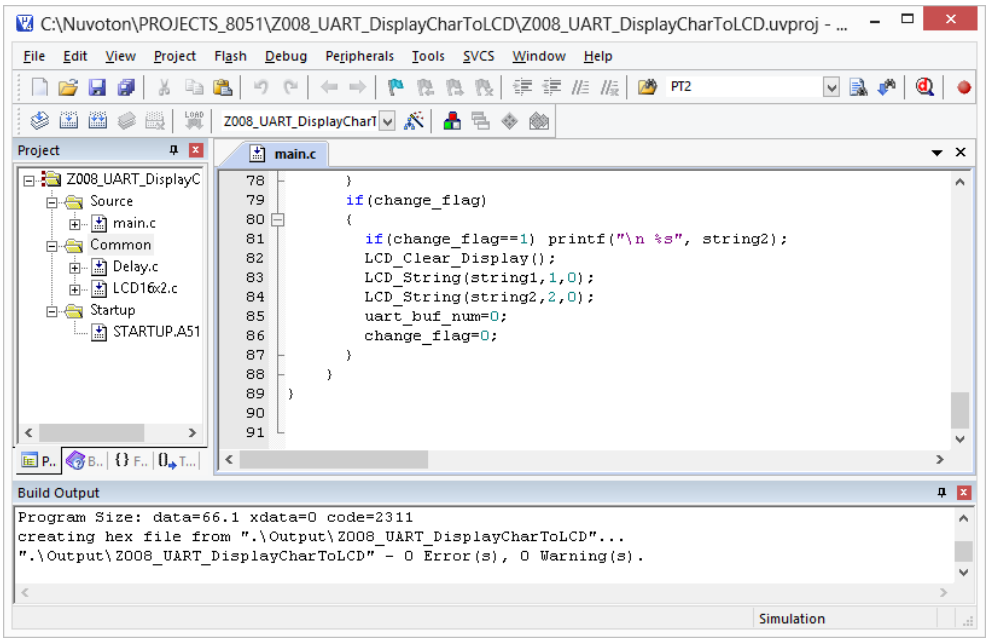
- ***Code:***

Thêm các dòng code này vào file ***main.c***:

```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "LCD16x2.h"
#include "Delay.h"
#include <stdio.h>              //for "printf"

#define BAUD_RATE       9600
#define U16BAUD_TIMER2  65536-(22118400/BAUD_RATE/32)       //65536-
(Fsys/Baud_Rate/32)

char uart_buf_num=0;
char change_flag=0;
```

```
char string1[17] = "PC<->UART<->LCD";
char string2[17] = "Click Button!";

void UART_ISR (void) interrupt 4    //interrupt address is 0x0023
{
    if (RI)                    //check Tx or Rx interrupt
    {
            if(uart_buf_num>15) uart_buf_num=0;
            string2[uart_buf_num]= SBUF;
            uart_buf_num++;
            string2[uart_buf_num]= '\0';
            change_flag =2;
            RI = 0;                //clear RI by software for next reception
    }
}
void Timer2_ISR (void) interrupt 5  //interrupt address is 0x002B
{
   EXF2 = 0;        //Clear Timer 2 External Flag
}
void Timer2_Init (void)
{
   TCLK = 1;                //select Timer 2 as transmission baud rate source
   RCLK = 1;                //select Timer 2 as receive baud rate source
   TL2 = RCAP2L = U16BAUD_TIMER2;  //baud rate 9600bps@22.1184MHz
   TH2 = RCAP2H = U16BAUD_TIMER2 >> 8;

   EXEN2 = 1;                //enable T2EX pin
   ET2 = 1;                //Timer 2 interrupt enable, in baud rate generator mode,
                        //T2EX pin interrupt still works
   TR2 = 1;                //Timer 2 run
}
void main (void)
{
    LCD_Init();                    //Init LCD
    LCD_String(string1,1,0);
    LCD_String(string2,2,0);
    LCD_PIN_LIGHT=0;
    Timer2_Init(); //Init timer 2 (source clock of UART)
    SCON = 0x52;  //initial UART as mode 1, receive enable,
                //TI should be set before using "printf"

    printf
("\n*=================================================
====*");
    printf ("\n*          N78E055A_059A_517A UART Mode 1 Demo     *");
    printf
("\n*=================================================
====*");
    printf ("\n 1. Put character in Terminal software -> display to LCD");
    printf ("\n 2. Press BT1, BT2 -> Display to PC (Terminal software)\n");
```

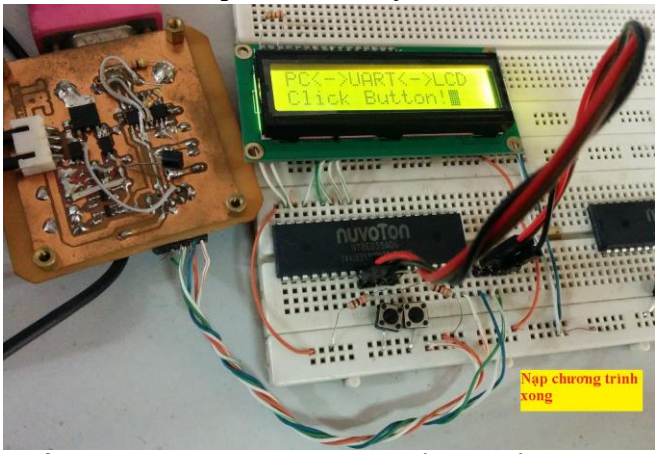
```
    ES = 1;             //enable UART interrupt
    EA = 1;             //enable global interrupt
    uart_buf_num=0;
    while(1)
    {
        if(P32 == 0)    //If P32 pin pressed -> increment number
        {
            while(!P32);    //Wait P32 release!
            sprintf(string2,"BT1 pressed!");
            change_flag=1;
        }
        if(P33 == 0)            //If P32 pressed -> increment number
        {
            while(!P33);    //Wait P32 release!
            sprintf(string2,"BT2 pressed!");
            change_flag=1;
        }
        if(change_flag)
        {
            if(change_flag==1) printf("\n %s", string2);
            LCD_Clear_Display();
            LCD_String(string1,1,0);
            LCD_String(string2,2,0);
            uart_buf_num=0;
            change_flag=0;
        }
    }
}
```



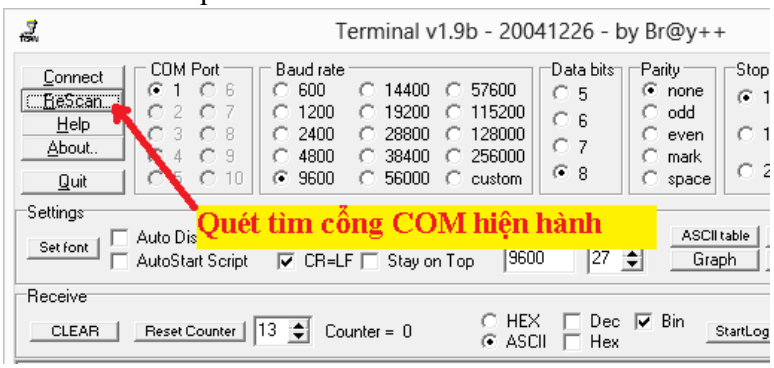
- ***Kết quả:***

  Bước 1: Ráp board mạch như bình thường, cắm cáp USB-COM vào máy tính trước (với những máy không có cổng COM sẵn) -> kết nối với board N78E055A, cung cấp nguồn cho board này như đã hướng dẫn nạp code ở phần trước.

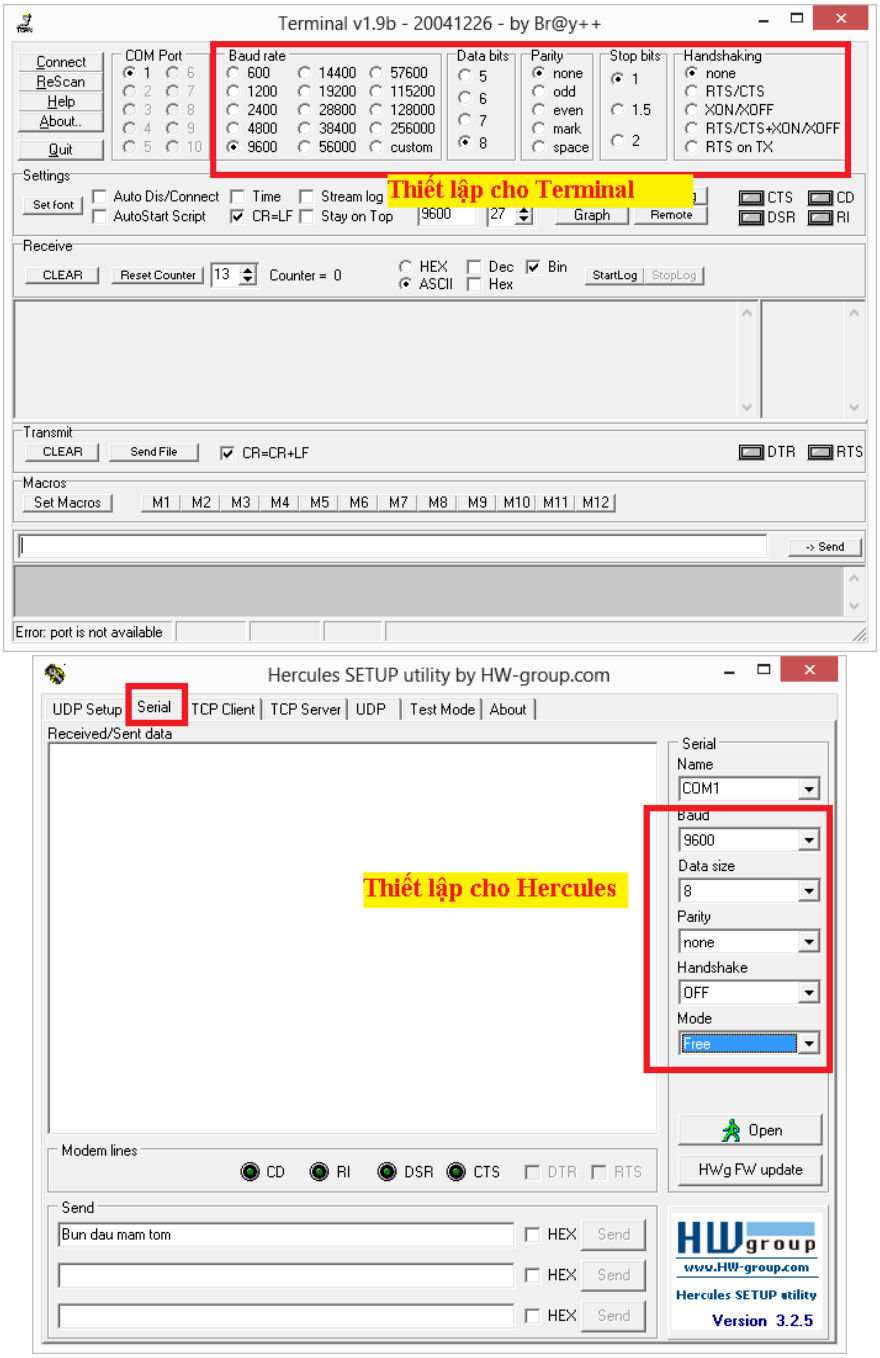
  Bước 2: mở phần mềm nạp code *Nuvoton ISP-ICP Utility* (nếu dùng windows7/8/8.1 thì nên mở bằng quyền administrator) -> nạp code *Z008_UART_DisplayCharToLCD* vào N78E055A -> sau đó, nên tắt phần mềm này đi.



  Bước 3: Mở phần mềm Terminal hoặc Hercules. Ở đây cổng COM (UART) mà tác giả thử nghiệm là COM1 (chính là cổng nạp ISP cho N78E055A). Nếu chưa nhận diện được cổng COM nào thì ta quét.



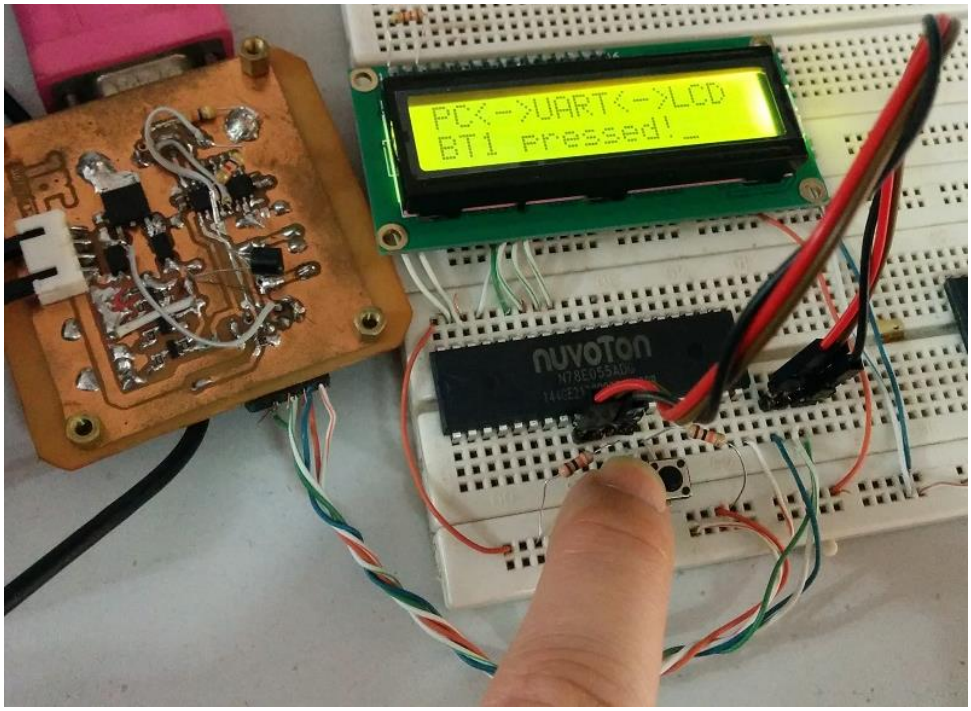
  Nếu nhận diện được thì ta tiến hành thiết lập thông số: chọn Baud rate 9600, 8 Databits, none Parity, 1 bit stop như trong code ta đã thiết lập.

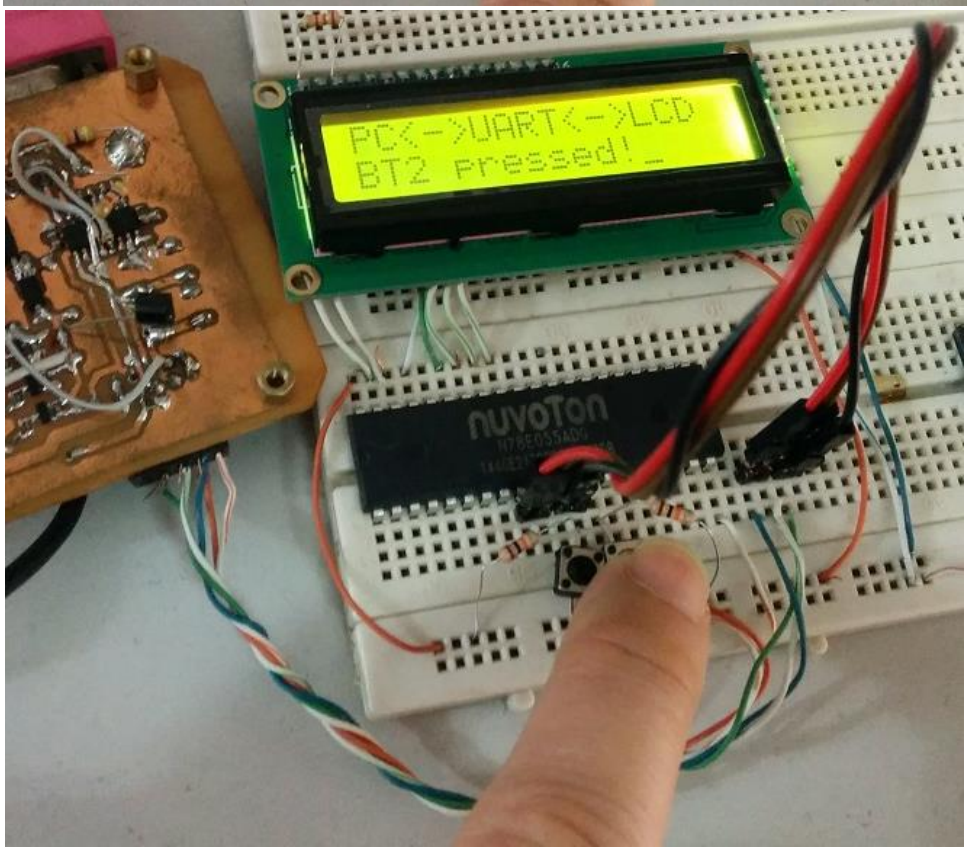


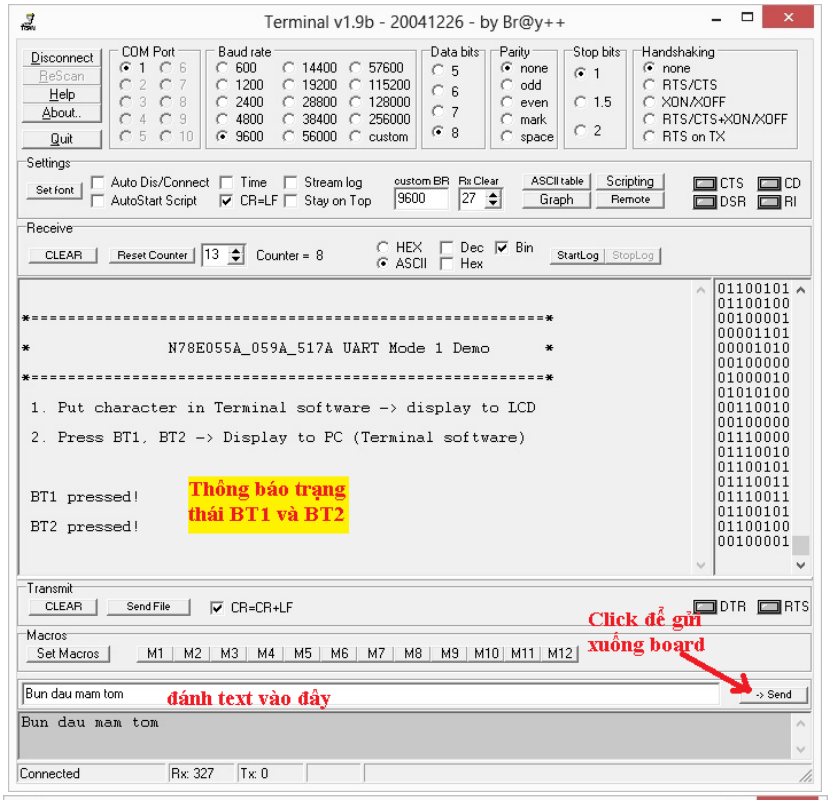
Bây giờ ta tiến hành kết nối phần mềm với cổng COM:

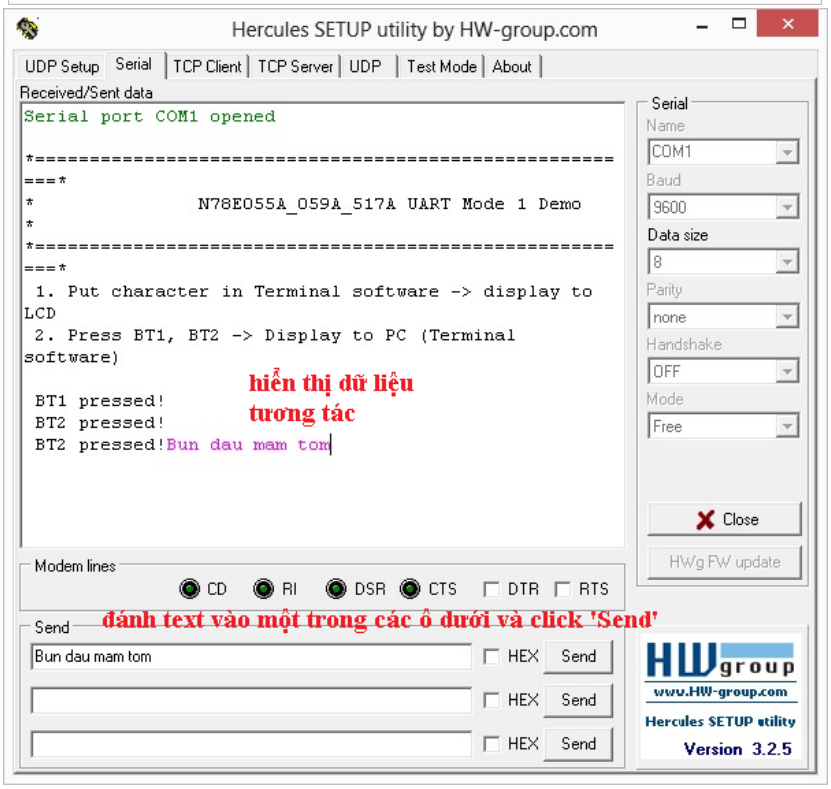
Click vào Connect trên Terminal hoặc Open với Hercules. Nếu với Terminal thì sẽ tự động reset mạch N78E055A, còn Hercules thì không (ta có thể reset board N78E055A để MCU chạy lại từ đầu).

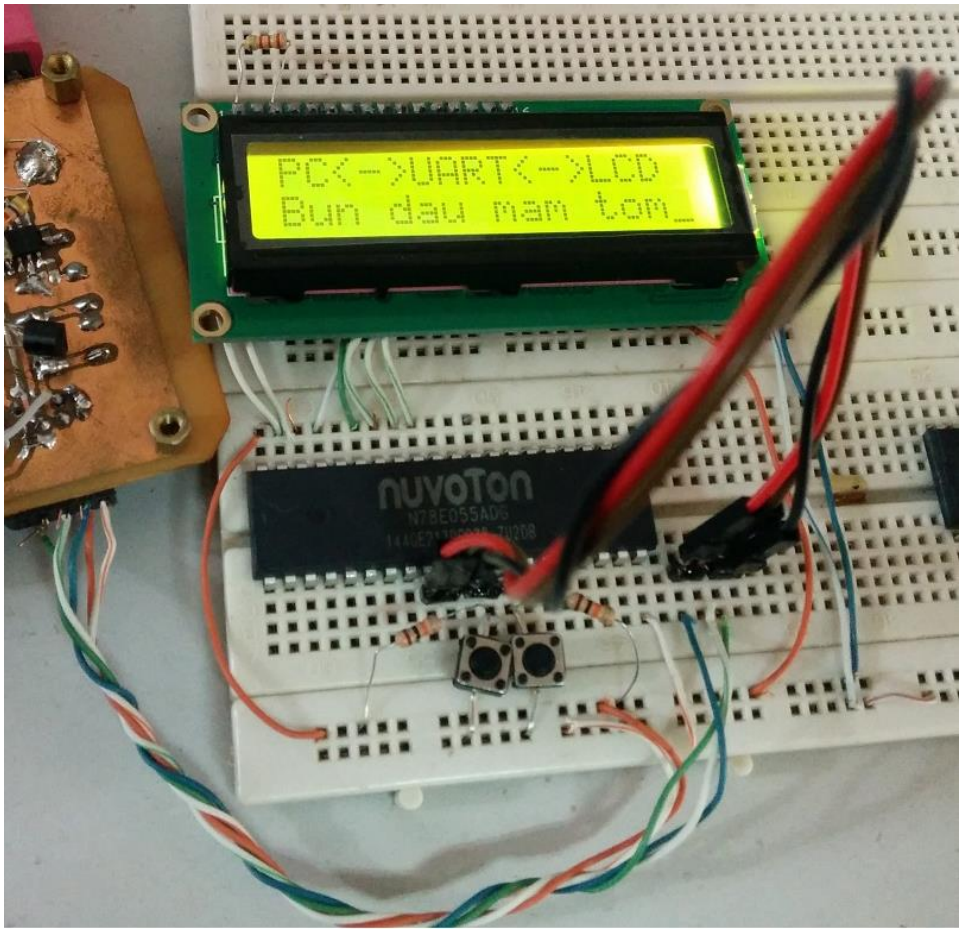
Bước 4: nhấn nút BT1 rồi BT2 sẽ nhận được thông báo trên Terminal/ Hercules. Đánh text vào Terminal rồi "Send" cho hiển thị ra LCD.


PC<->UART<->LCD
BT1 pressed!_
nuvoTon


PC<->UART<->LCD
BT2 pressed!_
nuvoTon

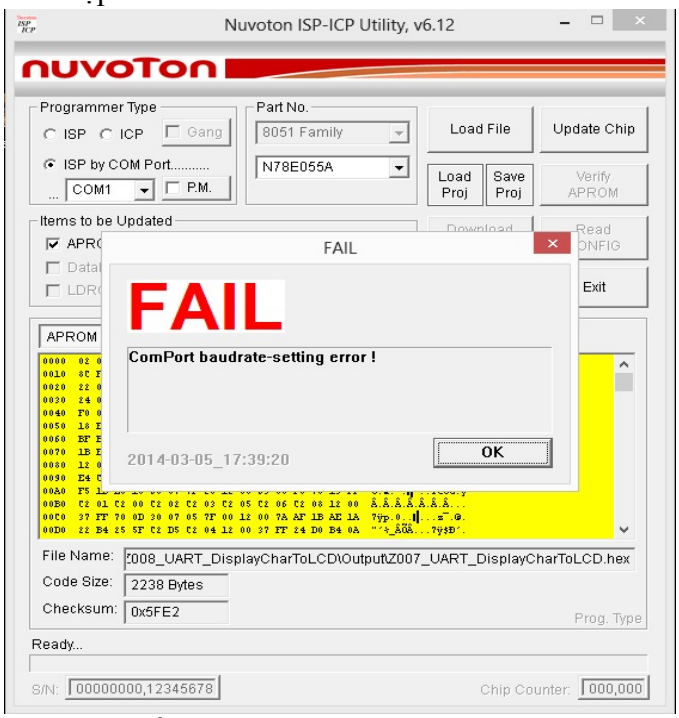

Terminal v1.9b - 20041226 - by Br@y++
N78E055A_059A_517A UART Mode 1 Demo
1. Put character in Terminal software -> display to LCD
2. Press BT1, BT2 -> Display to PC (Terminal software)
BT1 pressed!
BT2 pressed!
Thông báo trạng thái BT1 và BT2
Click để gửi xuống board
Bun dau mam tom
đánh text vào đây
Connected


Hercules SETUP utility by HW-group.com
Serial port COM1 opened
N78E055A_059A_517A UART Mode 1 Demo
1. Put character in Terminal software -> display to LCD
2. Press BT1, BT2 -> Display to PC (Terminal software)
BT1 pressed!
BT2 pressed!
BT2 pressed!Bun dau mam tom
hiển thị dữ liệu tương tác
đánh text vào một trong các ô dưới và click 'Send'
Bun dau mam tom
Version 3.2.5



c. Chú ý:

- Do N78E055A chỉ có 1 port UART, ta vừa dùng để nạp ISP, vừa làm giao tiếp UART như vì dụ trên -> khi nạp đổ code mới cho N78E055A có thể sẽ báo lỗi:

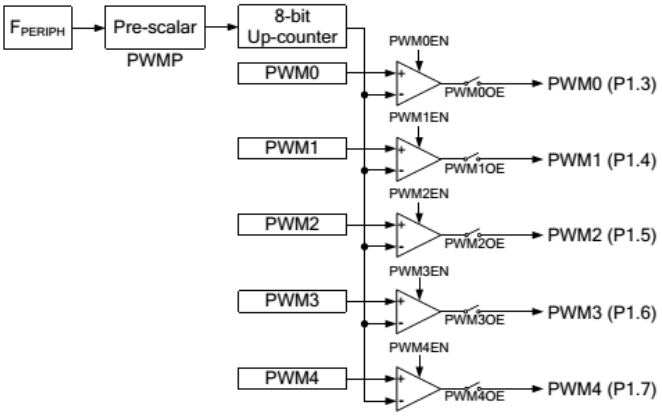


- Ta chỉ cần nạp lại 1 hoặc 2 lần là OK (click vào ‘’Update Chip’’).

5. PWM

a. Giới thiệu

- N78E059A/N78E055A có 5 kênh ngõ ra PWM (PWM0~PWM4) kết nối với các chân (P13~P17). Với tầm thang 8bit (0~255)
- Các thanh ghi liên quan:
    - PWMCON0, PWMCON1 – thiết lập cho các kênh PWM
    - PWMP – Chu kỳ PWM (8bit). Chung cho các kênh.
    - PWMx – Duty của các kênh, x = 0~4.



Hình: Sơ đồ khối chức năng PWM

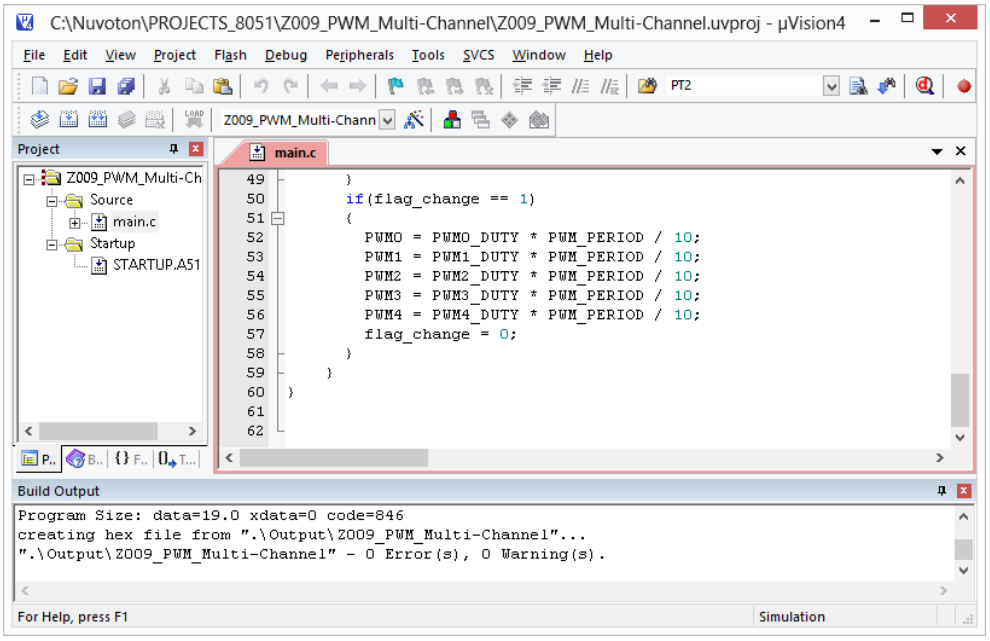
b. Ví dụ 1 – Băm xung đa kênh (*Z009_PWM_Multi-Channel)*

- ***Ý tưởng:*** N78E055A có 5 kênh ngõ ra PWM, ta test bằng cách dùng LED cắm vào các chân ngõ ra PWM hoặc tốt hơn là dùng dao động kế (Oscilloscope) để xem các dạng sóng ngõ ra.
- ***Xây dựng mạch:*** Kết nối BT1 với P32, BT2 với P33 với mục đính: tăng/ giảm duty cùa PWM. Ta dùng LED kết nối ngõ ra PWM (P13 ~ P17). Thực nghiệm thấy rằng các chân ra PWM tạo mức 1 khá yếu (LED sáng yếu) nên ta sẽ đấu <u>*ngược lại*</u> như hình dưới đây. Như vậy khi duty càng lớn thì LED càng sáng yếu. Hãy chú ý điều này.
- ***Code:***

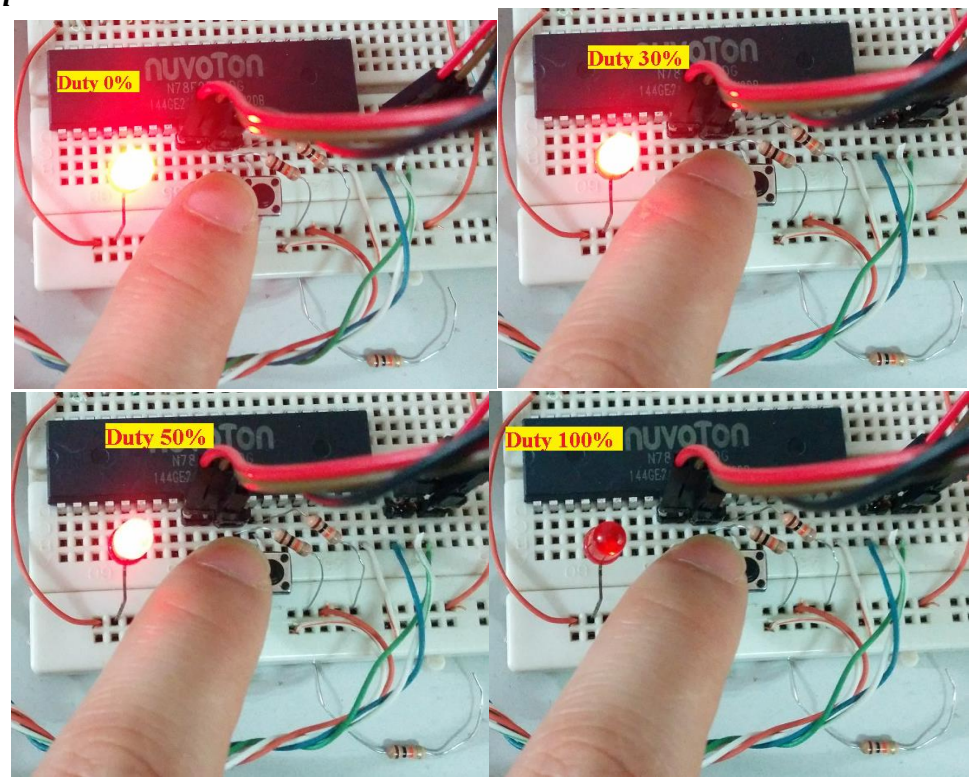
  Thêm các dòng code này vào file ***main.c***:

```
#include "N78E055A_059A_517A.h"

#define SYS_CLOCK      22118400
#define PWM_PERIOD     255

short PWM0_DUTY = 1;
short PWM1_DUTY = 3;
short PWM2_DUTY = 5;
short PWM3_DUTY = 7;
short PWM4_DUTY = 9;
void main(void)
{
    char flag_change = 0;
    P13 = P14 = P15 = P16 = P17 = 1; //PWM0~4 shared GPIO P1.3~P1.7 should set 1
before output PWM

  PWMP = PWM_PERIOD;                //initial PWM period and duty
  PWM0 = PWM0_DUTY * PWM_PERIOD / 10;
```

```
    PWM1 = PWM1_DUTY * PWM_PERIOD / 10;
    PWM2 = PWM2_DUTY * PWM_PERIOD / 10;
    PWM3 = PWM3_DUTY * PWM_PERIOD / 10;
    PWM4 = PWM4_DUTY * PWM_PERIOD / 10;

    PWMCON0 |= 0x55;          //PWM runs
    PWMCON1 |= 0x01;
    PWMCON0 |= 0xCC;          //PWM outputs
    PWMCON1 |= 0x04;

     while(1)
     {
            if(P32==0)
            {
                  while(!P32);
                  if( PWM0_DUTY<=9 ) PWM0_DUTY ++;      else PWM0_DUTY = 0;
                  if( PWM1_DUTY<=9 ) PWM1_DUTY ++;      else PWM1_DUTY = 0;
                  if( PWM2_DUTY<=9 ) PWM2_DUTY ++;      else PWM2_DUTY = 0;
                  if( PWM3_DUTY<=9 ) PWM3_DUTY ++;      else PWM3_DUTY = 0;
                  if( PWM4_DUTY<=9 ) PWM4_DUTY ++;      else PWM4_DUTY = 0;
                  flag_change = 1;
            }
            if(P33==0)
            {
                  while(!P33);
                  if( PWM0_DUTY>=1 ) PWM0_DUTY --; else PWM0_DUTY = 10;
                  if( PWM1_DUTY>=1 ) PWM1_DUTY --; else PWM1_DUTY = 10;
                  if( PWM2_DUTY>=1 ) PWM2_DUTY --; else PWM2_DUTY = 10;
                  if( PWM3_DUTY>=1 ) PWM3_DUTY --; else PWM3_DUTY = 10;
                  if( PWM4_DUTY>=1 ) PWM4_DUTY --; else PWM4_DUTY = 10;
                  flag_change = 1;
            }
            if(flag_change == 1)
            {
                  PWM0 = PWM0_DUTY * PWM_PERIOD / 10;
                  PWM1 = PWM1_DUTY * PWM_PERIOD / 10;
                  PWM2 = PWM2_DUTY * PWM_PERIOD / 10;
                  PWM3 = PWM3_DUTY * PWM_PERIOD / 10;
                  PWM4 = PWM4_DUTY * PWM_PERIOD / 10;
                  flag_change = 0;
            }
     }
}
```


C:\Nuvoton\PROJECTS_8051\Z009_PWM_Multi-Channel\Z009_PWM_Multi-Channel.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
PT2
Z009_PWM_Multi-Chann
Project
Z009_PWM_Multi-Ch
Source
main.c
Startup
STARTUP.A51
main.c
49 }
50 if(flag_change == 1)
51 {
52 PWM0 = PWM0_DUTY * PWM_PERIOD / 10;
53 PWM1 = PWM1_DUTY * PWM_PERIOD / 10;
54 PWM2 = PWM2_DUTY * PWM_PERIOD / 10;
55 PWM3 = PWM3_DUTY * PWM_PERIOD / 10;
56 PWM4 = PWM4_DUTY * PWM_PERIOD / 10;
57 flag_change = 0;
58 }
59 }
60 }
61
62
Build Output
Program Size: data=19.0 xdata=0 code=846
creating hex file from ".\Output\Z009_PWM_Multi-Channel"...
".\Output\Z009_PWM_Multi-Channel" - 0 Error(s), 0 Warning(s).
For Help, press F1
Simulation

- ***Kết quả:***

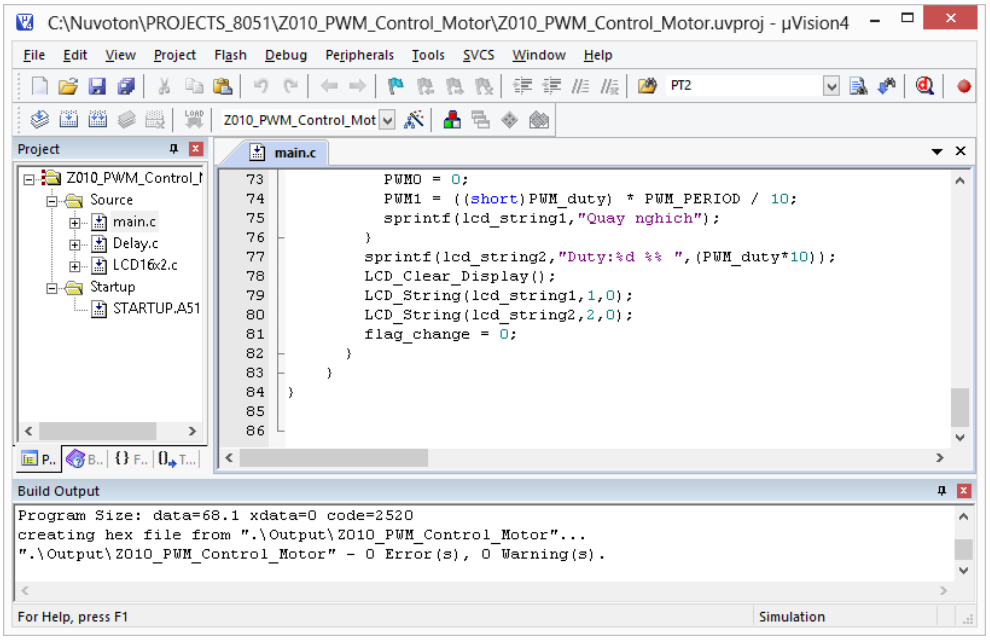

Duty 0%
Duty 30%
Duty 50%
Duty 100%

c. Ví dụ 2 – Điều khiển động cơ (*Z010_PWM_Control_Motor)*

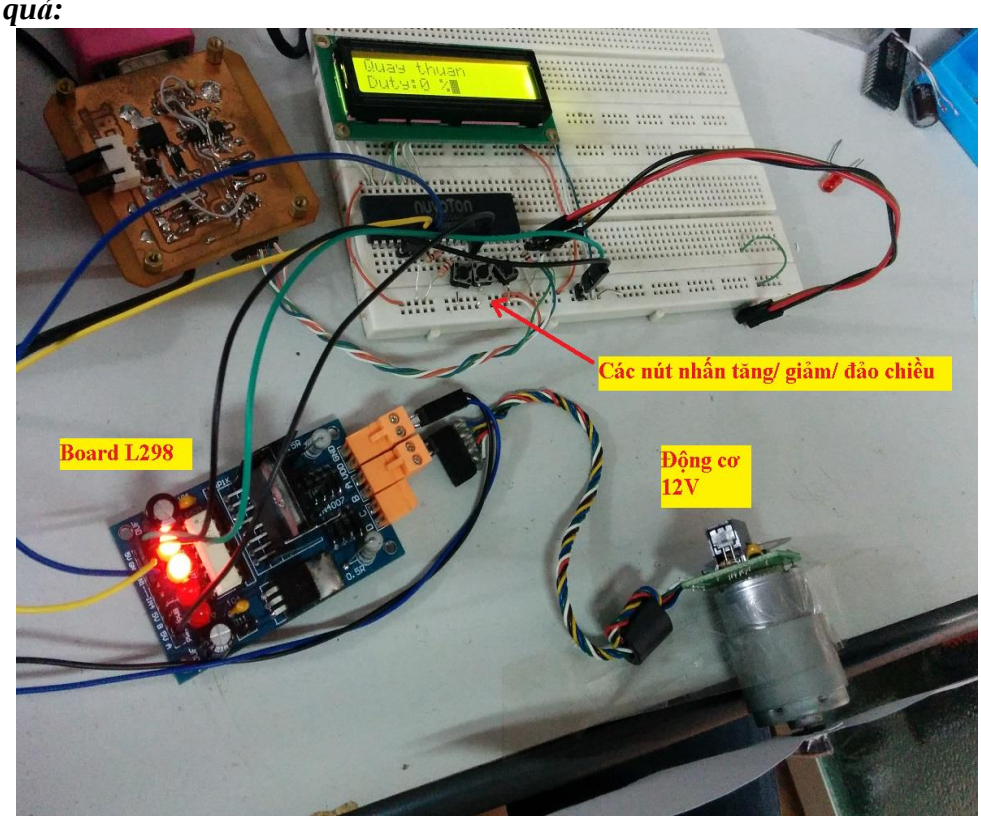
- ***Ý tưởng:*** Mạch điện dùng N78E055A điều khiển tốc độ động cơ, có thể đảo chiều động cơ, tương tác qua nút nhấn.
- ***Xây dựng mạch:*** Ráp mạch với BT1 kết nối P32 (tăng duty PWM), BT2 kết nối P33 (giảm duty PWM), BT3 kết nối P37 (đảo chiều động cơ), LCD Text qua port 0 (dùng hiển thị duty). Dùng board công suất L298 (có thể tìm trên mạng: banlinhkien.com), kết nối chân cấp phép L298 qua P34.
- ***Code:*** sử dụng thư viện LCD16x2 đã viết ở phần GPIO.
  Thêm các dòng code này vao file ***main.c***

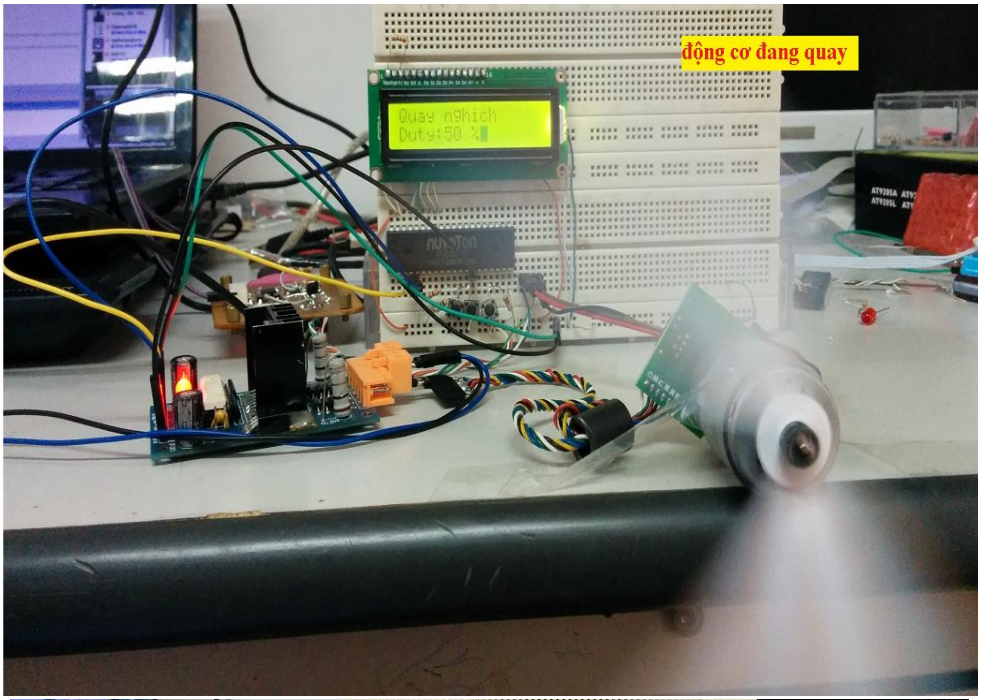
```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Delay.h"
#include "stdio.h"
#include "LCD16x2.h"

#define SYS_CLOCK      22118400
#define PWM_PERIOD     255

#define FORWARD                0
#define REVERSE                1

int PWM_duty = 0;
char direction = FORWARD;
void main(void)
{
    char lcd_string1[17]="FORWARD";
    char lcd_string2[17]="Duty: 0%";
    char flag_change = 1;
    LCD_Init();
    LCD_String("PWM",1,5);
    LCD_PIN_LIGHT=0;

    P37=0;
    P13 = P14 = 1;      //PWM0~1 shared GPIO P1.3~P1.4 should set 1 before output PWM

   PWMP = 255;    //initial PWM period and duty
   PWM0 = 0;
   PWM1 = 0;

   PWMCON0 |= 0x55;       //PWM runs
   PWMCON1 |= 0x00;
   PWMCON0 |= 0x0C;     //PWM outputs

    direction = FORWARD;
    P37 = 1;
     while(1)
    {
            if(P32==0)                              //Tang toc do
            {
                    while(!P32);
```

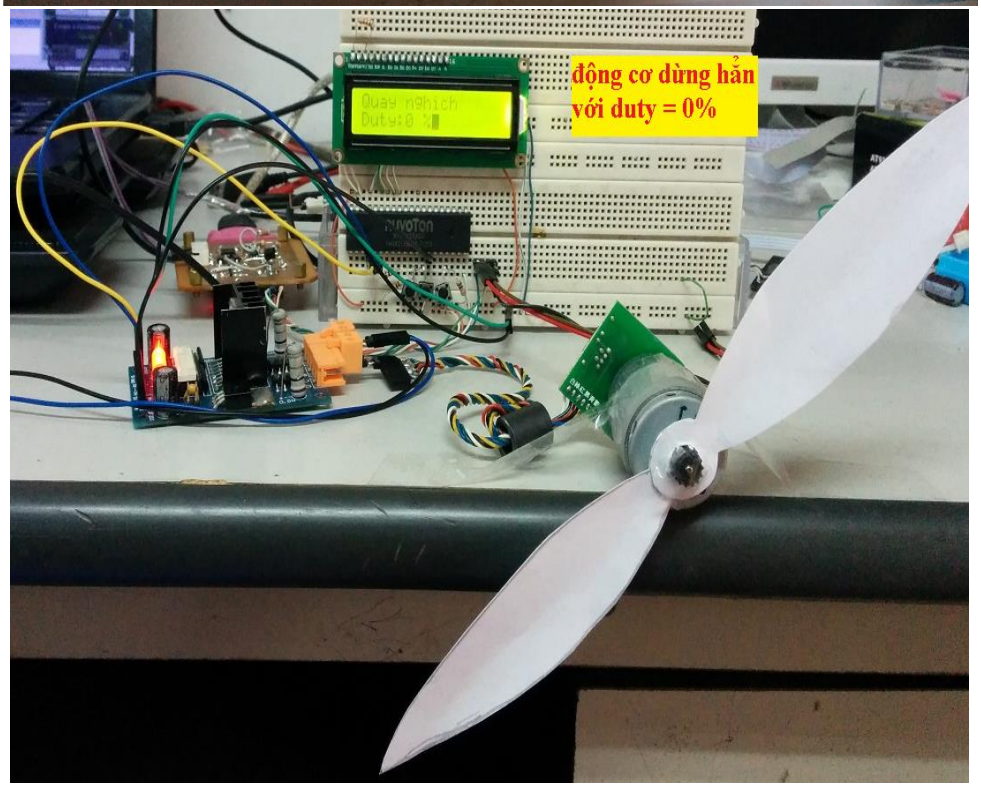
```
            if( PWM_duty<=9 ) PWM_duty ++;
            flag_change = 1;
        }
        if(P33==0)                          //Giam toc do
        {
            while(!P33);
            if( PWM_duty>=1 ) PWM_duty --;
            flag_change = 1;
        }
        if(P37==0)                          //Dao dong co
        {
            while(!P37);
            direction = ~ direction;
            flag_change = 2;
        }
        if(flag_change>0)
        {
            if(flag_change > 1)
            {
                PWM0 = 0;       //Cho PWM0 va PWM1 = 0 -> DC dung.
                PWM1 = 0;
                Delay1ms(500);//Doi DC dung an toan.
            }
            if(direction==FORWARD)
            {
                PWM1 = 0;
                PWM0 = ((short)PWM_duty) * PWM_PERIOD / 10;
                sprintf(lcd_string1,"Quay thuan");
            }
            else
            {
                PWM0 = 0;
                PWM1 = ((short)PWM_duty) * PWM_PERIOD / 10;
                sprintf(lcd_string1,"Quay nghich");

            }
            sprintf(lcd_string2,"Duty:%d %% ",(PWM_duty*10));
            LCD_Clear_Display();
            LCD_String(lcd_string1,1,0);
            LCD_String(lcd_string2,2,0);
            flag_change = 0;
        }
    }
}
```


C:\Nuvoton\PROJECTS_8051\Z010_PWM_Control_Motor\Z010_PWM_Control_Motor.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Z010_PWM_Control_M
Source
main.c
Delay.c
LCD16x2.c
Startup
STARTUP.A51
main.c
PWM0 = 0;
PWM1 = ((short)PWM_duty) * PWM_PERIOD / 10;
sprintf(lcd_string1,"Quay nghich");
sprintf(lcd_string2,"Duty:%d %% ",(PWM_duty*10));
LCD_Clear_Display();
LCD_String(lcd_string1,1,0);
LCD_String(lcd_string2,2,0);
flag_change = 0;
Build Output
Program Size: data=68.1 xdata=0 code=2520
creating hex file from ".\Output\Z010_PWM_Control_Motor"...
".\Output\Z010_PWM_Control_Motor" - 0 Error(s), 0 Warning(s).
For Help, press F1
Simulation

- ***Kết quả:***


Các nút nhấn tăng/ giảm/ đảo chiều
Board L298
Động cơ 12V

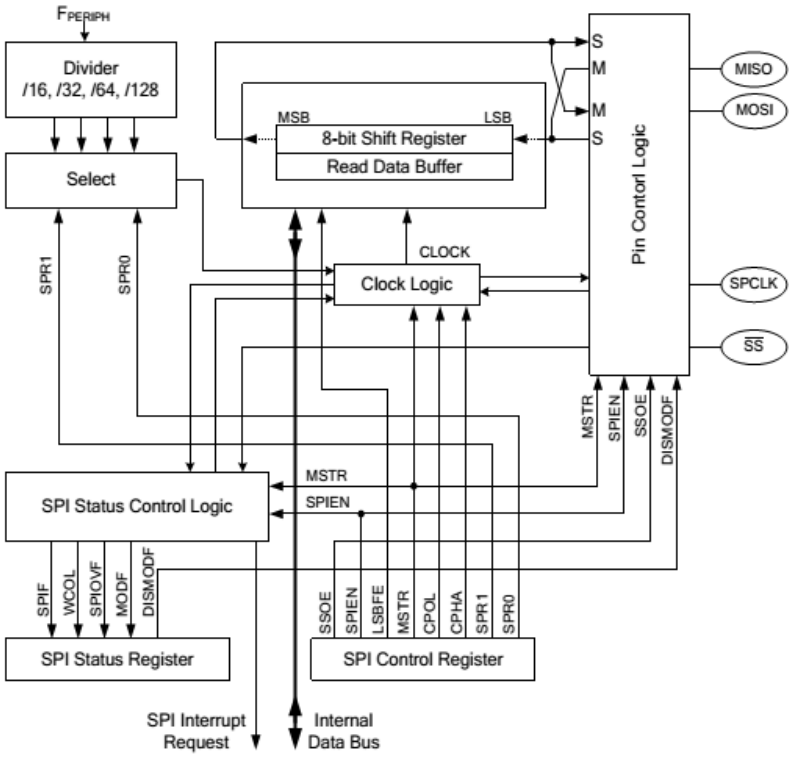

động cơ đang quay
Quay nghich
Duty:50 %


động cơ dừng hẳn
với duty = 0%
Quay nghich
Duty:0 %

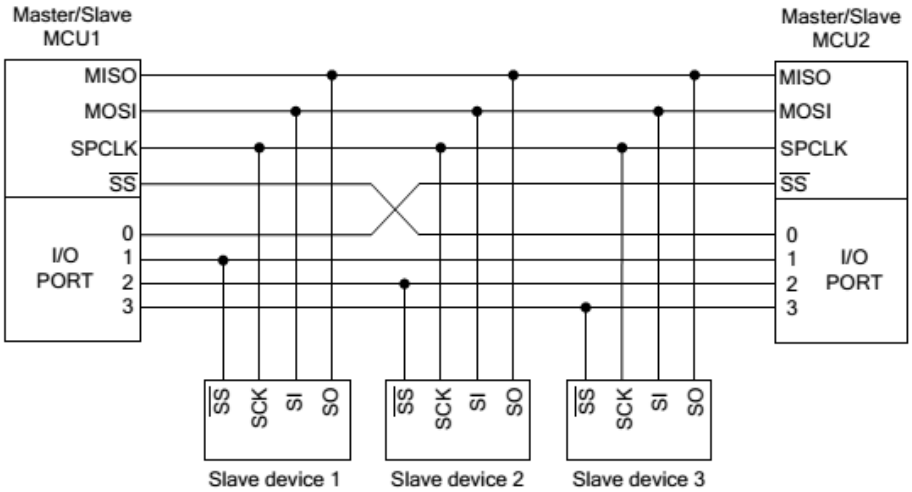
6. SPI

a. Giới thiệu

- N78E055A có 1 cổng truyền thông SPI, hỗ trợ chế độ Master và Slave.



Hình: Sơ đồ khối chức năng SPI



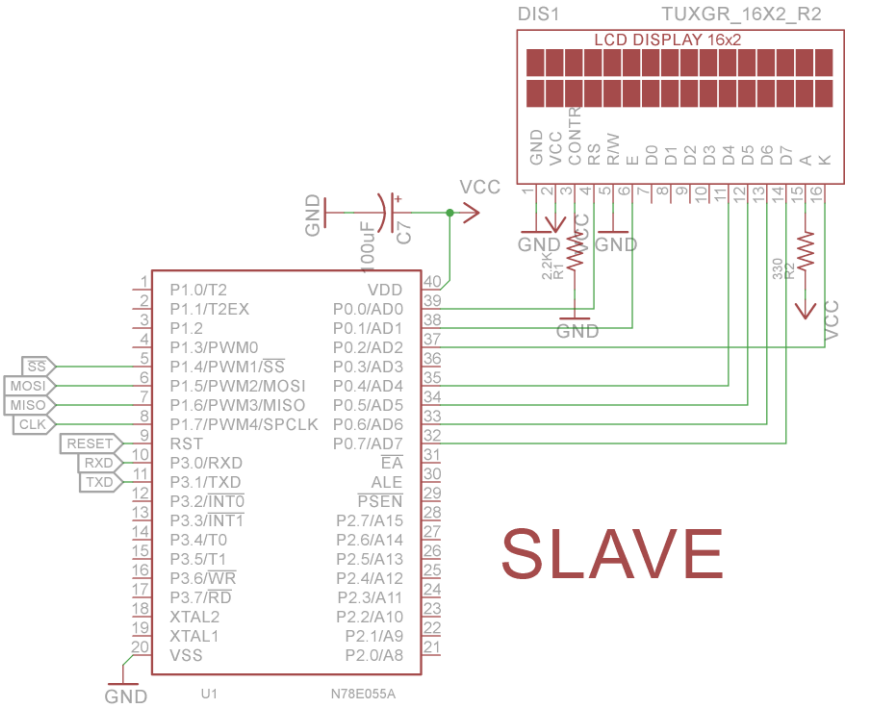
Hình: Kết nối nhiều Master, nhiều Slave trên SPI

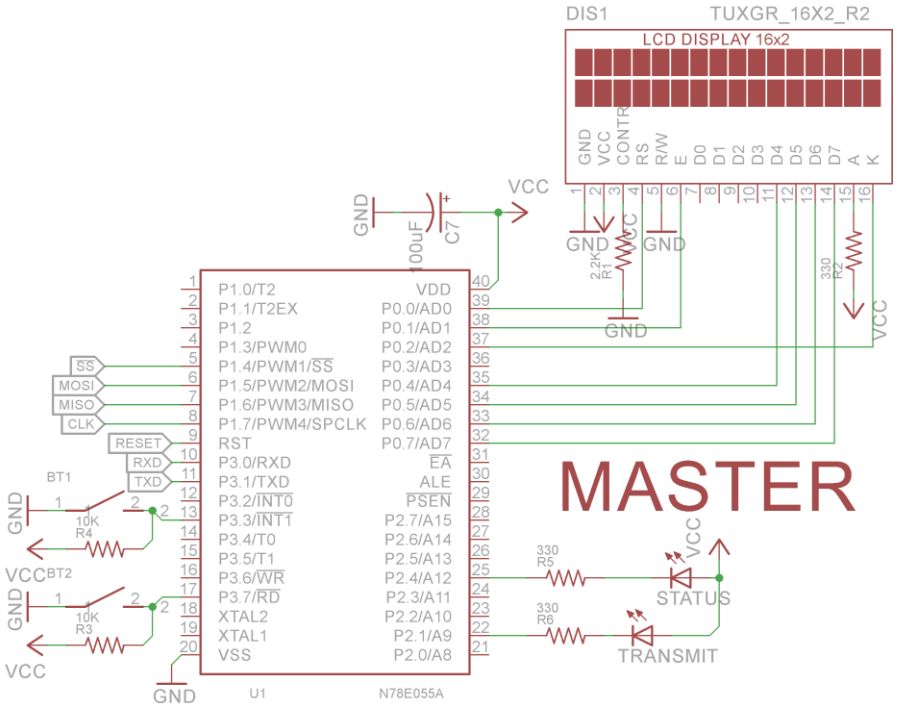
- Các thanh ghi liên quan:
  - SPCR: thiết lập cho SPI
  - SPSR: chứa các cờ báo của SPI
  - SPDR: bộ đệm dữ liệu SPI 8bit.

b. Ví dụ (*Z011_SPI_Transmit_2_MCU*)

- ***Ý tưởng:*** Truyền dữ liệu giữa 2 thiết bị qua giao tiếp SPI (2 MCU). đây, dùng 2 MCU N78E055A (1 đóng vai trò Master, 1 đóng vai trò Slave). Master có nút nhấn BT1, BT2, có 2 LED dùng báo trang thái lỗi, trạng thái đang truyền dữ liệuSPI. Slave có LCD hiển thị thông báo trang thái nút nhấn nào được nhấn.

- ***Xây dựng mạch:***
  - Master: BT1 kết nối P33, BT2 kết nối P37, LED báo trạng thái truyền nối P21, LED báo nối P24.
  - Slave: LCD Text 16x2 kết nối với port 0, chế độ truyền 4-bit.
  - Hai MCU kết nối với nhau qua SPI với các chân P14 ~ P17.





- ***Code:***
  - *Thư viện LCD16x2 và Delay ta đã tạo từ ví dụ phần GPIO*
  - *Thêm các dòng code này cho file* ***main_Master.c***

```
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Define.h"
```

```
#include "Delay.h"

#define set_SSOE      SPCR |= SET_BIT7
#define set_SPE       SPCR |= SET_BIT6
#define set_LSBFE     SPCR |= SET_BIT5
#define set_MSTR      SPCR |= SET_BIT4
#define set_CPOL      SPCR |= SET_BIT3
#define set_CPHA      SPCR |= SET_BIT2
#define set_SPR1      SPCR |= SET_BIT1
#define set_SPR0      SPCR |= SET_BIT0

#define set_DRSS      SPSR |= SET_BIT3
#define set_ESPI      EIE  |= SET_BIT0

#define clr_SSOE      SPCR &= CLR_BIT7
#define clr_SPE       SPCR &= CLR_BIT6
#define clr_LSBFE     SPCR &= CLR_BIT5
#define clr_MSTR      SPCR &= CLR_BIT4
#define clr_CPOL      SPCR &= CLR_BIT3
#define clr_CPHA      SPCR &= CLR_BIT2
#define clr_SPR1      SPCR &= CLR_BIT1
#define clr_SPR0      SPCR &= CLR_BIT0

#define clr_DRSS      SPSR &= CLR_BIT3
#define clr_SPIF      SPSR &= CLR_BIT7


#define SS_PIN                  P14
#define SPI_PIN_STATUS          P24
#define SPI_PIN_TRANSMIT        P21
#define BT1                                                     P33
#define BT2                                                     P37

#define CMD_CHECK       0x01
#define CMD_CHECK_BUS       0x02
#define CMD_BT1 0x03
#define CMD_BT2 0x04
#define OK          0x00
#define ERROR       0xFF

//-------------------------------------------------------
void SPI_Initial(void)
{
   set_SPR0;                //SPI clock = Fosc/128
   set_SPR1;

   set_DRSS;                //SS General purpose I/O ( No Mode Fault )
   clr_SSOE;

   set_MSTR;                    //SPI in Master mode
   clr_LSBFE;               //MSB first
```

```
   clr_CPOL;
   set_CPHA;
   set_SPE;     //Enable SPI function
}
//-------------------------------------------------------
//return 0 if ok
//return 1 if error
char SPI_Send_Byte(unsigned char _data)
{
    SPI_PIN_STATUS = 1;          //Tat LED bao Error
    SPI_PIN_TRANSMIT = 0;        //Dang truyen du lieu tren SPI
    SS_PIN = 0;                  //Chon Slave

    clr_SPIF;                    //Xoa co bao ngat SPI
    SPDR = _data;            //Day du lieu len SPI Bus
    while((SPSR&0x80) == 0x00);  //Cho khi du lieu dc truyen di -> khi truyen di dc -> nagt
bao truyen OK
    Delay1ms(300);
    clr_SPIF;                //Xoa co bao ngat
    SPDR = CMD_CHECK;
    while((SPSR&0x80) == 0x00);
    clr_SPIF;
    if(SPDR!=OK)
    {
            SPI_PIN_STATUS = 0;
            SPI_PIN_TRANSMIT = 1;
            return 1;
    }
    SPI_PIN_TRANSMIT = 1;
    SS_PIN=1;
return 0;}
//-------------------------------------------------------------
void main(void)
{
   SPI_Initial();
    while(1)
    {
            if(SPI_Send_Byte(CMD_CHECK_BUS)==0) break;
            Delay1ms(500);
    }
   while(1)
   {
            if(BT1==0)
            {
                    while(!BT1);
                    SPI_Send_Byte(CMD_BT1);
            }
            if(BT2==0)
            {
                    while(!BT2);
                    SPI_Send_Byte(CMD_BT2);
```

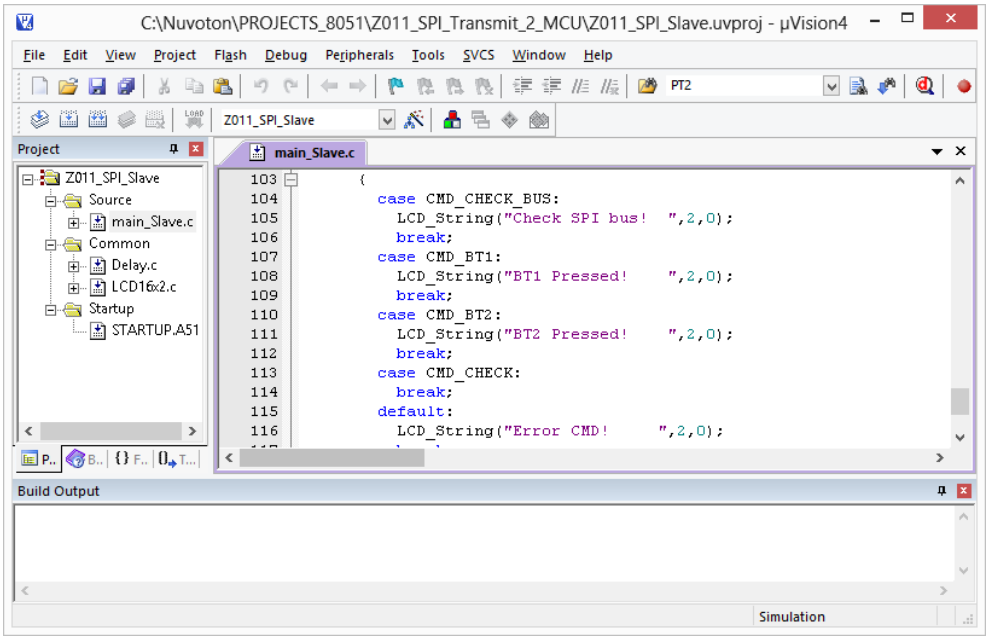
```
            }
        }
    }
//-------------------------------------------------------------
```

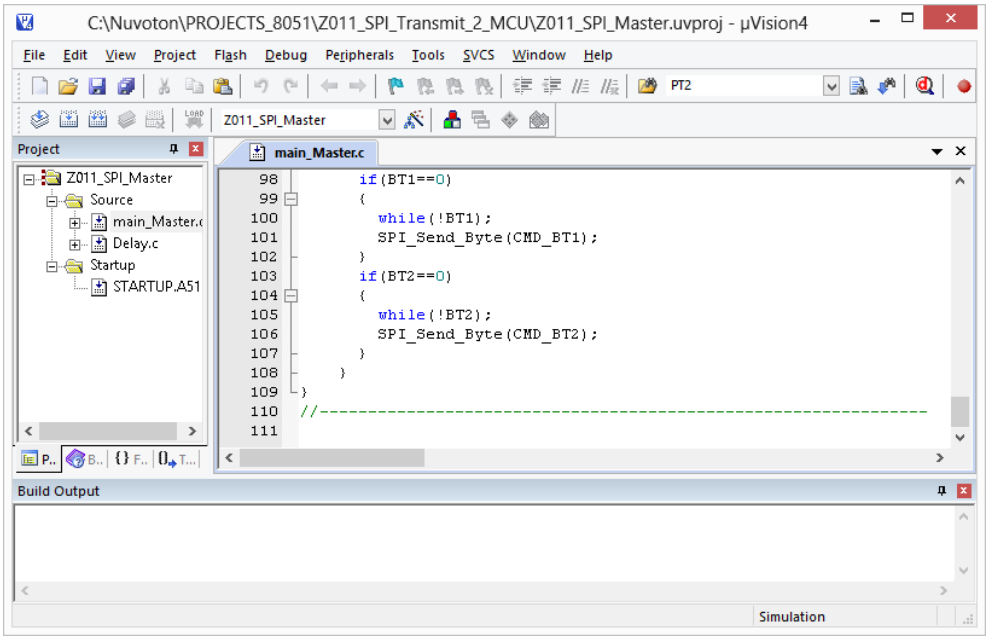
- Thêm các dòng code này vào file ***main_Slave.c***

```
#include <stdio.h>
#include "N78E055A_059A_517A.h"
#include "Typedef.h"
#include "Define.h"
#include "Delay.h"
#include "LCD16x2.h"

#define set_SSOE       SPCR |= SET_BIT7
#define set_SPE        SPCR |= SET_BIT6
#define set_LSBFE      SPCR |= SET_BIT5
#define set_MSTR       SPCR |= SET_BIT4
#define set_CPOL       SPCR |= SET_BIT3
#define set_CPHA       SPCR |= SET_BIT2
#define set_SPR1       SPCR |= SET_BIT1
#define set_SPR0       SPCR |= SET_BIT0

#define set_DRSS       SPSR |= SET_BIT3
#define set_ESPI       EIE  |= SET_BIT0

#define clr_SSOE       SPCR &= CLR_BIT7
#define clr_SPE        SPCR &= CLR_BIT6
#define clr_LSBFE      SPCR &= CLR_BIT5
#define clr_MSTR       SPCR &= CLR_BIT4
#define clr_CPOL       SPCR &= CLR_BIT3
#define clr_CPHA       SPCR &= CLR_BIT2
#define clr_SPR1       SPCR &= CLR_BIT1
#define clr_SPR0       SPCR &= CLR_BIT0

#define clr_DRSS       SPSR &= CLR_BIT3
#define clr_SPIF       SPSR &= CLR_BIT7

#define clr_ESPI       EIE  |= CLR_BIT0

#define SS_PIN       P14
char string1[17] = "   SPI BUS";
char string2[17] = "Wait";
#define CMD_CHECK               0x01
#define CMD_CHECK_BUS           0x02
#define CMD_BT1                 0x03
#define CMD_BT2                 0x04
#define OK                      0x00
#define ERROR                   0xFF
int temp;
//-----------------------------------------------
void SPI_Interrupt_ISR(void) interrupt 8
{
```

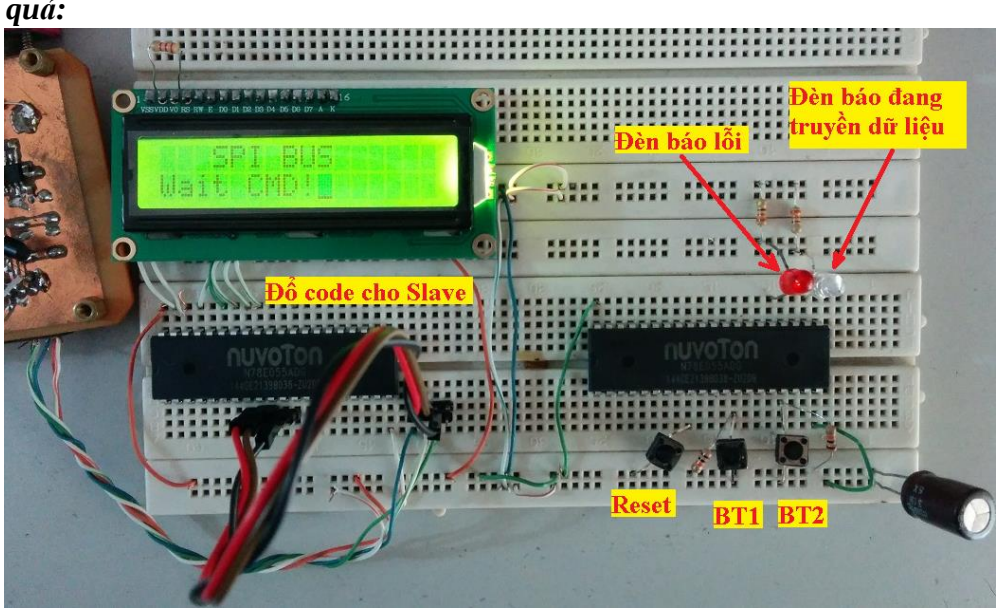
```
    clr_SPIF;
    switch(SPDR)
    {
            case CMD_CHECK_BUS:
                    SPDR = OK;
                    break;
            case CMD_BT1:
                    SPDR = OK;
                    break;
            case CMD_BT2:
                    SPDR = OK;
                    break;
            case CMD_CHECK:
                    SPDR = OK;
                    break;
            default:
                    SPDR = ERROR;
                    break;
    }
}
//-------------------------------------------------------
void Enter_Idle_Mode(void)
{
   PCON |= 0x01;
}
//-------------------------------------------------------
void SPI_Initial(void)
{
   clr_SPR0;                    //SPI clock = Fosc/16
   clr_SPR1;

   clr_MSTR;                    //SPI in Slave mode
   clr_LSBFE;       //MSB first

   clr_CPOL;
   set_CPHA;

   set_ESPI;                    //Enable SPI interrupt
   set_SPE;                     //Enable SPI function
}
//-------------------------------------------------------
void main(void)
{
    LCD_Init();
    LCD_PIN_LIGHT=0;
    LCD_String(string1,1,0);
    SPI_Initial();
    SPDR = ERROR;                   //Cho thanh ghi du lieu SPI= 0xFF
   LCD_String("Wait CMD!",2,0);
   EA = 1;                          //Interrupt global
   clr_SPIF;                        //xoa co ngat SPI
```

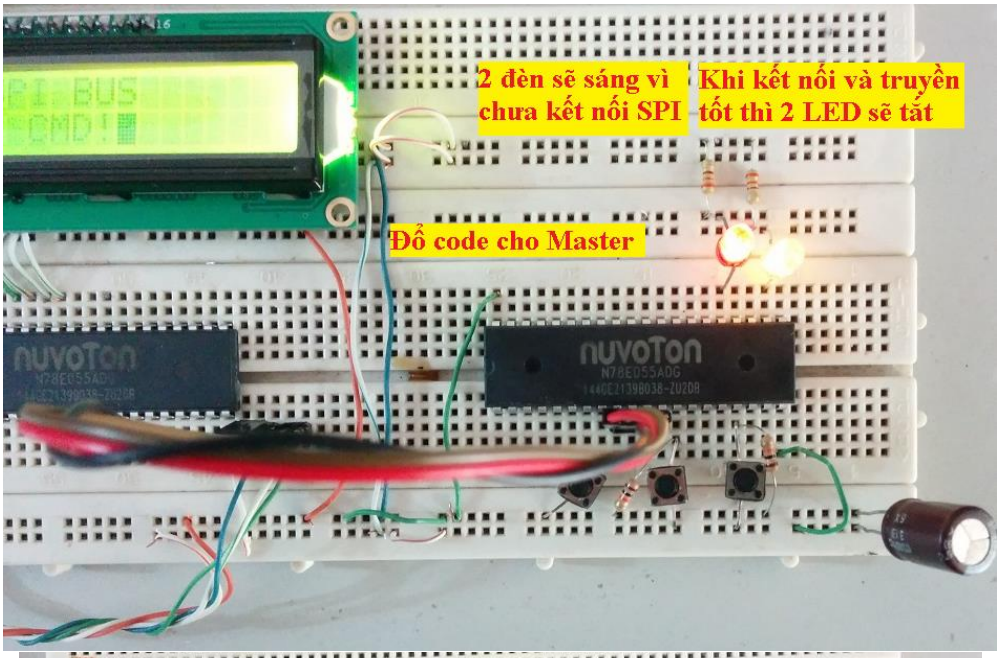
```
    while(1)
   {
            Enter_Idle_Mode();      //MCU vao che do ngu tu day. Doi khi co ngat SPI thi se
thuc thi lenh phia duoi.
            switch(SPDR)
            {
                    case CMD_CHECK_BUS:
                            LCD_String("Check SPI bus!  ",2,0);
                            break;
                    case CMD_BT1:
                            LCD_String("BT1 Pressed!    ",2,0);
                            break;
                    case CMD_BT2:
                            LCD_String("BT2 Pressed!    ",2,0);
                            break;
                    case CMD_CHECK:
                            break;
                    default:
                            LCD_String("Error CMD!      ",2,0);
                            break;
            }
    }
}
//-------------------------------------------------------------
```

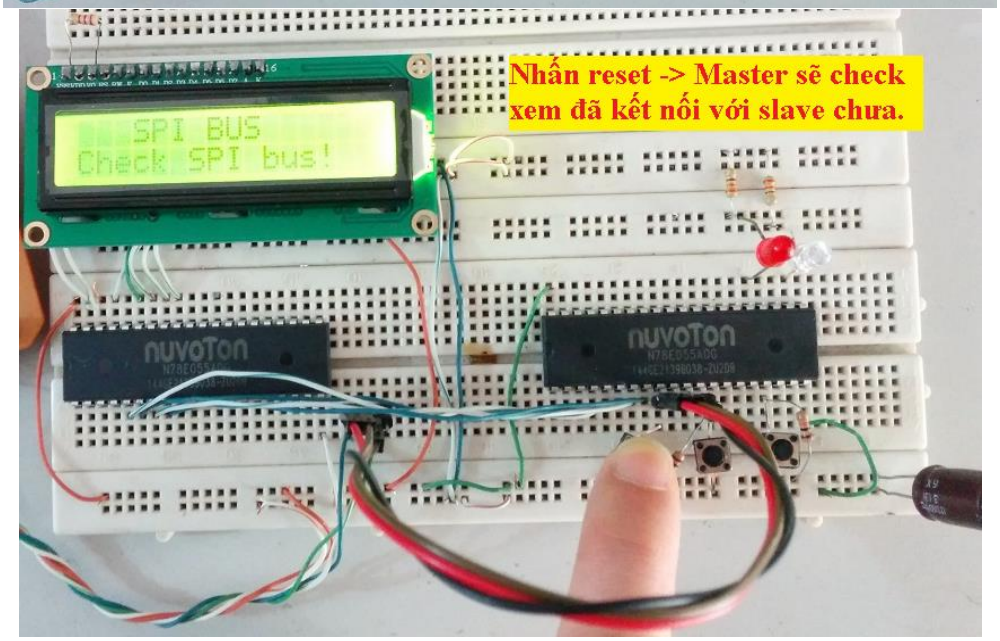


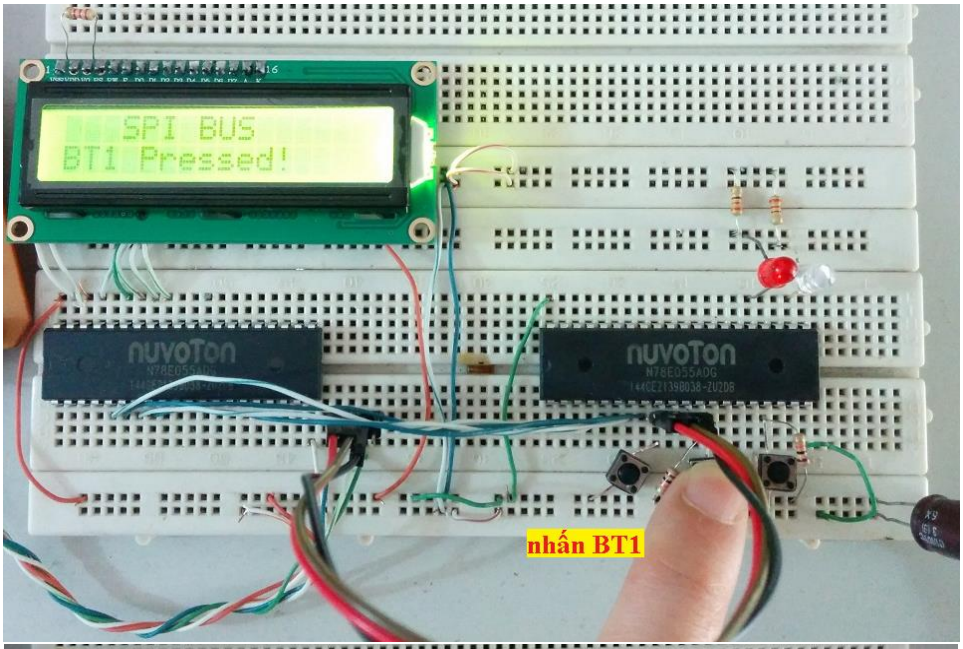

C:\Nuvoton\PROJECTS_8051\Z011_SPI_Transmit_2_MCU\Z011_SPI_Master.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
PT2
Z011_SPI_Master
Project
Z011_SPI_Master
Source
main_Master.c
Delay.c
Startup
STARTUP.A51
main_Master.c
98 if(BT1==0)
99 {
100 while(!BT1);
101 SPI_Send_Byte(CMD_BT1);
102 }
103 if(BT2==0)
104 {
105 while(!BT2);
106 SPI_Send_Byte(CMD_BT2);
107 }
108 }
109 }
110 //--
111
Build Output
Simulation

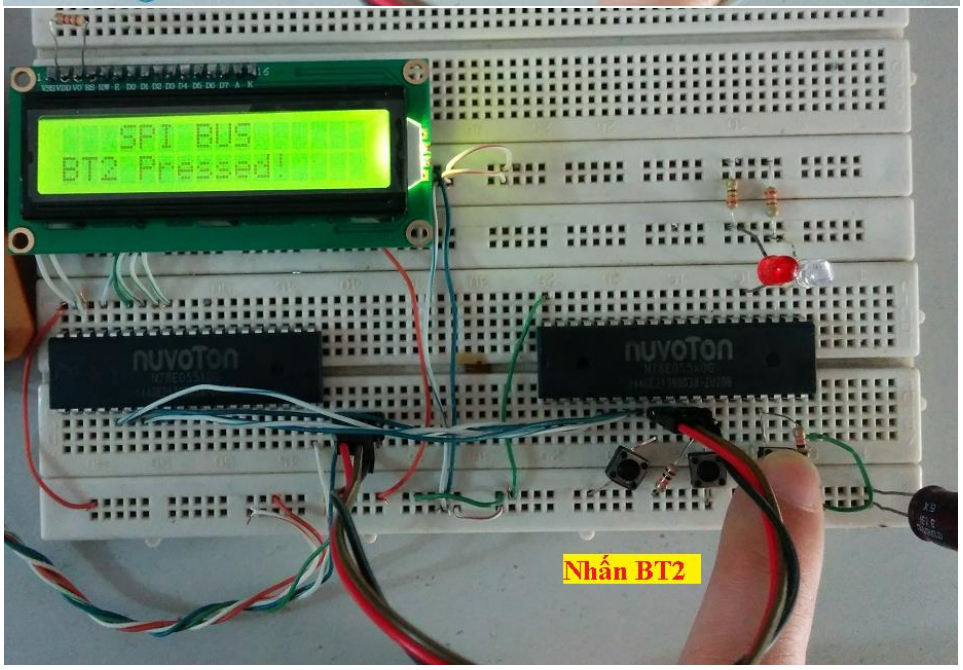
- ***Kết quả:***


SPI BUS
Wait CMD!
Đèn báo đang truyền dữ liệu
Đèn báo lỗi
Đổ code cho Slave
nuvoTon
nuvoTon
Reset
BT1
BT2


2 đèn sẽ sáng vì chưa kết nối SPI
Khi kết nối và truyền tốt thì 2 LED sẽ tắt
Đổ code cho Master


Nhấn reset -> Master sẽ check xem đã kết nối với slave chưa.
SPI BUS
Check SPI bus!


SPI BUS
BT1 Pressed!
nuvoTon
nuvoTon
nhấn BT1


SPI BUS
BT2 Pressed!
nuvoTon
nuvoTon
Nhấn BT2

V. LƯU Ý TÀI LIỆU

Trong quá trình biên soạn tài liệu, khó tránh khỏi các sai sót. Mong sự phản hồi tích cực từ người đọc để xây dựng một nguồn tài liệu mở đáng tin cậy.

Mọi thông tin, xin liên hệ về:

Công ty TNHH Giải pháp TULA - TULA Solution Co., Ltd
Trụ sở chính: Số 173 Tổ 15, Thị trấn Đông Anh, TP. Hà Nội, Việt Nam
Văn phòng giao dịch: Số 59 Tổ 6, Thị trấn Đông Anh, Hà Nội
Chi nhánh: Số 15 Đặng Thuỳ Trâm, Hoàng Quốc Việt, Cầu Giấy, Hà Nội
Tel./ Fax: 04. 39655633, Hotline: 04. 66831176, E-mail:info@tula.vn

LỊCH SỬ SỬA ĐỔI

Phiên bản	Ngày	Trang/Chương	Giải thích
Ver 1.00	08/03/2014	-	Phát hành đầu tiên
Ver 1.01	20/03/2014	93/ V	Chỉnh thông tin

The END