

CHUYỂN ĐỔI MÁY TÍNH THÀNH DAO ĐỘNG KÝ SỐ HAI KÊNH

Nguyễn Trường An, Phạm Thị Thu Phương, Huỳnh Hữu Thuận, Nguyễn Hữu Phương

Trường Đại học Khoa học Tự nhiên, ĐHQG – HCM

(Bài nhận ngày 01 tháng 12 năm 2005, hoàn chỉnh sửa chữa ngày 16 tháng 05 năm 2006)

TÓM TẮT: Dao động ký hai kênh (Two-channel oscilloscope) là thiết bị đo điện tử quan trọng và phổ biến, nó hiển thị dạng sóng và một số đặc tính khác của các tín hiệu. Có hai loại dao động ký: tương tự và số. Dao động ký số ưu việt hơn loại tương tự nhưng ít phổ biến vì giá cao. Bài báo trình bày cách thức chuyển đổi máy tính PC thành dao động ký số hai kênh, gồm phần cứng là mạch thu nhận dữ liệu tự thiết kế và phần mềm là tập nhiều chương trình viết bằng ngôn ngữ Visual C⁺⁺.

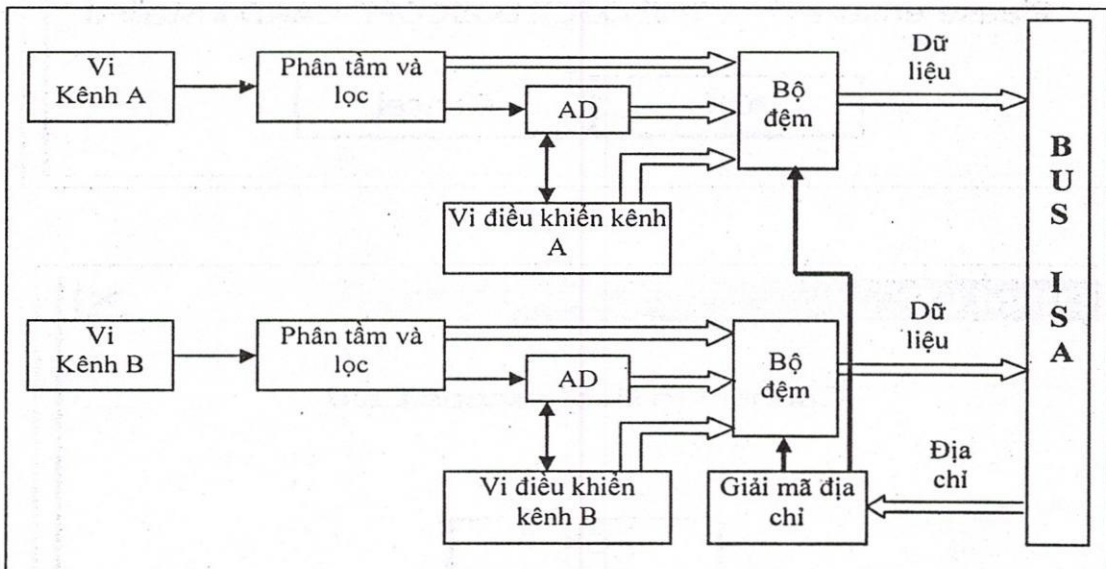
1. GIỚI THIỆU

Dao động ký hai kênh (Two-channel oscilloscope) là thiết bị rất cần thiết và khá phổ biến trong các phòng thí nghiệm điện tử và các phòng thí nghiệm khác. Nhưng dao động ký số hai kênh ít phổ biến hơn vì giá cao dù có nhiều ưu điểm, đặc biệt là khả năng đo, lưu trữ và xử lý dữ liệu. Mặt khác máy tính PC đã sẵn có khả năng cao về xử lý, lưu trữ và hiển thị. Do đó vấn đề là thiết kế mạch thu nhận dữ liệu (DAQ) giao tiếp với máy tính và trang bị các phần mềm cần thiết. Đã có nhiều card DAQ và phần mềm như vậy, điển hình nhất có lẽ là thiết bị và phần mềm LabView của hãng National Instruments (Mỹ). Ở đây, để giảm giá thành xuống mức thấp nhất chúng tôi tự nghiên cứu phát triển phần cứng và phần mềm.

Phần còn lại của bài báo như sau. Mục 2: Thu nhận dữ liệu, mục 3: Hiển thị dạng sóng và dữ liệu đo, mục 4: Tính toán tham số của tín hiệu, mục 5: Phân tích tần phổ, mục 6: Lưu trữ dữ liệu, và mục 7: Kết luận. Do giới hạn không gian nên nhiều giải thuật chương trình và hình minh họa bị bỏ bớt nhiều.

2. THU NHẬN DỮ LIỆU

Hình 1 là sơ đồ khối trình bày nguyên lý của card thu nhận dữ liệu (DAQ) để chuyển đổi máy tính thành dao động ký số hai kênh. Việc giao tiếp thực hiện qua bus ISA



Hình 1: Sơ đồ khối nguyên lý của card thu nhận dữ liệu

2.1 Mô tả mạch

Đầu tiên, tín hiệu vào ở kênh A hoặc B sẽ đến mạch phân tầm tự động (autoranging), phân ra các tầm đo khác nhau và qua lọc thông thấp. Mạch phân tầm là mạch khuếch đại có độ lợi thay đổi tùy theo khoảng biên độ tín hiệu vào. Việc này là cần thiết để tăng độ phân giải trong lúc phải phù hợp với độ dài dữ liệu. Biên độ tín hiệu vào V_i được phân thành 3 tầm khác nhau như sau:

Tầm	Khoảng biên độ	Mã số tầm
X5	$V_i < 0,5V$	0 0
X1	$0,5V < V_i < 2,5V$	0 1 1 0
+5	$2,5V < V_i < 12,5V$	1 1

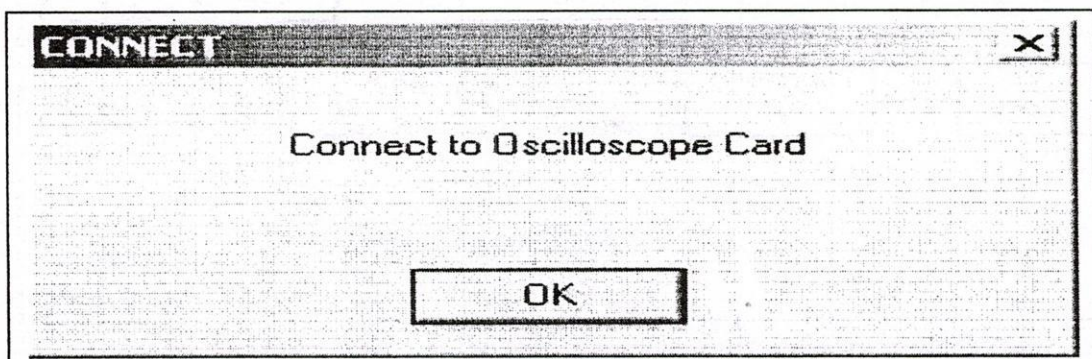
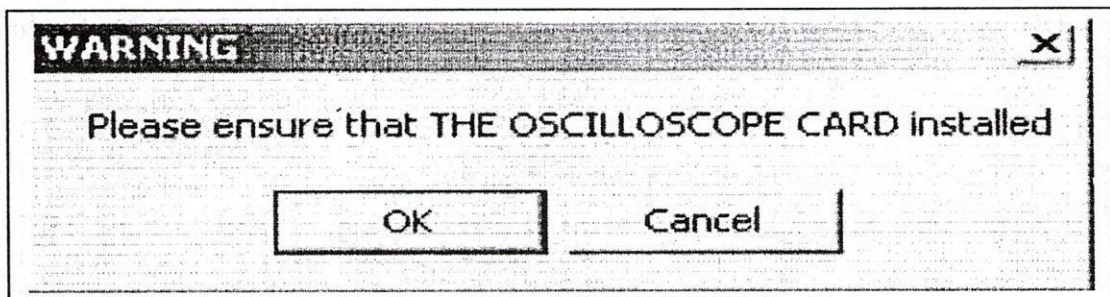
Tín hiệu ra khỏi mạch phân tầm sẽ qua mạch lọc thông thấp Butterworth bậc 5. Sau mạch lọc là mạch cộng để cộng thêm vào tín hiệu điện áp DC 2,5 Volt (gọi mạch chuyển mức điện áp một chiều – DC level shifter), lý do là mạch chuyển đổi tương tự – sang số (ADC) AD7875 thuộc loại có điện áp vào đơn cực 0 – 5V.

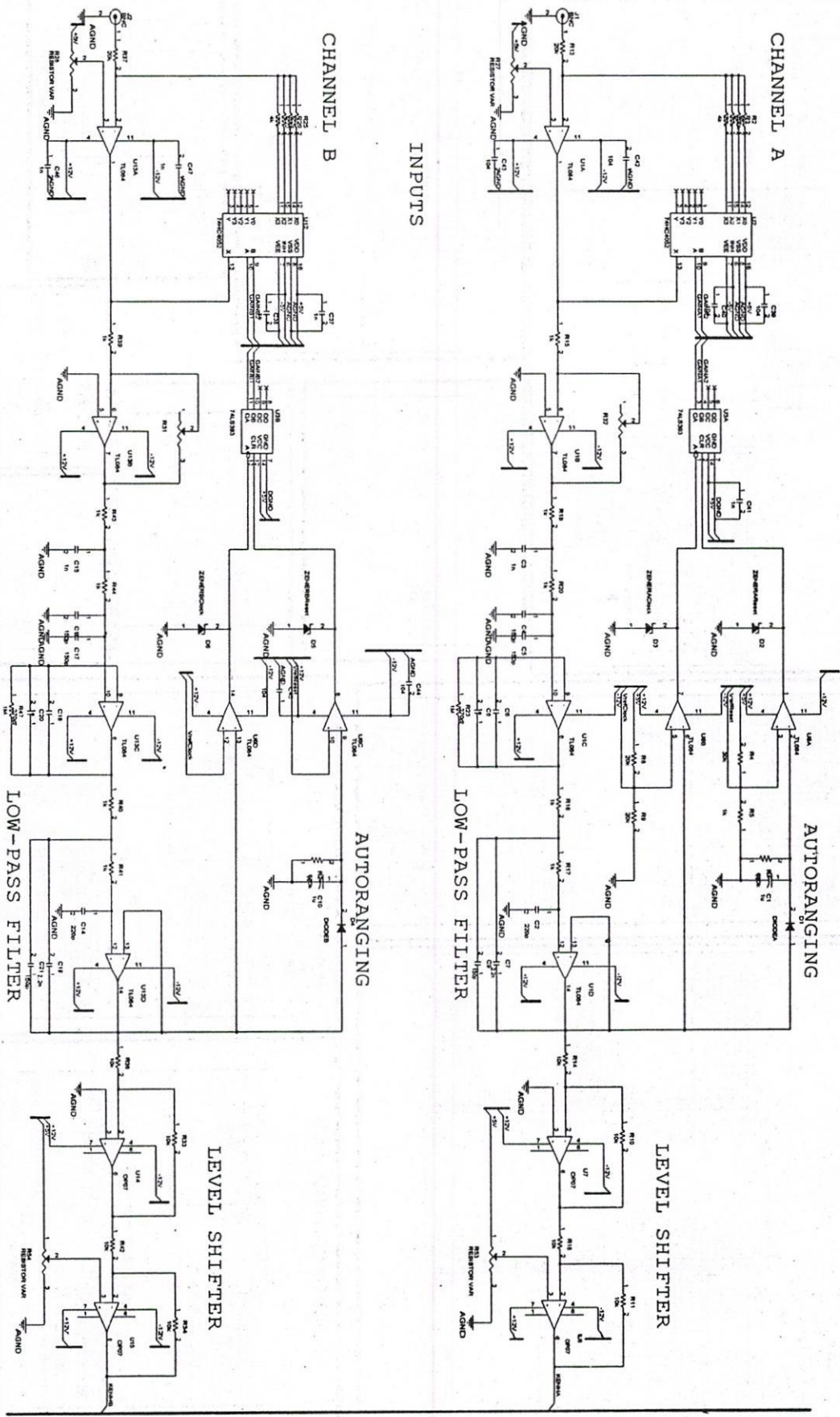
Sau đó, tín hiệu ngõ ra sẽ được AD7875 chuyển đổi sang số, đây là mạch 12 bit, tốc độ 100 kS/s và, đặc biệt, đã tích hợp mạch Track-and-Hold. AD7875 được điều khiển bởi vi điều khiển AT89C2051. Dữ liệu ra cùng với mã số phân tầm (2 bit) và bit trạng thái từ vi điều khiển được đưa đến bộ đệm để vào máy tính qua bus ISA. Bộ giải mã địa chỉ xác định địa chỉ cho các bộ đệm, đây là địa chỉ đọc dữ liệu.

Hình 2a và 2b là sơ đồ mạch đầy đủ.

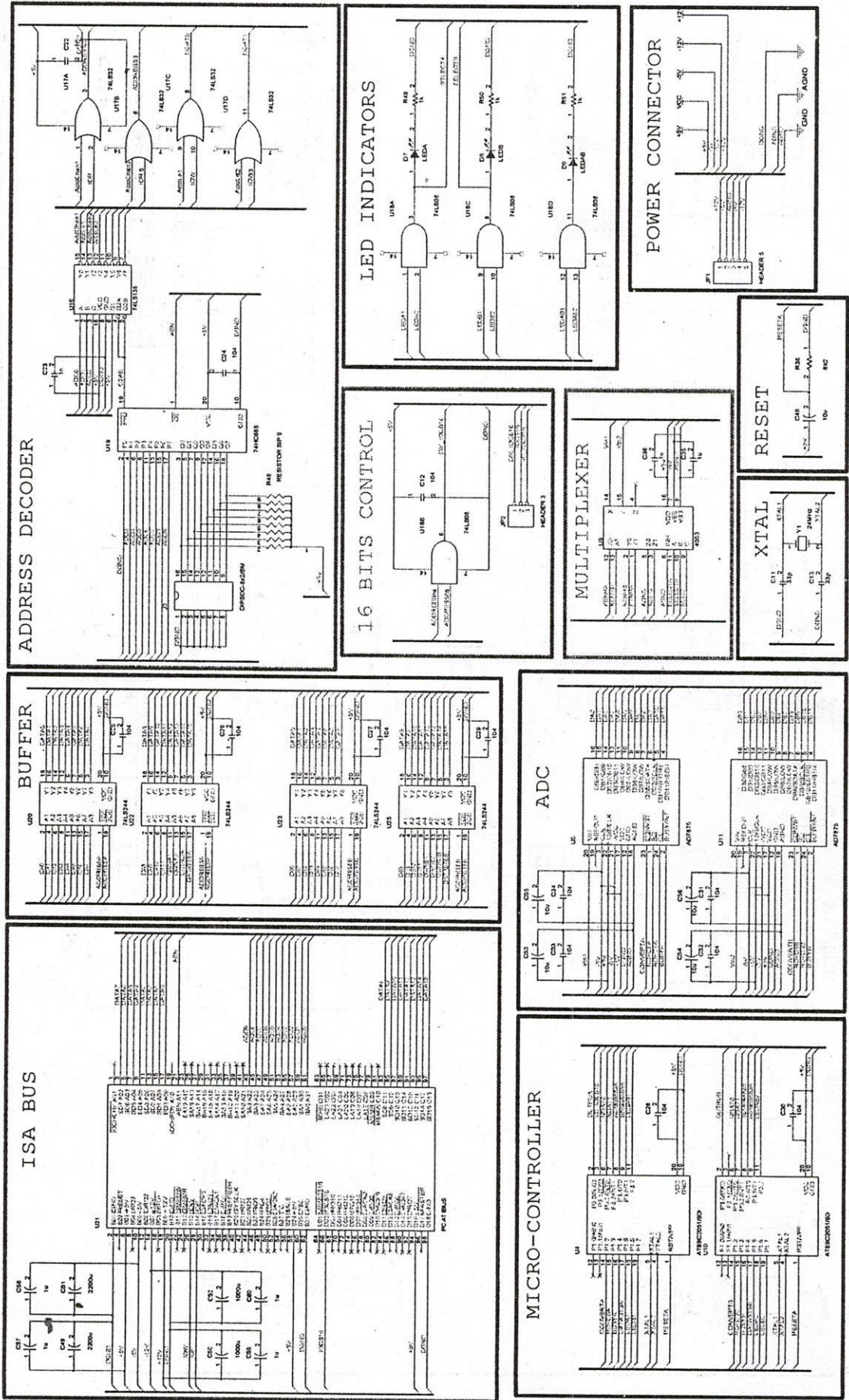
2.2 Phần mềm giao tiếp

Trước khi vào chương trình, phần mềm yêu cầu xác nhận đã có phần cứng gắn vào bus ISA chưa. Nếu đã có phần cứng thì bấm OK để vào chương trình chính.



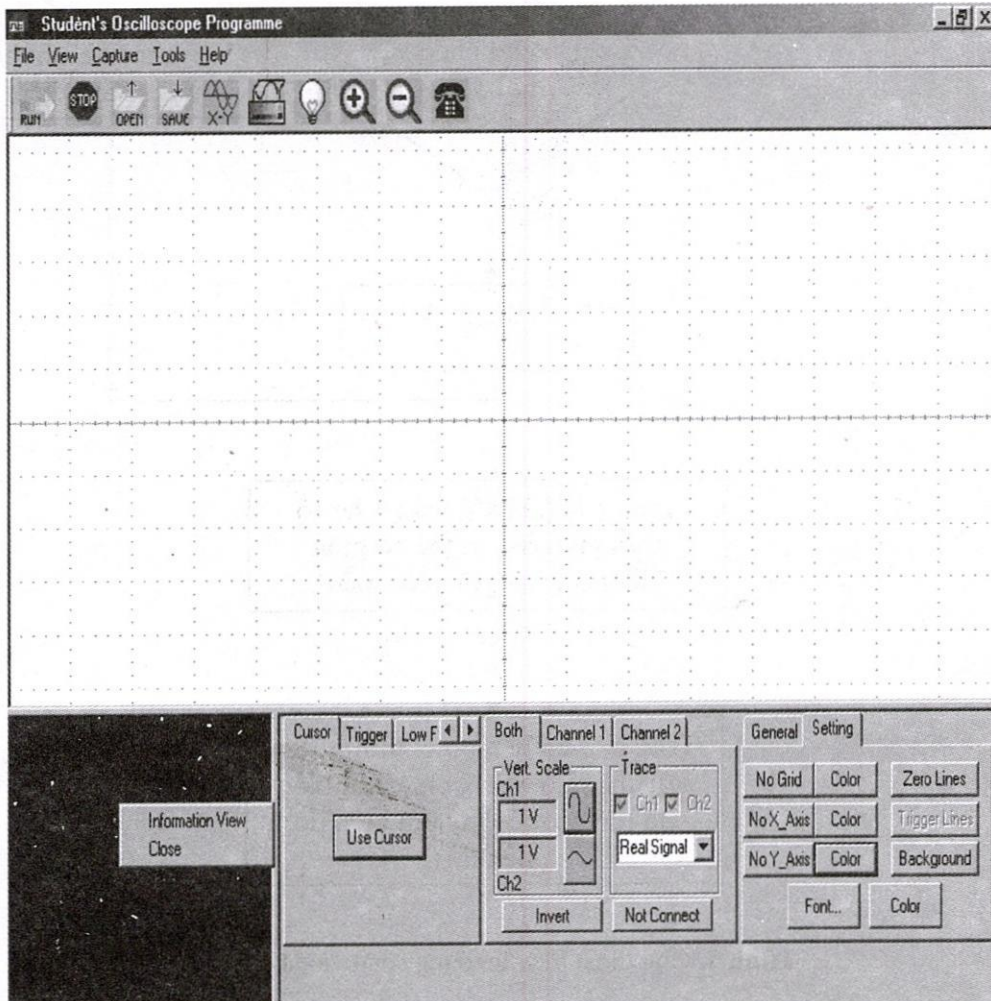


Hình 2a: Khối xử lý số và giao tiếp máy tính



Hình 2b. Khối xử lý số và giao tiếp máy tính của mạch thu nhận dữ liệu

Hình 3 là giao diện Oscilloscope.



Hình 3 Giao diện Oscilloscope

- **Nút Run:**

Khi muốn chạy chương trình, xem dữ liệu trên màn hình, ta bấm nút Run, hoặc vào menu Capture/Run.

- **Nút Stop:**

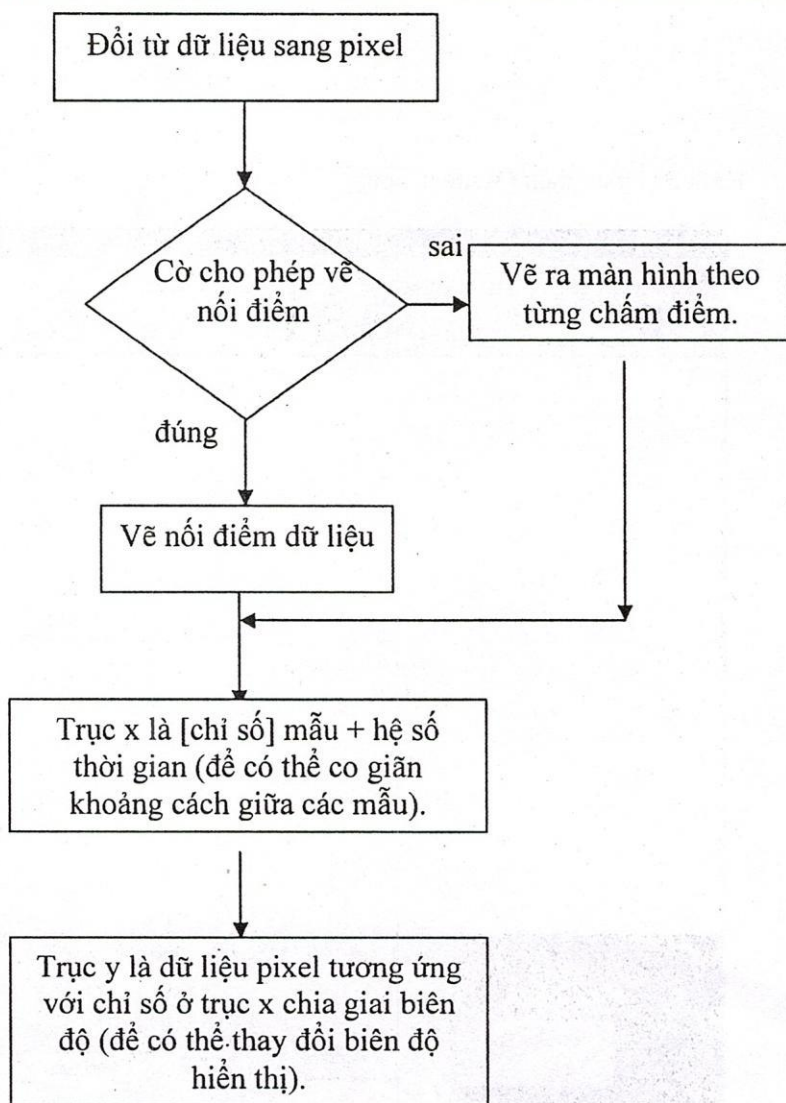
Khi muốn dừng việc lấy dữ liệu vào, bấm nút Stop hoặc vào menu Capture/Stop.

- **Lấy dữ liệu:**

Dữ liệu của kênh 1 và kênh 2 được nhập vào hai mảng dữ liệu khác nhau.

3. HIỂN THỊ DẠNG SÓNG VÀ DỮ LIỆU ĐO

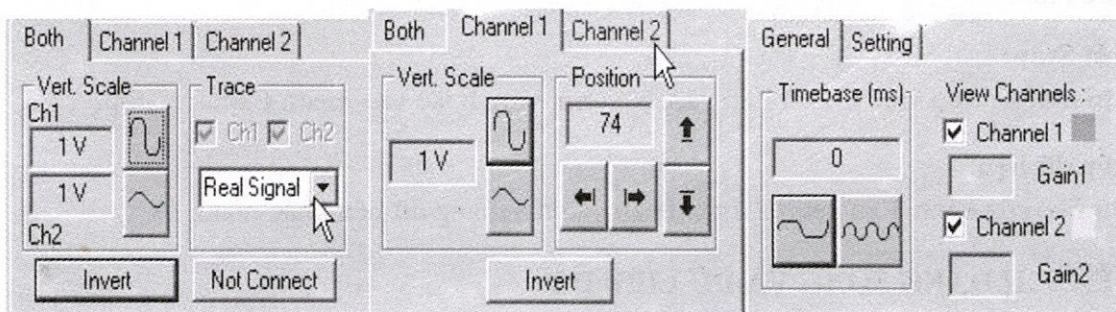
Việc vẽ tín hiệu ra màn hình chỉ là việc đưa dữ liệu tương ứng nhận được ra một vị trí xác định trên màn hình. Để điều khiển, kiểm soát được biên độ, thời gian, ta phải chuyển dữ liệu vào các giai xác định. Hình 4 là giải thuật.



Hình 4. Giải thuật hiển thị dạng sóng và dữ liệu đo

3.1 Giai dọc và ngang

Oscilloscope mô phỏng phải có hầu hết các chức năng của một Oscilloscope thật sự, trong đó có vấn đề về giai số dọc (Vert.scale) gốc thời gian (Timebase), vị trí dạng sóng (Position), độ lợi (Gain), chọn kênh (View channel) . . . Sau đây là giao diện điều khiển:

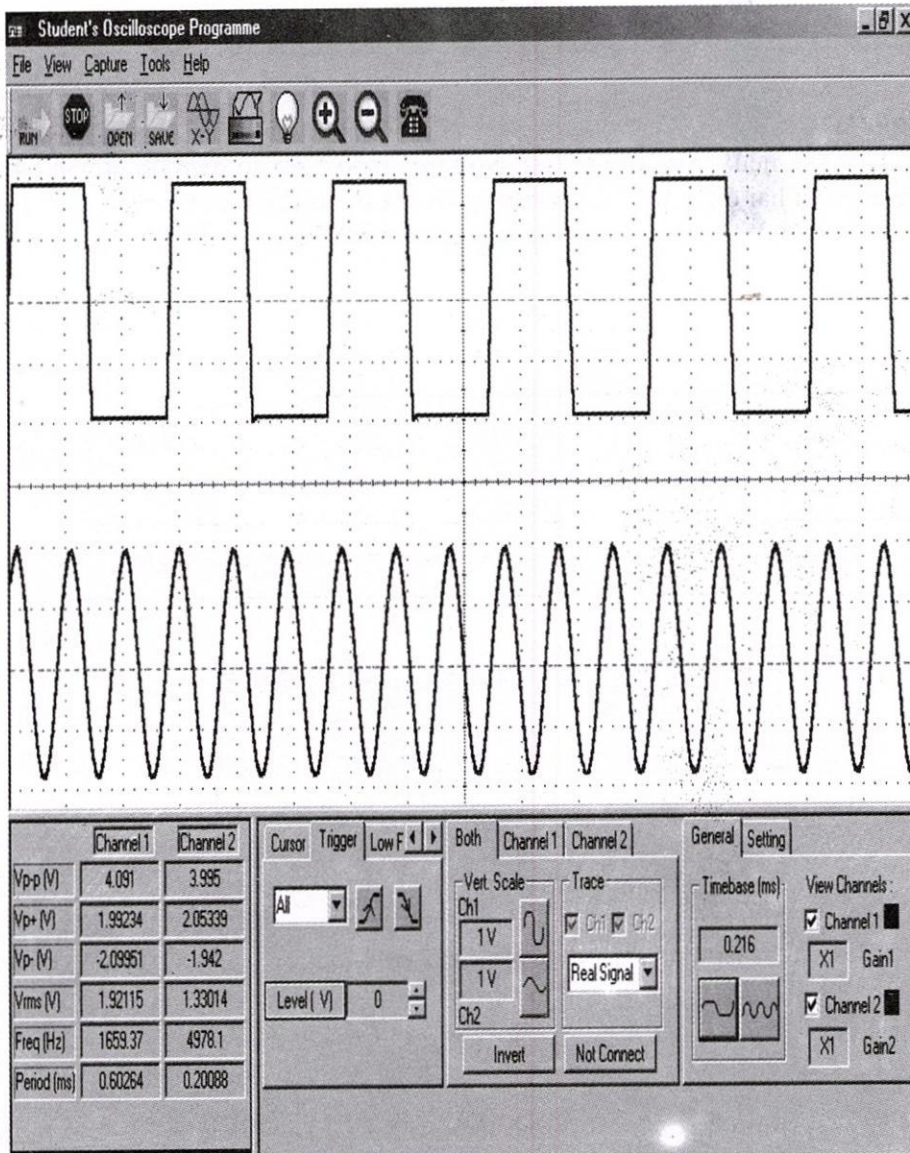


Để thay đổi giai biên độ (giai dọc) ta thay đổi giá trị biến *giaibiendo* trong chương trình; để thay đổi giai thời gian ta thay đổi giá trị biến *giaichuky*. Còn biến *vitri* là để thay đổi vị trí của hình vẽ (có thể di chuyển lên xuống), biến $k=1$ hoặc $k=-1$ để đảo dạng sóng tín hiệu (Invert).

3.2 Vẽ dạng sóng ra màn hình

Vẽ dạng sóng là hiển thị dữ liệu ra màn hình dựa vào các cách chọn lựa của người sử dụng: chọn vẽ tín hiệu thực, vẽ tín hiệu sau khi đã được làm trơn theo các cách khác nhau (xem sau). Người sử dụng còn có thể chọn các giai biên độ gốc thời gian và nhiều thuộc tính khác tùy ý như đã trình bày ở trên.

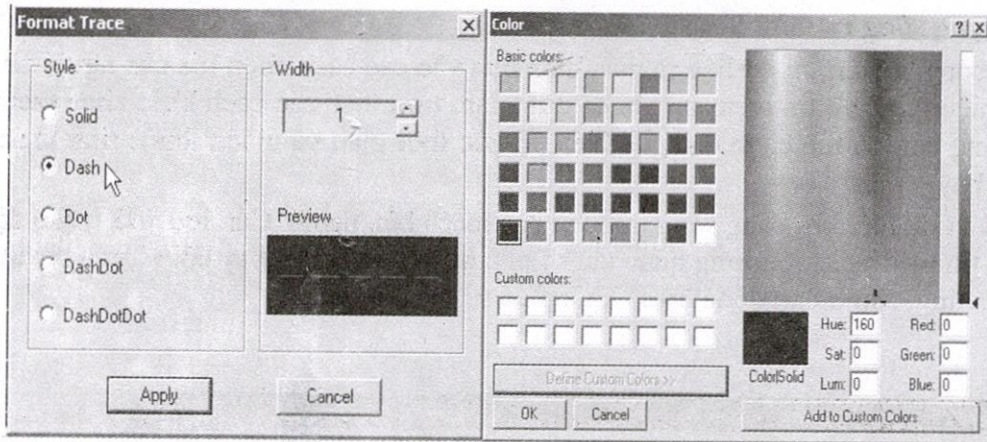
Hình 5 là ví dụ hiển thị dạng sóng tín hiệu của mạch bên ngoài. Cần lưu ý là ở dao động ký tương tự hai tín hiệu phải có tương quan tần số phù hợp (bằng nhau hay bội số của nhau) thì sự hiển thị mới vững, còn ở dao động ký số không có sự ràng buộc này.



Hình 5. Ví dụ hiển thị dạng sóng tín hiệu (dạng sóng vuông và sin có tần số độc lập)

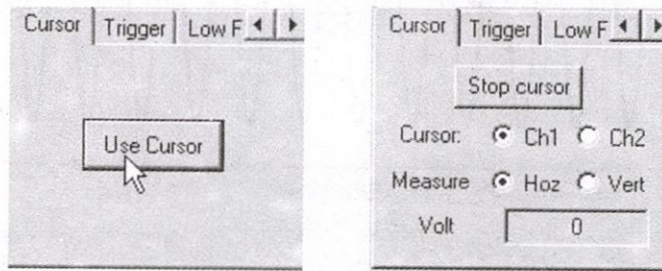
3.3 Chọn màu

Chương trình cho phép thay đổi nét vẽ, màu sắc, màu nền . . . Đây là các thuộc tính tính về đồ họa trong Visual C⁺⁺. Sau đây là màn hình chọn nét vẽ và màu:



3.4 Sử dụng con trỏ

Để có thể biết được tín hiệu tại một vị trí xác định trên màn hình có biên độ là bao nhiêu, khoảng cách thời gian giữa hai điểm tín hiệu là bao nhiêu, ta dùng con trỏ (Cursor). Giải thuật của phần sử dụng con trỏ chuột (kết hợp được với bàn phím) trình bày ở hình 6.



Hình 6. Giải thuật sử dụng cursor

3.5 Bộ định thời

Bộ định thời (Timer) trong Visual C⁺⁺ chính là hàm SetTimer. Dùng bộ định thời để lặp lại việc nhận dữ liệu và vẽ lên màn hình sau một khoảng thời gian xác định mà không cần phải thông qua thao tác hay sự kiện.

3.6 Phóng to, thu nhỏ

Để nhìn thấy rõ hơn dữ liệu hiển thị trên màn hình, ta phóng to hình lên bằng cách bấm nút Zoom in, đây là sự kết hợp giữa việc tăng chu kỳ ở trục x và tăng biên độ ở trục y, tương ứng là việc tăng giai trên màn hình.

Tương tự với phóng to, khi muốn thu nhỏ hình, ta bấm nút Zoom out, đây là sự kết hợp giữa việc giảm chu kỳ ở trục x và giảm biên độ ở trục y, tương ứng là việc giảm giai trên màn hình.

3.7 Hình Lissajous

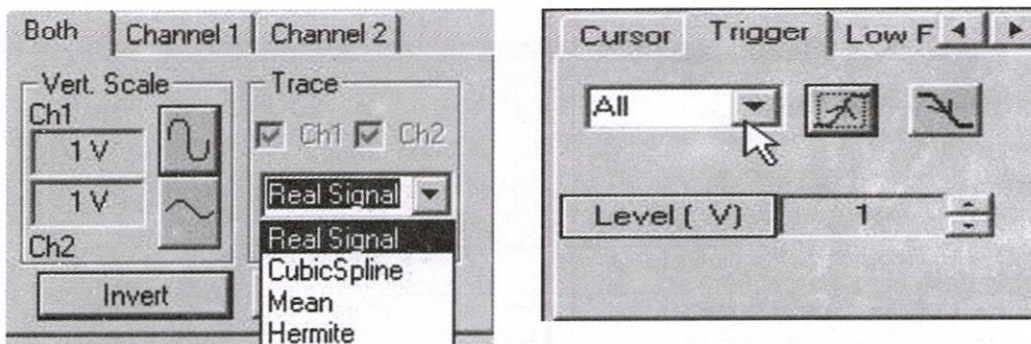
Chức năng của Lissajous là để thấy sự lệch pha của hai tín hiệu sin cùng biên độ, cùng tần số. Hai tín hiệu cùng pha: đường Lissajous là đường thẳng có hệ số góc là 1. Hai tín hiệu lệch pha 90° : đường Lissajous là đường tròn. Hai tín hiệu lệch pha 180° : đường Lissajous là đường thẳng có hệ số góc là -1. Hình Lissajous cũng là cách cổ điển để so sánh tần số hai tín hiệu sin. Vẽ đường Lissajous là vẽ biên độ của tín hiệu này theo biên độ tín hiệu kia.

3.8 Làm trơn dạng sóng

Một vấn đề then chốt của Oscilloscope hóa máy tính là phục hồi dạng sóng tín hiệu tương tự từ các trị lấy mẫu của nó (cung cấp bởi mạch ADC) như thế nào để cho dạng sóng trơn tru. Cách nội suy làm trơn tùy thuộc vào dạng sóng (sin, vuông . . .) và tần số tín hiệu (nếu xử lý thời gian thực). Hàm fit (làm khớp) là hàm nội suy làm trơn dạng sóng.

Sau đây là một số kiểu làm trơn.

- **Lấy trung bình:** Lấy trung bình qua một số điểm mẫu, thường là 3 hay 5, nói cách khác là cho tín hiệu qua một lọc trung bình di chuyển (moving average filter) bậc thấp.
- **Nội suy Spline bậc ba:** Phương pháp biểu diễn gần đúng các hàm số bằng những đa thức trên từng đoạn nhỏ rồi ghép lại sao cho tại các điểm nối thỏa điều kiện cho trước gọi là phương pháp Spline. Thường dùng là Spline bậc ba (cubic Spline), là một công thức nội suy sao cho làm trơn đạo hàm bậc nhất và liên tục tới đạo hàm bậc hai trong một khoảng tại biên của nó.
- **Nội suy Hermite:** Ở các nội suy Hermite ta dùng một đa thức bậc cao sao cho đa thức này và các đạo hàm bậc nhất, bậc hai, . . . có các giá trị đã biết tại điểm đầu, điểm cuối và các điểm trung gian. Màn hình bên trái cho thấy sự lựa ba kiểu làm trơn:



Việc làm trơn dạng sóng quá đáng sẽ làm giảm các biến động chuyển tiếp nhanh mà đôi khi rất cần thiết. Do đó phải tùy dạng tín hiệu vào mà chọn cách phù hợp.

3.9 Trigger

Mục đích của Trigger là hiển thị dữ liệu từ vị trí mức Trigger (Trigger level) đã chọn. Nếu dữ liệu tuần hoàn thì các lần vẽ lặp lại trùng nhau nên ta chỉ thấy một dạng sóng đứng yên (thuận tiện cho việc quan sát). Vị trí dữ liệu trong mảng có giá trị bằng hoặc gần mức Trigger

đã chọn được vẽ ra màn hình từ vị trí đó trở đi (cho đến hết số chu kỳ đã định). Cũng giống như Oscilloscope thật sự, ta có sự chọn lựa Trigger là +Slope (đốc lên) hay -Slope (đốc xuống), ngoài sự chọn mức. Màn hình ở trên (bên phải) cho thấy các kiểu trigger.

4. TÍNH TOÁN CÁC THAM SỐ CỦA TÍN HIỆU

Sự thuận lợi của Oscilloscope số (ở đây là máy tính dùng như Oscilloscope số) là có thể tính toán và hiển thị nhiều tham số (đặc tính) của tín hiệu như: chu kỳ, tần số, trị đỉnh - đỉnh...

4.1 Chu kỳ, tần số

Trước tiên tìm giải thuật đo chu kỳ tín hiệu tuần hoàn, sau đó suy ra tần số.

4.2 Trị hiệu dụng (RMS)

Ở đây ta tính trị hiệu dụng dựa vào phép lấy tích phân số. Thường công thức hình thang (trường hợp đặc biệt của công thức Newton-Cotet) được dùng:

$$\int_{y_0}^{(n+1)h} f(x)dx = h \frac{y_0 + y_1}{2}$$

với h = chu kỳ lấy mẫu, y_0 và y_1 là biên độ hai điểm lấy mẫu liên tiếp. Như vậy trị hiệu dụng là:

$$VRMS = \sqrt{\frac{1}{T} \int_0^T V^2 dt} = \sqrt{\frac{1}{T} \sum_{i=1}^{N-1} \frac{y_i^2 + y_{i+1}^2}{2} h}$$

Trong đó: T: chu kỳ tín hiệu vào

y_i, y_{i+1} : các biên độ của 2 mẫu liên tiếp.

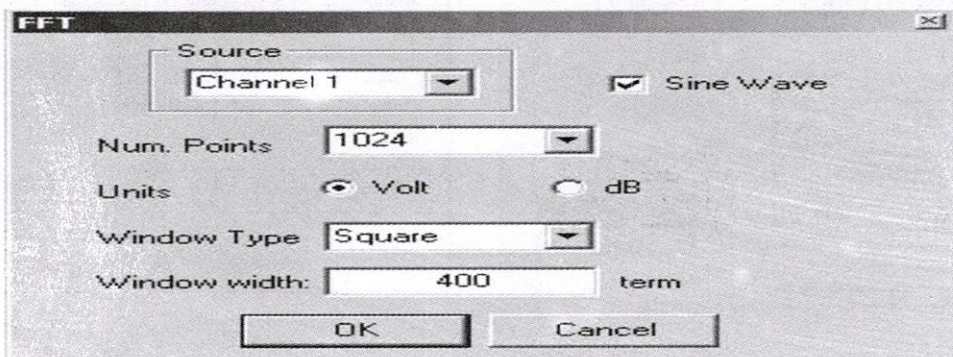
N: số điểm lấy mẫu trong 1 chu kỳ tín hiệu vào.

4.3 Trị đỉnh – đỉnh

Tìm trị của đỉnh dương và trị của đỉnh (đáy) âm trong mảng dữ liệu, sau đó suy ra trị đỉnh – đỉnh.

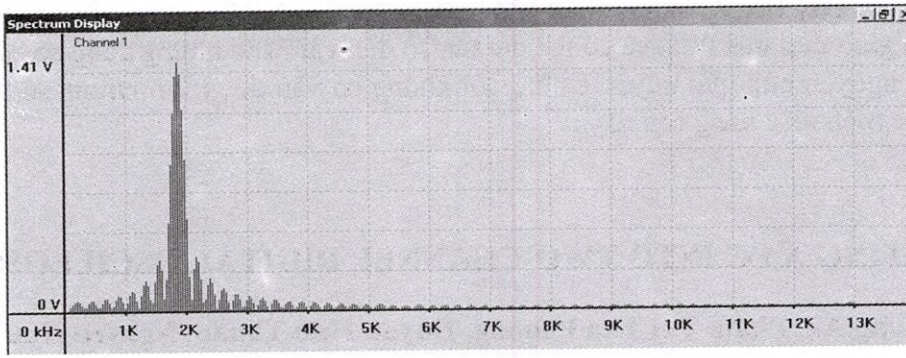
5. PHÂN TÍCH TẦN PHỔ TÍN HIỆU DÙNG FFT

FFT (Fast Fourier Transform) là thuật toán tính toán nhanh biến đổi Fourier rời rạc (Discrete Fourier Transform – DFT) trong việc phân tích tần phổ tín hiệu. Ta lấy một số lượng mẫu đúng bằng bội số nguyên của một chu kỳ tín hiệu, phần còn lại xem các mẫu như bằng không (đây là cách độn không):



5.1 Chương trình biến đổi FFT thuận và nghịch

Vì toàn bộ phần mềm cho phòng thí nghiệm dùng mạng máy tính được viết bằng ngôn ngữ Visual C++ (ngoại trừ phần cơ sở dữ liệu có kết hợp với Access), không dùng MATLAB nên chúng tôi tự viết chương trình biến đổi FFT đã biết. Hình 7 là một ví dụ phân tích.



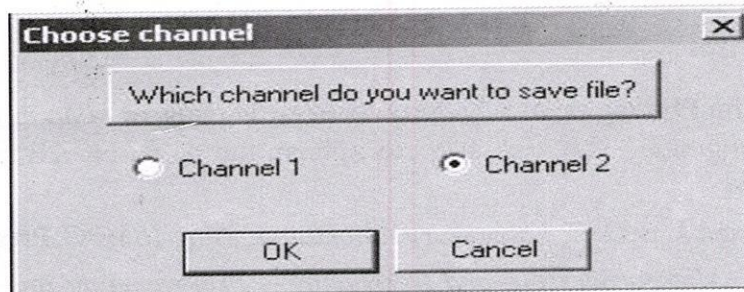
Hình 7. Phân tích phổ tín hiệu sin (méo và có nhiễu)

5.2 Độ méo hài

Phân tích FFT đã cho ra các thành phần hài của tín hiệu nên từ kết quả FFT ta có thể tính được độ méo hài toàn phần (Total Harmonic Distortion – THD). Vì FFT cho kết quả có đối xứng chẵn (đối xứng gương) nên khi tìm các hài chỉ cần dò tới $N/2$ là được.

6. LƯU TRỮ DỮ LIỆU

Chức năng cất dữ liệu cho phép người sử dụng lưu lại kết quả vào file, tiện cho việc lưu trữ và so sánh tín hiệu sau này. Ngược lại với chức năng cất dữ liệu là chức năng lấy dữ liệu để đưa file dữ liệu vào chương trình



In dữ liệu: Sau khi xem dữ liệu, ta có thể in ra giấy hình vẽ cùng với các thông số của tín hiệu đó.

7. KẾT LUẬN

Chúng tôi đã thiết kế và thực hiện độc lập một card thu nhận dữ liệu hai kênh với đặc điểm là có mạch tự động phân tâm (autoranging) và sử dụng vi điều khiển; hai yếu tố này làm tăng tính tự động và dễ sử dụng của Oscilloscope. Chúng tôi đã viết phần mềm gồm nhiều module cho hầu như tất cả các chức năng của một Oscilloscope số hai kênh: hiển thị (hai dạng sóng dịch chuyển và khuếch đại độc lập), đo (trị đỉnh – đỉnh, trị hiệu dụng, chu kỳ, tần số), phân tích (độ méo hài, tần phổ FFT). Chúng tôi đã thử nghiệm trên nhiều dạng sóng (sin, vuông, tam giác . . .) và so sánh với oscilloscope tương tự và oscilloscope số, độ chính xác về các phép đo tần số, biên độ, và các đại lượng khác đều tốt (tương đương) nếu tín hiệu vào có tần số dưới 10KHz tương ứng 1 chu kỳ tín hiệu lấy được trên 10 điểm mẫu. Với tín hiệu có tần số thấp, từ DC tới vài chục Hz, thường các oscilloscope tương tự không thể quan sát được, nhưng chúng tôi cũng viết các chương trình cho phép thu nhận tín hiệu dạng này và đảm bảo độ chính xác tương đương các oscilloscope số.

Phần mềm xem như đầy đủ với các giao diện thân thiện như Oscilloscope số thật sự, và cũng tương thích khi phần cứng thay đổi. Với linh kiện sử dụng hiện tại (khuếch đại thuật toán, mạch chuyển đổi ADC . . .) và với giao tiếp bus ISA, tần số hoạt động bị giới hạn ở dải tần số

từ DC đến 50KHz. Đây là giới hạn rất hiển nhiên của công trình. Việc nâng cấp (dùng linh kiện cao cấp hơn và giao tiếp bus PCI) sẽ có thể đạt tần số đến vài Mhz nhưng cũng sẽ có thêm một vài vấn đề cần nguyên cứu giải quyết. Chúng tôi không có vấn đề gì lớn nhưng sẽ tốn kém rất nhiều đang thực hiện việc nâng cấp này.

CONVERTING A PC INTO TWO-CHANNEL DIGITAL OSCILLOSCOPE

Nguyen Truong An, Pham Thi Thu Phuong, Huynh Huu Thuan, Nguyen Huu Phuong
University of Natural Sciences, VNU-HCM

ABSTRACT: *Oscilloscope is a very important and popular measuring instrument for electronics and many other fields. It displays the waveform and various characteristics of signals. There are two kinds of oscilloscope: analog and digital. Digital oscilloscope is superior to analog one but is not as popular because of its higher cost. This paper presents the conversion of a PC into a two-channel digital oscilloscope, consisting of hardware which is a self-designed data acquisition card, and the software which is a set of various programs written in Visual C++.*

TÀI LIỆU THAM KHẢO

- [1]. Nguyễn Hữu Phương et al, *Xây dựng cơ sở phần cứng và phần mềm phòng thí nghiệm điện tử dùng mạng máy tính*, báo cáo nghiệm thu đề tài NCKH trọng điểm ĐHQG – HCM, 2004
- [2]. S.C. Gate and J. Becker, *Laboratory Automation using IBMPC*, Prentice Hall, 1999
- [3]. D.V. Hall, *Microprocessors and Interfacing – Programming and Hardware*, 2nd Ed., McGraw-Hill, 1992
- [4]. A.D. Helfrick and W.D. Cooper, *Modern Electronic Instrumentation and Measurement Techniques*, Prentice Hall, 1994