

Électronique :

TP_Pic_C_Séance_2

Table des matières

TP_Pic_C_Séance_2.....	1
1. Structure sélective :.....	3
2. Structure sélective avec test appui :.....	4
2.2. Alternance en fonction des appuis :.....	5
3. Fonctions :.....	6
5. Structure sélective cas général :.....	7

1. Structure sélective :

Programme de chenillard :

2	<code>int led;</code>	initialisation variable : « led »
3	<code>void main()</code>	
4	<code>{</code>	<u>Tant que</u> vrais
5	<code>port_B_pullups(0xFF);</code>	Led = led numéro 1
6	<code>While (true)</code>	<u>tant que</u> led <= led numéro 5
7	<code>{</code>	allumer led
8	<code>led = 0x01;</code>	éteindre led
9	<code>while (led <= 0x20)</code>	passer de la led 1 à 2 de 2 à 3 [...]
10	<code>{</code>	<u>fin tant que</u>
11	<code>Output_b(led);</code>	<u>Tant que</u> led > led numéro 1
12	<code>delay_ms (500);</code>	allumer led
13	<code>Output_b(0x00);</code>	éteindre led
14	<code>delay_ms (0);</code>	passer de la led 5 à 4 de 4 à 3 [...]
15	<code>led=led*2;</code>	<u>fin tant que</u>
16	<code>}</code>	<u>Fin tant que</u>
17	<code>while (led > 0x01)</code>	
18	<code>{</code>	
19	<code>Output_b(led);</code>	
20	<code>delay_ms (500);</code>	
21	<code>Output_b(0x00);</code>	
22	<code>delay_ms (0);</code>	
23	<code>led=led/2;</code>	
24	<code>}</code>	
25	<code>}</code>	
26	<code>}</code>	

2. Structure sélective avec test appui :

Programme de chenillard qui marche si le bouton 1 du port A est actionné :

1	<code>#include <chenille.h></code>	initialisation variable : « led »
2	<code>int led;</code>	
3	<code>void main()</code>	
4	<code>{</code>	
5	<code>port_B_pullups(0xFF);</code>	<u>Tant que</u> vrais
6	<code>While (true)</code>	Led = led numéro 1
7	<code>{</code>	<u>SI</u> bouton du port A activé
8	<code>led = 0x01;</code>	<u>tant que</u> led <= led numéro 5
9	<code>If (input_A () == 0x01)</code>	
10	<code>{</code>	
11	<code>while (led <= 0x10) // led<5</code>	allumer led
12	<code>{</code>	éteindre led
13	<code>Output_b(led);</code>	
14	<code>delay_ms (250);</code>	passer de la led 1 à 2 de 2 à 3 [...]
15	<code>Output_b(0x00);</code>	
16	<code>delay_ms (0);</code>	<u>fin tant que</u>
17	<code>led=led*2;</code>	<u>Tant que</u> led > led numéro 1
18	<code>while (led > 0x01) // led>1</code>	allumer led
19	<code>{</code>	éteindre led
20	<code>Output_b(led);</code>	
21	<code>delay_ms (250);</code>	passer de la led 5 à 4 de 4 à 3 [...]
22	<code>Output_b(0x00);</code>	
23	<code>delay_ms (0);</code>	<u>fin tant que</u>
24	<code>led=led/2;</code>	<u>Sinon</u>
25	<code>}</code>	Allumer led (qui vaut led numéro 1)
26	<code>}</code>	<u>Fin tant que</u>
27	<code>}</code>	
28	<code>else</code>	
29	<code>{</code>	
30	<code>Output_b(led);</code>	
31	<code>}</code>	
32	<code>}</code>	
33	<code>}</code>	

Si on relâche le bouton poussoir, le programme continue jusqu'à qu'il finisse une exécution.

2.2. Alternance en fonction des appuis :

```

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```

```

if (input_A () == 0x00)
{
    led = 0x01;
    {
        Output_b(led);
    }
}

if (input_A () == 0x01)
{
    led = 0x01;
    while (led <= 0x20)
    {
        Output_b(led);
        delay_ms (250);
        Output_b(0x00);
        delay_ms (0);
        led=led*2;
    }
}

if (input_A () == 0x02)
{
    led = 0x20;
    while (led >= 0x01)
    {
        Output_b(led);
        delay_ms (250);
        Output_b(0x00);
        delay_ms (0);
        led=led/2;
    }
}

```

SI Pas d'appui : LB0 allumée

SI Appui PA0 : Chenillard de Droite à Gauche

SI Appui PA1 : Chenillard de gauche à Droite.

3. Fonctions :

```

Void sens_droit()
{
    led = 0x01;
    while (led <= 0x20)
    {
        Output_b(led);
        delay_ms (250);
        Output_b(0x00);
        delay_ms (0);
        led=led*2;
    }
}

Void sens_gauche()
{
    led = 0x20;
    while (led >= 0x01)
    {
        Output_b(led);
        delay_ms (250);
        Output_b(0x00);
        delay_ms (0);
        led=led/2;
    }
}

void main()
{
    port_B_pullups(0xFF);
    While (true)
    {
        if (input_A () == 0x00)
        {
            led = 0x01;
            {
                Output_b(led);
            }
        }
        if (input_A () == 0x01)
        {
            sens_droit ();
        }
        if (input_A () == 0x02)
        {
            sens_gauche();
        }
    }
}

```

Nous mettons maintenant nos « fonctions » dans des « void » au dessus du void main, la fonction principale

Fonction sens_droit

Fonction sens_gauche

Cette fonction principale en est donc extrêmement simplifiée .

← void sens_droit

← void sens_gauche

5. Structure sélective cas général :

```

23         delay_ms (250);
24         Output_b(0x00);
25         delay_ms (0);
26         led=led/2;
27     }
28 }
29
30
31 void main()
32 {
33     port_B_pullups(0xFF);
34     While (true)
35     {
36         lecture_A=input_A() & 0x03;// lecture clavier
37         switch (lecture_A)
38         {
39             case 0x00:led=0x01;
40             break;
41             case 0x01:sens_droit ();
42             break;
43             case 0x02:sens_gauche();
44         }
45     }
46 }
47

```

Nous avons ici utilisé un « switch » à la place de plusieurs « if »

Dans le switch nous mettons plusieurs cas.

Cas de si input_A=0
led 1 allumée

Cas de si input_A=1
Fonction sens droit

Cas de si input_A=2
Fonction sens gauche

Cette fonction est plus simple a utiliser que la fonction if car, elle ne demande pas de faire des boucles a chaque fois, et ne saturera moins la mémoire.

Pour information, nous avons aussi rajouter un masque, qui ne fais fonctionner le programme que avec la led 1 et 2 (& 0x03)