

ĐẠI HỌC CÔNG NGHỆ GIAO THÔNG VẬN TẢI



KỸ THUẬT VI ĐIỀU KHIỂN

(Lưu hành nội bộ)

Chủ biên: ThS. Hoàng Thế Phương

Hà Nội, 2020

MỤC LỤC

CHƯƠNG 1: MỞ ĐẦU VỀ VI ĐIỀU KHIỂN	6
1.1. Giới thiệu về vi điều khiển AVR.....	6
1.2. Vi điều khiển Atmega16	7
1.3. Các công cụ phần cứng.....	9
1.3.1. Mạch nạp:	9
1.3.2. KIT thực hành:	14
1.3.3. Các công cụ phần mềm	15
1.3.3.1. Hướng dẫn sử dụng CodeVisionAVR.....	15
1.3.3.2. Hướng dẫn sử dụng AVR Prog 1.4	25
CHƯƠNG 2: ĐIỀU KHIỂN VÀO/RA DỮ LIỆU	28
2.1. Giới thiệu	28
2.2. Điều khiển vào/ra với AVR.....	29
2.2.1. Các thanh ghi điều khiển vào/ra	29
2.2.2. Khởi tạo cho các cổng vào/ra	31
2.3. Lập trình ứng dụng	32
2.3.1. Giao tiếp nút bấm điều khiển LED đơn	32
2.3.2. Điều khiển LED 7 thanh.....	36
2.3.2.1. Giới thiệu LED 7 thanh	36
2.3.2.2. Điều khiển một LED 7 thanh.....	37
2.3.2.3. Điều khiển nhiều LED 7 thanh	38
2.3.3. Điều khiển hiển thị LCD	41
2.3.3.1. Giới thiệu LCD	41
2.3.3.2. Kết nối LCD với Vi điều khiển.....	43
2.3.3.3. Điều khiển hiển thị LCD	44
2.3.4. Giao tiếp với nút bấm ma trận	47
2.3.4.1. Kết nối nút bấm ma trận với vi điều khiển.....	47

2.3.4.2. Thuật toán lập trình	47
CHƯƠNG 3: HOẠT ĐỘNG CỦA NGẮT NGOÀI.....	50
3.1. Khái niệm.....	50
3.2. Thanh ghi điều khiển ngắt	53
3.3. Lập trình ngắt ngoài.....	55
3.3.1. Khởi tạo	55
3.3.2. Ví dụ:	56
3.4. Bài tập thực hành	57
CHƯƠNG 4: HOẠT ĐỘNG CỦA BỘ CHUYỂN ĐỔI ADC	59
4.1. Chức năng bộ chuyển đổi ADC.....	59
4.1.1. Khái niệm.....	59
4.1.2. Các bước chuyển đổi ADC.....	60
4.2. Lập trình ứng dụng ADC	62
4.2.1. Thiết lập các thanh ghi	62
4.2.2. Lập trình ứng dụng.....	63
4.3. Bài tập thực hành	64
CHƯƠNG 5: HOẠT ĐỘNG CỦA BỘ ĐỊNH THỜI TIMER/COUNTER.....	65
5.1. Hoạt động của bộ định thời.....	65
5.2. Thanh ghi điều khiển định thời.....	66
5.3. Các chế độ hoạt động của bộ định thời.....	69
5.3.1. Chế độ Normal	69
5.3.2. Chế độ CTC	71
5.3.3. Chế độ Fast PWM	72
5.3.4. Chế độ Phase Correct PWM.....	72
5.4. Ví dụ	73

DANH MỤC HÌNH ẢNH

Hình 1. 1. Vi điều khiển ATMEGA16	7
Hình 1. 2. Mạch nạp vi điều khiển	9
Hình 1. 3. Hộp thoại Device manager	10
Hình 1. 4. Cấu hình cổng COM giao tiếp	11
Hình 1. 5. Thay đổi cổng COM giao tiếp	12
Hình 1. 6. Kiểm tra mạch nạp trong Device manager	13
Hình 1. 7. KIT thực hành vi điều khiển AVR	14
Hình 1. 8. Giao diện chương trình CodeVision	15
Hình 1. 9. Tạo project mới trên CodeVision	16
Hình 1. 10. Cửa sổ CodeWizard	17
Hình 1. 11. Lưu các file khởi tạo	18
Hình 1. 12. Giao diện soạn thảo chương trình	18
Hình 1. 13. Cấu hình cho project	19
Hình 1. 14. Cấu hình cho mạch nạp	20
Hình 1. 15. Nạp chương trình vào chip	21
Hình 1. 16. Cấu hình fuse bit	24
Hình 1. 17. Giao diện AVRProg	25
Hình 1. 18. Thay đổi fuse bit của chip	27
Hình 2. 1. Mạch giao tiếp nút bấm điều khiển LED 7 thanh	28
Hình 2. 2. Sơ đồ cấu tạo chân vi điều khiển AVR	29
Hình 2. 3. Thanh ghi DDRx	29
Hình 2. 4. Thanh ghi PORTx	30
Hình 2. 5. Thanh ghi PINx	30
Hình 2. 6. Khởi tạo cổng vào/ra	31
Hình 2. 7. Mạch điều khiển LED đơn	32
Hình 2. 8. Mạch giao tiếp nút bấm	33
Hình 2. 9. Khởi tạo cổng vào/ra	33
Hình 2. 10. Cấu tạo LED 7 thanh Anode chung và Cathode chung	36
Hình 2. 11. Kết nối LED 7 thanh với vi điều khiển	37
Hình 2. 12. Sơ đồ kết nối 4 LED 7 thanh	40
Hình 2. 13. Hình ảnh LCD thực tế	41
Hình 2. 14. Kết nối LCD với vi điều khiển	43
Hình 2. 15. Sơ đồ mạch giao tiếp LCD	44
Hình 2. 16. Khởi tạo giao tiếp LCD bằng CodeWizard	44
Hình 2. 17. Kết nối nút bấm ma trận với vi điều khiển	47
Hình 3. 1. Giản đồ thực thi một chương trình	50
Hình 3. 2. Thanh ghi GICR	53
Hình 3. 3. Thanh ghi MCUCR	53

Hình 3. 4. Thanh ghi MCUSCR.....	54
Hình 3. 5. Thanh ghi GIFR.....	54
Hình 4. 1. Lấy mẫu	60
Hình 4. 2. Khởi tạo ADC	63
Hình 5. 1. Đồng hồ	65
Hình 5. 2. Đếm số lượng xe.....	65
Hình 5. 3. Điều chỉnh độ sáng bóng đèn	65
Hình 5. 4. Thanh ghi TCNT0.....	67
Hình 5. 5. Thanh ghi OCR0	67
Hình 5. 6. Thanh ghi TCCR0.....	67
Hình 5. 7. Thanh ghi TIMSK.....	69
Hình 5. 8. Sơ đồ hoạt động của chế độ Normal	70
Hình 5. 9. Sơ đồ hoạt động của chế độ CTC	71
Hình 5. 10. Sơ đồ hoạt động chế độ Fast PWM.....	72
Hình 5. 11. Sơ đồ hoạt động chế độ Phase Correct PWM	73

DANH MỤC BẢNG BIỂU

Bảng 2. 1. Bảng mã hiển thị LED 7 thanh Anode chung.....	38
Bảng 2. 2. Ý nghĩa các chân của LCD	41

CHƯƠNG 1: MỞ ĐẦU VỀ VI ĐIỀU KHIỂN

1.1. Giới thiệu về vi điều khiển AVR

AVR là họ vi điều khiển 8 bit theo công nghệ mới, với những tính năng rất mạnh được tích hợp trong chip của hãng Atmel theo công nghệ RISC, nó mạnh ngang hàng với các họ vi điều khiển 8 bit khác như PIC, Pisoc. Do ra đời muộn hơn nên họ vi điều khiển AVR có nhiều tính năng mới đáp ứng tối đa nhu cầu của người sử dụng, so với họ 8051 89xx sẽ có độ ổn định, khả năng tích hợp, sự mềm dẻo trong việc lập trình và rất tiện lợi.

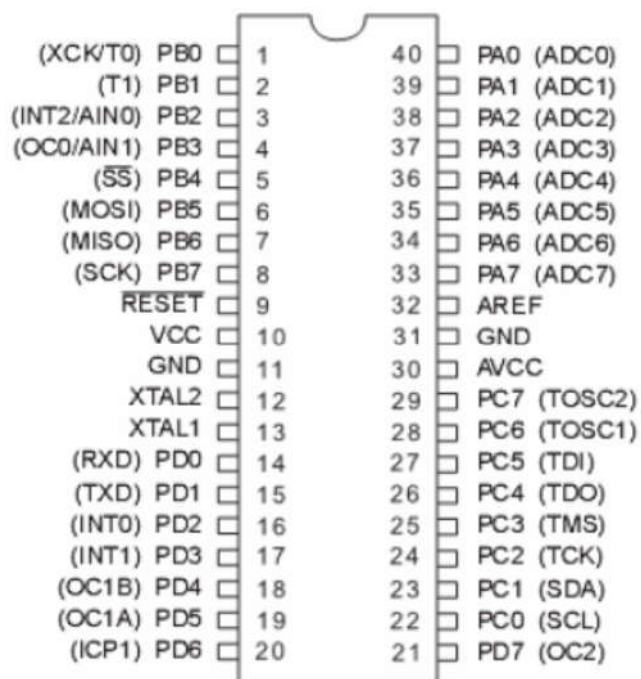
* Tính năng mới của họ AVR:

- Có thể sử dụng xung clock lên đến 16MHz.
- Bộ nhớ Flash có thể lập trình nhiều lần với dung lượng lớn
- Giao diện SPI đồng bộ.
- Các đường dẫn vào/ra (I/O) 2 hướng lập trình được.
- Giao tiếp I2C.
- Bộ biến đổi ADC 10 bit.
- Các kênh băm xung PWM.
- Các chế độ tiết kiệm năng lượng như sleep, stand by..vv.
- Một bộ định thời Watchdog.
- Bộ Timer/Counter 8 bit, 16 bit.
- 1 bộ so sánh analog.
- Bộ nhớ EEPROM.
- Giao tiếp USART..vv..

* Một số chip AVR:

AT90S4414 and AT90S8515
 AT90S4434 and AT90S8535
 AT90C8534
 ATtiny10, ATtiny11 and ATtiny12
 ATtiny15
 ATtiny22
 ATtiny26
 ATtiny28
ATmega8/8515/8535
 ATmega16
 ATmega161
 ATmega162
 ATmega163
 ATmega169
ATmega32
 ATmega323
 ATmega103
ATmega64/128
 AT86RF401.

1.2. Vi điều khiển Atmega16



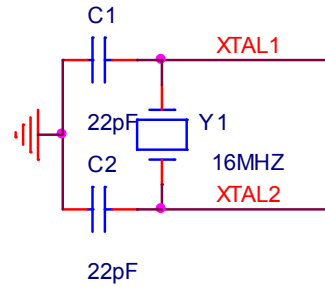
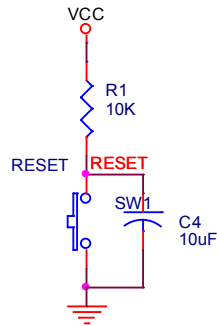
Hình 1. 1. Vi điều khiển ATMEGA16

Atmega16 có đầy đủ tính năng của họ AVR, về giá thành so với các loại khác thì giá thành là vừa phải khi nghiên cứu và làm các công việc ứng dụng tới vi điều khiển. Tính năng:

- Bộ nhớ 16K(flash) . - 512 byte (EEPROM). - 1 K (SRAM).
- Đóng vỏ 40 chân , trong đó có 32 chân vào ra dữ liệu chia làm 4 PORT A,B,C,D. Các chân này đều có chế độ pull_up resistors.
- Giao tiếp SPI.
- Giao diện I2C.
- Có 8 kênh ADC 10 bit.
- 1 bộ so sánh analog.
- 4 kênh PWM.
- 2 bộ timer/counter 8 bit, 1 bộ timer/counter 16 bit.
- 1 bộ định thời Watchdog.
- 1 bộ truyền nhận UART lập trình được.

Mô tả các chân:

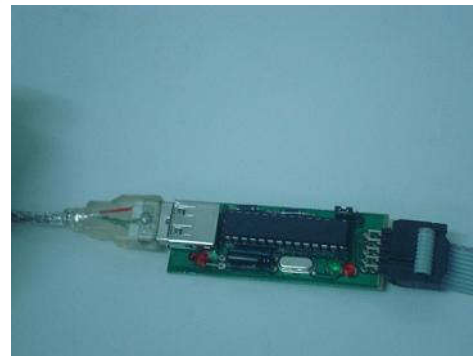
- 4 cổng vào/ra lập trình được, chia làm 4 PORT A,B,C,D.
- Vcc và GND 2 chân cấp nguồn cho vi điều khiển hoạt động. Chú ý cần cấp điện áp 5V ổn định.
- Reset: đây là chân Reset cứng khởi động lại mọi hoạt động của hệ thống.
- 2 chân XTAL1, XTAL2 các chân tạo bộ dao động ngoài cho vi điều khiển, các chân này được nối với thạch anh (hay sử dụng loại 8M), tụ gốm (22p).
- Chân Aref thường nối lên 5v(Vcc), nhưng khi sử dụng bộ ADC thì chân này được sử dụng làm điện áp so sánh, khi đó chân này phải cấp cho nó điện áp cố định, có thể sử dụng diode zener
- Chân Avcc là nguồn nuôi cho bộ ADC, thường được nối lên Vcc.



1.3. Các công cụ phần cứng.

1.3.1. Mạch nạp:

Đây là 1 công cụ không thể thiếu khi thực hành VDK.



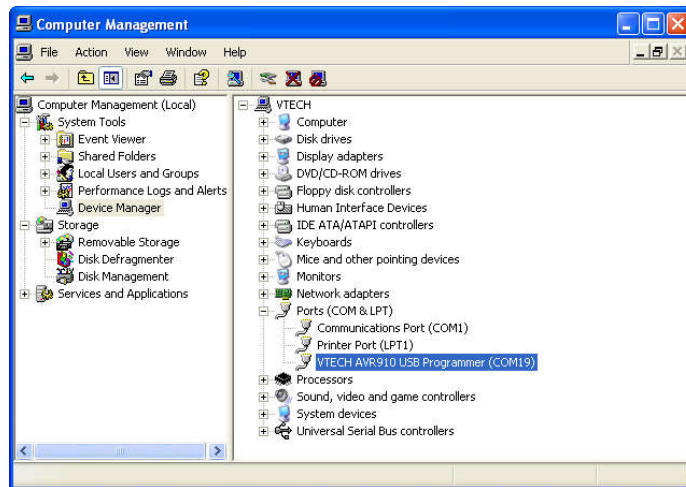
Hình 1. 2. Mạch nạp vi điều khiển

Có 1 số loại mạch nạp phổ biến bao gồm: STK200 (nạp qua cổng máy in), AVR910, STK500 (nạp qua cổng USB).

Cài đặt, cấu hình mạch nạp AVR910

Sau khi cài đặt driver cho mạch nạp xong:

Vào **Control Panel**→**System**→**Advanced**→**Device Manager**, ta có hộp thoại **Device Manager** hiện lên:



Hình 1. 3. Hộp thoại Device manager

+Mở nhánh Ports (COM & LPT), ta sẽ thấy có thông báo mạch nạp AVR910 đang được kết nối với máy tính, cổng COM gán cho mạch là COM19. Ở đây có một số điểm cần lưu ý:

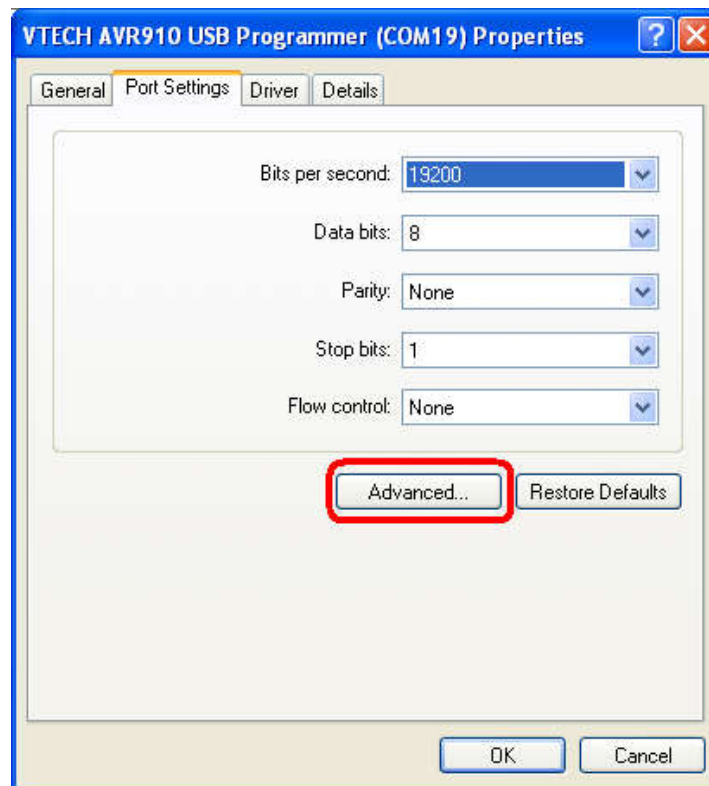
- Cần kết nối mạch nạp với máy tính thì trong phần Ports (COM&LPT) mới hiện lên thông báo về thiết bị.

- Với các máy tính xách tay không có các cổng COM và LPT thực. Phần Ports (COM & LPT) sẽ không hiện lên nếu không có thiết bị nào được gắn vào.

- Tên cổng COM gán cho thiết bị phụ thuộc vào từng máy tính, không phải đều giống nhau, tuy nhiên cần lưu ý rằng **AVR Prog** chỉ hoạt động với các cổng **COM từ 1-4** và **CodeVisionAVR** chỉ hoạt động với các cổng **COM từ 1-6**. Do đó nếu cổng COM được gán tự động ra ngoài khoảng này, ta cần tiến hành gán lại cổng COM bằng tay cho phù hợp!

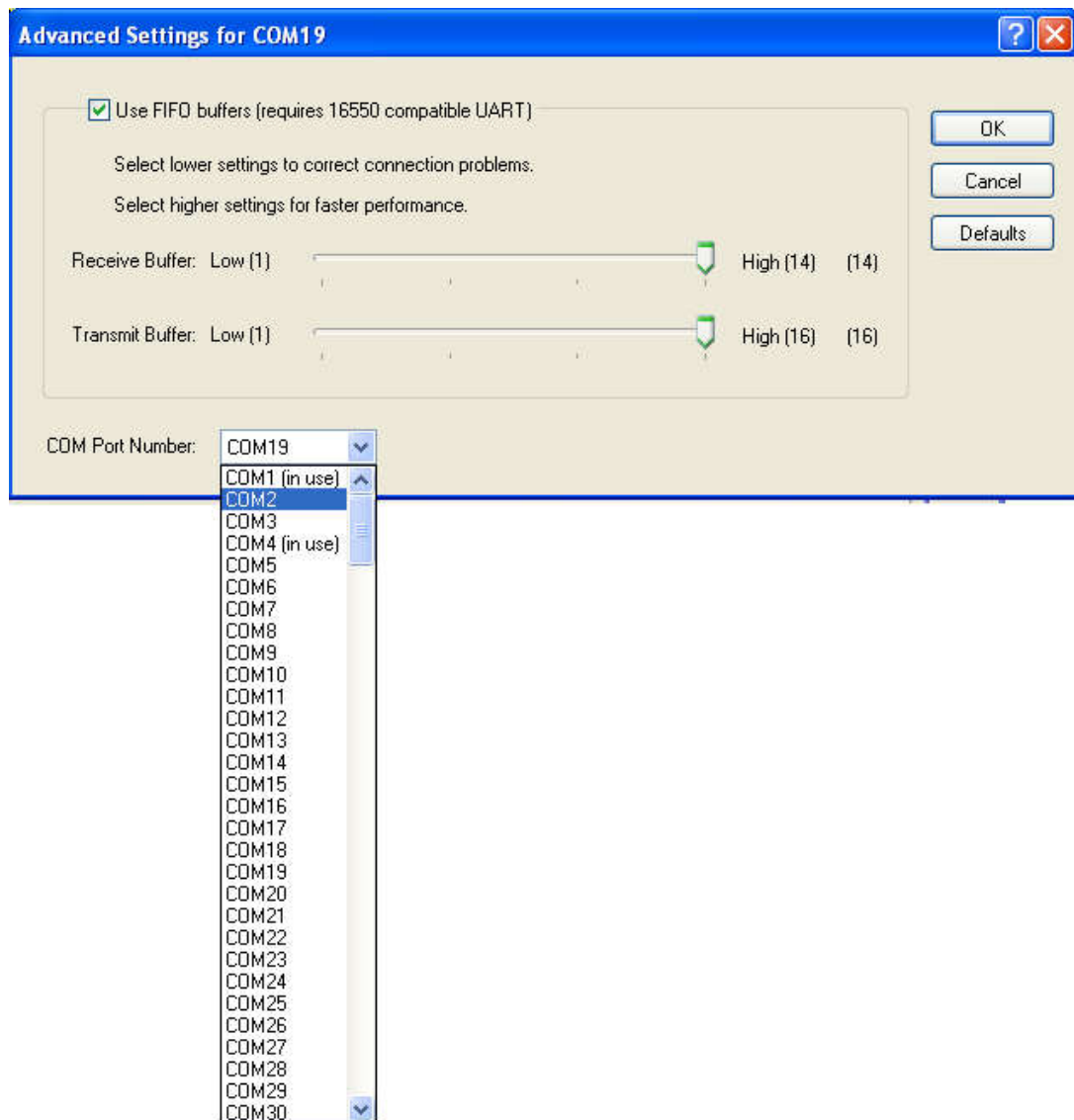
Cách gán lại tên cổng COM cho thiết bị!

-Kích đúp chuột lên nhánh **VTECH AVR910 USB Programmer** để mở hộp thoại **Properties** của thiết bị (hoặc click chuột phải vào và chọn **Properties**:



Hình 1. 4. Cấu hình cổng COM giao tiếp

-Chọn Tab **Port Settings**, nhấn nút **Advanced...** ta sẽ có hộp thoại **Advanced Settings**:



Hình 1. 5. Thay đổi cổng COM giao tiếp

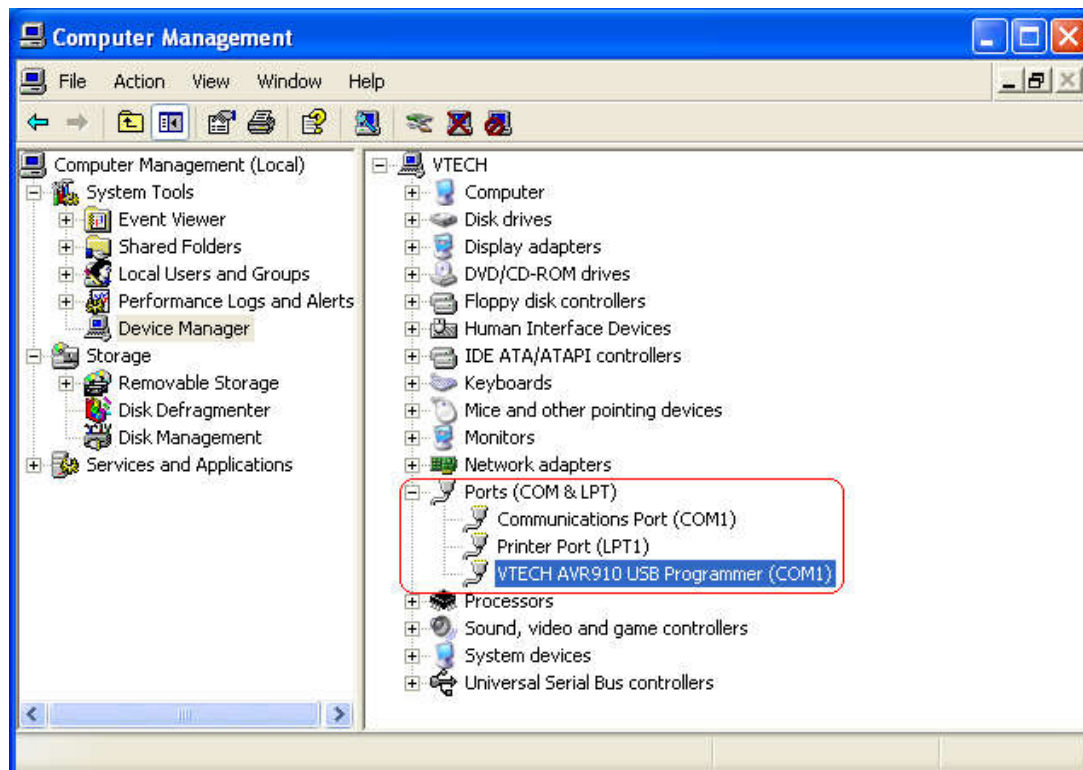
-Trong danh sách COM Port Number, ta chọn một trong các cổng COM 1-4 (để có thể sử dụng với cả 2 phần mềm), tốt nhất là chọn một cổng COM không bị đánh dấu *in use* (đã được sử dụng bởi một phần mềm khác), trong trường hợp tất cả đã bị đánh dấu, ta vẫn có thể lựa chọn cổng *in use*, ở đây mình họa với cổng COM1, chọn **COM1** và nhấn **OK**.



-Ngay lập tức Windows đưa ra thông báo, nhấn **Yes** để chấp nhận dùng chung cổng với thiết bị khác, sau đó nhấn **OK** trên hộp thoại **Properties**.

(Lưu ý rằng trong đa số các trường hợp, việc dùng chung cổng này không ảnh hưởng gì đến hoạt động của thiết bị, tuy nhiên cần lưu ý vì nếu cả 2 thiết bị dùng chung cổng cùng được kết nối thì sẽ xảy ra tranh chấp và cả 2 thiết bị sẽ không hoạt động – trường hợp này có thể xảy ra với các thiết bị Bluetooth hoặc một mạch nạp khác sử dụng COM ảo.)

-Rút mạch nạp khỏi cổng USB và cắm lại, theo dõi trong **Device Manager** sẽ thấy cổng COM mới đã được gán cho thiết bị:



Hình 1. 6. Kiểm tra mạch nạp trong Device manager

+ **Một vài lưu ý nhỏ:**

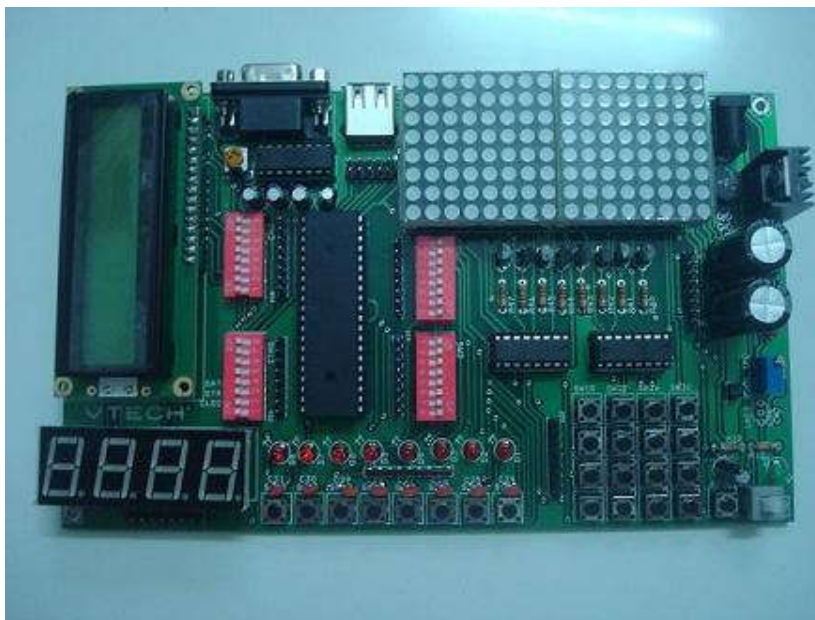
- Quá trình cài đặt trên chỉ có giá trị với một cổng USB (được chọn để kết nối với mạch nạp), máy tính thông thường có khá nhiều cổng USB, nếu cắm mạch nạp vào một cổng khác, quá trình cài đặt sẽ cần làm lại từ đầu, Hệ điều hành sẽ gán một cổng COM mới tương ứng. Vì vậy nên chọn cố định một cổng USB để kết nối với mạch nạp để có thể nắm rõ tên cổng COM tương ứng (với mục đích không cần thay đổi thiết lập khi chạy phần mềm).

- Không nên sử dụng HUB USB với mạch nạp, vì HUB USB có thể không cung cấp đủ nguồn điện cho mạch nạp hoạt động và cung cấp cho mạch được nạp.

- Việc sử dụng dây USB kéo dài có thể ảnh hưởng đến hoạt động của mạch, lưu ý sử dụng các loại dây có chất lượng tốt, có chống nhiễu

1.3.2. KIT thực hành:

Đây là 1 công cụ dùng để thực hành các chức năng của VĐK. Mỗi người học VĐK nên có 1 cái.



Hình 1. 7. KIT thực hành vi điều khiển AVR

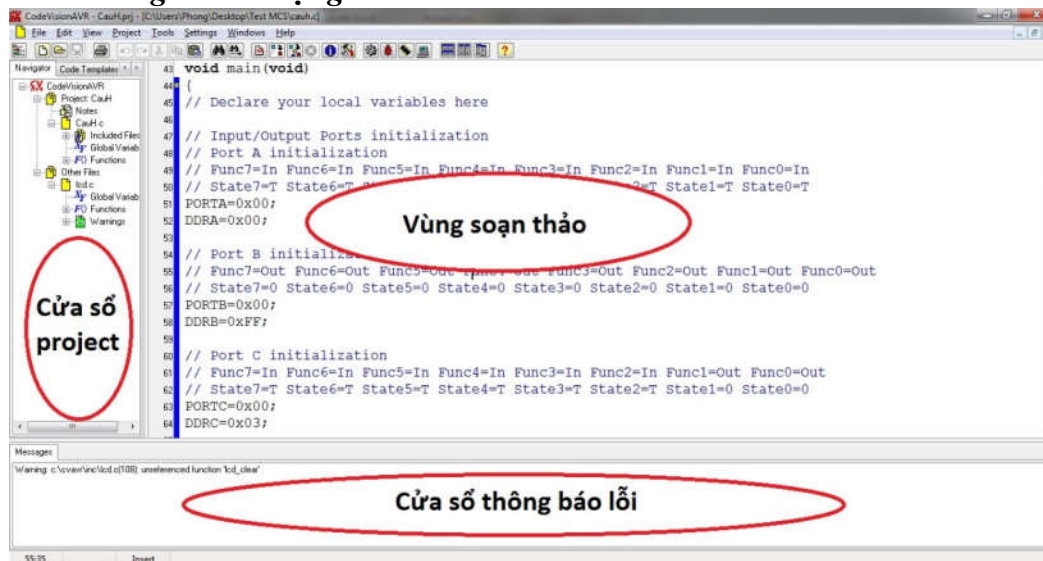
1.3.3. Các công cụ phần mềm

Trình biên dịch: có rất nhiều trình biên dịch bạn có thể sử dụng để biên dịch code của bạn thành file intel hex để nạp vào chip, một số trình dịch quen thuộc có thể kể đến như sau:

- **AvrStudio:** là trình biên dịch ASM chính thức cung cấp bởi Atmel, đây là trình biên dịch hoàn toàn miễn phí và tất nhiên là tốt nhất cho lập trình AVR bằng ASM. Phiên bản hiện tại là 4.12 SP4, bạn có thể download phần mềm AvrStudio tại trang web chính thức của Atmel: http://atmel.com/dyn/products/tools_card.asp?tool_id=2725
- **WinAvr hay Avr gcc:** là bộ chương trình được phát triển bởi gnu, ngôn ngữ sử dụng là C và thường được viết tích hợp với AvrStudio (dùng Avrstudio làm trình biên tập – editor). Đặc biệt bộ biên dịch này cũng miễn phí và đa số nguồn source code C được viết bằng bộ này, vì vậy nó rất lí tưởng cho bạn khi viết các ứng dụng chuyên nghiệp.
- **CodeVisionAvr:** một chương trình bằng ngôn ngữ C rất hay cho AVR, hỗ trợ nhiều thư viện lập trình. Tuy nhiên là chương trình thương mại. Bạn có thể download bản 1.25.9 tại trang web của công ty: www.vtechco.org

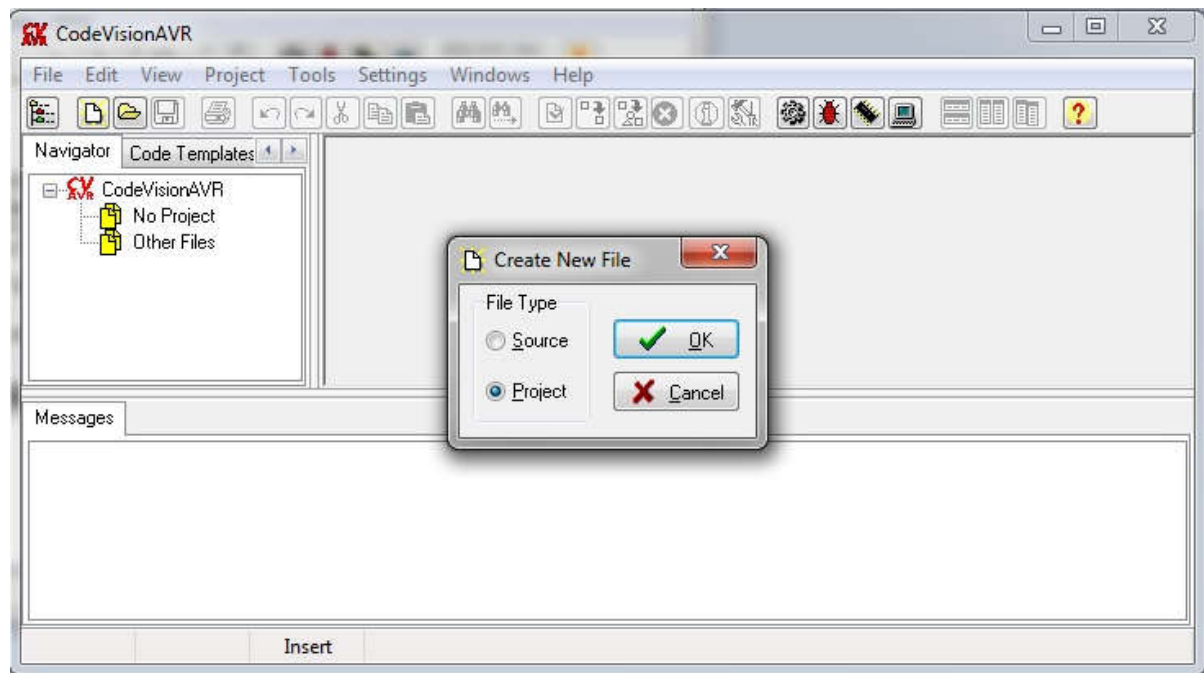
Chương trình nạp (Chip Programmer): đa số các trình biên dịch (AvrStudio, CodeVisionAVR, Bascom...) đều tích hợp sẵn 1 chương trình nạp chip hỗ trợ nhiều loại mạch nạp nên bạn không quá lo lắng. Trong trường hợp khác, bạn có thể sử dụng các chương trình nạp như Icprow hay Ponyprog... là các chương trình nạp miễn phí cho AVR. Việc chọn và sử dụng chương trình nạp sẽ được giới thiệu trong các bài sau.

1.3.3.1. Hướng dẫn sử dụng CodeVisionAVR



Hình 1. 8. Giao diện chương trình CodeVision

Để tạo Project mới chọn trên menu: **File → New** được như sau:

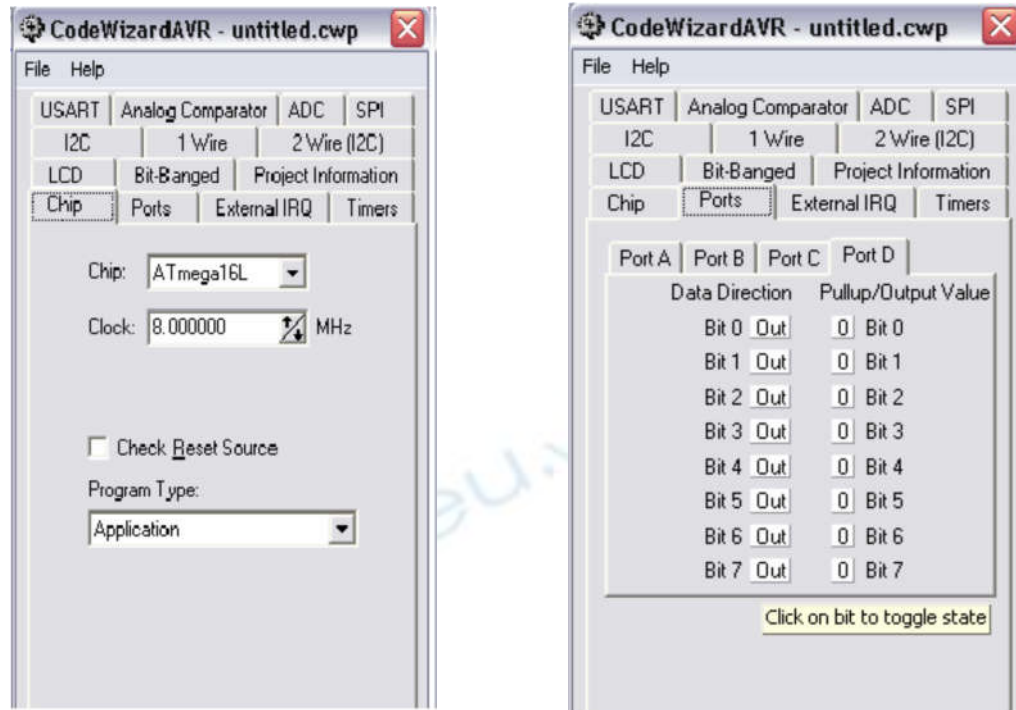


Hình 1. 9. Tạo project mới trên CodeVision

Chọn Project sau đó click chuột vào OK được cửa sổ hỏi xem có sử dụng CodeWinzard không.



Chọn Yes được cửa sổ CodeWinzardAVR như sau:

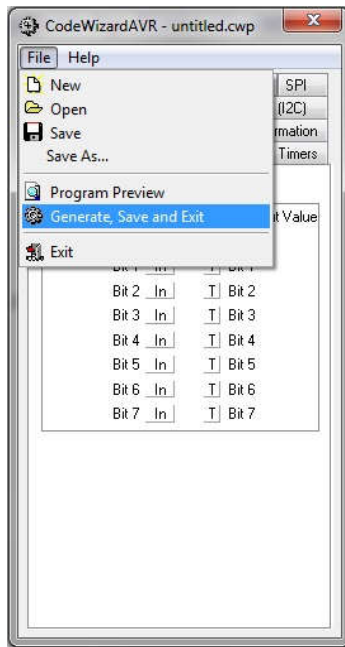


Hình 1. 10. Cửa sổ CodeWizard

Sử dụng chip AVR nào và thạch anh tần số bao nhiêu ta nhập vào tab Chip.

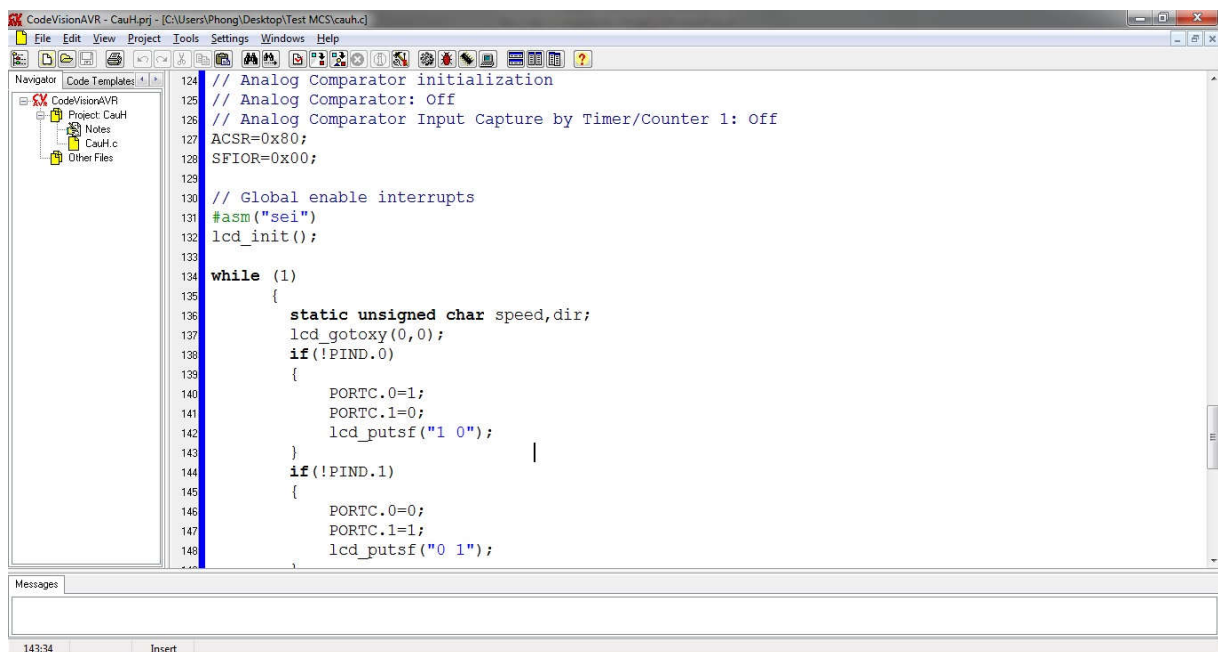
Để khởi tạo cho các cổng IO ta chuyển qua tab **Ports**.

Sau đó chọn **File→Generate, Save and Exit**, lưu lại 3 file có đuôi là .C, .prj, .cwp



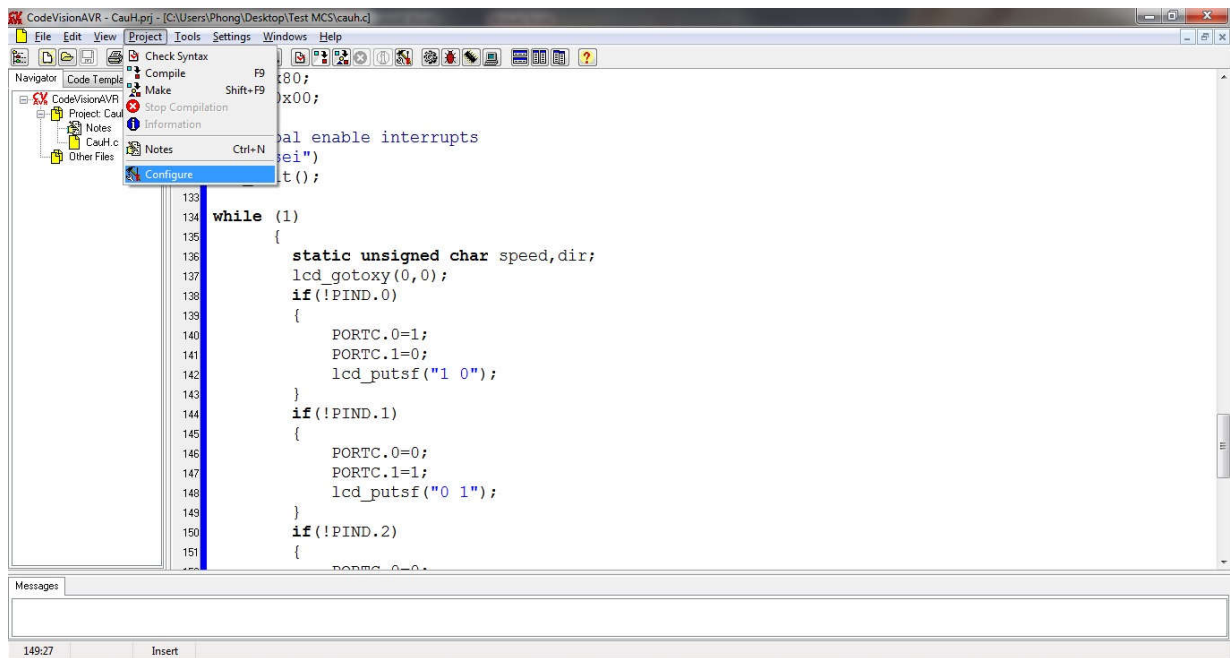
Hình 1. 11. Lưu các file khởi tạo

CodeWizard đã khởi tạo sẵn cho ta giá trị của các thanh ghi theo như khởi tạo bên trên, ta chỉ việc tiếp tục lập trình vào vùng soạn thảo.



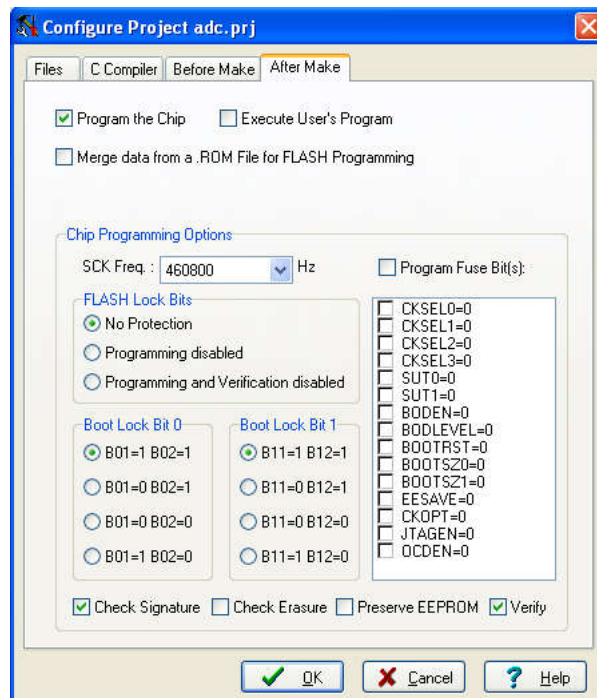
Hình 1. 12. Giao diện soạn thảo chương trình

Sau khi lập trình xong, bây giờ ta cần nạp chương trình vào VDK. Đầu tiên ta vào **Project→Configure**:

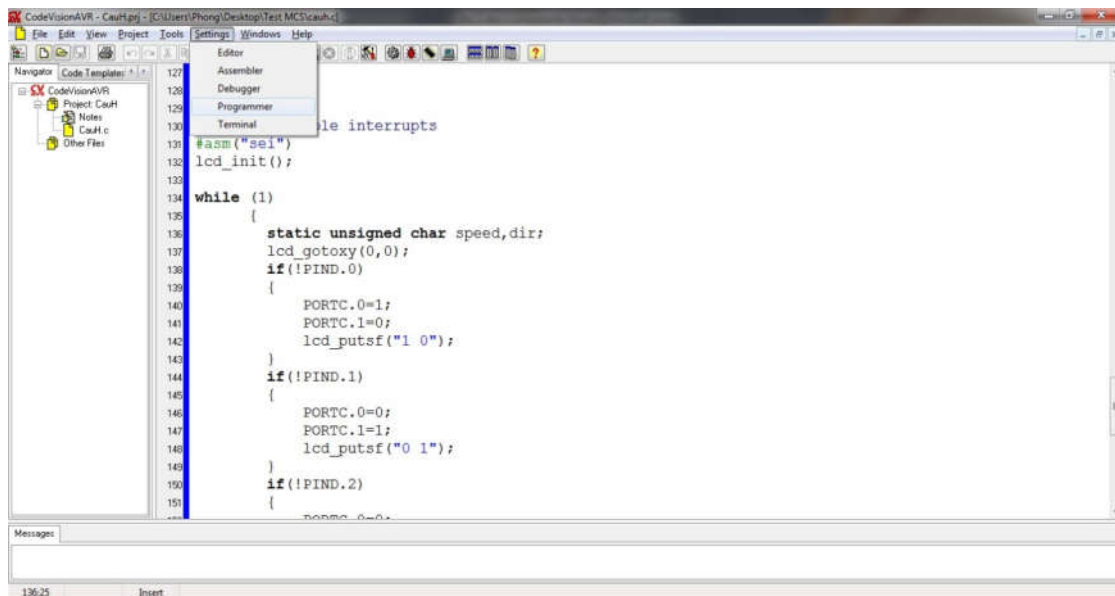


Hình 1. 13. Cấu hình cho project

Sang thẻ **Apter Make**, và tích vào ô **Program the chip**:



Sau đó, vào **Settings**→**Programmer** để cấu hình cho mạch nạp:



Hình 1. 14. Cấu hình cho mạch nạp

Một hộp thoại hiện ra, dòng đầu tiên ta chọn tên mạch nạp (mặc định là STK500), dòng thứ 2 chọn cổng giao tiếp với máy tính:

