



LNF STAGE REPORT

By

Davide Bottoni & Giorgio Castellani

Stage aim

The main purpose of the stage was to introduce us to the following topics:

- C++ language and object-oriented programming.
- Linux operating system.
- Digital electronics and computers architecture.

As conclusion of the stage and synthesis of the acquired knowledge, we have designed and realized the hardware and the software of a device for interfacing a PC parallel port with a 7-segment display and a buzzer.

Materials and instruments employed

We have used a PC equipped with Red Hat Linux 7.2 operating system and the GNU C++ compiler for software development. The design and realization of the interface circuit has required the use of some electronic components and of their datasheets, a breadboard, a soldering iron, a power supply and a flat cable for connecting our circuit to the parallel port of the computer.

What we have learned

- LINUX
 - How to install the Linux Red Hat 7.2 operating system.
 - Users' settings.
 - Basic commands of the "shell" . (We used the "Tcshell").
- C++
 - The concept of object-oriented programming.
 - Classes and objects.
 - Basic functions of the C++ language.
 - The typical structure of a C++ program.
 - How to compile a C++ program.
- COMPUTER ARCHITECTURE
 - Differences between analog and digital electronics: the sampling.
 - Number systems (binary, octal, hexadecimal).

- Logical operators (AND, OR, XOR, NOR, NAND ...) and truth tables.
- Electronic implementation of the logic operators: the “gates”.
- Design example: a four-bit comparator.
- Combinatorial and sequential networks.
- Flip-flops.
- Typical digital functions and their implementation in integrated circuits: registers, counters, comparators, shift registers, memories.
- What is a microprocessor:
 - the internal structure of a CPU.
 - the concept of “bus”.
 - addressing modes.
 - memory and I/O handling.
 - interrupts.
 - Machine language and assembler language.

Results: design and realization of the interface circuit

- THE INTERFACE HARDWARE

- How to read a datasheet and made a simple project.
- Practical realization of an electronic circuit:
 - Arrangement of the components on the breadboard.
 - Welding and wiring the components on the breadboard.

- THE INTERFACE SOFTWARE

The software integrates the low-level functions for accessing the PC parallel port with a main structure that allows to drive our interface circuit using the PC keyboard. The various functions of the software are invoked by pressing a proper key:

- Count up / count down with final sound of the buzzer.
- Display an alphanumeric character.
- Display a short phrase.

The software is structured in the following source files:

- DISPLAY.H

It is the Class definition file: it contains the declarations of the data and functions used inside the class.

- DISPLAY.C

It is the file containing class and functions implementation.

- DISPLAY_MAIN.C

It is the Main Program where we create an object of the class “Display” and recall its functions.

Appendices

SOURCE CODE LISTS

DISPLAY.H

```
class Display{
public:
    Display();
    Display(int port_addr);
    void Scrivi(char c);
    void Bip(int n);
    int Cerca (char d);
    void Contasu();
    void Contagiu();
    void Frase();
private:
    int port;
    void Clock();

};
```

DISPLAY.C

```
# include "display.h"
# include < sys/stat.h>
# include < sys/io.h>
# include < stdlib.h>
# include < iostream.h>
# include < stdio.h>
# include < unistd.h>

Display::Display(){
    port=0x378;
    if (ioperm(port,3,1)!=0) {
        cout<< "la porta non si apre"<< endl;
        exit(1);
    }
}

Display::Display(int port_addr){
    port=port_addr;
    if (ioperm(port,3,1)!=0) {
        cout<< "la porta non si apre"<< endl;
        exit(1);
    }
}

void Display::Bip(int n){
    char bip=0xff;
```

```

    Scrivi(bip);
    usleep(n);
    Scrivi(blk);
}
void Display::Clock(){
    outb(0x00,port+ 2);
    outb(0x01,port+ 2);
    outb(0x00,port+ 2);
}
int Display::Cerca(char d){
    char lista[2][29]={

        {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','L','N','O','P','R','S','T','U','Y'},
        {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x18,0x08,0x03,0x46,0x21,0x06,0x0e,0x42,0x09,0x79,
0x61,0x47,0x2b,0x23,0x0c,0x2f,0x12,0x07,0x63,0x19}
    };
    if (d> =97 && d< = 122) d= d-32;
    for (int i=0;i< 29;i++){
        if (lista[0][i]== d) return (int)lista[1][i];
    }
    return -1;
}
void Display::Scrivi(char c){
    outb(c,port);
    Clock();
}
void Display::Contasu(){
    const int numeri[10]= {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x18};
    for (int i=0;i< 10;i++){
        outb(numeri[i],port);
        Clock();
        sleep(1);
    }
}
void Display::Contagiu(){
    const int numeri[10]= {0x40,0x79,0x24,0x30,0x19,0x12,0x02,0x78,0x00,0x18};
    for (int i= 10;i> =0;i--){
        outb(numeri[i],port);
        Clock();
        sleep(1);
    }
}
void Display::Frase(){
    const int bella_ci[12]= {0x03,0x06,0x47,0x47,0x08,0x7f,0x46,0x4f,0xff,0x7f,0xff,0x7f};
    for (int i=0;i< 12;i++){
        outb(bella_ci[i],port);
        Clock();
        sleep(1);
    }
}
}

```

```

#include < iostream.h>
#include < stdio.h>
#include < stdlib.h>
#include < unistd.h>
#include < curses.h>
#include "display.h"

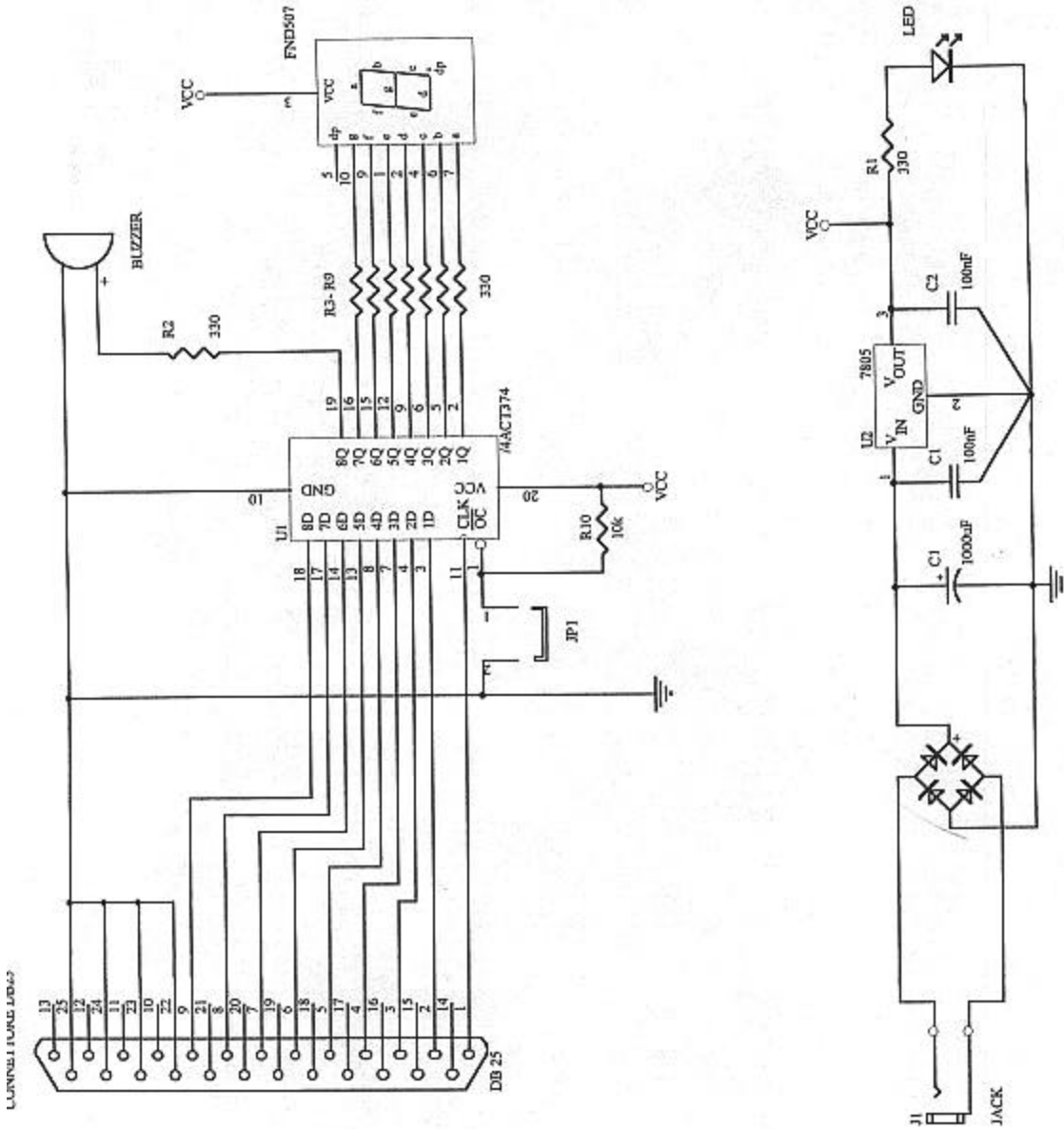
main(){
    int b;
    char a=99;

    Display Disp;
    while (a!=27){
        system("clear");
        cout << "\tQuesto e' il programma per la gestione del display" << endl;
        cout << endl;
        cout << "\tBIP          [$] " << endl;
        cout << "\tCONTA IN SU      [+] " << endl;
        cout << "\tCONTA IN GIU'   [-] " << endl;
        cout << "\tFRASE          [!] " << endl;
        cout << "\tCARATTERE ALFA-NUMERICO [a-z][0-9] " << endl;
        cout << "\tEXIT          [esc] " << endl;
        cin >> a;
        switch (a){
            case ' ':
                break;
            case '$':
                Disp.Bip(1000000);
                break;
            case '+':
                Disp.Contasu();
                Disp.Bip(1000000);
                break;
            case '-':
                Disp.Contagiu();
                Disp.Bip(1000000);
                break;
            case '!':
                Disp.Frase();
                break;
            default:
                b= Disp.Cerca(a);
                if (b< 0) {
                    system("xkbbell");
                    cout << "Carattere non valido\n";
                    sleep(1);
                }
                else {
                    Disp.Bip(20000);
                    Disp.Scrivi((char)b);
                }
        }
    }
}

```


THE INTERFACE CIRCUIT

SCHEMATIC DIAGRAM



PHOTOGRAPH OF THE CIRCUIT

