TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

**Báo cáo**

Bài thực hành 02 môn hệ nhúng

Nhóm sinh viên thực hiện : Nhóm HĐ

|  |  |
| --- | --- |
| Họ và tên | MSSV |
| Nguyễn Quang Hưng | 20183760 |
| Đỗ Trọng Đức | 20183710 |

Lớp : Kỹ thuật máy tính– Khoá 63

Giảng viên hướng dẫn : Thầy Đỗ Công Thuần

***Hà Nội, tháng 5 năm 2021***

# MỤC LỤC

[MỤC LỤC 2](#_Toc73304753)

[PHÂN CÔNG THÀNH VIÊN TRONG NHÓM 3](#_Toc73304754)

[Câu hỏi 1: 4](#_Toc73304755)

[Câu hỏi 2: 4](#_Toc73304756)

[Câu hỏi 3: 4](#_Toc73304757)

[1. Chương trình trên PC và chương tình trên 8051 4](#_Toc73304758)

[2. Giải thích nguyên tắc phát ra các âm thanh Do, Re, Mi,… 4](#_Toc73304759)

[3. Giải thích cấu trúc và mã nguồn chương trình trên 8051. 5](#_Toc73304760)

[4. Giải thích chương trình trên PC. 11](#_Toc73304761)

# PHÂN CÔNG THÀNH VIÊN TRONG NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ và tên** | **MSSV** | **Công việc thực hiện** | **Đánh giá** |
| Nguyễn Quang Hưng | 20183760 | * Code các module đọc dữ liệu từ cổng serial, phát nhạc ra loa, ghi chữ ra LCD, code chương trình trên 8051 * Tính toán các giá trị thanh ghi tạo trễ timer0 tương ứng với các nốt nhạc * Viết báo cáo | Hoàn thành tốt |
| Đỗ Trọng Đức | 20183710 | * Viết mã nguồn cho chương trình PlayMusic chạy trên PC sử dụng ngôn ngữ C#. Chương trình đọc 1 phím ấn và gửi phím đó đến cổng COM5 gửi sang 8051 * Viết báo cáo miêu tả chương trình chạy trên PC | Hoàn thành tốt |

# Câu hỏi 1:

Trong bài 3.4, với tần số clock của 8051 là 11.0592 MHz thì giá trị các thanh ghi TH0, TL0 cần đặt là bao nhiêu để phát ra tín hiệu tần số 261.63 Hz ra loa?

Trả lời:

Với tần số clock của 8051 là 11.0592 MHz, 8051 có chu kỳ là T = 1/f \* 12= 1/11.0592\*12 = 1.085 (micro giây).

Với tín hiệu có tần số 261.63 Hz, ta có chu kỳ T = 1/f = 1/261.63 = 3822 (micro giây) là chu kỳ của sóng vuông. Trong 1 chu kỳ chia làm 2 nửa chu kỳ.

Vậy số nhịp cần cho timer nhảy để tạo chu kỳ đó là 3822 / 2 / 1.085 = 1761

Giá trị cần nạp vào các thanh ghi TH0, TL0 là 65536 - 1761 = 63875 đổi sang mã Hex là F91E. Nạp 0xf9 cho TH0 và 0x1E cho TL0.

# Câu hỏi 2:

Với các giá trị TH0, TL0 ở trên, cho biết tần số thực sự được phát ra loa là bao nhiêu? Tại sao lại không thể tạo tần số chính xác?

Trả lời

Với giá trị TH0 = 0xF9 và TH0 là 0x1E, ta có giá trị của thanh ghi là 0xF91E, đổ sang thập phân là 63875. => đã tạo trễ 1761 timer.

Ta có nửa chu kỳ của nốt nhạc là 1761\*1.085 = 1911 (micro giây)

=>Chu kỳ của nốt nhạc là 1911\*2 = 3822 (micro giây)

=>Tần số là 1/f = 1/3822 \* 1e6 = 261.64 Hz

Không thể tạo tần số chính xác, vì chu kỳ timer cỡ micro giây, không thể tính chính xác đến mức nhỏ hơn chu kỳ timer nên không thể tạo trễ chính xác. Mặt khác, em đã làm tròn nhiều chỗ như tính chu kỳ clock của 8051, chu kỳ của nốt nhạc, nên kết quả cuối cùng không chính xác nữa.

# Câu hỏi 3:

## Chương trình trên PC và chương tình trên 8051

Nhóm em đã gửi kèm file chương trình PC, file code chương trình và file code cho 8051 trong file nén cũng với báo cáo.

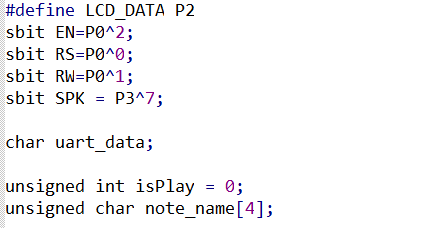
## Giải thích nguyên tắc phát ra các âm thanh Do, Re, Mi,…

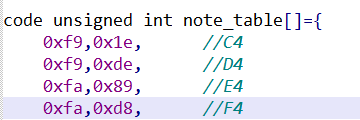
Các âm thanh Đồ, Rê, Mi, … sẽ mang 1 tần số âm nhất định. Ví dụ 1 âm Đồ C5 có tần số âm là 523.25 Hz. Vậy ta chỉ cần phát ra loa sóng có tần số là 523.25 Hz thì nó sẽ là âm Đồ C5.

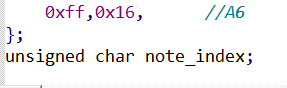
Vậy với âm Đồ C5 tần số 523.25 Hz và tình tương tự câu 1, ta có giá trị thanh ghi sẽ là 0xfc8f. Vậy ta truyền cho thanh ghi TH0 là 0xfc và thanh ghi TL0 là 0x8f tạo trễ và loa sẽ phát ra âm có tần số gần bằng tần số âm của nốt nhạc (sai số đã giải thích ở câu hỏi 2).

## Giải thích cấu trúc và mã nguồn chương trình trên 8051.

Ta có khai báo 1 số biến toàn cục và cấu trúc cho chương trình như sau:



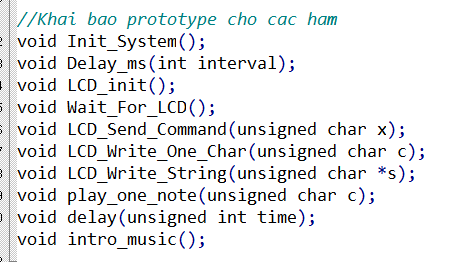




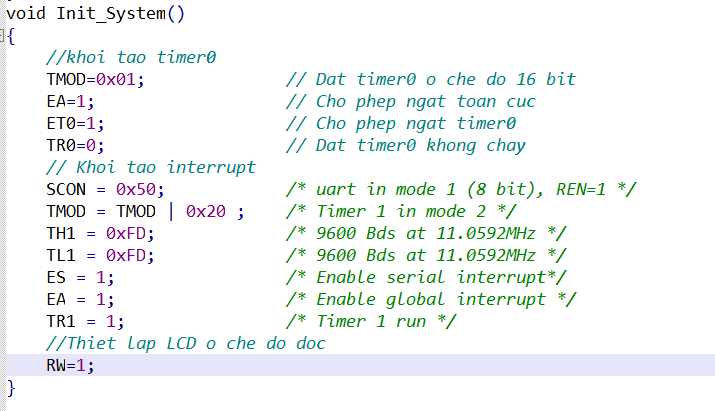
Giải thích các biến:

* Đặt biến LCD-Data là cổng P2 để điều khiển LCD (Các chân từ D0 đến D7 đã nối với các chân từ P2.0 đến P2.7 của cổng P2)
* Khai báo các biến điều khiển các chân: biến EN kiểu bit là chân P0.2, biến RS là chân P0.0, RW là chân P0.1, biến SPK là P3.7. biến SPK sẽ có chức năng điều khiển loa, phát tín hiệu ra loa
* Biến uart-char có chức năng lưu trữ 1 ký tự gửi qua cổng UART và gửi lại ký tự đó trở lại cổng UART.
* Biến isPlay có chức năng điều khiển hoạt động chơi nhạc hay không. Nếu có 1 ký tự gửi vào cổng UART, cờ isPlay được bật và sẽ chơi nốt nhạc tương ứng. Ngược lại, cờ isPlay sẽ là 0;
* Biến note-name là 1 xâu ký tự chứa tên của nốt nhạc và hiển thị lên màn hình LCD
* Mảng note-table gồm 42 phần tử đi theo cặp, là các giá trị của 2 thanh ghi TH0 và TL0 (em đã tính toán các giá trị này từ trước), tương ứng với 21 nốt nhạc đồ, rê, mi,… từ E4 – B4, E5 – B5, E6 – B6
* Biến note-index là chỉ số của nốt tương ứng với phím bấm. Nhận các giá trị từ 0 đến 20 tương ứng với 21 phím bấm.

Tiếp theo, em sẽ giải thích các prototype và các hàm con



Đầu tiên, ta có hàm Init-System có nhiệm vụ khởi tạo các giá trị, set các bit là các chân ở chế độ đọc/ghi dữ liệu, set ngắt 4, time0.



TMOD=0x01: đặt timer0 ở chế độ 16 bit tràn.

EA=1: cho phép ngắt toàn cục

ET0=1: cho phép ngắt trên timer0

TR0=0: khởi tạo, đặt timer0 chưa chạy

SCON = 0x50 : đặt cổng uart đọc ghi 8 bit

TMOD = TMOD | 0x20 : đặt timer1 là chết độ 2 8 bit auto reload

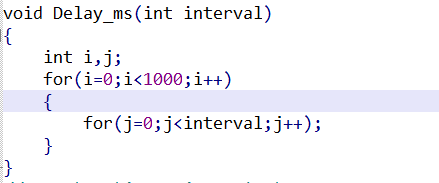
TH1 = 0xFD, TL1 = 0xFD : đặt baudrate là 9600

TR1=1: timer1 bắt đầu chạy

RW=1: thiết lập màn hình LCD ở chế độ đọc

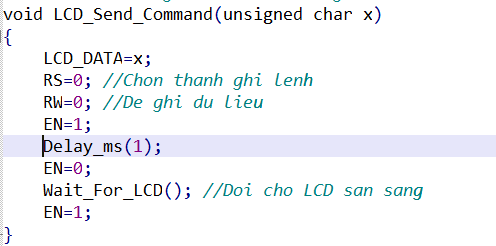
Ở đây, timer0 sẽ xử lý tạo trễ để tạo ra tần số phù hợp với từng nốt nhạc, còn timer1 sẽ xử lý ngắt khi có dữ liệu vào ở cổng UART.

Tiếp theo, hàm Delay-ms(int interval) tạo delay số interval ms. Hàm náy chỉ tạo trễ tương đối dựa theo xung clock thạch anh.



Trong hàm này, vòng lạp for chạy khoảng interval\*1000 lần. Với xung nhịp của 80051 là 11.0592, mỗi 1 chu kỳ lệnh sẽ xấp xỉ 1.085 micro giây. Vậy interval\*1000 lệnh sẽ xấp xỉ interval – ms.

Tiếp theo, hàm LCD-Send-Command gửi 1 lệnh đến cổng P2 cho LCD.

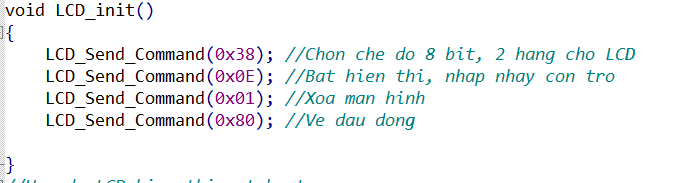


Tham số đầu vào là x là giá trị gửi đến cổng P2, bằng câu lệnh LCD-DATA = x.

RS = 0: LCD sẽ nhận tham số đề vào là lệnh và ghi vào thanh ghi lệnh

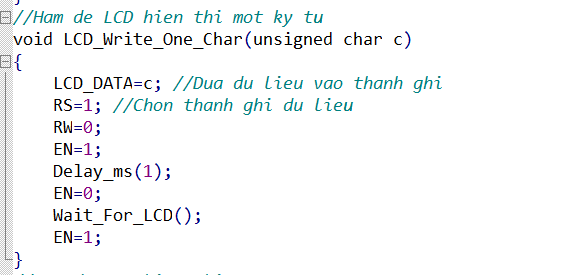
RW=0: chọn chế độ LCD ghi dữ liệu

Hàm LCD-init có chức năng thiết lập các khởi tạo cho LCD, đặt chể độ 8 bit, 2 hàng chữ, bật hiện thị con trỏ, xóa màn hình, về đầu dòng.



Hàm này dùng hàm LCD-Send-Command để gửi các tham số đến cho LCD. Các giải thích cho code đã có như trên hình.

Tiếp theo, hàm LCD-Write-One-Char có chức năng hiển thị 1 ký tự kiểu char ra màn hình lcd.

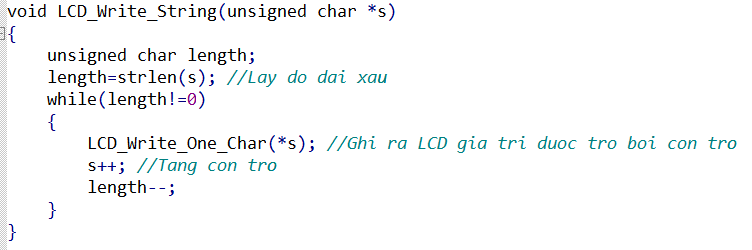


LCD-Data = c : đưa ký tự đầu vào ra thanh ghi

RS=1 : chọn thanh ghi là dữ liệu

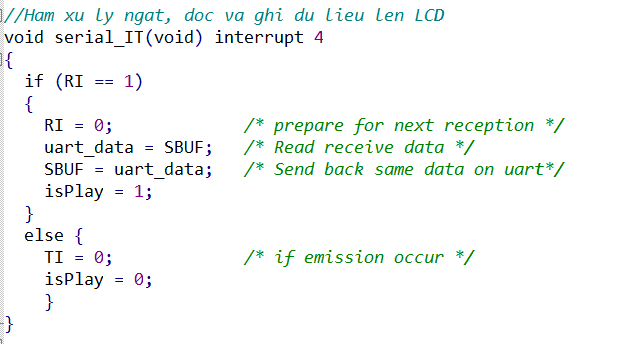
RW=0: chọn chết độ ghi cho LCD

Hàm LCD-Write-String (unsigned char \*s) ghi dữ liệu đầu vào là 1 chuỗi ký tự ra LCD.



Hàm này sẽ sử dụng hàm LCD-Write-One-Char để ghi lần lượt các ký tự của chuỗi ra màn hình LCD.

Tiếp theo, ta có hàm xử lý ngắt ở cổng seial port.

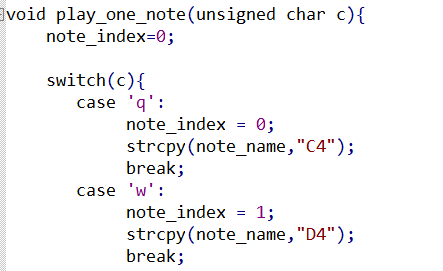


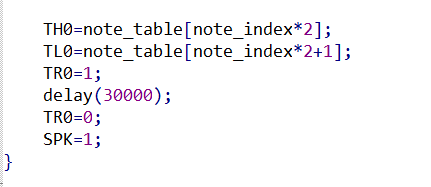
Nếu RI=1 cờ receive bật tức là nhận được 1 byte ở cổng serial, đặt lại RI=0 cho lần ngắt kế tiếp, đọc ký tự nhận được từ thanh ghi đệm SBUF, và ghi luôn ký tự đó ra SBUF ghi ra cổng serial. Sau đó, đặt cờ isPlay = 1 để trong hàm main() sẽ phát nhạc.

Nếu RI=0, không có ngắt hoặc đã truyền byte đi thì đặt cờ transmit TI=0 và cờ isPlay = 0;

Sau đó, ta có hàm play-one-note(unsigned char c) để phát 1 nốt nhạc như sau:

Vì hàm này dài nên em sẽ chụp đoạn đầu và đoạn cuối.





Tham số đầu vào là ký tự đọc được từ cổng uart. Em sử dụng cấu trúc switch-case để ánh xạ từ ký tự sang nốt tương ứng. Mỗi 1 note dẽ đặc trưng bởi chỉ số note-index và tên note là note-name. em quy định bàn phím nhạc của em gồm 21 phím từ C4 đến B6 với 3 nhóm phím như sau:

* Các phím từ q-w-e-r-t-y-u ứng với 7 phím nhạc từ C4 đến B4
* Các phím từ a-s-d-f-g-h-j ứng với 7 phím nhạc từ C5 đến B5
* Các phím từ z-x-c-v-b-n-m ứng với 7 phím nhạc từ C6 đến B6
* Các phím còn lại sẽ tự động phát nôt nhạc C4

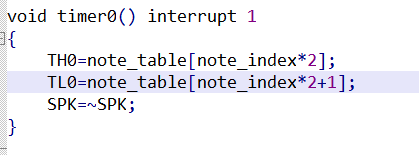
Đặt 2 thanh ghi TH0 và TL0 lấy từ mảng note-table với phần tử ứng với nốt nhạc cần chơi, quản lý bằng chỉ số note-index.

TR=1 : timer0 bắt đầu chạy

Delay(30000) : delay 30ms đễ chơi nốt nhạc đó

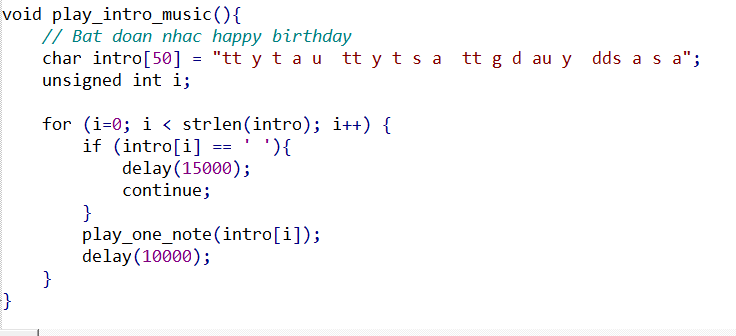
SPK = 1: tắt loa, đặt lại loa cho các lần chơi nhạc tiếp theo

Hàm xử lý ngắt timer0 như sau:

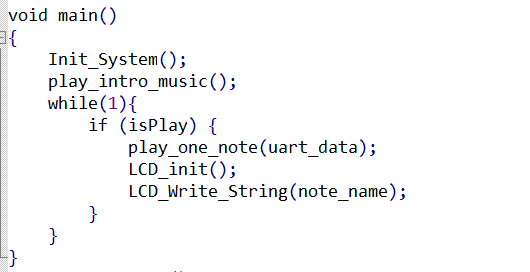
.

Sau khi có ngắt tại timer0, đặt lại các giá trị cho 2 thanh ghi TH0 và TL0 và SPk = ~SPK = 0, phát tín hiệu 0 ra loa, tức là bật loa vì loa đang neo bởi nguồn.

Tiếp theo, hàm play-intro-music sẽ chơi 1 đoạn nhạc mở đầu là bài hát happy birthday khi người dùng bật ứng dụng, cũng có chức năng test loa và hệ thống. Bài hát happy birthday đã được ánh xạ thành 1 đoạn ký tự lưu ở biến intro



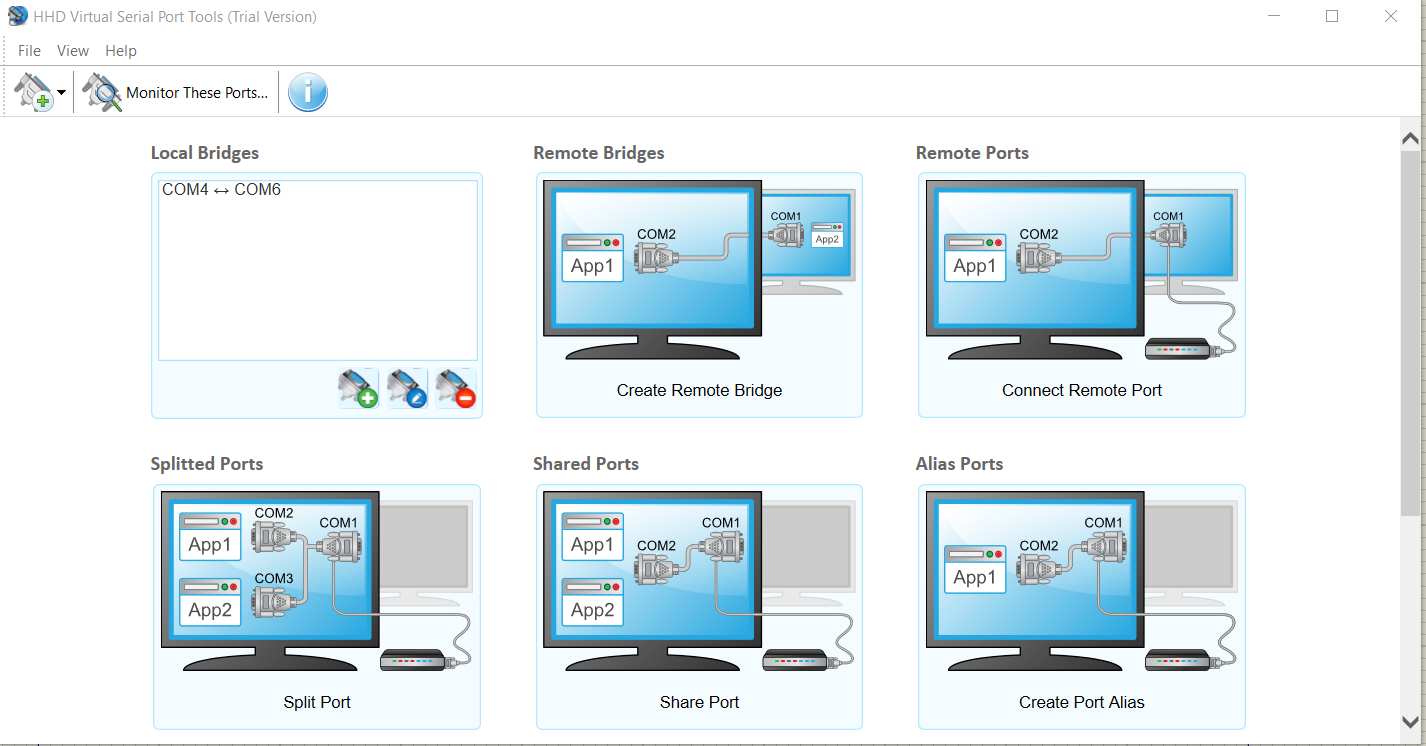
Cuối cùng, em xin được giải thích hàm main() :



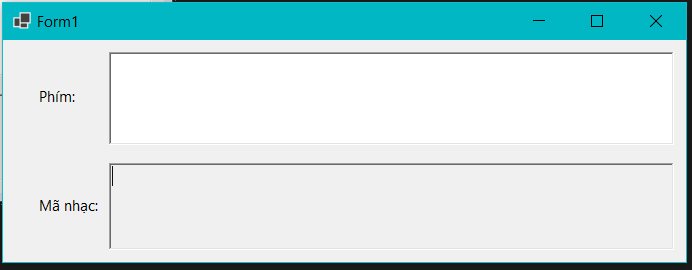
Bước đầu, gọi hàm Init-System để khởi tạo và thiết lập các giá trị, các thanh ghi, timer. Gọi hàm play-intro-music để chơi đoạn nhạc mở đầu.

Tiếp theo, trong vòng lặp while, nếu có ngắt khi nhận được ký tự ở cổng serial, cờ isPlay sẽ được bật., chương trình sẽ gọi hàm play-one-note chơi nốt nhạc đó. Gọi hàm LCD-init để xóa màn hình cũ, gọi hàm LCD-write-string để hiển thị tên của nốt tương ứng lên màn hình LCD. ở đây em sẽ hiện thị mã của nốt nhạc, từ là các giá trị C4, D4, E4, …, A6, B6.

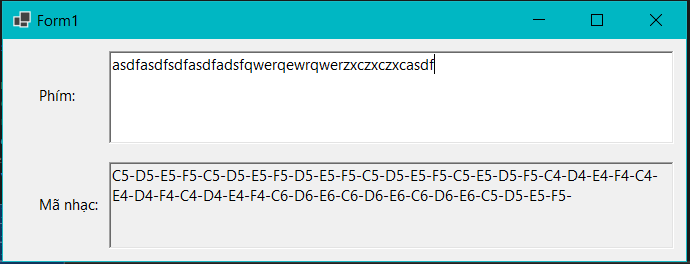
## Giải thích chương trình trên PC.



* Tạo cổng ảo COM4 và COM6, cổng COM4 nối trực tiếp với COMPIM trên Proteus. Cổng COM6 sẽ được nối với phía bên Application PC.
* Giao diện của chương trình phía PC.



* Khi gõ 1 phím, kết quả đầu ra là phần textbox phía trên hiển thị phím vừa gõ, phần Mã nhạc hiển thị nốt nhạc tương ứng.
* Kết quả thử nghiệm:

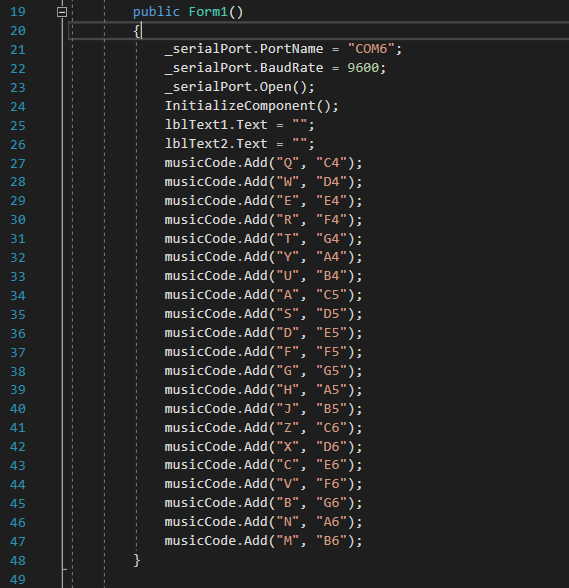


Class Form1:

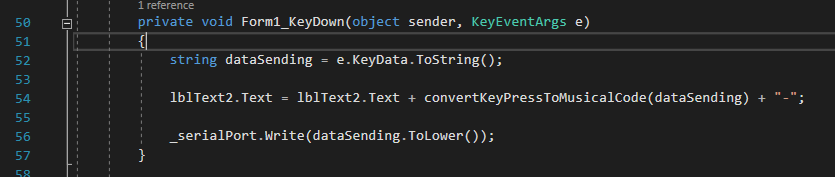
* 1. Các thuộc tính:
* \_serialPort: SerialPort: cổng serial.
* musicCode: Dictionary<string, string>: hàm ánh xạ từng phím trên bàn phím PC sang code của nốt nhạc.

1. Các phương thức:

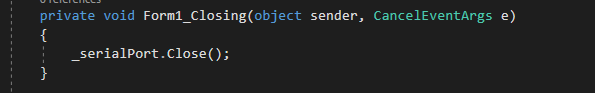
* Constructor:



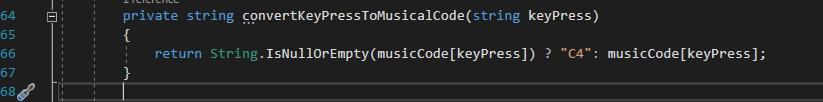
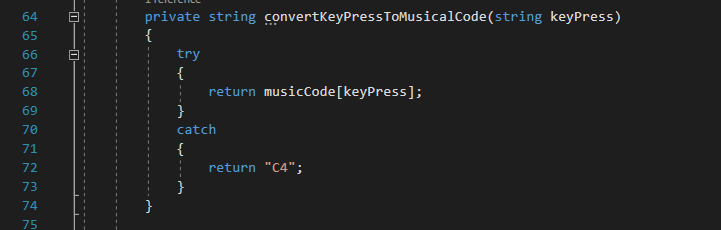
* + - Khởi tạo các giá trị của serialPort và musicCode.
    - Chọn cổng sử dụng phía PC là COM6, BaudRate là 9600 (mặc định).
    - Mở cổng bằng phương thức Open().
    - Gán các cặp giá trị ánh xạ từng string sang string như trên.
* Form1\_KeyDown(object, KeyEventArgs):



* + - Bắt sự kiện KeyDown từ bàn phím.
    - Đầu vào của hàm là sender và KeyEventArgs e.
    - e là tham số đặc trưng cho phím vừa gõ vào => lấy dữ liệu phím vừa KeyDown bằng cách sử dụng e.KeyData.ToString().
    - In ra lblText2 (phần textbox thứ 2 ở trên Form1) thêm các nốt dưới dạng Nốt nhạc.
    - Write dataSending dưới dạng chữ viết thường => send vào COM6 một string chứa 1 ký tự.
* Form1\_Closing():



* + - Thực hiện công việc đóng cổng lại khi tắt form1.
* convertKeyPressToMusicalCode(string): Ánh xạ từ phím đầu vào sang nốt nhạc:



* + - Nếu xảy ra ngoại lệ, trực tiếp gửi tới nốt “C4” và in “C4” ra màn hình.