HƯỚNG DẪN THỰC HÀNH TRÊN KIT VI ĐIỀU KHIỂN AVR

Phạm Thế Duy

ptduy@yahoo.com

GIỚI THIỆU VỀ VI ĐIỀU KHIỂN AVR

AVR Microcontroller

- Là họ vi điều khiển 8 bit của Atmel.
- Tất cả các vi điều khiển AVR sử dụng chung tập lệnh và cấu trúc CPU kiểu Hardward.
- Có 32 thanh ghi đa dụng 8 bit.
- Hầu hết các lệnh thực hiện trong 1 clock.
- Được thiết kế làm việc hiệu quả với các chương trình C.

LÀM VIỆC HIỆU QUẢ VỚI NGÔN NGỮ C

- Xem đoạn mã lệnh C sau:

```
int max(int *array)
{
  char a;
  int maximum = -32768;
  for (a = 0; a < 16; a++)
    if (array[a] > maximum)
      maximum = array[a];
  return (maximum);
}
```

KẾT QUẢ THỬ NGHIỆM CỦA ATMEL

MCU	ĐỘ DÀI BIÊN DỊCH (BYTE)	THỜI GIAN THỰC HIỆN (CHU KỲ)
AVR	46	335
8051	112	9,384
PIC16C74	87	2,492
68HC11	57	5,244

TỐC ĐỘ AVR

KHI CHẠY CÙNG MỘT TỐC ĐỘ:

- Nhanh hơn PIC16 - 07 lần.
- Nhanh hơn 68HC11 – 15 lần.
- Nhanh hơn 8051 – 28 lần.

AVR Microcontroller

- Bao gồm nhiều loại vi điều khiển với các đặc tính khác nhau như:
  - Kích thước bộ nhớ chương trình.
  - Kích thước bộ nhớ EEPROM.
  - Số chân vào ra.
  - Số các khối chức năng.
- AVR nhỏ nhất là ATTiny11 với 1KB flash ROM, 0 byte RAM và 6 I/O.
- AVR lớn nhất là ATMega128 với 128KB flash ROM, 4KB RAM, 53 I/O và rất nhiều khối chức năng.

CÁC LOẠI AVR

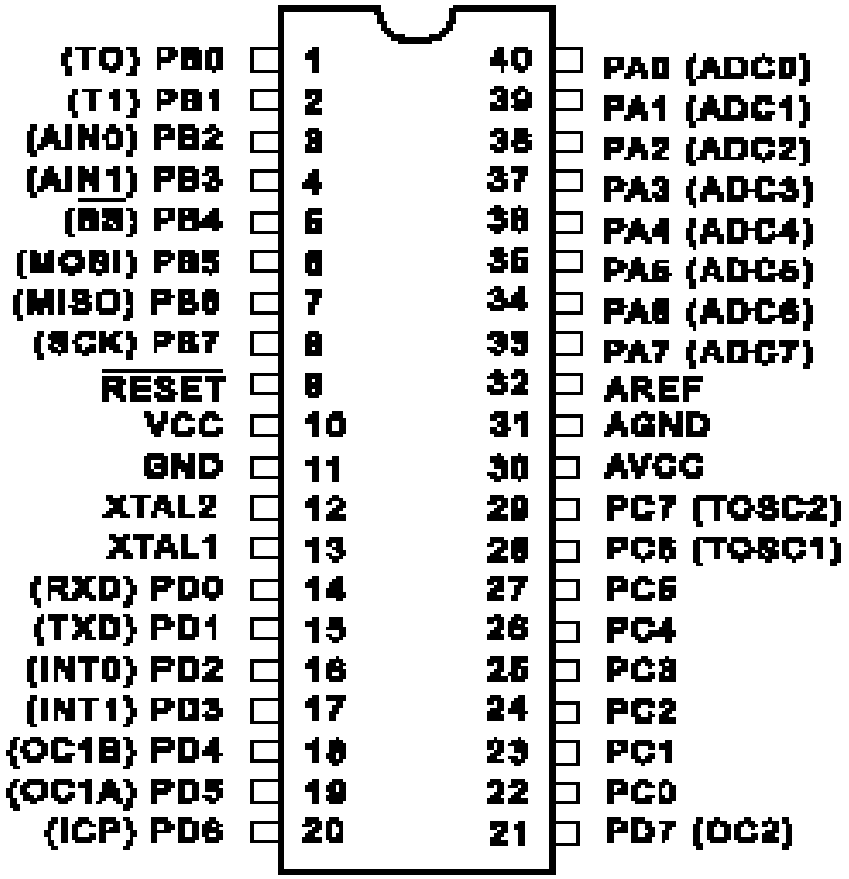
- Có 03 họ AVR cơ bản:
  - Họ AVR gốc là: AT90xxxx.
  - Họ ATMega cho các ứng dụng phức tạp.
  - Cho các ứng dụng nhỏ co họ Attiny.
- Họ FPGA với lõi AVR.
- AVR hỗ trợ giao tiếp USB:AT43USB355 …
- AVR hỗ trợ truyền vô tuyến:AT86RF401 …
- AVR hỗ trợ BUS mạng công nghiệp CAN.

Part Number	Pins	Flash	EEPROM	RAM
90S1200	20	1K	64 Bytes	0
90S2313	20	2K	128	128
90S2323	8	2K	128	128
90S2333	28	2K	128	128
90S4433	28	4K	256	128
90S4414	40	4K	256	256
90S8515	40	8K	512	512
90S4434	40	4K	256	256
90S2343	8	2K	128	128
Mega103	64	128K	4096	4096
Mega603	64	64K	2048	4096
Tiny10	8	1K	64	0
Tiny12	8	1K	64	0
Tiny13	8	2K	128	128

CẤU TRÚC AT90S8535

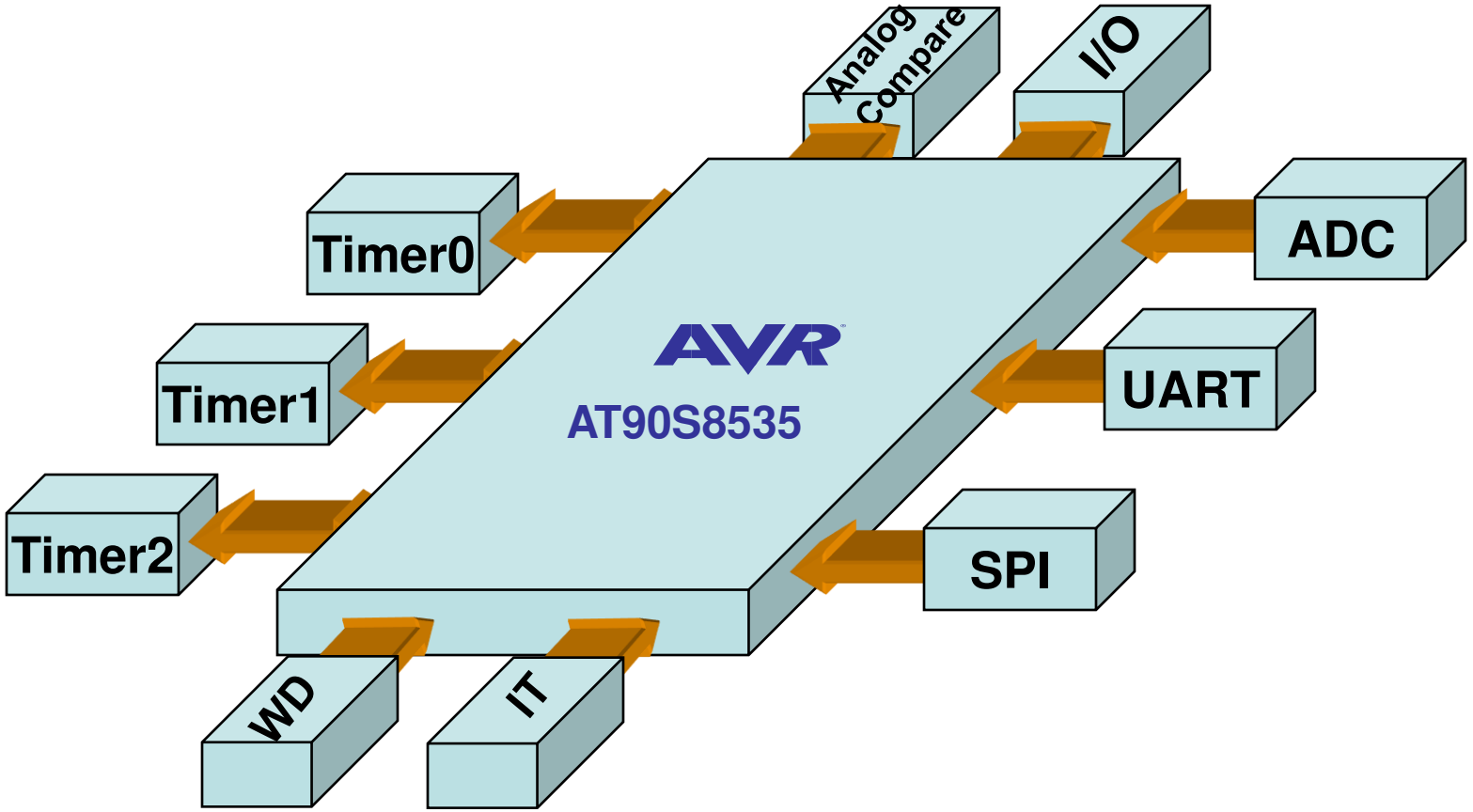
- Giới thiệu chung về vi điều khiển.
- Vi điều khiển AVR.
- **Vi điều khiển AVR AT90S8535.**

AVR AT90S8535



- HARVARD,RISK Architecture
- Flash 8K Bytes
- EEPROM 512 Bytes
- SRAM 512 Bytes
- ADC 8-channel / 10-bit
- UART programmable
- Timer/Counter 2(8-bit) 1(16-bit)
- Watchdog Timer programmable
- Analog Comparator on chip
- SPI on chip
- Power-on Reset Circuit
- Real Time Clock (RTC)
- External and Internal Interrupt sources
- 3-Sleep Modes
- Operating Voltage 4.0-6.0V
- Speed grades 0-8MHz
- 40-pinPDIP package

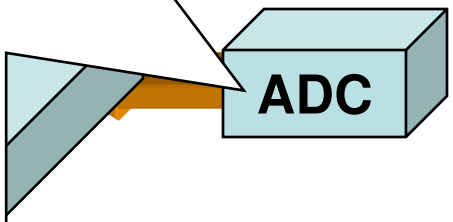
AVR AT90S8535



AVR AT90S8535

ADC

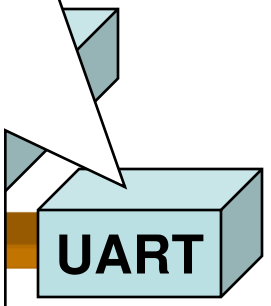
- 8 kênh ngõ vào multiplex.
- Các chế độ biến đổi 8 hoặc 10 bit.
- Độ chính xác ± LSB
- Thời gian biến đổi 65 - 260µsec
- Có hai chế độ biến đổi Free run và single.
- Có ngắt báo kết thúc quá trình biến đổi.
- Có chế độ ngủ chống nhiễu.



AVR AT90S8535

UART

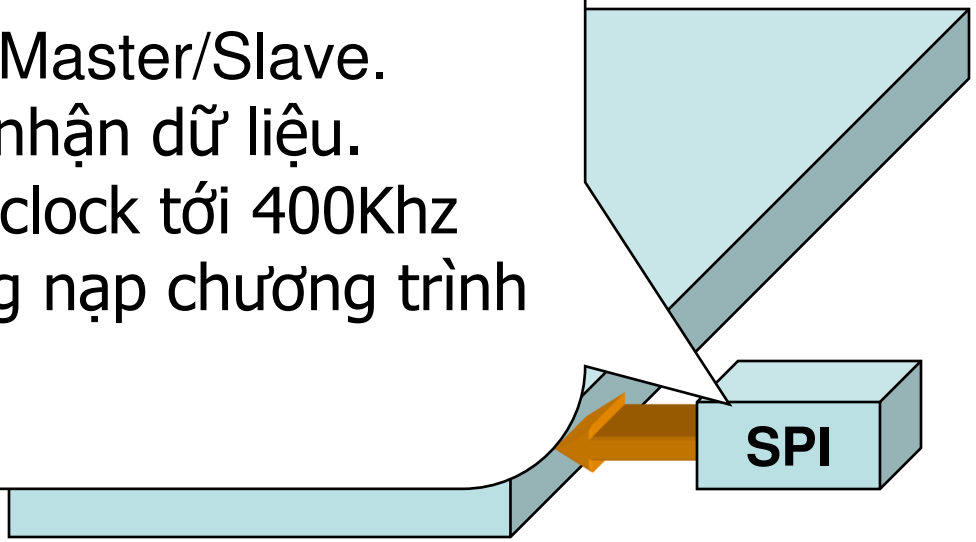
- Truyền nhận song công
- 8 hoặc 9 bit dữ liệu
- Phát hiện sai khung
- Báo lỗi bit start
- Có khả năng chống nhiễu
- Tốc độ truyền cao với tần số thạch anh thấp
- Có thể truyền với nhiều tốc độ
- Có chế độ truyền đa xử lý
- Có 3 nguồn ngắt riêng



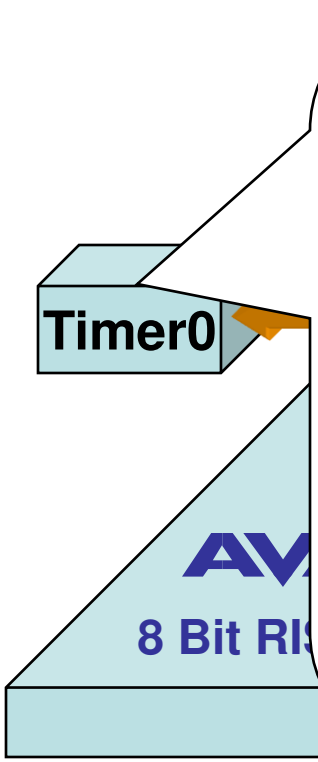
AVR AT90S8535

SPI

- Chế độ Master/Slave.
- Truyền nhận dữ liệu.
- Tốc độ clock tới 400Khz
- Sử dụng nạp chương trình



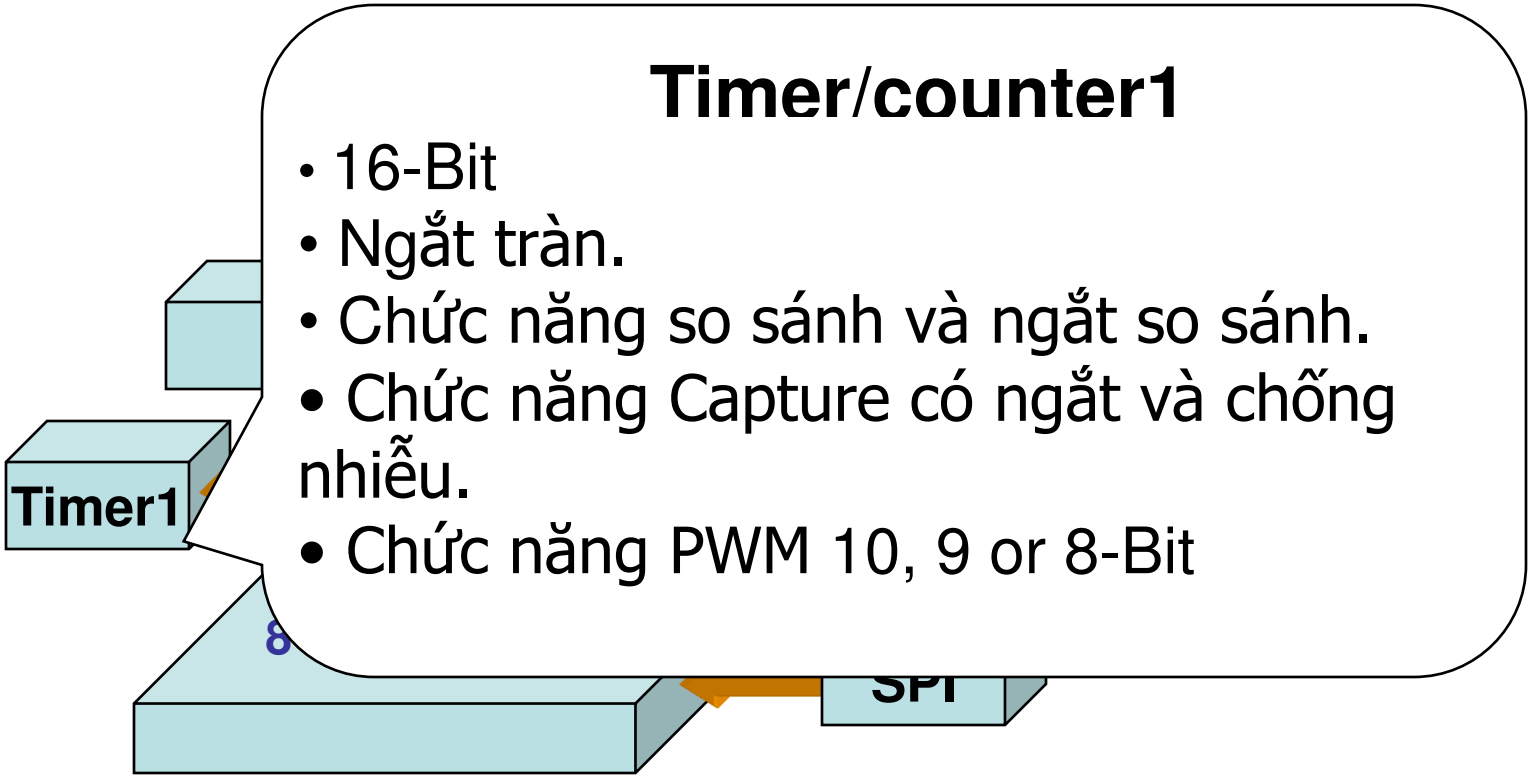
AVR AT90S8535



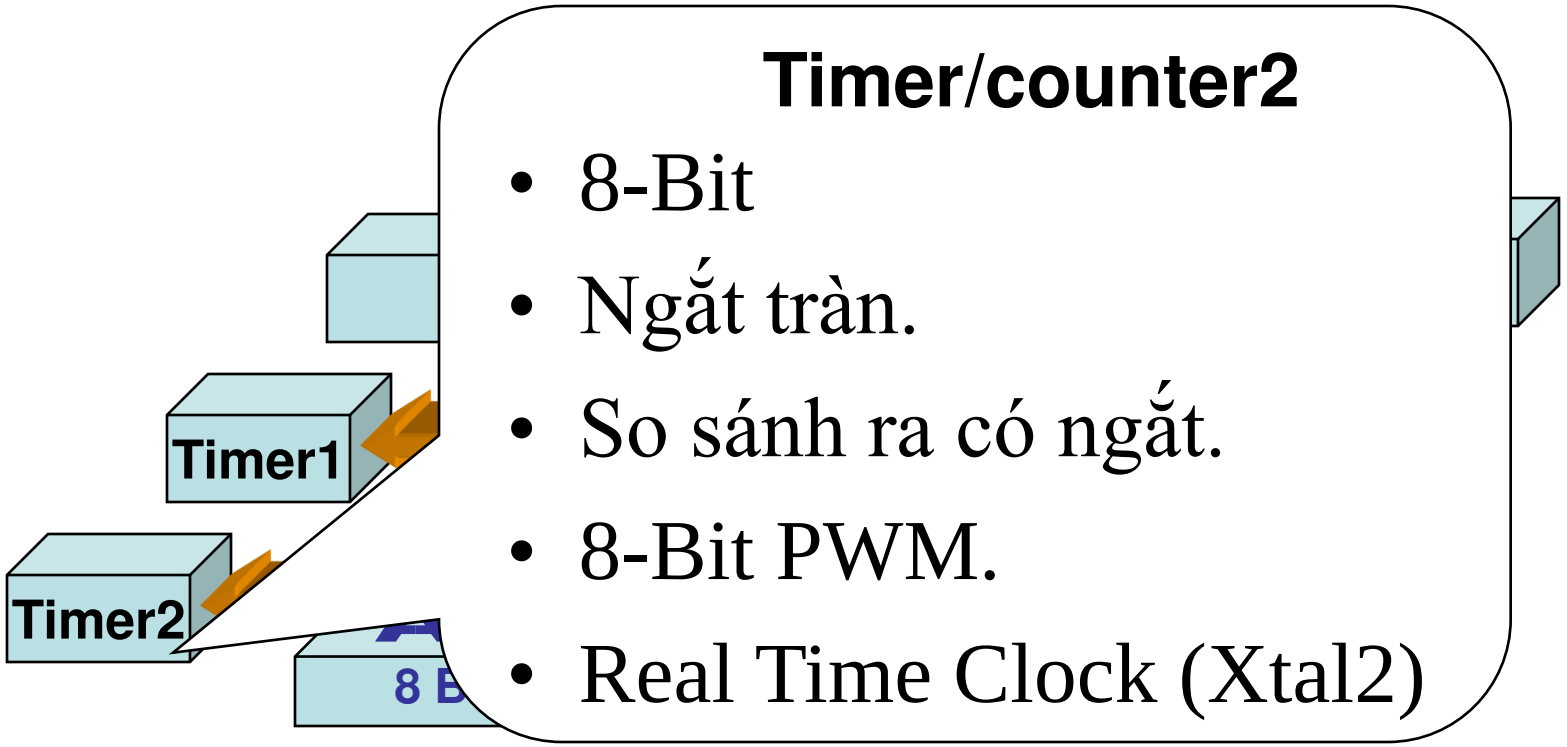
Timer/counter0

- 8-Bit
- Ngắt tràn
- Có chức năng tràn có ngắt tràn
- 8-Bit PWM.

AVR AT90S8535



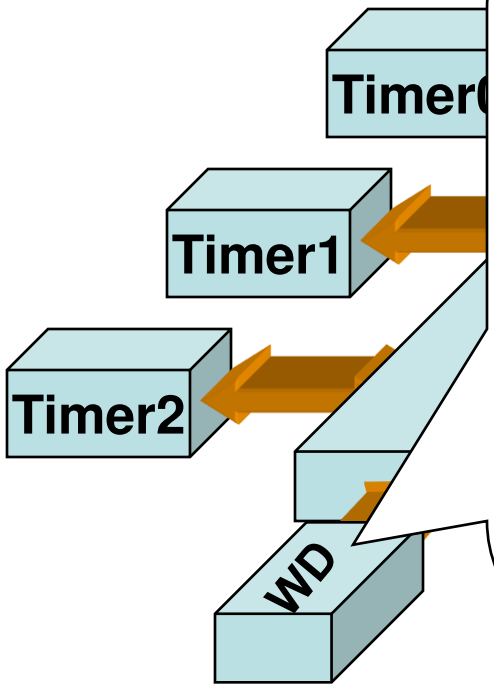
AVR AT90S8535



Timer/counter2

- 8-Bit
- Ngắt tràn.
- So sánh ra có ngắt.
- 8-Bit PWM.
- Real Time Clock (Xtal2)

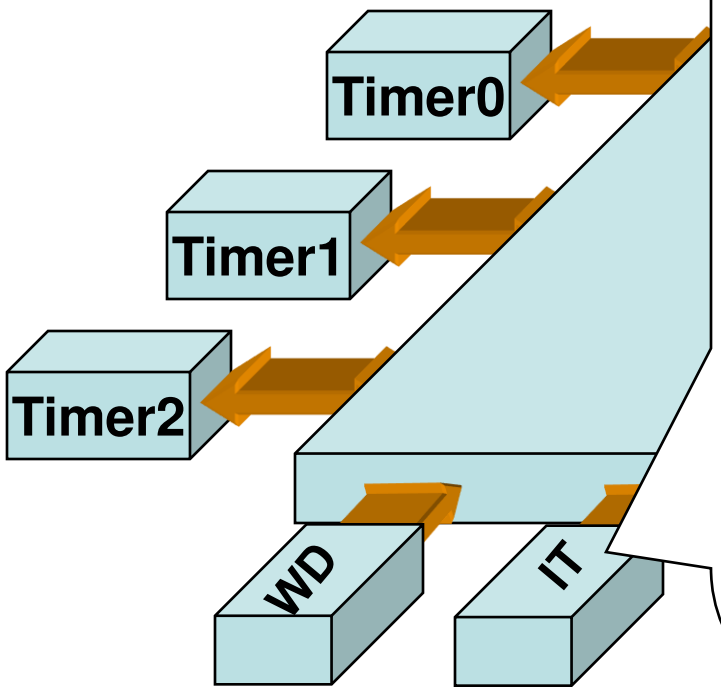
AVR AT90S8535



Watchdog/Timer

- Cấp clock từ bộ dao động RC 1 Mhz bên trong.
- Chỉnh Time-Out 47ms - 6s.
- Reset bằng lệnh “WDR”.

AVR AT90S8535



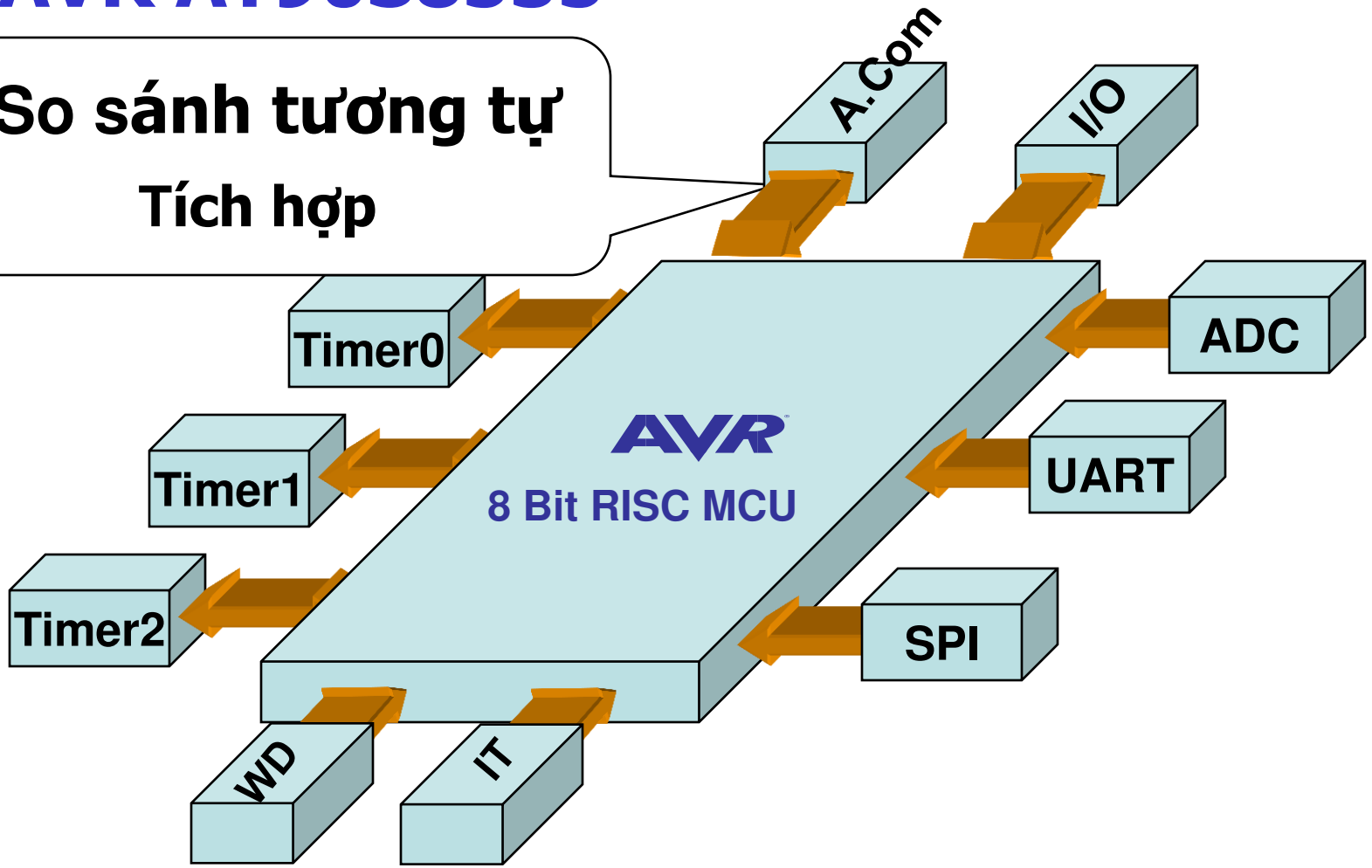
2 Nguồn ngắt ngoài

- Thời gian đáp ứng ngắn (4 Clock + RJMP)
- Tự động xoá cờ ngắt.
- Tự động cấm các chương trình ngắt bên trong.

AVR AT90S8535

So sánh tương tự

Tích hợp

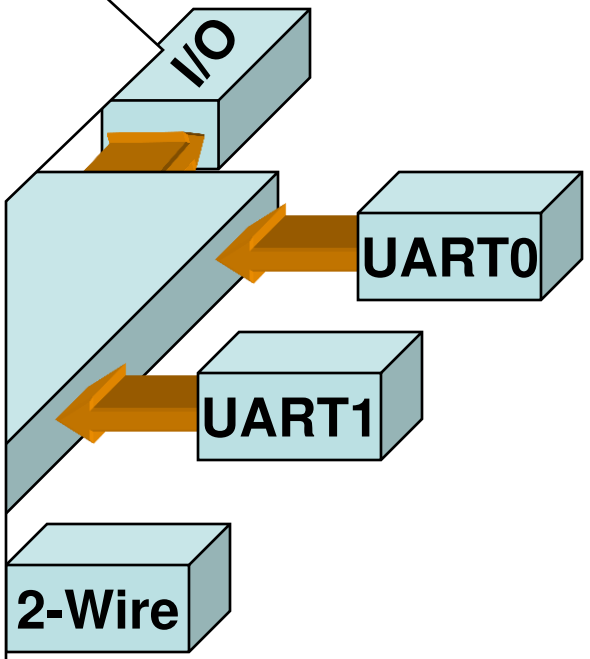


AVR AT90S8535

I/O

- 04 Cổng vào ra A,B,C,D.
- Cấp dòng (6 or 20 mA)
- Điều khiển điện trở Pull-Up
- Điều khiển 3 trạng thái cho từng ngõ vào ra.
- Mỗi cổng có 03 thanh ghi:

  DDRx, PORTx, and PINx



CÁC VẤN ĐỀ CHÍNH TRONG CÁC BÀI THỰC HÀNH

- Cổng giao tiếp song song.
- Cổng giao tiếp nối tiếp.
- Bộ định thời.
- ADC.
- Ngắt.

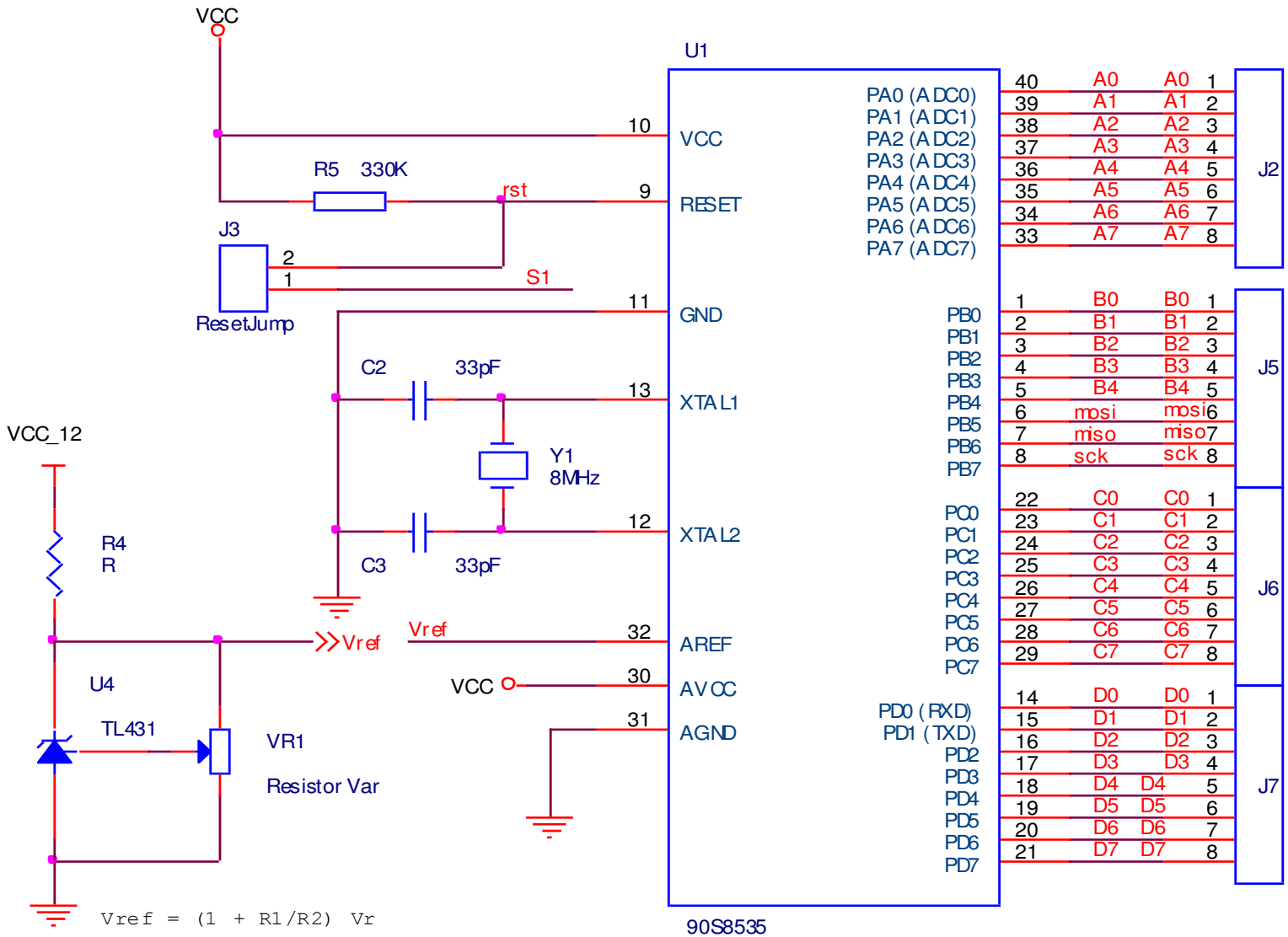
CÁC BƯỚC THỰC HIỆN BÀI THỰC HÀNH

- Thiết kế phần cứng trên Protues.
- Viết chương trình trên CodeVision AVR.
- Chạy thử hệ thống trên Protues.
- Hỏi đáp trên lớp.
- Nộp chương trình protues đã chạy trên phòng thực hành.
- Chạy thử nghiệm trên KIT của phòng thực hành.
- Viết báo cáo thực hành
- Thi trắc nghiệm thực hành.

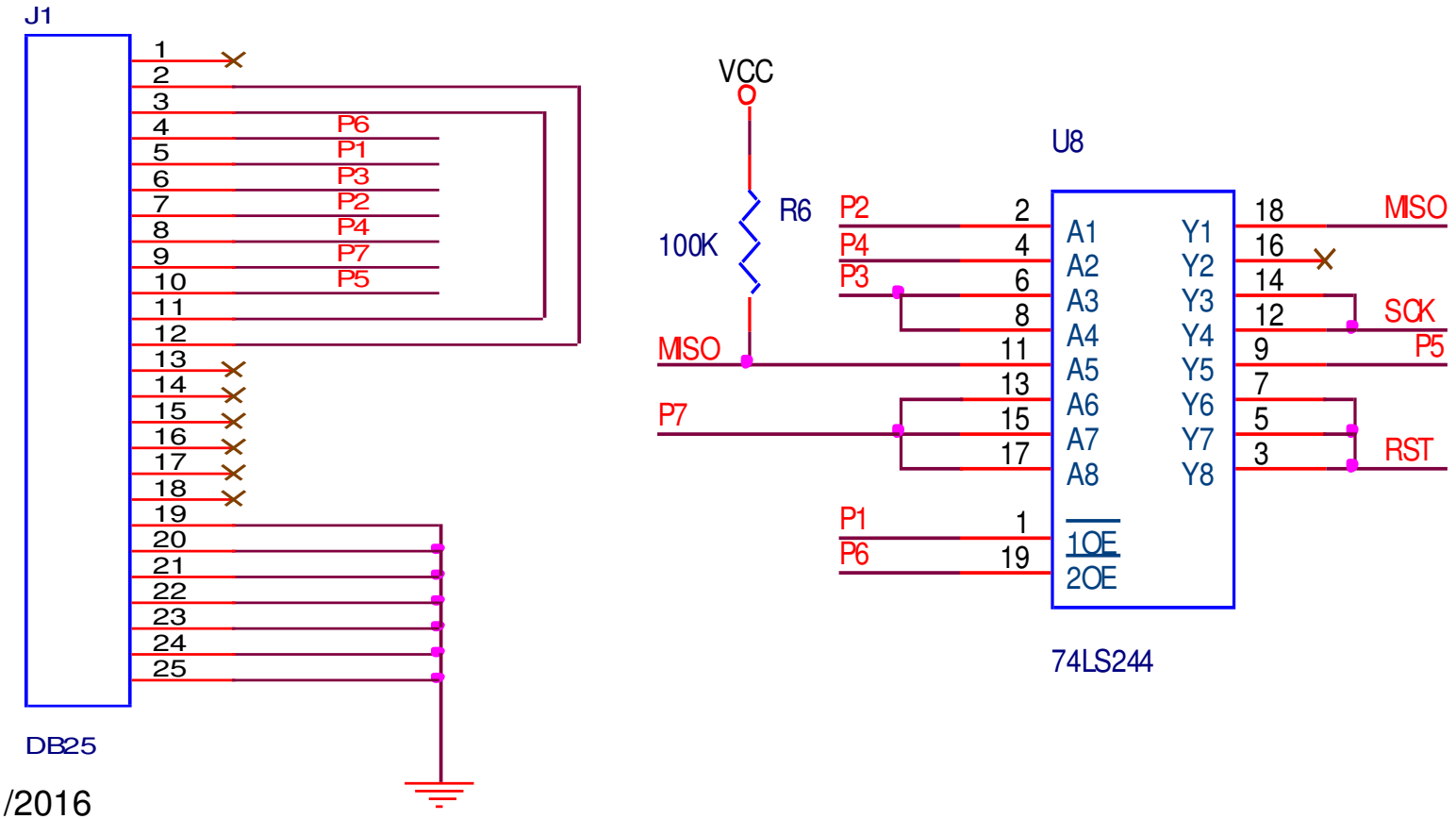
MẠCH ĐIỆN PHẦN CỨNG KIT AVR

- Mạch vi điều khiển.
- Mạch lập trình cho vi điều khiển.
- Mạch nguồn cung cấp.
- Mạch công tắc đơn.
- Mạch phím nhấn và công tắc đơn.
- Mạch hiển thị LED 7 đoạn.
- Mạch hiển thị LCD.
- Mạch biến trở cấp tín hiệu tương tự.
- Mạch tạo tín hiệu tương tự.
- Mạch đo nhiệt độ.
- Mạch giao tiếp công suất.
- Mạch giao tiếp nối tiếp.

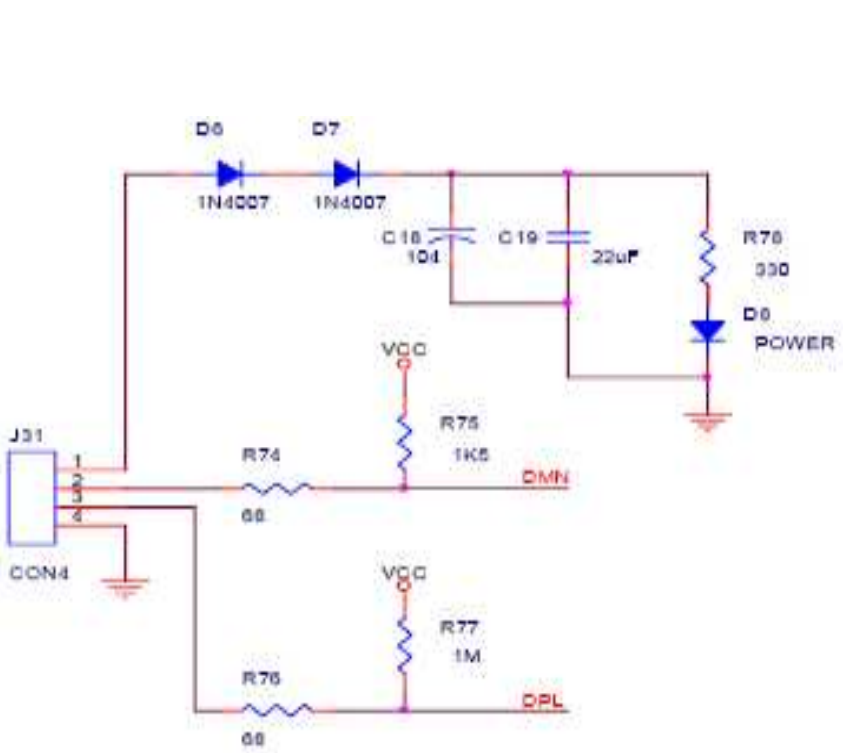
MẠCH VI ĐIỀU KHIỂN

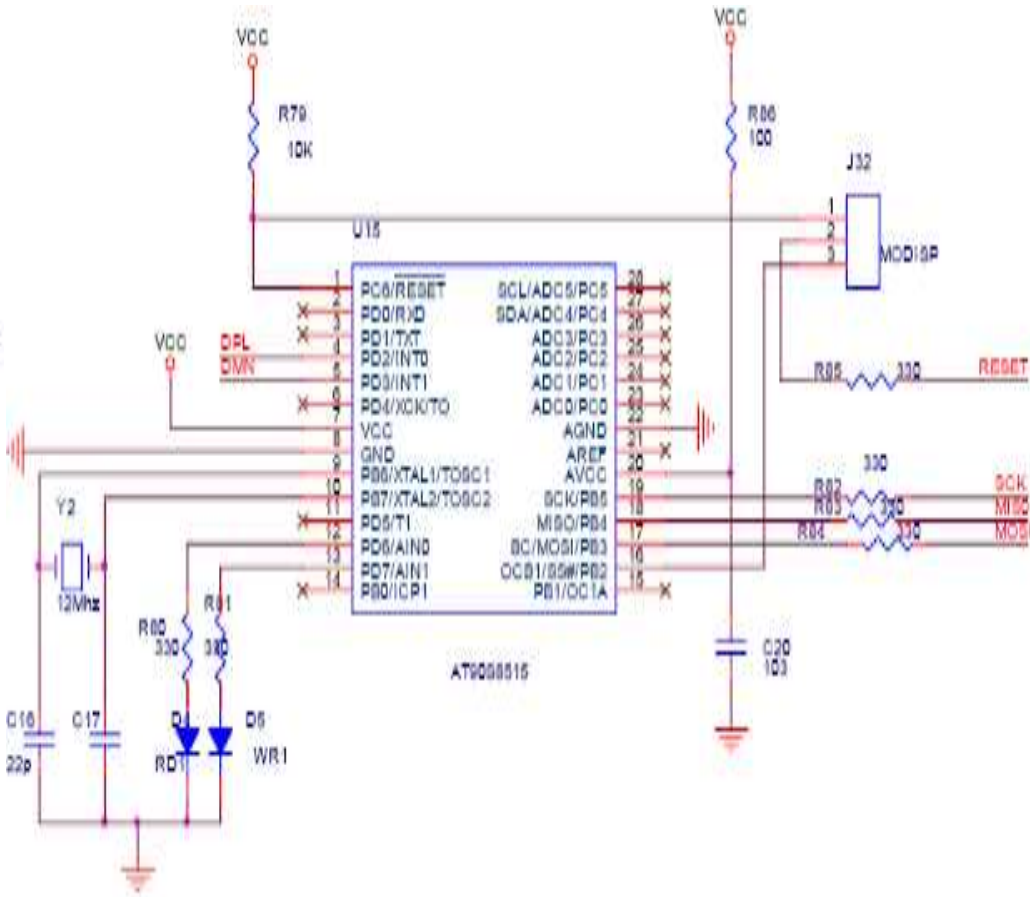


KẾT NỐI LẬP TRÌNH VI ĐIỀU KHIỂN QUA CỔNG MÁY IN

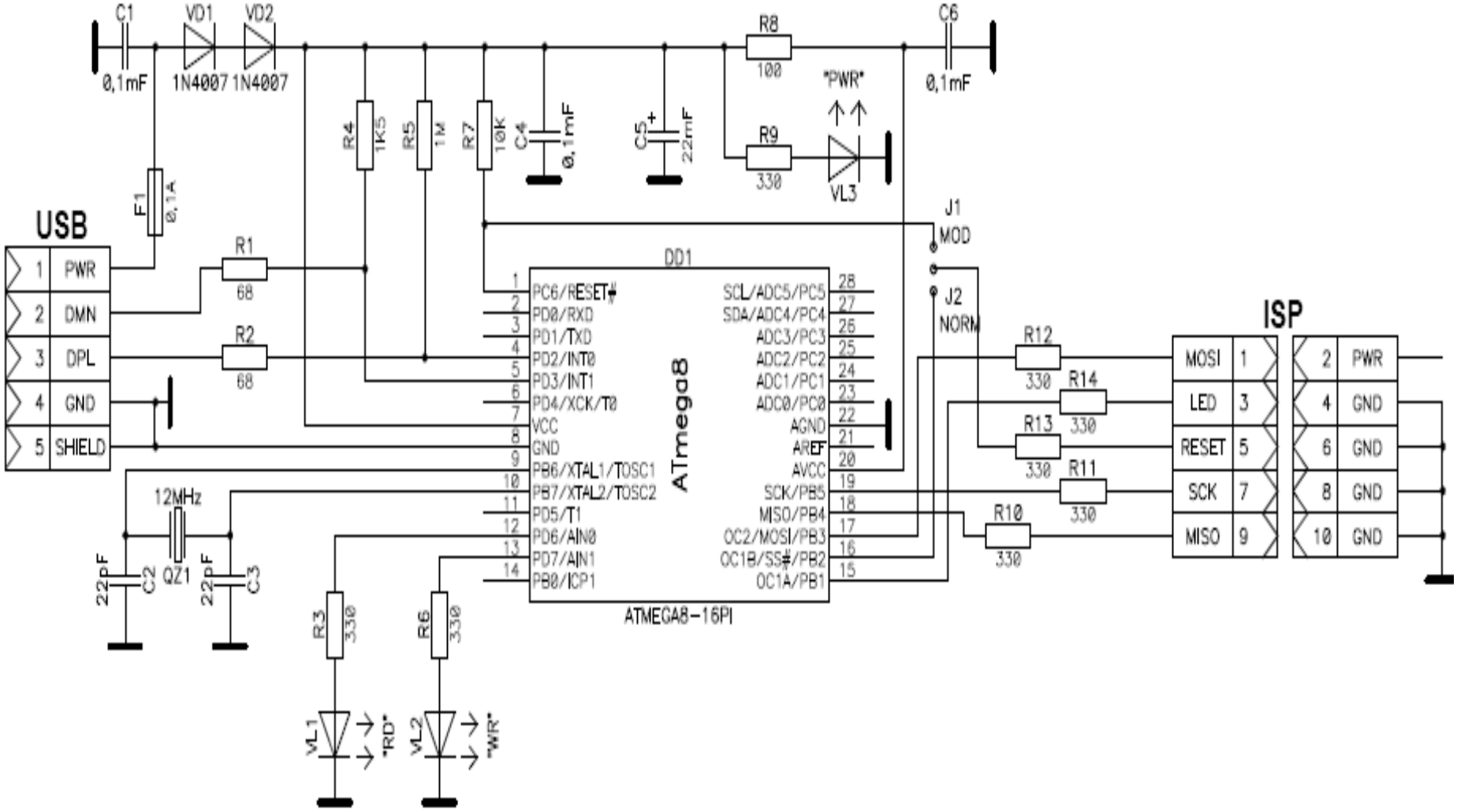


MẠCH NẠP QUA USB

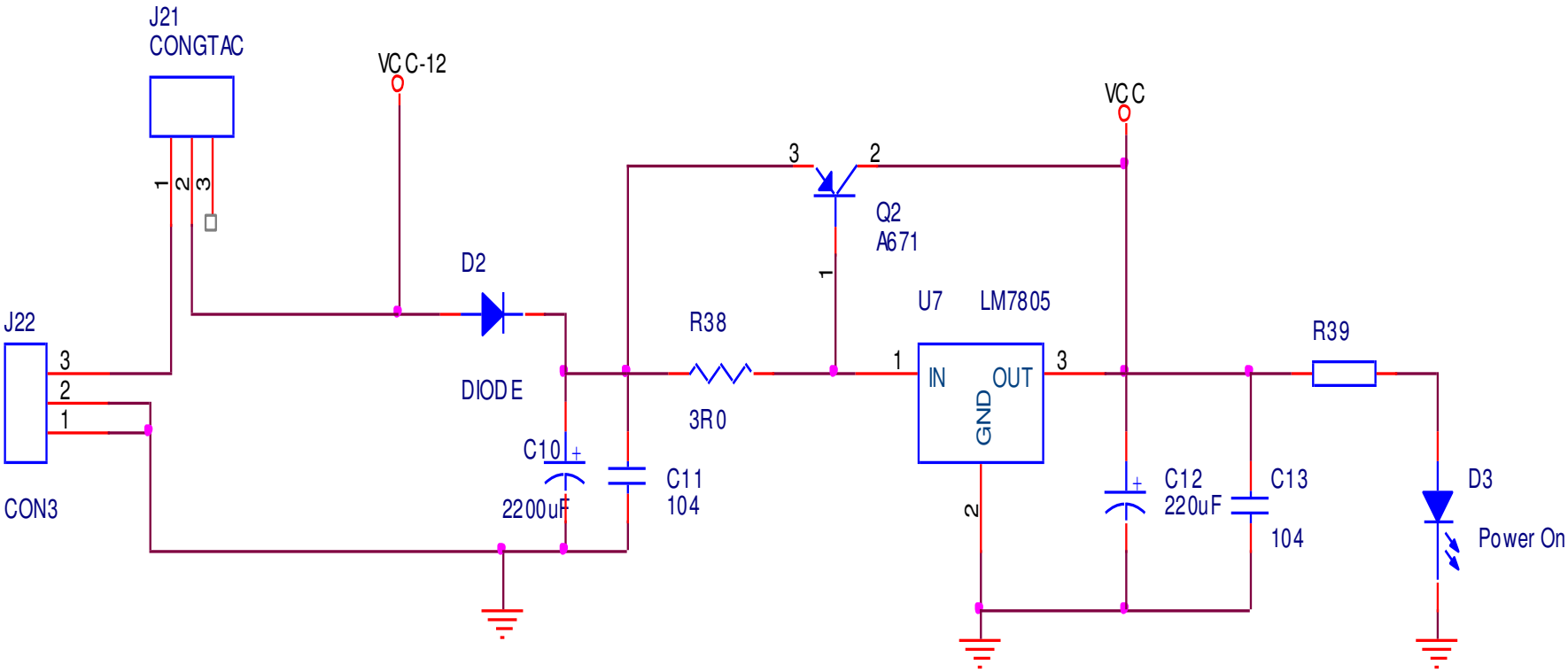




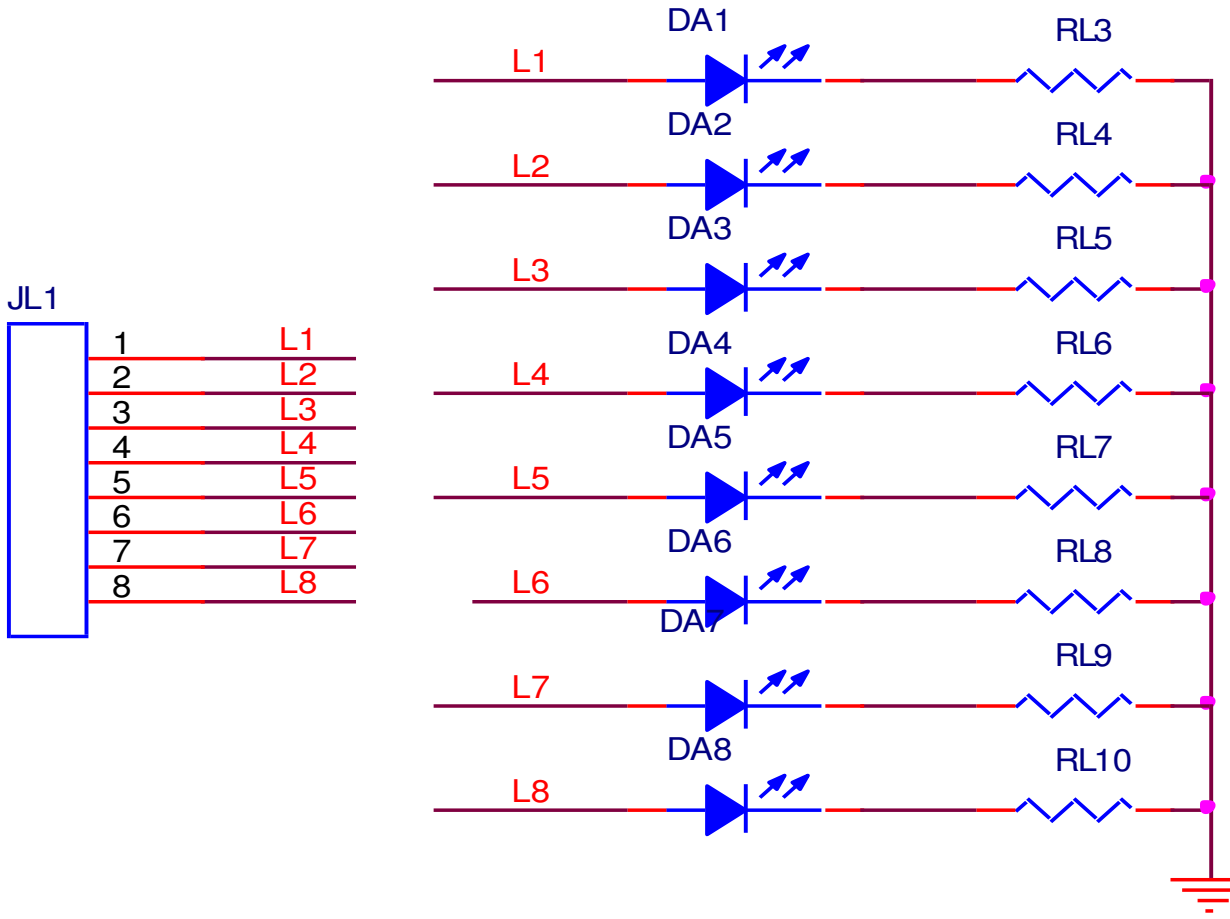
MẠCH NẠP AVR QUA CỔNG USB



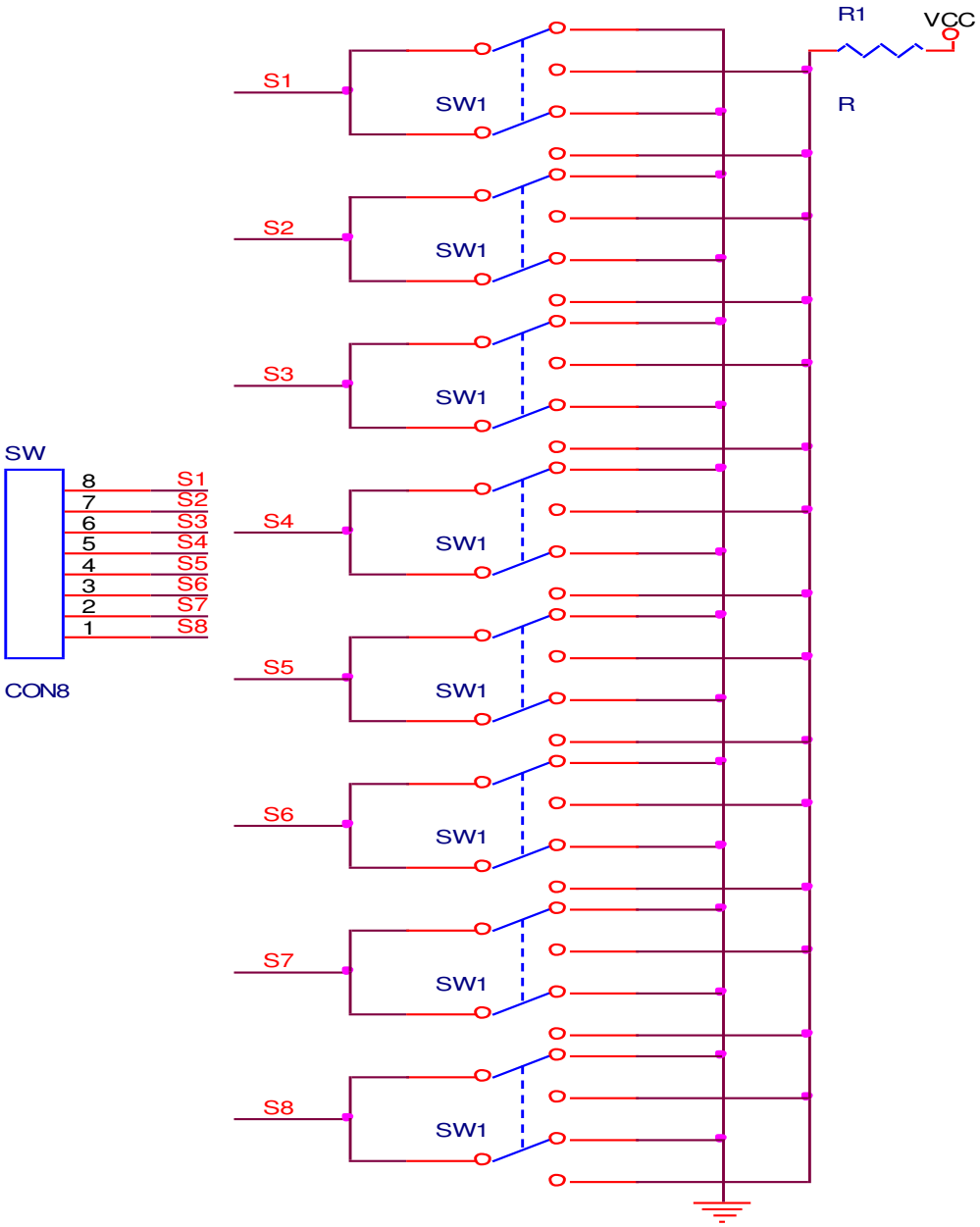
MẠCH NGUỒN CUNG CẤP



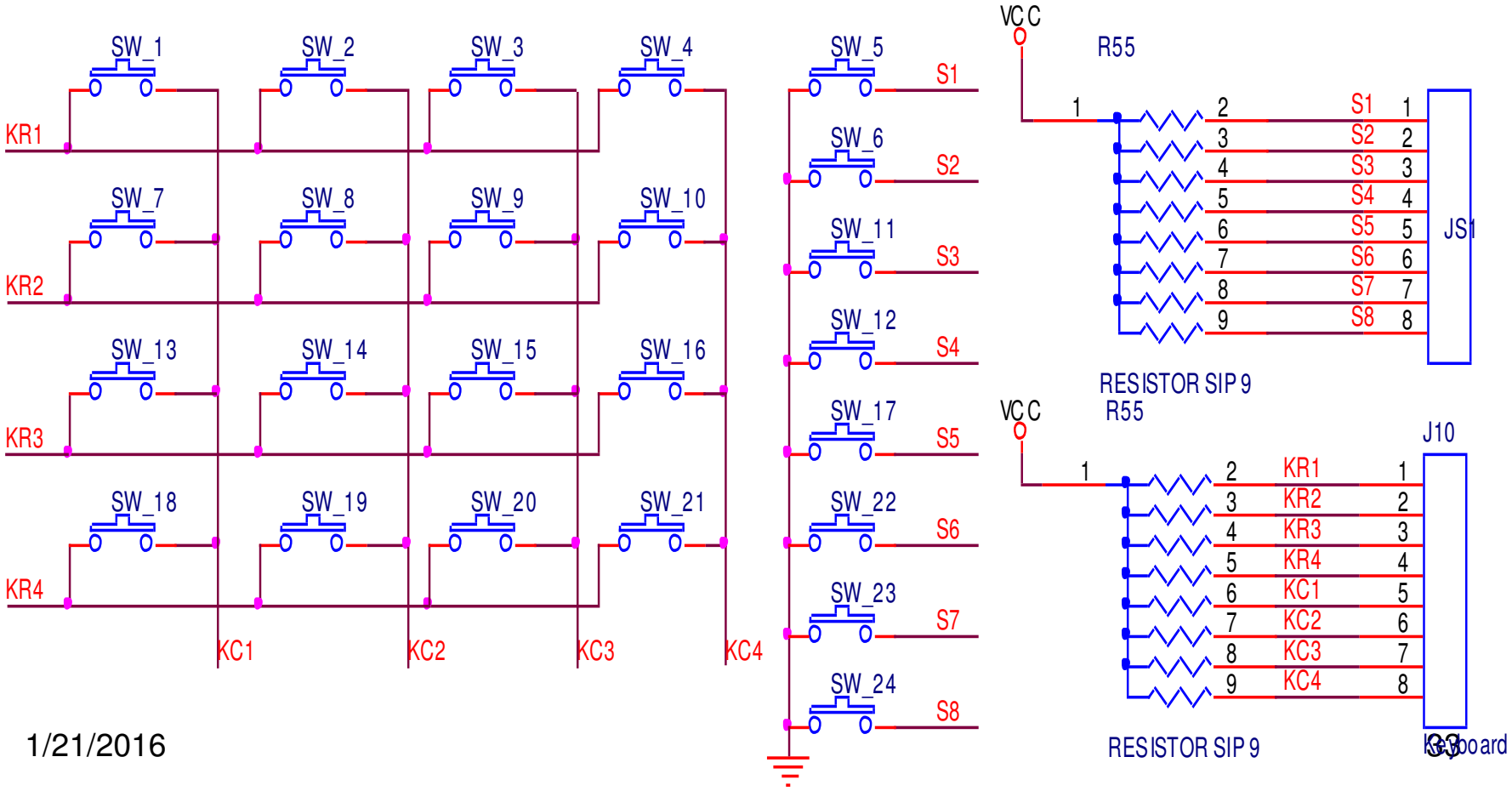
MẠCH LED ĐƠN


JL1
1
2
3
4
5
6
7
8
L1
L2
L3
L4
L5
L6
L7
L8
DA1
DA2
DA3
DA4
DA5
DA6
DA7
DA8
RL3
RL4
RL5
RL6
RL7
RL8
RL9
RL10

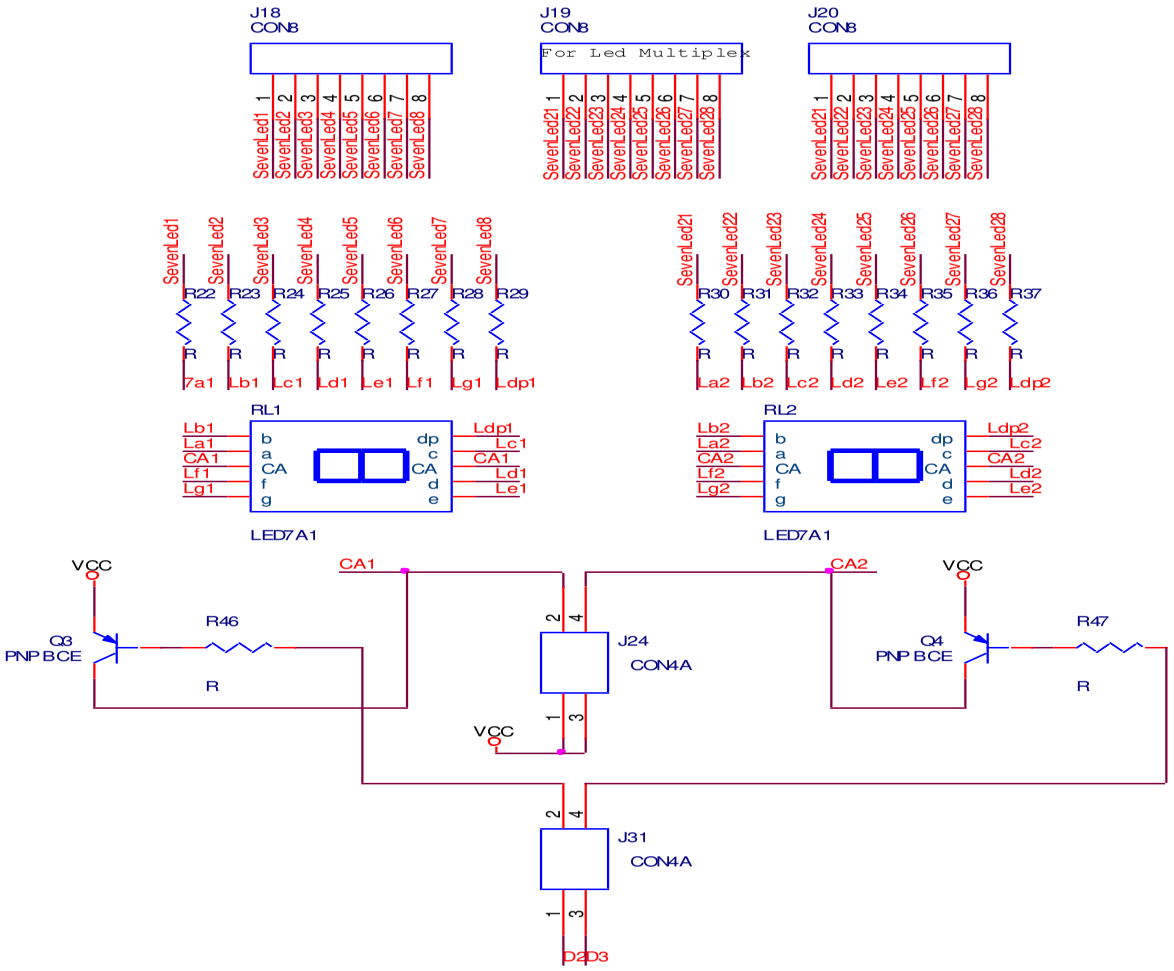
MẠCH CÔNG TẮC ĐƠN

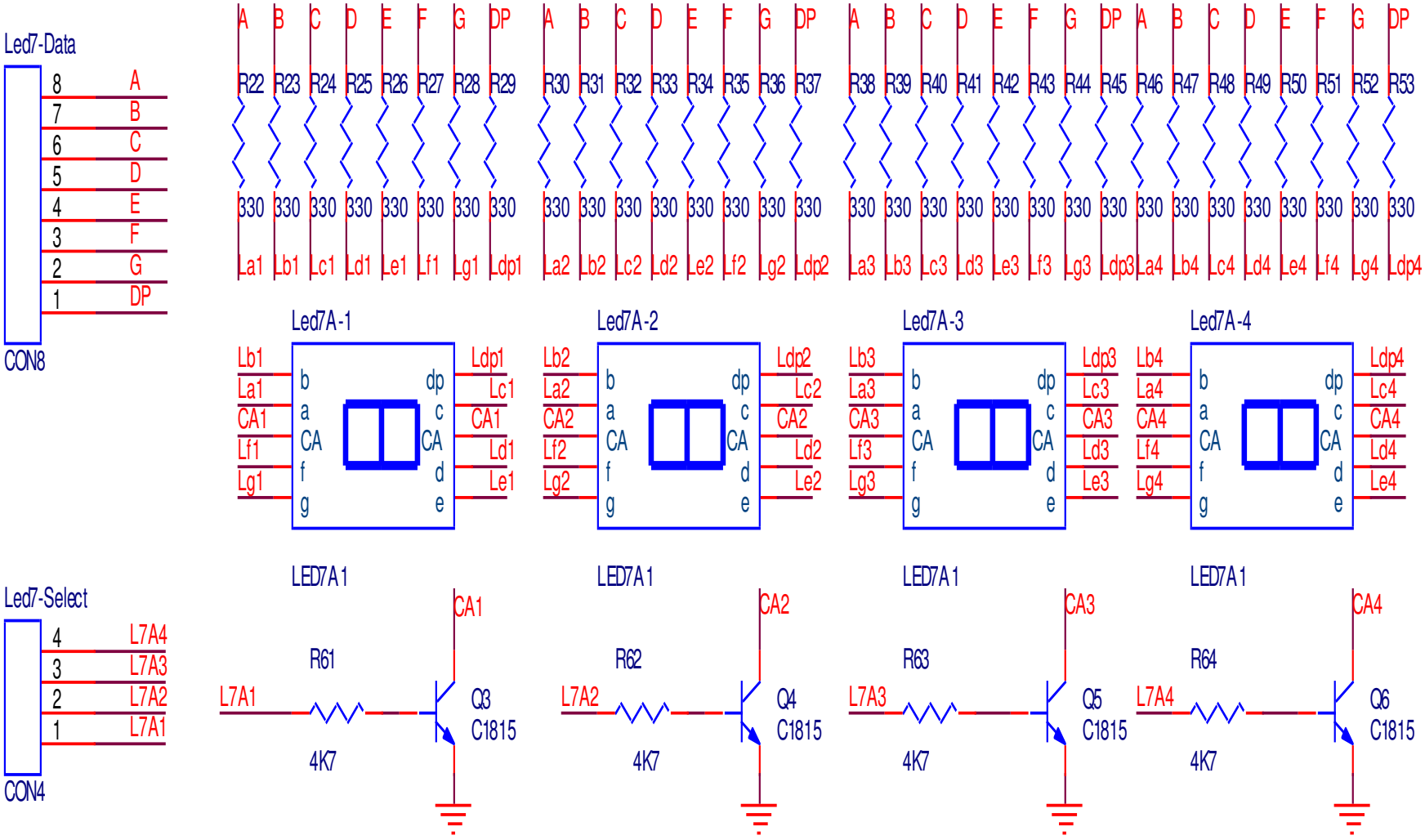


MẠCH PHÍM NHẤN

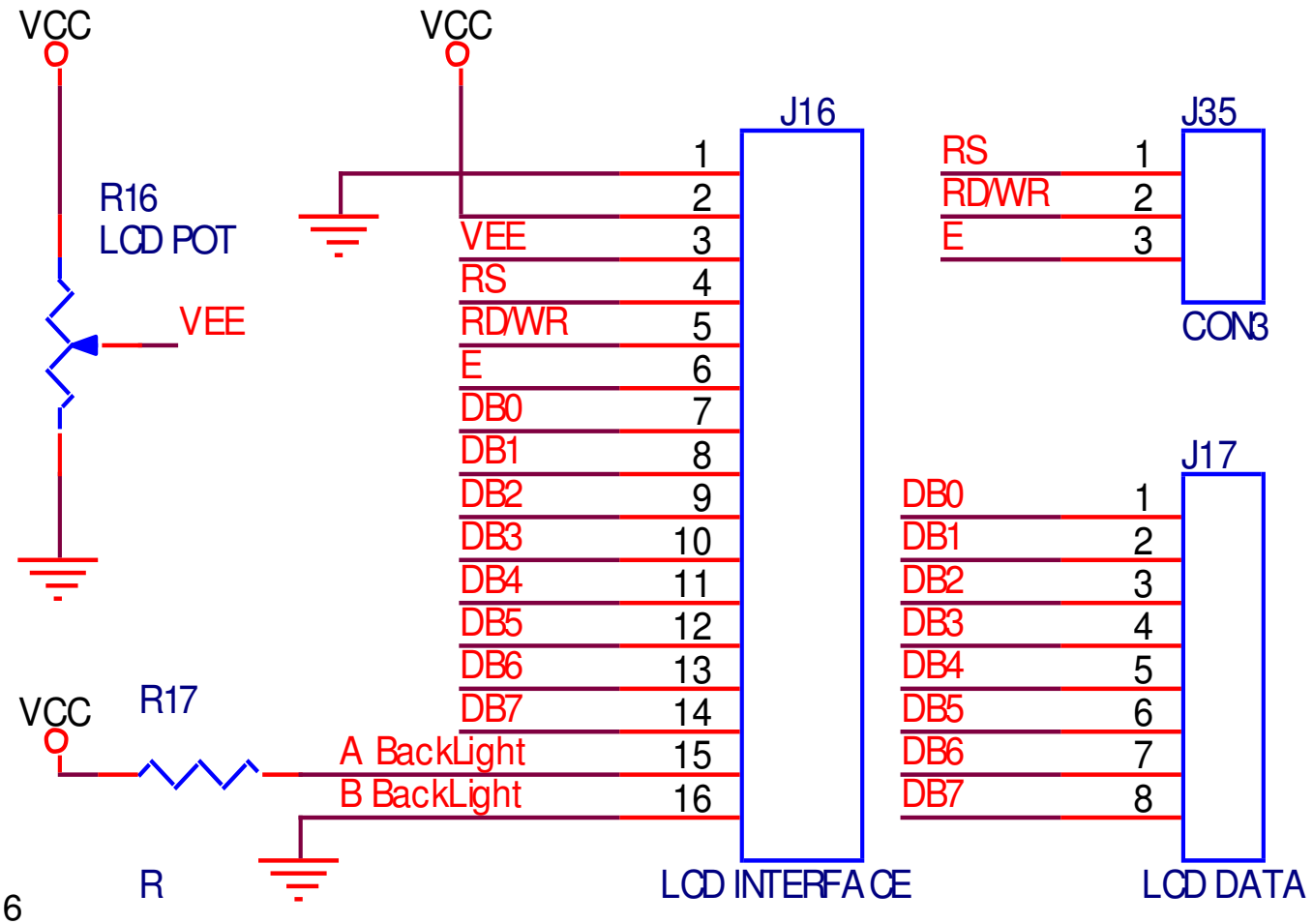


MẠCH HIỂN THỊ LED 7 ĐOẠN

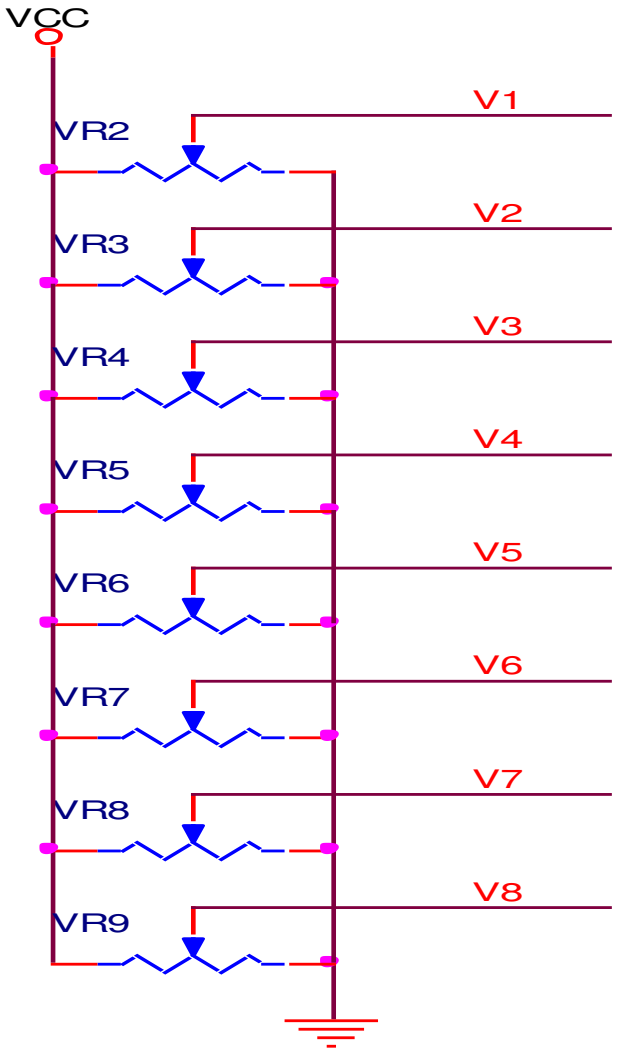



Led7-Data
8 A
7 B
6 C
5 D
4 E
3 F
2 G
1 DP
CON8
A B C D E F G DP
R22 R23 R24 R25 R26 R27 R28 R29
330 330 330 330 330 330 330 330
La1 Lb1 Lc1 Ld1 Le1 Lf1 Lg1 Ldp1
A B C D E F G DP
R30 R31 R32 R33 R34 R35 R36 R37
330 330 330 330 330 330 330 330
La2 Lb2 Lc2 Ld2 Le2 Lf2 Lg2 Ldp2
A B C D E F G DP
R38 R39 R40 R41 R42 R43 R44 R45
330 330 330 330 330 330 330 330
La3 Lb3 Lc3 Ld3 Le3 Lf3 Lg3 Ldp3
A B C D E F G DP
R46 R47 R48 R49 R50 R51 R52 R53
330 330 330 330 330 330 330 330
La4 Lb4 Lc4 Ld4 Le4 Lf4 Lg4 Ldp4
Led7A-1
Lb1 La1 CA1 Lf1 Lg1
b a CA f g
dp c CA d e
Ldp1 Lc1 CA1 Ld1 Le1
Led7A-2
Lb2 La2 CA2 Lf2 Lg2
b a CA f g
dp c CA d e
Ldp2 Lc2 CA2 Ld2 Le2
Led7A-3
Lb3 La3 CA3 Lf3 Lg3
b a CA f g
dp c CA d e
Ldp3 Lc3 CA3 Ld3 Le3
Led7A-4
Lb4 La4 CA4 Lf4 Lg4
b a CA f g
dp c CA d e
Ldp4 Lc4 CA4 Ld4 Le4
Led7-Select
4 L7A4
3 L7A3
2 L7A2
1 L7A1
CON4
LED7A1
CA1
R61
L7A1
Q3
C1815
4K7
LED7A1
CA2
R62
L7A2
Q4
C1815
4K7
LED7A1
CA3
R63
L7A3
Q5
C1815
4K7
LED7A1
CA4
R64
L7A4
Q6
C1815
4K7

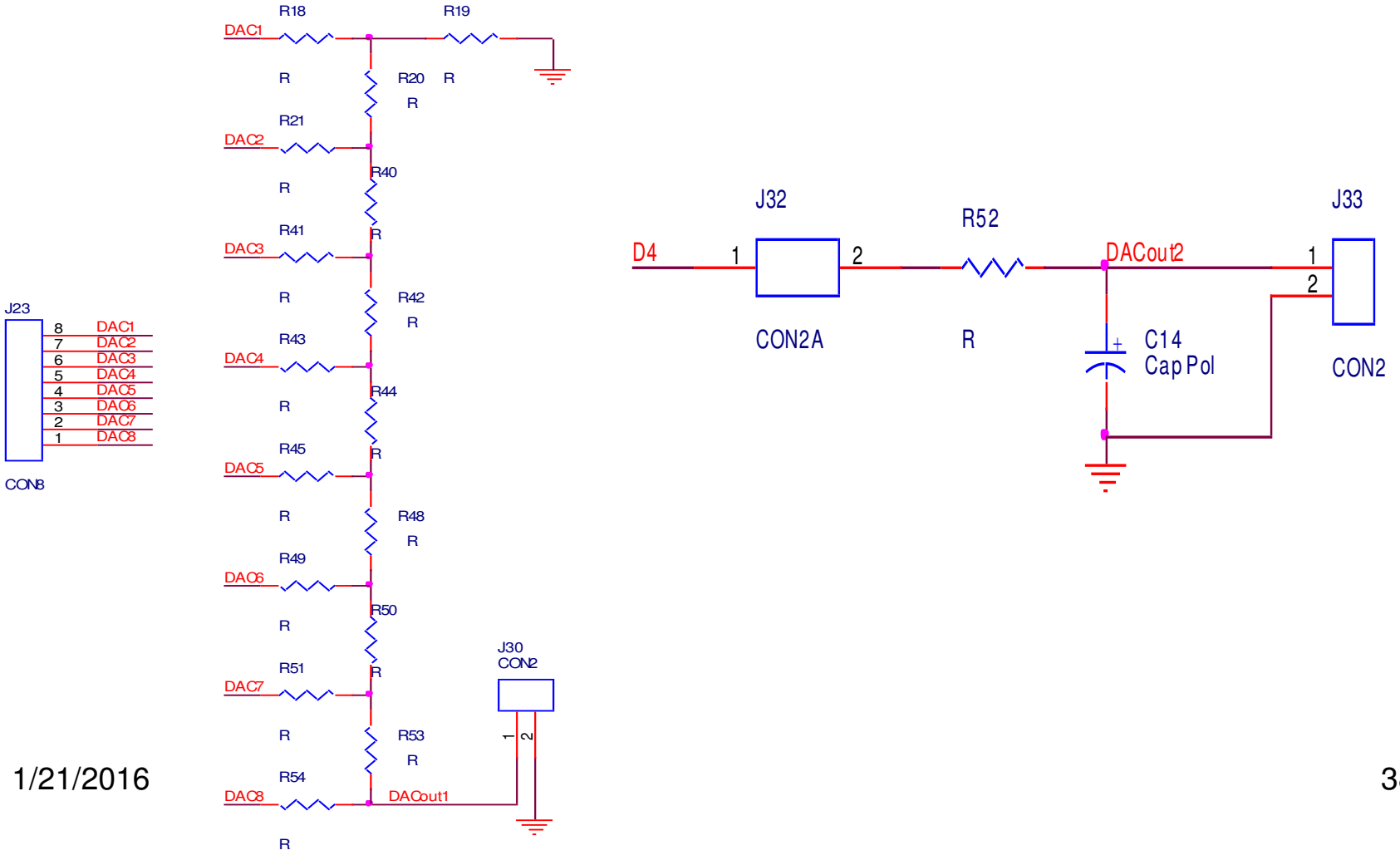
MẠCH HIỂN THỊ LCD



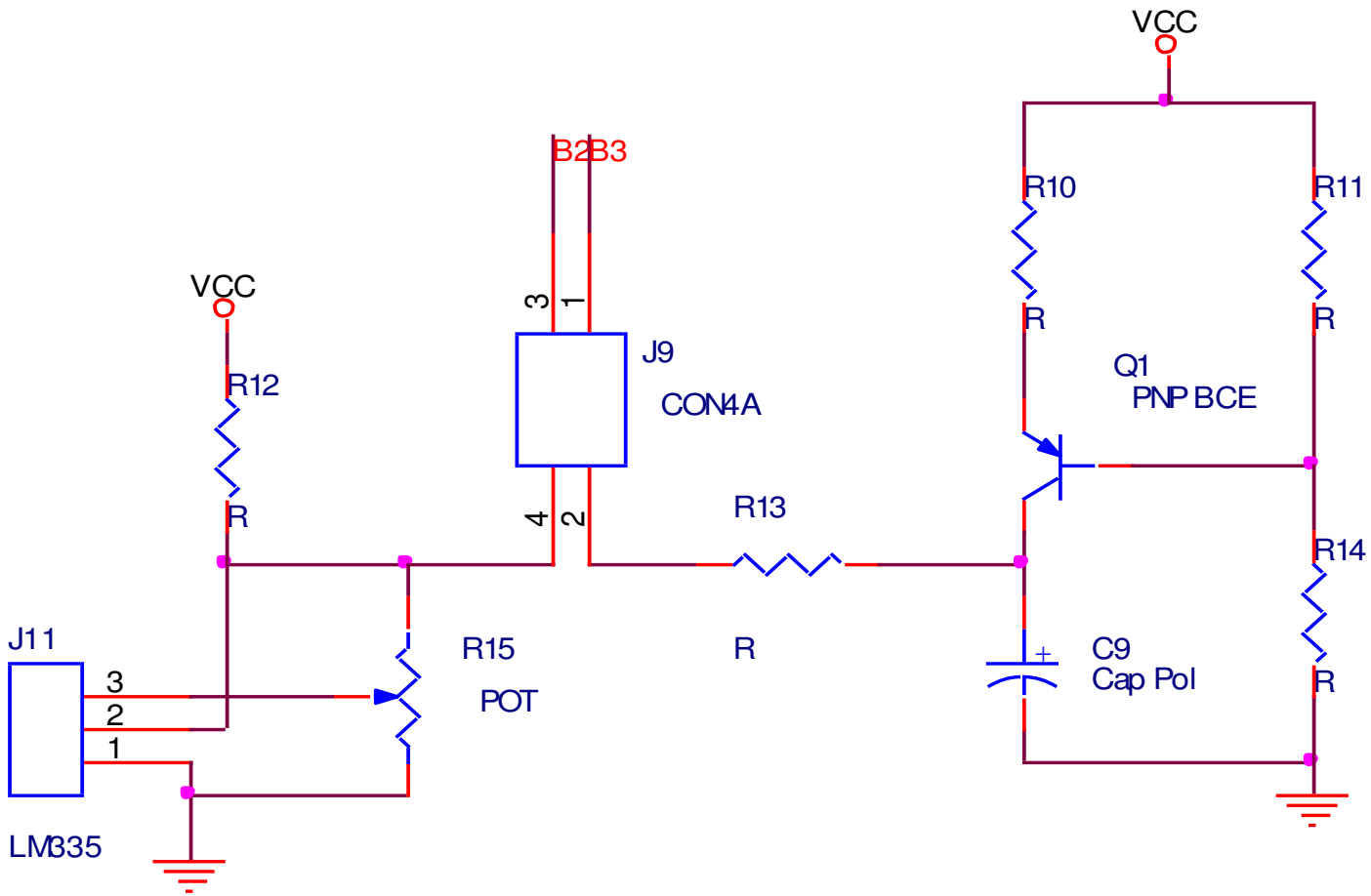
MẠCH BIẾN TRỞ CẤP TÍN HIỆU TƯƠNG TỰ



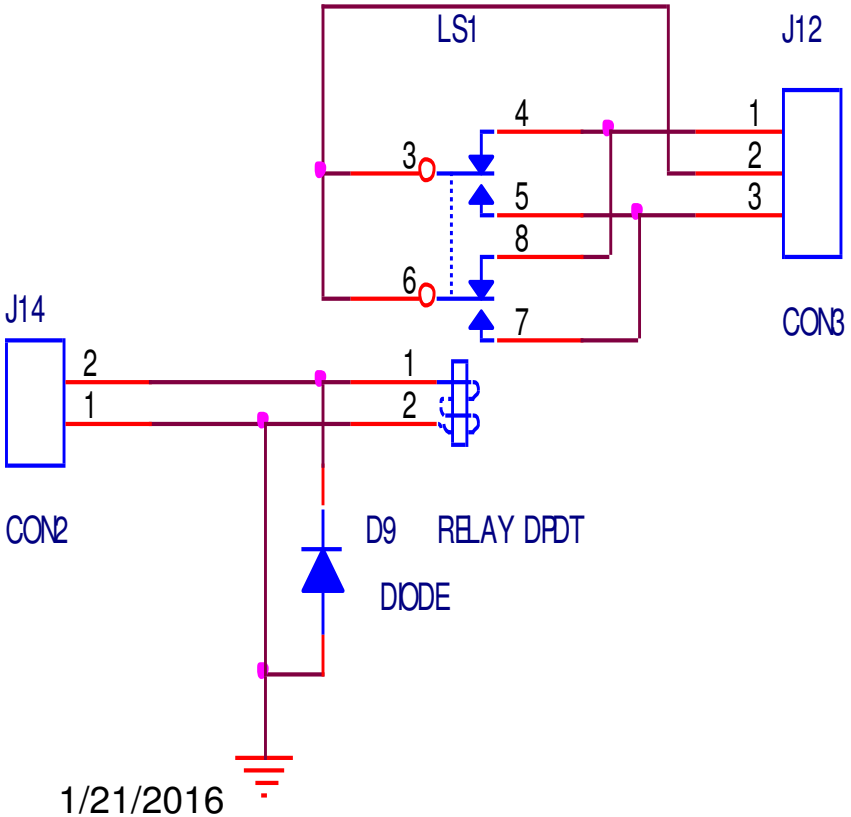
MẠCH DAC

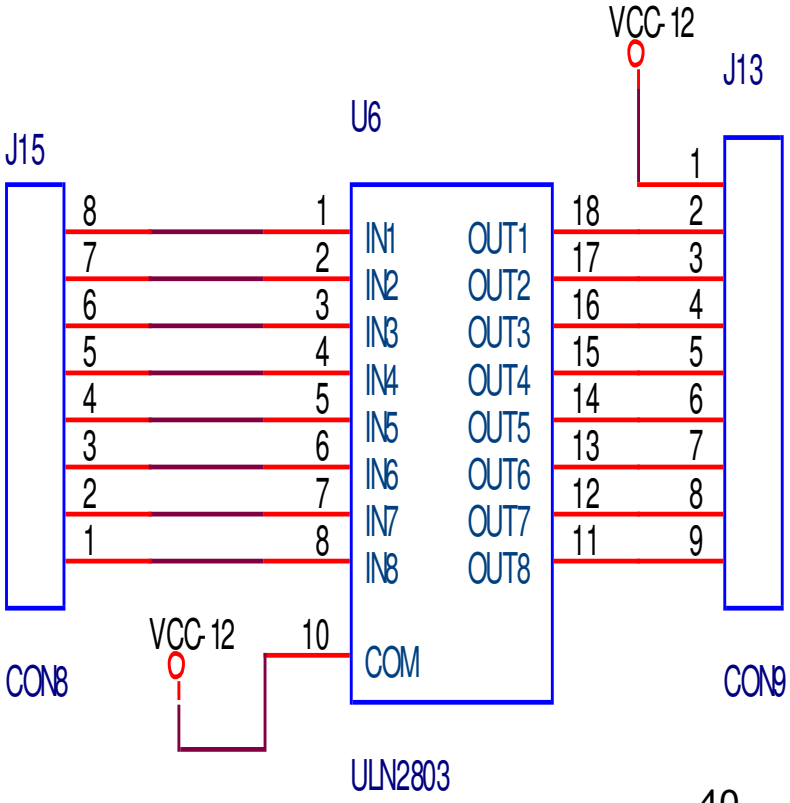

J23
8 DAC1
7 DAC2
6 DAC3
5 DAC4
4 DAC5
3 DAC6
2 DAC7
1 DAC8
CON8
R18
R19
DAC1
R
R20
R
R
R21
DAC2
R
R40
R
R41
DAC3
R
R42
R
R43
DAC4
R
R44
R
R45
DAC5
R
R48
R
R49
DAC6
R
R50
R
R51
DAC7
R
R53
R
R54
DAC8
R
DACout1
J30
CON2
1
2
J32
D4
1
2
CON2A
R52
R
DACout2
C14
Cap Pol
J33
1
2
CON2

MẠCH ĐO NHIỆT ĐỘ

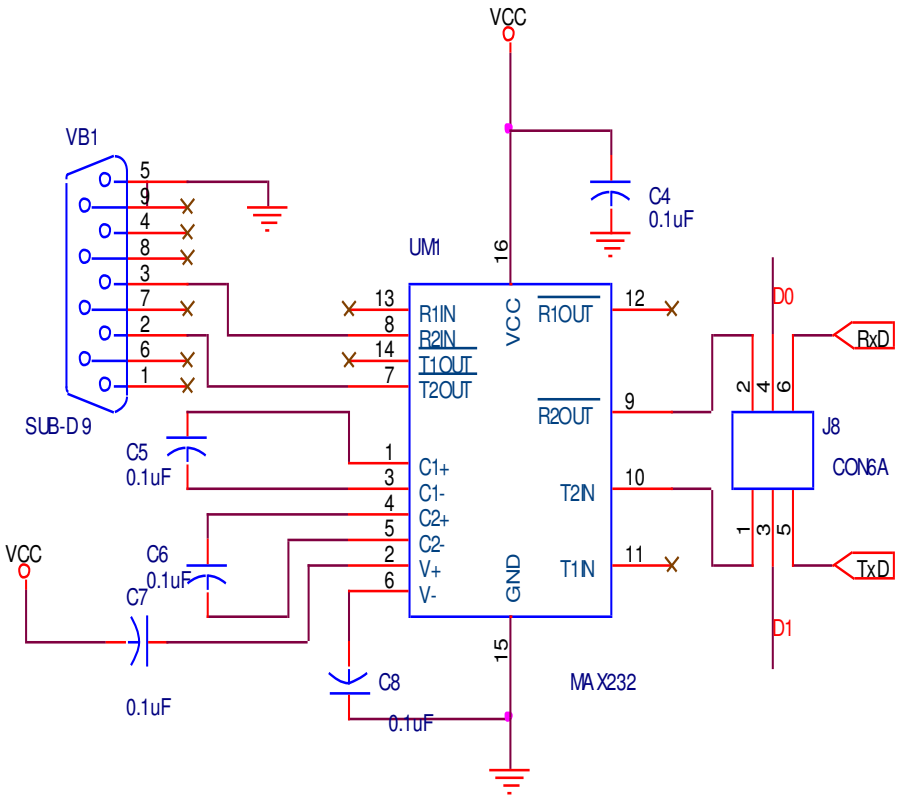


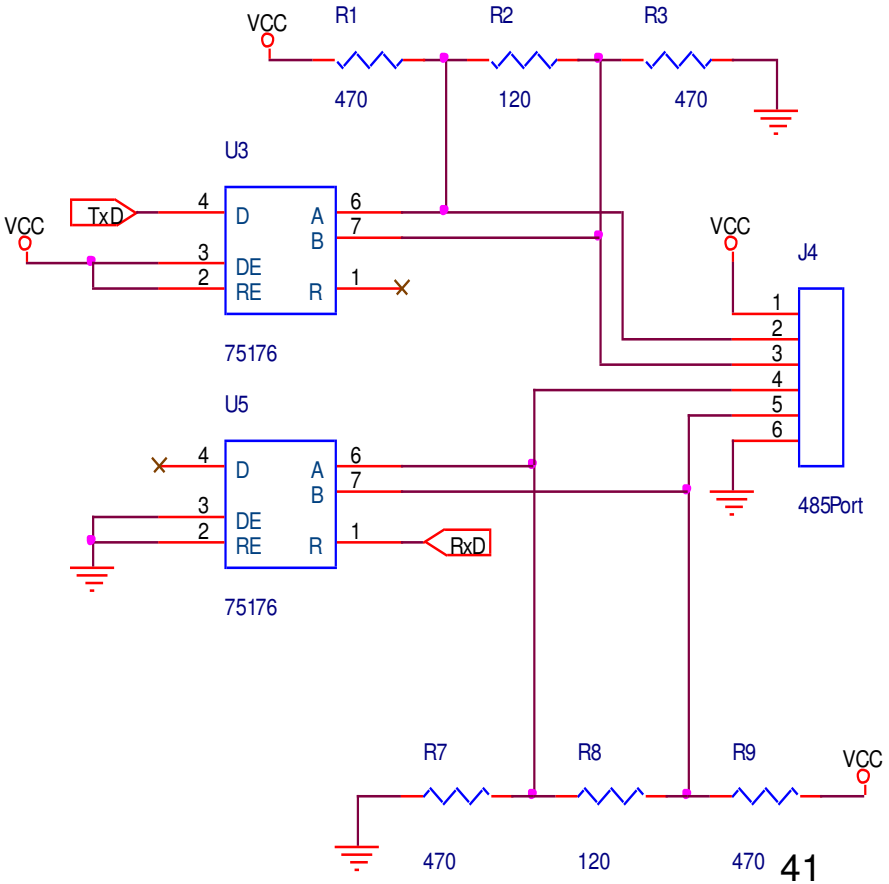
MẠCH GIAO TIẾP CÔNG SUẤT





MẠCH GIAO TIẾP NỐI TIẾP





GIỚI THIỆU VỀ CodeVision AVR

Đọc file: LaptrinhC.doc

CÁC BÀI THỰC HÀNH

- Bài 1: Điều khiển công tắc đơn và LED đơn.
- Bài 2: Điều khiển hiển thị trên LED 7 đoạn.
- Bài 3: Điều khiển bàn phím ma trận hiển thị lên LED 07 đoạn.
- Bài 4: Điều khiển hiển thị chữ trên LCD.
- Bài 5: Đọc ADC hiển thị lên LCD.
- Bài 6: Đo nhiệt độ bằng Analog Compare hiển thị trên LED 7 đoạn.
- Bài 7: Tạo xung bằng DAC.
- Bài 8: Giao tiếp nối tiếp hai KIT vi điều khiển.
- Bài 9: Thực hiện đồng hồ điện tử trên KIT vi điều khiển.
- Bài 10: Thực hiện giao tiếp với tải AC.

BÀI SỐ 1: ĐIỀU KHIỂN CÔNG TẮC ĐƠN VÀ LED ĐƠN

1. Mục đích: Sử dụng cổng vào ra song song điều khiển các thiết bị vào ra cơ bản.
2. Yêu cầu thực hiện:
    1. Thực hiện chương trình điều khiển các LED đơn chớp tắt.
    2. Thực hiện chương trình kiểm tra 8 công tắc đơn. Tương ứng với 1 công tắc ở mức 1, 8 LED sẽ sáng theo một kiểu định trước:

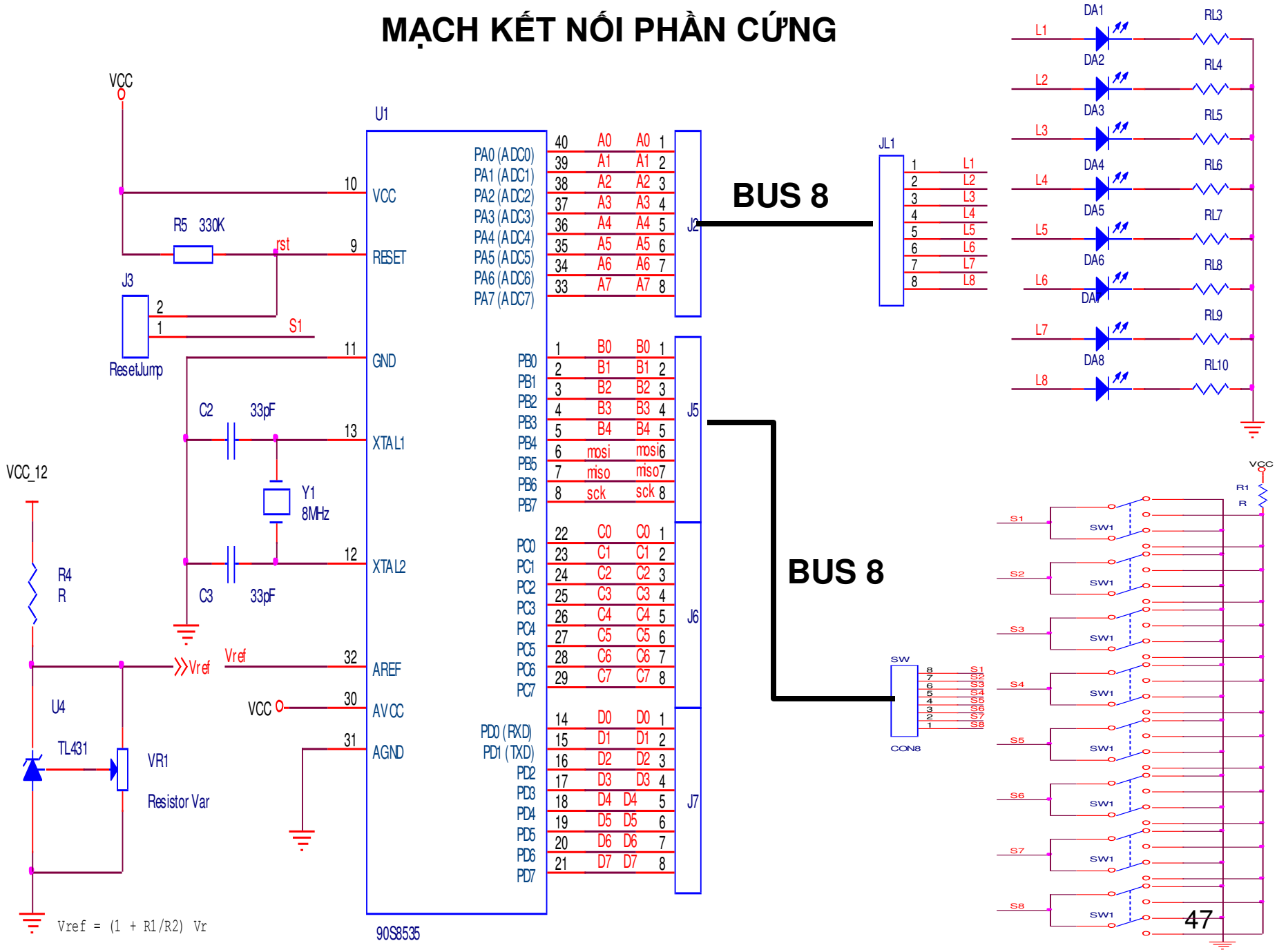
BÀI SỐ 1: ĐIỀU KHIỂN CÔNG TẮC ĐƠN VÀ LED ĐƠN

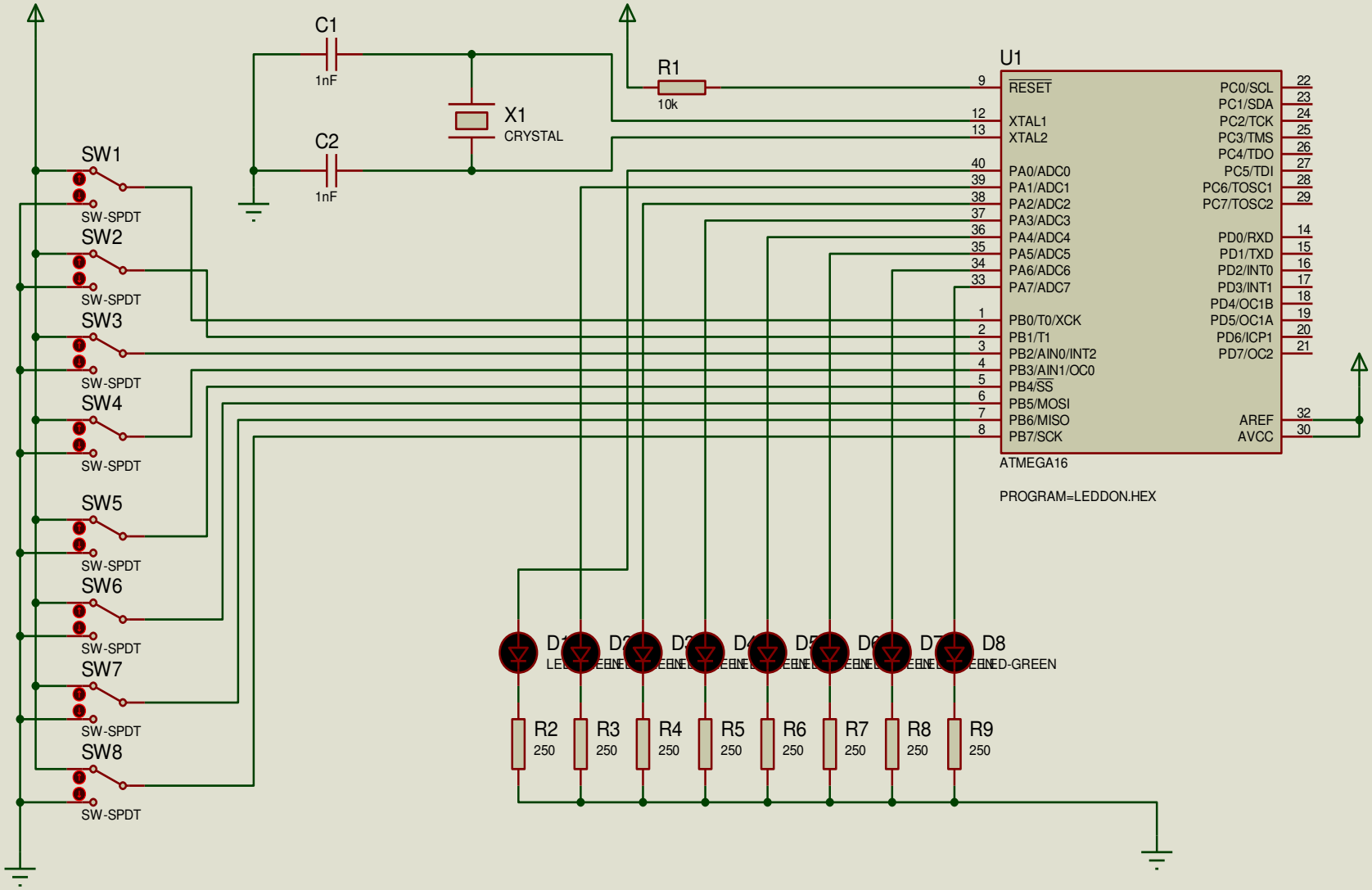
- Công tắc 1: 1 LED sáng chạy qua lại.
- Công tắc 2: 8 LED sáng dần sau đó tắt dần.
- Công tắc 3: 8 LED sáng từ hai bên vào sau đó tắt từ hai bên vào.
- Công tắc 4: 1 LED sáng chạy từ trái qua phải sau đó dừng lại sáng ở LED bên phải cho đến khi 8 LED cùng sáng.
- Công tắc 5: 8 LED chớp tắt, 4 sáng 4 tắt xen kẽ, 4 sáng 4 tắt mỗi bên.
- Công tắc 6: Số LED chạy qua lại tăng dần.
- Công tắc 7: LED sáng chạy từ hai bên vào và sáng dần từ giữa ra.
- Công tắc 8: 2 LED sáng chạy vào từ hai bên và chạy ra hai bên.

BÀI SỐ 1: ĐIỀU KHIỂN CÔNG TẮC ĐƠN VÀ LED ĐƠN

- Hướng dẫn thực hiện phần cứng:
  - Kết nối một cổng tới LED đơn: PORTA
  - Kết nối một cổng tới công tắc đơn: PORTB
- Hướng dẫn lập trình:
  - Cổng kết nối tới LED phải được khởi động là OUT bằng cách gán thanh ghi định hướng DDRx=FFh (DDRA=FFh).
  - Cổng kết nối tới công tắc phải được khởi động là IN bằng cách gán thanh ghi định hướng DDRx=00h (DDRB=00h).
  - Nhập dữ liệu từ PORTB sau đó chọn kểu sáng tương ứng bằng cấu trúc lập trình "Switch case".

MẠCH KẾT NỐI PHẦN CỨNG





MẠCH MÔ PHỎNG TRÊN PROTUES

MACH LED DON

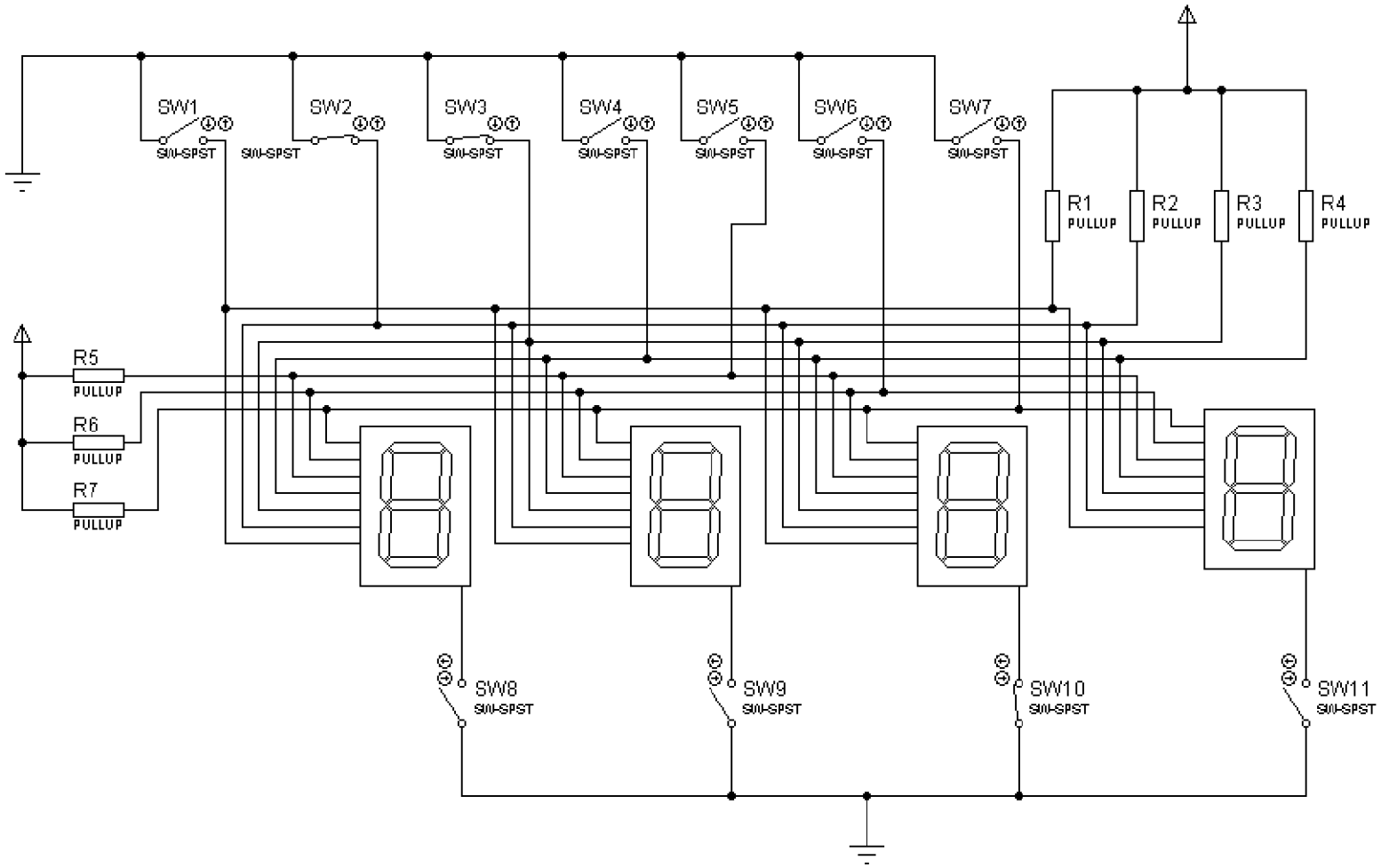
BÀI SỐ 2: HIỂN THỊ TRÊN LED 7 ĐOẠN

1. Mục đích: Sử dụng cổng vào ra song song hiển thị các số trên LED 7 đoạn.
2. Yêu cầu thực hiện:
    1. Thực hiện chương trình hiển thị các số từ 1 đến 4 lên 4 LED 7 đoạn.
    2. Thực hiện chương trình hiển thị các số từ 0 đến 9 chạy tuần tự từ phải qua trái.

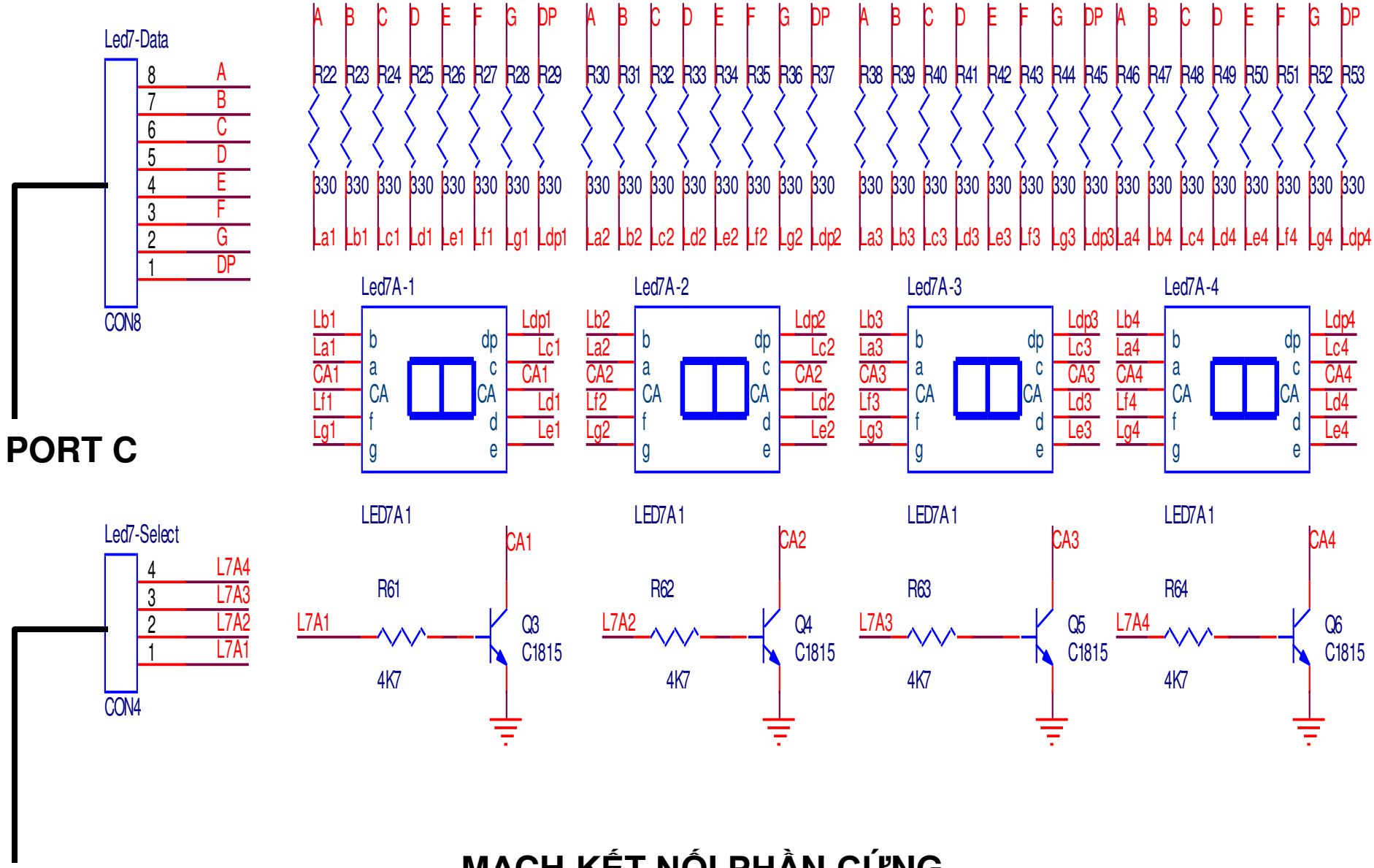
BÀI SỐ 2: HIỂN THỊ TRÊN LED 7 ĐOẠN

- Hướng dẫn thực hiện phần cứng:
  - Kết nối một cổng tới ngõ vào dữ liệu của các LED: PORTC
  - Kết nối một cổng tới ngõ vào chọn LED: PORTD
- Hướng dẫn lập trình:
  - Các cổng đều khởi động OUT.
  - Muốn đèn LED nào sáng cấp dữ liệu số tương ứng sau đó cấp bit chọn bằng 1. Như vậy tại một thời điểm chỉ có một LED sáng. Nhưng nếu chương trình thực hiện nhanh sẽ thấy tất cả các LED đều sáng.
  - Để số này không lem sang LED kế bên cần xoá toàn bộ các LED trước khi hiển thị số kế tiếp.
  - Giải thuật sẽ là: cấp dữ liệu – chọn LED sáng – tắt tất cả các LED – cấp dữ liệu khác – chọn LED kế tiếp.
  - Để các LED chạy cần có biến cung cấp dữ liệu cho các LED.

File mô phỏng tại D:\Baigiangvixuly\slidetiengviet\led7doan



MẠCH MÔ PHỎNG NGUYÊN LÝ QUÉT LED

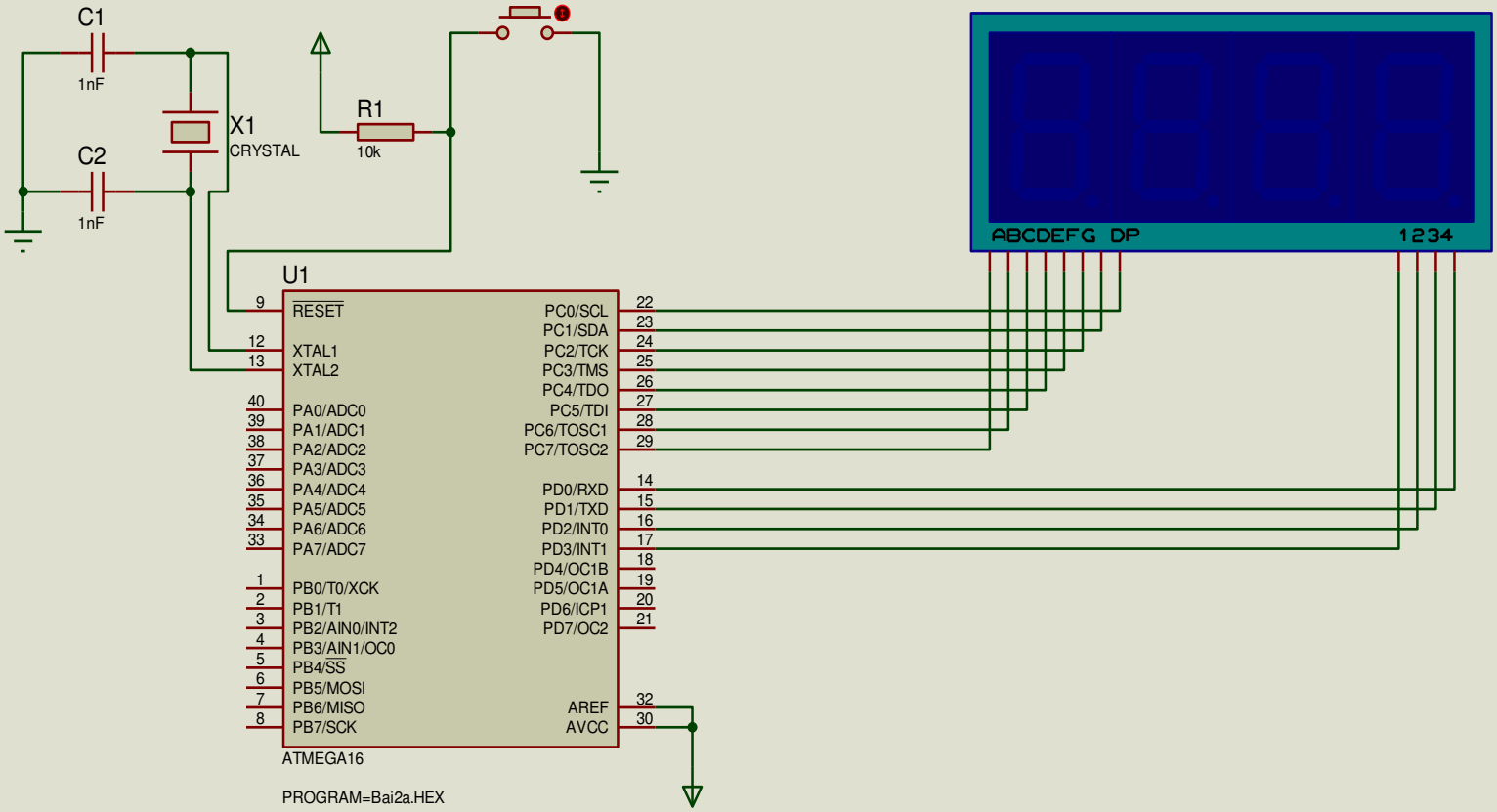


PORT C

PORT D

MẠCH KẾT NỐI PHẦN CỨNG

MẠCH MÔ PHỎNG TRÊN PROTUES



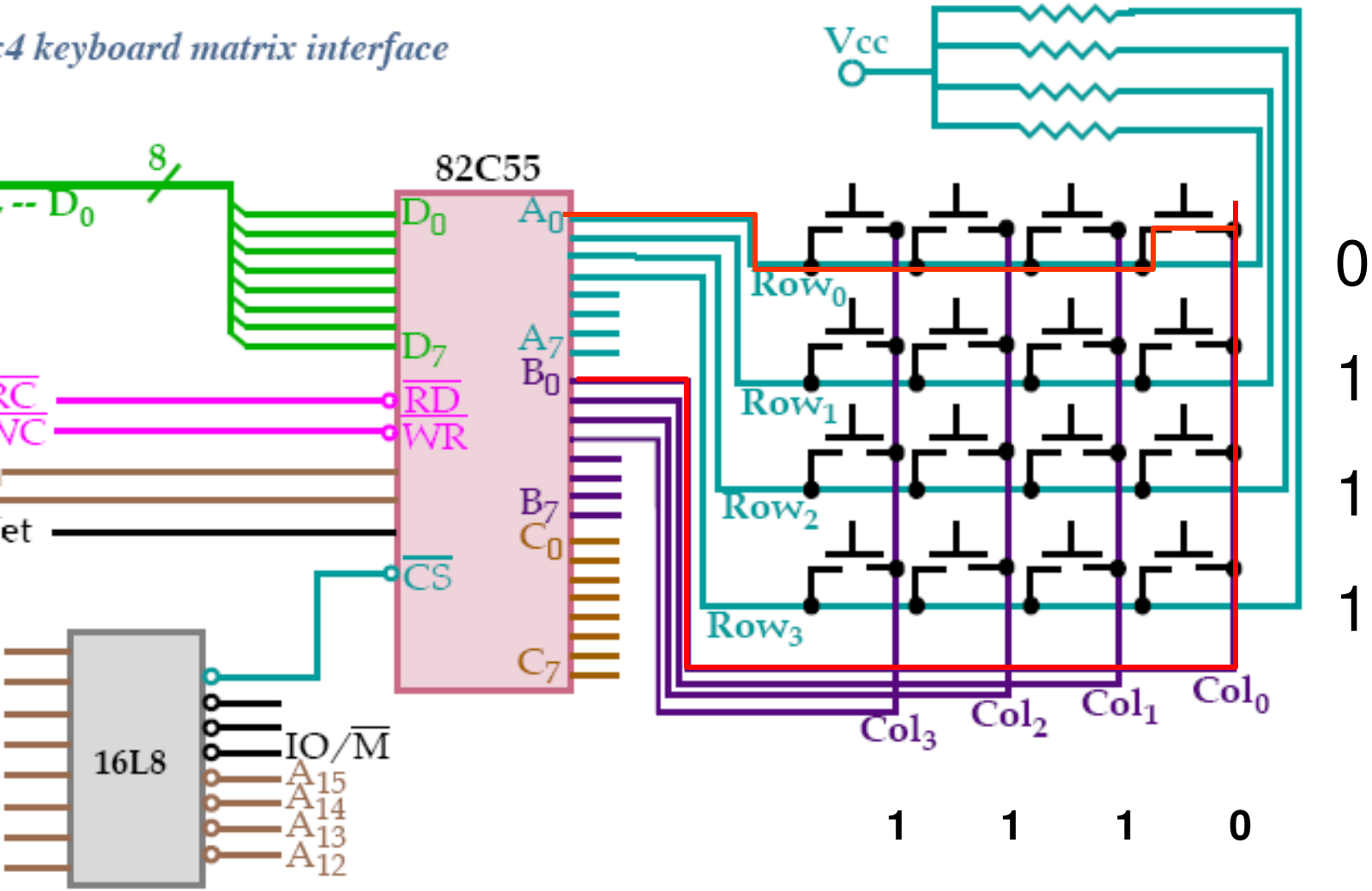
BÀI SỐ 3: ĐIỀU KHIỂN ĐỌC BÀN PHÍM MA TRẬN HIỂN THỊ MÃ PHÍM TRÊN LED 7 ĐOẠN

1. Mục đích: Thực hiện giải thuật quét bàn phím ma trận.
2. Yêu cầu thực hiện:
    1. Thực hiện chương trình đọc mã bàn phím hiển thị lên 1 LED 7 đoạn.
    2. Thực hiện chương trình đọc mã bàn phím ma trận hiện thị lên 4 LED 7 đoạn, khi một phím mới được nhấn, mã phím cũ sẽ chuyển qua LED bên trái một vị trí, để mã phím mới hiện lên trên LED bên phải.

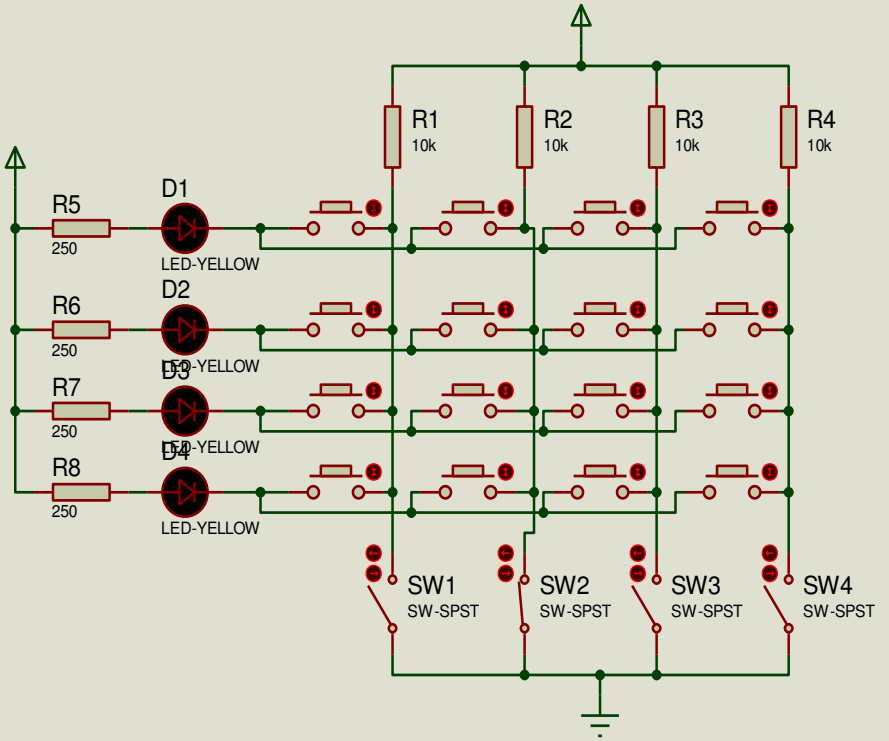
BÀI SỐ 3: ĐIỀU KHIỂN ĐỌC BÀN PHÍM MA TRẬN HIỂN THỊ MÃ PHÍM TRÊN LED 7 ĐOẠN

- Hướng dẫn thực hiện phần cứng:
  - Kết nối bộ hiển thị LED 7 đoạn tới các cổng AVR như trong bài 2: PORTC và PORTD.
  - Kết nối một cổng tới bàn phím ma trận: PORTA
- Hướng dẫn lập trình:
  - Các cổng PORTC và PORTD khởi động OUT như bài 2.
  - Cổng A khởi động các bit cao OUT để cấp dữ liệu quét hàng, các bít thấp IN để nhận dữ liệu từ cột.
  - Để nhận được phím nhấn lần lượt cấp mức 0 ra các hàng, và đọc dữ liệu từ các cột vào. Gọi vị trí bit 0 ở mã hàng là x, vị trí bit 0 ở mã cột là y thì mã phím nhấn sẽ tính bằng: 4x+y (sẽ là 1 số hex từ 0 – F)

uyên tắc quét bàn phím ma trận



Mô phỏng mạch nguyên lý bàn phím ma trận
D:\Baigiangvixuly\Slidetiengviet\machmophong

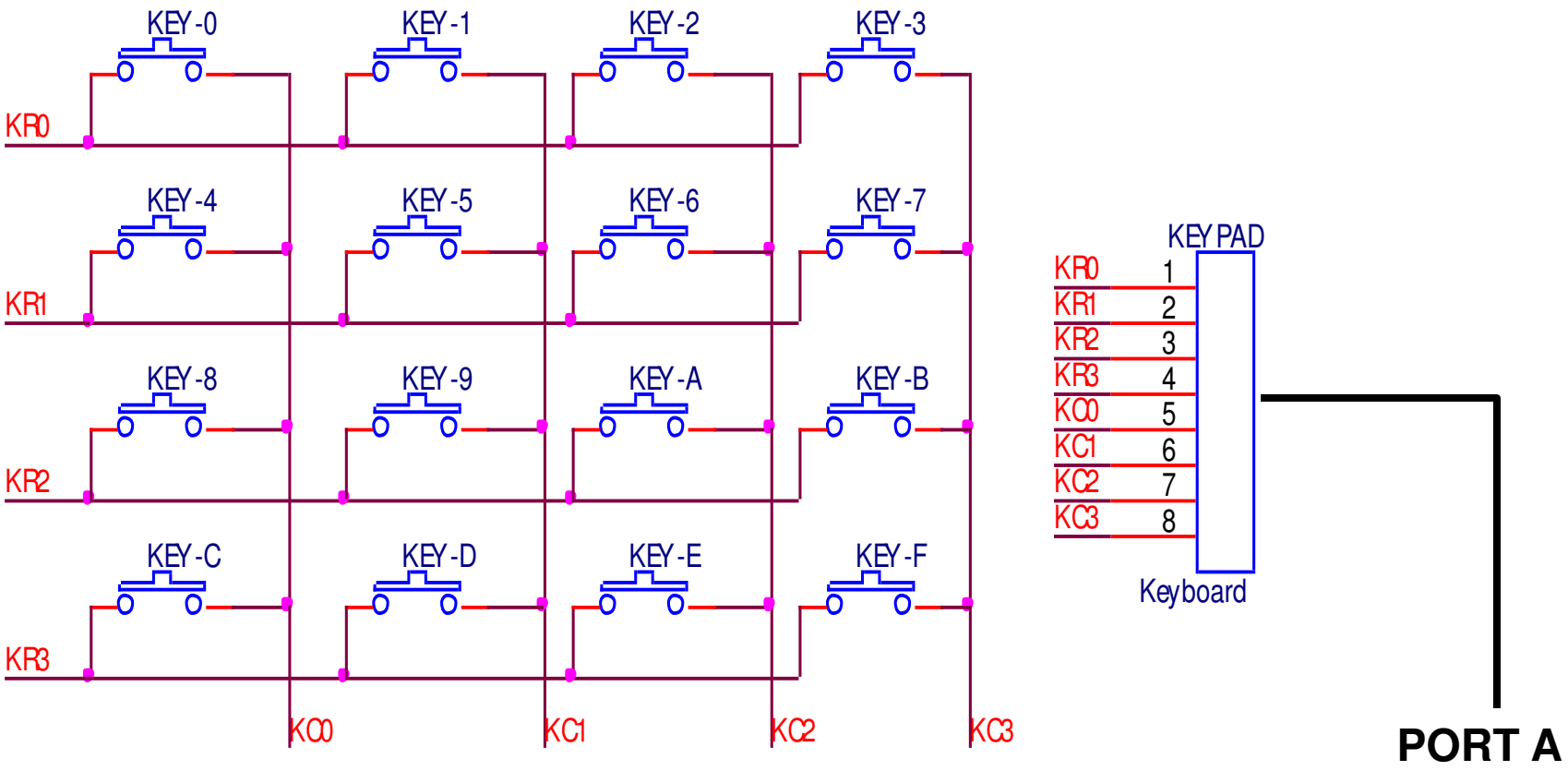


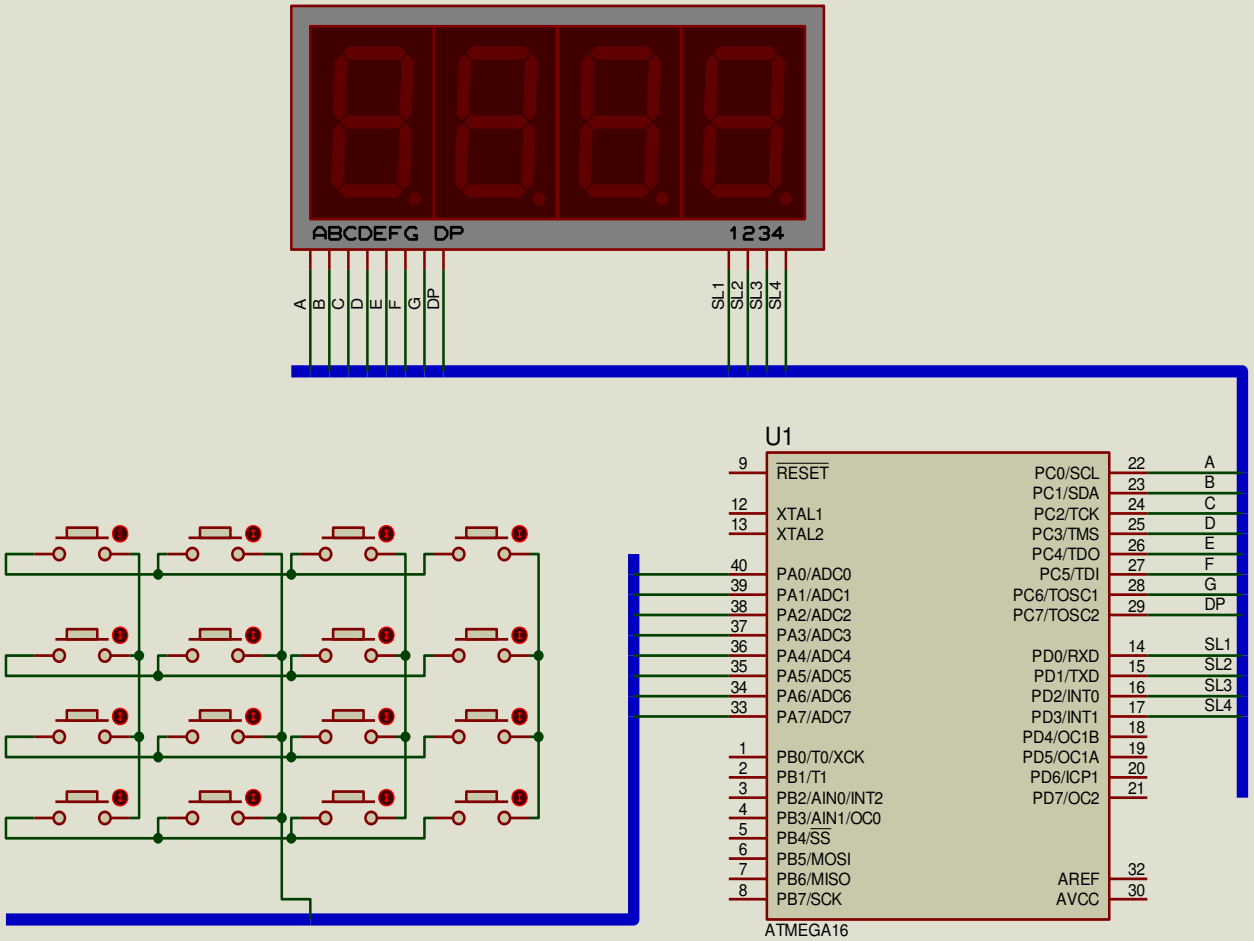
MẠCH MÔ PHỎNG NGUYÊN LÝ QUÉT BÀN PHÍM MA TRẬN TRÊN PROTUES

BÀI SỐ 3: ĐIỀU KHIỂN ĐỌC BÀN PHÍM MA TRẬN HIỂN THỊ MÃ PHÍM TRÊN LED 7 ĐOẠN

- Để các số hiển thị đẩy từ trái qua phải cần khai báo 4 biến chứa mã hiển thị. Khi có một số mới nhập từ bàn phím giá trị mã hiển thị sẽ được chuyển đi một vị trí, và mã mới nhập sẽ được ghi vào biến bên phải, mã hiển thị bên trái sẽ bị mất.

MẠCH KẾT NỐI PHẦN CỨNG BÀN PHÍM



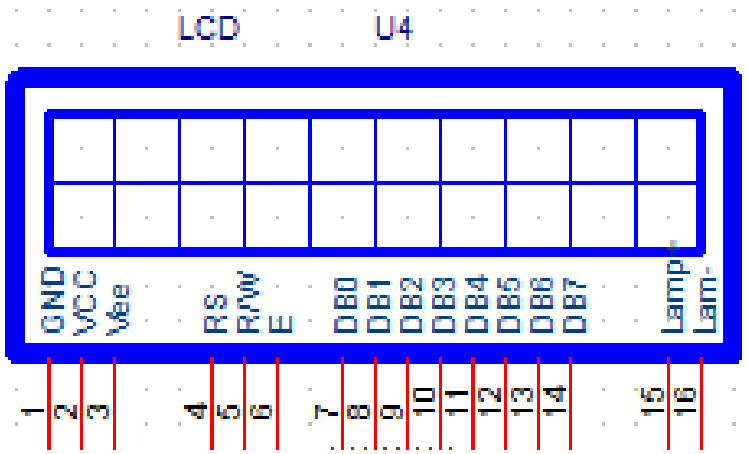


MẠCH MÔ PHỎNG TRÊN PROTUES

BÀI SỐ 4: ĐIỀU KHIỂN HIỂN THỊ TRÊN LCD

1. Mục đích: Tìm hiểu về màn hình LCD và cách điều khiển hiển thị các ký tụ trên màn hình LCD.
2. Yêu cầu thực hiện:
    1. Hiển thị hai dòng chữ trên màn hình LCD.
    2. Hiển thị dòng chữ trôi trên màn hình LCD.

Các tín hiệu của LCD



Khởi động LCD

- Ghi tới LCD hai lệnh:
  - Lệnh 0x38: hiển thị 2 hàng 16 ký tự
  - Lệnh 0x0e: Display on – cussor on
- Để ghi một lệnh tới LCD
  - Cấp lệnh tới data bus: DB7-DB0
  - E=0; R/W=0; RS=0
  - E=1; delay
  - E=0; delay

Ghi ký tự để hiển thị trên LCD

- Cấp mã ASCII của ký tự muốn hiện tới DB7-DB0
- E=0; R/W=0; RS=1
- E=1; delay
- E=0; delay

BÀI SỐ 4: ĐIỀU KHIỂN HIỂN THỊ KÝ TỰ TRÊN MÀN HÌNH LCD.

- Hướng dẫn thực hiện phần cứng:
  - Kết nối màn hình LCD tới các cổng AVR: PORTC nối tới cổng dữ liệu (DB0 – DB7), PORTD cấp các tín hiệu E, RS, R/W#.
- Hướng dẫn lập trình:
  - Muốn hiển thị ký tự trên màn hình LCD trước hết phải khởi động nó bằng cách ghi giá trị 38H tới thanh ghi điều khiển của nó, sau đó bật bộ hiển thị bằng lệnh 0eH. Chọn thanh ghi điều khiển bằng cách cấp RS=0, ghi bằng cách cấp R/W = 0 và cho E lến mức 1 sau đó kéo xuống mức 0.
  - Trước khi ghi dữ liệu tới hiển thị lên LCD cần kiểm tra trạng thái sẵn sàng của nó bằng cách đọc thanh ghi trạng thái (RS=0, R/W=1 và E = L-H-L). Nếu bit cao nhất của thanh ghi trạng thái bằng 1 thì LCD đã sẵn sàng nhận dữ liệu. Xem bit cao nhất của từ trạng thái, nếu = 1 thì LCD sẵn sàng.
  - Để hiển thị ký tự trên LCD cần ghi mã ASCII của ký tự cần hiển thị tới nó (RS=1, R/W=0, E = L-H-L, DB7-0=ASCII).
  - Để dòng chữ trôi trên LCD có thể ghi lệnh 18H (dịch trái) hoặc 1CH (dịch phải) tới LCD.

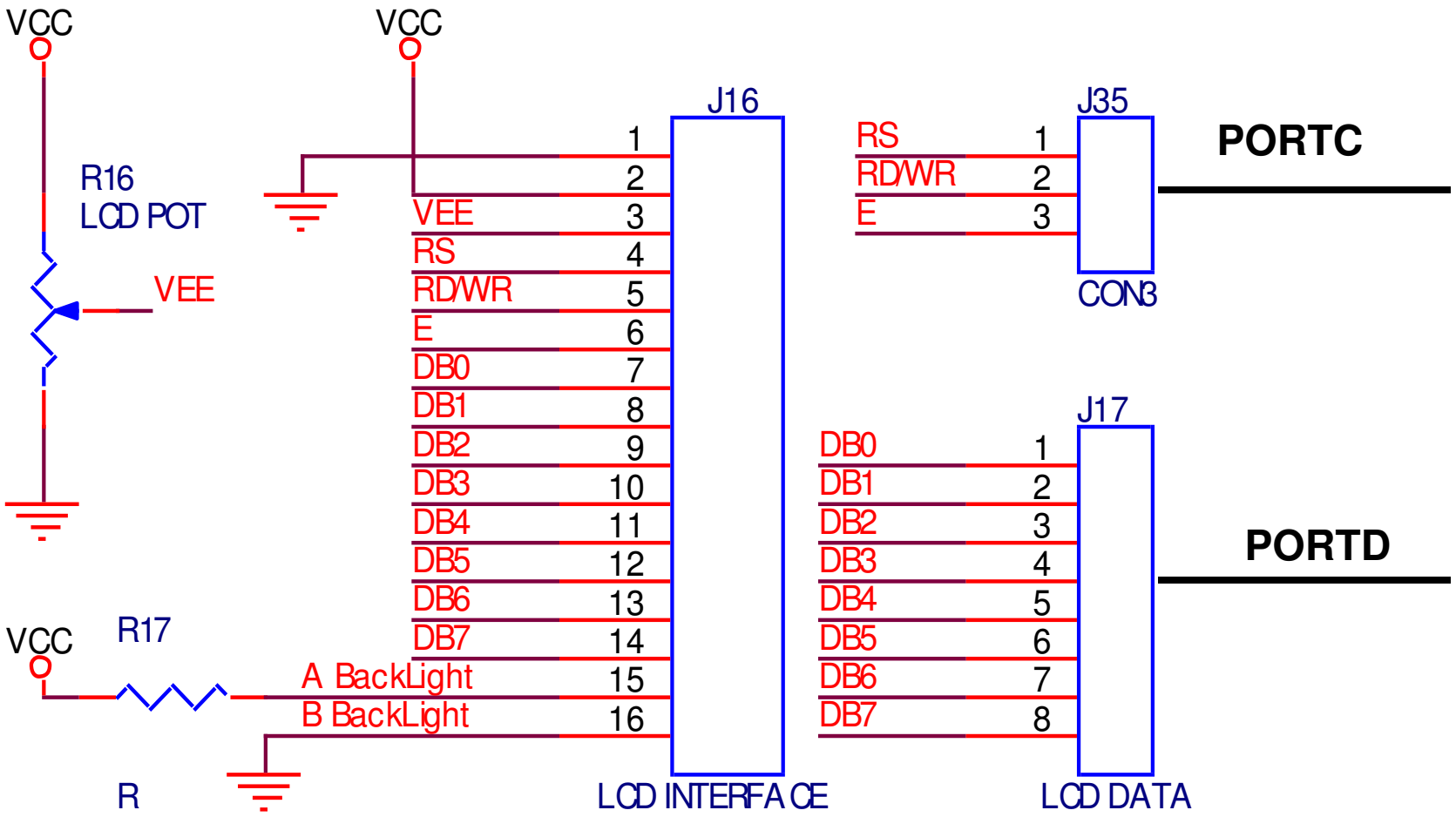
Các mã lệnh lập trình cho LCD

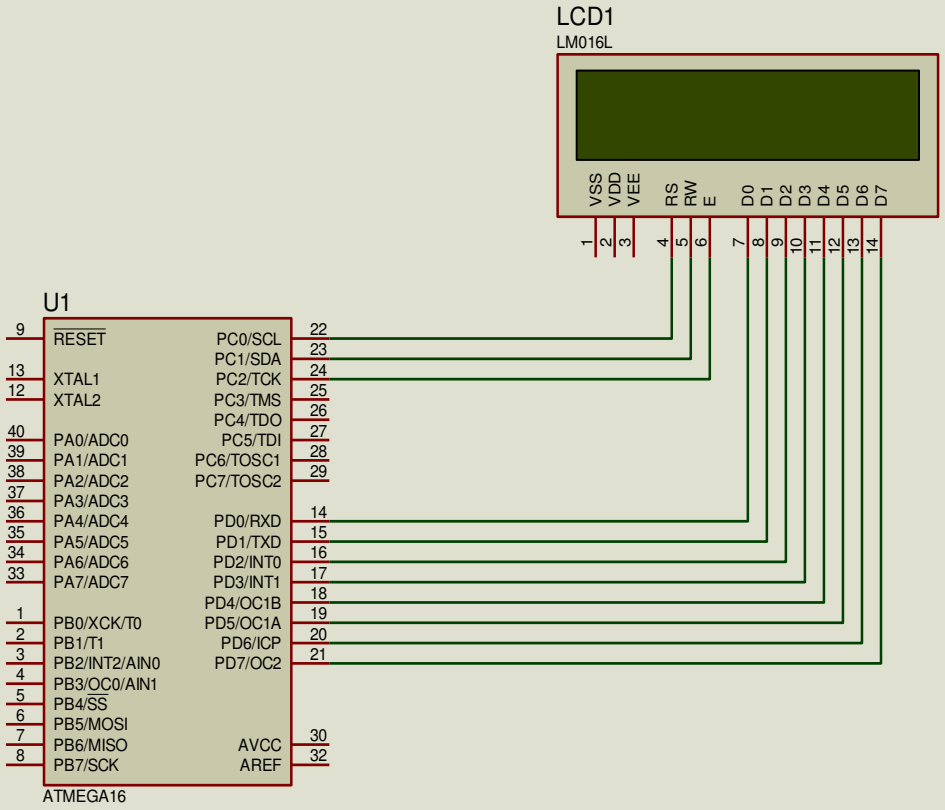
Table 4-8: LCD Command Codes

Code (hex)	Command to LCD Instruction Register
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to left)
6	Increment cursor (shift cursor to right)
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor on
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
C0	Force cursor to beginning of 2nd line
38	2 lines and 5x7 matrix

Note: This table is extracted from Table 4-10.

MẠCH KẾT NỐI PHẦN CỨNG





MẠCH MÔ PHỎNG TRÊN PROTUES

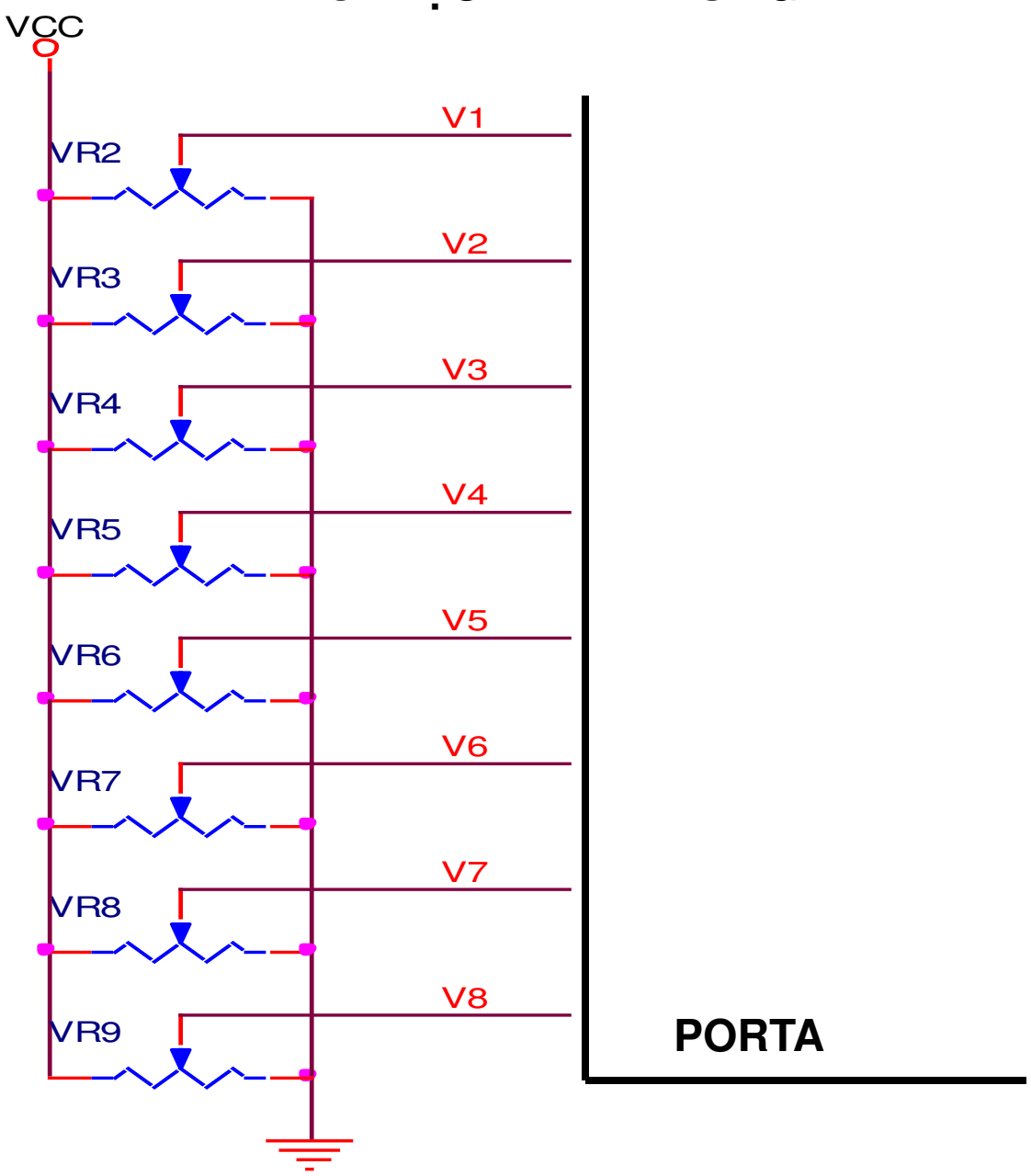
BÀI SỐ 5: ĐIỀU KHIỂN ĐỌC ĐIỆN ÁP DC BẰNG ADC VÀ HIỂN THỊ LÊN LCD

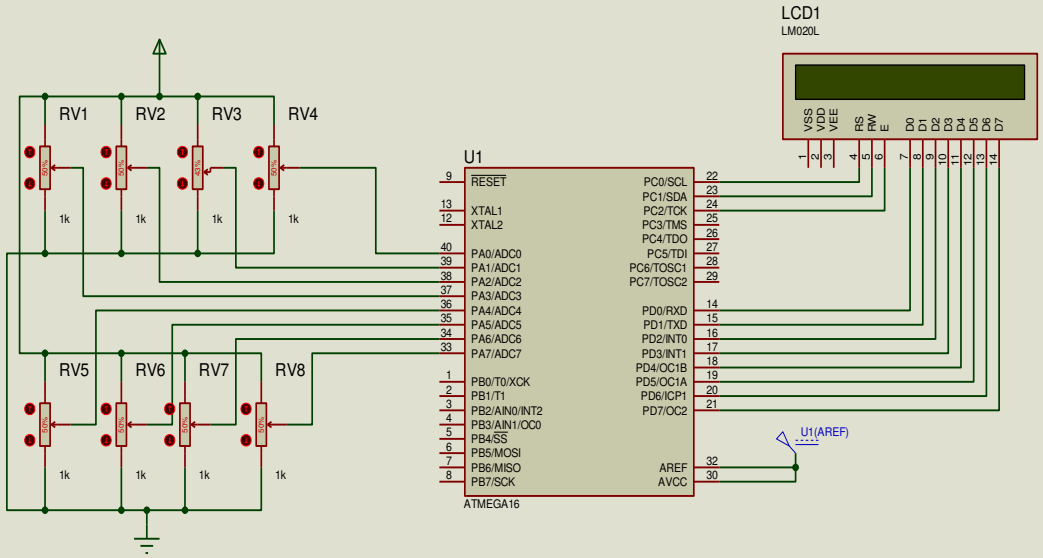
1. Mục đích: Tìm hiểu bộ biến đổi tương tự sang số (ADC – Analogue to Digital Converter) tích hợp trong AVR.
2. Yêu cầu thực hiện:
    1. Đọc một ngõ vào PA0 và hiển thị giá trị HEXA (DEC) (BIN) trên màn hình LCD.
    2. Đọc 8 kênh vào ADC và hiển thị giá trị điện áp thực trên màn hình LCD.

BÀI SỐ 5: ĐIỀU KHIỂN ADC ĐỌC ĐIỆN ÁP DC HIỂN THỊ TRÊN MÀN HÌNH LCD.

- Hướng dẫn thực hiện phần cứng:
  - Kết nối màn hình LCD tới các cổng AVR: PORTD nối tới cổng dữ liệu (DB0 – DB7), PORTC cấp các tín hiệu E, RS, R/W#. (Như trong bài số 4).
  - Kết nối PORTA tới các biến trở để cung cấp điện áp DC tới các ngõ vào ADC
- Hướng dẫn lập trình:
  - Tạo Project mới trong CodeVisionAVR dùng CodeWizard cho phép ADC ở chế độ 8 bit, CodeWizard sẽ tự tạo ra chương trình con đọc ADC và trả về giá trị trong thanh ghi ADCH.
  - Giá trị số trong thanh ghi ADCH sẽ tương ứng với giá trị điện áp tương tự ở ngõ vào PORTA của AVR.
  - Cần phải biến đổi dữ liệu HEX trong ADCH thành số ASCII để hiển thị lên LCD.
  - Khi hiển thị điện áp cần đổi số trong ADCH thành dạng float. Sau đó nhân với 1000 và chia 10 lấy số dư để được giá trị thập phân cần hiển thị.

KẾT NỐI MẠCH PHẦN CỨNG





MẠCH MÔ PHỎNG TRÊN PROTUES

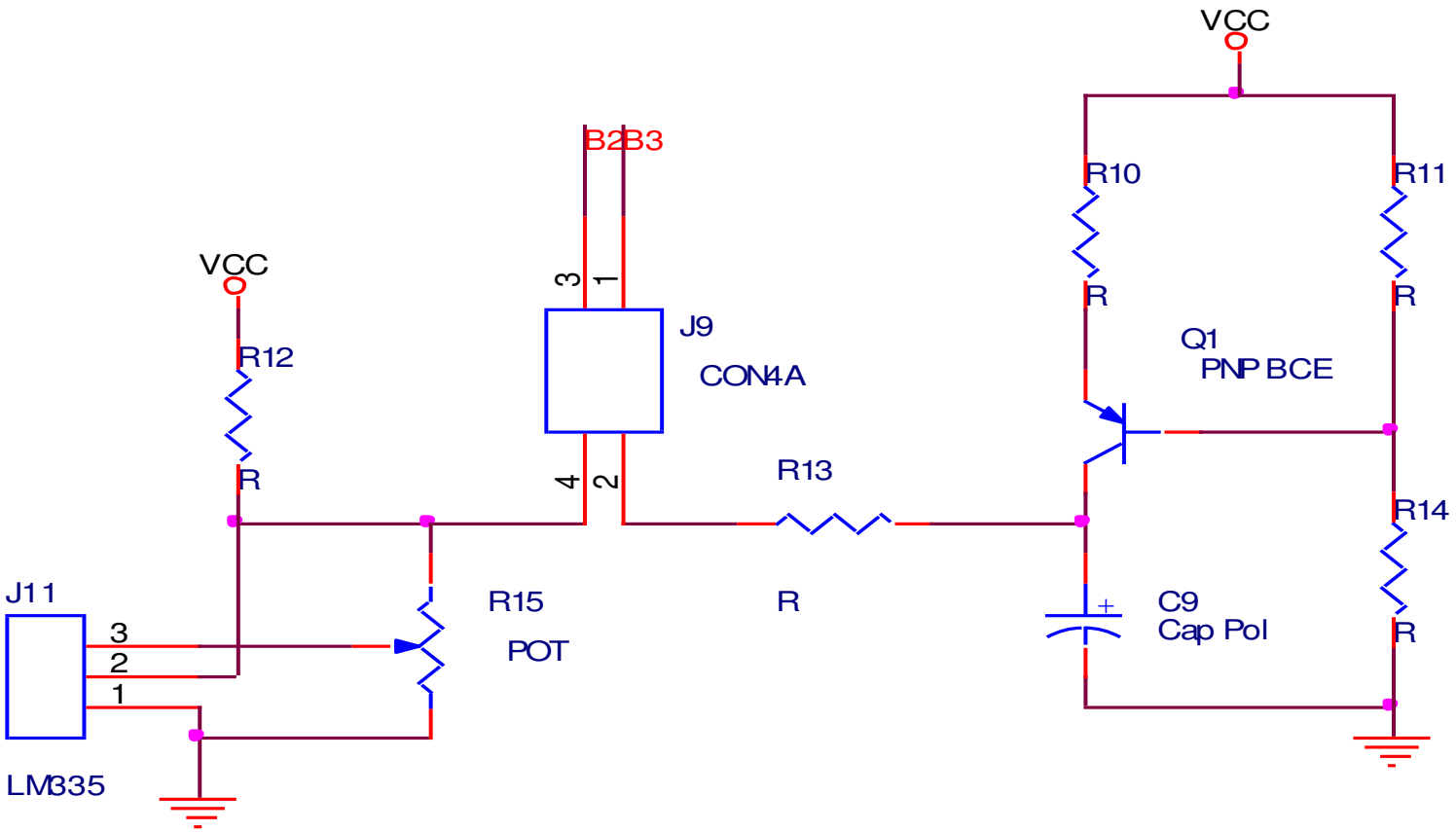
BÀI SỐ 6: ĐO NHIỆT ĐỘ VÀ HIỂN THỊ TRÊN LED 7 ĐOẠN DÙNG MẠCH SO SÁNH TƯƠNG TỰ TÍCH HỢP TRONG AVR

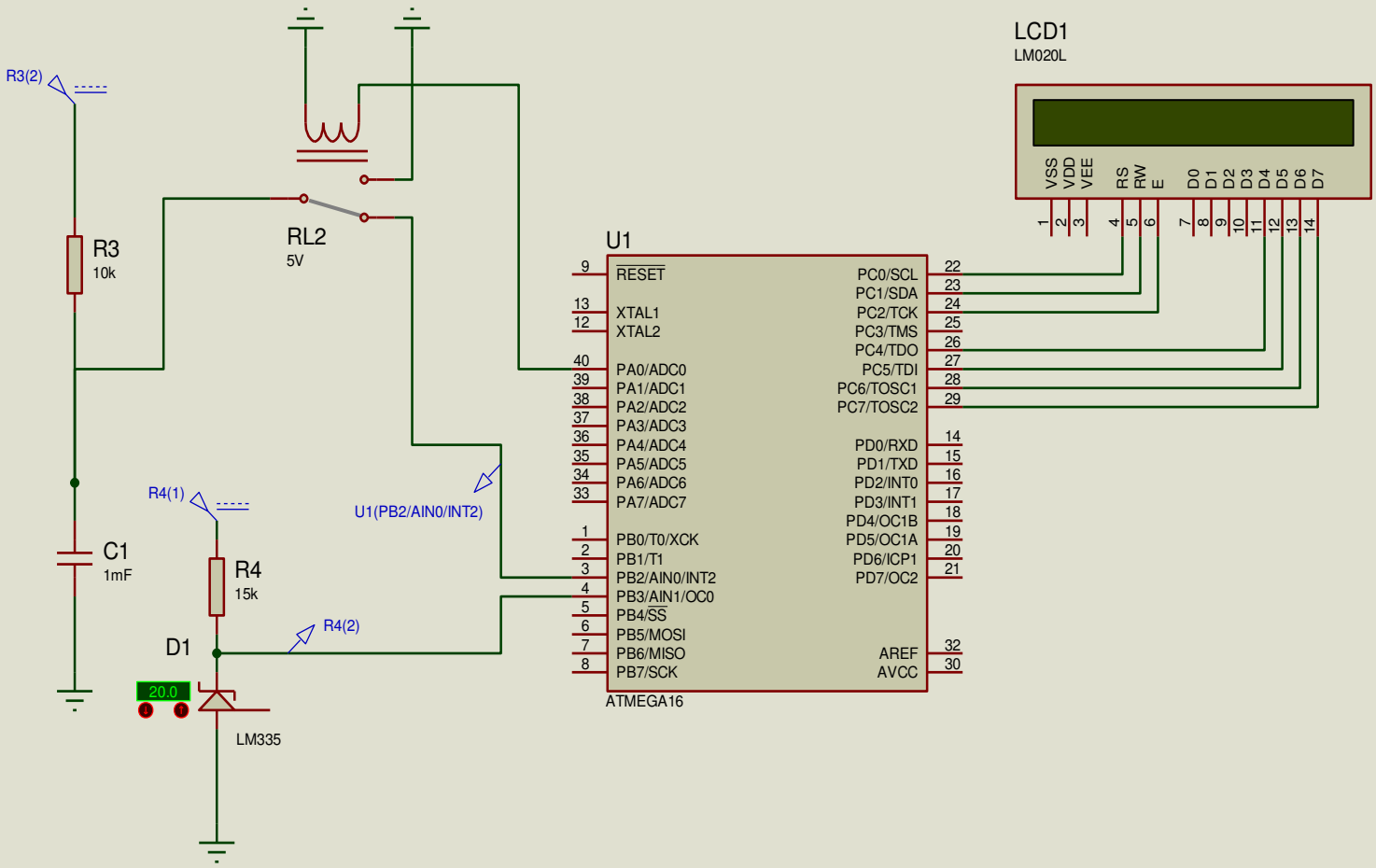
1. Mục đích:
   1. Tìm hiểu về ứng dụng của mạch so sánh tương tự .
   2. Tìm hiểu về IC cảm biến nhiệt độ LM335.
2. Yêu cầu thực hiện:
   1. Đọc thời gian nạp tụ điện hiển thị trên LED 7 đoạn.
   2. Đổi thời gian nạp tụ thành giá trị điện áp hiển thị trên LED 7 đoạn.
   3. Đổi điện áp trên tụ thành giá trị nhiệt độ hiển thị trên LED 7 đoạn.

BÀI SỐ 6: ĐO NHIỆT ĐỘ VÀ HIỂN THỊ TRÊN LED 7 ĐOẠN DÙNG MẠCH SO SÁNH TƯƠNG TỰ CỦA AVR

- Hướng dẫn thực hiện phần cứng:
  - Kết nối bộ hiển thị LED 7 đoạn tới AVR giống như trong bài số 2.
  - Cắm các Jumper để kết nối các mạch đo nhiệt độ và mạch nạp tụ tới bộ so sánh tương tự PINB2 và PINB3 như hình phần cứng.
- Hướng dẫn lập trình:
  - Trước hết khởi động PORTB.3 là ngõ ra và cấp mức 0 tới ngõ ra này tạo trễ một khoảng thời gian để tụ điện xả hết.
  - Cho phép mạch so sánh tương tự (Analogue Compare), khi đó PORTB.3 trở thành ngõ vào của mạch Op-Amp bên trong AVR (có tổng trở vào rất lớn), nên tụ điện sẽ được nạp qua transistor.
  - Tại thời điểm tụ bắt đầu nạp cho phép Timer hoạt động, khi điện áp trên tụ bằng với điện áp của mạch LM335 ngõ ra Analogue compare sẽ lên mức 1. Với giá trị bộ đếm Timer sẽ biết được thời gian nạp tụ.
  - Tuỳ theo giá trị linh kiện trong mạch nạp tụ sẽ tính được điện áp trên tụ.
  - Điện áp trên tụ cũng chính bằng điện áp ngõ ra mạch đo nhiệt độ LM335, nên theo thông số của LM335 có thể suy ra được giá trị nhiệt độ tương ứng.

KẾT NỐI MẠCH PHẦN CỨNG





MẠCH MÔ PHỎNG TRÊN PROTUES

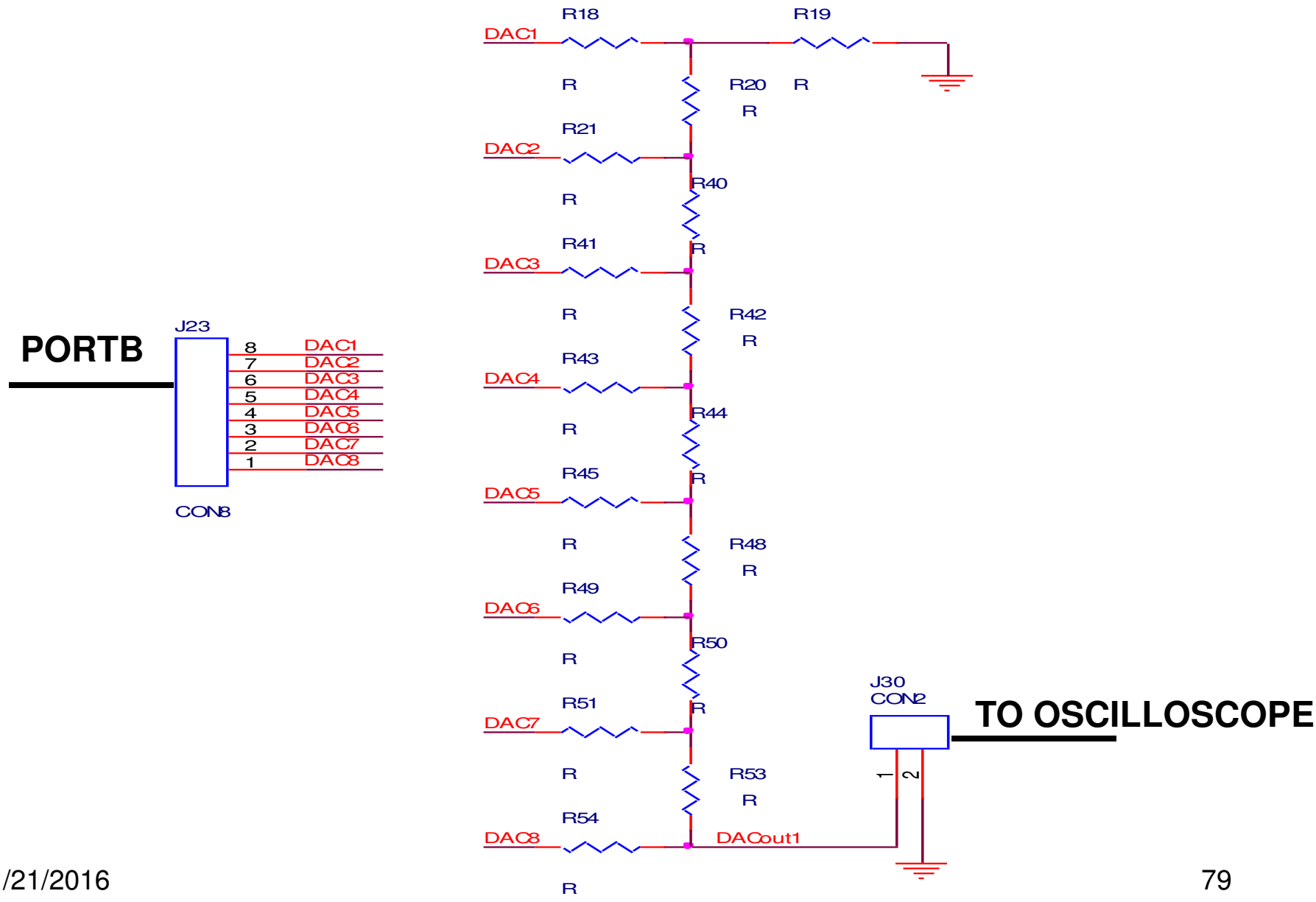
BÀI SỐ 7: TẠO XUNG VUÔNG, SIN, TAM GIÁC BẰNG DAC

1. Mục đích:
   1. Tìm hiểu mạch biến đổi DAC .
   2. Giải thuật số tạo ra các tín hiệu tương tự.
2. Yêu cầu thực hiện:
   1. Tạo mức điện áp DC ngõ ra thay đổi theo vị trí biến trở nối tới ngõ vào kênh ADC.
   2. Thực hiện hệ thống tạo ra các xung Vuông, Sin, Tam giác được chọn bằng các SW. Tần số và biên độ của các xung vuông có thể thay đổi bằng hai biến trở nối tới hai ngõ vào ADC.

BÀI SỐ 7: TẠO XUNG VUÔNG, SIN, TAM GIÁC BẰNG DAC

- Hướng dẫn thực hiện phần cứng:
  - Nối mạch biến trở tới PORTA của vi điều khiển.
  - Nối PORTB tới ngõ vào mạch DAC.
  - Ngõ ra mạch DAC nối tới Oscilloscope.
- Hướng dẫn lập trình:
  - Viết chương trình đọc giá trị ADC từ ngõ vào PINA.0 chuyển giá trị ra PORTB đưa tới DAC. Xoay biến trở và quan sát trên Oscilloscope để thấy mức tín hiệu thay đổi.
  - Để tạo ra xung vuông với biên độ thay đổi cần cấp giá trị đọc từ ADC ra DAC tạo Delay sau đó cấp giá trị 0 ra DAC rồi tạo Delay. Để thay đổi tần số sóng vuông cần viết chương trình delay với thời gian delay phụ thuộc vào giá trị đọc được từ kênh 1 ADC (PINA.1).
  - Để tạo xung sin cần cấp ra DAC các giá trị theo bảng sin được định nghĩa trước. Giá trị biên độ sẽ xác định bằng giá trị cực đại đọc vào từ ADC. Tần số sóng sin sẽ thay đổi theo thời gian delay giữa các lần xuất dữ liệu ra DAC.
  - Xung tam giác sẽ tạo ra bằng cách xuất giá trị tăng dần cho tới giá trị cực đại, rồi giảm dần cho tới 0.

KẾT NỐI MẠCH PHẦN CỨNG



MẠCH MÔ PHỎNG TRÊN PROTUES

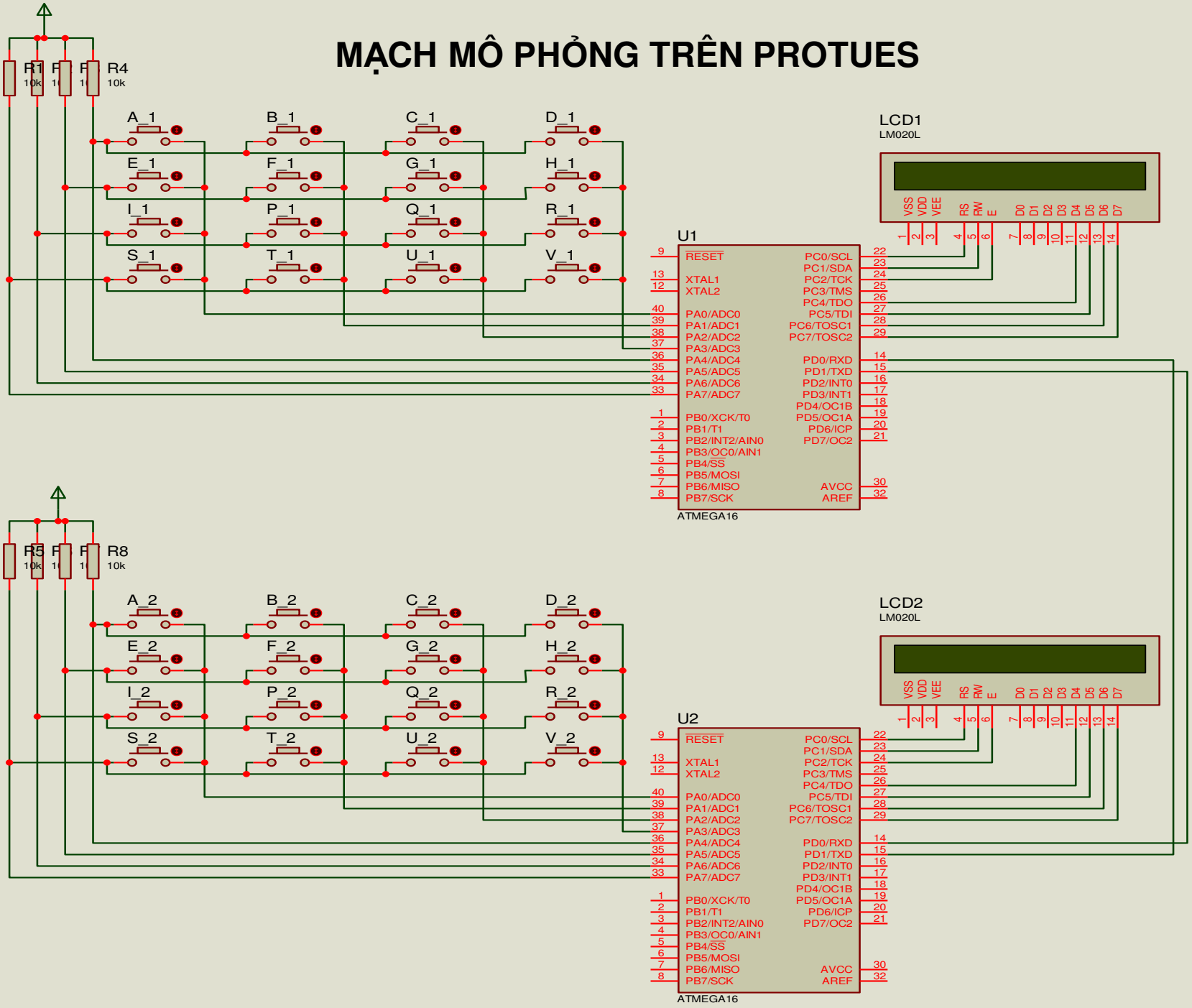
BÀI SỐ 8: GIAO TIẾP NỐI TIẾP USART

1. Mục đích:
   1. Tìm hiểu về giao tiếp nối tiếp USART của AVR .
   2. Tìm hiểu về giao tiếp với máy tính qua cổng COM.
2. Yêu cầu thực hiện:
   1. Kết nối hai KIT AVR bằng cổng nối tiếp USART, viết các chương trình truyền nhận dữ liệu để nhấn phím KIT bên này ký tự sẽ hiện lên màn hình LCD của KIT bên kia.
   2. Kết nối KIT AVR với máy tính, thực hiện các chương trình truyền nhận trên KIT AVR và trên máy tính để bật các công tắt đơn trên KIT, một điểm sáng tương ứng trên màn hình máy tính sẽ đổi màu, và nhấn chuột vào một điểm trên màn hình máy tính đèn LED đơn trên KIT sẽ sáng tắt.

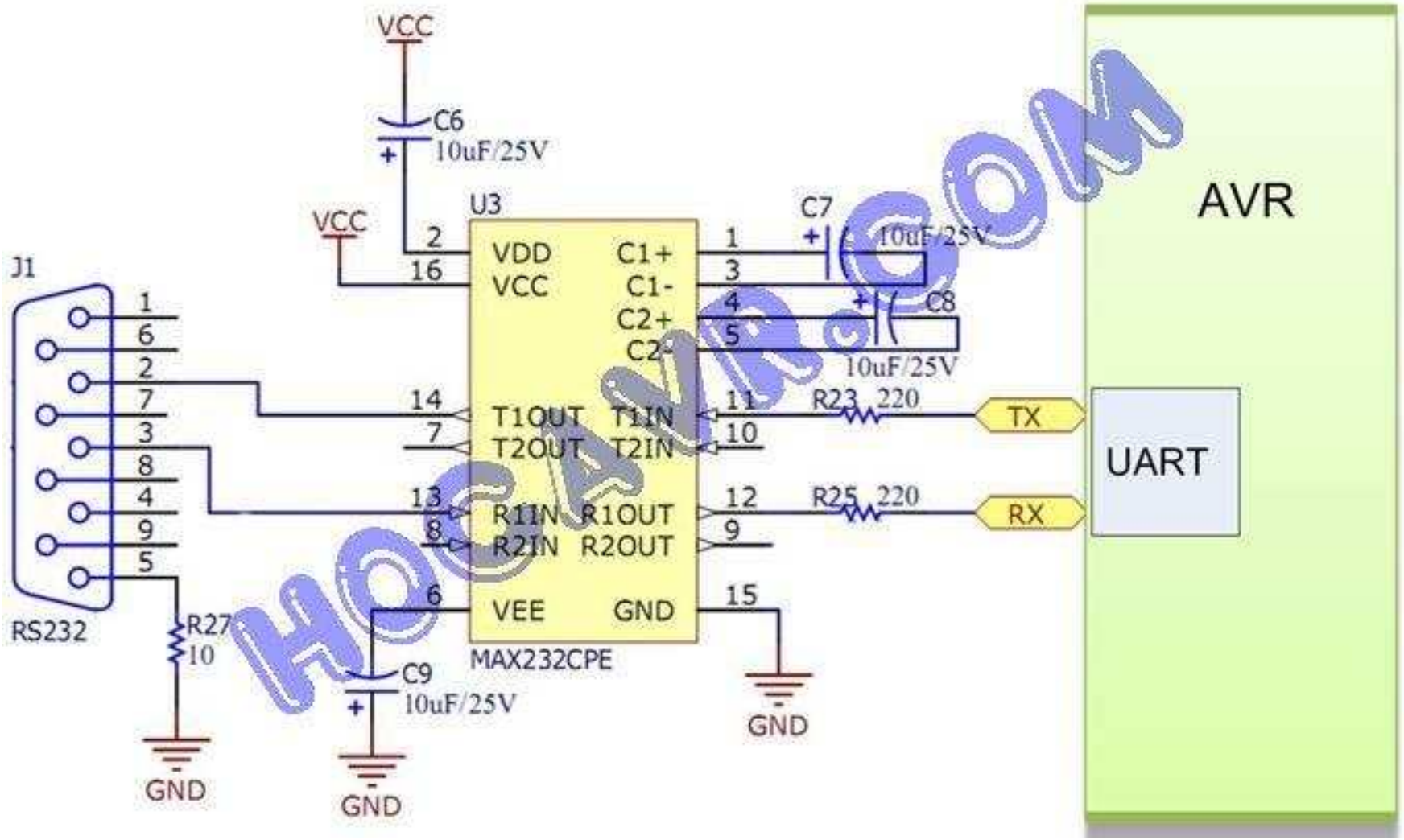
BÀI SỐ 8: GIAO TIẾP NỐI TIẾP USART

- Hướng dẫn thực hiện phần cứng:
  - Kết nối cổng nối tiếp của hai KIT. Trên mỗi KIT kết nối bàn phìm ma trận và màn hình LCD như hình vẽ để thực hiện phần 01.
  - Kết nối cổng RS232 của KIT AVR với cổng COM máy tính, nối LED đơn và SW tới các công AVR để thực hiện phần 2.
- Hướng dẫn lập trình:
  - Dùng CodeWizard để khởi động cổng USART, chọn chức năng ngắt truyền nhận, CodeWizard sẽ tự động tạo ra các chương trình ngắt truyền nhận dữ liệu trên cổng nối tiếp.
  - Viết các chương trình con đọc bàn phím ma trận và hiển thị LCD. Chương trình chính thấy có phím nhấn thì giử tới cổng nối tiếp, và nhận dữ liệu từ cổng nối tiếp thì hiển thị trên LCD.
  - Tham khảo chương trình truyền nhận dữ liệu bằng Visual Basic trên máy tính. Tương tự thực hiện các chương trình truyền nhận dữ liệu trên KIT AVR như trong phần 1.

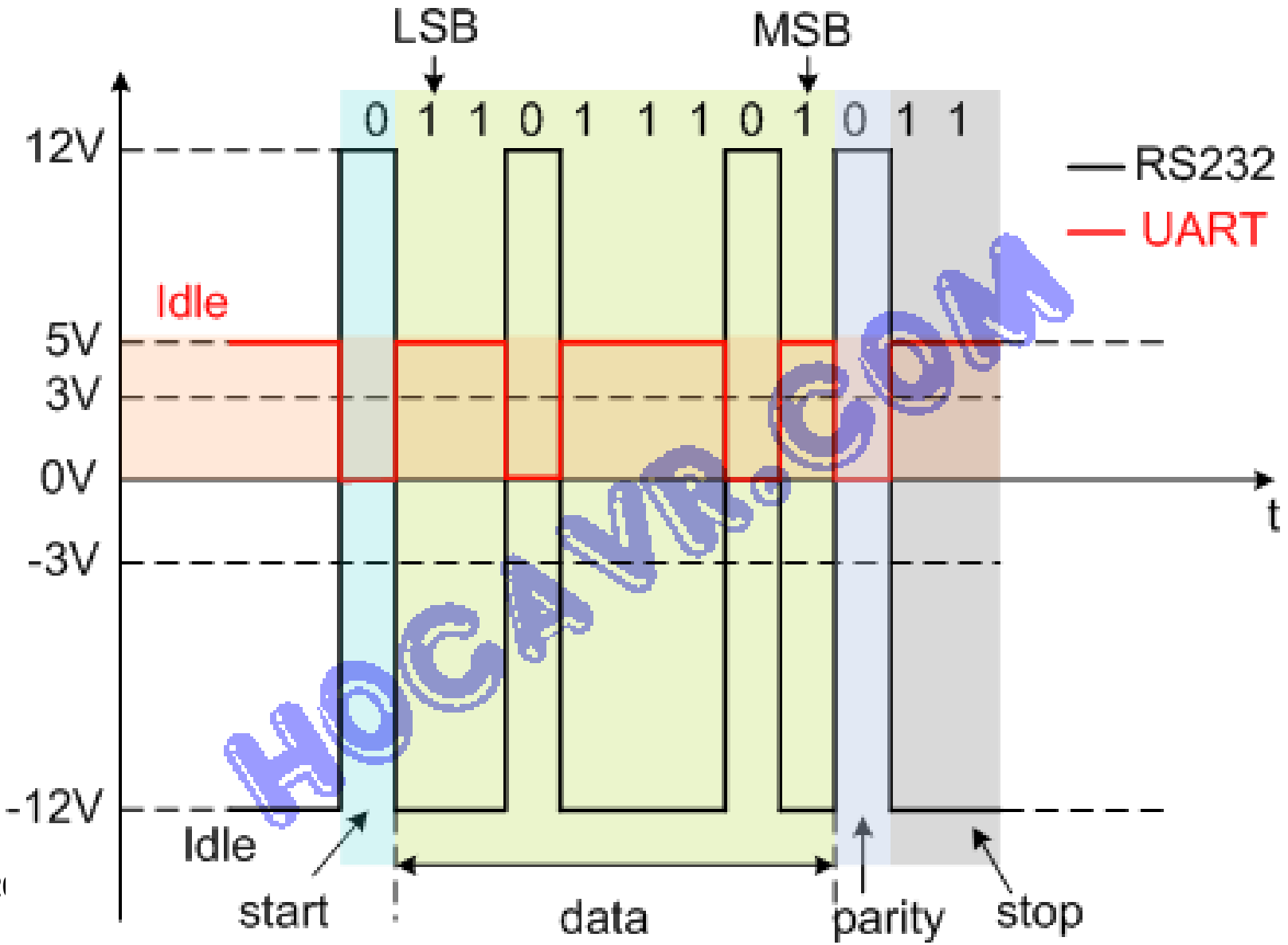
MẠCH MÔ PHỎNG TRÊN PROTUES



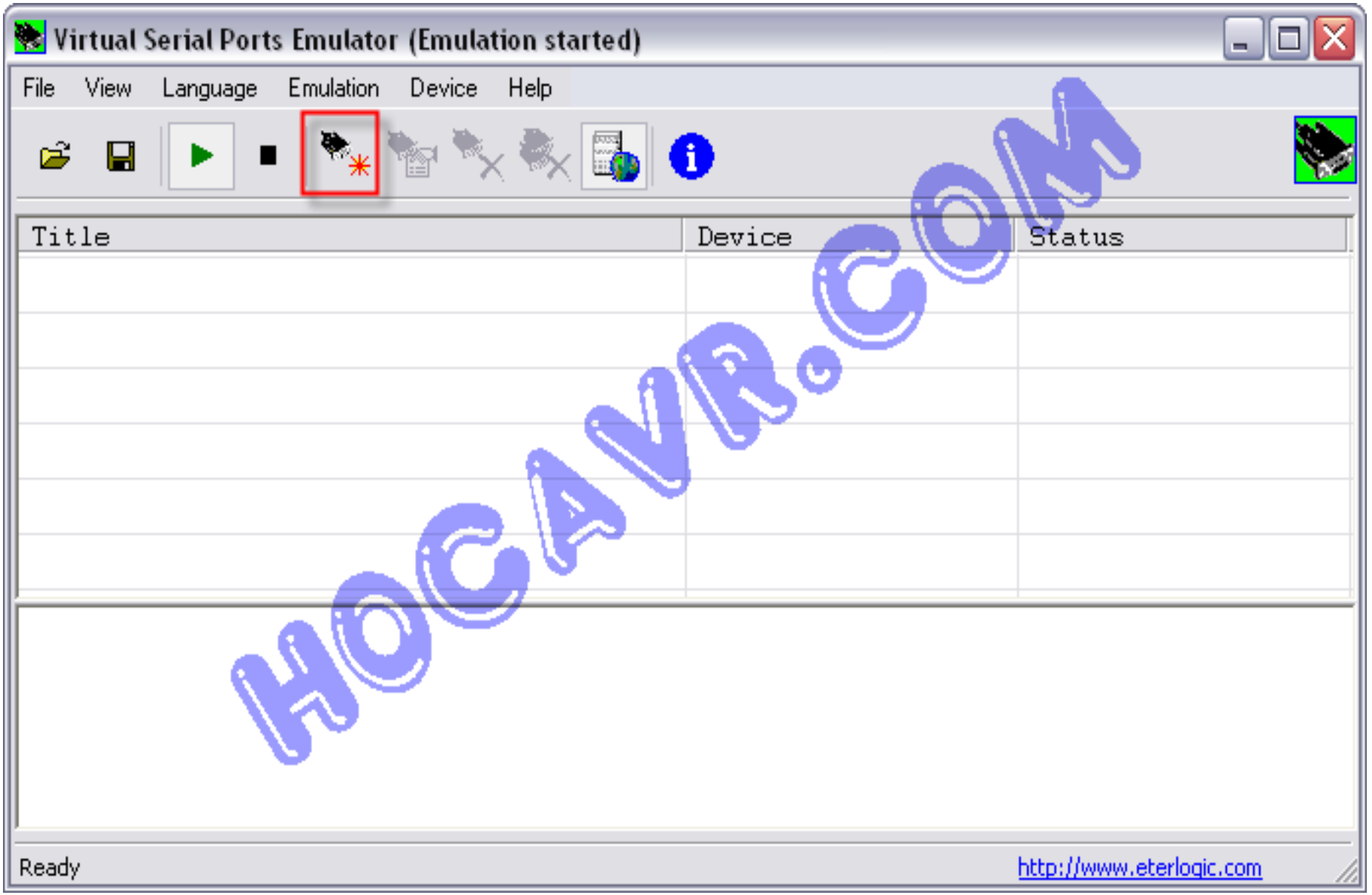
KẾT NỐI VỚI MÁY TÍNH



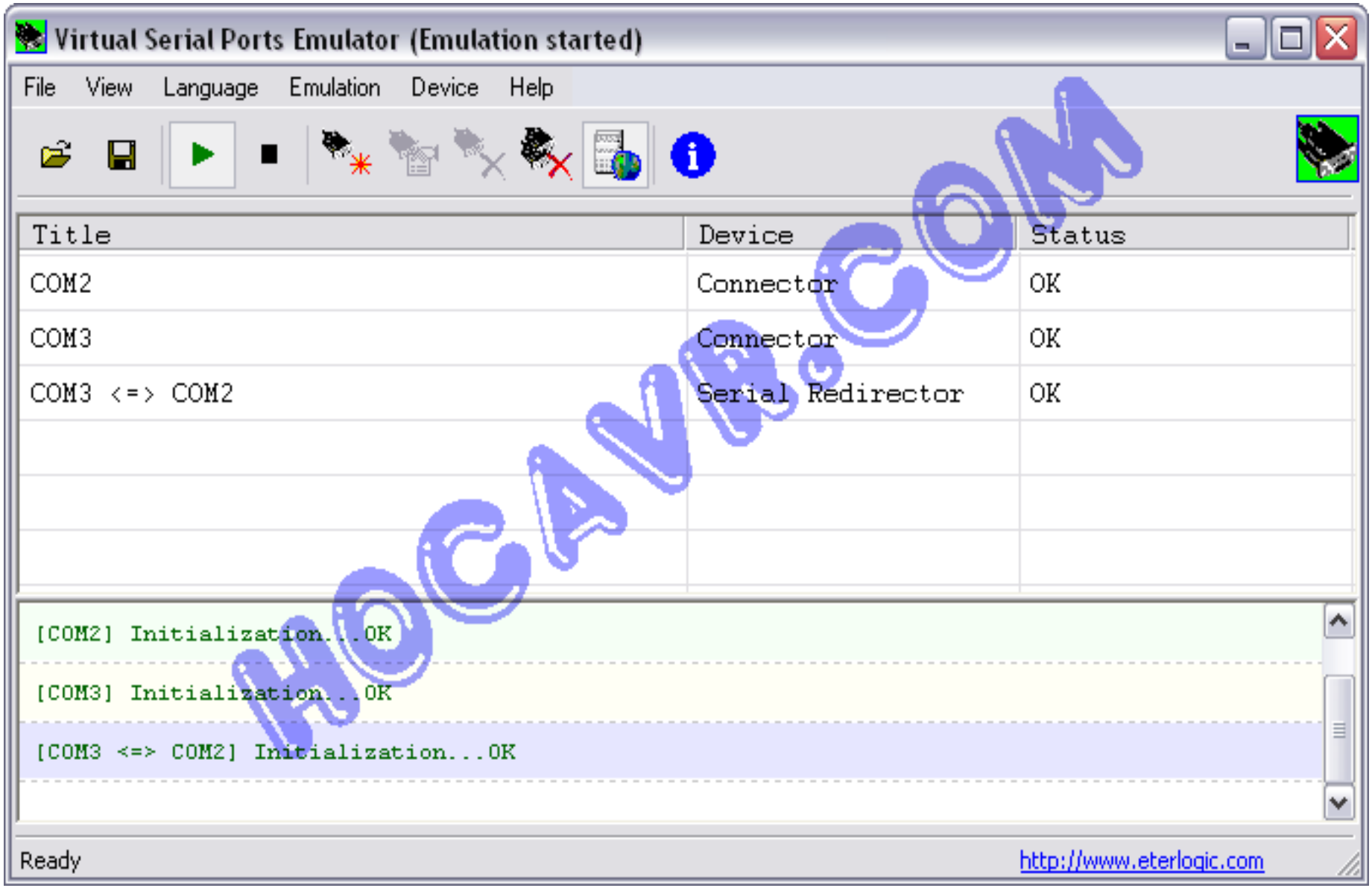
MỨC TÍN HIỆU RS232

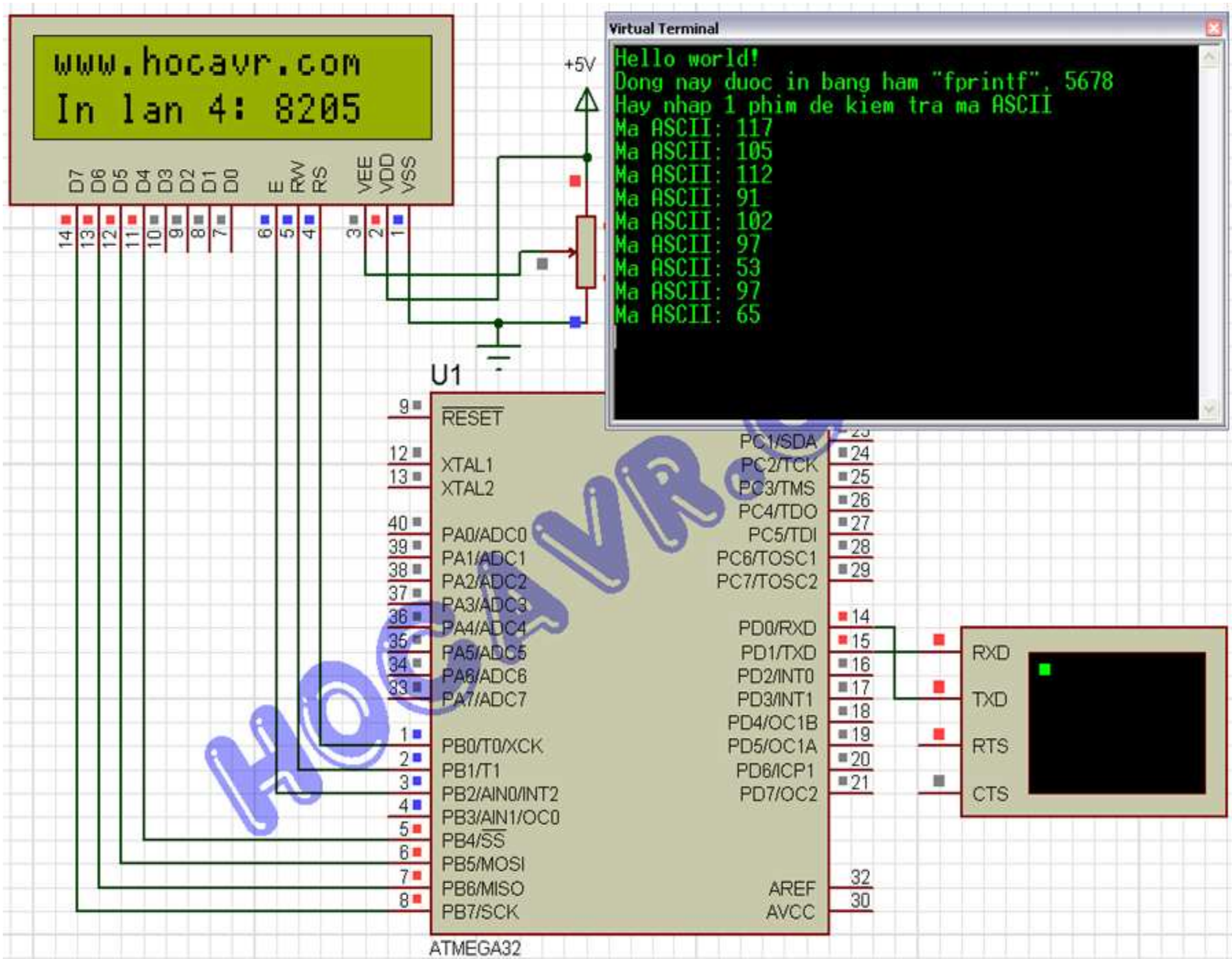


CỔNG COM ẢO TRÊN MÁY TÍNH

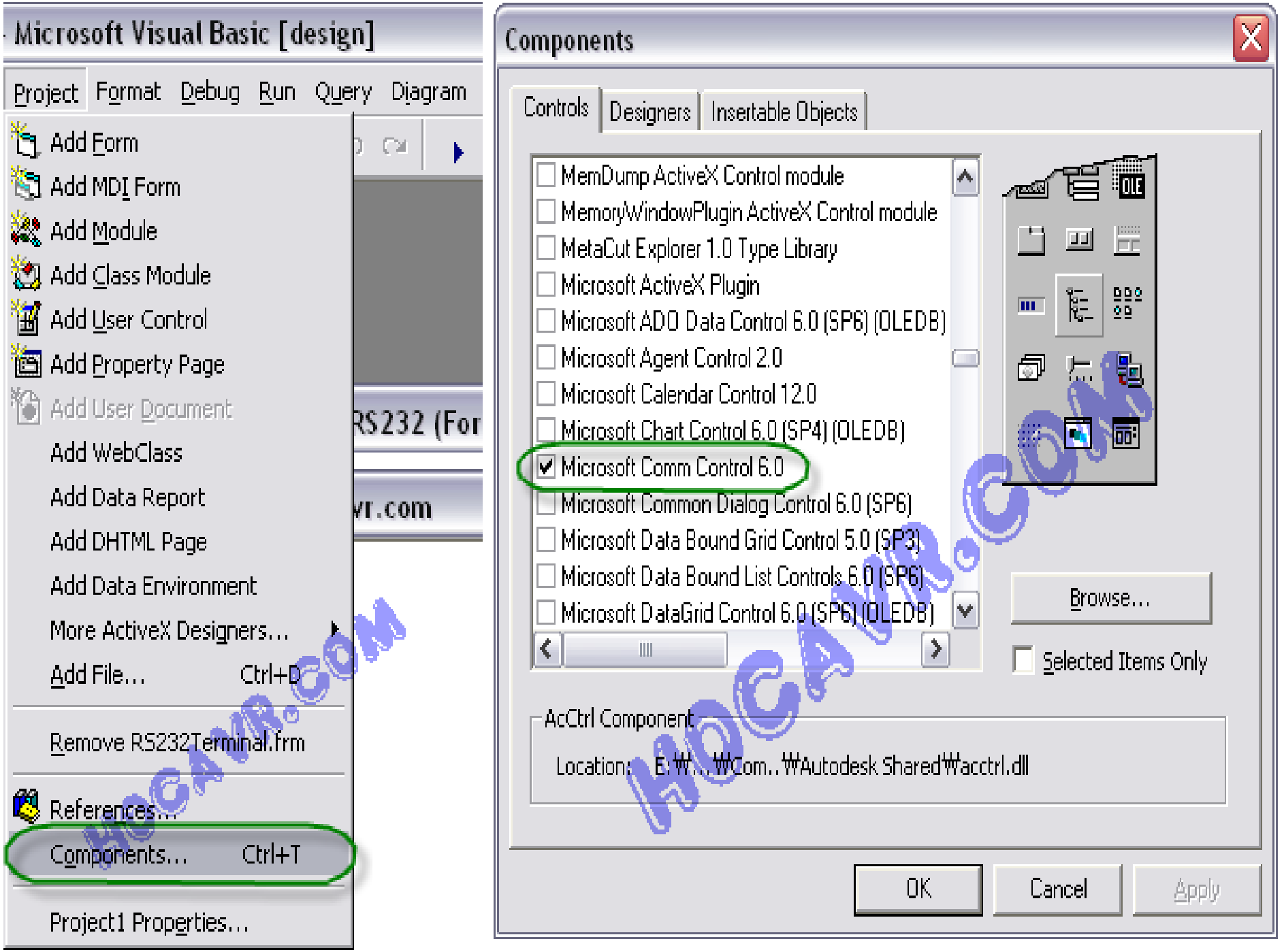


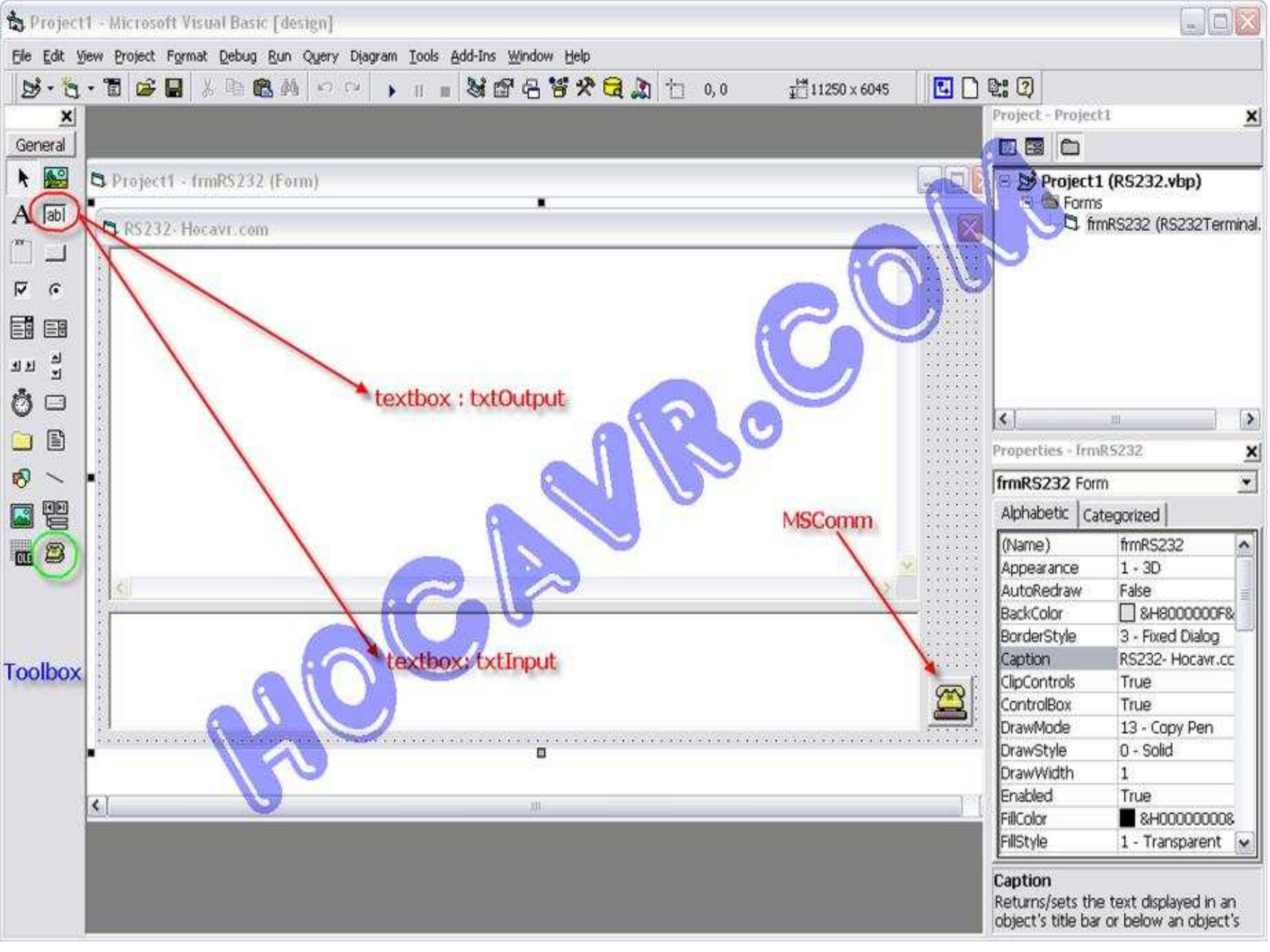
KẾT NỐI CÁC COM ẢO




www.hocavr.com
In lan 4: 8205
D7 D6 D5 D4 D3 D2 D1 D0 E RW RS VEE VDD VSS
14 13 12 11 10 9 8 7 6 5 4 3 2 1
+5V
Virtual Terminal
Hello world!
Dong nay duoc in bang ham "fprintf", 5678
Hay nhap 1 phim de kiem tra ma ASCII
Ma ASCII: 117
Ma ASCII: 105
Ma ASCII: 112
Ma ASCII: 91
Ma ASCII: 102
Ma ASCII: 97
Ma ASCII: 53
Ma ASCII: 97
Ma ASCII: 65
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA32
RXD
TXD
RTS
CTS

```c
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>
#include "myLCD.h"
//chuong trinh con phat ky tu uart (ham co ban)
void uart_char_tx(unsigned char chr){
    if (chr == '\n')  uart_char_tx('\r');
    while (bit_is_clear(UCSRA,UDRE)) {}; //cho den khi bit UDRE=1
    UDR=chr;
}
//tao 2 "FILE" (stream) tuong ung voi LCD va UART
static FILE lcdstd = FDEV_SETUP_STREAM(putChar_LCD,NULL,_FDEV_SETUP_WRITE);
static FILE uartstd= FDEV_SETUP_STREAM(uart_char_tx, NULL,_FDEV_SETUP_WRITE);
int main(void){
        //UART: set baud, 38.4k ung voi f=8Mhz ------------------
        UBRRH=0;
        UBRRL=12;
        UCSRA=0x00;
        UCSRC=(1<<URSEL)|(1<<UCSZ1)|(1<<UCSZ0);
        UCSRB=(1<<RXEN)|(1<<TXEN)|(1<<RXCIE); //cho phep ngat RxD
        sei(); //cho phep ngat toan cuc
        //LCD: khoi dong va in LCD-------------
        init_LCD();
        clr_LCD();
        int x=8205;
        printf("In lan 1");
        fprintf(&lcdstd,"www.hocavr.com");
        move_LCD(2,1);
        printf("In lan 3: %i", x);
        stdout=&lcdstd;
        printf("In lan 4: %i", x);
        //UART: in ra uart ---------------------
        stdout=&uartstd;
        printf("Hello world!\n");
        fprintf(&uartstd,"Dong nay duoc in bang ham \"fprintf\", %i\n", 5678);
        printf("Hay nhap 1 phim de kiem tra ma ASCII\n");
        while(1){
        };
}
ISR(SIG_UART_RECV){ //trinh phuc vu ngat USART
        //In ma ASCII cua phim duoc nhan
        fprintf(&uartstd,"Ma ASCII: %i\n", UDR);
}
```


Microsoft Visual Basic [design]
Project Format Debug Run Query Diagram
Add Form
Add MDI Form
Add Module
Add Class Module
Add User Control
Add Property Page
Add User Document
Add WebClass
Add Data Report
Add DHTML Page
Add Data Environment
More ActiveX Designers...
Add File... Ctrl+D
Remove RS232Terminal.frm
References...
Components... Ctrl+T
Project1 Properties...
Components
Controls
Designers
Insertable Objects
MemDump ActiveX Control module
MemoryWindowPlugin ActiveX Control module
MetaCut Explorer 1.0 Type Library
Microsoft ActiveX Plugin
Microsoft ADO Data Control 6.0 (SP6) (OLEDB)
Microsoft Agent Control 2.0
Microsoft Calendar Control 12.0
Microsoft Chart Control 6.0 (SP4) (OLEDB)
Microsoft Comm Control 6.0
Microsoft Common Dialog Control 6.0 (SP6)
Microsoft Data Bound Grid Control 5.0 (SP3)
Microsoft Data Bound List Controls 6.0 (SP6)
Microsoft DataGrid Control 6.0 (SP6) (OLEDB)
Browse...
Selected Items Only
AcCtrl Component
Location: E:₩...₩Com...₩Autodesk Shared₩acctrl.dll
OK
Cancel
Apply


Project1 - Microsoft Visual Basic [design]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
General
Project1 - frmRS232 (Form)
RS232- Hocavr.com
textbox : txtOutput
textbox: txtInput
MSComm
Toolbox
Project - Project1
Project1 (RS232.vbp)
Forms
frmRS232 (RS232Terminal.
Properties - frmRS232
frmRS232 Form
Alphabetic
Categorized
(Name) frmRS232
Appearance 1 - 3D
AutoRedraw False
BackColor &H8000000F&
BorderStyle 3 - Fixed Dialog
Caption RS232- Hocavr.cc
ClipControls True
ControlBox True
DrawMode 13 - Copy Pen
DrawStyle 0 - Solid
DrawWidth 1
Enabled True
FillColor &H00000000&
FillStyle 1 - Transparent
Caption
Returns/sets the text displayed in an object's title bar or below an object's

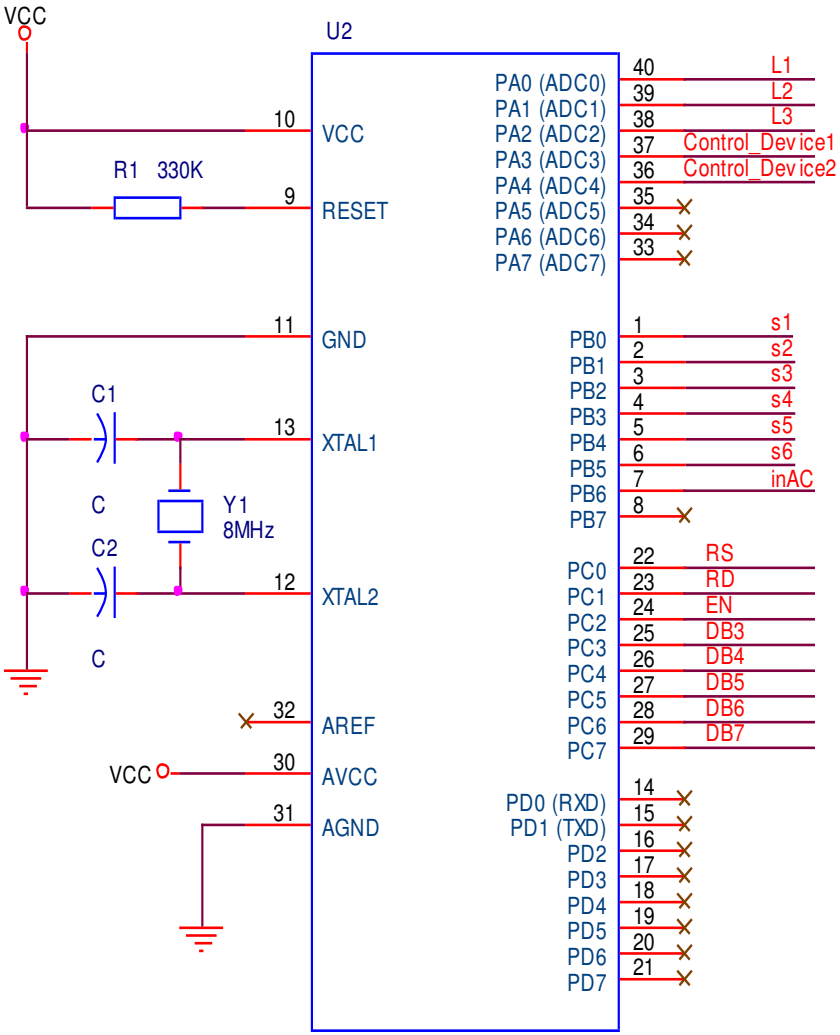
FORM CHÍNH

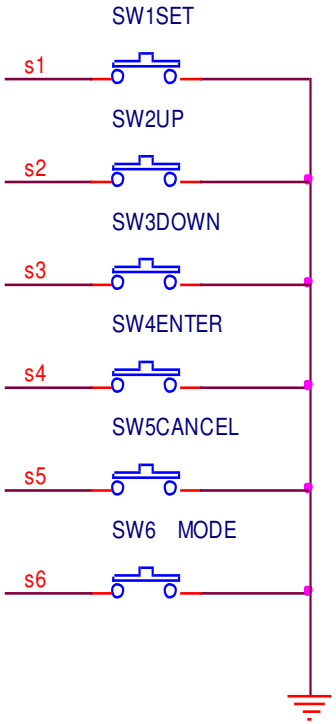
```
1  Private Sub Form_Load()
2     MSComm1.CommPort = 3
3     MSComm1.Settings = "38400,N,8,1"
4     Me.MSComm1.RThreshold = 1
5     MSComm1.InputLen = 1
6     MSComm1.PortOpen = True
7  End Sub
```

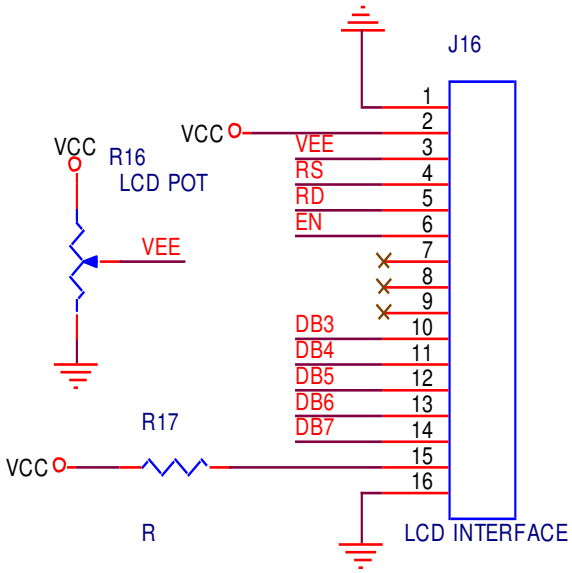
```
1  Private Sub MSComm1_onComm()
2     Dim InputText As String
3     If Me.MSComm1.CommEvent = comEvReceive Then
4        InputText = MSComm1.Input
5        txtOutput.Text = txtOutput.Text + InputText
6        txtOutput.SelStart = Len(txtOutput.Text)
7     End If
8  End Sub
```

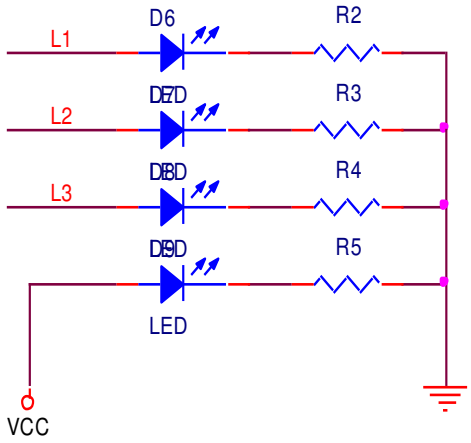
```
1  Private Sub txtInput_KeyPress(KeyAscii As Integer)
2     Me.MSComm1.Output = Chr(KeyAscii)
3  End Sub
```

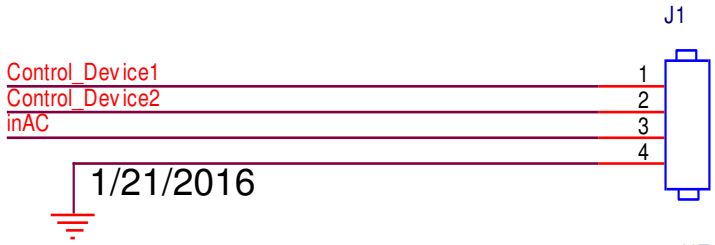

AIR-CONDITIONALER CONTROL SYSTEM
POWER
PROGRAM
DEVICE1
DEVICE2
SET
UP
ENTER
MODE
DOWN
CANCEL

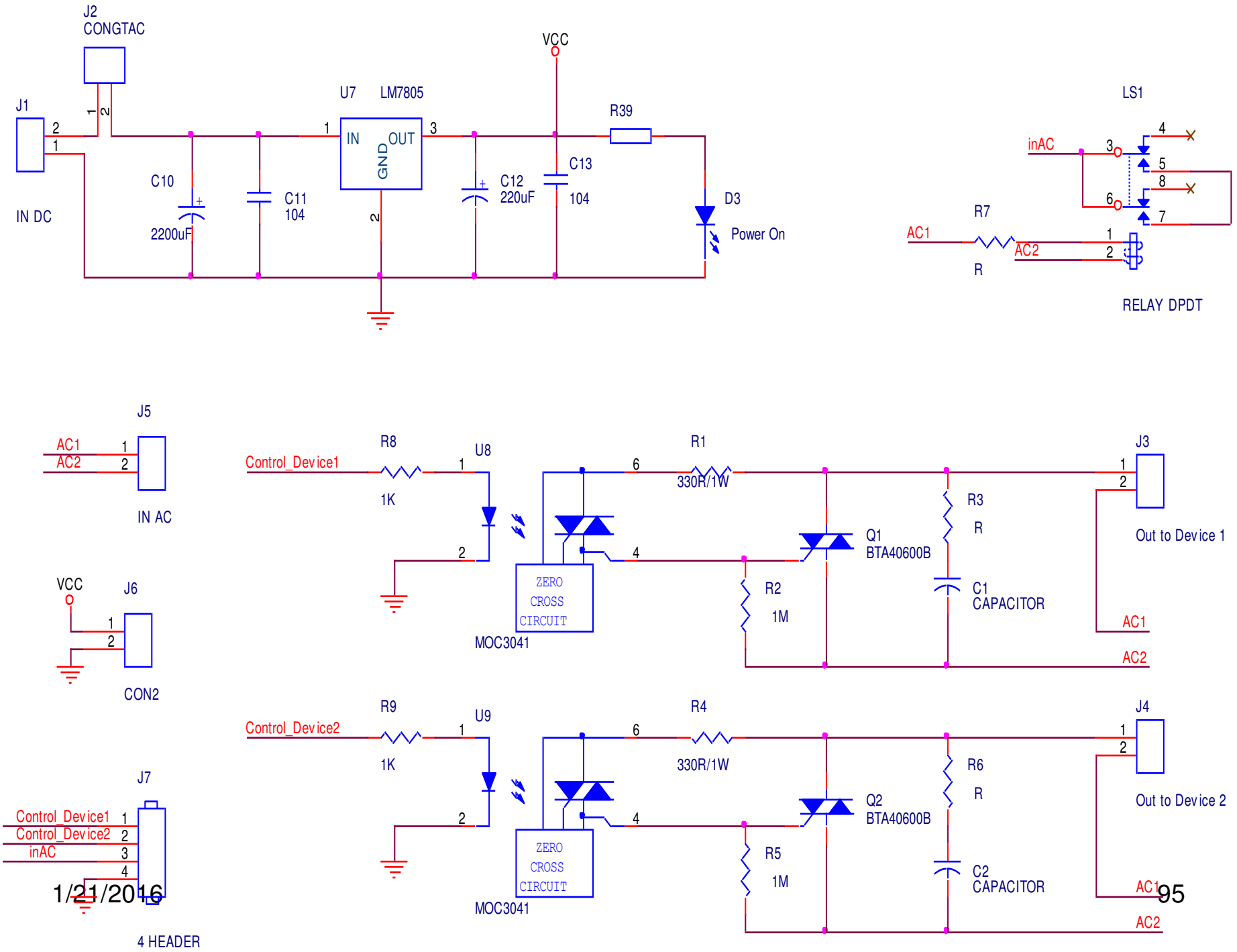

VCC
U2
10
VCC
R1 330K
9
RESET
11
GND
C1
13
XTAL1
C
Y1
8MHz
C2
12
XTAL2
C
32
AREF
VCC
30
AVCC
31
AGND
PA0 (ADC0)
PA1 (ADC1)
PA2 (ADC2)
PA3 (ADC3)
PA4 (ADC4)
PA5 (ADC5)
PA6 (ADC6)
PA7 (ADC7)
40 L1
39 L2
38 L3
37 Control_Device1
36 Control_Device2
35
34
33
PB0 1 s1
PB1 2 s2
PB2 3 s3
PB3 4 s4
PB4 5 s5
PB5 6 s6
PB6 7 inAC
PB7 8
PC0 22 RS
PC1 23 RD
PC2 24 EN
PC3 25 DB3
PC4 26 DB4
PC5 27 DB5
PC6 28 DB6
PC7 29 DB7
PD0 (RXD) 14
PD1 (TXD) 15
PD2 16
PD3 17
PD4 18
PD5 19
PD6 20
PD7 21
ATMega16


SW1SET
s1
SW2UP
s2
SW3DOWN
s3
SW4ENTER
s4
SW5CANCEL
s5
SW6 MODE
s6


J16
VCC
R16
LCD POT
VEE
VCC
R17
R
VEE
RS
RD
EN
DB3
DB4
DB5
DB6
DB7
LCD INTERFACE


L1
L2
L3
D6
R2
R3
R4
R5
LED
VCC


J1
Control_Device1
Control_Device2
inAC
1
2
3
4
4 HEADER


J2
CONGTAC
J1
IN DC
C10
2200uF
C11
104
U7 LM7805
IN
GND
OUT
C12
220uF
VCC
C13
104
R39
D3
Power On
LS1
inAC
R7
AC1
AC2
R
RELAY DPDT
J5
AC1
AC2
IN AC
VCC
J6
CON2
R8
Control_Device1
1K
U8
ZERO
CROSS
CIRCUIT
MOC3041
R1
330R/1W
Q1
BTA40600B
R3
R
C1
CAPACITOR
R2
1M
J3
Out to Device 1
AC1
AC2
R9
Control_Device2
1K
U9
ZERO
CROSS
CIRCUIT
MOC3041
R4
330R/1W
Q2
BTA40600B
R6
R
C2
CAPACITOR
R5
1M
J4
Out to Device 2
AC1
AC2
J7
Control_Device1
Control_Device2
inAC
4 HEADER

Bài 9: Timer