

CHAPITRE 4. MISE EN ŒUVRE DE LA MAQUETTE DE SIMULATION

4.1. Introduction

Le but de notre application est de mettre en œuvre les connaissances apprises précédemment afin de simuler en modèle réduit un système automatisé piloté par un ordinateur. Celui-ci permet l'allumage des feux de croisement d'un carrefour commandé grâce au port parallèle d'un PC avec un logiciel programmé avec Free pascal.

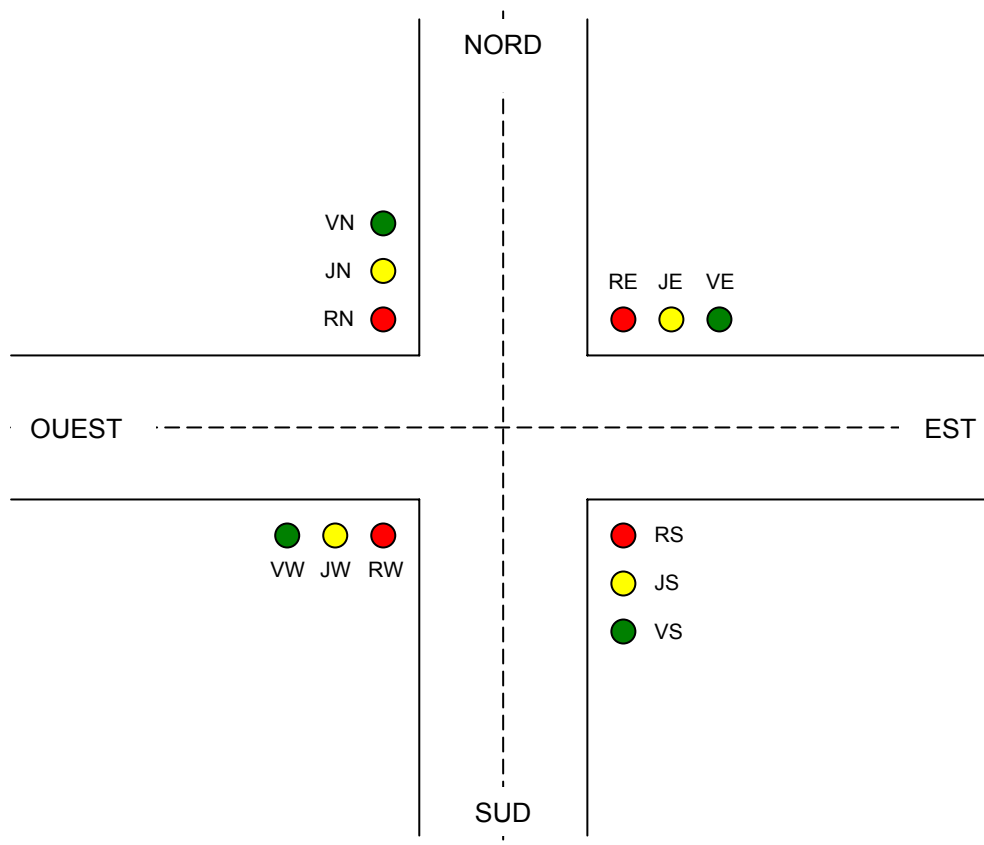
4.2. Description de la mise en œuvre

Nous supposons que ce carrefour est composé de deux routes dont une à un débit par heure de véhicules très important, l'autre à un débit moins important.

La route principale sera celle avec le débit de voiture important et la route secondaire avec le débit plus faible. Le principe de fonctionnement étant que quand dans un axe le feu est au vert, dans l'autre celui-ci soit au rouge, le feu orange n'intervenant que pour prévenir aux automobilistes le passage imminent au feu rouge.

Sur la route principale (Axe Nord – Sud), le temps permettant le franchissement du carrefour sera fixé à 10 secondes (durée d'allumage du feu vert). Sur la route secondaire (Axe Est – Ouest), le temps de franchissement du carrefour sera de 7 secondes (durée d'allumage du feu vert).

Sur les deux routes, la durée des feux orange est fixée à 3 secondes. Pendant qu'un des feux est à l'orange, il faut bien évidemment que les feux sur l'autre route reste au rouge.



Comme nous pouvons le voir avec la figure ci-dessus, nous aurons besoin de 3 LEDs (Rouge, Orange et Verte) à chaque point cardinal :

$$4 \times 3 = 12 \text{ LEDs}$$

au total pour les 4 points cardinaux.

Le tableau ci-dessous nous montre la disposition de ces 12 LEDs :

Route / Axe	Point Cardinal	Couleur LED	Nomenclature
Route Principale (Axe Nord - Sud)	NORD	LED Rouge	RN
		LED Orange	JN
		LED Verte	VN
	SUD	LED Rouge	RS
		LED Orange	JS
		LED Verte	VS
Route Secondaire (Axe Est - Ouest)	EST	LED Rouge	RE
		LED Orange	JE
		LED Verte	VE
	OUEST	LED Rouge	RW
		LED Orange	JW
		LED Verte	VW

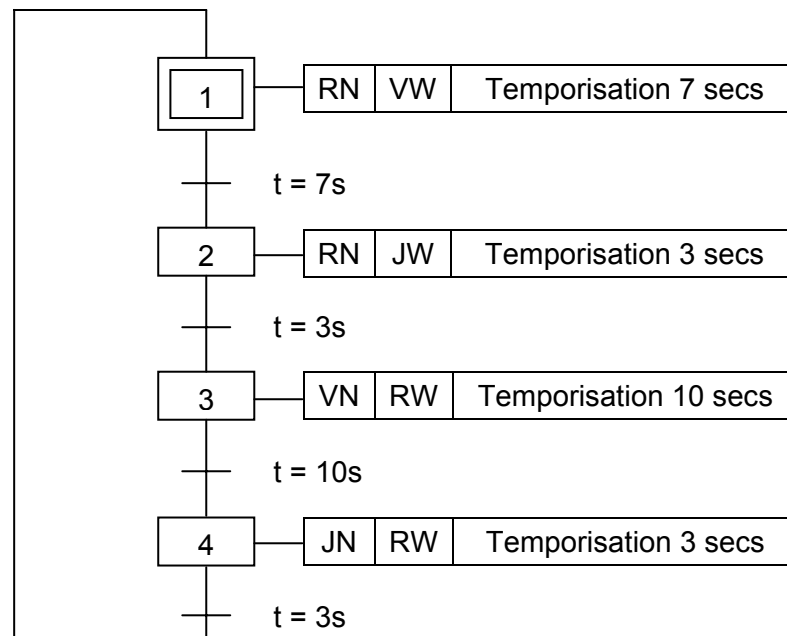
4.3. Analyse des étapes d'allumage des LEDS

Vu que l'état des LEDs d'un point cardinal sera identique à celui du point opposé, nous nous simplifierons la tâche en n'étudiant que l'état de 6 LEDs. Ce qui nous amène au tableau ci-dessous.

Route / Axe	Point Cardinal	Couleur LED	Simplification
Route Principale (Axe Nord - Sud)	NORD	LED Rouge	RN = RS
		LED Orange	JN = JS
		LED Verte	VN = VS
Route Secondaire (Axe Est - Ouest)	OUEST	LED Rouge	RW = RE
		LED Orange	JW = JE
		LED Verte	VW = VE

Nous n'aurons ainsi besoin que de 6 sorties du port parallèle pour commander l'allumage de nos LEDs.

Le Grafcet de notre système sera représenté comme ceci :



Dans la première étape, le feu sur la route principale est rouge et celle de la route secondaire au vert. La temporisation est enclenchée pendant 7 secondes pour rester dans cette configuration.

Une fois ces 7 secondes atteintes, on passe à la deuxième étape qui consiste à passer au feu orange sur la route secondaire pendant 3 secondes tout en laissant le feu au rouge sur l'autre route.

Après ces 3 secondes, on passe à l'étape 3 pour mettre le feu au vert dans la route principale et le feu au rouge sur la route secondaire. Cette étape a une durée de 10 secondes.

Une fois ces 10 secondes passés, on passe à la dernière étape qui consiste à passer le feu à l'orange dans la route principale tout en laissant le feu au rouge sur la route secondaire. Une fois les 3 secondes écoulées, on boucle jusqu'à l'étape principale pour pouvoir refaire ce même cycle indéfiniment.

En supposant la LED VW branchée sur la sortie D0 du port parallèle, JW branchée sur la sortie D1, RW sur la sortie D2, VN sur la sortie D3, JN sur la sortie D4 et RN sur la sortie D5 nous pouvons déjà représenter l'état des sorties du port parallèle en fonction des étapes données par le Grafcet avec les tableaux ci-dessous :

Etape 1 / Durée 7 secondes					
D5	D4	D3	D2	D1	D0
RN	JN	VN	RW	JW	VW
1	0	0	0	0	1

Etape 2 / Durée 3 secondes					
D5	D4	D3	D2	D1	D0
RN	JN	VN	RW	JW	VW
1	0	0	0	1	0

Etape 3 / Durée 10 secondes					
D5	D4	D3	D2	D1	D0
RN	JN	VN	RW	JW	VW
0	0	1	1	0	0

Etape 4 / Durée 3 secondes					
D5	D4	D3	D2	D1	D0
RN	JN	VN	RW	JW	VW
0	1	0	1	0	0

Ainsi nous avons les valeurs binaires des nombres à envoyer sur les sorties des ports parallèles pour commander nos LEDs :

Pour l'étape 1 :

00100001 en binaire = 33 en décimal

Pour l'étape 2 :

00100010 en binaire = 34 en décimal

Pour l'étape 3 :

00001100 en binaire = 12 en décimal

Pour l'étape 4 :

00010100 en binaire = 20 en décimal

Nous retenons ces valeurs que nous utiliserons dans le programme qui commandera les LEDs

4.4. Le programme de commande en Free Pascal

Notre programme a été conçu de façon à séparer les procédures d'affichage à l'écran des procédures de commande proprement dites. Nous décrirons plus en détail les procédures de commande, épargnant ainsi au lecteur du présent travail les longues lignes décrivant les procédures d'affichage à l'écran. Des lignes de commentaire sont introduites pour faciliter la compréhension du programme.

4.4.1. Description du programme

Au lancement du programme, un écran apparaît affichant à gauche de l'écran une maquette de carrefour et à droite les options de menu pour la simulation accessible par l'appui d'une touche de clavier correspondante au numéro affiché à gauche de l'option de menu le tout surmonté de l'inscription « MENU PRINCIPAL ».

L'option de menu 1 ou Automatique bascule le programme en mode commande automatique. Il lance le cycle que nous avons vu ci-dessus dans l'analyse que nous avons fait sur les étapes de fonctionnement du feu de carrefour. Pour sortir de ce mode, il suffit d'appuyer sur la touche « ESC » comme indiqué dans le mode.

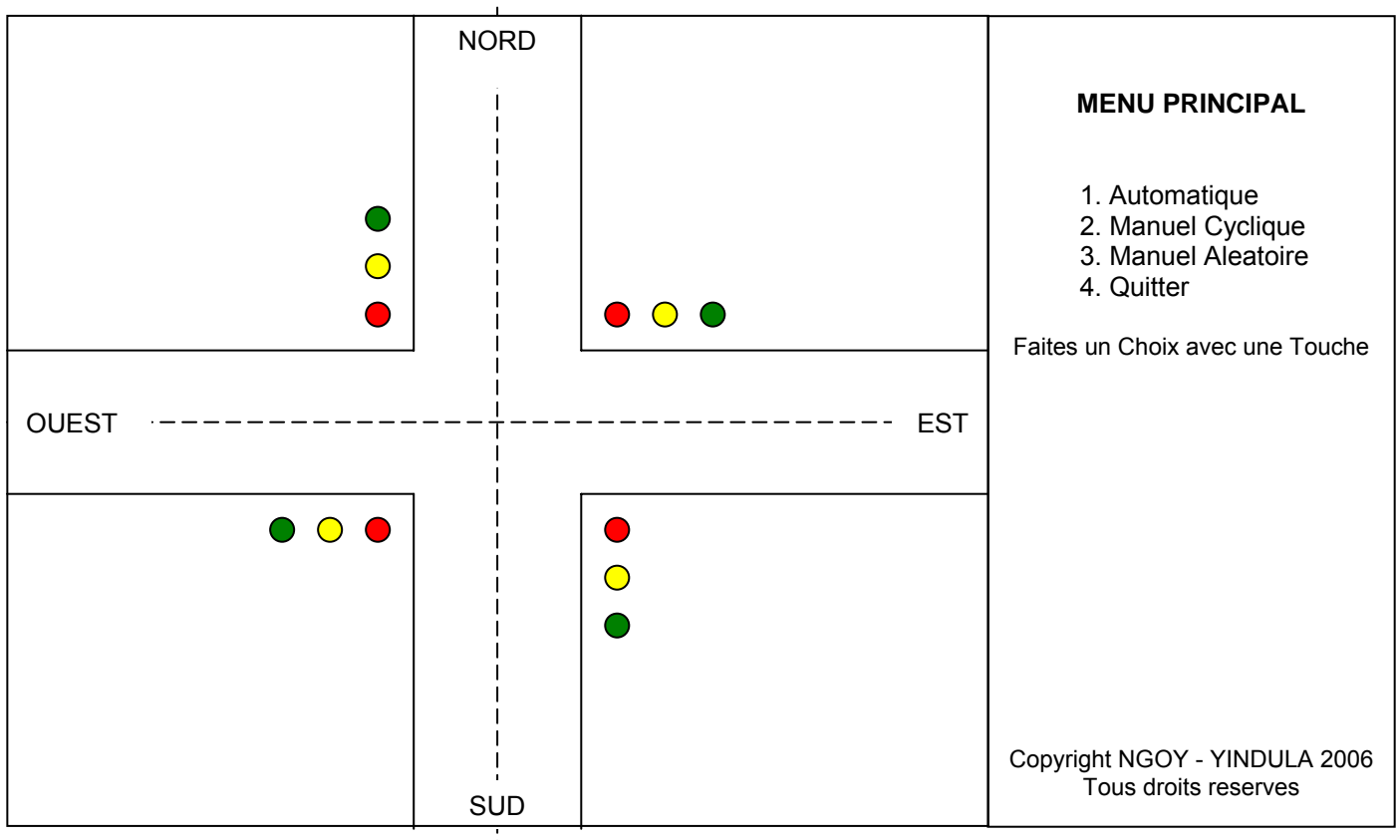
L'option de menu 2 ou Manuel Cyclique nous permet de changer les étapes du cycle manuellement grâce à la barre d'espacement. Ce mode change d'étapes à chaque appui sur la

touche barre d'espace et suit le cycle décrit par le Grafset. Comme pour le mode automatique, il suffit d'appuyer sur la touche « ESC » pour quitter le mode.

L'option de menu 3 ou Manuel quant à lui nous donne la possibilité d'afficher une étape individuellement sans passer nécessairement par l'étape précédente. Pour le quitter, il suffit d'appuyer sur la touche 5 comme indiqué dans le mode.

Dans la partie gauche de l'écran où se situe la maquette du carrefour, l'état des points représentant les LEDs évolue en fonction de l'affichage réel des LEDs. Quand les LEDs sont éteintes, les points sont gris.

La figure ci-dessous montre l'écran d'accueil du programme FeuxCarrefour.



4.4.2. Constitution du programme

4.4.2.1. Les procédures

Notre programme est constitué de 16 procédures qui sont :

1. AfficheEcranPrin : Elle affiche à l'écran le dessin de la maquette du carrefour ainsi que le cadre où seront placé les différents menus.
2. AfficheMenuPrin : Celle-ci affiche à l'écran le menu principal
3. AfficheModeAuto : Elle affiche à l'écran le menu du mode Automatique
4. AfficheModeManuelCycl : Affiche à l'écran le menu du mode Manuel Cyclique
5. AfficheModeManuelAlea : Affiche à l'écran le menu du mode Manuel Aléatoire

6. AfficheEtat01 : Change l'état des points du dessin correspondants aux LEDs à l'écran pour afficher l'étape 1
7. AfficheEtat02 : Change l'état des points du dessin correspondants aux LEDs à l'écran pour afficher l'étape 2
8. AfficheEtat03 : Change l'état des points du dessin correspondants aux LEDs à l'écran pour afficher l'étape 3
9. AfficheEtat04 : Change l'état des points du dessin correspondants aux LEDs à l'écran pour afficher l'étape 4
10. Etat01 : Routine pour la commande à effectuer à l'étape 1 du cycle. Nous le décrirons plus en détail dans la suite de ce chapitre.
11. Etat02 : Routine pour la commande à effectuer à l'étape 2 du cycle. Nous la décrirons plus en détail dans la suite de ce chapitre.
12. Etat03 : Routine pour la commande à effectuer à l'étape 3 du cycle. Nous la décrirons plus en détail dans la suite de ce chapitre.
13. Etat04 : Routine pour la commande à effectuer à l'étape 4 du cycle. Nous la décrirons plus en détail dans la suite de ce chapitre.
14. Automatique : Exécute les séquences nécessaires pour le mode Automatique
15. ManuelCycl : Exécute les séquences nécessaires pour le mode Manuel cyclique
16. ManuelAlea : Exécute les séquences nécessaires pour le mode Manuel aléatoire

4.4.2.2. Les routines Etat01, Etat02, Etat03 et Etat04

Le listing ci-dessous représente la routine Etat01. Toutes les autres routines de commande ont la même structure. Celles-ci représentent chacune des quatre étapes trouvées avec notre Grafcet. A la ligne 295, nous déclarons la procédure. Le bloc d'instructions la composant commence à partir de la ligne 296 avec l'instruction « Begin » et se clôture à la ligne 300 avec l'instruction « End » comme on l'a vu ci-dessus dans le chapitre présentant Free pascal.

L'instruction à la ligne 297 permet d'envoyer un octet à l'adresse situé à gauche dans la parenthèse, dans notre cas \$378 qui est l'adresse de base du port parallèle. La valeur de l'octet envoyé est située à droite dans la parenthèse. C'est là que se trouvera la différence entre les 4 routines, le nombre envoyé sur les registre data du port parallèle est différent selon l'état désiré. C'est ici que nous réutilisons les nombres trouvés dans l'analyse de fonctionnement de notre système.

Pour l'étape 1 du cycle, La procédure Etat01 met le chiffre binaire 00100001 dans le data register (\$378) soit 33 en décimal. L'instruction se trouve à la ligne 297 du listing. L'instruction à la ligne 298 permet d'actualiser l'affichage des points représentant les LEDs à l'écran dans la maquette du carrefour. Ici aussi nous observerons une différence en fonction des étapes.

A la ligne 299, nous vérifions si le programme est en mode « Automatique » avec la variable ModeAuto pour faire une temporisation de 7000 millisecondes soit 7 secondes obtenu avec l'instruction « Delay ». La temporisation variera selon les étapes du cycle aussi.

```

293 // Routine pour Premiere Etape du cycle
294
295 Procedure Etat01;
296 Begin
297   WritePortB($378,33);
298   AfficheEtat01;
299   If ModeAuto then Delay (7000);
300 End;

```

Après cette explication succincte des routines de commande, voici ci-dessous le programme complet de notre simulation, celui-ci contient 480 lignes de code, 7 variables et 16 procédures. Relevons ici l’instruction à la ligne 460. Sans celui-ci, nous ne pouvons pas faire d’entrée/sortie avec Linux.

```

001 Program FeuxCarrefour;
002
003 Uses Crt, x86 ;
004
005 // Definition des variables
006
007 Var
008   Finir, ManFinir, ModeAuto : Boolean;
009   ChoixMenu, CharControl : Char;
010   x, y : LongInt;
011
012 // Affichage Mode Automatique
013
014 Procedure AfficheModeAuto;
015
016 Begin
017   TextColor (LightRed);
018   Gotoxy (99, 7); Write('MODE AUTOMATIQUE');
019   Textcolor(Green);
020   Gotoxy (99, 10); Write(' ');
021   Gotoxy (99, 12); Write(' ');
022   Gotoxy (99, 14); Write(' ');
023   Gotoxy (99, 16); Write(' ');
024   TextColor (Yellow);
025   Gotoxy (92, 20); Write (' Appuyez ESC pour quitter le Mode');
026   Gotoxy (48, 47);
027 End;
028

```

```

029 // Affichage Mode Manuel cyclique
030
031 Procedure AfficheModeManuelCycl;
032

```

```

033 Begin
034   TextColor (LightRed);
035   Gotoxy (96, 7); Write('MODE MANUEL CYCLIQUE');
036   Textcolor(Green);
037   Gotoxy (99, 10); Write(' ');
038   Gotoxy (99, 12); Write(' ');
039   Gotoxy (99, 14); Write(' ');
040   Gotoxy (99, 16); Write(' ');
041   TextColor (Yellow);
042   Gotoxy (91, 20); Write ('Appuyez ESC pour quitter le Mode');
043   Gotoxy (91, 24); Write ('Tapez ESPACE pour changer Etats');
044   Gotoxy (48, 47);
045 End;
046
047 // Affichage Mode Manuel Aleatoire
048
049 Procedure AfficheModeManuelAlea;
050
051 Begin
052   TextColor (LightRed);
053   Gotoxy (96, 7); Write('MODE MANUEL ALEATOIRE');
054   Textcolor(Green);
055   Gotoxy (99, 10); Write('1. Etat 01 ');
056   Gotoxy (99, 12); Write('2. Etat 02 ');
057   Gotoxy (99, 14); Write('3. Etat 03 ');
058   Gotoxy (99, 16); Write('4. Etat 04');
059   Gotoxy (99, 18); Write('5. Quitter le Mode');
060   TextColor (Yellow);
061   Gotoxy (92, 20); Write ('Faites un Choix avec une Touche');
062   Gotoxy (48, 47);
063 End;
064
065
066 // Routine pour afficher le menu principal
067
068 Procedure AfficheMenuPrin;
069
070 Begin
071   TextColor (LightRed);
072   Gotoxy (100, 7); Write('MENU PRINCIPAL');
073   Textcolor(Green);
074   Gotoxy (99, 10); Write('1. Automatique');
075   Gotoxy (99, 12); Write('2. Manuel Cyclique');
076   Gotoxy (99, 14); Write('3. Manuel Aleatoire');
077   Gotoxy (99, 16); Write('4. Quitter');
078   TextColor (Yellow);
079   Gotoxy (92, 20); Write ('Faites un Choix avec une Touche');
080   Gotoxy (48, 47);
081 End;
082
083

```



```

084 // Routine pour afficher la fenetre principale
085
086 Procedure AfficheEcranPrin;
087
088 Begin
089     // Tracage Presentation Ecran
090     ClrScr;
091     TextColor (Blue);
092     For x:= 5 to 124 do
093         Begin
094             Gotoxy (x, 46); Write(Chr(205));
095             Gotoxy (x, 3); Write(Chr(205));
096         End;
097     For y:= 4 to (45) do
098         Begin
099             Gotoxy (4, y); Write(Chr(186));
100             Gotoxy (125, y); Write(Chr(186));
101             Gotoxy (88, y); Write(Chr(186));
102         End;
103     Gotoxy (4, 3); Write (Chr(201));
104     Gotoxy (125, 3); Write (Chr(187));
105     Gotoxy (4, 46); Write (Chr(200));
106     Gotoxy (125, 46); Write (Chr(188));
107     Gotoxy (88, 3); Write (Chr(203));
108     Gotoxy (88, 46); Write (Chr(202));
109
110     // Affichage Points Cardinaux
111     TextColor (Yellow);
112     Gotoxy (42, 4); Write ('NORD');
113     Gotoxy (43, 45); Write ('SUD');
114     Gotoxy (6, 24); Write ('OUEST');
115     Gotoxy (84, 24); Write ('EST');
116
117     // Affichage Copyright
118     TextColor (LightRed);
119     Gotoxy (92, 43); Write ('Copyright NGOY - YINDULA 2006');
120     Gotoxy (96, 44); Write ('Tous droits reserves');
121
122     // Tracage route EST - OUEST
123     TextColor (LightGray);
124     For x:= 5 to 87 do
125         Begin
126             Gotoxy (x, 21); Write(Chr(196));
127             Gotoxy (x, 27); Write(Chr(196));
128         End;
129     For x:= 37 to 50 do
130         Begin
131             Gotoxy (x, 21); Write(Chr(32));
132             Gotoxy (x, 27); Write(Chr(32));
133         End;
134

```

```

135 Gotoxy (13, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
136 Gotoxy (22, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
137 Gotoxy (31, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
138 Gotoxy (40, 24); Write(Chr(223) + Chr(223) + Chr(223));
139 Gotoxy (46, 24); Write(Chr(223) + Chr(223) + Chr(223));
140 Gotoxy (53, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
141 Gotoxy (62, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
142 Gotoxy (71, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
143 Gotoxy (79, 24); Write(Chr(223) + Chr(223) + Chr(223) + Chr(223));
144
145 // Tracage route NORD - SUD
146 For y:= 4 to 45 do
147     Begin
148         Gotoxy (37, y); Write (Chr(179));
149         Gotoxy (51, y); Write (Chr(179));
150     End;
151 For y:= 21 to 27 do
152     Begin
153         Gotoxy (37, y); Write(Chr(32));
154         Gotoxy (51, y); Write(Chr(32));
155     End;
156 Gotoxy (44, 6); Write (Chr(219));
157 Gotoxy (44, 7); Write (Chr(219));
158 Gotoxy (44, 11); Write (Chr(219));
159 Gotoxy (44, 12); Write (Chr(219));
160 Gotoxy (44, 16); Write (Chr(219));
161 Gotoxy (44, 17); Write (Chr(219));
162 Gotoxy (44, 21); Write (Chr(219));
163 Gotoxy (44, 22); Write (Chr(219));
164 Gotoxy (44, 26); Write (Chr(219));
165 Gotoxy (44, 27); Write (Chr(219));
166 Gotoxy (44, 31); Write (Chr(219));
167 Gotoxy (44, 32); Write (Chr(219));
168 Gotoxy (44, 36); Write (Chr(219));
169 Gotoxy (44, 37); Write (Chr(219));
170 Gotoxy (44, 41); Write (Chr(219));
171 Gotoxy (44, 42); Write (Chr(219));
172
173 // Tracage Carrefour
174 Gotoxy (37,21); Write (Chr(217));
175 Gotoxy (37,27); Write (Chr(191));
176 Gotoxy (51,21); Write (Chr(192));
177 Gotoxy (51,27); Write (Chr(218));
178
179 // Affichage LED eteintes NORD RJV
180 Gotoxy (35, 19); Write (Chr(220));
181 Gotoxy (35, 18); Write (Chr(220));
182 Gotoxy (35, 17); Write (Chr(220));
183

```

```

184 // Affichage LED eteintes SUD RJV
185 Gotoxy (53, 28); Write (Chr(220));
186 Gotoxy (53, 29); Write (Chr(220));
187 Gotoxy (53, 30); Write (Chr(220));
188
189 // Affichage LED eteintes EST RJV
190 Gotoxy (54, 20); Write (Chr(220));
191 Gotoxy (56, 20); Write (Chr(220));
192 Gotoxy (58, 20); Write (Chr(220));
193
194 // Affichage LED eteintes OUEST RJV
195 Gotoxy (34, 28); Write (Chr(220));
196 Gotoxy (32, 28); Write (Chr(220));
197 Gotoxy (30, 28); Write (Chr(220));
198 Gotoxy (48, 47);
199 End;
200
201 // Affichage Etape 1
202
203 Procedure AfficheEtat01;
204
205 Begin
206     TextColor (LightRed);
207     Gotoxy (35, 19); Write (Chr(220));
208     Gotoxy (53, 28); Write (Chr(220));
209     TextColor (LightGreen);
210     Gotoxy (30, 28); Write (Chr(220));
211     Gotoxy (58, 20); Write (Chr(220));
212     TextColor (LightGray);
213     Gotoxy (35, 18); Write (Chr(220));
214     Gotoxy (53, 29); Write (Chr(220));
215     Gotoxy (35, 17); Write (Chr(220));
216     Gotoxy (53, 30); Write (Chr(220));
217     Gotoxy (34, 28); Write (Chr(220));
218     Gotoxy (54, 20); Write (Chr(220));
219     Gotoxy (32, 28); Write (Chr(220));
220     Gotoxy (56, 20); Write (Chr(220));
221     Gotoxy (48, 47);
222 End;
223
224 // Affichage Etape 2
225
226 Procedure AfficheEtat02;
227
228 Begin
229     TextColor (LightRed);
230     Gotoxy (35, 19); Write (Chr(220));
231     Gotoxy (53, 28); Write (Chr(220));
232     TextColor (Yellow);

```

```

233     Gotoxy (32, 28); Write (Chr(220));
234     Gotoxy (56, 20); Write (Chr(220));
235     TextColor (LightGray);
236     Gotoxy (35, 18); Write (Chr(220));
237     Gotoxy (53, 29); Write (Chr(220));
238     Gotoxy (35, 17); Write (Chr(220));
239     Gotoxy (53, 30); Write (Chr(220));
240     Gotoxy (34, 28); Write (Chr(220));
241     Gotoxy (54, 20); Write (Chr(220));
242     Gotoxy (30, 28); Write (Chr(220));
243     Gotoxy (58, 20); Write (Chr(220));
244     Gotoxy (48, 47);
245 End;
246
247 // Affichage Etape 3
248
249 Procedure AfficheEtat03;
250
251 Begin
252     TextColor (LightRed);
253     Gotoxy (34, 28); Write (Chr(220));
254     Gotoxy (54, 20); Write (Chr(220));
255     TextColor (LightGreen);
256     Gotoxy (35, 17); Write (Chr(220));
257     Gotoxy (53, 30); Write (Chr(220));
258     TextColor (LightGray);
259     Gotoxy (35, 19); Write (Chr(220));
260     Gotoxy (53, 28); Write (Chr(220));
261     Gotoxy (35, 18); Write (Chr(220));
262     Gotoxy (53, 29); Write (Chr(220));
263     Gotoxy (32, 28); Write (Chr(220));
264     Gotoxy (56, 20); Write (Chr(220));
265     Gotoxy (30, 28); Write (Chr(220));
266     Gotoxy (58, 20); Write (Chr(220));
267     Gotoxy (48, 47);
268 End;
269
270 // Affichage Etape 4
271
272 Procedure AfficheEtat04;
273
274 Begin
275     TextColor (LightRed);
276     Gotoxy (34, 28); Write (Chr(220));
277     Gotoxy (54, 20); Write (Chr(220));
278     TextColor (Yellow);
279     Gotoxy (35, 18); Write (Chr(220));
280     Gotoxy (53, 29); Write (Chr(220));
281     TextColor (LightGray);
282     Gotoxy (35, 19); Write (Chr(220));

```

```

283     Gotoxy (53, 28); Write (Chr(220));
284     Gotoxy (35, 17); Write (Chr(220));
285     Gotoxy (53, 30); Write (Chr(220));
286     Gotoxy (32, 28); Write (Chr(220));
287     Gotoxy (56, 20); Write (Chr(220));
288     Gotoxy (30, 28); Write (Chr(220));
289     Gotoxy (58, 20); Write (Chr(220));
290     Gotoxy (48, 47);
291 End;
292
293 // Routine pour Premiere Etape du cycle
294
295 Procedure Etat01;
296 Begin
297     WritePortB($378,33);
298     AfficheEtat01;
299     If ModeAuto then Delay (7000);
300 End;
301
302 // Routine pour Seconde Etape du cycle
303
304 Procedure Etat02;
305 Begin
306     WritePortB($378,34);
307     AfficheEtat02;
308     If ModeAuto then Delay (3000);
309 End;
310
311 // Routine pour Troisieme Etape du cycle
312
313 Procedure Etat03;
314 Begin
315     WritePortB($378,12);
316     AfficheEtat03;
317     If ModeAuto then Delay (10000);
318 End;
319
320 // Routine pour Quatrieme Etape du cycle
321
322 Procedure Etat04;
323 Begin
324     WritePortB($378,20);
325     AfficheEtat04;
326     If ModeAuto then Delay (3000);
327 End;
328
329 // Routine d'Allumage Manuel Aleatoire
330

```

```

331 Procedure ManuelAlea;
332
333 Begin
334     ChoixMenu:= '0';
335     ModeAuto:= FALSE;
336     AfficheModeManuelAlea;
337     Repeat
338         ChoixMenu:= ReadKey;
339         Case ChoixMenu of
340             '1': Begin
341                 Etat01;
342             end;
343             '2': Begin
344                 Etat02;
345             end;
346             '3': Begin
347                 Etat03;
348             end;
349             '4': Begin
350                 Etat04;
351             end;
352         end;
353     Until ChoixMenu = '5';
354     WritePortB($378,0);
355 End;
356
357 // Routine d'Allumage Manuel Cyclique
358
359 Procedure ManuelCycl;
360
361 Begin
362     AfficheModeManuelCycl;
363     Finir := FALSE;
364     ManFinir:= FALSE;
365     ModeAuto:= FALSE;
366
367     Repeat
368         Finir := FALSE;
369         Repeat
370             Etat01;
371             CharControl:= ReadKey;
372             If CharControl = Chr(27) then Begin
373                 WritePortB($378,0);
374                 exit;
375             End;
376             If CharControl = Chr(32) then Finir := TRUE;
377         Until Finir;
378
379         Finir := FALSE;
380

```

```

381      Repeat
382          Etat02;
383          CharControl:= ReadKey;
384          If CharControl = Chr(27) then Begin
385              WritePortB($378,0);
386              exit;
387          End;
388          If CharControl = Chr(32) then Finir := TRUE;
389      Until Finir;
390
391      Finir := FALSE;
392
393      Repeat
394          Etat03;
395          CharControl:= ReadKey;
396          If CharControl = Chr(27) then Begin
397              WritePortB($378,0);
398              exit;
399          End;
400          If CharControl = Chr(32) then Finir := TRUE;
401      Until Finir;
402
403      Finir := FALSE;
404
405      Repeat
406          Etat04;
407          CharControl:= ReadKey;
408          If CharControl = Chr(27) then Begin
409              WritePortB($378,0);
410              exit;
411          End;
412          If CharControl = Chr(32) then Finir := TRUE;
413      Until Finir;
414
415  Until ManFinir;
416
417 End;
418
419 // Routine d'Allumage Automatique
420
421 Procedure Automatique;
422
423 Begin
424     AfficheModeAuto;
425     Finir := FALSE;
426     ModeAuto:= TRUE;
427
428     Repeat
429         Etat01;
430         If (KeyPressed and (ReadKey = chr(27))) then Begin

```

```

431         WritePortB($378,0);
432         exit;
433     End;
434
435     Etat02;
436     If (KeyPressed and (ReadKey = chr(27))) then Begin
437         WritePortB($378,0);
438         exit;
439     End;
440
441     Etat03;
442     If (KeyPressed and (ReadKey = chr(27))) then Begin
443         WritePortB($378,0);
444         exit;
445     End;
446
447     Etat04;
448     If (KeyPressed and (ReadKey = Chr(27))) then Begin
449         WritePortB($378,0);
450         exit;
451     End;
452     Until Finir;
453 End;
454
455
456 // Debut du Programme Principal
457
458 Begin
459     ChoixMenu:= '0';
460     fpIOperm($378,1,1);
461     Repeat
462         Clrscr;
463         AfficheEcranPrin;
464         AfficheMenuPrin;
465         ChoixMenu:= ReadKey;
466         Case ChoixMenu of
467             '1': Begin
468                 Automatique;
469             end;
470             '2': Begin
471                 ManuelCycl;
472             end;
473             '3': Begin
474                 ManuelAlea;
475             end;
476         end;
477     Until ChoixMenu = '4';
478     ClrScr;
479 End.
480 // Fin du programme principal

```


4.5. L'interface

Par expérimentation, nous avons constaté une faible luminosité quand les LEDs étaient directement branchés sur le port parallèle. Pour cette raison, nous avons conçu cette interface qui utilise le transistor en commutation pour amplifier les courants de sortie du port parallèle pour commander nos LEDs.

4.5.1. Les composants du circuit

Notre circuit comprendra :

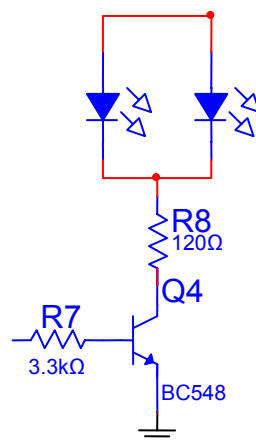
- 4 LEDs Rouges : Pour représenter les feux rouge.
- 4 LEDs Vertes : Pour représenter les feux vert
- 4 LEDs Oranges : Pour représenter les feux orange
- 6 transistors BC548 : Ce sont les éléments actifs du circuit, ils amplifient les signaux issus des sorties parallèle et commandent les LEDs qui sont branchés sur leurs Collecteurs. Ceux-ci nous permettent d'obtenir une meilleure luminosité de nos LEDs.
- 6 résistances de $3,3k\Omega$: Celles-ci limitent les courants de la base et protègent les sorties du port parallèle.
- 6 résistances de 120Ω : Ils limitent les courants fournis par le transistor aux LEDs et protègent celles-ci.
- 1 Condensateur Electrolytique de $470\mu F$: Nous avons jugé bon de le placer pour éviter des grandes variations de tension en cas de surcharge du port USB d'où nous tirons l'alimentation de notre circuit. Il fait office de filtrage.
- 1 Capacité de $100nF$: pour filtrer les fluctuations haute fréquence pouvant provenir du port USB.
- 1 Fusible à fusion rapide : pour protéger le Port USB en cas de court-circuit pendant la démonstration de la simulation.
- 1 Interrupteur : Pour la mise en marche de l'interface.
- 1 connecteurs DSUB25 Femelle : pour connecter l'interface au port parallèle du PC.
- 1 Connecteur USB : Celui-ci nous permettra de récupérer les 5 volts pour l'alimentation de l'interface à partir du port parallèle.
- 1 Plaque à pistes parallèles : C'est sur celle-ci que reposeront tous les composants.

Ces composants sont visibles sur la photo ci-dessous prise avant le montage :

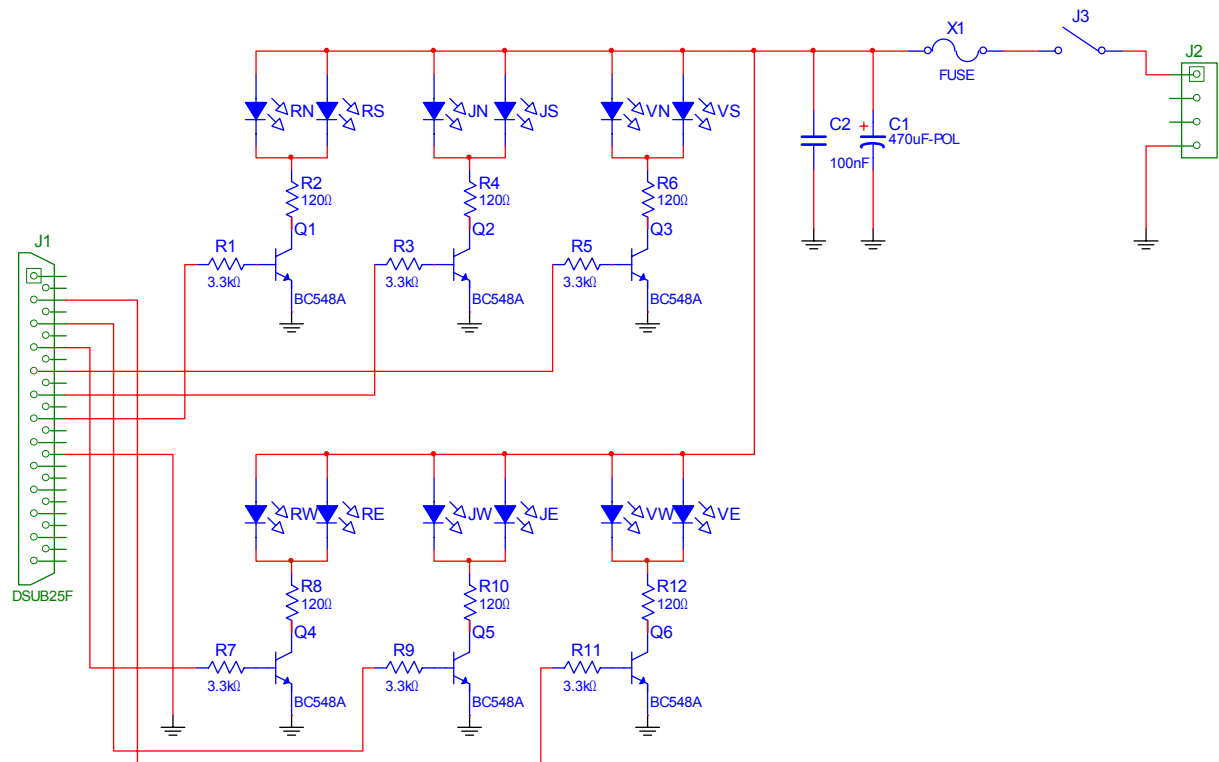


4.5.2. Le montage de l'interface

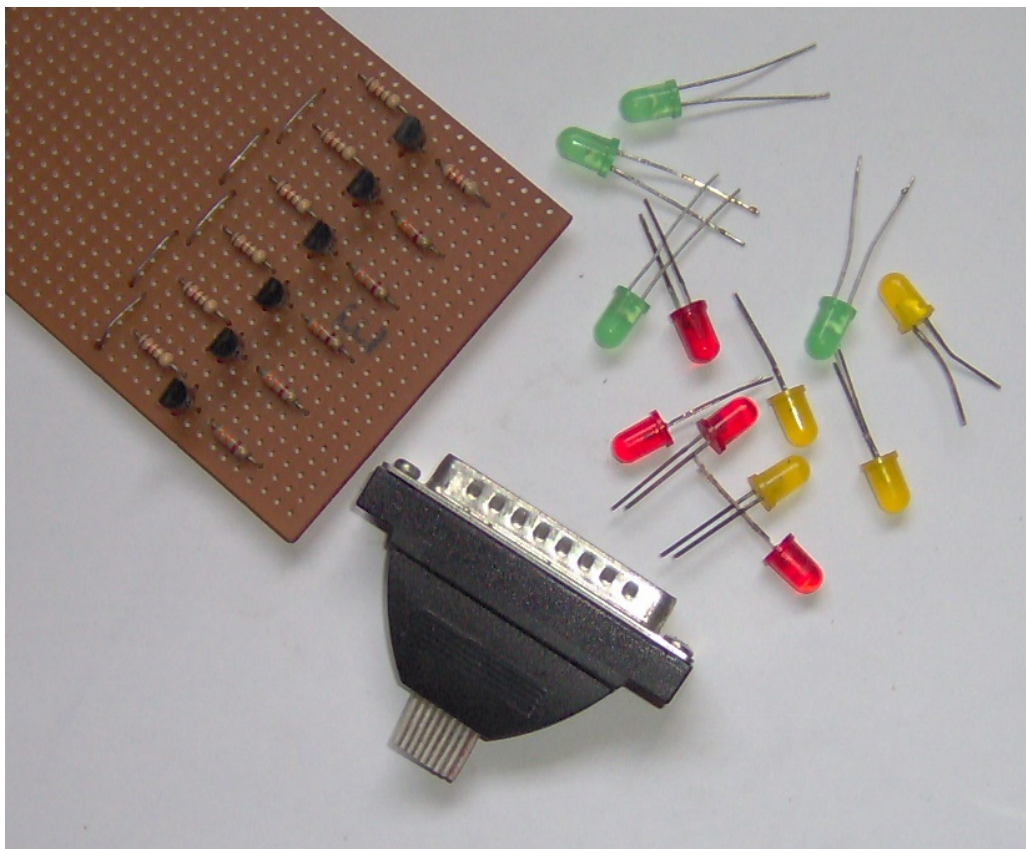
Chaque association de 2 LEDs est commandée par ce circuit de base :

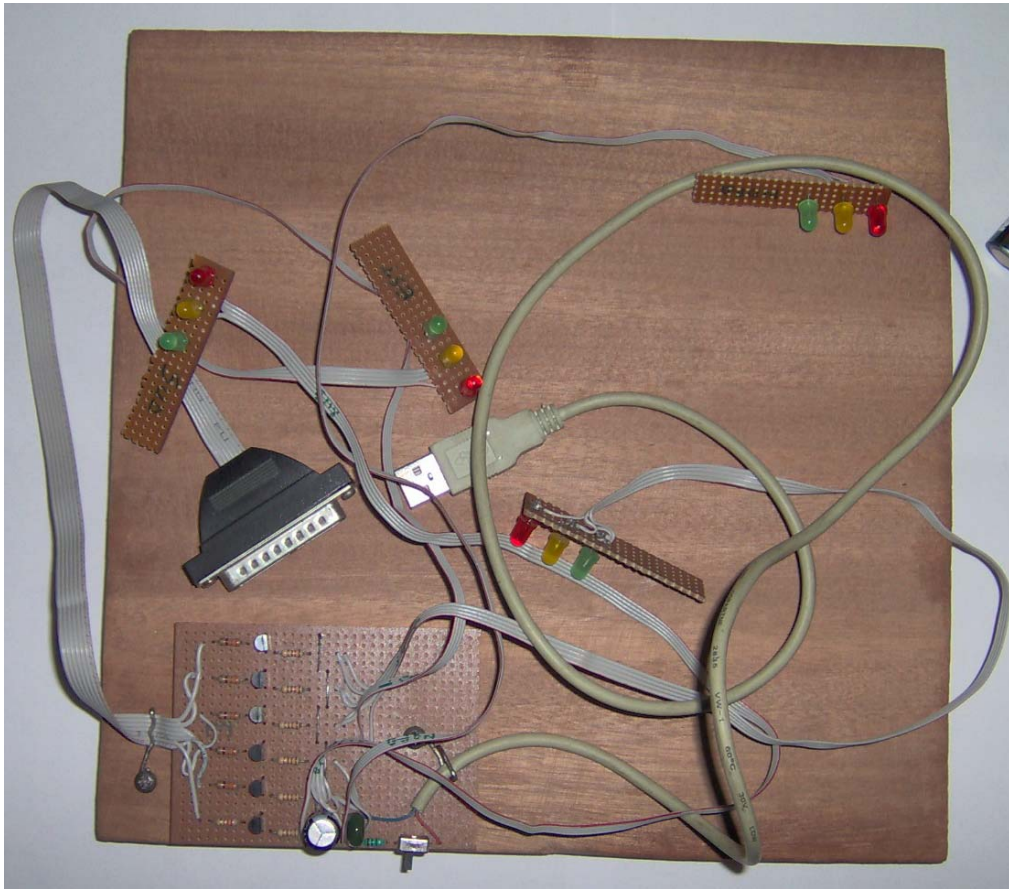
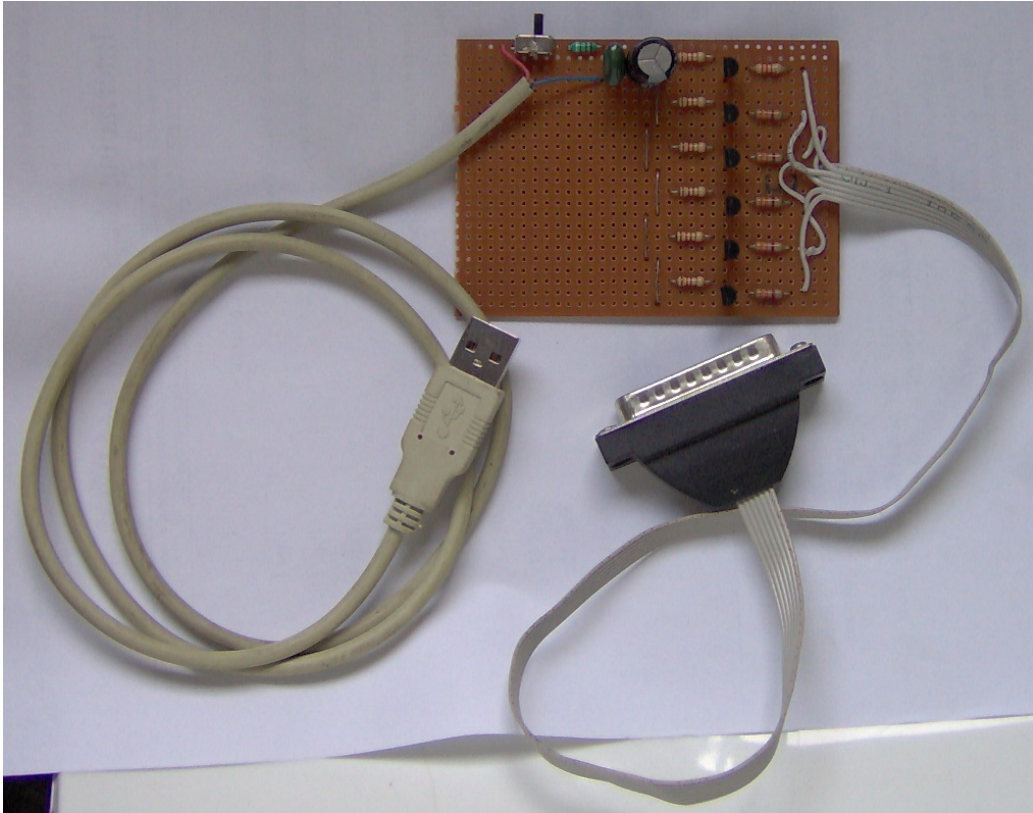


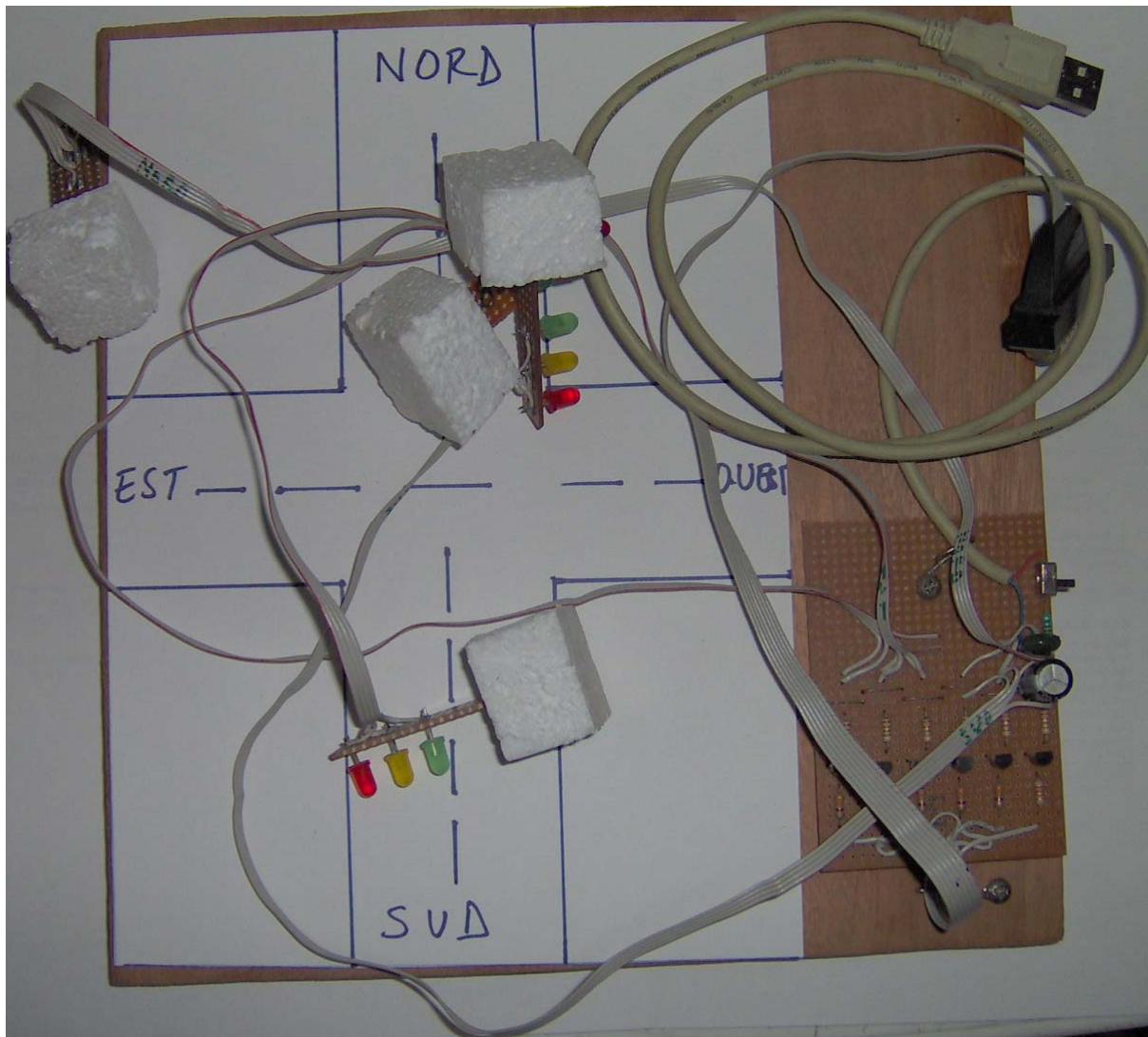
L'ensemble de ces circuits nous donne le schéma final de l'interface de commande qui est représentée sur la figure ci-dessous :



Les photos ci-dessous montrent les étapes progressives du montage de l'interface :







La maquette finalement terminée.

