## **TD** Logique combinatoire

# 1 Identification des produits par code barres EAN 1

L'identification et la tracabilité des produits est un des enjeux essentiels de la distribution et la logistique. Les codes barres représentent le moyen le plus courant et le plus efficace pour réaliser cette opération. Ils présentent les avantages d'être très peu coûteux (impression de quelques centimétres carrés directement sur l'emballage ou sur une étiquette), d'identifier un très grand nombre de produit, d'être très faciles à lire (quelque milli-secondes à l'aide d'un lecteur optique) et d'être fiables (contrôle des erreurs de lecture).



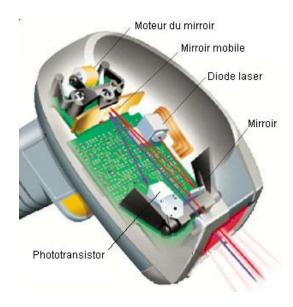


FIGURE 1 – Codes barres présents sur les produits de consommation courante

FIGURE 2 – Lecteur de code barres laser

Le standard EAN permet d'identifier de façon unique les produits fabriqués et/ou commercialisés partout dans le monde. La figure 1 montre une bouteille d'eau minérale où le code EAN est clairement visible.

La lecture se fait par un lecteur laser (un faisceau laser balaie une zone de lecture) ou par un lecteur CCD (une caméra CCD filme la zone de lecture). Lorsque le lecteur identifie une séquence correspondant à un code barres, le nombre binaire est enregistré et traduit en un identifiant décimal.

On s'intéresse dans cet étude à la traduction de l'identifiant en code barres et inversement.

#### 1.1 Composition du code barres EAN

Le code barres EAN est constitué de 13 chiffres (figure 3). On distingue un chiffre isolé à gauche : le préfixe No 1 (P1), une suite de 6 chiffres codés dans la partie gauche, et une seconde suite de 6 chiffres codés dans la partie droite.

La partie gauche contient un second préfixe No 2 (P2) et 5 chiffres identifiant la société distribuant l'article.

<sup>1.</sup> Pour plus d'informations, consulter le site internet http://www.gomaro.ch/

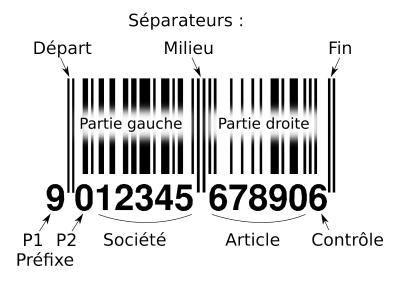


FIGURE 3 – Composition d'un code barres EAN.

La partie droite contient 5 chiffres identifiant l'article et un chiffre de contrôle. Le chiffre de contrôle permet de vérifier la lecture du code barres.

Le codage sous forme de barres verticales se fait en binaire : une barre élémentaire noir équivaut à un 1 logique et une barre élémentaire blanche équivaut à un 0 logique. Il faut donc lire la succession de barre comme un mot binaire de 95 bits.

Le mot binaire commence par une combinaison de début (D) codée par 101 (deux barres noires séparées par une blanche). Viens ensuite le codage de la partie gauche, puis une combinaison de milieu (M) codée par 01010, puis le codage de la partie droite. On termine par une combinaison de fin (F) codée par 101.

Le codage de la partie droite est constitué de 6 séquences (pour chacun des 6 chiffres) de 7 bits. Chaque séquence commence par 1, se termine par 0 et code le chiffre à l'aide d'un code à 5 bits. Le tableau 1 de gauche indique le codage binaire associé à chaque chiffre.

### 1.2 Décodage du code barres EAN

Le chiffre 1 est donné en exemple sur le document réponse figure 9. On distingue une barre épaisse noire correspondant à deux barres élémentaires, suivie d'une barre épaisse blanche, puis noire, puis une barre blanche d'épaisseur élémentaire. En effet, on trouve dans la table de codage pour le chiffre 1 en partie droite 1100110.

**Q.1.** Déterminer pour la partie droite les 4 chiffres suivants et compléter le document réponse figure 9.

La partie gauche est un peu plus complexe. Historiquement, le préfixe No 1 a été introduit plus tard, pour étendre les possibilités du code. Pour des raisons de compatibilité avec les codes déjà existants, la forme du code barre n'a pas été modifiée.

Le préfixe No 1 est codé à partir de la parité du codage des 5 chiffres identifiant la société. En effet, on utilise deux types de codages : impaire (si le nombre de 1 est impaire) et paire (si le nombre de 1 est paire). Le second tableau indique le chiffre de préfixe No 1 (P1) en fonction de la parité des codages de la partie gauche.

**Q.2.** Déterminer la parité de chacune des séquences correspondant aux chiffres de la partie gauche. En déduire le chiffre du préfixe P1.

Partie gauche	Partie gauche	Partie droite
Impaire (I)	Pair (P)	
0001101	0100111	1110010
0011001	0110011	1100110
0010011	0011011	1101100
0111101	0100001	1000010
0100011	0011101	1011100
0110001	0111001	1001110
0101111	0000101	1010000
0111011	0010001	1000100
0110111	0001001	1001000
0001011	0010111	1110100
	Impaire (I)  0001101  0011001  0010011  0111101  0100011  0101111  0111011	Impaire (I)         Pair (P)           0001101         0100111           0011001         0110011           0010011         0011011           0111101         0100001           0100011         0011101           0110001         0111001           0101111         0000101           0110111         0001001

P1	P2	1	2	3	4	5
0	I	I	I	I	Ι	Ι
1	I	I	P	I	P	P
2	I	I	P	P	I	P
3	I	I	P	P	P	I
4	I	P	I	I	P	P
5	I	P	P	I	I	P
6	I	P	P	P	I	I
7	I	P	I	P	I	P
8	I	P	I	P	P	I
9	I	P	P	I	P	I

TABLE 1 – Tables de codage du code EAN.

Le tableau de gauche indique, en fonction de la parité, le codage de chaque chiffre pour la partie gauche.

**Q.3.** Déterminer les 6 chiffres de la partie gauche et compléter le document réponse figure 9. Le chiffre de contrôle permet de vérifier la lecture du code barres. Le calcul de ce chiffre se fait par une formule magique à partir des 12 premiers chiffres du code barres.

Notons  $n_1, n_2 \dots n_{13}$  les 13 chiffres du code barres ( $n_{13}$  étant le chiffre de contrôle) :

$$\alpha = n_1 + n_2 \times 3 + n_3 + n_4 \times 3 + n_5 + n_6 \times 3 + n_7 + n_8 \times 3 + n_9 + n_{10} \times 3 + n_{11} + n_{12} \times 3$$

Il faut ensuite diviser  $\alpha$  par 10 et déterminer le reste  $\beta$  de la division. Le chiffre de contrôle est alors  $n_{13} = 10 - \beta$ . (Ainsi, en ajoutant  $n_{13}$  à  $\alpha$ , le reste de la division par 10 doit être nul).

- **Q.4.** Vérifier sur l'exemple de la figure 3 le calcul du chiffre de contrôle. Déterminer ensuite le chiffre de contrôle du code barres du document réponse 9.
- **Q.5.** Dessiner clairement sur votre copie la séquence du code barre correspondant au chiffre de contrôle.

## 2 Driver de pont d'un hacheur à deux quadrants

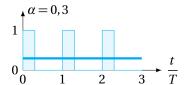
L'alimentation en énergie d'un moteur électrique à courant continu se fait grâce à l'utilisation d'un élément appelé hacheur (*electric chopper* en anglais).

Le hacheur assure une distribution variable de tension sous la forme d'une « modulation de la largeur d'impulsion » (MLI, aussi appelée PWM comme *pulse width modulation* en anglais <sup>2</sup>) selon les ordres envoyés sous forme logique par un micro-contrôleur.

La gestion du sens de rotation d'un moteur électrique en fonction de la position des interrupteurs du hacheur est précisée sur la figure 10. Les éléments A, B, C et D sont des transistors à effet de champs (FET : *Field Effect Transistor*) qui sont assimilables à des interrupteurs commandés par les tensions  $V_A$ ,  $V_B$ ,  $V_C$  et  $V_D$ .

Comme le montre la figure 10, le moteur tourne dans un sens si les interrupteurs A et D sont ouverts et les interrupteurs B et C sont fermés. Il tourne dans le sens opposé si les interrupteurs A et D sont fermés et les interrupteurs B et C sont ouverts. Cette structure à quatre interrupteurs est appelée « pont en H » de par sa structure.

<sup>2.</sup> Le terme anglais est précisé car c'est celui qui apparaît sur toutes les cartes électroniques de commande ainsi que sur tous les documents techniques, communément appelés *datasheets*.



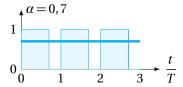


FIGURE 4 – Principe du hachage de courant par MLI/PWM et rapport cyclique  $\alpha = \frac{\text{Temps ferm\'e}}{\text{P\'eriode }T}$  : en ordonnée, le signal d'ordre de fermeture des hacheurs et en gras la valeur moyenne.

Si les interrupteurs situés en diagonale (donc A - D ou B - C selon le sens choisi) sont commutés en restant, sur une période T donnée (T est de l'ordre de la milliseconde), fermé pendant  $\alpha T$  secondes et ouvert pendant  $(1-\alpha)T$  secondes avec  $0 \le \alpha \le 1$  le « rapport cyclique », le moteur recevra une valeur moyenne fonction de  $\alpha$  et de la tension d'alimentation (voir figure 4). Le hacheur L293D / SN754410NE est composé de deux parties (voir figure 5) :

- La partie « driver », étudiée dans la suite, convertit l'information logique envoyée par un micro-contrôleur en un ordre d'ouverture/fermeture des interrupteurs symétriques A D ou B C selon le sens de rotation souhaité.
- La partie « interface de puissance » comporte les interrupteurs et des diodes de protection pour la transmission de l'alimentation de puissance (notée VCC) aux bornes du moteur via les interrupteurs.

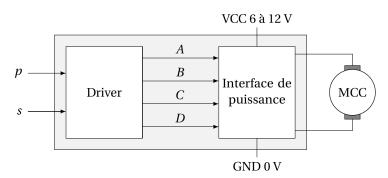


FIGURE 5 – Synoptique général de la structure d'un hacheur avec le driver, l'interface de puissance et un moteur électrique à courant continu alimenté.

Le circuit logique de pilotage étudié comporte donc deux entrées et quatre sorties logiques :

- Les deux entrées correspondent à l'ordre PWM (entrée p) et au signe (entrée s) qui vaudra 0 ou 1 selon le sens de rotation souhaité.
- Les quatre sorties binaires *A*, *B*, *C* et *C* correspondent aux commandes d'ouverture (valeur logique 0) ou de fermeture (valeur logique 1) des interrupteurs *A*, *B*, *C* et *D* du pont ≪ en H ≫ du hacheur.

Le cahier des charges du fonctionnement d'un tel « driver » de hacheur est le suivant :

- Si s = 0, le moteur tourne dans le sens trigonométrique, si s = 1, il tourne dans le sens anti-trigonométrique (horaire) : voir figure 10.
- Par construction, les interrupteurs situés sur les mêmes lignes d'alimentation, donc A B à gauche et C D à droite, ne peuvent être fermés simultanément.
- Si p = 0, l'alimentation du moteur est coupée et ses bornes sont reliées à la masse (GND).

S	p	A	В	C	D
0	0				
0	1				
1	0				
1	1				

TABLE 2 – Table de vérité du fonctionnement logique du pilote du hacheur.

- **Q.6.** Compléter la table de vérité fournie sur la table 2.
- **Q.7.** En déduire les expressions logiques A(p,s), B(p,s), C(p,s) et D(p,s).
- **Q.8.** Tracer les logigrammes de ces quatre fonctions logiques.
- **Q.9.** Compléter le chronogramme de la figure 11.

## 3 Traitement des signaux d'un codeur incrémental

Un codeur incrémental permet de mesurer une position ou une vitesse à partir de signaux numériques de type créneaux. Ces signaux sont généralement générés par un disque strié associé à une fourche optique, ou par un aimant associé à des capteurs à effet Hall (figure 6). Deux voies « en quadrature » sont transmises au microcontrôleur, lui permettant de compter ou décompter les fronts montant et/ou descendant des deux voies. Le sens de rotation se déduit du déphasage entre les voies.



FIGURE 6 – Codeur incrémental magnétique en bout d'arbre moteur

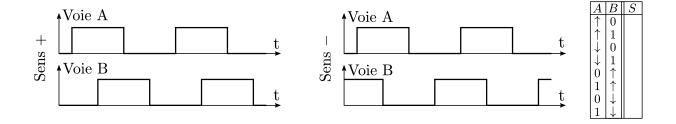


FIGURE 7 – Signaux des deux voies A et B selon le sens de rotation

Le traitement des signaux peut être matériel ou logiciel. Le traitement est matériel lorsque le microcontroleur est équipé d'une interface QEI (quadrature encoder interface) : des portes logiques s'occupent du traitement et le processeur n'a plus qu'à consulter le résultat dans un

registre dédié à l'interface. Lorsque le microcontroleur ne dispose pas d'interface QEI, il reste possible de réaliser un traitement logiciel en activant des interruptions sur changement d'état des deux broches reliées au voies de mesure, et en traitant le comptage/décomptage dans le programme. Le traitement matériel est évidement plus rapide (et permet des vitesses plus élevées) et libère le temps de calcul du microcontrôleur.

La figure 7 montre les signaux issus du capteur pour les deux sens de rotation.

**Q.10.** Compléter la table de vérité figure 7 en précisant un symbole + lorsque le compteur doit être incrémenté et un symbole – lorsqu'il doit être décrémenté.

Le traitement logiciel est implémenté sur un ATMEL ATmega 328 (microcontrôleur de l'Arduino). Ce microcontrôleur dispose de deux broches (2 et 3) permettant de générer des interruptions (respectivement numérotées 0 et 1) à chaque changement d'état de la broche (front montant ou descendant). Chaque interruption lance une fonction (respectivement interruptA et interruptB) assurant le comptage ou décomptage d'une variable de position pos. Le programme figure 8 montre le traitement réalisé.

```
int pos = 0; // Variable contenant la position actuelle
void setup() {
  pinMode(2, INPUT);  // Codeur voie A
pinMode(3, INPUT);  // Codeur voie B
  attachInterrupt(0,interruptA,CHANGE); // Interruption 0
  attachInterrupt(1,interruptB,CHANGE); // Interruption 1
  int A=digitalRead(2); // int 0...
int B=digitalRead(2); // Lecture voie A
void interruptA() {
  int B=digitalRead(3); // Lecture voie B
  TEST_A = ???
  if (TEST_A ){ // Test si rotation positive
  } else {
                            // Sinon rotation négative
   pos--;
  int A=digitalRead(2);  // Lecture voie A
int R=digitalRead(2)
void interruptB() {
  int B=digitalRead(3); // Lecture voie B
  TEST B = ???
  if (TEST_B ){ // Test si rotation positive
    pos++:
                             // Sinon rotation négative
  } else {
    pos--;
}
```

FIGURE 8 – Programme de traitement logiciel du codeur incrémental.

- **Q.11.** Indiquer quelles sont les actions réalisées dans la fonction set up ().
- **Q.12.** À quoi servent les deux premières lignes de la fonction interruptA()? Que vaudront les variables A et B suite à un front montant de la voie A dans le sens de rotation positif?
- **Q.13.** Déterminer les expressions des conditions booléennes TEST\_A et TEST\_B permettant de compter et décompter correctement la variable de position pos.

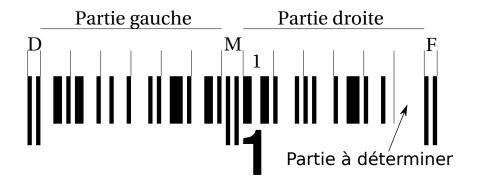


FIGURE 9 – Document réponse – Code barres EAN à décoder

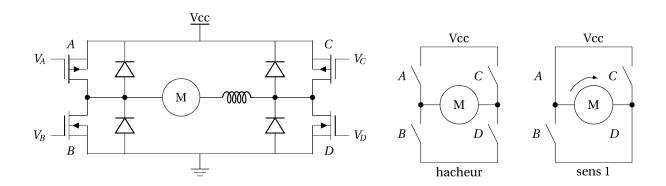


FIGURE 10 – Principe général du fonctionnement d'un hacheur à quatre quadrants.

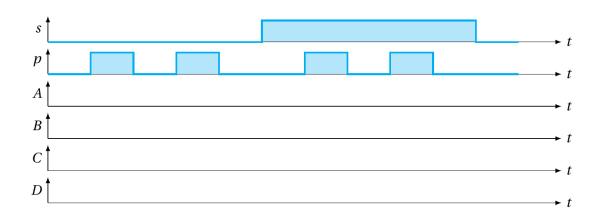


FIGURE 11 – Chronogramme de fonctionnement à compléter.