

Bài thực hành số 2

LẬP TRÌNH GHEP NOI PC VA NGOAI VI VỚI 8051 (Online)

1. Mục đích

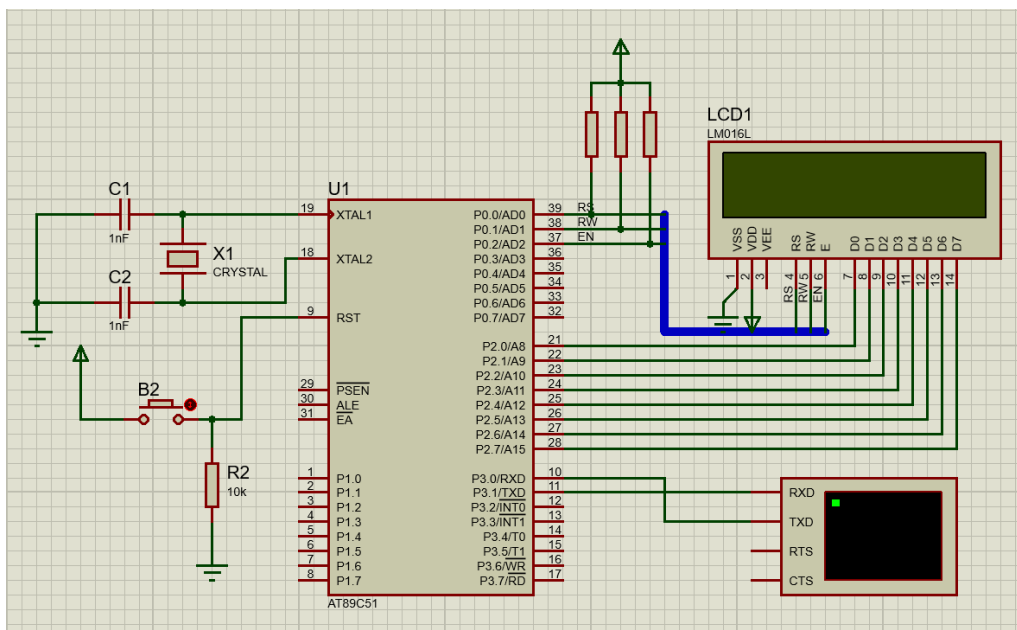
- Lập trình với timer trong 8051 để phát tín hiệu với tần số xác định.
- Ghep noi LCD ký tự với 8051, lập trình hiển thị dữ liệu trên LCD.
- Ghep noi hệ 8051 với PC qua cổng UART, lập trình truyền/nhận dữ liệu giữa PC và 8051.
- Xây dựng được giao thức ghep noi và trao đổi dữ liệu giữa PC với hệ 8051. Áp dụng xây dựng ứng dụng demo.

2. Chuẩn bị

- Tài liệu thực hành + mạch và code mẫu.
- Cài đặt các phần mềm: Công cụ **Keil C51**, **Proteus 8.9** (hoặc mới hơn), **Free Virtual Serial Port** (<https://freevirtualserialports.com>), **Hercules** (<https://www.hw-group.com/software/hercules-setup-utility>).
- Môi trường lập trình ứng dụng trên PC: **python**, **Java**, hoặc **C#** (Visual Studio).

3. Tiến hành

3.1. Lập trình UART cho 8051 trên Proteus

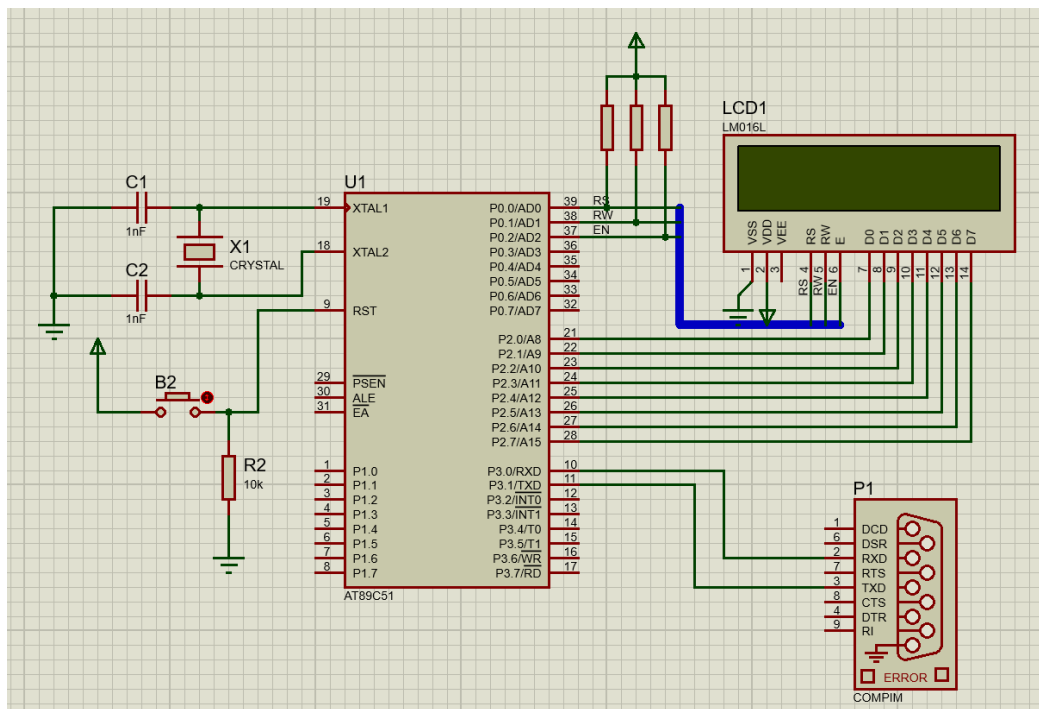


- Dùng Proteus thiết kế mạch như hình vẽ trên, đặt tần số clock cho 8051 là **11.0592 MHz**. Dùng chức năng **Place Virtual Instrument** đặt một **Virtual Terminal** vào mạch.

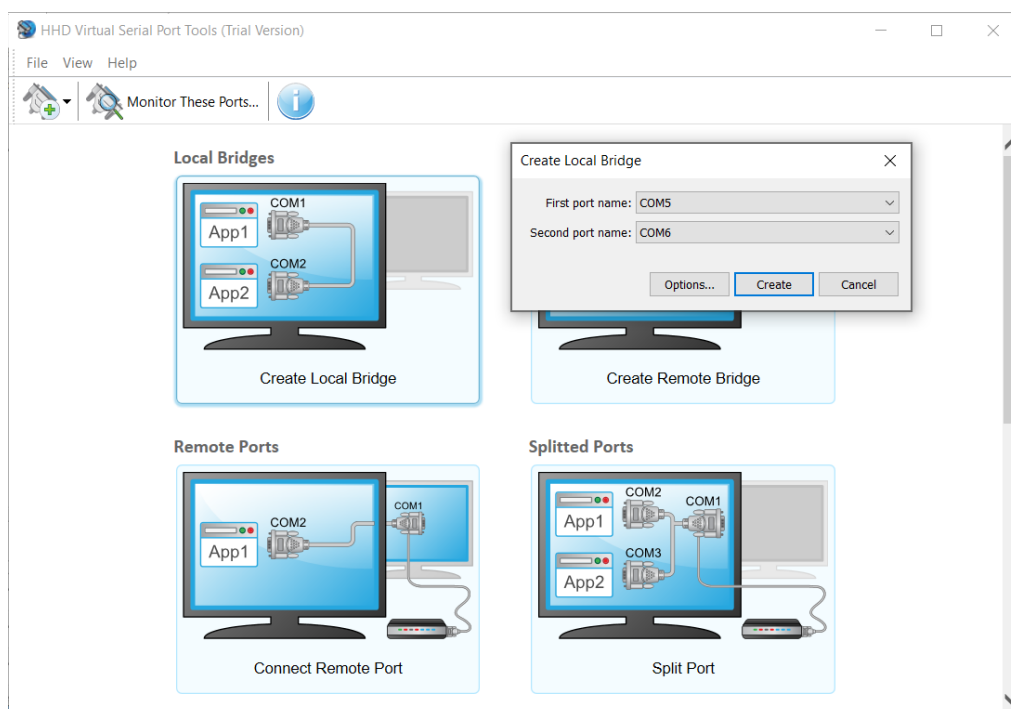
- Tham khảo code mẫu trong file **com.c** lập trình cho 8051 thực hiện chức năng *loopback*: các ký tự do người dùng gõ vào Virtual terminal sẽ được gửi đến 8051, 8051 gửi lại đúng ký tự đó tới Virtual Terminal.

3.2. Ghép nối 8051 với máy tính thông qua cổng serial

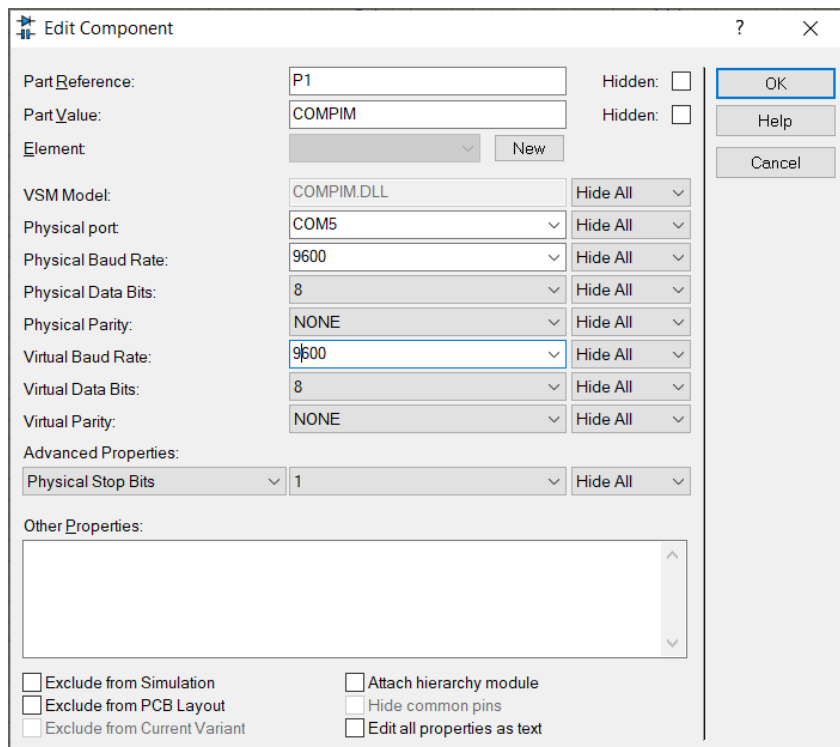
- Sử dụng thiết kế mạch trong câu trên, thay Virtual Terminal bằng một thiết bị COMPIM (xem hình dưới).



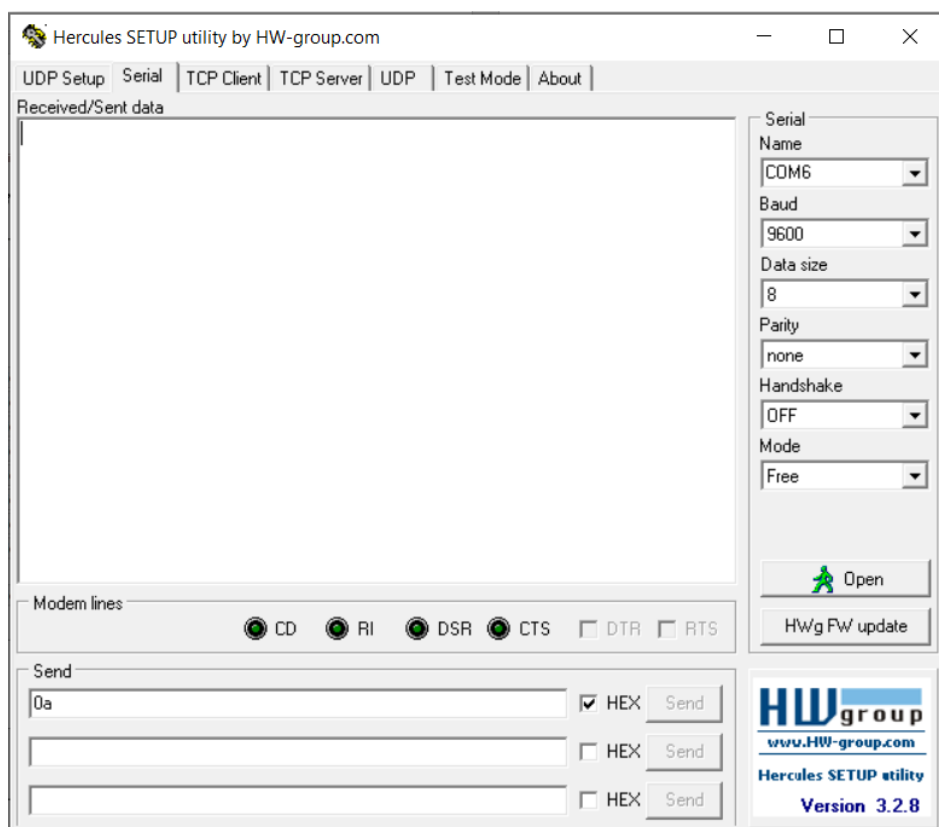
- Sử dụng Free Virtual Serial Port, chức năng **Create Local Bridge** tạo một cặp cổng COM ảo, đặt tên là COM5 và COM6.



- Cấu hình cho COMPIM trong Proteus nối với COM5 ở baudrate 9600 bps.



- Mở phần mềm Hercules, mở cổng COM6 trong phần Serial với baudrate 9600.



- Kiểm tra kết nối giữa Hercules với 8051 xem có hoạt động giống với Virtual Terminal ở câu 3.1 hay không? Nếu không thì kiểm tra lại các bước kết nối cho chính xác.

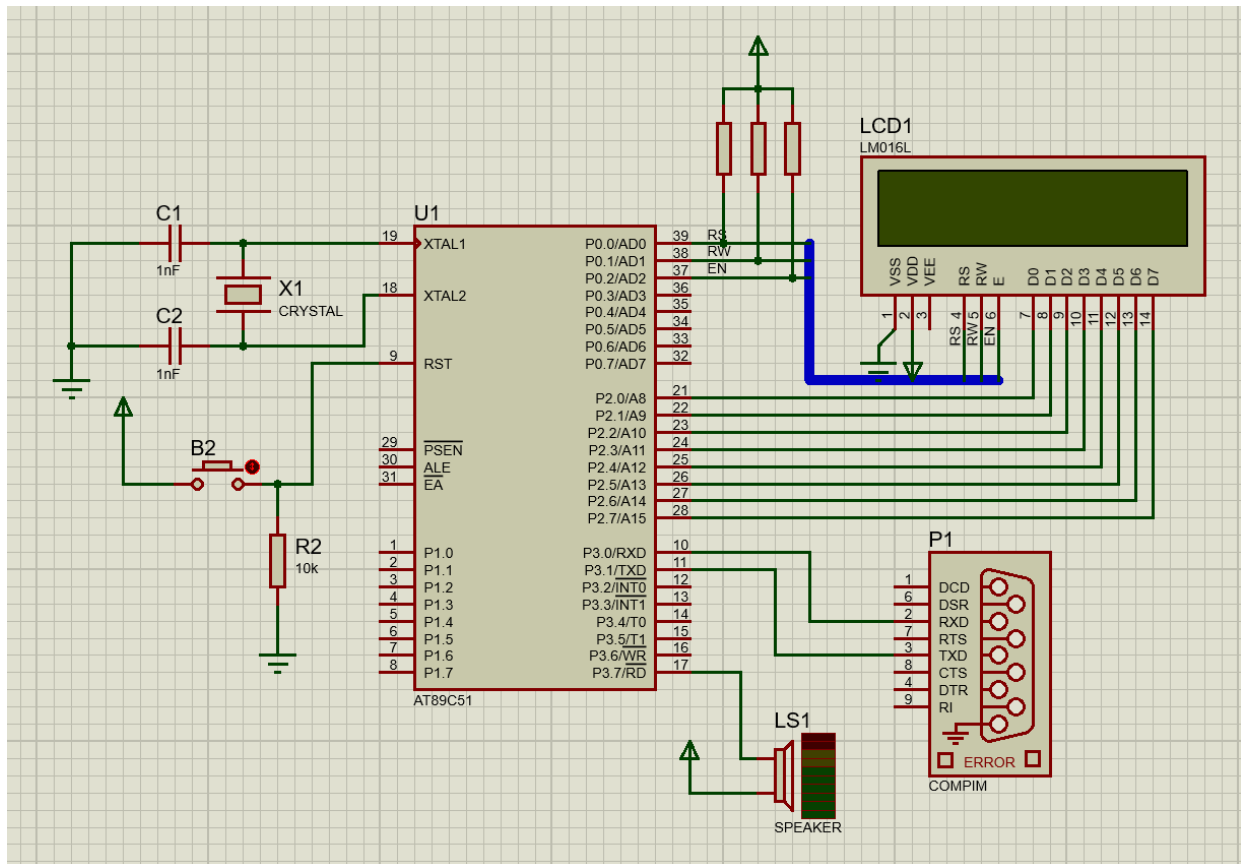
3.3. Lập trình ghép nối 8051 với LCD

- Tiếp tục sử dụng mạch ở câu 3.2, tham khảo code mẫu trong file **lcd.c**.
- Lập trình điều khiển hiển thị chuỗi ký tự sau trên 2 hàng LCD:

“Hello
World...”

3.4. Phát tín hiệu với tần số xác định

- Sử dụng mạch câu 3.2, thêm một thiết bị **SPEAKER (Loudspeaker model (Analog) – Outputs via Sound Card)** vào mạch và kết nối như hình dưới.



- Lập trình sử dụng timer 0 để phát tín hiệu có tần số 261.63 Hz ra loa.
- Cho biết tần số thực sự được phát ra loa là bao nhiêu? Tại sao lại không thể tạo tần số chính xác?
- Lập trình để phát ra loa các nốt nhạc Do, Re, Mi, Fa, Sol, La, Si... tương đương từ C4 đến F5. Mỗi nốt nhạc kéo dài trong khoảng 1 giây (không cần chính xác tuyệt đối).

Chú ý: Tần số các nốt nhạc được tham khảo trong bảng sau:

<https://pages.mtu.edu/~suits/notefreqs.html>

3.5. Bài tập tự làm: Ghép nối hệ 8051 và PC xây dựng ứng dụng Electric piano

Xây dựng ứng dụng gồm các tính năng.

- Chơi một đoạn nhạc ngắn trên 8051.
- Sử dụng bàn phím máy tính để chơi piano, tại một thời điểm chỉ nhấn một phím.
- Mỗi phím trên bàn phím ứng với một phím trên đàn piano.
- Khi bấm 1 phím trên bàn phím thì mã phím được gửi qua cổng COM đến mạch 8051. Mạch sẽ tạo âm thanh tương ứng với phím cho tới khi phím được nhả ra.
- Tên của nốt nhạc được hiển thị trên màn hình LCD.

Ứng dụng gồm 2 thành phần:

- **Firmware trên 8051:** chơi 1 đoạn nhạc ngắn khi mạch khởi tạo, sau đó liên tục nhận lệnh từ PC qua cổng COM, chơi nốt nhạc + hiện tên nốt nhạc ra màn hình LCD.
- **Ứng dụng trên PC:** nhận sự kiện bấm trên bàn phím, gửi lệnh chơi nốt nhạc/tắt nốt nhạc xuống 8051 qua cổng COM. Gợi ý: có thể sử dụng pyserial ([pyserial · PyPI](#)), hoặc Serial (C# [SerialPort Class \(System.IO.Ports\) | Microsoft Docs](#))

Viết báo cáo theo nhóm gồm các nội dung:

1. Phân công nhiệm vụ.
2. Nộp chương trình trên PC, chương trình trên 8051.
3. Giải thích nguyên tắc phát ra các âm thanh Do, Re, Mi,...
4. Giải thích cấu trúc chương trình, giải thích mã nguồn chương trình trên 8051.
5. Giải thích chương trình trên PC.

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 blinking
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
80	Force cursor to beginning to 1st line
C0	Force cursor to beginning to 2nd line
38	2 lines and 5x7 matrix