

GIỚI THIỆU SƠ LƯỢC VỀ VI ĐIỀU KHIỂN P89V51RD2



Bùi Trung Hiếu
Webmaster: [Khoa học và tuổi trẻ](#)

Như mọi lần trước, năm nay, Philips tiếp tục tài trợ cho tất cả các đội thi Robocon vi điều khiển P89V51RD2 thông qua bộ môn Điều khiển tự động-khoa Điện-điện tử trường ĐHBK TP Hồ Chí Minh, mỗi đội sẽ được nhận 2 MCUs <và nếu thấy các MCUs sử dụng tiện lợi, các đội có thể mua với số lượng không hạn chế!>

Nhằm giới thiệu với các bạn một số tính năng nổi bật của MCU P89V51RD2 phục vụ cuộc thi Robocon, trong bài viết này, tôi xin đưa ra một số đề nghị về mạch nạp, mạch ứng dụng, các chức năng điều xung <PWM>, Capture, giao tiếp PC qua cổng COM<UART>... để tiện lợi cho bạn trong khi sử dụng vi điều khiển này. Code của chương trình, tôi viết bằng ngôn ngữ C, qua đó, bạn thấy được tính tiện ích của C-Compiler.

Các đề mục trong bài:

1.	Khái quát các tính năng:	3
2.	Tổ chức bộ nhớ:	4
a.	Lựa chọn bank bộ nhớ chương trình flash:	4
b.	Thực thi đoạn code Reset nguồn sau khi bật nguồn: <Power-on reset code execution>	5
c.	Trình tự sử dụng mạch nạp:	6
3.	Định nghĩa các thanh ghi chức năng:	8
4.	Tạo chương trình giao tiếp PC-MCU:	9
5.	Chức năng PCA:	13
a.	Thanh ghi chức năng đặc biệt CMOD:	13
b.	Thanh ghi chức năng đặc biệt CCON:	14
c.	Thanh ghi chức năng so sánh/capture cho các modul PCA: CCAPMn	14
d.	Các chế độ hoạt động của modul PCA	14
e.	Chế độ Capture của PCA:	14
f.	Chức năng PWM của PCA:	16
6.	Kết luận:	17

Các hình vẽ trong bài:

Hình 1: Sơ đồ khối của MCU P89V51RD2.....	3
Hình 2: Mạch Reset hợp lệ.....	5
Hình 3: Mạch MCU P89V51RD2 khi sử dụng ISP.....	5
Hình 4: Mạch giao tiếp RS232 qua cổng COM.....	6
Hình 5: Màn hình khởi động FlashMagic.....	6
Hình 6: Thiết đặc các thông số cần thiết cho FlashMagic.....	7
Hình 7: Màn hình xuất hiện sau khi bấm Start của FlashMagic.....	7
Hình 8: Giao diện của chương trình Keil uV3.....	8
Hình 9: Bảng thông báo khởi tạo code khởi động cho họ 8051.....	9
Hình 10: Timer 2 ở chế độ sinh tốc độ baud.....	9
Hình 11: Kết quả chương trình giao tiếp UART.....	12
Hình 12: Hệ thống các ngắt ở chức năng PCA.....	13
Hình 13: Chế độ capture của modul PCA.....	15
Hình 14: Mô hình sử dụng chức năng Capture đề nghị với 4 chế độ quy định bởi Mode Select.....	15
Hình 15: Chức năng PWM của PCA.....	16

Các bảng số liệu:

Bảng 1: Các tốc độ baud thường dùng với tần số thạch anh 12MHz.....	10
Bảng 2: Các bit định nghĩa cho thanh ghi CMOD.....	13
Bảng 3 : Quy định cách chọn xung nhịp cho PCA.....	13
Bảng 4: Các bit định nghĩa của thanh ghi CCON.....	14
Bảng 5: Các bit định nghĩa thanh ghi CCAPMn.....	14
Bảng 6: Các chế độ hoạt động modul PCA.....	14

1. Khái quát các tính năng:

Trước tiên, ta lướt qua các tính năng của P89V51RD2:

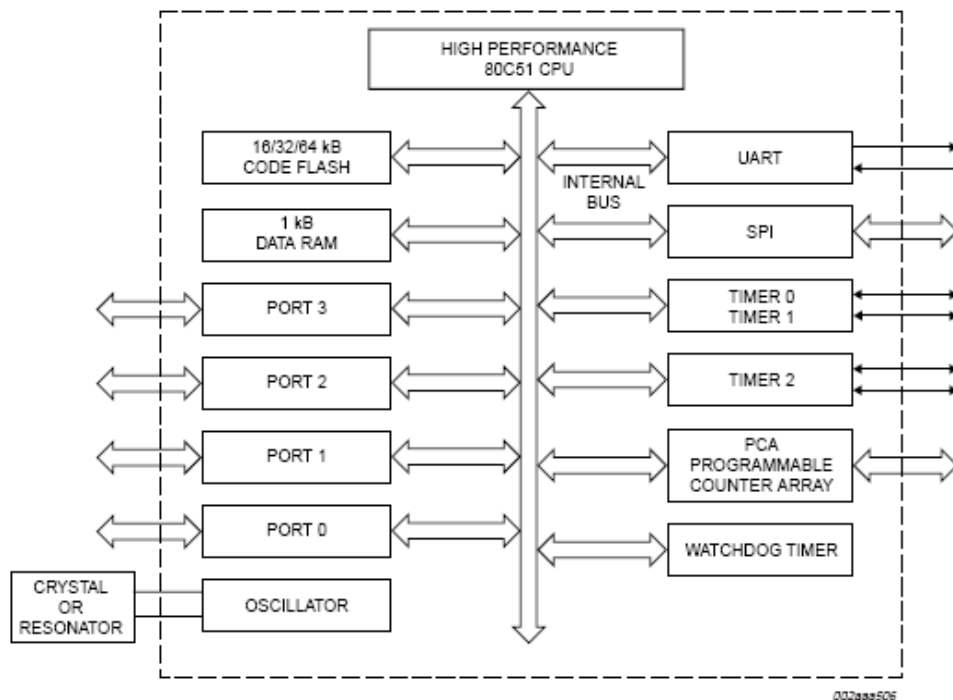
+Khái quát:

- Y P89V51RD2 là vi điều khiển 80C51 có 64kB Flash và 1024bytes<1kB> bộ nhớ dữ liệu RAM.
- Y Tính năng đặc biệt của P89V61RD2 là ở chế độ hoạt động mode x2. Người thiết kế chọn chạy ứng dụng của mình ở chế độ này để nâng đôi tốc độ khi hoạt động ở cùng tần số dao động<một chu kỳ máy=6 chu kỳ xung nhịp>
- Y Bộ nhớ chương trình Flash cho phép lập trình ISP hoặc/và song song. Chế độ lập trình song song được đưa ra để thích ứng với tốc độ cao, giảm thời gian và giá thành.
- Y IAP/ISP.

+Các tính năng:

- Y CPU 80C51.
- Y Hoạt động ở 5VDC trong tầm tần số dao động đến 40MHz.
- Y 64kB ISP.
- Y SPI
- Y 5 PCA với chức năng PWM/capture/compare 16bits.
- Y 4 cổng xuất nhập.
- Y 3 Timers/Couters 16bits.
- Y Watchdog Timer có thể lập trình được.
- Y 8 nguồn ngắt.
- Y 2 thanh ghi DPTR.
- Y Tương thích mức logic TTL và CMOS.
- Y Phát hiện nguồn yếu <Brownout Detect>
- Y Chế độ Low-power, Power down, Idle.

Sơ đồ khối của MCU P89V51RD2:



Hình 1: Sơ đồ khối của MCU P89V51RD2

Sơ qua về các chân của vi điều khiển:

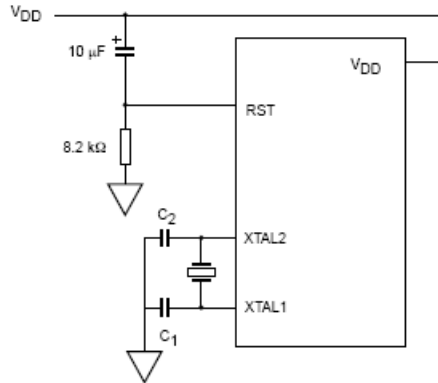
- + Port 0, Port 1, Port 2, Port 3: Như cấu trúc 8051 kinh điển.
- + P1.0 - T2: Ngõ vào Counter cho Timer/Counter 2 hoặc ngõ ra cho Counter/Timer 2.
- + P1.1 - T2EX: Điều khiển hướng và cạnh kích chức năng Capture cho timer/Counter 2.
- + P1.2 – ECI: Ngõ vào xung nhịp. Tín hiệu này là nguồn xung nhịp ngoài cho chức năng PCA.
- + P1.3 – CEX0: ngõ vào xung nhịp cho chức năng Capture/Compare modul 0.
- + P1.4:
 - o \overline{SS} : Chọn cổng phụ vào cho SPI.
 - o CEX1: ngõ vào xung nhịp cho chức năng Capture/Compare modul 1.
- + P1.5:
 - o MOSI: phục vụ SPI
 - o CEX2: ngõ vào xung nhịp cho chức năng Capture/Compare modul 2.
- + P1.6:
 - o MISO: phục vụ SPI
 - o CEX3: ngõ vào xung nhịp cho chức năng Capture/Compare modul 3.
- + P1.7:
 - o SCK: phục vụ SPI
 - o CEX4: ngõ vào xung nhịp cho chức năng Capture/Compare modul 4.
- + \overline{PSEN} : Cho phép dùng bộ nhớ chương trình ngoài. Khi MCU sử dụng bộ nhớ chương trình trong chip, \overline{PSEN} không tích cực. Khi sử dụng bộ nhớ chương trình ngoài, \overline{PSEN} thường ở mức tích cực 2 lần trong mỗi chu kỳ máy. Sự chuyển mức cao sang thấp \downarrow trên \overline{PSEN} cưỡng bức từ bên ngoài khi ngõ vào RST đang ở mức cao trong hơn 10 chu kỳ máy sẽ đưa MCU vào chế độ lập trình host từ bên ngoài.
- + RST: Khi nguồn dao động đang hoạt động, mức cao trên chân RST trong ít nhất 2 chu kỳ máy sẽ Reset lại hệ thống. Nếu chân \overline{PSEN} chuyển mức \downarrow trong khi RST vẫn còn ở mức cao, MCU sẽ vào chế độ lập trình host từ bên ngoài, nếu không, sẽ hoạt động bình thường.
- + \overline{EA} : Cho phép sử dụng bộ nhớ chương trình ngoài.
 - o $\overline{EA} = '0'$ \rightarrow Bộ nhớ ngoài.
 - o $\overline{EA} = '1'$ \rightarrow Bộ nhớ trong chip.
- + ALE/ \overline{PROG} : Cho phép khóa địa chỉ <Như 8051 cổ điển> ngoài ra, chân này còn được dùng để đưa vào chế độ lập trình FLASH.

2. Tổ chức bộ nhớ:

MCU P89V51RD2 có 2 vùng không gian địa chỉ riêng biệt: vùng lưu trữ cho bộ nhớ chương trình và cùng địa chỉ cho bộ nhớ dữ liệu <cấu trúc Harvard>

a. Lựa chọn bank bộ nhớ chương trình flash:

Có 2 vùng bộ nhớ nội flash trong MCU, Block 0 có 64kB và được tổ chức thành 512 sector, mỗi sector chứa 128 Bytes. Block 1 chứa chương trình ISP/ICP và được kích hoạt khi chọn kết hợp bit reset mềm (SWR) <FCF.1> và bit chọn bank (BSEL) <FCF.0>.



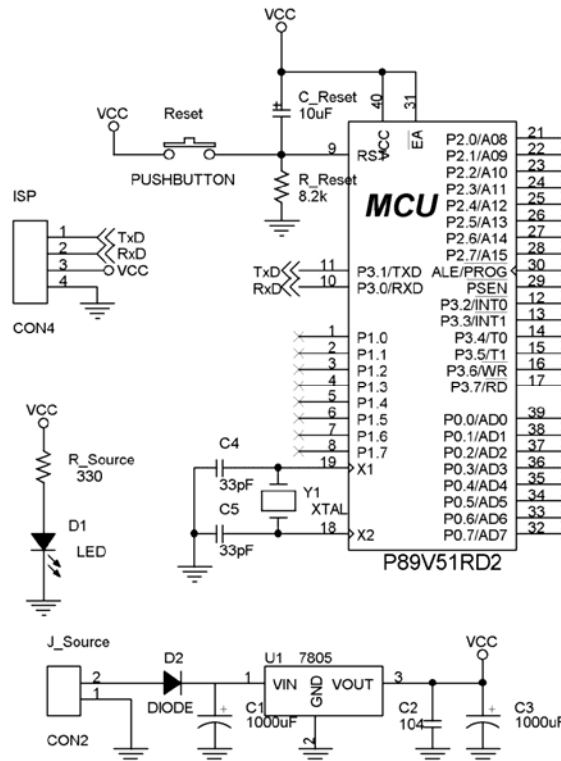
Hình 2: Mạch Reset hợp lệ

Quá trình tuần tự sau khi nguồn được bật, chương trình boot sẽ tự động thực thi và cố gắng lấy tín hiệu autobaud từ máy chủ. Nếu không có quá trình này xảy ra trong vòng 400ms và bit cờ SoftICE không được bật, chương trình boot sẽ tự động vào đoạn chương trình được người dùng lập trình.

b. Thực thi đoạn code Reset nguồn sau khi bật nguồn: <Power-on reset code execution>

Khi mới cấp điện vào MCU, tất cả các port đều ở trạng thái bất kì nào đó tới khi oscillator bắt đầu hoạt động và quá trình reset kéo tất cả các pin lên mức cao. Quá trình bật nguồn mà không đi đôi với việc Reset MCU hợp lý sẽ dẫn tới tình trạng ta không biết MCU thực thi lệnh ở vùng nhớ nào! Điều đó có thể dẫn đến mã lệnh trong flash bị thay đổi. Reset hệ thống trong khi chương trình đang hoạt động sẽ không có ảnh hưởng tới RAM của MCU, tuy nhiên, nội dung của vùng nhớ RAM trên chip trong quá trình bật nguồn là các giá trị ngẫu nhiên.

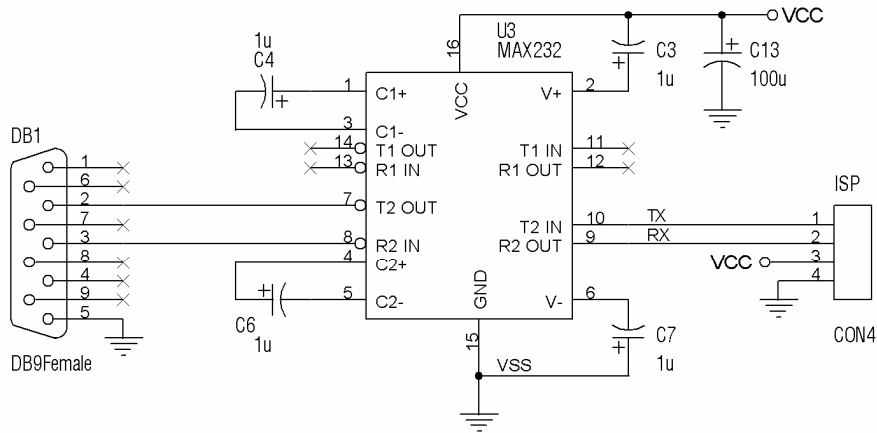
Khi bắt đầu mở nguồn, chân RST phải được giữ ở mức cao một khoảng thời gian đủ cho oscillator khởi động <thường là vài mili giây đối với dao động ở tần số thấp>, thêm nữa, để quá trình reset là hợp lệ, cần giữ mức cao ít nhất 2 chu kỳ máy. Các giá trị trở và tụ được chọn trên hình 2.



Hình 3: Mạch MCU P89V51RD2 khi sử dụng ISP

Nói một cách dễ hiểu, để quá trình Reset là hợp lệ, ta nên mắc mạch RC như hình 2. Khi dùng ở chức năng ISP/ICP, cần thêm một Push button kéo lên nguồn như hình 3. Quá trình Reset ngoài <chỉ nhấn nút RESET> và POR là có tác dụng như nhau.

Ta lưu ý cần nối các chân TxD, RxD đúng theo trình tự ở mạch giao tiếp RS232:



DB1: DB9 female nối cổng COM của PC
ISP: JACK kết nối hai mạch để lập trình

Hình 4: Mạch giao tiếp RS232 qua cổng COM

c. **Trình tự sử dụng mạch nạp:**

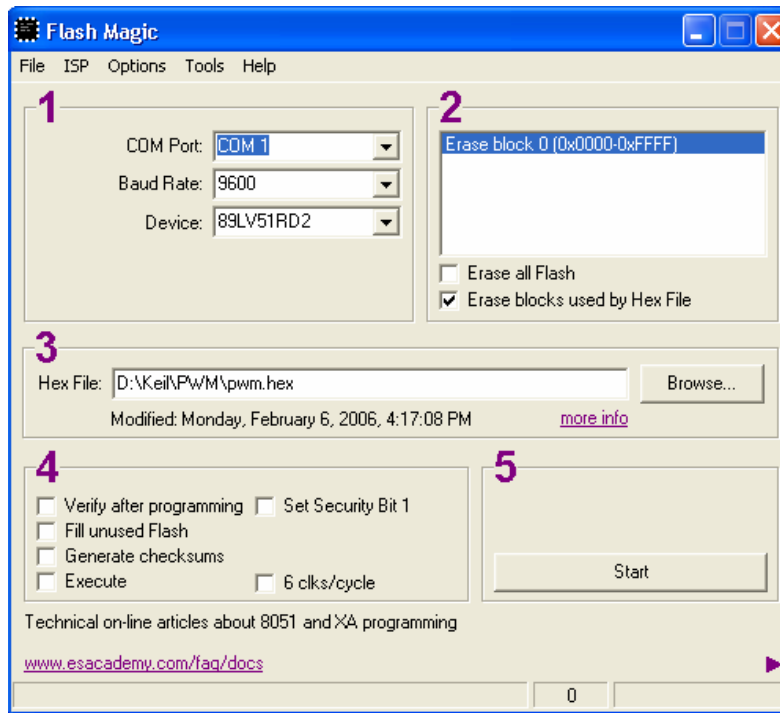
Tắt nguồn, gắn mạch cổng COM-RS232

- Mở chương trình nạp <FlashMagic, chọn Device, BaudRate, thiết lập các thông số>
- Browse đến File .hex cần nạp vào MCU.
- Bật nguồn.
- Nhấp Start trên trình FlashMagic.
- Bấm Reset ngoài.<Có thể đến lúc này, bạn mới bật nguồn, đây là quá trình POR>
- Kết nối ISP, chờ báo hiệu Complete
- Tắt FlashMagic

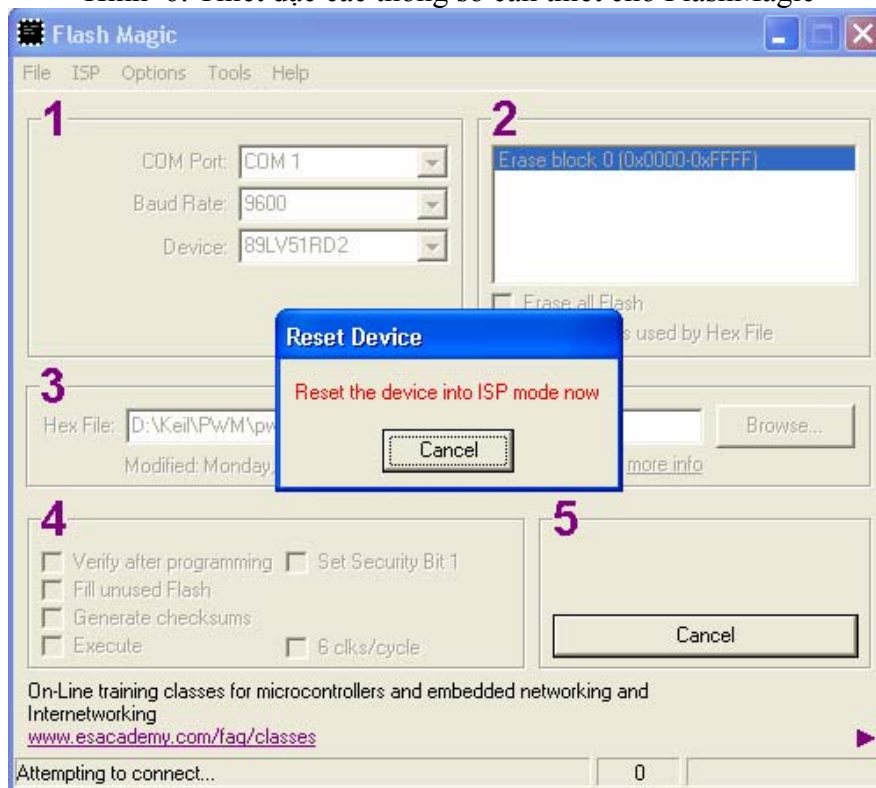
Các hình minh họa:



Hình 5: Màn hình khởi động FlashMagic



Hình 6: Thiết đặt các thông số cần thiết cho FlashMagic



Hình 7: Màn hình xuất hiện sau khi bấm Start của FlashMagic

Đến lúc này, bạn bấm nút Reset trên board mạch. Màn hình sẽ mất bảng thông báo Reset Device và đi vào chương trình ISP như đã phân tích ở trên. Do hiện tại, gặp một số trục trặc về thiết bị thí nghiệm nên tôi chưa thể lấy kết quả thực tế đưa vào bài viết. Tuy nhiên, với lý thuyết khảo sát từ Datasheet, kết quả là đáng tin cậy.

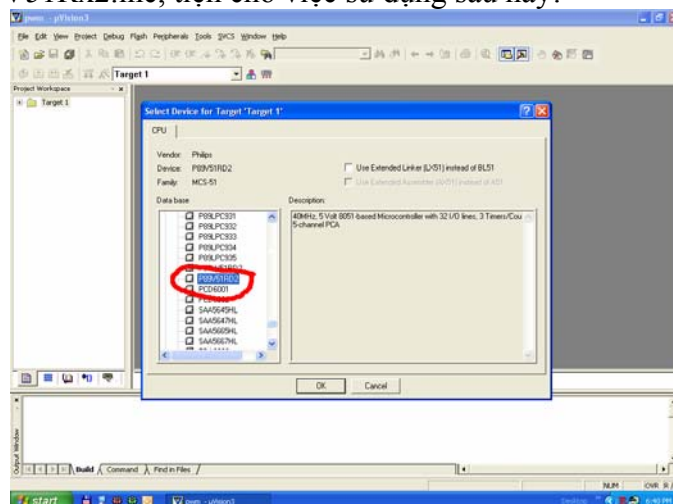
3. Định nghĩa các thanh ghi chức năng:

Do P89V51RD2 có thêm một số thanh ghi chức năng đặc biệt, nếu bạn đang sử dụng các trình biên dịch thông thường, cần phải khai báo các thanh ghi chức năng đó. Có thể xem các trang 11-12-13 của datasheets để định nghĩa các thanh ghi chức năng. Ví dụ:

*;Đoạn code dưới đây để ví dụ cho định nghĩa các thanh ghi chức năng khi dùng ASM.
;Khởi tạo 200601312020*

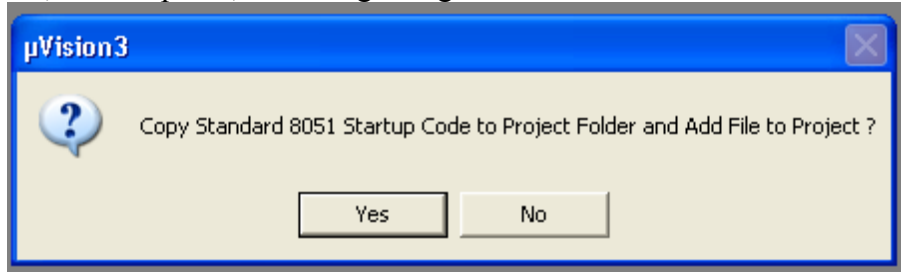
CCAP0H	DATA	0FAH
CCAP1H	DATA	0FBH
CCAP2H	DATA	0FCH
CCAP3H	DATA	0FDH
CCAP4H	DATA	0FEH
CCAP0L	DATA	0EAH
CCAP1L	DATA	0EBH
CCAP2L	DATA	0ECH
CCAP3L	DATA	0EDH
CCAP4L	DATA	0EEH
CCAPM0	DATA	0DAH
CCAPM1	DATA	0DBH
CCAPM2	DATA	0DCH
CCAPM3	DATA	0DDH
CCAPM4	DATA	0DEH
ECOM_0	BIT	CCAPM0.6
CAPP_0	BIT	CCAPM0.5
CAPN_0	BIT	CCAPM0.4
MAT_0	BIT	CCAPM0.3
TOG_0	BIT	CCAPM0.2
PWM_0	BIT	CCAPM0.1
ECCF_0	BIT	CCAPM0.0

Ở đây, tôi sử dụng phần mềm Keil-uV3 để làm chương trình soạn thảo chính và viết bằng ngôn ngữ C nên chỉ cần `#include<P89V51Rx2.h>`: [<Bạn có thể load tại đây>](#) Nếu viết bằng ASM, có thể đối chiếu với file header này để tạo file P89V51Rx2.inc, tiện cho việc sử dụng sau này!



Hình 8: Giao diện của chương trình Keil uV3

Sau khi chọn Device, bạn nhấp OK, hiện bảng thông báo:



Hình 9: Bảng thông báo khởi tạo code khởi động cho họ 8051

Bạn chọn **Yes**.

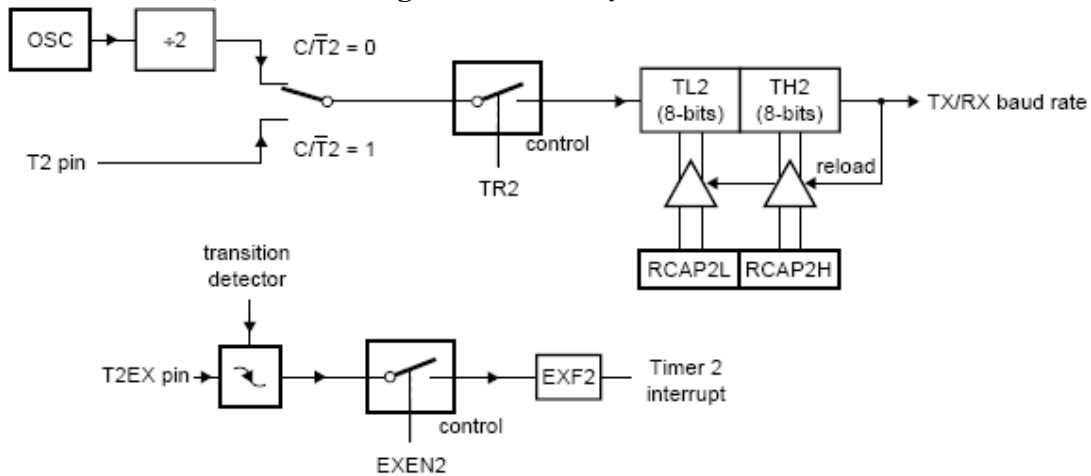
Bấm tổ hợp phím Ctrl+N, tạo một trình soạn thảo mới.

4. Tạo chương trình giao tiếp PC-MCU:

Theo thông lệ khi viết một Embedded system, tôi viết trình giao tiếp MCU-PC, với vi điều khiển P89V51RD2 tôi cũng bắt đầu như vậy:

Bit TCLK và/hoặc RCLK trong thanh ghi T2CON cho phép chọn nguồn sinh tốc độ Baud UART truyền phát từ timer1 hoặc 2:

- ❖ **SCON: SM0-SM1=0-1:** UART mode 1: 8-bit UART variable
- ❖ **TI:** cờ ngắt truyền phát
- ❖ **RI:** cờ ngắt thu nhận.
- ❖ **REN:** cho phép thu.
- ❖ **SM2='0'** ở mode 1.
- ❖ **Khi TCLK = 0, Timer 1 là nguồn sinh tốc độ Baud,**
- ❖ **Khi TCLK = 1, Timer 2 là nguồn sinh tốc độ Baud.**



Hình 10: Timer 2 ở chế độ sinh tốc độ baud.

$$BaudRate = \frac{OscillatorFreq}{16 \times (65536 - (RCAP2H, RCAP2L))} \quad \text{<Công thức 1>}$$

với RCAP2H-RCAP2L là giá trị Integer 16 bit

không dấu.

Tốc độ baud	Tần số thạch anh	Timer 2	
		RCAP2H	RCAP2L
750000	12MHz	FF	FF
19200	12MHz	FF	D9
9600	12MHz	FF	B2
4800	12MHz	FF	64
2400	12MHz	FE	C8
600	12MHz	FB	1E
220	12MHz	F2	AF

Bảng 1: Các tốc độ baud thường dùng với tần số thạch anh 12MHz

Chương trình được viết bằng C: <[load về](#)>

```

/*****
Chương trình thu giao tiếp UART
Tac gia       :      Bui Trung Hieu - Webmaster: http://www.khvt.com
Su dung       :      Co the su dung voi moi muc dich ma khong can xin phep
                :      tac gia ve tac quyen, tac gia khong chiu moi trach nhem ve
                :      cac tac hai gay ra va khong doi hoi cac loi ich co the co

Khoi tao      :      Ngay 31 thang 1 nam 2006 luc 9h15'PM
Hoan thanh    :      Ngay 31 thang 1 nam 2006 luc 10h15'PM
*****/

#include <P89V51Rx2.h>
#include "uart.h"
#include "stdio.h"

void main (void)
{
    //Khoi tao chuc nang UART
    uart_init();
    //Cho phep ngat chung
    EA=1;
    //Tich cuc co ngat phat
    TI=1;
    //In cac ki tu tren man hinh PC
    printf("\n*****");
    printf("\n    Hom qua tat nuoc dau dinh    ");
    printf("\n    Bo quen cai ao tren canh hoa sen ");
    printf("\n    Em duoc thi cho anh xin    ");
    printf("\n    Hay la em de lam tin trong nha? ");
    printf("\n***** ");
    //Xoa co ngat phat, bao hieu khong truyen phat nua
    TI=0;
    //Vong lap vo tan, chuong trinh dung tai day!
    while(1);
}
//End

```

Trong đó, hàm khai báo cho chức năng UART là:

```
// Khai bao thu vien thanh ghi cua P89V51RD2 va cac ham dung trong uart
#include <P89V51Rx2.h>
#include "uart.h"

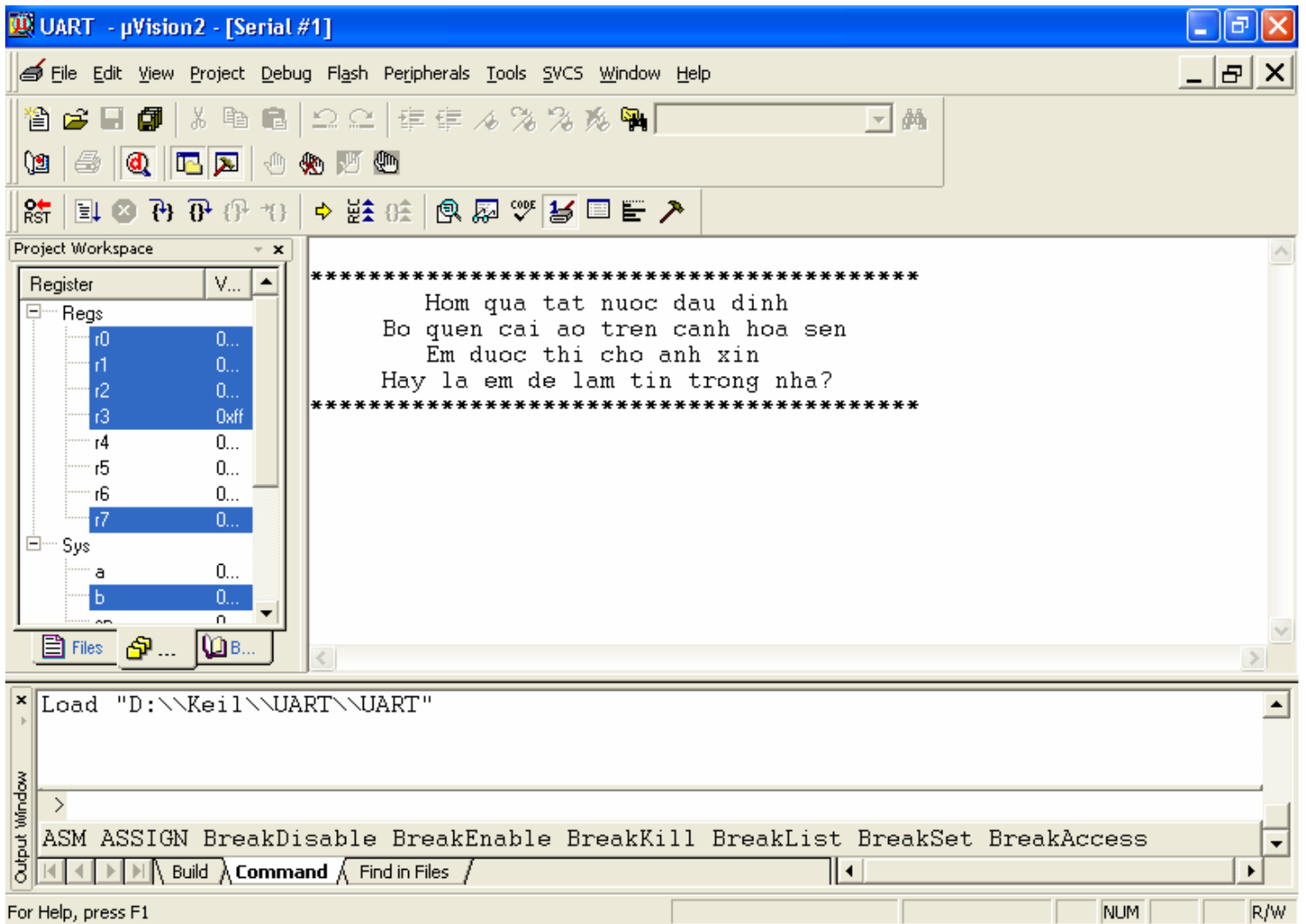
/*****
Chuc nang      :   Khoi tao UART o mode 1,
                  -Su dung dao dong Osc 12Mhz
                  -BAUDRATE:   19200
Gia tri tra ve:   Khong co
Luu y          :   De dung ket hop voi ngat uart, phai dat EA=1
*****/
void uart_init(void)
{
    //Cau hinh cho UART
    //Chon mode 1
    SM0=0;
    SM1=1;
    SM2=0;      //Chu y neu ban load file header tu trang ban dan cua Philips
                //phai sua lai SM3 thanh SM2, toi khong hieu tai sao co truc trac nay
    //Chon Timer 2 la nguon sinh toc do Baud
    //Cau hinh cho Timer 2, xem them o phan Timer 2 trong Datasheet
    TCLK=1;
    //T2OE=0; luu y thanh ghi T2MOD khong can thiep tung bit duoc
    T2MOD|=0x02;
    //Cho Timer 2 chay
    TR2=1;

    //Toc do Baud la 19200, luu y cong thuc 3 hoac bang 24 trong Datasheet
    //Tuc la cong thuc 1 hoac bang 1 trong bai gioi thieu nay
    RCAP2H=0xFF;
    RCAP2L=0xD9;

    // Khong co uu tien ngat
    IP0&=0xEF;
    IPOH&=0xEF;
    // Cho phep ngat noi tiep
    ES = 1;
} //Ket thuc ham khoi tao UART

void uart_isr(void) interrupt 4 using 0
{
}
//Ham ngat khong lam gi ca() ☺
```

Kết quả với chương trình Debug:

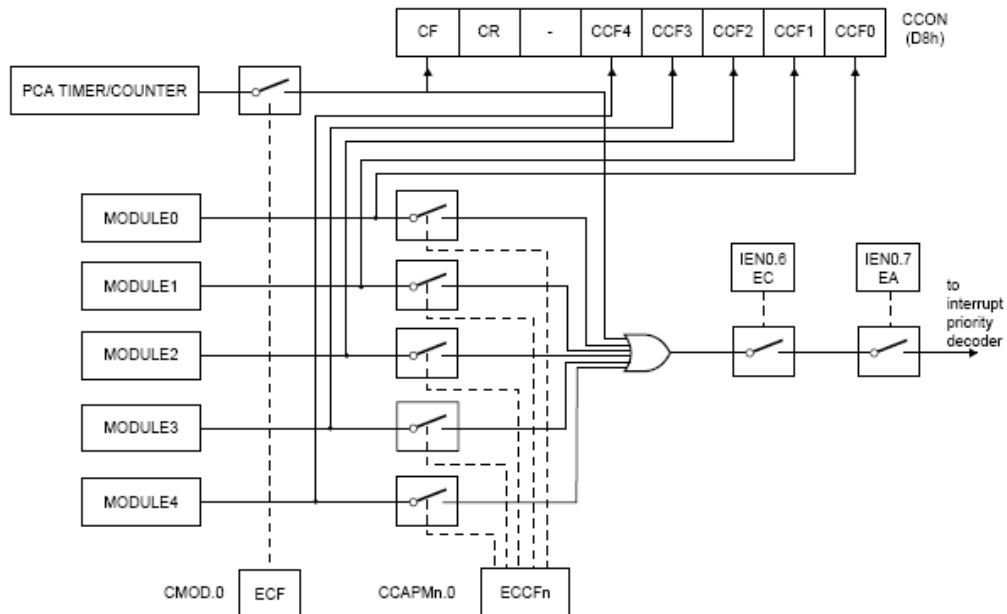


Hình 11: Kết quả chương trình giao tiếp UART

Nếu bạn để ý, chương trình này tôi viết hoàn toàn giống hàm main cho chương trình giao tiếp của P89LPC922<xem thêm trong phần P89LPC922>, chỉ có các khai báo cần thiết cho `uart_init()` là thay đổi để thích hợp. Tính linh hoạt của chương trình được viết bằng C thể hiện rất rõ nét. Trong trường hợp bạn đã có một số thư viện khai báo ngoại vi cần thiết, thời gian để có thể hoàn thành một chương trình là rất ngắn.

5. Chức năng PCA:

Chức năng nổi bật của P89V51RD2 là có 5 kênh PCA, các thanh ghi chịu tác động: CMOD, CCON, CCAPMn.



Hình 12: Hệ thống các ngắt ở chức năng PCA

a. Thanh ghi chức năng đặc biệt CMOD:

CMOD	PCA Counter Mode	D9H	CIDL	WDTE	-	-	-	CPS1	CPS0	ECF
------	------------------	-----	------	------	---	---	---	------	------	-----

Không thể can thiệp vào từng bit.

Bit	Kí hiệu	Chức năng
7	CIDL	Điều khiển trạng thái rỗi của Counter PCA. CIDL=0 sẽ cho Counter PCA tiếp tục hoạt động bất chấp đang trong trạng thái rỗi. CIDL=1 sẽ lập trình cho nó không hoạt động trong trạng thái rỗi.
6	WDTE	Cho phép Watchdog Timer trên modul 4.
5-3	-	Chưa được khai báo
2-1	CPS1-CPS0	Lựa chọn nguồn xung đếm cho PCA <Xem bảng dưới>
0	ECF	Cho phép ngắt khi tràn Counter PCA.

Bảng 2: Các bit định nghĩa cho thanh ghi CMOD

Ta sẽ lợi dụng chức năng PWM của PCA để điều xung, nhằm chỉnh mức áp đặt vào động cơ, qua đó, thay đổi tốc độ của nó.

PCA là Timer 16bit đặc biệt, gồm 5 modul. Mỗi modul có thể lập trình để vận hành 1 trong 4 chế độ: capture cạnh lên/xuống, timer, ngõ ra tốc độ cao, hay PWM. Timer PCA có thể vận hành nhờ các nguồn xung nhịp: $\frac{1}{2}$ OscFreq ; $\frac{1}{6}$ OscFreq; tốc độ tràn của timer 0; hay từ ngõ vào trên chân ECI(P1.2). Nguồn xung nhịp của PCA được chọn nhờ các bit CPS1-CPS0 trên thanh ghi đa dụng CMOD.

CPS1	CPS0	Chọn xung nhịp cho PCA
0	0	Fosc/6
0	1	Fosc/2
1	0	Tràn từ Timer 0
1	1	Xung nhịp ngoài (ECI) Tốc độ lớn nhất Fosc/4

Bảng 3 : Quy định cách chọn xung nhịp cho PCA

b. *Thanh ghi chức năng đặc biệt CCON:*

CCON*	PCA Counter Control	D8H	CF	CR	-	CCF4	CCF3	CCF2	CCF1	CCF0
-------	---------------------	-----	----	----	---	------	------	------	------	------

Có thể can thiệp vào từng bit

Bit	Kí hiệu	Chức năng
7	CF	Cờ đếm tràn PCA.
6	CR	Bit điều khiển chạy counter PCA
5	-	Chưa định nghĩa
4	CCF4	Cờ ngắt modul PCA 4
3	CCF3	Cờ ngắt modul PCA 3
2	CCF2	Cờ ngắt modul PCA 2
1	CCF1	Cờ ngắt modul PCA 1
0	CCF0	Cờ ngắt modul PCA 0

Bảng 4: Các bit định nghĩa của thanh ghi CCON

c. *Thanh ghi chức năng so sánh/capture cho các modul PCA: CCAPMn*

CCAPM0	Module 0 Mode	DAH	-	ECOM_0	CAPP_0	CAPN_0	MAT_0	TOG_0	PWM_0	ECCF_0
CCAPM1	Module 1 Mode	DBH	-	ECOM_1	CAPP_1	CAPN_1	MAT_1	TOG_1	PWM_1	ECCF_1
CCAPM2	Module 2 Mode	DCH	-	ECOM_2	CAPP_2	CAPN_2	MAT_2	TOG_2	PWM_2	ECCF_2
CCAPM3	Module 3 Mode	DDH	-	ECOM_3	CAPP_3	CAPN_3	MAT_3	TOG_3	PWM_3	ECCF_3
CCAPM4	Module 4 Mode	DEH	-	ECOM_4	CAPP_4	CAPN_4	MAT_4	TOG_4	PWM_4	ECCF_4

Không can thiệp được vào từng bit.

Bit	Kí hiệu	Chức năng
7	-	Chưa định nghĩa
6	ECOMN	Cho phép bộ so sánh
5	CAPPN	Cho phép capture cạnh lên
4	CAPNN	Cho phép capture cạnh xuống
3	MATN	Cho phép kết nối với cờ ngắt CCFn ở thanh ghi CCON
2	TOGN	Lật mức ngõ ra. Tác động đến chân CEXn
1	PWMN	Chế độ điều xung. Tác động đến chân CEXn
0	EXCFN	Cho phép ngắt CCFn.

Bảng 5: Các bit định nghĩa thanh ghi CCAPMn

d. *Các chế độ hoạt động của modul PCA*

Xem bảng sau:

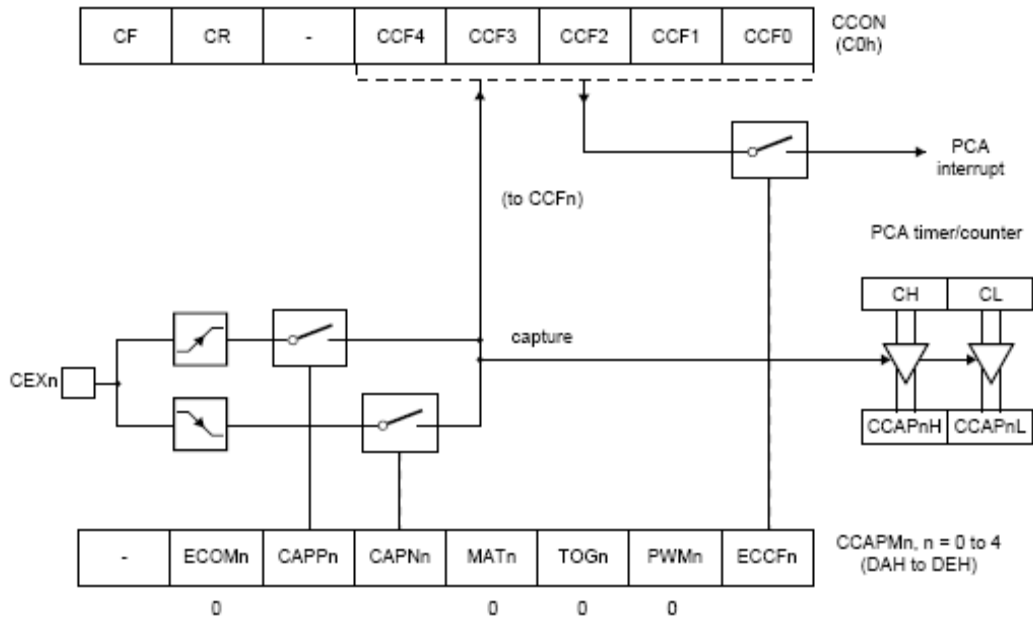
ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Chế độ hoạt động
0	0	0	0	0	0	0	Không hoạt động
x	1	0	0	0	0	x	Capture 16 bit kích cạnh lên CEX
x	0	1	0	0	0	x	Capture 16 bit kích cạnh xuống CEX
x	1	1	0	0	0	x	Capture 16 bit kích cạnh CEX
1	0	0	1	0	0	x	Timer 16 bit
1	0	0	1	1	0	x	Ngõ ra tốc độ cao 16 bit
1	0	0	0	0	1	0	PWM 8 bit
1	0	0	1	x	0	x	Đồng hồ Watchdog.

Bảng 6: Các chế độ hoạt động modul PCA

e. *Chế độ Capture của PCA:*

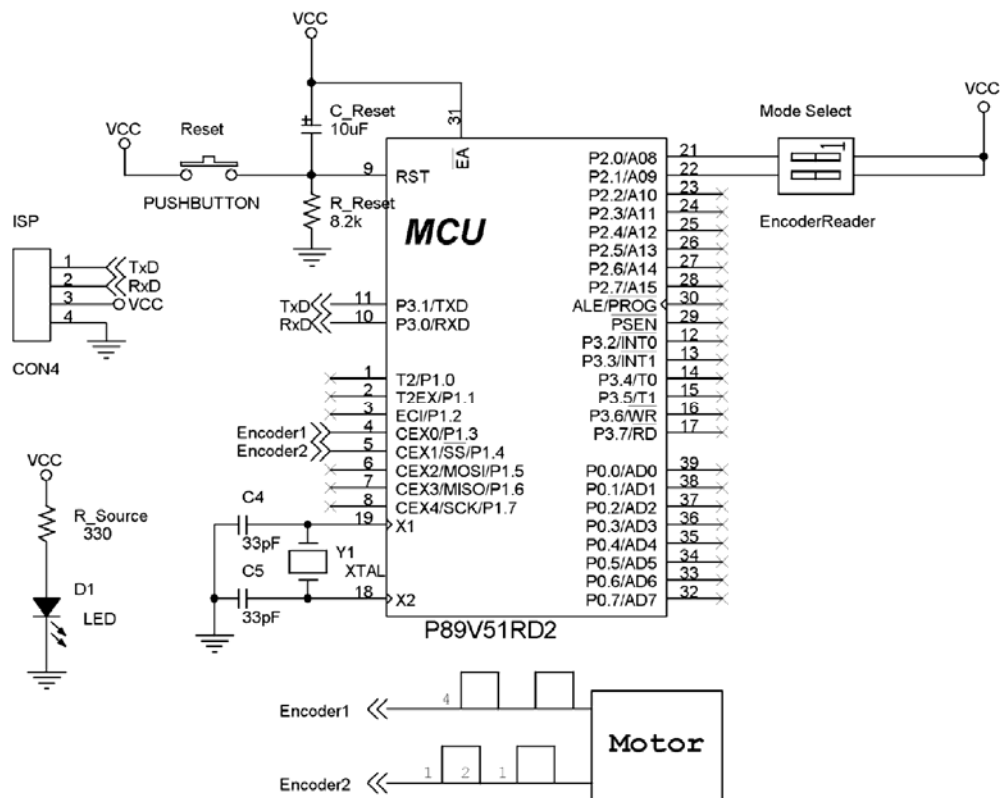
Phải có ít nhất 1 bit trong CAPP hoặc CAPN được bật; sự chuyển mức CEX sẽ thành ngõ vào Capture cho modul này. Khi có sự chuyển mức hợp lệ trên CEX, phần cứng của modul PCA sẽ tự nạp các

giá trị thanh ghi Timer/Counter của PCA (CH và CL) vào các thanh ghi Capture modul PCA đó <CCAPnL, CCAPnH>. Xem hình vẽ dưới để thấy rõ hơn chức năng đây.



Hình 13: Chế độ capture của modul PCA

Ứng dụng của chức năng Capture vào việc đọc Encoder gắn cạnh ở các mode Normal, x2, x4 tỏ ra rất tiện lợi, dùng kết hợp 2 modul này để phát hiện thêm hướng di chuyển cũng rất tốt. Mô hình đề nghị của tôi đưa ra dưới đây, nếu bạn thực hiện thành công, nhớ gửi mã nguồn, để các bạn khác cùng tham khảo nhé!

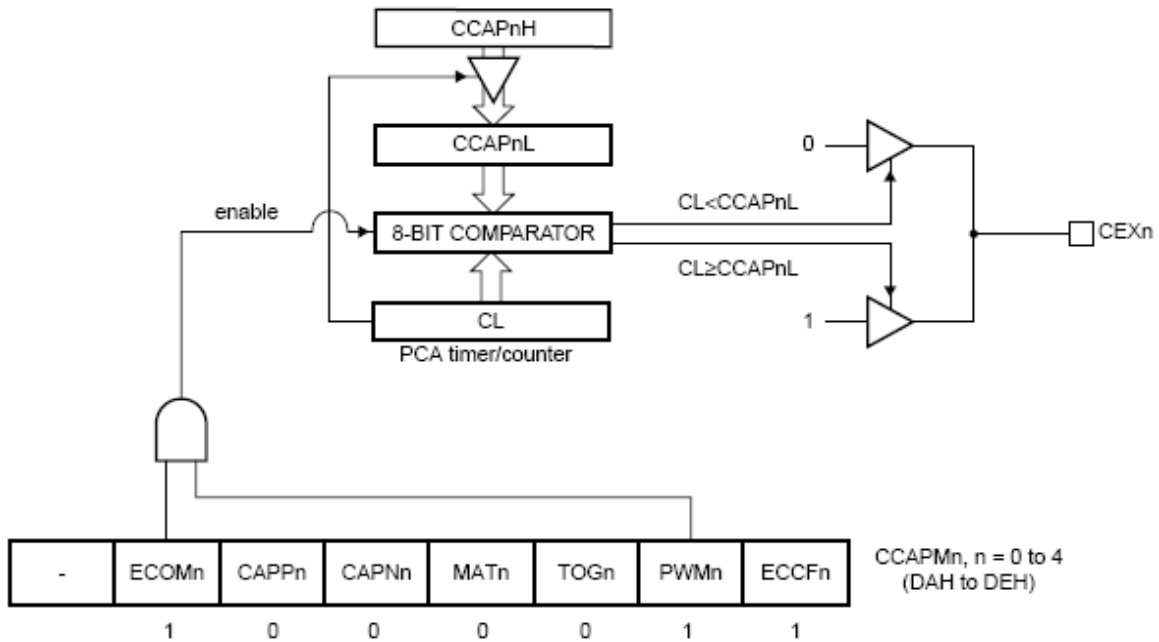


Hình 14: Mô hình sử dụng chức năng Capture đề nghị với 4 chế độ quy định bởi Mode Select

Khi sử dụng hai dây Encoder, giản đồ xung của chúng như hình vẽ trên, giả sử bạn dùng ngắt cạnh lên, 2 số 1-1 biểu thị mode 1, số 1-2 biểu hiện mode x2: bạn dùng chế độ ngắt cạnh, số 1-4 biểu hiện mode x4. Đồng thời, đọc theo thứ tự ngắt cạnh lên E1-E2 là hướng tới, theo thứ tự E2-E1 là hướng lùi, ta cũng có thể dùng mode x4 để triệt tiêu nhiễu đo: trường hợp khi dừng, bánh xe không dừng hẳn mà nhích tới nhích lui, dao động quanh vị trí cân bằng, bằng cách tính tổng số xung tiến lùi trong một lần lấy mẫu, ta có thể biết được vị trí góc quay tương đối của động cơ.

Vì chức năng Capture còn tương đối mới với một số bạn mới làm quen với vi điều khiển <tôi cũng vậy☺> nên tôi muốn nói thêm một chút về chức năng này: Giả sử bạn dùng một chương trình điều khiển vận tốc động cơ <PID chẳng hạn, hay một chương trình có chức năng gần như thế> bạn cần biết vận tốc chính xác của hai động cơ Robot mình đang vận hành, cách thông thường hiện nay, là ta đếm số xung trong 1 chu kì lấy mẫu, từ đó suy ra vận tốc; cách này không đạt độ chính xác cao <trước kia, tôi đã sai lầm do vội vã và phân tích quá sơ sài thời gian thực và độ chính xác khi so sánh 2 cách đọc vận tốc-bạn xem thêm ở bài DC Motor>. Nếu ta dùng chức năng Capture, đọc khoảng thời gian giữa 2 lần ngắt cạnh <giả sử đang dùng ở mode 1>, biết số xung của encoder, ta có thể dùng công thức $[v]=s/t$ để tính vận tốc trung bình trong khoảng T_s . Trong công thức, s là khoảng cách giữa 2 xung cạnh lên liên tiếp, giả sử Encoder có 100 xung thì $s=2*\pi/100$, t là khoảng thời gian lưu có đánh dấu tương đối trong CCAP. Giả sử trong khoảng thời gian T_s , bạn được n mẫu, lấy trung bình sẽ được vận tốc trung bình lấy mẫu, cách này tỏ ra ưu thế nếu khoảng thời gian lấy mẫu ngắn và yêu cầu chính xác cao. Bạn chú ý thêm tới ngưỡng trên của chức năng Capture, và có thể Timer của PCA bị tràn trong quá trình lấy mẫu, những chú ý trên nếu bạn khắc phục được, tôi nghĩ chắc chắn Capture sẽ rất tốt trong ứng dụng này. Khi bạn hoàn thành mỹ mãn, nhớ gọi cho tôi đoạn code và chương trình giải thuật để cùng chia sẻ với mọi người nhé!

f. Chức năng PWM của PCA:



Hình 15: Chức năng PWM của PCA

Ở chức năng này, giá trị ở ngõ ra CEX phụ thuộc vào giá trị của 2 thanh ghi: CCAPnL và CL, khi CL tăng lớn hơn CCAPnL thì CEX sẽ chuyển sang mức 1, ngược lại, sẽ ở mức 0, mỗi lần tràn, CCAPnL sẽ nạp lại giá trị từ CCAPnH, chính điều này cũng cho phép ta cập nhật duty cycle mới mà không gây ảnh hưởng tới quá trình PWM.

Chức năng PWM này có độ phân giải 8 bit, tức là ta được 256 mức chia <trung đối mìn với áp 24V thì mỗi mức sẽ được $24/256 \approx 94mV$ >

6. Kết luận:

Trong phạm vi bài viết này, do thời gian cũng rất hạn chế nên không thể nêu tất cả các tính năng của một vi điều khiển, các chức năng PWM, Capture, UART, tôi nghĩ bạn sẽ cần tới trong kì thi Robocon cho điều khiển động cơ, kiểm tra một số chức năng, nên tôi mạnh dạn đưa ra, có thể còn nhiều sai sót, mong bạn thông cảm. Các chức năng còn lại, tôi nghĩ hoặc bạn đã quen khi sử dụng với AT89C51/52, hoặc sẽ ít dùng tới nên không được đưa ra trong bài viết này.

Mục đích của tôi trong bài viết này, là nêu một cách khái quát để bạn có thể ứng dụng một cách dễ dàng những tính năng của vi điều khiển P89V51RD2. Bạn nên làm quen với trình biên dịch C-Compiler của Keil-uV3, nó sẽ giúp bạn nhiều trong các ứng dụng sau này, cho cả những vi điều khiển khác, không phải chỉ riêng P89V51RD2.

Khi đã thực hiện tốt các ứng dụng của bạn, tôi nghĩ sự sẻ chia sẽ rất cần thiết! Một chương trình nhỏ, một sub function... của nhiều cá nhân sẽ tạo thành một thư viện lớn để cộng đồng sử dụng có thể nhanh chóng kế thừa và phát huy các ứng dụng phong phú. Đây là mong muốn không phải của riêng tôi vậy!

Mọi trao đổi xin liên hệ qua:

E-mail : buitrunghieu@khvt.com

YM : khvt_sites@yahoo.com

Cell : (+84)98.3210.906

Chúc các bạn có nhiều ứng dụng thú vị với vi điều khiển P89V51RD2!

[Tài liệu tham khảo chính: Datasheet P89V51RD2 của Philips Semiconductor]