	SUPPORT		CLASSE :
	Radar Pédagogique Icare-M		☑ Première ☑ Terminale
ACTIVITE	**Enseignement Technologique Transversal**	**Enseignement Technologique de Spécialité**	**Thème Sociétal**
☐ E.D.T. ☑ Activité pratique ☐ Projet	☐ Matière ☐ Energie ☐ Information	☑ S.I.N. ☐ E.E. ☐ I.T.E.C. ☐ A.C.	**Améliorer le confort d'un environnement**
Durée	**Centre(s) d'Intérêt(s) :**		
4 H	**CI3** Communication de l'information		
Objectif(s) :	O7 - Imaginer une solution, répondre à un besoin O8 - Valider des solutions techniques		
Compétences attendues :	CO7.sin1. Décoder la notice technique d'un système, vérifier la conformité du fonctionnement CO7.sin3. Exprimer le principe de fonctionnement d'un système à partir des diagrammes SysML pertinents. CO8.sin1. Rechercher et choisir une solution logicielle ou matérielle au regard de la définition d'un système		
Pré-requis :	- Numération - Algorithme - Langage de programmation arduino		

Données et conditions :

- Carte texte, afficheur à matrice de Leds et carte arduino.
- Rendre le TP à la fin de la séance.

Mise en situation :

Le système Radar Pédagogique Icare-M mesure la vitesse des véhicules et affiche des informations à l'aide d'un afficheur à matrice de Led.

Ce TP porte sur la fonction "afficher", correspondant au diagramme (ou analyse par SysML) spécifié dans le chapitre "approche fonctionnelle" du dossier technique. Il concerne l'analyse de la gestion de l'affichage par le microcontrôleur.

Problématique :

Comment afficher des symboles ou des lettres ?

1 Approche fonctionnelle du système « radar pédagogique »

Q1 : Le radar présent dans la salle est le radar ICARE-M. Le fichier pdf « plaquette_icare.pdf » présente le radar I-CARE . Quelle est la différence entre les deux radars en ce qui concerne l'affichage ?

_____

_____

Q2 : Compléter le diagramme des exigences du radar ICARE-M en annexe pour prendre en compte la différence du radar I-CARE.

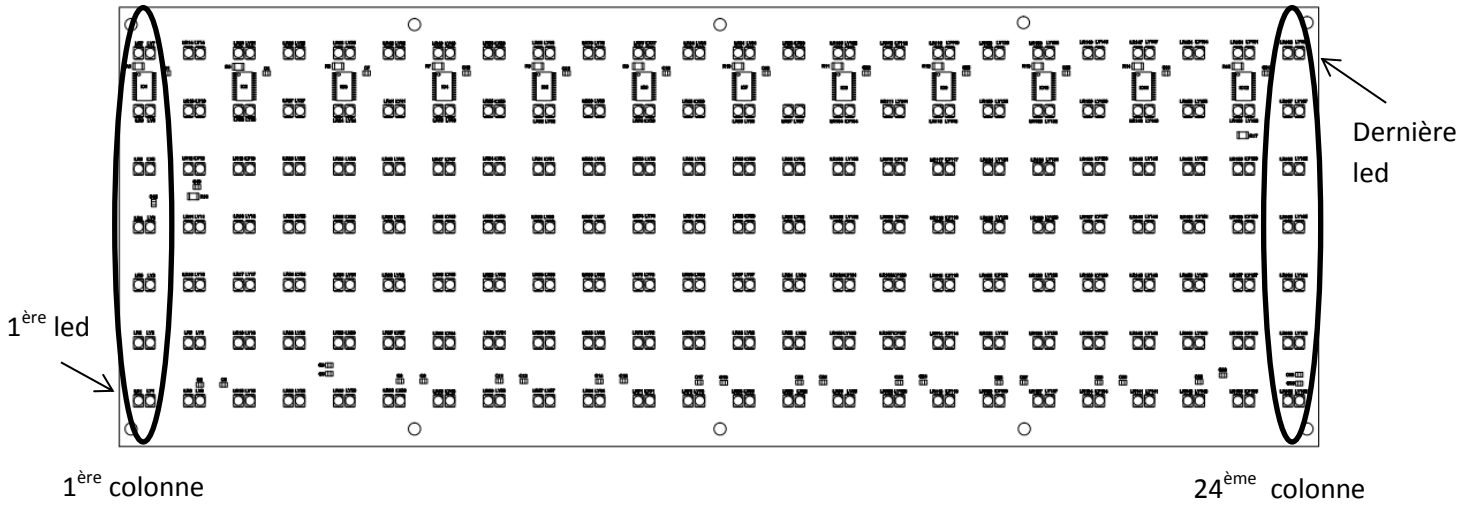
2 Affichage des caractères sur la matice de leds

Dans la suite de cette activité, vous comprendrez tout d'abord comment est gérée la matrice de leds supplémentaire du radar I-CARE. Ensuite, vous afficherez un caractère « P » puis un caractère ou un symbole de votre choix et pour finir un mot de 4 lettres.

2.1 Présentation

Les composants qui gèrent la matrice de leds sont constitués de registres à décalage mis en série, commandés par un signal d'horloge. Il y a donc une entrée série (SDI) et des sorties parrallèles.

La carte est constituée de 24 x 8 sorties parrallèles. Sur chaque groupe de 8 sorties, les 7 premières sorties sont reliées à une led, la 8ème sortie n'est pas reliée. Chaque groupe de 7 leds constitue une colonne sur la carte d'affichage. Il y a donc 24 colonnes au total.



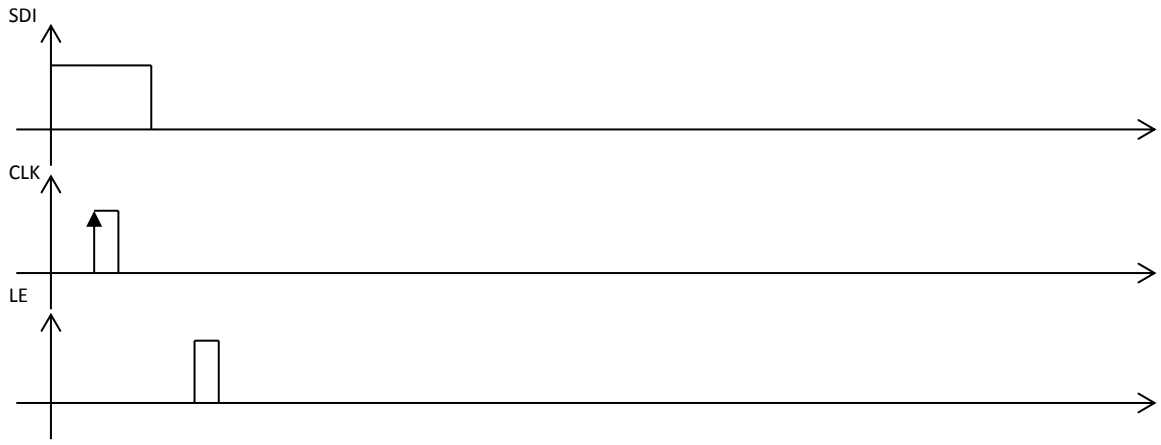
Trois sorties sont donc nécessaires pour pouvoir commander l'affichage à matrice de Leds.

- SDI : Permet de décider si une led va être allumée (SDI=1) ou etteinte (SDI=0)
- CLK : Horloge de commande du registre à décalage.
- LE : Latch enable, commande permettant de recopier le contenu du registre à décalage sur les sorties parrallèles du composant pour mettre à jour l'état des leds.

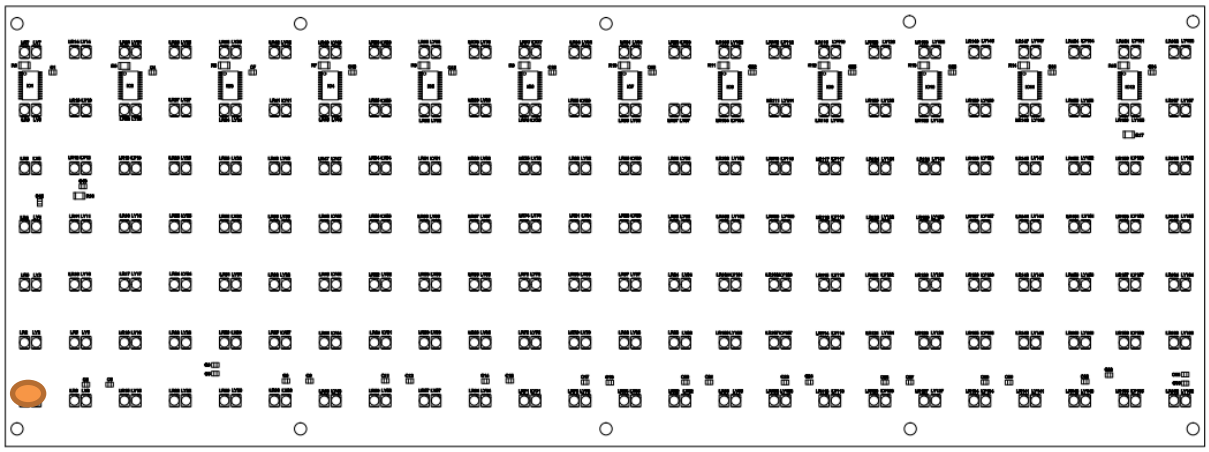
A chaque front montant de l'horloge CLK, l'état d'une led est transféré à la led suivante. L'état des 168 leds est mis à jour au même moment. Pourvisualiser le nouvel état de chaque led, il faut alors générer une impulsion sur le signal LE.

Exemple :

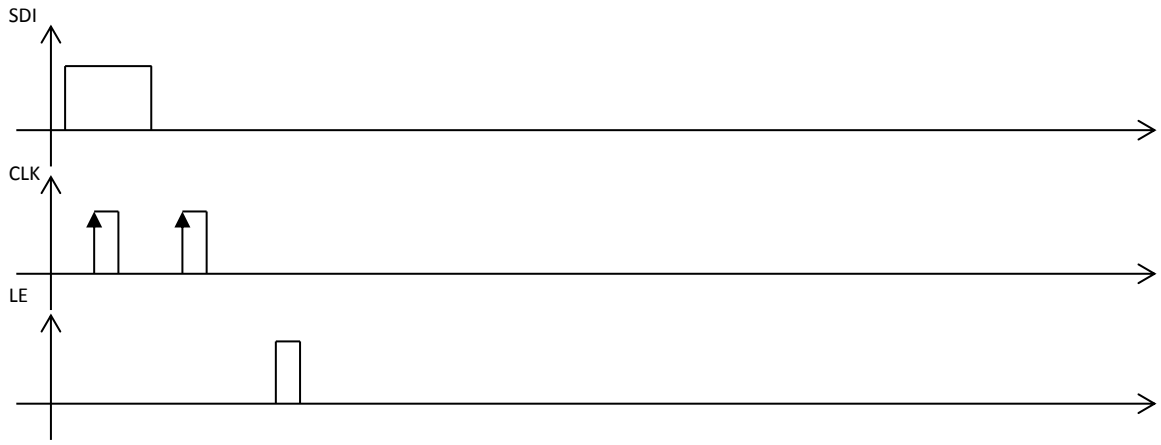
Etat initiale : toutes les led sont etteintes



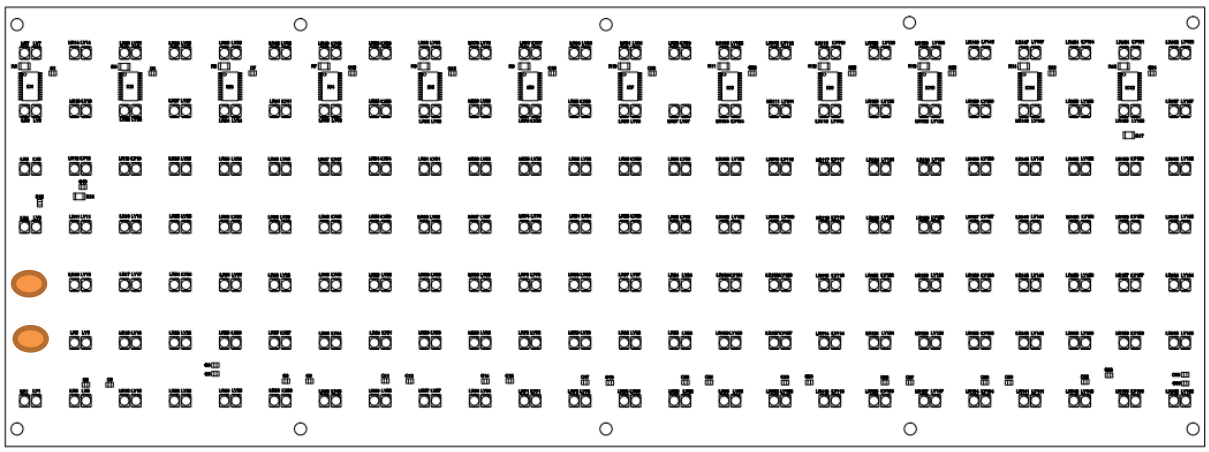
Résultat :



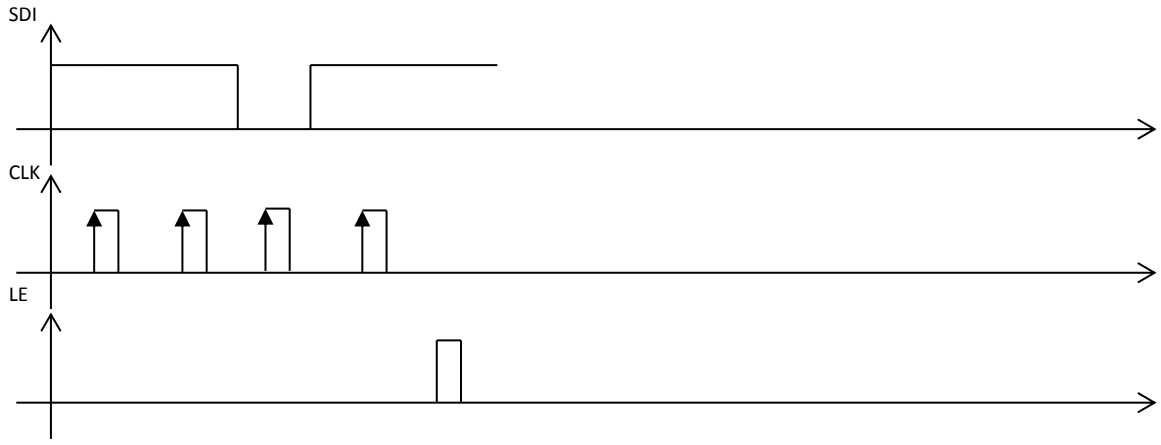
Ensuite en appliquant les signaux suivants :



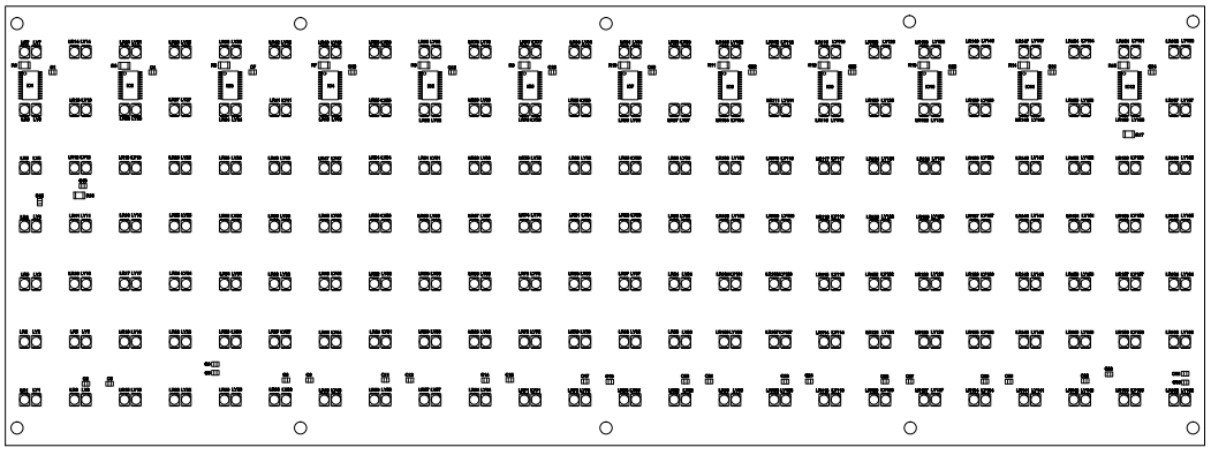
Résultat :



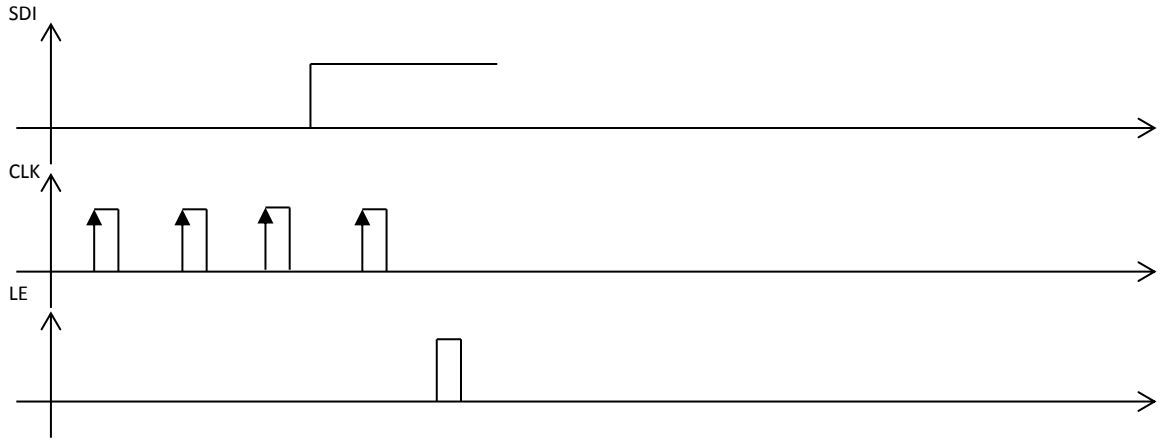
Les signaux suivants sont ensuite appliqués :



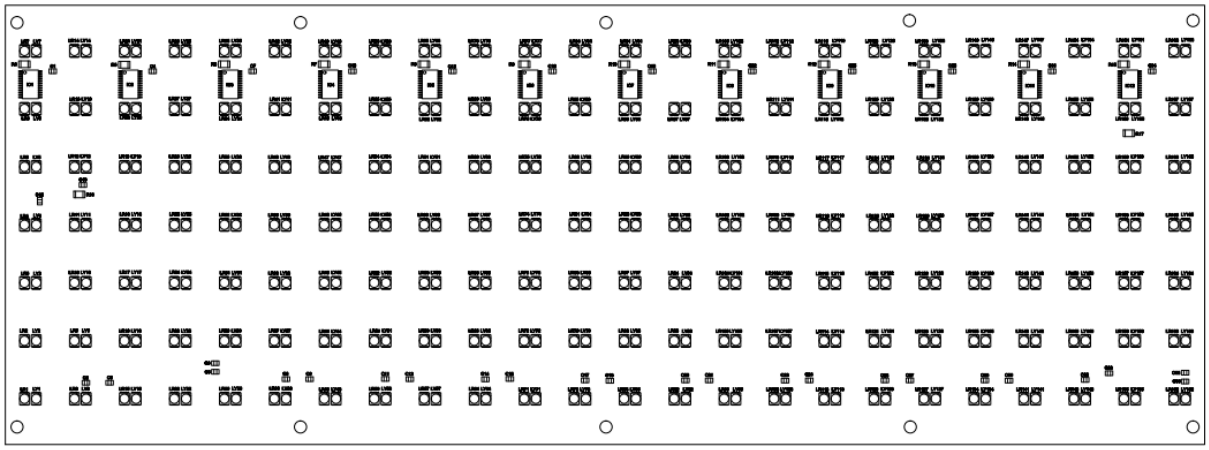
Q3 : Déterminez l'état des leds ci-dessous :



On continue avec les signaux suivants :



Q4 : Déterminez l'état des leds ci-dessous (Rappel : La huitième sorties n'est pas reliée à une led) :



2.2 Affichage du caractère « P ».

Cette matrice est prévue pour afficher 4 caractères, chaque caractère occupera donc une place de 6x7 leds. Exemple pour le caractère P (Rappel : 1 pour allumer, 0 pour éteindre) :

0	1	1	1	1	0
0	1	0	0	0	1
0	1	0	0	0	1
0	1	1	1	1	0
0	1	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	0

```
void loop() {
  digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,HIGH);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,HIGH);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
  digitalWrite(LE,HIGH);
  digitalWrite(LE,LOW);
  while(1);
}
```

Vous allez maintenant essayer d'afficher le caractère P en commençant par afficher une première colonne.

Q5 : Par quelle colonne devez-vous commencer ?

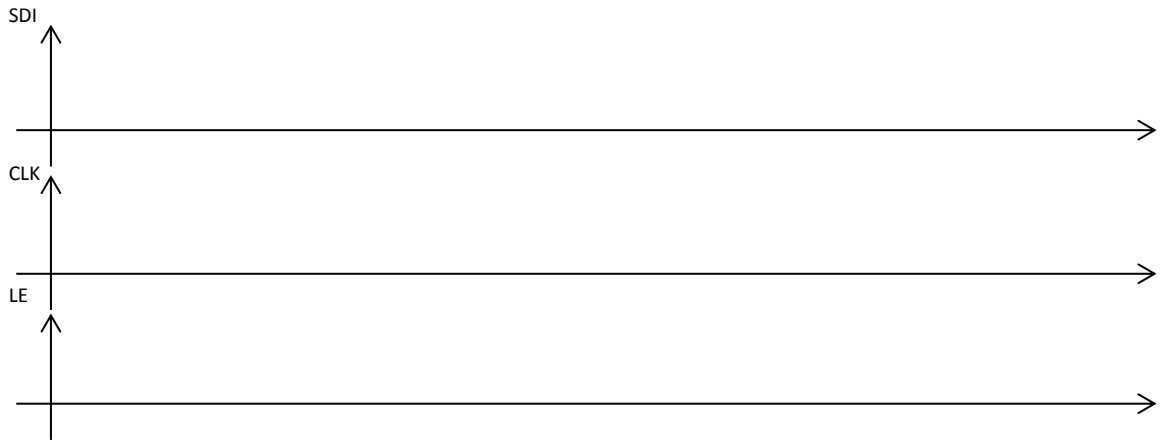
_____

Q6 : En analysant le code de la fonction loop ci-contre, complétez les chronogrammes page suivante.

Q7 : Complétez la fonction loop du sketch « affiche0.ino » avec le code ci-contre dans votre environnement « arduino ».Compiler et téléverser le programme dans le module arduino. Alimentez la carte fille Arduino en 12V.

Q8 :Le résultat est-il conforme à votre attente ? Pourquoi utilise-t-on huit coup d'horloge ?

Réponses Q6 :



Q9 : Complétez le programme de manière à afficher complètement le caractère P. Appeler le professeur lorsque vous aurez réussi.

2.3 Première optimisation du programme

Il est possible de prédéfinir le caractère dans un tableau :

```
const byte P[48]={
 1,1,0,0,1,1,1,1,
 1,0,1,1,0,1,1,1,
 1,0,1,1,0,1,1,1,
 1,0,1,1,0,1,1,1,
 1,0,0,0,0,0,0,0,
 1,1,1,1,1,1,1,1} ;
```

Le programme principal devient alors :

```
void loop() {
 char i;

 for(i=0;i<48;i++){
  if (P[i]==1)
   digitalWrite(SDI,HIGH);
  else
   digitalWrite(SDI,LOW);
  digitalWrite(CLK,HIGH);
  digitalWrite(CLK,LOW);
 }
 digitalWrite(LE,HIGH);
 digitalWrite(LE,LOW);
 while(1);
}
```

Q10 : Quelles valeurs prendra la variable i ?

_____

Q11 : Pour quels indices (valeurs de i) P[i] sera-t-il égale à 1 ?

_____

_____

Q12 : Chargez le programme « affiche3.ino » et exécutez le. Le caractère s'affiche-t-il correctement ?

_____

Q13 : Configurer le caractère de votre choix dans le tableau ci-dessous (Ne pas oublier que la première colonne est utilisée pour séparer deux caractères juxtaposés) :

Q14 : Mettez à jour le tableau dans le programme précédent et validez votre caractère. Appeler le professeur lorsque votre caractère s'affiche correctement.

2.4 Deuxième optimisation du programme

Il est inutile d'utiliser autant de place mémoire, on peut donc optimiser le tableau en utilisant les 8 bits de chaque octet du tableau. Chacun de ces octets correspondra donc à une colonne de votre caractère. Pour le caractère P, le nouveau tableau est le suivant :

```
const byte P[6]={ 0x30,0X48,0x48,0x48,0x7F,0x00} ;
```

Le programme principal devient alors :

```
void loop() {
 char Test,i,j;

 for(i=0;i<6;i++){
  for(j=0;j<8;j++) {
    Test=(P[i]>>(7-j))&0x01;
     if (Test==0x01)
       digitalWrite(SDI,HIGH);
     else
       digitalWrite(SDI,LOW);
    digitalWrite(CLK,HIGH);
    digitalWrite(CLK,LOW);
  }
 }
 digitalWrite(LE,HIGH);
 digitalWrite(LE,LOW);
 while(1);
}
```

La syntaxe (P[i]>>n) permet de faire un décalage à droite **n** fois du contenu de la variable P[**i**]. Pour chaque décalage, un 0 est placé sur le bit de poids fort.

Q15 : Remplissez les valeurs ci-dessous lorsque i=1 et j= 0 à 3 :

P[i] | | | | | | | | |

P[i]>>(7-0) | | | | | | | | |

P[i]>>(7-1) | | | | | | | | |

P[i]>>(7-2) | | | | | | | | |

P[i]>>(7-3) | | | | | | | | |

P[i]>>(7-4) | | | | | | | | |

Un « et » avec l'octet 0x01 est ensuite appliqué avec le résultat du décalage précédent.

Q16 : Complétez le tableau ci-dessous :

P[i]>>(7-4)	0	0	0	0	1	0	0	1
0x01	0	0	0	0	0	0	0	1
(P[i]>>3)&0x01								

Q17 : Quel sera le résultat final si au départ le bit de poid (7-j) de P[i] était égal à 1 ?

_____

Q18 : Quel sera le résultat final si au départ le bit de poid (7-j) de P[i] était égal à 0 ?

_____

Q19 : Expliquez alors quel est l'utilité du « if (Test==1) »

_____

_____

Q20 : Faire l'organigramme de la fonction « loop ».

Q21 : Chargez le programme « affiche4.ino », exécutez le. Le caractère s'affiche-t-il correctement ?

_____

Q22 : Mettez à jour le tableau de façon à faire apparaitre votre caractère de la question 13. Testez votre tableau en modifiant le programme « affiche4.ino »

```
const byte P[6]={                                                    } ;
```

3 Synthèse : comment afficher le mot « STOP » ?

Q23 : Combien de coup d'horloge sont nécessaires pour afficher directement 4 caractères ?

_____

Q24 : Proposez un tableau et modifiez la fonction « loop » de façon à afficher le mot STOP sur l'afficheur :

```
const byte P[  ]={
```

Diagramme des exigences :