

# Tổng quan về VĐK AVR

# Sơ lược về kiến trúc VĐK

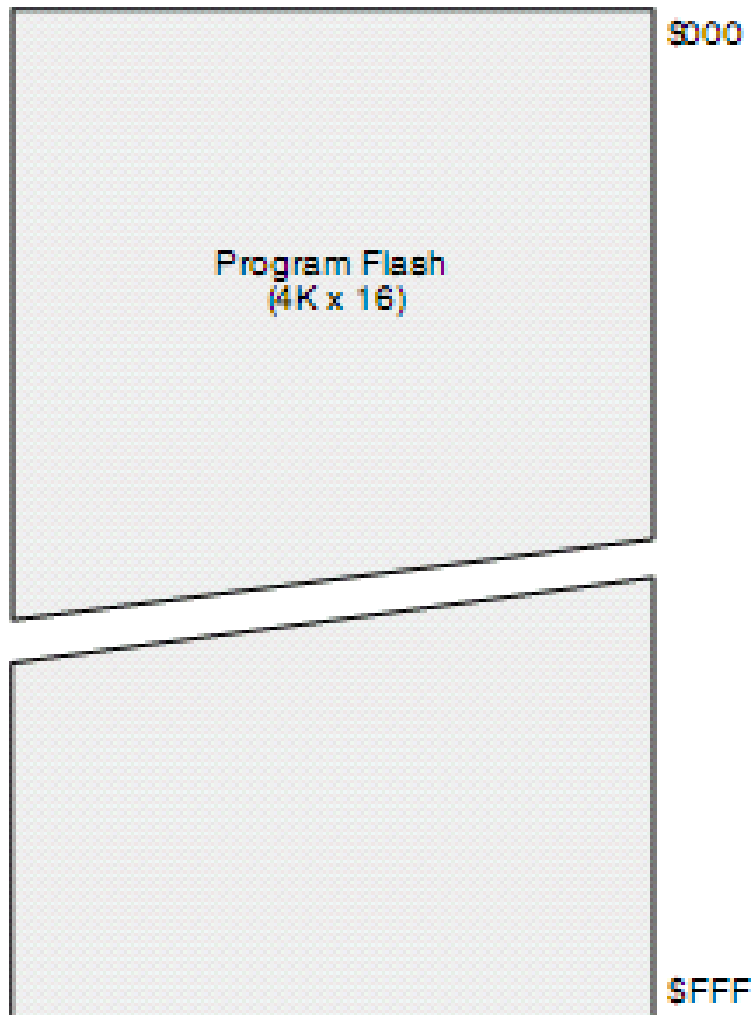
- AVR do hãng Atmel chế tạo ra, là họ VĐK 8bit theo công nghệ mới với những tính năng rất mạnh được tích hợp trong chip.
- AVR Ổn định hơn rất nhiều so với dòng VĐK 8051
- Tính năng mới của họ AVR :
  - Giao diện SPI đồng bộ, giao tiếp I2C, USART.
  - Các lối vào ra lập trình được.
  - Tích hợp bộ biến đổi ADC 10bit.
  - Có sẵn các kênh băm xung PWM.
  - Bộ Timer/Counter 8bit và 16bit rất mạnh.
  - Bộ định thời Watchdog.

Bộ nhớ EEPROM

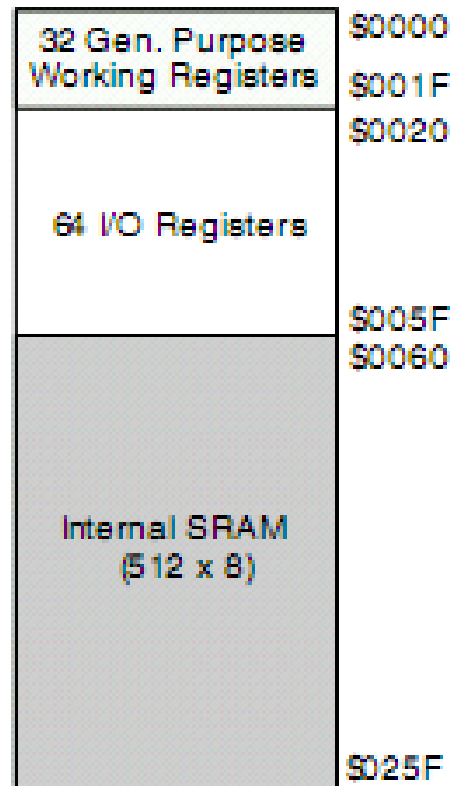
# Cấu trúc bộ nhớ

Cũng như mọi vi điều khiển khác AVR có cấu trúc Harvard tức là có bộ nhớ và đường bus riêng cho bộ nhớ chương trình và bộ nhớ dữ liệu.

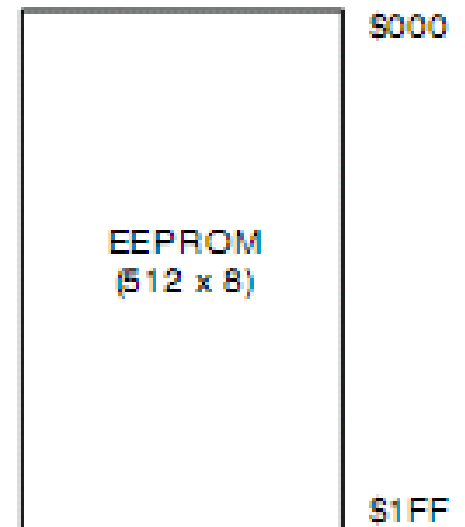
## Program Memory



## Data Memory



## Data Memory





Ta thấy không gian bộ nhớ của bộ nhớ chương trình gồm 4Kx8 và có địa chỉ từ 0000H tới FFFH. Bộ nhớ dữ liệu gồm hai phần: bộ nhớ RAM và bộ nhớ EEPROM trong đó không gian bộ nhớ RAM lại chia làm 3 phần: Các thanh ghi chức năng chung, các thanh ghi vào ra và cuối cùng là 512 byte bộ nhớ SRAM. Bộ nhớ EEPROM mặc dù cũng là một phần của bộ nhớ dữ liệu nhưng lại hoàn toàn đứng độc lập như một bộ nhớ độc lập và cũng được đánh địa chỉ riêng.

# Bộ nhớ dữ liệu

- AVR có 32 thanh ghi chức năng chung và chúng được liên kết trực tiếp với ALU đây là điểm khác biệt của AVR và tạo cho nó một tốc độ xử lý cực cao. Các thanh ghi được đặt tên từ R0 tới R31. Và đặc biệt cặp 6 thanh ghi cuối (từ R6 tới R31) từng đôi một tạo thành các thanh ghi 16 bit sử dụng làm con trỏ trỏ tới bộ nhớ chương trình và dữ liệu. Chúng lần lượt có tên là X, Y, Z.

7

0

Addr.

R0	\$00
R1	\$01
R2	\$02
...	
R13	\$0D
R14	\$0E
R15	\$0F
R16	\$10
R17	\$11
...	
R26	\$1A
R27	\$1B
R28	\$1C
R29	\$1D
R30	\$1E
R31	\$1F



# Bộ nhớ chương trình

- Bộ nhớ chương trình có địa chỉ từ 0000H tới 0010H được dành cho bảng véc tơ ngắt.

**Table 2.** Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	\$000	RESET	Hardware Pin, Power-on Reset and Watchdog Reset
2	\$001	INT0	External Interrupt Request 0
3	\$002	INT1	External Interrupt Request 1
4	\$003	TIMER2 COMP	Timer/Counter2 Compare Match
5	\$004	TIMER2 OVF	Timer/Counter2 Overflow
6	\$005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	\$006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	\$007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	\$008	TIMER1 OVF	Timer/Counter1 Overflow
10	\$009	TIMER0 OVF	Timer/Counter0 Overflow
11	\$00A	SPI, STC	SPI Serial Transfer Complete
12	\$00B	UART, RX	UART, Rx Complete



**Table 2.** Reset and Interrupt Vectors (Continued)

Vector No.	Program Address	Source	Interrupt Definition
13	\$00C	UART, UDRE	UART Data Register Empty
14	\$00D	UART, TX	UART, Tx Complete
15	\$00E	ADC	ADC Conversion Complete
16	\$00F	EE_RDY	EEPROM Ready
17	\$010	ANA_COMP	Analog Comparator

# Các dòng vĐK AVR

- Atmega 88.
- Atmega 16.
- Atmega 128.
- Atmega8.
- Attiny 2113.
- ...v...v...vv

# Giới thiệu VĐK Atmega 16

## Pinout ATmega16

### PDIP

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
( $\overline{SS}$ ) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
$\overline{RESET}$	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)



- Atmega 16 có đầy đủ tính năng của họ AVR, có giá thành phù hợp khi nghiên cứu và ứng dụng.
- Tính năng:
  - Bộ nhớ 16K(Flash) – 512 byte EEPROM – 1 K.
  - Đóng vỏ 40 chân, trong đó có 32 chân vào ra dữ liệu.
  - Giao tiếp SPI, I2C, USART.
  - 8 kênh ADC 10bit, 1 bộ so sánh analog.
  - 4 kênh PWM.
  - 2 bộ Timer/Counter 8bit, 1 bộ Timer/Counter 16bit.
  - 1 bộ định thời Watchdog.

# Mô tả các chân

- Vcc và Gnd là 2 chân cấp nguồn cho VĐK.
- Reset là chân khởi động lại mọi hoạt động của hệ thống.
- XTAL1 và XTAL2 là các chân tạo bộ dao động ngoài cho VĐK.
- Chân Vref thường nối lên 5V, nhưng khi sử dụng ADC thì chân này được sử dụng làm điện áp so sánh. Khi đó chân này phải được cấp 1 điện áp cố định.
- Các chân AVCC thường được nối lên VCC nhưng khi sử dụng ADC thì các chân này được nối qua 1 cuộn cảm lên VCC với mục đích ổn định điện áp cho bộ biến đổi.
- Các PORTA, PORTB, PORTC, PORTD đều 8 bit có trở kéo lên trong vừa có thể làm đầu ra, vừa làm đầu vào.

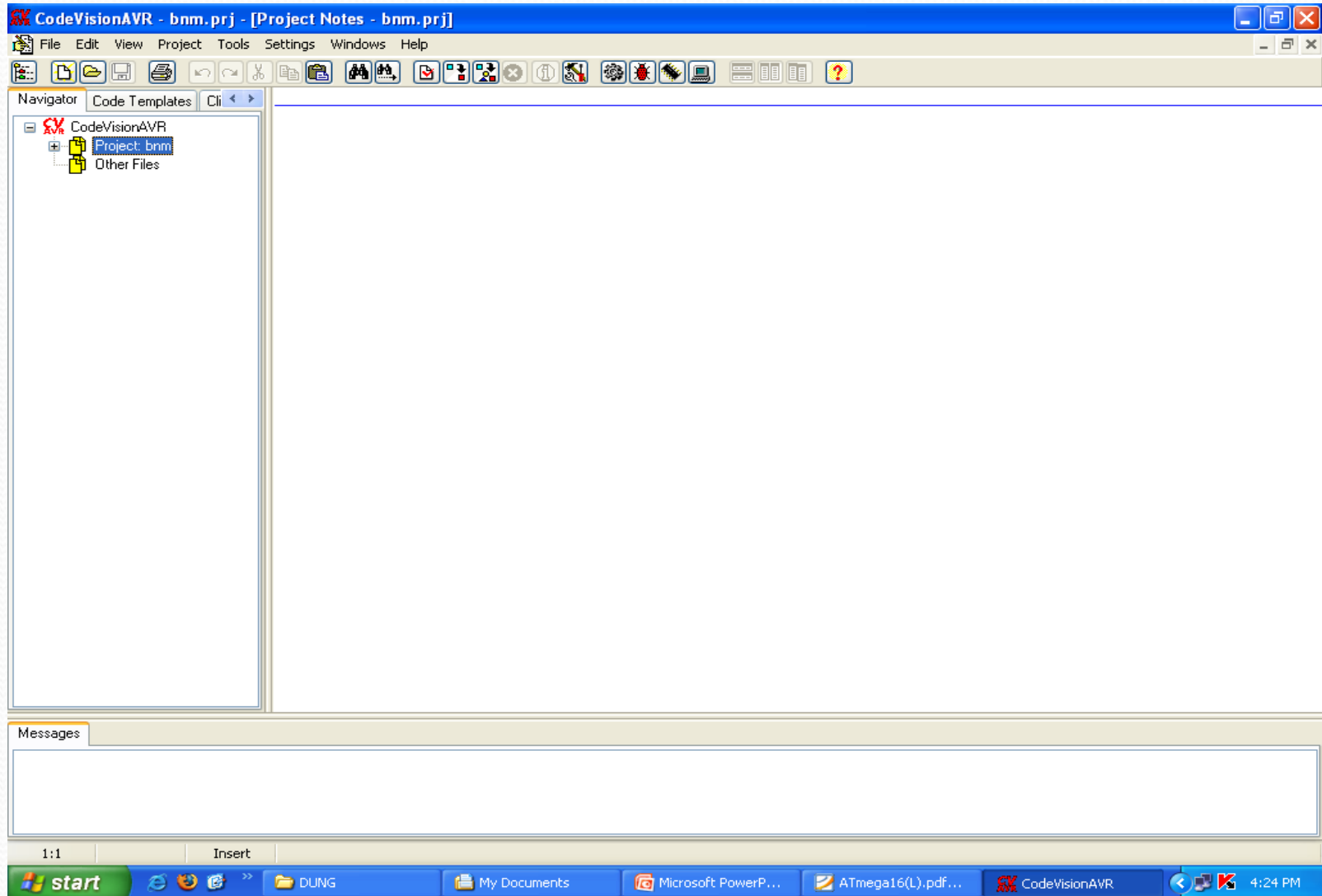


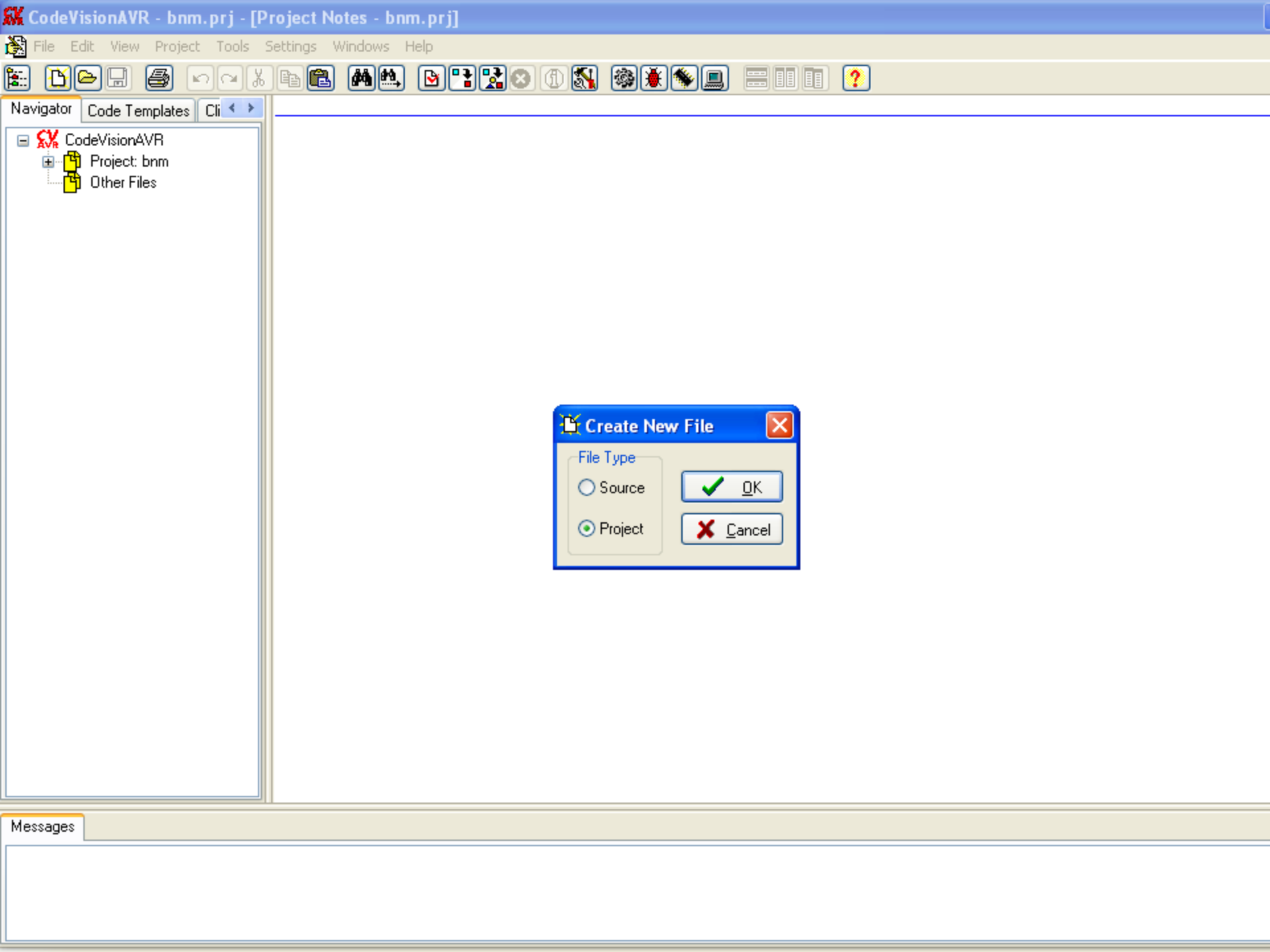
# Lập trình cổng vào ra dùng CodeVision

- Bài này ta sử dụng phần mềm thông dụng CodeVision để lập trình cổng vào ra cho Atmega16 bằng cách tác động vào thanh ghi PORTxx và DDRxx.
- DDRxx : để điều khiển các hướng dữ liệu các chân của cổng. Khi DDRxx=0 thì dùng làm cổng vào, ngược lại, khi DDRxx=1 thì dùng làm cổng ra.
- PORTxx: truy cập tại các địa chỉ xuất nhập của PORTx



Chạy chương trình CodeVision. Chọn File->New->Project->OK





Chọn YES khi được hỏi có sử dụng Code Wizard.





## Cửa sổ Code Vision Winzard:

**CodeWizardAVR - untitled.cwp**

File Help

USART Analog Comparator ADC SPI  
I2C 1 Wire 2 Wire (I2C)  
LCD Bit-Banged Project Information  
Chip Ports External IRQ Timers

Chip: ATmega16L

Clock: 8.000000 MHz

☐ Check Reset Source

Program Type:  
Application

**CodeWizardAVR - untitled.cwp**

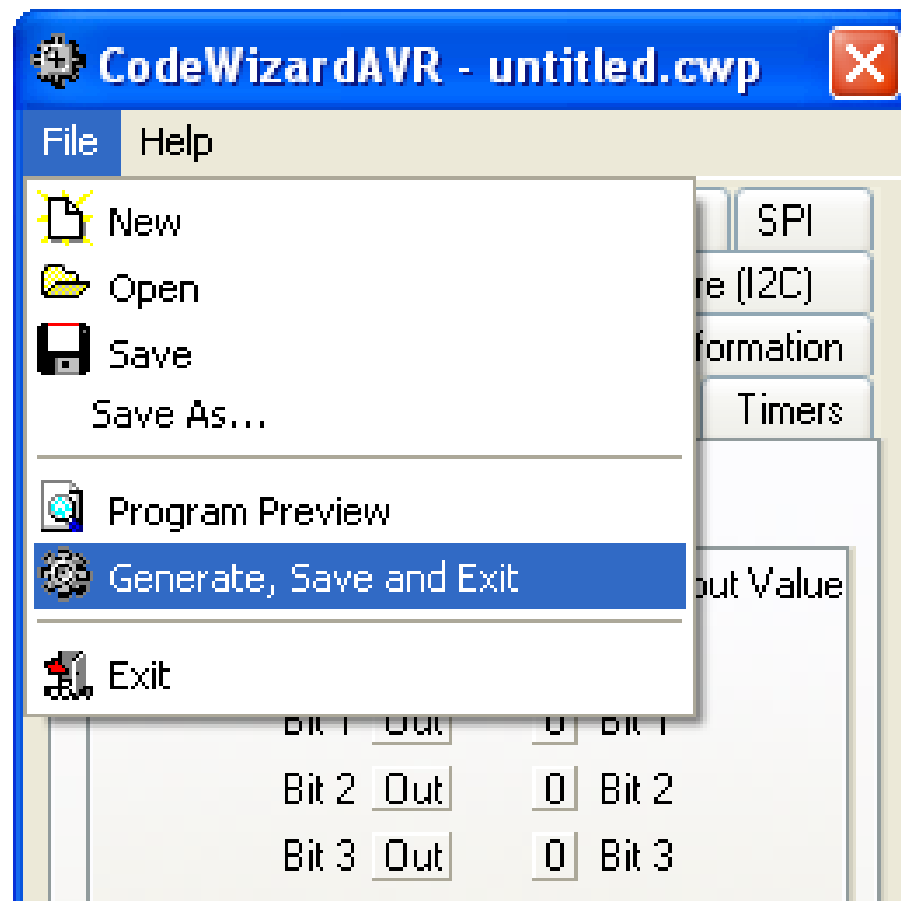
File Help

USART Analog Comparator ADC SPI  
I2C 1 Wire 2 Wire (I2C)  
LCD Bit-Banged Project Information  
Chip Ports External IRQ Timers

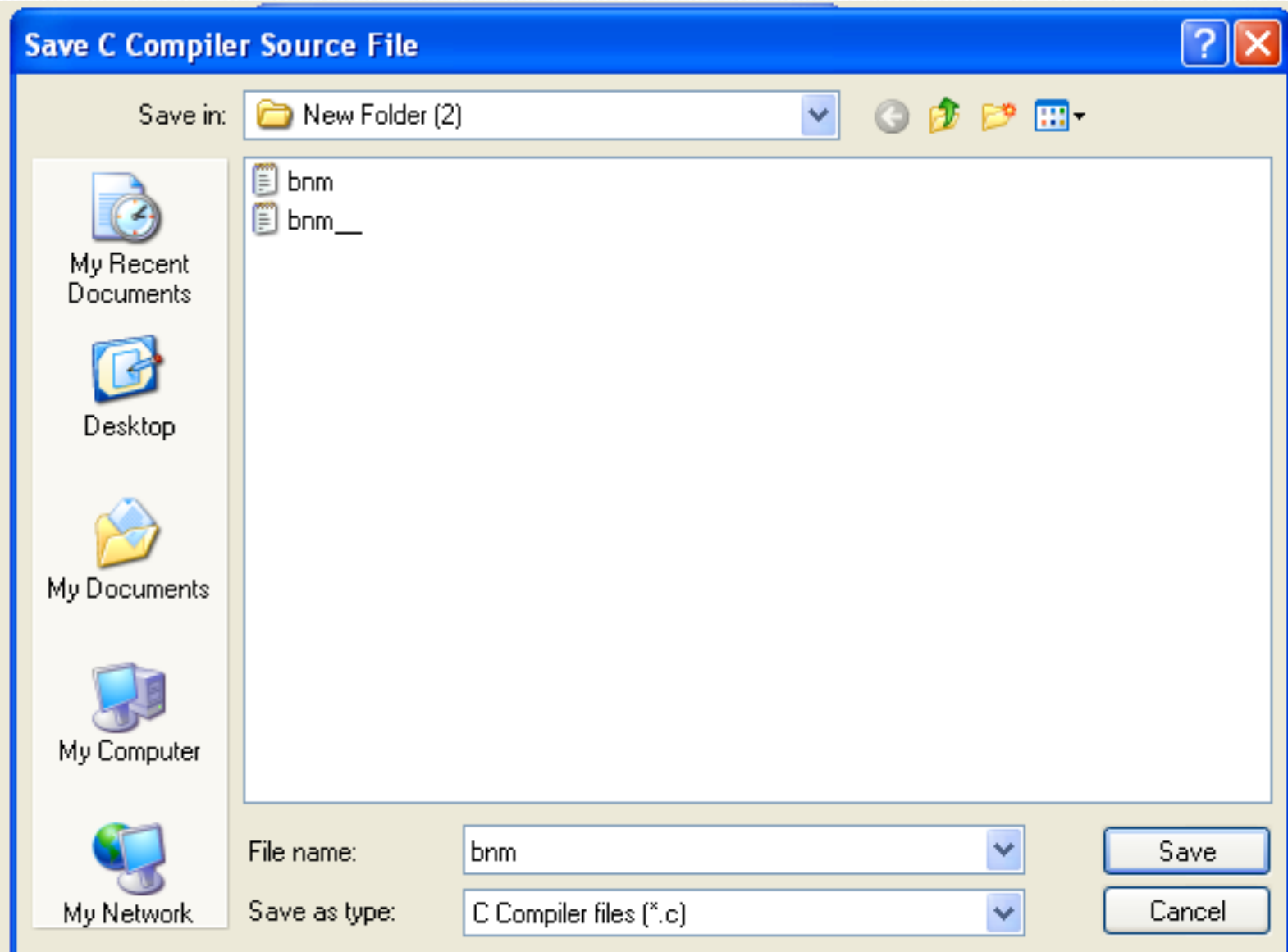
Port A Port B Port C Port D

Data Direction		Pullup/Output Value	
Bit 0	Out	0	Bit 0
Bit 1	Out	0	Bit 1
Bit 2	Out	0	Bit 2
Bit 3	Out	0	Bit 3
Bit 4	Out	0	Bit 4
Bit 5	Out	0	Bit 5
Bit 6	Out	0	Bit 6
Bit 7	Out	0	Bit 7

Chọn File->Generate, Save and Exit.

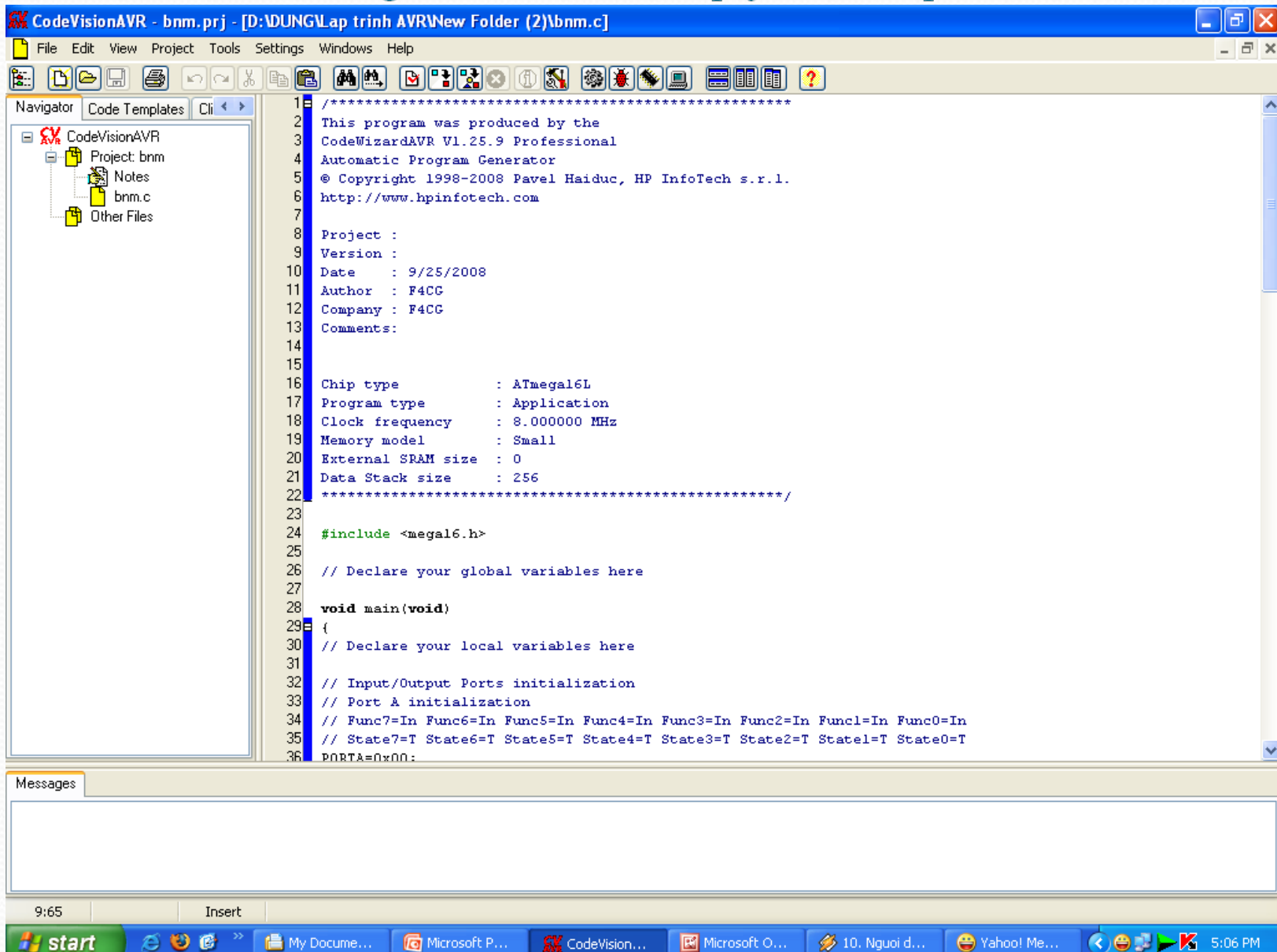


Được cửa sổ yêu cầu nhớ các file của Project. Ta đặt tên và Save lại

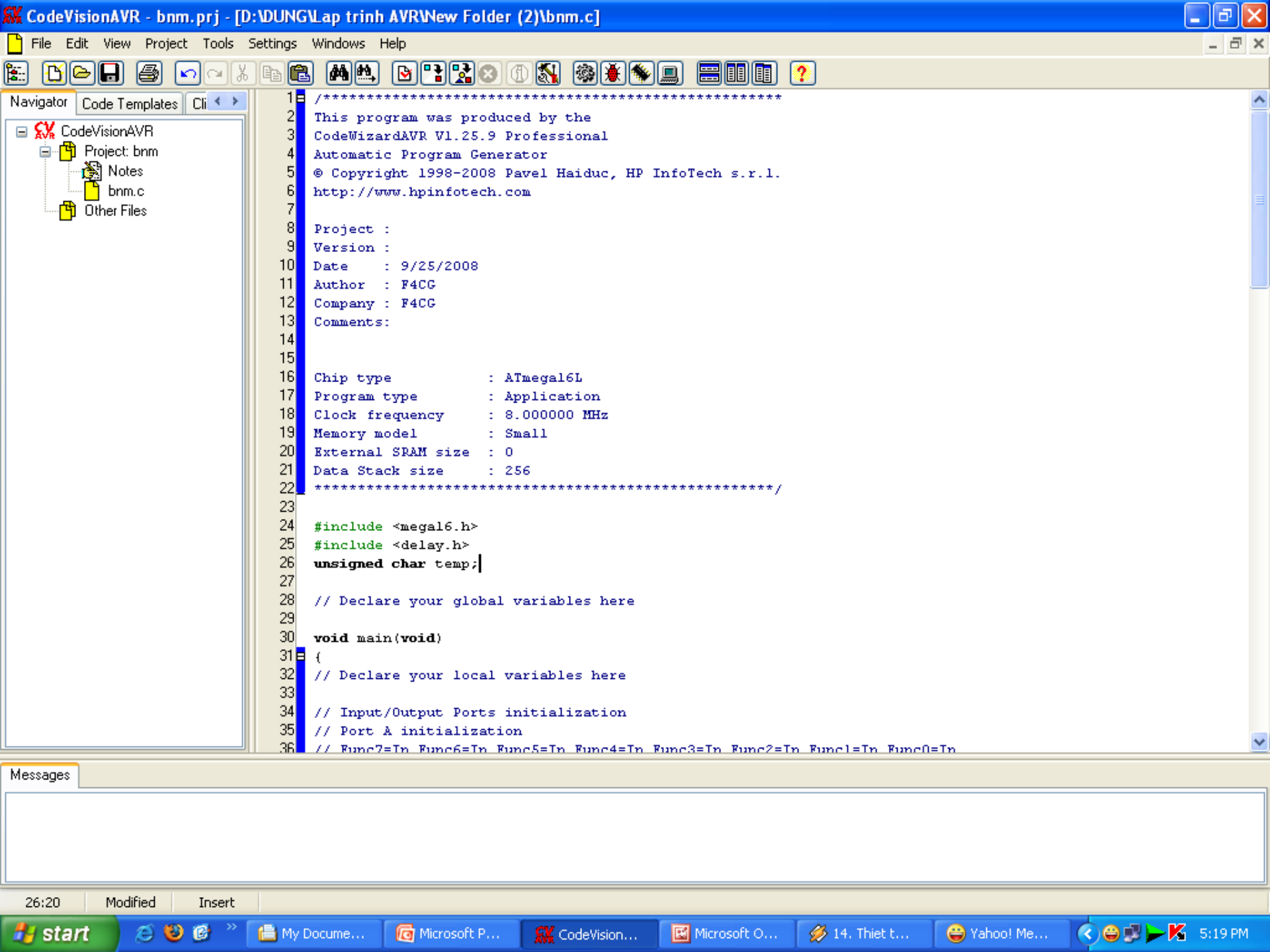
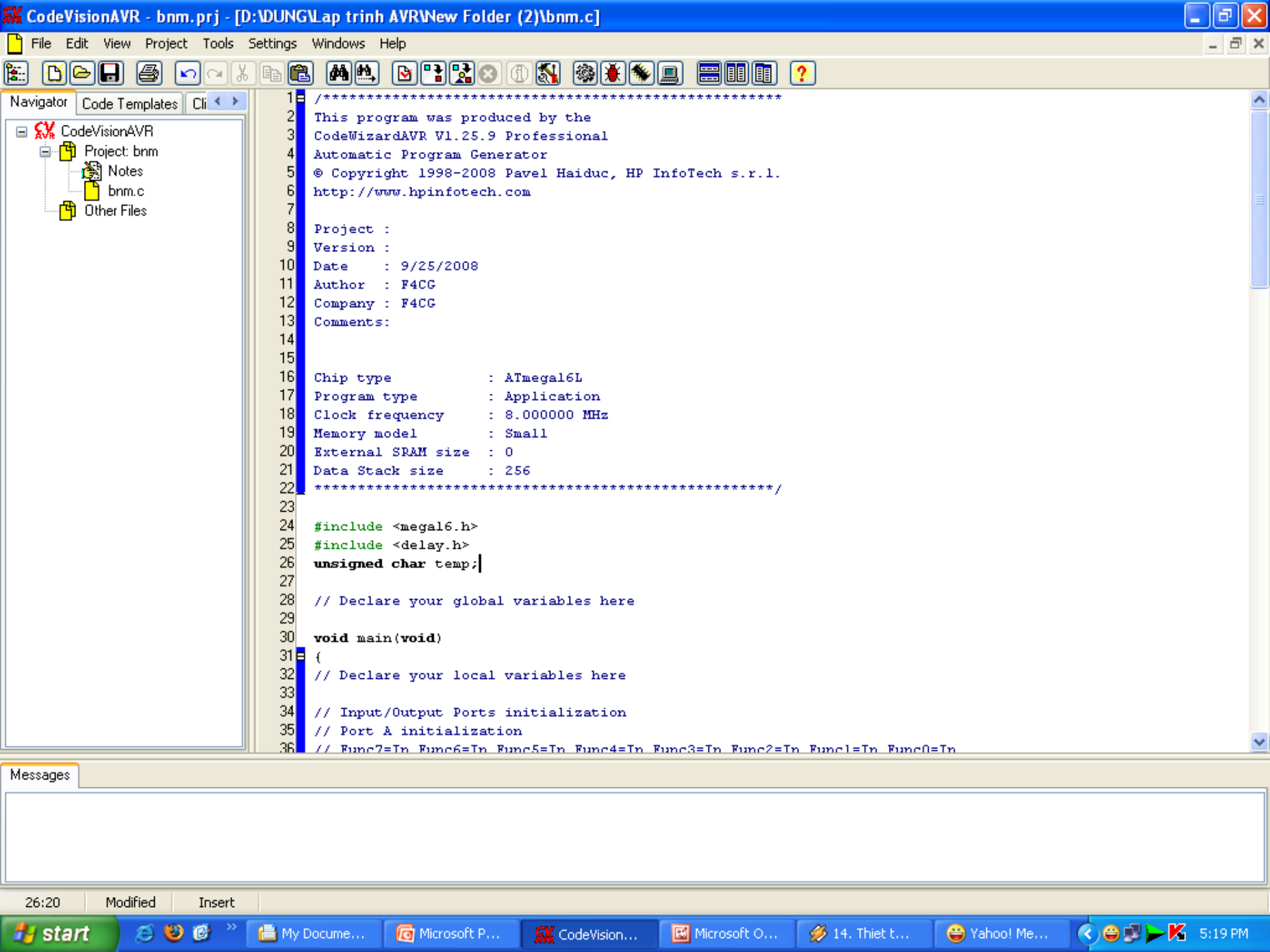




Sau khi nhớ xong 3 file IO.c – IO.prj – IO.cwp được cửa sổ như sau:

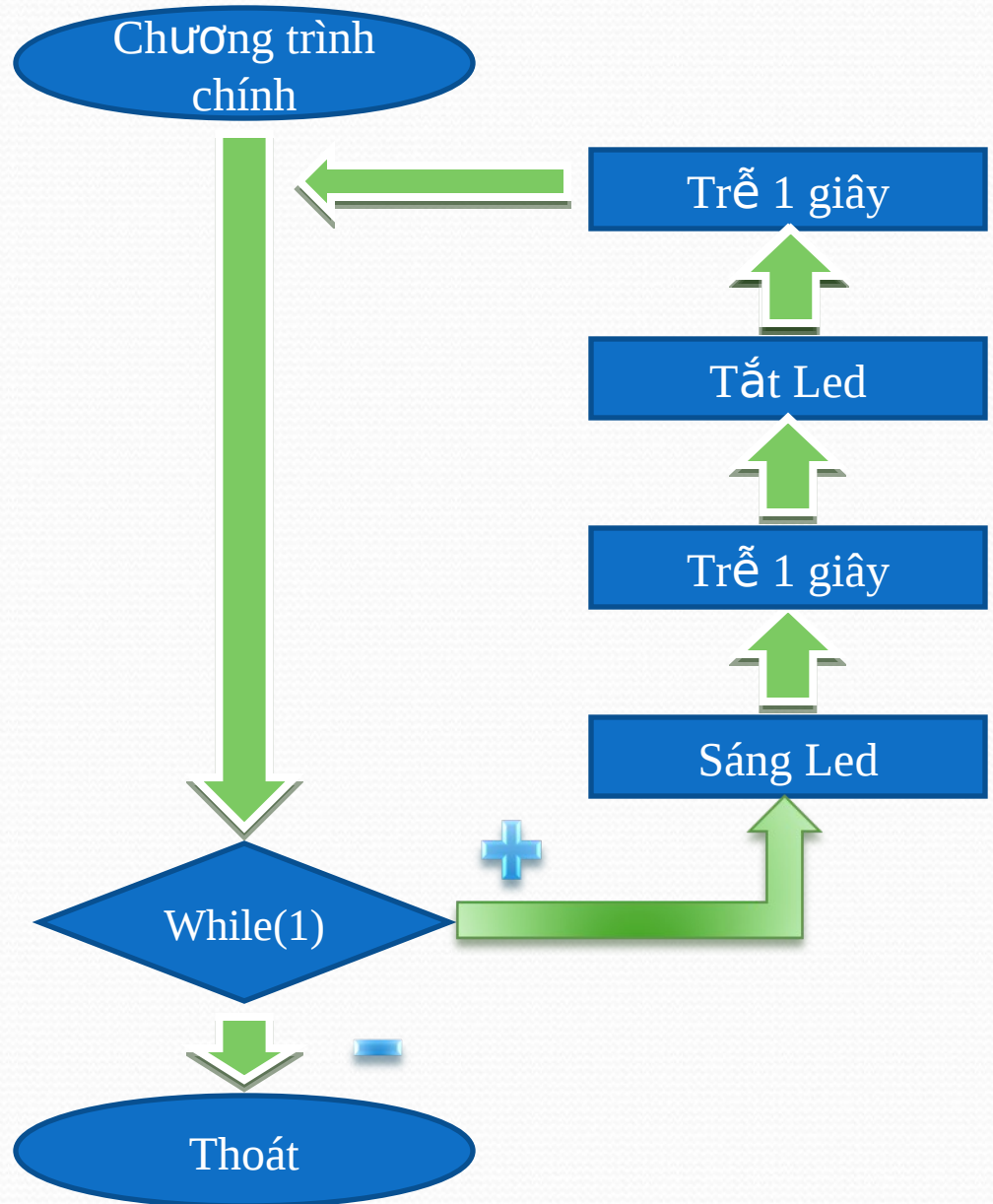


- Trong bài này chúng ta xây dựng chương trình đầu tiên, ct này sẽ làm nhấp nháy led.
- Ta phải sử dụng hàm `delay_ms()` trong thư viện “`delay.h`” để tạo trễ trong chương trình. Do đó ta phải chèn thư viện này vào. Các bạn thêm dòng `#include<delay.h>` vào ngay sau dòng `#include<mega16.h>` .
- Để nhấp nháy led, ta đưa vào cổng I/O một biến Temp dạng unsigned char có giá trị tăng dần từ 0 đến 255. Do đó ta thêm dòng `unsigned char temp;` vào ngay dòng dưới.



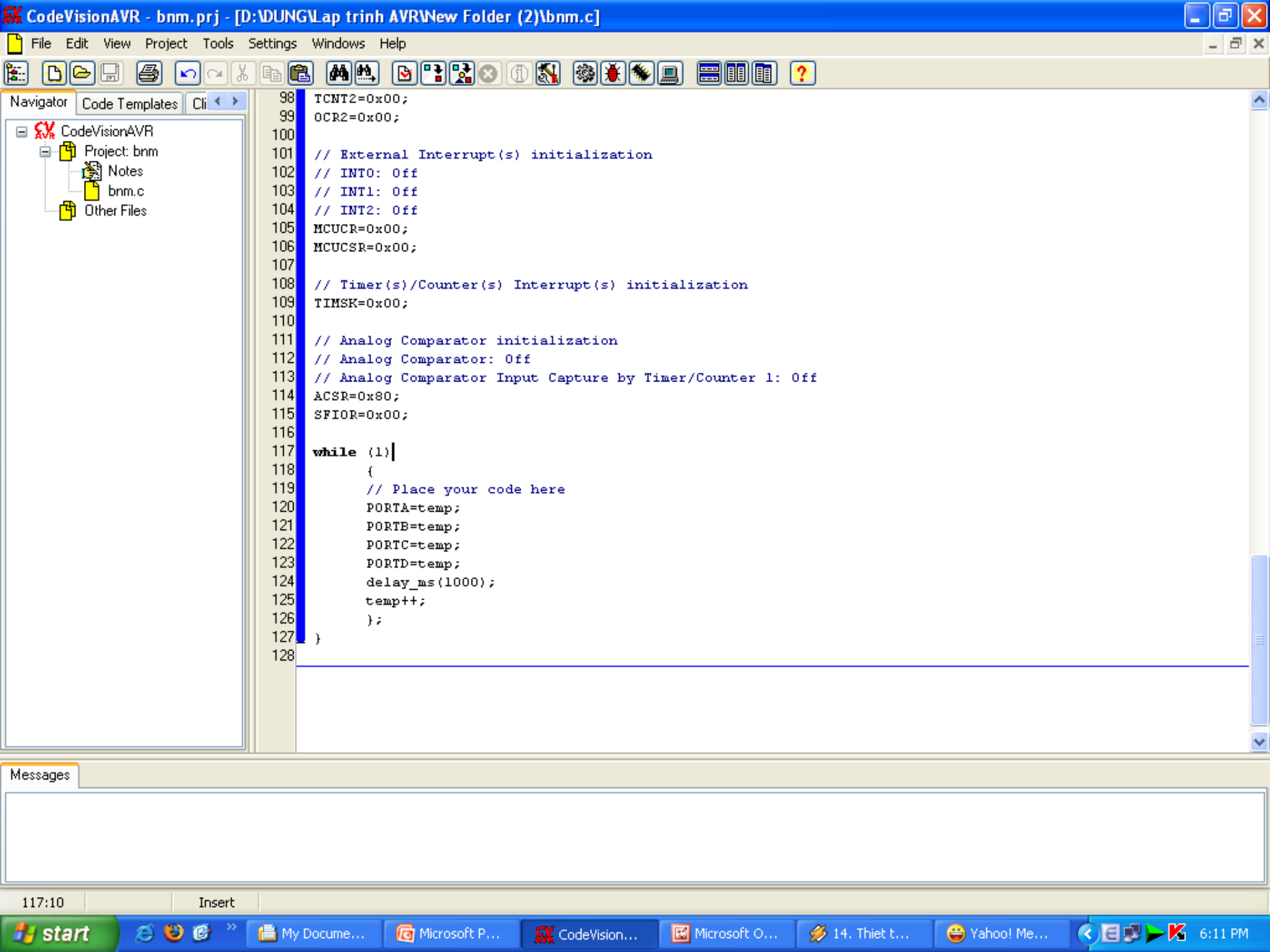


# Sơ đồ thuật toán



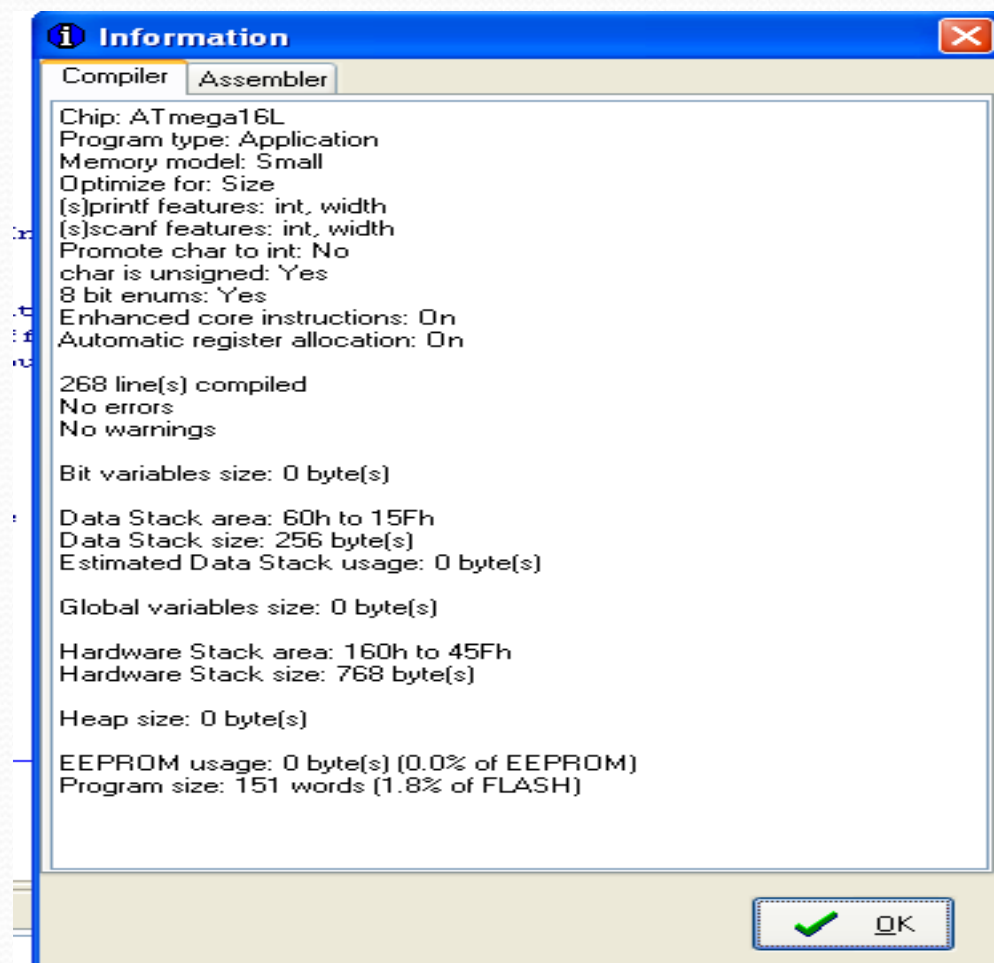
# CODE

```
while (1)
{
    // Place your code here
    PORTA=temp;
    PORTB=temp;
    PORTC=temp;
    PORTD=temp;
    delay_ms(1000);
    temp++;
};
```





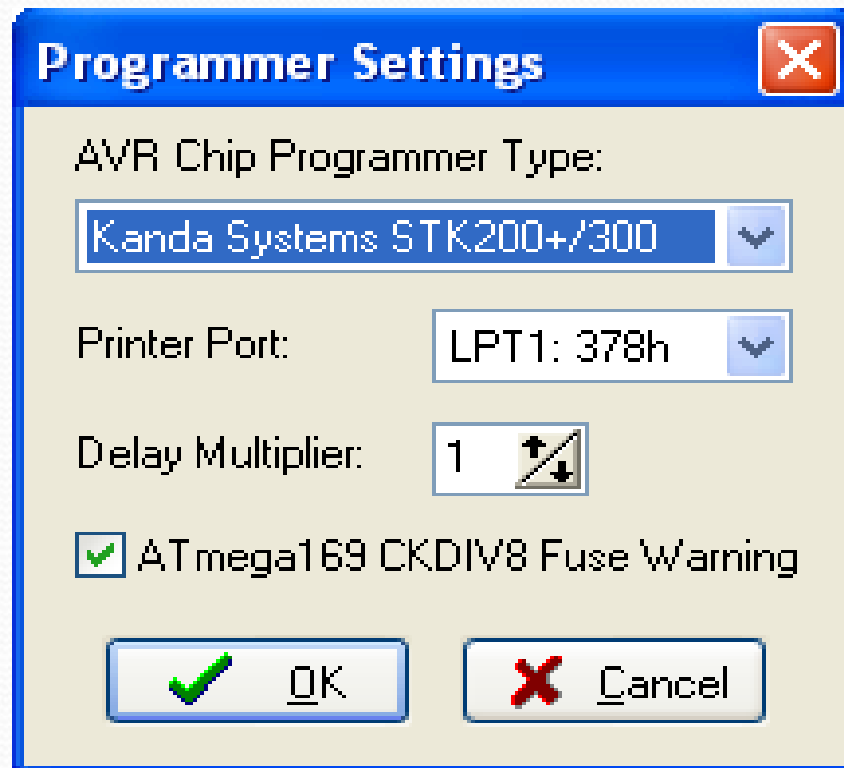
Để dịch chương trình ấn **Shift + F9** hoặc vào Menu : **Project** → **Make the Project**  
Được cửa sổ Information như sau:



Chương trình không có lỗi. Nhấp OK.

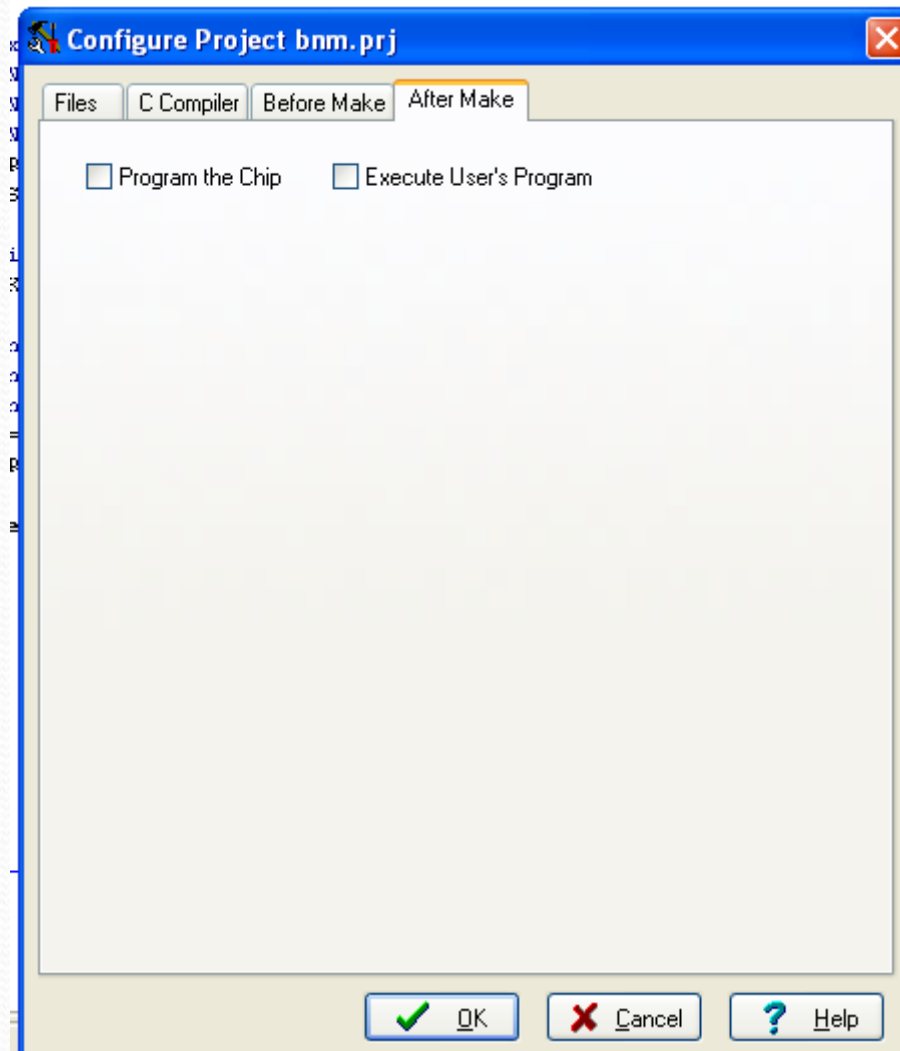
Để nạp chương trình các bạn cần cấu hình cho mạch nạp.

Vào menu : Setting-> Programmer được cửa sổ sau:



Chọn các mục như hình vẽ .

Chọn trên menu: Project -> Configure được cửa sổ như sau:





Trong Tab **After Make** các bạn đánh dấu vào **Program the Chip** và nhấn OK.

Nhấn tổ hợp phím **Shift + F9** được như hình bên.

Cắm Jum mạch nạp vào. Click vào **Program**. Đợi nạp xong nhõ jump nạp ra ấn Reset sẽ thấy led chạy.

