

Exercice N°1 : (5pts)

1- Donner le rôle du bouton RAZ

pour initialiser le programme

2- Donner le nom du composant X1

c'est un quartz (génère le signal d'horloge)

3- Donner à chaque fois la valeur du portB selon la direction de la voiture.

Avant : portB=0b...00111010... ; portB=0x...3A ←

Arrière : portB=0b...00110101... ; portB=0x...35

Gauche : portB=0b...10100011... ; portB=0x...A3

Droite : portB=0b...01010011... ; portB=0x...53

4- Donner la valeur du portA selon la direction de la voiture.

Avant : portA=0b...0001... ; portA=...1 /

Arrière : portA=0b...0010... ; portA=...2 /

Gauche : portA=0b...0100... ; portA=...4 /

Droite : portA=0b...1000... ; portA=...8 /

5- Compléter le programme de commande de la voiture

```
void main()
```

```
{
```

```
trisa=0xFF; configurer le portA en entrée. → 1111 1111
```

```
trisb=0x00; configurer le portB en sortie.
```

```
portb=0; //initialisation mettre à zéro les sorties
```

```
for(;;) // boucle infinie
```

```
switch(porta) // le port des entrées
```

```
{
```

```
case 1: portb=0x3A;break;
```

```
case 2: portb=0x35;break;
```

```
case 4: portb=0xA3;break;
```

```
case 8: portb=0x53;break;
```

```
default: portb=0x00;
```

```
} }
```



في دارك... انتهى على قرابتك إصغارك

Exercice N°2 : (4.5pts)

En se référant a la page 2/2 du dossier technique et a la figure N°2

1- Donner le nom des variables d'entrée avec le type et l'adresse

A, bit, RA0

B, bit, RA1

2- Compléter le programme de fonctionnement .

programme	Déclaration des variables
<code>sbit A at RA0_bit;</code>	
<code>Sbit B at RA1_bit;</code>	
<code>void main()// Note clé du programme.</code>	
<code>{ // début programme.</code>	
<code>TRISA=0xFF// configurer le portA en entrée.</code>	
<code>TRISB=0x00;</code>	
<code>PORTB=0; // initialisation.</code>	
<code>while (1) // boucle de répétition infinie.</code>	
<code>{</code>	
<code>if(A==0&B==0) portb=0xFF;</code>	
<code>else if(A==1 & B==0) {portb=0xFF;delay_ms(750);</code>	
<code>portB=0x00;delay_ms(750);} //</code>	
<code>else if(A==0 & B==1) {portb=0x55;delay_ms(500);</code>	
<code>portB=0xAA;delay_ms(500);} //</code>	
<code>else if(A==1 & B==1) {portb=0x3C;delay_ms(1000);</code>	
<code>portb=0xC3;delay_ms(1000);} //</code>	
<code>else portB=0; ←</code>	
<code>} // fin while 1</code>	
<code>} // fin prog</code>	



في دارك... إتهون على قرابتك إصفاك

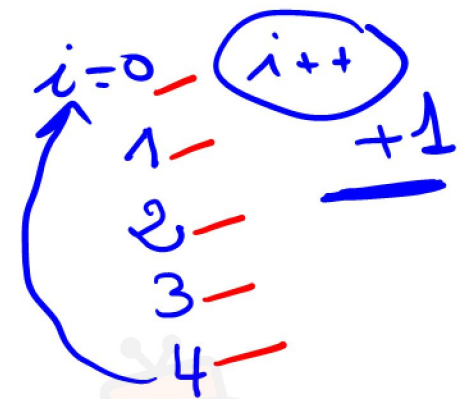
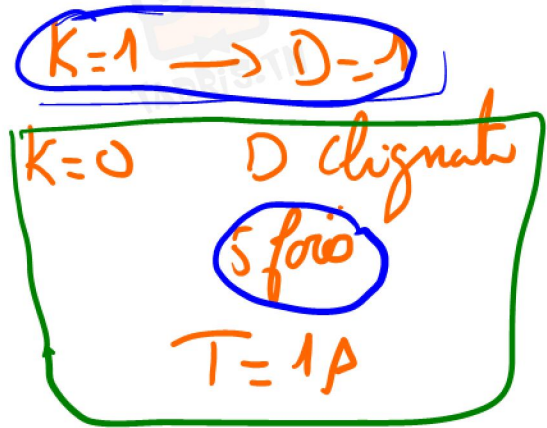
Exercice N°3 : (4.5pts)

Sur la commande de la voiture il y a une diode led qui indique l'état de la commande

Si on ferme sur l'interrupteur K D=1.

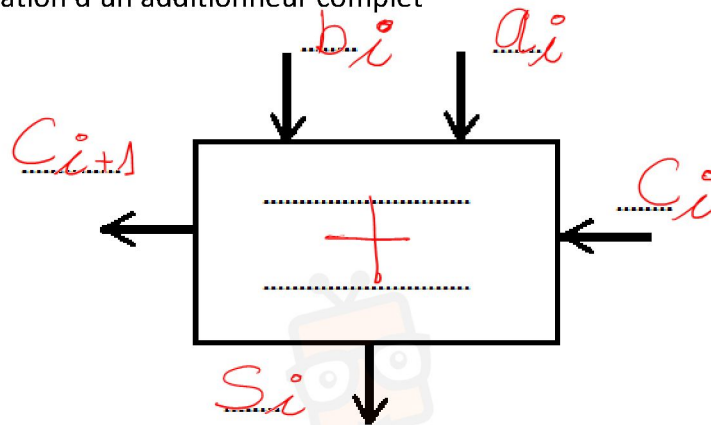
Si on ouvre K D clignote 5 fois puis s'éteint. T=1s

```
#define DRB0_bit #define K RA0_bit
char i;
void main()
{
  TRISA=0xFF; configurer le portA en entrée.
  TRISB=0x00; configurer le portB en sortie
  D=0; //initialisation ou bien portB=0;
  while(1)
  {
    do // faire
    {
      D=1;
    }
    while (K); // si K=1
    for(i=0; i<5; i++) //boucle repetition 5fois
    {
      D=1; delay_ms(500);
      D=0; delay_ms(500);
    }
    while (!K); // si K=0
  }
}
```



Exercice N°4 : (6pts)

1. Compléter la modélisation d'un additionneur complet

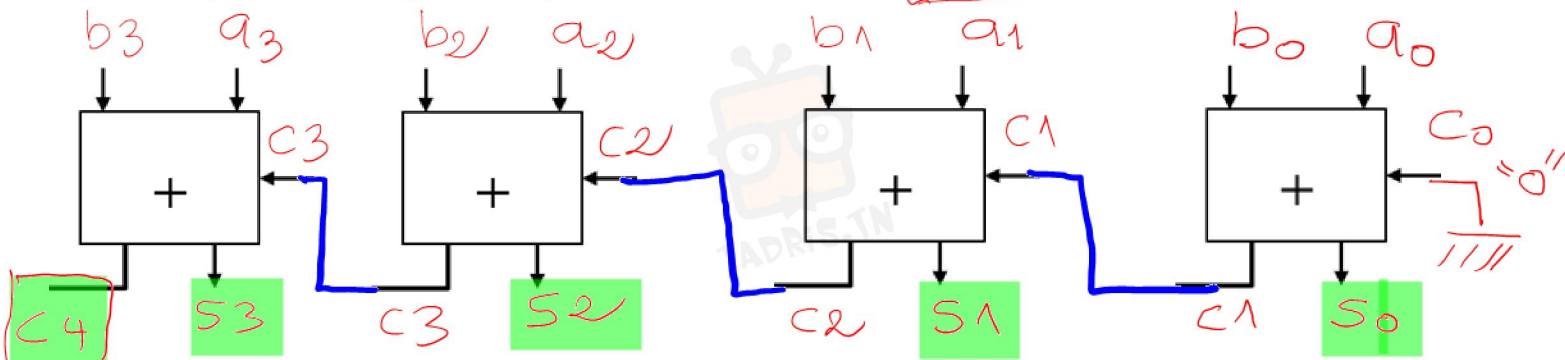


في دارك... ايتهاي علي قرابت اصفارك

2. Réaliser les opérations d'addition en binaire suivantes

$ \begin{array}{r} 10100011_{(2)} + 11001001_{(2)} \\ \begin{array}{r} 10100011 \\ + 11001001 \\ \hline 101101100 \end{array} \end{array} $	$ \begin{array}{r} 110100011_{(2)} + 00001001_{(2)} \\ \begin{array}{r} 110100011 \\ + 00001001 \\ \hline 110101100 \end{array} \end{array} $
---	---

3. Compléter le montage suivant pour avoir un additionneur a 4 bits. Nommer les entrées et les sorties.

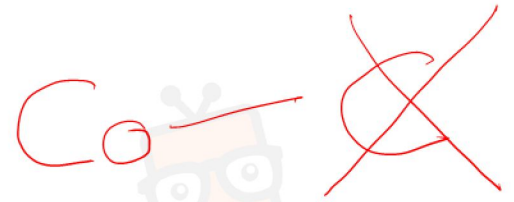


4. Compléter le programme en mikroC pour réaliser un additionneur a 4bits(voir figure N°3 page 2/2 dossier technique)

```

char S, A, B, Co;
void main()
{
    trisD = 0b11111111;
    trisC = 0b11111111;
    trisA = 0b00000000;
    trisE = 0b11111111;
    INTCON = 0x06;
    portA = 0; // initialisation
    while (1) // boucle infinie
    {
        A = portD; // établir A
        B = portC; // établir B
        Co = portE.B0; // établir Co
        S = A + B + Co; // calculer S
        PortA = S; // afficher S
    }
}

```



S = 0;

Co = portE.B0; // établir Co → Co = portE