

Projet de qualification :

**EFFET DE LUMIÈRE  
COMMANDÉ PAR  
ORDINATEUR**

Année 1999 – 2000  
Institut Robert Schuman  
Rosengarten Daniel

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

Table des matières :

1. Introduction
2. Plan général
3. Explications techniques avec plan électrique
4. Mode d'emploi
5. Programme C
6. Organigrammes
7. Lay-out
8. Documentation

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

1) Introduction :

Software :

Le montage sert à commander quatre moteurs pas à pas et un triac.

Le fichier d'exécution du programme est « ROBOSCAN.EXE ».

Le menu est en mode texte et tous le reste en mode graphique en résolution 640\*480  
16 couleurs.

Il est possible de programmer des séquences, chaque programmations est enregistrée  
dans un fichier « \*.ROB » coder en binaire. Si on lit une programmation le programme  
boucle jusqu'à ce que l'on pousse sur « q ».

Il est possible d'effacer un fichier d'une programmation depuis le menu du programme.

Il est également possible de jouer avec le roboscan en temps réel.

Hardware :

On utilise 8 bits de sorties et 2 bits d'entrée du port parallèle de l'ordinateur.

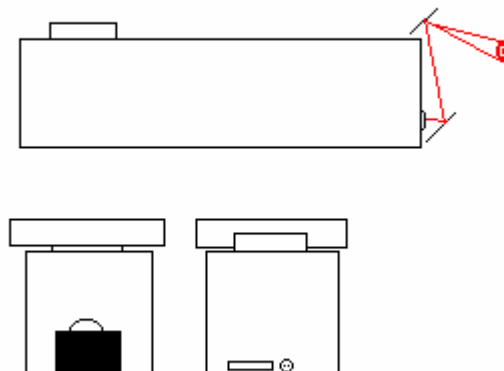
Les 8 bits de sorties servent à enclencher un triac, à commander les 4 circuits SAA  
1027 et à allumer les 2 capteurs optiques.

Les 6 signaux pour les circuits SAA 1027 passent d'abord par 6 coupleurs optiques  
pour avoir une isolation galvanique entre le port du PC et les circuits.

Les 2 derniers bits de sorties servent à allumer le triac qui est commander par un  
MOC 3041 qui fait une isolation galvanique entre le PC et le circuit et a enclenché les  
2 capteurs optiques.

La lumière passe à travers un filtre de couleur puis à travers une fente et pour  
terminer est focaliser sur le miroir à l'aide d'une lentille.

Un miroir peut pivoter vers la gauche et vers la droite et un autre vers le haut et vers le  
bas.



Rosengarten  
Daniel  
6A Elo

Lontzen, le 8 mai 2000

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

Plan général :

Rosengarten  
Daniel  
6A Elo

Lontzen, le 8 mai 2000

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

3) Explications des différentes parties :

**1<sup>ère</sup> partie : l'alimentation 2 x 12V / connecteur DB25 / ventilateur**

Plan : voir page « alimentation 2 x 12V / connecteur DB25 / ventilateur »

Le circuit représente une alimentation 2 x 12V.

Le transformateur est un transformateur à point milieu 220 / 0-12-24V 1A.

Le redressement est un redressement double alternance à point milieu.

$$12V * \sqrt{2} = 16,9V$$

$$16,9 V - 1,4V = 15,5V$$

il y a 1 condensateur de filtrage pour chaque tension

$$C = 0,7A / (100Hz * 7V) = 1000\mu F$$

Il n'y a pas eu besoin de mettre un régulateur de tension car les circuits SAA 1027 ont une tension de fonctionnement comprise entre 9,5V et 18V.

Il n'y a plus eu besoin d'un régulateur de tension pour le ventilateur.

## 2<sup>ème</sup> partie : Commande du triac

Plan : voir page « commande d'un triac par ordinateur »

Le montage représente la commande d'un triac par un pc.

Le circuit MOC 3041 se charge d'enclencher et de déclencher le triac. S'il reçoit un état haut sur la PIN 1 le triac est enclencher, s'il reçoit un état bas sur la PIN 1 le triac est déclencher.

Le branchement de ce circuit a été copié de la documentation du MOC 3041.

La résistance de  $39\Omega$  et le condensateur de  $0,01\mu\text{F}$  servent à protéger la sortie contre les parasites. Les résistances de  $330\Omega$  et de  $360\Omega$  servent à polariser le triac.

Le calcul de la résistance d'entrée du MOC 3041 a été basé sur les caractéristiques de la diode d'entrée de ce circuit.  $V_R = 6\text{V}$   $I_F = 30\text{mA}$

$$R_{in} = U / I = 6\text{V} / 25\text{mA} = 200 \Omega \rightarrow 220\Omega$$

PIN 1 : sortie du port parallèle

PIN 2 : masse du port parallèle

PIN 4 : gâchette du triac

PIN 6 : A2 du triac

PIN 3 et 5 : n.c.

## 3<sup>ème</sup> partie : circuit des capteurs optiques

Plan : voir page « capteurs optiques en fourche »

Le circuit est le circuit de 2 capteurs optiques en fourche.

Les résistances ont été testées jusqu'à ce que le signal ne soit ni trop fort ni trop faible pour le photo-transistor. C'était malheureusement la seule solution car nous n'avons pas trouvé de documentation sur ce circuit qui ne se trouve plus sur le marché.

Résistance pour la diode :

$$R = 47\Omega$$

Résistance pour le transistor :

$$R = 1\text{K}\Omega$$

## 4<sup>ème</sup> partie : commande des moteurs pas à pas

Plan : voir page « circuits de commande des moteurs pas à pas »

Le montage représente la commande de 4 moteurs pas à pas.

Le circuit SAA 1027 se charge de faire avancer le moteur d'un pas lorsqu'il reçoit une impulsion sur la PIN 15. Pour donner le sens de rotation il faut mettre 12V ou 0V sur la PIN 3.

Les circuits SAA 1027 sont isolé galvanique ment de du port parallèle par 6 opto-coupleurs.

Le branchement de ce circuit a été copié de la documentation du SAA 1027.

Séquence	PIN 3 = L				PIN 3 = H			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
0	L	H	L	H	L	H	L	H
1	H	L	L	H	L	H	H	L
2	H	L	H	L	H	L	H	L
3	L	H	H	L	H	L	L	H
0	L	H	L	H	L	H	L	H

PIN 3 : mode gauche/droite

PIN 4 : résistance externe de l'ampli

PIN 5 et 12 : masse

PIN 6 et 8 : 1<sup>er</sup> bobine du moteur

PIN 9 et 11 : 2<sup>ème</sup> bobine du moteur

PIN 13 : alimentation des sorties vers les bobines

PIN 14 : alimentation du circuit SAA 1027 avec filtrage

## 5<sup>ème</sup> partie : protection galvanique

La séparation galvanique entre le PC et les circuits est faite par le circuit CNY 74-2 qui contient 2 capteurs optiques.

J'ai testé la résistance d'entrée la meilleur pour ce circuit. Avec une résistance d'entrée de 1,8K $\Omega$  et une résistance de 10K en sortie on peut monter jusqu'à une fréquence de 300Khz en gardant encore un beau signal.

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

4) Mode d'emploi :

1) Installation :

introduisez la disquette d'installation dans le lecteur de disquettes  
Tapez A : puis INSTALL  
Pour entrer dans le programme, sur C:\  
Tapez CD ROBOSCAN puis ROBOSCAN

2) Raccordement du boîtier

Branchez la fiche dans une prise 230V 50Hz  
Branchez la fiche DB25 mâle au boîtier et l'autre côté de la fiche DB25 femelle au port  
parallèle du PC

3) Programme

1. Menu Principal

Pour utiliser le menu il suffit d'appuyer sur une des touches suivantes pour effectuer la  
tâche :

- 1 utilisation en direct
- 2 programmer une séquence
- 3 utiliser une programmation
- 4 supprimer une programmation
- E quitter le programme

2. Initialisation automatique

Les miroirs se recentrent et le disque des couleurs et le disque des formes se remettent  
sur une position initiale qui est la couleur mauve et la forme carré, pour finir la lampe  
est éteinte.

3. Utilisation en direct

Pour utiliser le roboscan en direct il suffit d'appuyer sur une des touches suivantes  
pour effectuer la tâche :

- U forme carrée
- I forme cercle

O forme triangle

P forme rond

J couleur mauve

K couleur jaune

L couleur verte

M couleur rouge

← miroir vers la gauche

↑ miroir vers le haut

→ miroir vers la droite

↓ miroir vers le bas

Q retour au menu principal

C changer le temps entre 2 actions

R réinitialiser le robosca

#### 4. Programmer une séquence

Il faut d'abord entrer le nom du fichier qui contiendra la programmation.

La programmation est faite avec une visualisation en direct pour pouvoir faire des réglages précis.

Pour programmer il suffit d'appuyer sur une des touches suivantes pour effectuer la tâche :

U forme carrée

I forme cercle

O forme triangle

P forme rond

J couleur mauve

K couleur jaune

L couleur verte

M couleur rouge

← miroir vers la gauche

↑ miroir vers le haut

→ miroir vers la droite

↓ miroir vers le bas

A annuler

Q retour au menu principal et fin de programmation

C changer le temps entre 2 actions

□ enregistrement du positionnement des élément

#### 5. Utiliser une séquence programmée

Il faut d'abord entrer le nom du fichier qui contient la programmation.

Ensuite l'ordinateur joue la séquence en boucle

Pour utiliser le roboscan en mode séquence il suffit d'appuyer sur une des touches suivantes pour effectuer la tâche :

Q retour au menu principal et fin de lecture

#### 6. Supprimer un fichier qui contient une programmation

Il faut entrer le nom du fichier qui contient la programmation et il sera automatiquement supprimé.

#### 7. Quitter

Retour au mode MS-DOS

## 8. Configuration nécessaire

PC compatible IBM

Carte graphique VGA

MS-DOS 6.0 (pas le MS-DOS 7.0 ou supérieur de WIN95, WIN98, WIN NT)

Rosengarten  
Daniel  
6A Elo

Lontzen, le 8 mai 2000

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

5) Programme C :

```

#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<io.h>
#include<fcntl.h>
#include<sys\stat.h>
#include<dos.h>
#include<graphics.h>
#include<complex.h>
#include<math.h>

#define tempo 10
#define mirgd 1
#define mirhb 2
#define col 4
#define fo 8
#define sensmir 16
#define senscofo 32
#define lamp 64
#define capt 128

struct str{
    int prog;
    int posx;
    int posy;
    int coul;
    int form;
    int tmp;
    int lmp;
}save;

long int del=0;
int
program=0,lecture=0,nf=0,cont=0,max=0,valeur=0,forme=1,couleur=1,lampe=0,
    positionx=0,positiony=0,charge=0,positionxp=0,positionyp=0,couleurp=1,
    formep=1;

void initialisation();
void direct();
void cadre();
long int saisiegraph(int x,int y,int nbcaract,int bkcolor, int txcolor);
void programmation();
void utilisation();
void port();
void supprime();

void main()
{
clrscr();
char choix;
int i;
initialisation();
do
    {
    _setcursortype(_NOCURSOR);
    textbackground(0);
    clrscr();
    cadre();
    textcolor(14);

```

```

textbackground(1);
gotoxy(28,10);
cprintf("1.Utilisation en direct");
gotoxy(28,12);
cprintf("2.Programmer une s,quence");
gotoxy(28,14);
cprintf("3.Utiliser une programmation");
gotoxy(28,16);
cprintf("4.Supprimer une programmation");
gotoxy(34,18);
cprintf("<ESC> = Sortie");
textbackground(0);
textcolor(7);
if(charge==1)
    choix='1';
else
    choix=getch();
switch(choix)
{
case'1':
{
if(charge==0)
    initialisation();
else
    charge=0;
direct();
break;
}
case'2':
{
initialisation();
programmation();
break;
}
case'3':
{
initialisation();
utilisation();
break;
}
case'4':
{
supprime();
break;
}
}
}while(choix!=27);
textbackground(0);
clrscr();
textcolor(3);
cprintf("Merci d'avoir utilis, un Programme Rosengarten Technologie\r\nNous
esp,rions que vous vous ^tes bien amus,?\r\nA la prochaine fois!");
textcolor(7);
}

void initialisation()                /*initialisation du roboscan*/
{
char resultat=0;
int i=0,j=0;
int driver=DETECT,mode=2,attente=0;
if(charge==0)

```

```

initgraph(&driver,&mode,"f:\\borlandc\\bgi");

setcolor(0);
for(j=0;j<480;j++)
    line(1,j,639,j);

setcolor(7);
settextstyle(2,0,8);
outtextxy(120,200,"Veuillez passientez s.v.p. ...");
setcolor(56);
rectangle(109,400,531,420);
setfillstyle(SOLID_FILL,7);
floodfill(121,401,56);
settextstyle(0,0,0);

forme=1;
couleur=1;
lampe=0;
positionx=0;
positiony=0;
del=0;
outport(0x378,capt);
do
{
    resultat=inportb(0x379);
    delay(20);
    if(resultat==0x3f)
    {
        //delay(1);
        //printf("%d) 1:OK    2:--\n",attente);
        outport(0x378,capt|col|mirgd|mirhb);
    }
    if(resultat==0x7f)
    {
        //delay(1);
        //printf("%d) 1:--    2:--\n",attente);
        outport(0x378,capt|mirgd|mirhb);
    }
    if(resultat==0xffff)
    {
        //delay(1);
        //printf("%d) 1:--    2:OK\n",attente);
        outport(0x378,capt|fo|mirgd|mirhb);
    }
    if(resultat==0xffbf)
    {
        //delay(1);
        //printf("%d) 1:OK    2:OK\n",attente);
        outport(0x378,capt|fo|col|mirgd|mirhb);
    }
    delay(20);
    outport(0x378,capt);
    attente++;
    if(charge==1)
        setcolor(4);
    else
        setcolor(1);
    line(109+i,401,109+i,419);
    line(110+i,401,110+i,419);
    line(111+i,401,111+i,419);
    line(112+i,401,112+i,419);
}

```

```

    i=i+4;
    }while(attente!=100);

for(j=0;j<22;j++)
{
    if(j<12)
        outport(0x378,mirgd|mirhb|sensmir);
    else
        outport(0x378,mirgd|sensmir);
    delay(20);
    outport(0x378,sensmir);
    delay(20);
    line(109+i,401,109+i,419);
    i++;
}
if(charge==0)
    closegraph();
outport(0x378,0);
}

void direct()
{
    int
    driver=DETECT,mode=2,color=7,change=0,change2=0,i=0,direction=0,cont2=0;
    char choix=0,vitesse[50],impr[6],etape[20];

    initgraph(&driver,&mode,"f:\\borlandc\\bgi");

    //ligne du menu

    setcolor(7);
    rectangle(1,1,639,12);
    setfillstyle(SOLID_FILL,7);
    floodfill(2,2,7);
    setcolor(4);
    outtextxy(3,3,"Q");
    setcolor(0);
    outtextxy(12,3,"uit");
    if(program==1)
    {
        setcolor(4);
        outtextxy(50,3,"C");
        setcolor(0);
        outtextxy(59,3,"hanger vitesse de r,action");
        setcolor(4);
        outtextxy(284,3,"A");
        setcolor(0);
        outtextxy(293,3,"nnuler");
    }
    else
    {
        if(lecture==0)
        {
            setcolor(4);
            outtextxy(50,3,"C");
            setcolor(0);
            outtextxy(59,3,"hanger vitesse de r,action");
            setcolor(4);
            outtextxy(284,3,"R");
            setcolor(0);
            outtextxy(293,3," ,initialisation");
        }
    }
}

```



```

}
case'u':
{ forme=1;break; }
case'i':
{ forme=2;break; }
case'o':
{ forme=3;break; }
case'p':
{ forme=4;break; }
case'j':
{ couleur=1;break; }
case'k':
{ couleur=2;break; }
case'l':
{ couleur=3;break; }
case'm':
{ couleur=4;break; }
case 0xd:
{
choix='.';
if(lampe==1)
lampe=0;
else
lampe=1;
break;
}
case 0x4b:
{
if(positionx>-45)
{
direction=1;
positionx--;
}
break;
}
case 0x48:
{
if(positiony<25)
{
direction=2;
positiony++;
}
break;
}
case 0x4d:
{
if(positionx<45)
{
direction=3;
positionx++;
}
break;
}
case 0x50:
{
if(positiony>-25)
{
direction=4;
positiony--;
}
break;
}

```

```

    }
    case ' ':
    {
        if(program==1)
        {
            choix='.';
            save.prog=cont;
            save.posx=positionx;
            save.posy=positiony;
            save.coul=couleur;
            save.form=forme;
            save.tmp=del;
            save.lmp=lampe;
            write(nf,&save,sizeof(str));
            cont++;
            change2=1;
        }
        break;
    }
    case 'a':
    {
        if(program==1 && cont>0)
        {
            choix='.';
            cont--;
            save.prog=-1;
            save.posx=-1;
            save.posy=-1;
            save.coul=-1;
            save.form=-1;
            save.tmp=-1;
            save.lmp=-1;
            lseek(nf,cont*(sizeof(str)),SEEK_SET);
            write(nf,&save,sizeof(str));

            cont2=cont-1;
            if(cont2>0)
            {
                lseek(nf,cont2*sizeof(str),SEEK_SET);
                read(nf,&save,sizeof(str));
                positionx=save.posx;
                positiony=save.posy;
                couleur=save.coul;
                forme=save.form;
                del=save.tmp;
                lampe=save.lmp;
                lseek(nf,cont*(sizeof(str)),SEEK_SET);
            }
            else
                lseek(nf,0,SEEK_SET);
            change2=1;
        }
    }
}

/***** affichage *****/

//afficher l',tape

if(program==1 || lecture==1)

```



```

//bouton du trou

if(forme==4) { color=1; }
else        { color=7; }
setcolor(color);
setfillstyle(SOLID_FILL,color);
rectangle(260,200,310,250);
floodfill(261,201,color);
setfillstyle(SOLID_FILL,0);
setcolor(0);
circle(285,225,15);
floodfill(286,226,0);

//bouton de la couleur violet

if(couleur==1) { color=5; }
else          { color=0; }
setcolor(5);
rectangle(50,270,100,320);
setfillstyle(SOLID_FILL,color);
floodfill(51,271,color);

//bouton de la couleur jaune

if(couleur==2) { color=62; }
else          { color=0; }
setcolor(62);
rectangle(120,270,170,320);
setfillstyle(SOLID_FILL,color);
floodfill(121,271,color);

//bouton de la couleur vert

if(couleur==3) { color=2; }
else          { color=0; }
setcolor(2);
rectangle(190,270,240,320);
setfillstyle(SOLID_FILL,color);
floodfill(191,271,color);

//bouton de la couleur rouge

if(couleur==4) { color=4; }
else          { color=0; }
setcolor(4);
rectangle(260,270,310,320);
setfillstyle(SOLID_FILL,color);
floodfill(261,271,color);

//bouton lampe

if(lampe==1) { color=1; }
else        { color=7; }
setcolor(color);
setfillstyle(SOLID_FILL,color);
rectangle(420,235,470,285);
floodfill(421,236,color);
setcolor(0);
outtextxy(427,258,"LIGHT");

```

```
//bouton de direction gauche
```

```
setcolor(7);  
setfillstyle(SOLID_FILL,7);  
rectangle(360,235,410,285);  
floodfill(361,236,7);  
setcolor(0);  
line(370,260,400,260);  
putpixel(371,259,0);  
putpixel(371,261,0);
```

```
if(direction==1)  
{  
  choix='.';  
  direction=0;  
  setcolor(1);  
  setfillstyle(SOLID_FILL,1);  
  rectangle(360,235,410,285);  
  floodfill(361,236,1);  
  setcolor(0);  
  line(370,260,400,260);  
  putpixel(371,259,0);  
  putpixel(371,261,0);  
  delay(del);  
}
```

```
//bouton de direction haut
```

```
setcolor(7);  
setfillstyle(SOLID_FILL,7);  
rectangle(420,175,470,225);  
floodfill(421,176,7);  
setcolor(0);  
line(445,185,445,215);  
putpixel(444,186,0);  
putpixel(446,186,0);
```

```
if(direction==2)  
{  
  choix='.';  
  direction=0;  
  setcolor(1);  
  setfillstyle(SOLID_FILL,1);  
  rectangle(420,175,470,225);  
  floodfill(421,176,1);  
  setcolor(0);  
  line(445,185,445,215);  
  putpixel(444,186,0);  
  putpixel(446,186,0);  
  delay(del);  
}
```

```
//bouton de direction droite
```

```
setcolor(7);  
setfillstyle(SOLID_FILL,7);  
rectangle(480,235,530,285);  
floodfill(481,236,7);  
setcolor(0);  
line(490,260,520,260);
```







```

    }
    if(enter==0x0d || i==7)
    {
        fichier[i]='.';
        fichier[i+1]='r';
        fichier[i+2]='o';
        fichier[i+3]='b';
        fichier[i+4]='\0';
        i=8;
    }
}
nf=open(fichier,O_CREAT|O_BINARY|O_RDWR,S_IREAD|S_IWRITE);
if(nf==-1)
{
    textcolor(4+128);
    textbackground(0);
    clrscr();
    gotoxy(17,10);
    printf("!!! ERREUR : IMPOSSIBLE DE CREER LE FICHIER !!!");
    textcolor(15);
    textbackground(0);
    gotoxy(14,25);
    printf("Appuyer sur une touche pour revenir au menu principale");
    getch();
}
if(nf!=-1)
{
    cont=0;
    program=1;
    direct();
    if(cont==0)
    {
        open(fichier,O_TRUNC|O_RDWR,S_IREAD|S_IWRITE);
    }
    else
    {
        save.prog=-1;
        save.posx=-1;
        save.posy=-1;
        save.coul=-1;
        save.form=-1;
        save.tmp=-1;
        save.lmp=-1;
        write(nf,&save,sizeof(str));
    }
    close(nf);
    cont=0;
    program=0;
}
textcolor(15);
textbackground(0);
}

void utilisation()
{
    int i=0,j=0,mem=0;
    char fichier[13],enter;
    clrscr();
    textcolor(15);
    textbackground(1);
    gotoxy(25,10);

```



```

    max=save.prog;
    }while(max!=-1 && max!=0);
if(max!=0)
{
    max=j;
    j=0;
    lecture=1;
    cont=0;
    direct();
    lecture=0;
    cont=0;
}
else
{
    textcolor(4+128);
    textbackground(0);
    clrscr();
    gotoxy(15,10);
    cprintf("!!! ERREUR : LE FICHER EST VIDE, SUPRIMEZ LE  !!!");
    textcolor(15);
    textbackground(0);
    gotoxy(14,25);
    cprintf("Appuyer sur une touche pour revenir au menu principale");
    getch();
}
}
close(nf);
textcolor(15);
textbackground(0);
}

void port()
{
int tab[4][4]={ {0,12,24,-12} , {-12,0,12,24} ,
                {24,-12,0,12} , {12,24,-12,0} };
int j=1,i=0;

    //lampe

if(lampe==0)
{
    //disque des couleurs

j=tab[couleurp-1][couleur-1];
couleurp=couleur;

if(j>0)
{
    for(i=0;i<j;i++)
    {
        outport(0x378,col|senscofo);
        delay(tempo);
        outport(0x378,0);
        delay(tempo);
    }
}
if(j<0)
{
    for(i=0;i>j;i--)
    {
        outport(0x378,col);
    }
}
}

```

```

    delay(tempo);
    outport(0x378,0);
    delay(tempo);
}

//disque des formes

j=tab[formep-1][forme-1];
formep=forme;
if(j>0)
{
    for(i=0;i<j;i++)
    {
        outport(0x378,fo|senscofo);
        delay(tempo);
        outport(0x378,0);
        delay(tempo);
    }
}
if(j<0)
{
    for(i=0;i>j;i--)
    {
        outport(0x378,fo);
        delay(tempo);
        outport(0x378,0);
        delay(tempo);
    }
}

//position x

if(positionx!=positionxp)
{
    j=positionx;
    if(positionxp<positionx)
    {
        do {
            outport(0x378,mirgd|sensmir);
            delay(tempo);
            outport(0x378,0);
            delay(del);
            j++;
        }while(positionxp>j);
    }
    else
    {
        do {
            outport(0x378,mirgd);
            delay(tempo);
            outport(0x378,0);
            delay(del);
            j--;
        }while(positionxp<j);
    }
    positionxp=positionx;
}

//position y

```

```

if(positiony!=positionyp)
{
j=positiony;
if(positionyp<positiony)
{
do {
outport(0x378,mirhb|sensmir);
delay(tempo);
outport(0x378,0);
delay(del);
j++;
}while(positionyp>j);
}
else
{
do {
outport(0x378,mirhb);
delay(tempo);
outport(0x378,0);
delay(del);
j--;
}while(positionyp<j);
}
positionyp=positiony;
}
outport(0x378,0);
}

else

{
//disque des couleurs

j=tab[couleurp-1][couleur-1];
couleurp=couleur;

if(j>0)
{
for(i=0;i<j;i++)
{
outport(0x378,col|senscofo|lamp);
delay(tempo);
outport(0x378,lamp);
delay(tempo);
}
}
if(j<0)
{
for(i=0;i>j;i--)
{
outport(0x378,col|lamp);
delay(tempo);
outport(0x378,lamp);
delay(tempo);
}
}

//disque des formes

j=tab[formep-1][forme-1];
formep=forme;

```

```

if(j>0)
{
  for(i=0;i<j;i++)
  {
    outport(0x378,fo|senscofo|lamp);
    delay(tempo);
    outport(0x378,lamp);
    delay(tempo);
  }
}
if(j<0)
{
  for(i=0;i>j;i--)
  {
    outport(0x378,fo|lamp);
    delay(tempo);
    outport(0x378,lamp);
    delay(tempo);
  }
}

//position x

if(positionx!=positionxp)
{
  j=positionx;
  if(positionxp<positionx)
  {
    do {
      outport(0x378,mirgd|sensmir|lamp);
      delay(tempo);
      outport(0x378,lamp);
      delay(del);
      j++;
    }while(positionxp>j);
  }
  else
  {
    do {
      outport(0x378,mirgd|lamp);
      delay(tempo);
      outport(0x378,lamp);
      delay(del);
      j--;
    }while(positionxp<j);
  }
  positionxp=positionx;
}

//position y

if(positiony!=positionyp)
{
  j=positiony;
  if(positionyp<positiony)
  {
    do {
      outport(0x378,mirhb|sensmir|lamp);
      delay(tempo);
      outport(0x378,lamp);
      delay(del);
    }
  }
}

```

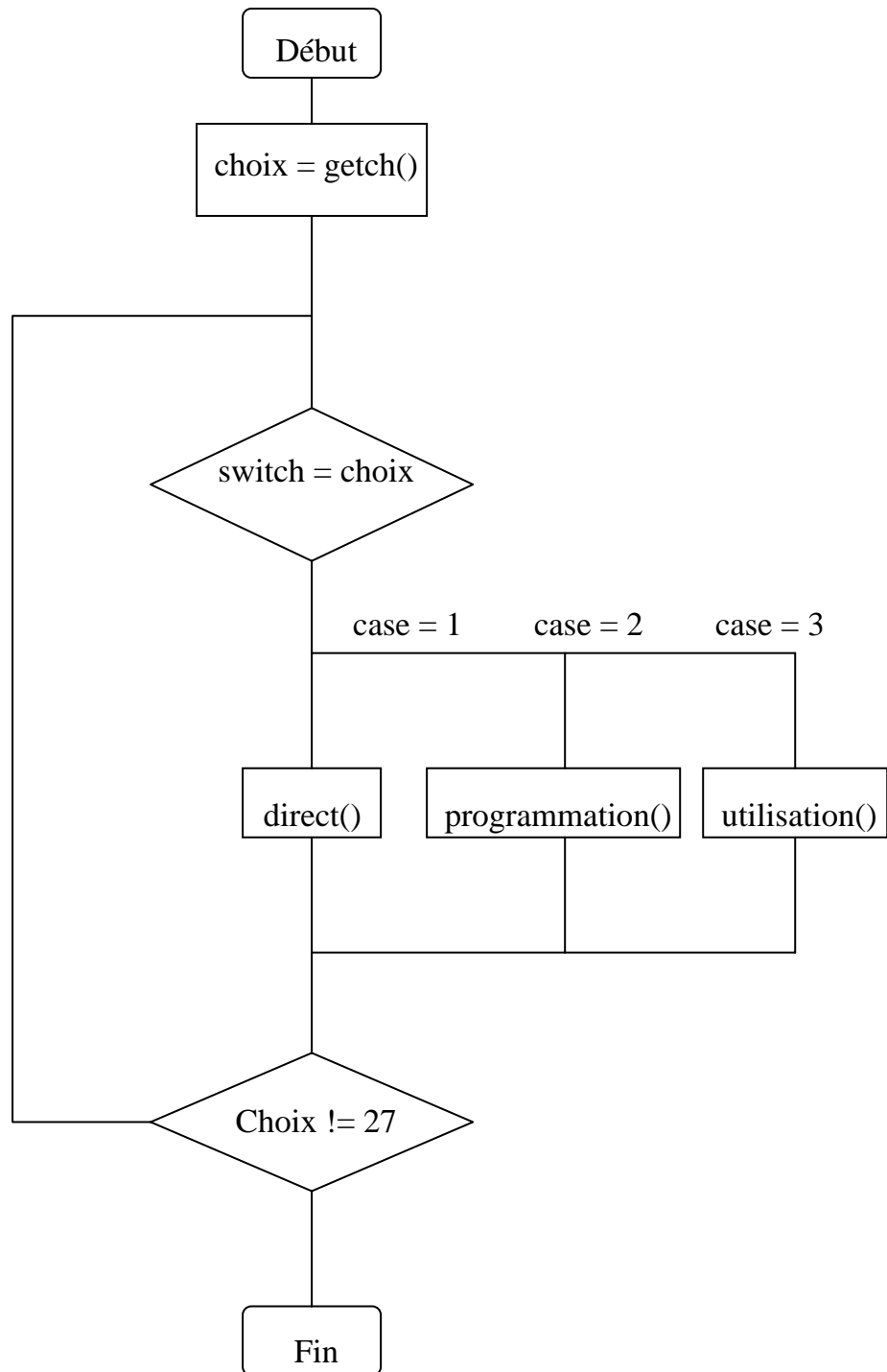


```
        fichier[i+2]='o';
        fichier[i+3]='b';
        fichier[i+4]='\0';
        i=8;
    }
}
strcpy(com,"del ");
strcat(com,fichier);
system(com);
}
```

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

**6) Organigrammes :**

**Void main :**



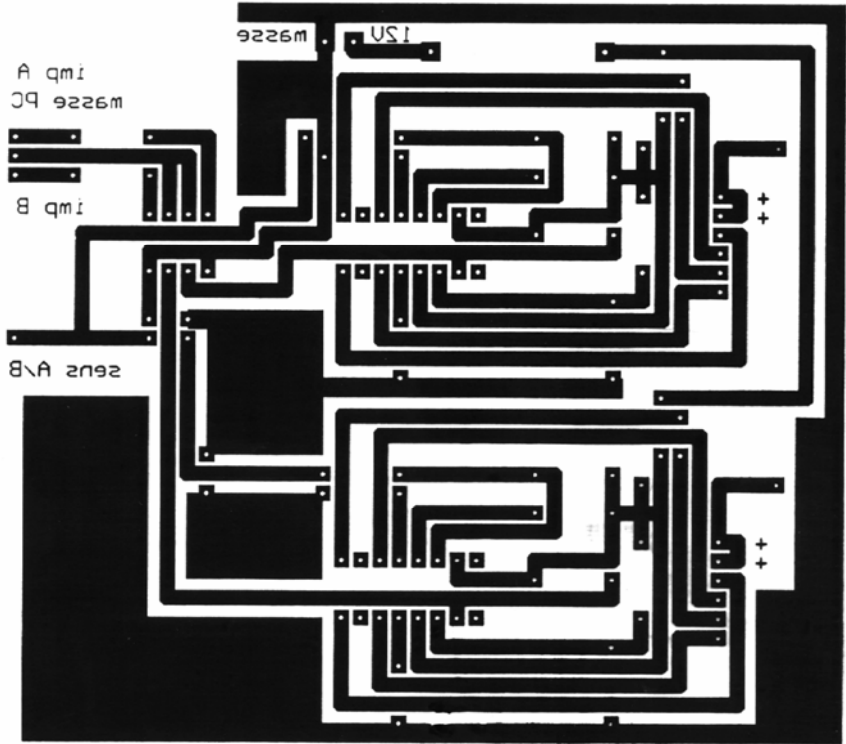
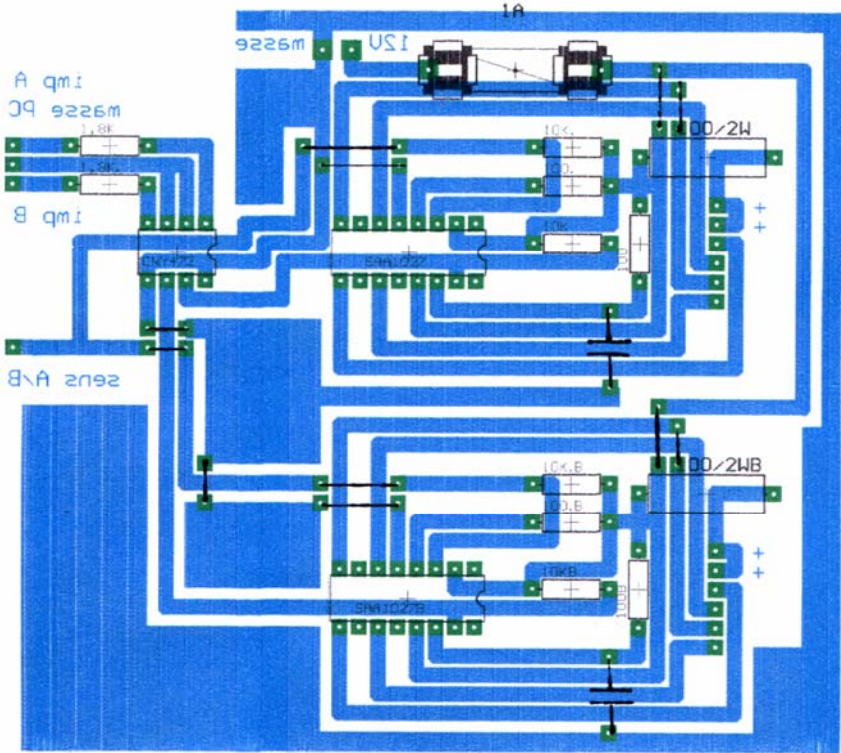
Rosengarten  
Daniel  
6A Elo

Lontzen, le 8 mai 2000

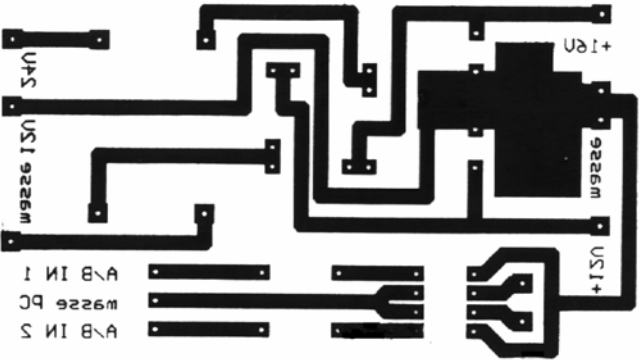
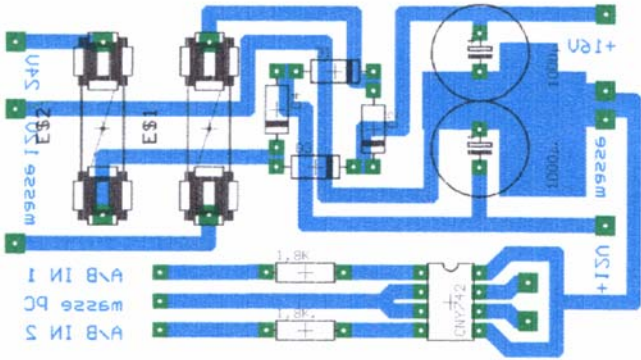
**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

7) Lay-out :

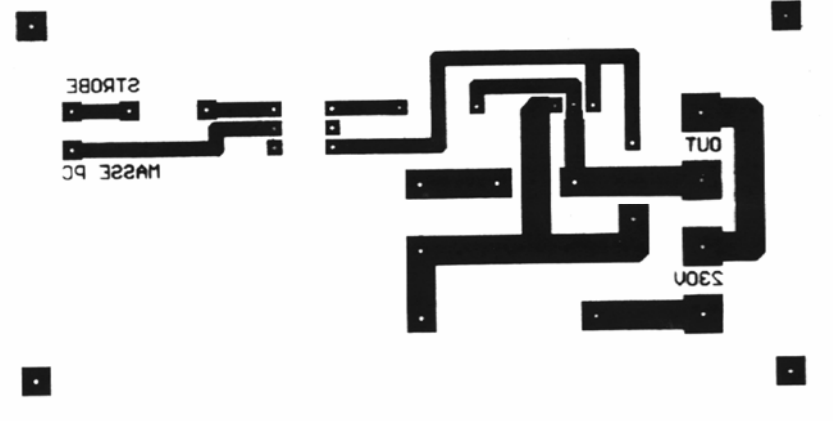
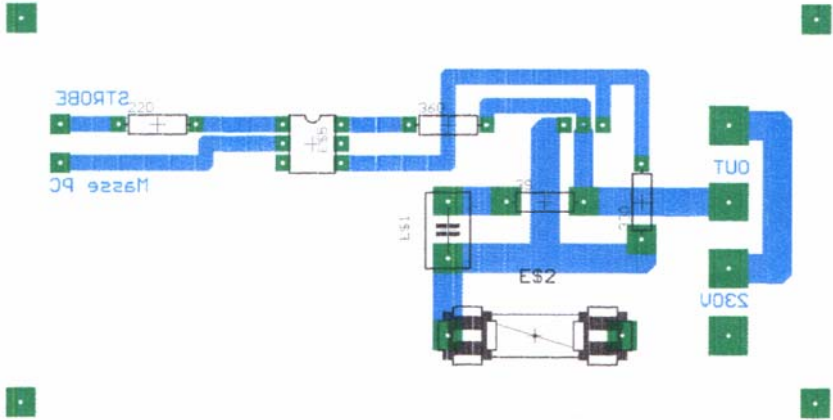
Commande de 2 moteurs pas à pas + 1 opto-coupleur.



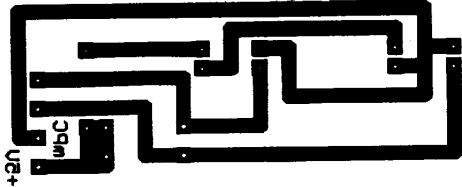
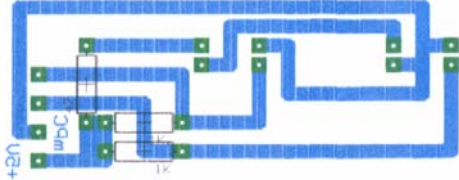
Alimentation 2 x 12V + 1 opto-coupleur



commande du triac avec un circuit MOC 3041



Capteurs optiques en fourche



Rosengarten  
Daniel  
6A Elo

Lontzen, le 8 mai 2000

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

8) Documentation :

Rosengarten  
Daniel  
6A Elo

Lontzen, le 8 mai 2000

**Projet de Qualification : « Effet de lumière commandé par ordinateur »**

Liste de matériel :

1 X MOC 3041  
2 X CNY 37  
3 X CNY 742  
1 X TIC 206 D  
4 X SAA 1027  
4 X socket 16 PIN  
3 X socket 8 PIN  
1 X socket 6 PIN  
1 X socket ampoule E4  
1 X transformateur 0-24V / 10A  
1 X transformateur à point milieu 0-12-24V / 1A  
4 X diode 1N4007  
8 X résistance 10 K $\Omega$  / 0,5 Watt  
8 X résistance 100  $\Omega$  / 0,5 Watt  
4 X résistance 100  $\Omega$  / 2 Watt  
1 X résistance de 390  $\Omega$  / 0,5 Watt  
1 X résistance de 330  $\Omega$  / 0,5 Watt  
1 X résistance de 220  $\Omega$  / 0,5 Watt  
6 X résistance de 1,8 K $\Omega$  / 0,5 Watt  
1 X résistance de 1 K $\Omega$  / 0,5 Watt  
1 X résistance de 47  $\Omega$  / 0,5 Watt  
4 X condensateur de 0,1 $\mu$ F  
2 X condensateur de 1000  $\mu$ F  
6 X porte fusible  
5 X fusible 1A / 230V  
1 X fusible 10A / 230V  
1 X ampoule halogène 24V / 150 Watt → Philips 7158 socket E4  
1 X ventilateur 16V DC