

6. Conclusions

- Développement logiciel et hardware
- Projet très varié
- Informations Pilot, assez difficile à aquérir

Logiciel:

- Programme Musée fonctionne comme prévu
- On arrive à commander le MiniDisc

- Améliorer l'interface graphique
- Langage script dans fichier interprété

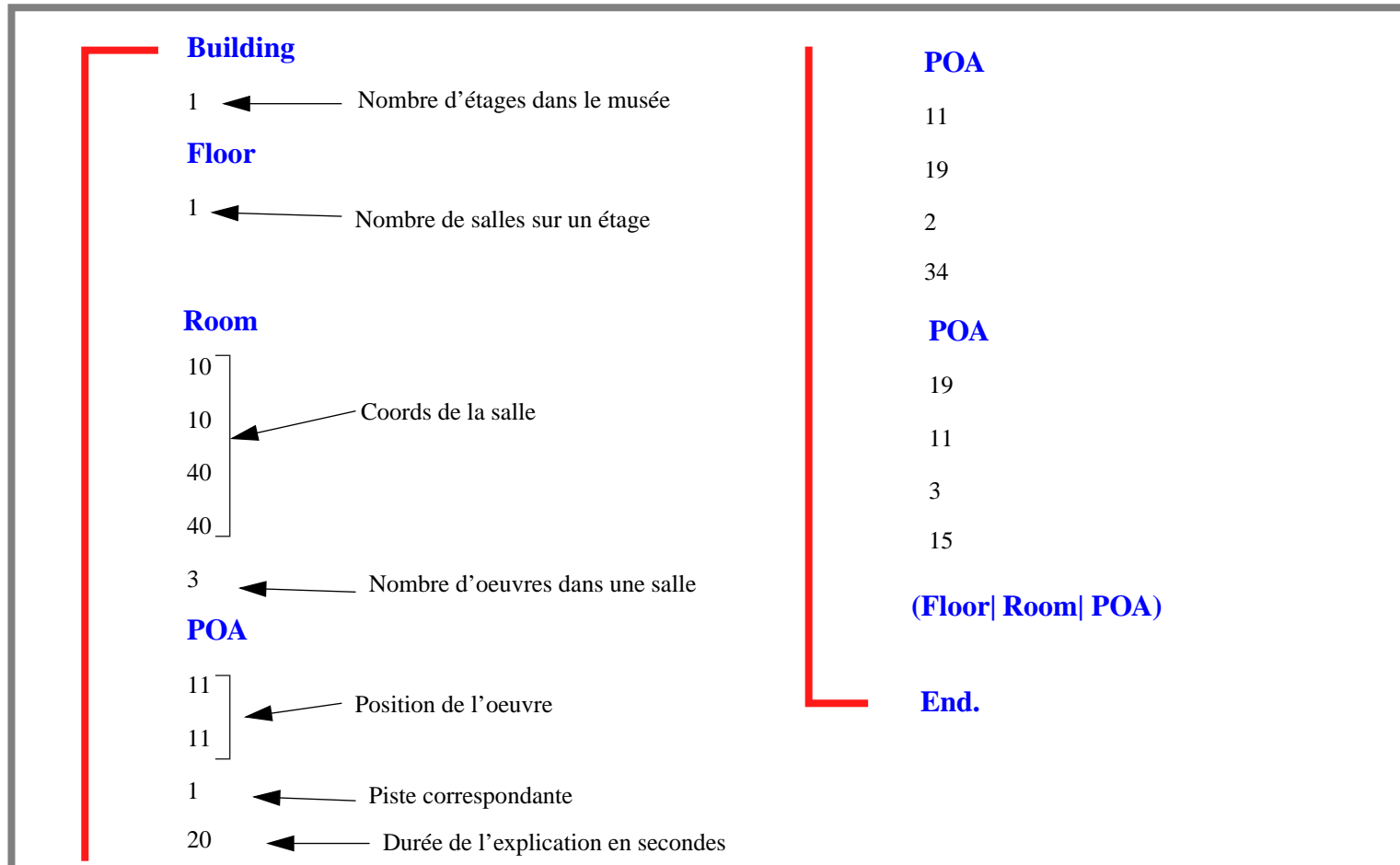
Hardware:

- L'interface fonctionne comme prévu
- "Bruits" de courant

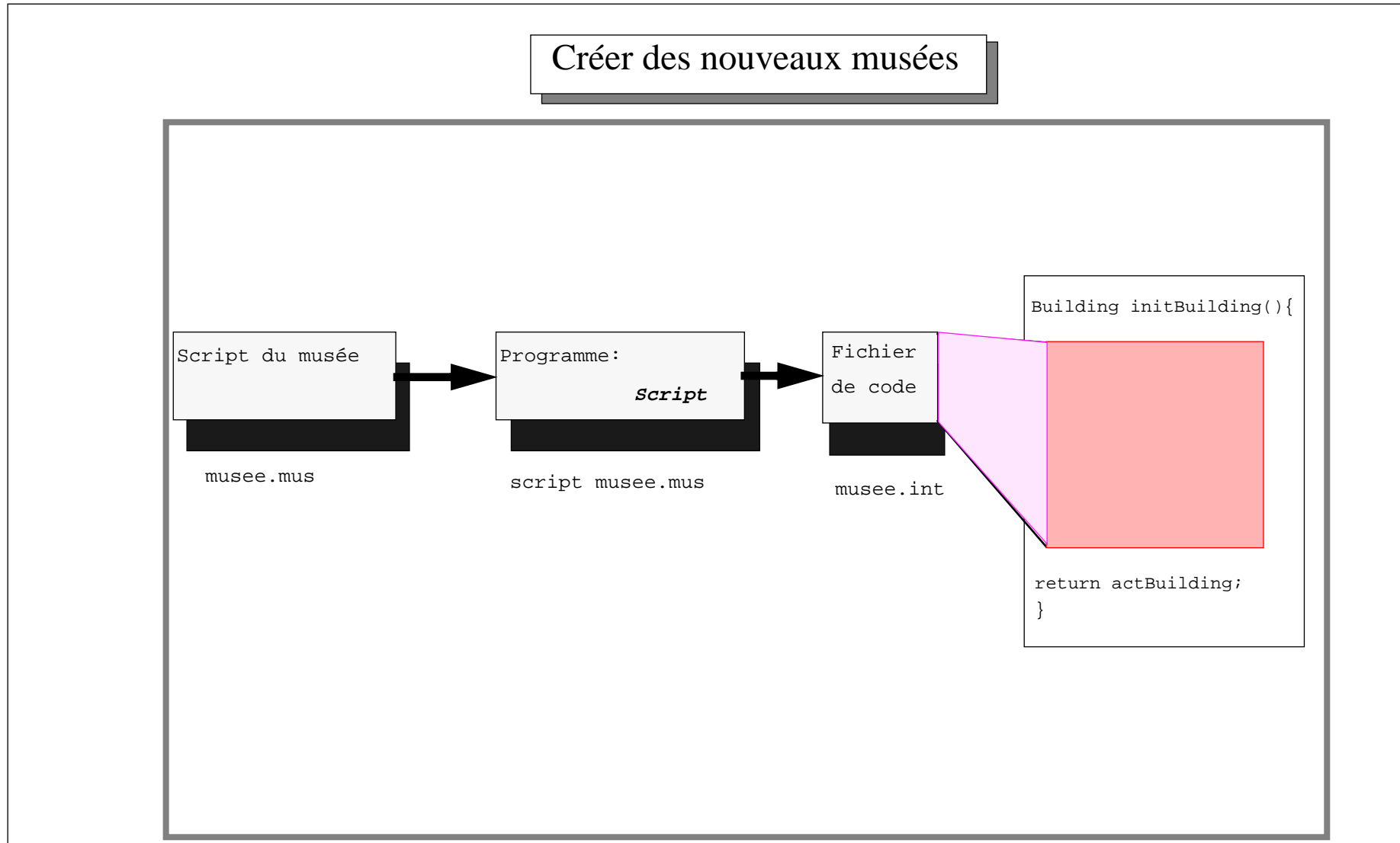
- Réaliser une solution one-chip

4. Le programme Musée (cont.)

Le langage script

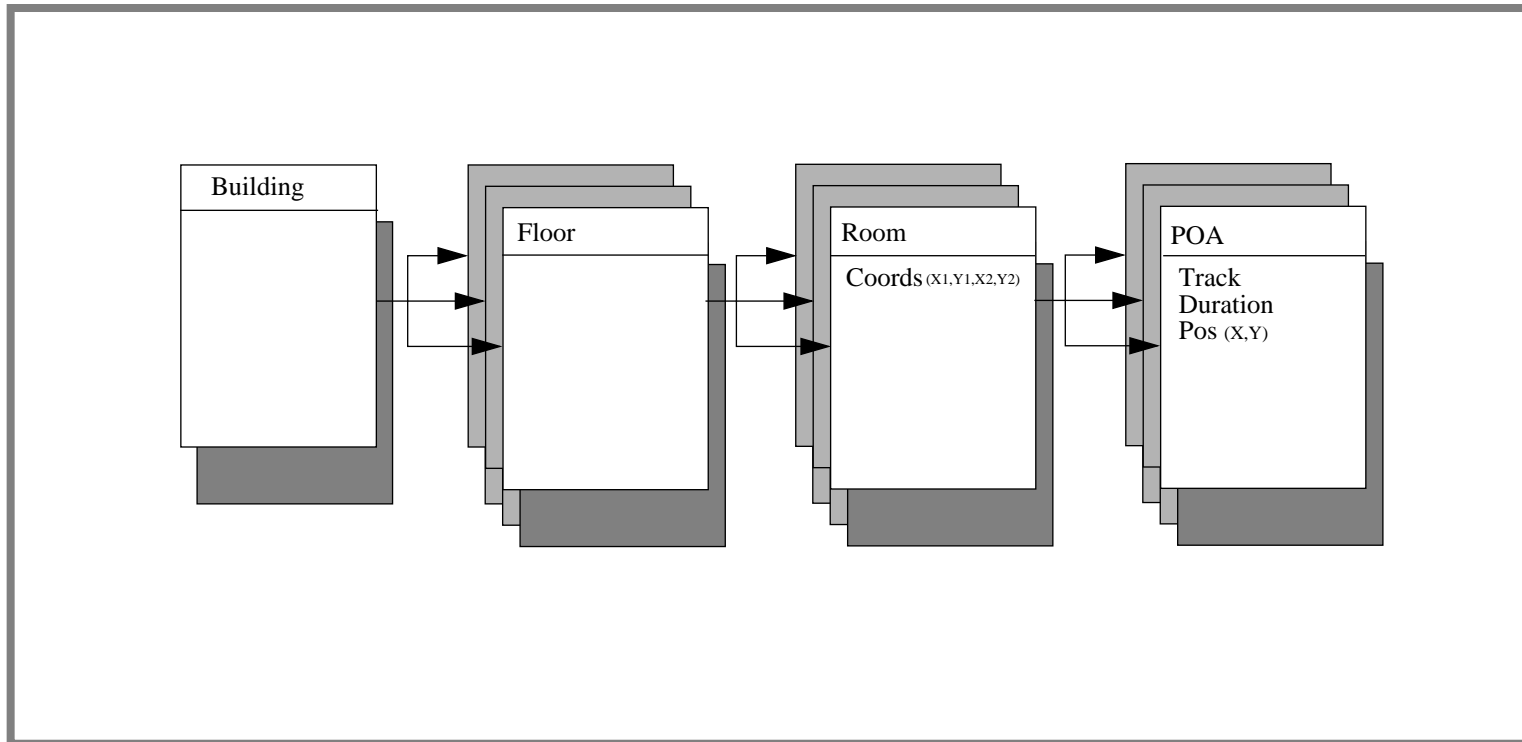


4. Le programme Musée (cont.)

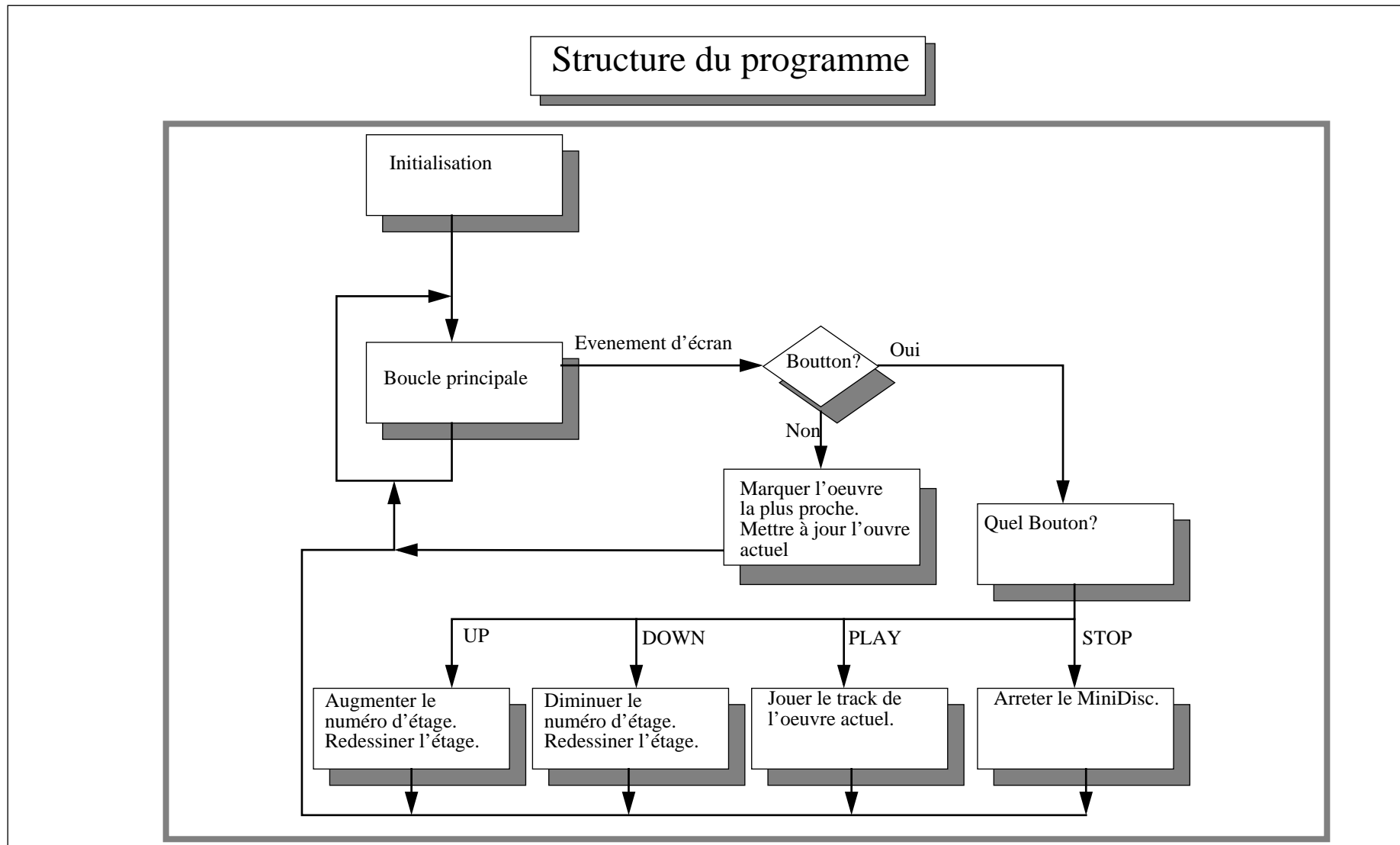


4. Le programme Musée (cont.)

Structure du programme:
Les objets utilisés



4. Le programme Musée (cont.)



4. Le programme Musée

L'interface utilisateur



Up Monter d'un étage

Down Descendre d'un étage

Play Donner l'explication pour l'oeuvre sélectionné

Stop Interrompre l'explication en cours

3.2 Jump - Pila (cont.)

Le fichier .res:
- contient la description des ressources

```
res 'WBMP', $7ffe, "Hello.bmp"  
res 'tFRM', 1000, "tFRM03e8.bin"
```

Nom de l'icône pour
la représentation Pilot

3.2 Jump - Pila (cont.)

Le fichier .rcp:

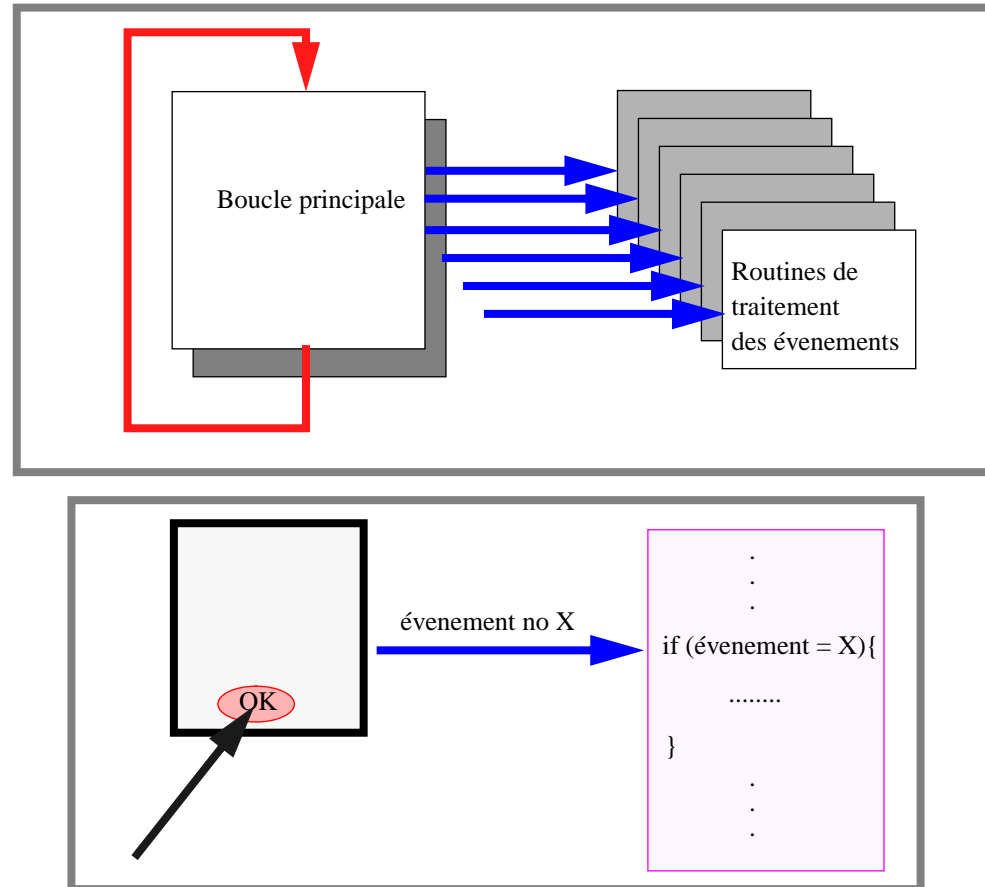
- contient la description de l'interface graphique
- doit être compilé avec PilRC

```
FORM (1000) 0 0 160 160
USABLE
NOFRAME
BEGIN
    TITLE "Hello World"
    LABEL "Hello, world!" 1001 60 70
END
```

Numéro d'identification
pour l'affichage

3.2 Jump - Pila (cont.)

Structure d'un programme



3.2 Jump - Pila (cont.)

```

import palmos.*;

class Hello {
    static final int idfMain = 1000;

    public static int (PilotMain) int cmd, int cmdBPB, int launchFlags,
    {
        if (cmd != 0) {
            return 0;
        }

        Palm.FrmGotoForm(idfMain);

        Event e = new Event();
        Short err = new Short();
        while (e.eType != Event.appStopEvent) {
            Palm.EvtGetEvent(e, -1);
            if (!Palm.SysHandleEvent(e)) {
                if (!Palm.MenuHandleEvent(0, e, err)) {
                    if (!appHandleEvent(e)) {
                        Palm.FrmHandleEvent(Palm.FrmGetActiveForm(), e);
                    }
                }
            }
        }
        return 0;
    }

    static boolean appHandleEvent(Event e)
    {
        if (e.eType == Event.frmLoadEvent) {
            int form = Palm.FrmInitForm(e.formID());
            Palm.FrmSetActiveForm(form);
            return true;
        } else if (e.eType == Event.frmOpenEvent) {
            Palm.FrmDrawForm(Palm.FrmGetActiveForm());
            return true;
        }
        return false;
    }
}
    
```

importation de la classe spécifique Pilot

classe de base contenant le programme

Programme principal

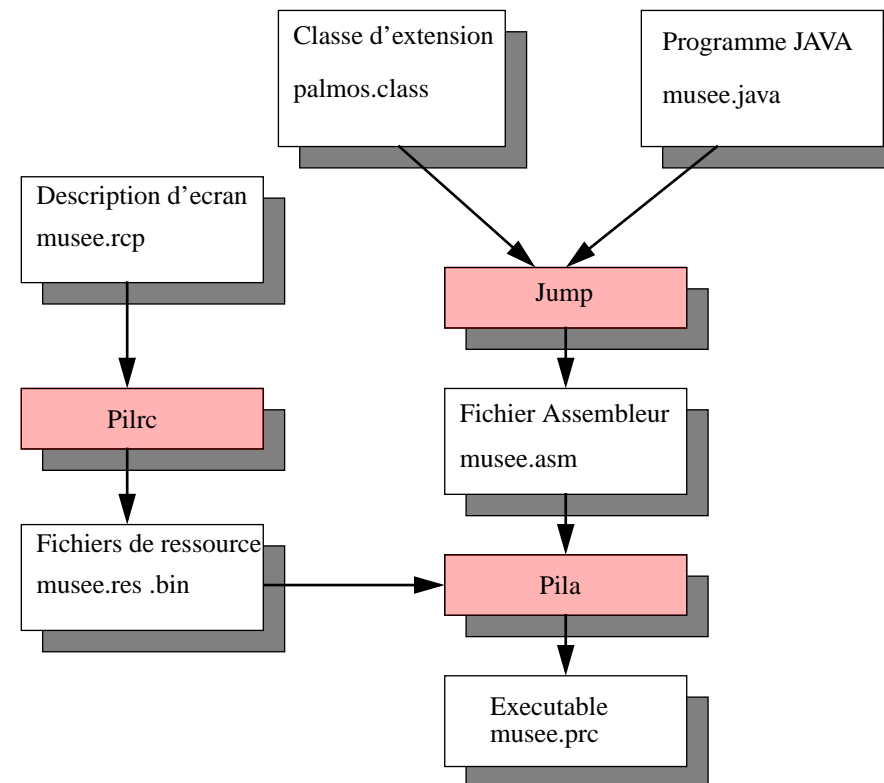
Boucle de lecture des événements

Routines de traitement des événements

3.2 Jump - Pila (cont.)

Programmation:

En Java avec la combinaison de Jump
et Pila.



3.1 Programmation en Palm OS - Possibilités

Possibilités de Programmation:

En Java avec la combinaison de Jump et Pila.

En C/C++ avec Code Warrior

On a choisi JAVA/Jump:

- logiciels disponibles pour PC
- logiciels sont Shareware

2.4 Le Pilot de USRobotics

Pilot 5000

Display: tactile, 160 x 160 pixels sans éclairage

Programmable en C/C++ ou Java

Mémoire: 512K

Autonomie: 8-12 semaines avec 2 piles AAA

Connexion série: RS-232C 9-Pin


Compatible Windows et Macintosh



2.3 Le Sony MiniDisc (cont.)

Commandé par niveaux de tension sur une ligne.

Toutes les commandes sont accessibles par des niveaux.

Prise de télécommande	
 1 2 3 4	1. Gnd 2. Télécommande 3. Volume limite 4. Vcc

Niveaux pour les commandes courantes					
Play/Fwd	Rew	Stop	Vol.Up	Vol.Down	Pause
2.2 V	3.0 V	2.9 V	1.8 V	1.9 V	2.1 V

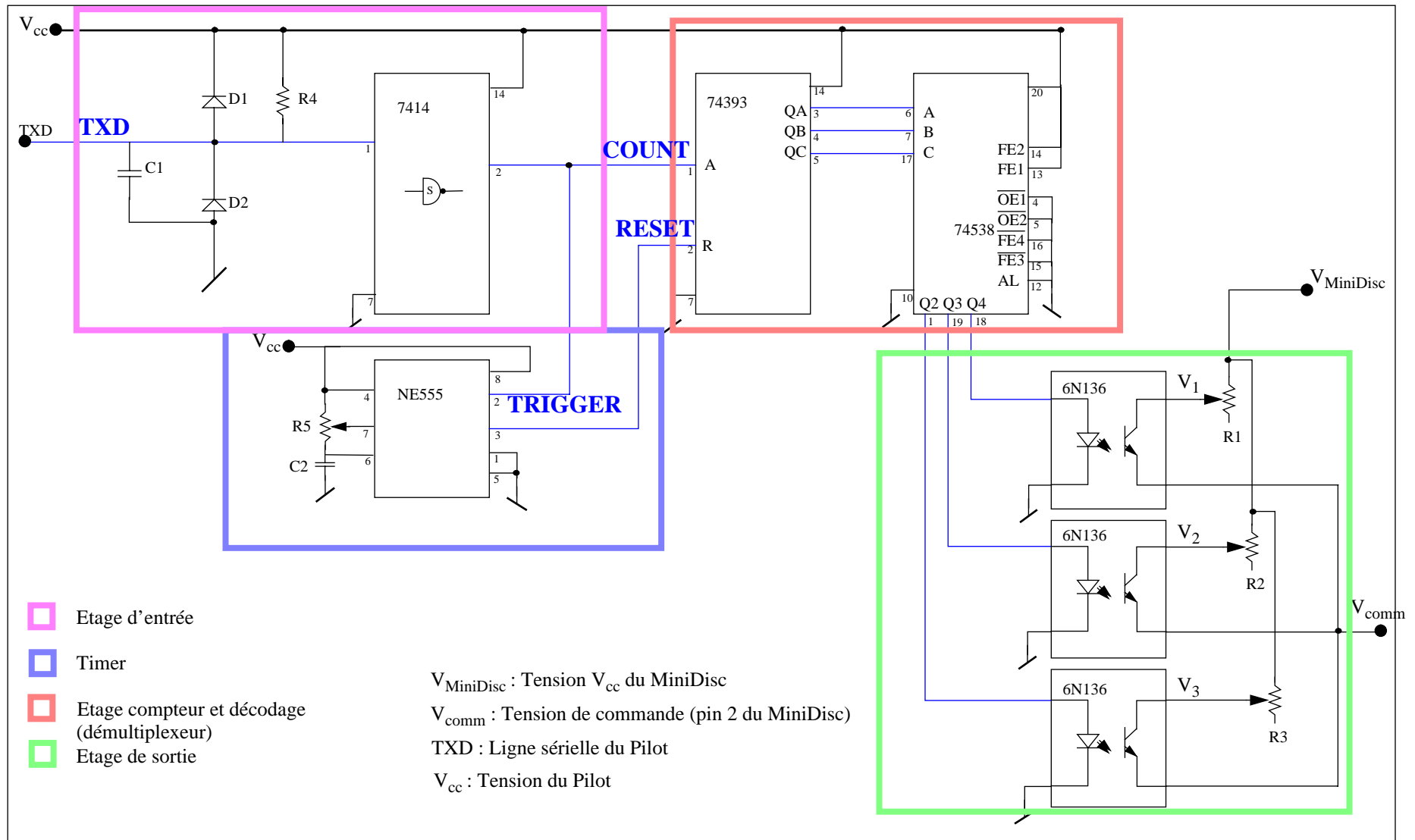
2.3 Le Sony MiniDisc

- Qualité CD.
- Possibilité d'enregistrement.
- Accès direct à des tracks.
- Possibilité de télécommander.



Le MZ-R3

2.2 Interface RS232 - MiniDisc (cont.)



2.2 Interface RS232 - MiniDisc

Seule possibilité: Utiliser les commandes de changement de track.
Générer des différents niveaux de tension à partir d'un signal sériel.

→ Solution facile: **Compter les flancs** pendant un certain temps et en déduire la commande à envoyer.

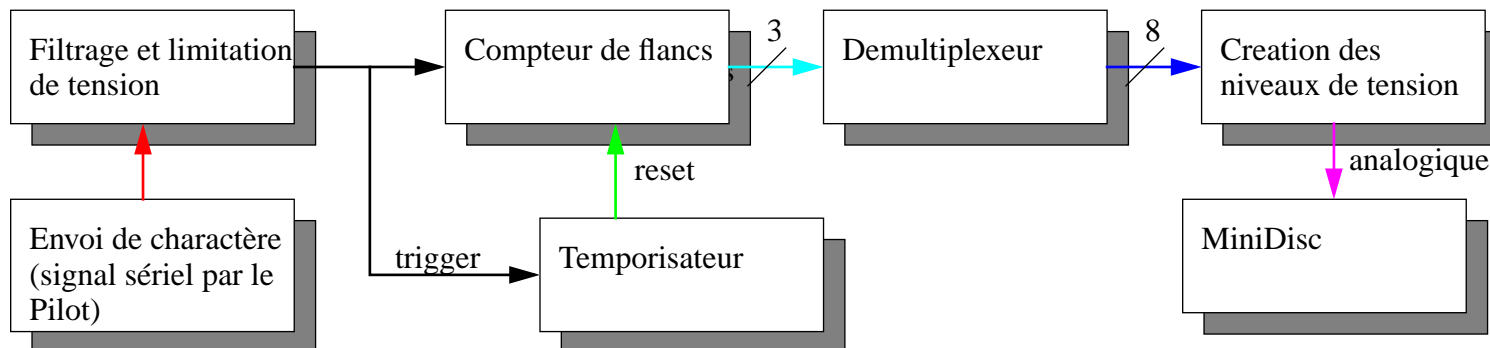
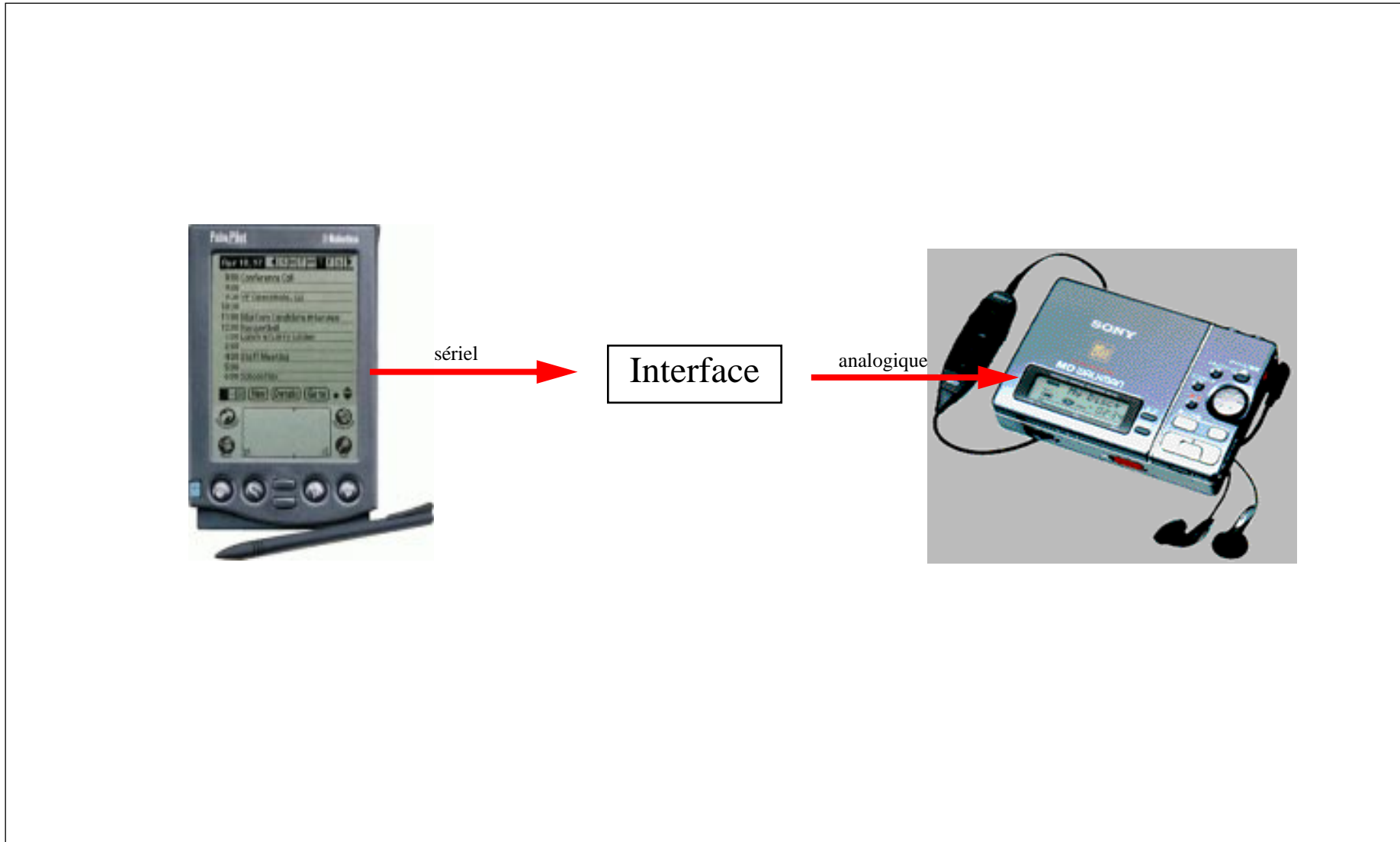


Schéma bloc

2.1. Idée de base



2. Solution choisie

Lecteur MiniDisc commandé par Pilot

- + - Coût modéré
- Echange de disque pour différentes langues
- Facile à enregistrer
- Résistant aux vibrations

- - Pas en une pièce (boîtier)

1. Introduction

Conception d'un guide de musée interactif sur un ordinateur portable, capable de donner les explications demandées par le visiteur.

Buts:

- commodité:** la solution doit être **légère** et **compacte** pour assurer la portabilité.
- facilité:** le dispositif doit être **facile à utiliser**.
- économie:** le dispositif doit **coûter le moins possible**.

On a plusieurs possibilités tout en essayant d'assurer le mieux les trois contraintes.

Table des Matières

1. Introduction
2. Solution choisie
 - 2.1 Idée de base
 - 2.2 Interface RS232 - MiniDisc
 - 2.3 Le Sony MiniDisc
 - 2.4 Le Pilot de USRobotics
3. Programmation en Palm OS
 - 3.1 Possibilités
 - 3.2 Jump - Pila
4. Le programme Musée
5. Remarques
6. Conclusion



Présentation finale
du projet de semestre

Guide interactif pour visite de musées et expositions sur pen computer

Niklaus Hirt
Info 4ème