

# Modélisme ferroviaire avec Arduino et le système de commande numérique DCC

JO BONNÉ

31 décembre 2016

## Résumé

Ce document rend compte d'une progression relative aux réalisations d'une centrale de pilotage numérique de trains, de ses accessoires et des décodeurs ainsi que du programmeur selon la norme DCC, basées sur des cartes Arduino .

Le choix s'est porté sur des modèles de la marque LGB<sup>1</sup> commercialisés bien avant le développement du numérique.

---

1. Lehmann Gross Bahn – repris par Märklin. LGB est spécialisé dans les trains de jardin à l'échelle G soit 1 : 22,5

# -Sommaire

<b>1 De l'analogique au numérique</b>	<b>3</b>
1.1 Loi de Laplace	3
1.2 Le moteur à courant continu	3
1.3 Le moteur à courant continu de la locomotive	3
<b>2 Opérations booléennes sur les nombres binaires</b>	<b>4</b>
2.1 Tables de vérité	4
2.2 Opérations binaires bit à bit	4
<b>3 Le standard DCC</b>	<b>4</b>
3.1 Généralités	4
3.2 Exemple type	5
3.3 La trame idle	6
3.4 Les principales trames	6
3.4.1 Trame destinée au décodeur d'une locomotive	6
3.4.2 Trame destinée à un accessoire d'une locomotive ou des wagons	7
3.4.3 Trame destinée à un décodeur d'accessoire extérieur au train	7
3.4.4 Modifications des CV – variables de configuration – d'un décodeur	7
<b>4 Présentation du circuit — cahier des charges</b>	<b>8</b>
<b>5 Progression</b>	<b>8</b>
5.1 Informatique	8
5.1.1 Tension carrée	8
5.1.2 Production de l'idle	9
5.1.3 Modification de la vitesse	10
5.2 Matériel de base	11
5.2.1 Caractéristiques d'un décodeur Digitrax DH126D	11
5.2.2 Caractéristiques d'une locomotive LGB	11
5.3 Fabrication de la centrale DCC	12
5.3.1 Le booster LMD18200	12
5.4 Fabrication du décodeur de locomotive	13
5.4.1 Fonctionnement d'un décodeur du commerce	13
5.4.2 Principe de gestion du moteur de la locomotive	13

# 1-De l'analogique au numérique

## 1.1-Loi de Laplace

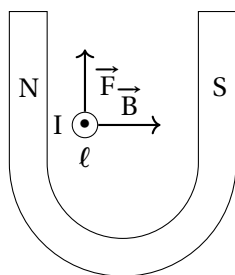


FIGURE 1 - câble dans un champ magnétique

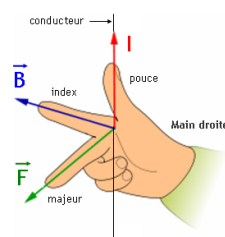


FIGURE 2 - règle des 3 doigts

C'est cette loi qui explique le fonctionnement du moteur à courant continu qui tracte la locomotive.

On considère un aimant en U. Il possède 2 pôles Nord et Sud. Dans l'espace entre ces 2 pôles, appelé entrefer, règne un champ magnétique noté  $\vec{B}$ , vecteur orienté du pôle Nord vers le pôle Sud.

On place une longueur  $\ell$  d'un fil conducteur perpendiculairement au vecteur  $\vec{B}$  dans l'entrefer. Ce fil est parcouru par un courant d'intensité  $I$ . Sur la figure, on a représenté  $I$  par un point  $\odot$  : le courant est dirigé perpendiculairement et sortant de la feuille. (Dans l'autre sens, c'est-à-dire pénétrant dans la feuille, on l'aurait représenté par  $\otimes$ )

La loi de Laplace dit qu'une longueur  $\ell$  d'un fil conducteur parcouru par une intensité  $I$  dans un champ magnétique  $B$  est soumise à une force  $\vec{F}$  dont la valeur vaut :

$$F = B \cdot I \cdot \ell$$

orientée selon la « règle des 3 doigts de la main droite » :

## 1.2-Le moteur à courant continu

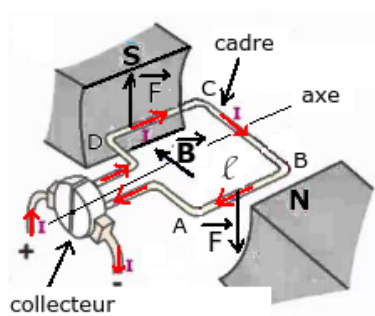


FIGURE 3 - moteur à courant continu

La figure montre le principe de fonctionnement d'un moteur à courant continu.

Un cadre ABCD est parcouru par un courant  $I$ . En appliquant la loi de Laplace aux parties AB et CD, on obtient 2 forces comme indiquées sur la figure. Ces forces créent une rotation non seulement du cadre mais aussi du collecteur par où rentre le courant dans le cadre et par où il en sort<sup>2</sup>. Au moment où le cadre se positionne perpendiculairement au champ magnétique, le cadre reçoit du collecteur un courant dans l'autre sens, ce qui inverse le sens des vecteurs force sur les parties AB et CD entraînant la poursuite du mouvement de rotation.

Une augmentation de l'intensité du courant augmente la valeur du couple de forces responsable de la rotation et en conséquence l'augmentation de la vitesse de rotation.

Un changement de sens du courant inverse le sens des vecteurs forces donc la rotation se fait dans l'autre sens.

Remarque : Pour augmenter la valeur de la force de rotation du moteur, les fabricants :

- multiplient le nombre de tours de cadre,
- le nombre de cadres
- et placent ces cadres autour d'une masse en fer qui tourne avec ces cadres augmentant ainsi la valeur du champ magnétique  $B$ .

## 1.3-Le moteur à courant continu de la locomotive

Autrefois la plupart des fabricants de trains électriques miniatures plaçaient dans la locomotive un moteur à courant continu. Par l'intermédiaire de patins métalliques qui glissaient sur chacun des rails, une tension continue réglable permettait de modifier la vitesse et si le courant changeait de sens, le moteur tournait dans l'autre sens, entraînant la locomotive en marche arrière.

Dans le numérique, la tension continue réglable est remplacée par une tension « PWM », hachée, périodique de fréquence suffisamment élevée : pendant une partie<sup>3</sup> de la période, la tension est maximum et pendant le reste de la période, la tension est nulle. Le moteur à courant continu fonctionne tout aussi bien avec ce type de tension.

2. Le collecteur est constitué de 2 parties identiques séparées par un isolant. Le contact entre le collecteur et le circuit d'alimentation se fait par l'intermédiaire de « balais » qui frottent sur chacune des 2 parties. Ces balais sont souvent en graphite.

3. c'est le rapport cyclique réglable de 0 à 100%

## 2–Opérations booléennes sur les nombres binaires

Ces opérations ont des applications dans le domaine logiciel et dans le domaine matériel (portes logiques). L'opération XOR est utilisée dans chaque trame envoyée par la centrale DCC vers les décodeurs pour vérifier la bonne transmission des données.

### 2.1–Tables de vérité

AND	0	1
0	0	0
1	0	1

OR	0	1
0	0	1
1	1	1

XOR	0	1
0	0	1
1	1	0

NAND	0	1
0	1	1
1	1	0

NOR	0	1
0	1	0
1	0	0

XNOR	0	1
0	1	0
1	0	1

### 2.2–Opérations binaires bit à bit

	1	0	1	0
AND	1	1	0	0
	1	0	0	0

	1	0	1	0
OR	1	1	0	0
	1	1	1	0

	1	0	1	0
XOR	1	1	0	0
	0	1	1	0

	1	0	1	0
NAND	1	1	0	0
	0	1	1	1

	1	0	1	0
NOR	1	1	0	0
	0	0	0	1

	1	0	1	0
XNOR	1	1	0	0
	1	0	0	1

## 3–Le standard DCC



### 3.1–Généralités

Le générateur ou centrale DCC « Digital Command Control »<sup>4</sup> crée une tension formée par une succession ininterrompue de trames de 2 types :

- des trames appelées « idle »<sup>5</sup> apportant la puissance permettant aux trains de se déplacer. Elles sont brièvement interrompues par :
- des trames spécifiques destinées aux décodeurs de chaque locomotive ou aux décodeurs d'accessoires. Chaque locomotive, chaque accessoire interne (lampe d'éclairage d'un wagon ou signal sonore par exemple) d'une locomotive a une adresse spécifique ainsi que chaque accessoire externe (aiguillage par exemple). Pour chaque trame, les décodeurs vérifient si l'adresse correspond à celle du matériel pour lequel il est installé et si c'est le cas, il met en place l'instruction portée par la trame.

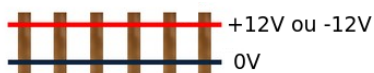


FIGURE 4 – potentiel de chaque rail

Une voie reste au potentiel 0 V tandis que l'autre reçoit *en permanence* un potentiel alternatif carré symétrique de valeurs -12 V et +12 V constitué par des périodes pouvant avoir 2 valeurs différentes :

4. en français « Système de Commande Numérique »

5. on peut traduire par *tâche routinière de fond*

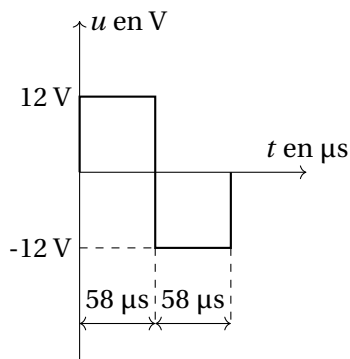


FIGURE 5 – bit 1

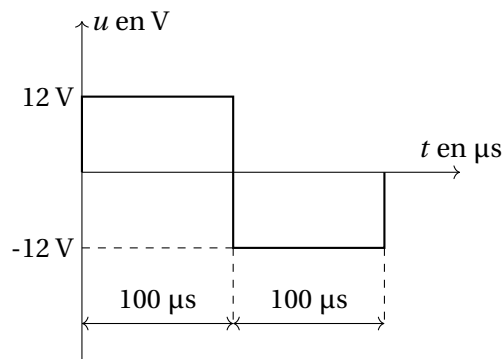


FIGURE 6 – bit 0

La période de  $2 \times 58 \mu\text{s}$ <sup>6</sup> correspondra dans ce qui suit à l'envoi ou la réception d'un bit 1 et la période de  $2 \times 100 \mu\text{s}$ <sup>7</sup> à l'envoi ou la réception d'un bit 0.<sup>8</sup>

Cette tension d'alimentation se rapproche de la modulation de fréquence (FM) : l'amplitude ne change pas mais on travaille avec 2 valeurs de fréquence.<sup>9</sup>

### 3.2–Exemple type

Chaque trame est constitué de plusieurs parties :

- ① Un préambule qui prévient qu'une nouvelle commande suit : il est constitué d'au moins 12 bits 1,
- ② puis un bit 0 signifiant que ce qui suit est l'adresse,
- ③ puis l'adresse spécifique :  
il s'agit généralement d'un octet (ou byte, de 8 bits) mais l'adresse d'une locomotive peut être constituée de 2 octets.
- ④ puis un bit 0 signifiant que ce qui suit est l'instruction qui doit être exécutée à cette adresse.
- ⑤ puis cette instruction :  
il s'agit généralement d'un octet parfois 2 ou 3 octets.
- ⑥ puis un bit 0 signifiant que ce qui suit est la détection d'erreur.  
Elle est justifiée car il y a de nombreux parasites en raison de mauvais contacts entre les rails et la locomotive.
- ⑦ Il s'agit du résultat de l'opération bit à bit (adresse)XOR(instruction)
- ⑧ enfin un bit 1 signifiant que la trame est terminée.

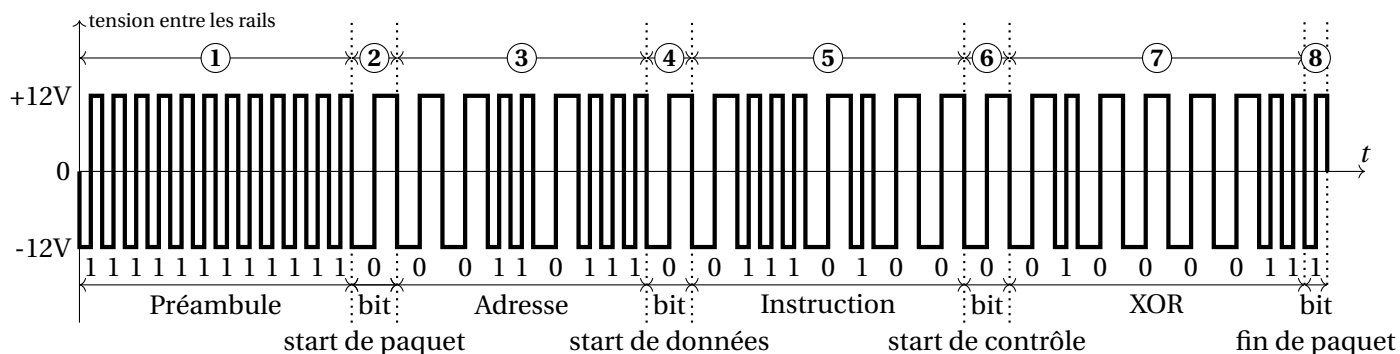


FIGURE 7 – Exemple de trame spécifique envoyée par la centrale DCC

Dans cet exemple :

voir les détails au §3.4.1.1

- L'adresse est 0 0110111 : le premier 0 car l'adresse est sur 1 octet, 0110111 soit  $1 + 2 + 4 + 16 + 32 = 55$  en décimal.
- L'instruction est 01 1 1 0100 : 01 car elle est destinée au moteur de la locomotive puis 1 pour la marche avant puis 1 pour indiquer qu'il y a 28 paliers de vitesse et enfin 0100 valeur correspondant au 6<sup>e</sup> palier de vitesse
- 01000011 est le résultat de l'opération bit à bit (00110111)XOR(01110100)

voir §2.2

6.  $55 \mu\text{s}$  à  $61 \mu\text{s}$  selon la norme

7.  $> 100 \mu\text{s}$  selon la norme

8. Cette méthode peut faire penser au code morse.

9. Ce système est défini par une norme du NMRA (National Model Railroad Association) adopté par le MOROP (Union Européenne des Modélistes Ferroviaires et des Amis des Chemins de Fer)



L'adresse est à **7 bits** dans le même octet, marqués A.

Il y a  $2^7 = 128$  paliers de vitesse.

### 3.4.1.3–locomotive ayant une adresse sur 2 octets, 14 ou 28 paliers de vitesse

préambule 0 11AAAAAA 0 AAAAAAAA 0 01DVV<sub>4</sub>V<sub>3</sub>V<sub>2</sub>V<sub>1</sub> 0 contrôle 1

L'adresse est à **14 bits** répartis sur 2 octets, marqués A.

### 3.4.1.4–locomotive ayant une adresse sur 2 octets, 128 paliers de vitesse

préambule 0 11AAAAAA 0 AAAAAAAA 0 00111111 0 DV<sub>7</sub>V<sub>6</sub>V<sub>5</sub>V<sub>4</sub>V<sub>3</sub>V<sub>2</sub>V<sub>1</sub>V<sub>0</sub> 0 contrôle 1

L'adresse est à **14 bits** répartis sur 2 octets, marqués A.

Il y a  $2^7 = 128$  paliers de vitesse.

### 3.4.2–Trame destinée à un accessoire d'une locomotive ou des wagons

préambule 0 0AAAAAAAA 0 100F<sub>ℓ</sub>F<sub>4</sub>F<sub>3</sub>F<sub>2</sub>F<sub>1</sub> 0 contrôle 1

préambule 0 0AAAAAAAA 0 1010F<sub>8</sub>F<sub>7</sub>F<sub>6</sub>F<sub>5</sub> 0 contrôle 1

préambule 0 0AAAAAAAA 0 1011F<sub>12</sub>F<sub>11</sub>F<sub>10</sub>F<sub>9</sub> 0 contrôle 1

préambule 0 0AAAAAAAA 0 11011110 0 F<sub>20</sub>F<sub>19</sub>F<sub>18</sub>F<sub>17</sub>F<sub>16</sub>F<sub>15</sub>F<sub>14</sub>F<sub>13</sub> 0 contrôle 1

préambule 0 0AAAAAAAA 0 11011111 0 F<sub>28</sub>F<sub>27</sub>F<sub>26</sub>F<sub>25</sub>F<sub>24</sub>F<sub>23</sub>F<sub>22</sub>F<sub>21</sub> 0 contrôle 1

On reconnaît l'adresse de la locomotive (ici sur 7 bits). Les différentes fonctions  $F_i$  sont activées par 1 et désactivée par 0.  $F_\ell$  est spécifique des feux de la locomotive.

Les différentes fonctions sont décrites dans le manuel du décodeur. Elles gèrent les lampes, sons ...

### 3.4.3–Trame destinée à un décodeur d'accessoire extérieur au train

Par exemple un aiguillage.

préambule 0 10AAAAAA 0 1AAA1DDD 0 contrôle 1

L'adresse est à **9 bits** répartis sur 2 octets. L'instruction est sur **3 bits**.

### 3.4.4–Modifications des CV – variables de configuration – d'un décodeur

Un décodeur du commerce est vendu configuré, notamment à l'adresse 3 pour la locomotive dans laquelle il sera installé. Si l'on souhaite agir sur 2 locomotives utilisant le même circuit, il faut pouvoir modifier l'adresse de l'une des 2.

préambule 0 0AAAAAAAA 0 111001CVCV 0 CVCVCVCVCVCVCVCVCV 0 DDDDDDDD 0 contrôle 1

Le manuel du décodeur permet de connaître le nombre décimal relatif à chaque CV. 10 bits sont réservés pour indiquer le nombre binaire correspondant.

8 bits D permettent de placer le nombre binaire de la valeur à placer dans le CV

Exemple : on veut changer l'adresse 3 en 4 dans le CV1 :

préambule 0 00000011 0 11100100 0 00000001 0 00000100 0 contrôle 1

## 4-Présentation du circuit — cahier des charges

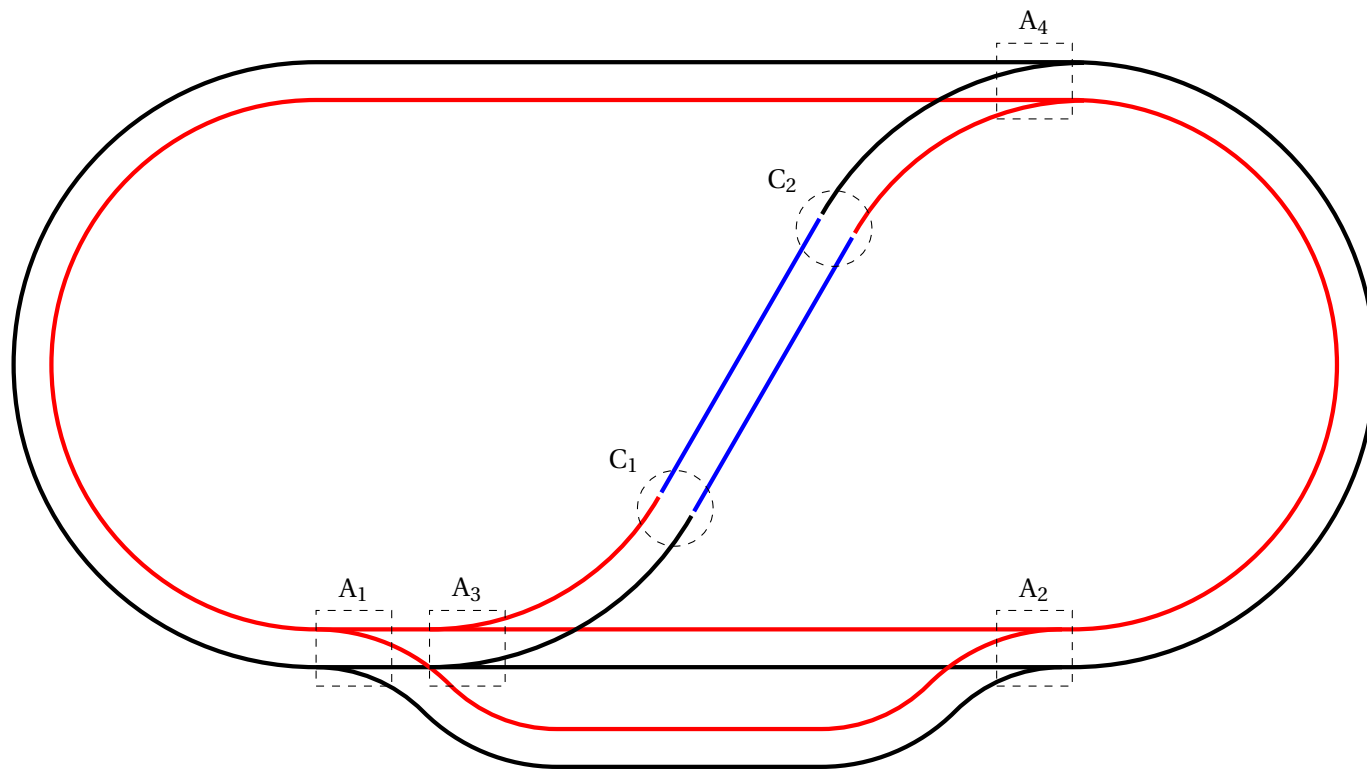


FIGURE 9 – circuit

Ce circuit doit être alimenté en numérique en respectant la norme DCC.

Deux locomotives doivent pouvoir être commandées indépendamment l'une de l'autre. Le sens de parcours et la vitesse doivent pouvoir être réglés. Elles doivent pouvoir rouler en sens avant et en sens arrière.

Des procédures d'évitement et d'arrêt d'urgence doivent être étudiées et mises en œuvre.

Les aiguillages A<sub>1</sub>, A<sub>2</sub>, A<sub>3</sub> et A<sub>4</sub> doivent être motorisés et réglables.

La partie centrale permet un retournement du sens de parcours de la locomotive sur le voie extérieure. On placera donc en C<sub>1</sub> et C<sub>2</sub> un dispositif permettant d'éviter un court-circuit.

Les 2 locomotives devraient pouvoir circuler de façon automatisée.

## 5-Progression

Un oscilloscope numérique est indispensable pour effectuer cette progression <sup>10</sup>.

### 5.1-Informatique

#### 5.1.1-Tension carrée

```
void setup(){
  TCCR2A=0;
  TCCR2B=B010;
  TIMSK2=0x01;
  DDRD |= B100000;
  PORTD &= ~(1<<5);
}
```

```
ISR(TIMER2_OVF_vect){
  TCNT2=141;
  PORTD ^= (1<<5);
}
```

```
void loop() {
}
```

Ce programme crée une tension carrée entre 0 V et 5 V de période  $2 \times 58\mu\text{s} = 116\mu\text{s}$ .

TCCR2A=0; mode normal

TCCR2B=B010; prescaler 8 : fréquence  $\frac{16}{8} = 2 \text{ MHz}$  soit  $T = 0,5\mu\text{s}$

ISR(TIMER2\_OVF\_vect){ } est une boucle au cours de laquelle un compteur TCNT2 incrémente par défaut une valeur de 0 à 255. En indiquant TCNT2=142; le compteur part de 142 jusque 255 soit 113 intervalles de  $0,5\mu\text{s}$  donc  $56,5\mu\text{s}$ .

PORTD ^= (1<<5); à chaque boucle le potentiel de la broche 5 est changé : 0 V ou 5 V.

10. Par exemple Rigol DS1052E – ≈ 330 €

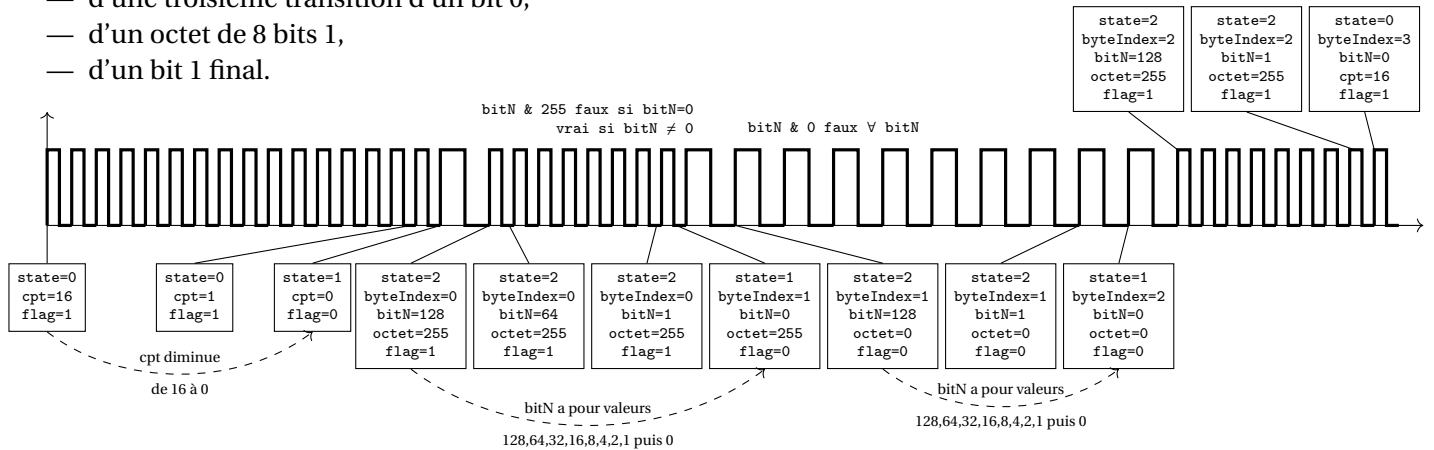


Pour créer une tension carrée entre 0 V et 5 V de période  $2 \times 116 \mu\text{s} = 232 \mu\text{s}$ , il suffit de remplacer  $\text{TCNT2}=142$ ; par  $\text{TCNT2}=26$ ; . Le compteur part de 26 jusque 255 soit 229 intervalles de  $0,5 \mu\text{s}$  donc  $114,5 \mu\text{s}$ .

### 5.1.2–Production de l'idle

L'idle est un signal dont le motif est constitué successivement

- d'un préambule de 16 bits 1,
- d'une transition d'un bit 0,
- d'un octet de 8 bits 1,
- d'une deuxième transition d'un bit 0,
- d'un octet de 8 bits 0,
- d'une troisième transition d'un bit 0,
- d'un octet de 8 bits 1,
- d'un bit 1 final.



```
#define BIT1 143
#define BIT0 30
byte timer0=BIT1;
byte flag=1;
byte updown = 1;
#define PREAMBULE 0
#define TRANSITION 1
#define INSTRUCTION 2
byte state= PREAMBULE;
byte cpt = 16;
byte octet = 0;
byte bitN = 0x80;
```

```
struct Message {
    byte data[7];
    byte len;
};
```

```
#define MAXMSG 1
```

```
struct Message msg[MAXMSG] = {
    { { 0xFF, 0, 0xFF, 0, 0, 0, 0 }, 3}
};
int msgIndex=0;
int byteIndex=0;
```

```
void setup() {
    DDRD|=(1<<4);
    DDRB&=~(1<<4);
    SetupTimer2();
}
```

```
void SetupTimer2(){
    TCCR2A = 0;
    TCCR2B = 0<<CS22 | 1<<CS21 | 0<<CS20; //010
    TIMSK2 = 1<<TOIE2;
    TCNT2=BIT1;
}
```

La broche 4 est déclarée en sortie puis portée au potentiel 0 V.

Pour produire le motif, on utilise le Timer2 de l'Atmega 328. Le compteur TCNT2 garde une valeur de 0 à 255 pendant une durée définie par le prescaler TCCR2B. Ici, les 3 bits de poids faible sont 010 ce qui correspond à un prescaler de 8. La fréquence est donc de  $\frac{16}{8} = 2 \text{ MHz}$  soit une durée de  $\frac{1}{2M} = 0,5 \mu\text{s}$  entre 2 valeurs du compteur.

Le compteur est initialisé à  $\text{BIT1}=143$ , valeur qui augmente progressivement jusque 255 soit 112 durées de  $0,5 \mu\text{s} = 56 \mu\text{s}$ , ce qui correspond à un peu moins de la valeur normalisée de la demi-période du bit 1 ( $58 \mu\text{s}$ )<sup>11</sup>.

```
ISR(TIMER2_OVF_vect) {
    if (updown==1) {
        PORTD|=(1<<4);
        updown = 0;
        TCNT2=timer0;
    } else {
        PORTD&=~(1<<4);
        updown = 1;

        switch(state) {
            case PREAMBULE:
                flag=1;
                cpt--;
                if (cpt == 0) {
                    state = TRANSITION;
                    byteIndex = 0;
                }
                break;
            case TRANSITION:
                flag=0;
                state = INSTRUCTION;
        }
    }
}
```

11. On rappelle que la norme précise que cette durée doit se situer entre 55 et 61  $\mu\text{s}$ . On signale que la durée effective de la demi-période est aussi affectée par la durée de traitement des instructions du programme.

```

    bitN = 0x80;
    octet = msg[msgIndex].data[byteIndex];
    break;
case INSTRUCTION:
    if (octet & bitN) {
        flag = 1; //tant que bitN non nul
    } else {
        flag = 0;
    }
    bitN = bitN >> 1;
    if (bitN == 0) {
        byteIndex++;
        if (byteIndex >= msg[msgIndex].len) {
            state = PREAMBULE;
            cpt = 16;
        } else {
            state = TRANSITION ;
        }
    }
    break;
} //fin switch

```

```

    if (flag) {
        TCNT2=BIT1;
        timer0=BIT1;
    } else {
        TCNT2=BIT0;
        timer0=BIT0;
    } //fin if flag
} //fin if upown
} //fin ISR

void loop() {
}

```

ISR(TIMER2\_OVF\_vect){} est un programme d'interruption interne du Timer2 basé sur un compteur TCNT2 qui évolue jusque 255<sup>12</sup>.

Ce programme traite en boucle d'abord la partie haute (5 V) d'un bit 1 ou d'un bit 0 puis la partie basse (0 V) du même bit. C'est au cours de la partie basse que sont modifiées les variables (sauf updown).

### 5.1.3–Modification de la vitesse

On considère une locomotive ayant une adresse 3 sur 1 octet avec 128 paliers de vitesse donc sur 2 octets conformément au §3.4.1.2 page 6

La broche 4 servira de sortie DCC.

On déclare en entrée la broche 12. Si cette broche est portée à 5 V le moteur tournera dans un sens. Si elle est portée à 0 V, elle tournera dans l'autre sens.

Un potentiomètre est branché entre 5 V et 0 V, le curseur sur l'entrée A2.

```

#define BIT1 143
#define BIT0 30
byte timer0=BIT1;
byte flag=1;
byte updown = 1;
#define PREAMBULE 0
#define TRANSITION 1
#define INSTRUCTION 2
byte state= PREAMBULE;
byte cpt = 16;
byte octet = 0;
byte bitN = 0x80;

int locoAdr=3;
int vitesse=0;
int vitesse0=0;
int dir;
int dir0;

```

On déclare donc locoAdr l'adresse de la locomotive, une variable vitesse et une variable direction dir.

```

struct Message {
    byte data[7];
    byte len;
} ;

#define MAXMSG 2

struct Message msg[MAXMSG] = {
    { { 0xFF, 0, 0xFF, 0, 0, 0, 0}, 3},
    { { locoAdr, 0x3F, 0, 0, 0, 0, 0}, 4}
};

int msgIndex=0;

```

```
int byteIndex=0;
```

Le programme va d'abord créer le motif permanent de l'idle puis immédiatement après le motif relatif à la dernière vitesse souhaitée en procédant exactement de la même façon que pour l'idle.

```

void setup() {
    DDRD|=(1<<4);
    DDRB&=~(1<<4);
    PORTB|=(1<<4); //pull-up
    SetupTimer2();
    lectureVitesse();
    dccMsg();
}

void SetupTimer2(){
    TCCR2A = 0;
    TCCR2B = 0<<CS22 | 1<<CS21 | 0<<CS20; //010
    TIMSK2 = 1<<TOIE2;
    TCNT2=BIT1;
}

ISR(TIMER2_OVF_vect) {
    if (updown==1) {
        PORTD|=(1<<4);
        updown = 0;
        TCNT2=timer0;
    } else {
        PORTD&=~(1<<4);
        updown = 1;

        switch(state) {

```

12. OVF overflow avec mise à 1 d'un flag.

```

case PREAMBULE:
    flag=1;
    cpt--;
    if (cpt == 0) {
        state = TRANSITION;
        msgIndex++;
        if (msgIndex >= MAXMSG) {msgIndex = 0;}
        byteIndex = 0;
    }
    break;
case TRANSITION:
    flag=0;
    state = INSTRUCTION;
    bitN = 0x80;
    octet = msg[msgIndex].data[byteIndex];
    break;
case INSTRUCTION:
    if (octet & bitN) {
        flag = 1; //tant que bitN non nul
    } else {
        flag = 0;
    }
    bitN = bitN >> 1;
    if (bitN == 0) {
        byteIndex++;
        if (byteIndex >= msg[msgIndex].len) {
            state = PREAMBULE;
            cpt = 16;
        } else {
            state = TRANSITION ;
        }
    }
    break;
} //fin switch

if (flag) {
    TCNT2=BIT1;
    timer0=BIT1;
} else {
    TCNT2=BIT0;
    timer0=BIT0;
} //fin if flag
} //fin if upown
} //fin ISR

void loop() {
    delay(200);
    if (lectureVitesse()) {
        dccMsg();
    }
}

```

```

}

```

La fonction `lectureVitesse` permet d'obtenir en temps réel la valeur de la variable `vitesse` comprise entre 0 et 127. Elle permet aussi d'obtenir la valeur de la variable `dir` égale à 0 ou 1. Le résultat de cette fonction est une variable booléenne `changed` qui prend la valeur `true` en cas de changement d'une des valeurs sinon `false`.

```

boolean lectureVitesse() {
    boolean changed = false;
    vitesse = (127L * analogRead(A2))/1023;
    if (vitesse != vitesse0) {
        changed = true;
        vitesse0 = vitesse;
    }
    dir = digitalRead(12);
    if (dir != dir0) {
        changed = true;
        dir0 = dir;
    }
    return changed;
}

```

La fonction `dccMsg` crée le bit de vérification XOR de la commande de vitesse, introduit les valeurs `data` et `xdata` respectivement en 3<sup>e</sup> place et 4<sup>e</sup> place dans la structure du message relatif à la vitesse.

```

void dccMsg() {
    byte data;
    byte xdata;

    if (dir) {
        data = 0x80;
    } else {
        data = 0;
    }

    data |= vitesse;

    xdata = (msg[1].data[0] ^ msg[1].data[1]) ^ data;

    noInterrupts();
    msg[1].data[2] = data;
    msg[1].data[3] = xdata;
    interrupts();
}

```

## 5.2–Matériel de base

### 5.2.1–Caractéristiques d'un décodeur Digitrax DH126D

Ce décodeur va servir pour les essais avant de réaliser un décodeur à partir de la carte Arduino. Il ne doit pas être soumis à une tension supérieure à 16 V et l'intensité totale ne doit pas excéder 0,5 A.

Les fils rouge et noir prennent la tension aux bornes des rails. Les fils orange et gris sont reliés au moteur. Les 5 autres fils permettront notamment de gérer les lampes.

Remarque : Un décodeur du commerce bien que placé entre les rails et le moteur ne doit pas empêcher le fonctionnement de la locomotive en analogique. Dans ce cas, on dit qu'il est à l'adresse 0. Seul le seuil de tension du générateur à partir duquel la locomotive commence à se déplacer est modifié.

### 5.2.2–Caractéristiques d'une locomotive LGB

Il s'agit de 2 locomotives issues de kits (analogiques) de démarrage vendus par LGB dans les années 90. Le générateur fournit une tension continue réglable entre -25 V et +25 V et peut alimenter un réseau de trains jusqu'à une intensité maximum de 1 A. L'intensité dans le moteur de la locomotive est de l'ordre de 0,4 A pour 12 V.

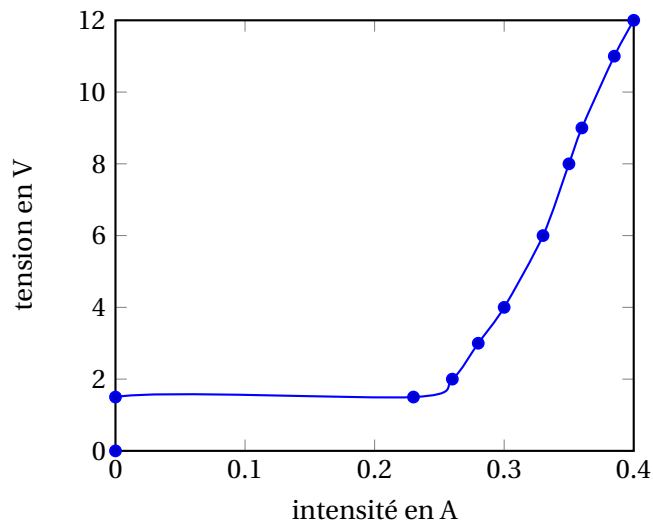


FIGURE 10 – Caractéristique  $u = f(i)$  du moteur de locomotive LGB

Le générateur alimentera la centrale DCC. La tension maximum sera réglée pour ne pas dépasser 12 V aux bornes du décodeur du commerce. Par souci de sécurité il sera souhaitable de placer un régulateur de tension. La tension d'alimentation recommandée pour les cartes Arduino doit être comprise entre 7 et 12 V.

Bien que la locomotive puisse être alimentée jusqu'à 25 V, on limitera la tension à 12 V valeur pour laquelle la vitesse est déjà suffisante et le bruit acceptable. Pour une tension de 12 V, les roues font 100 tours en à peu près 19 s. Chaque roue a un diamètre de 36 mm. Donc la vitesse atteinte est d'environ :

$$v = \frac{100 \times 2 \times 3,14 \times 0,018}{19} = 0,59 \text{ m/s}$$

Chaque locomotive (référence LGB 62201) contient un moteur Bühler dont l'axe supporte de chaque côté une crémaillère. Chaque crémaillère entraîne un pignon sur l'essieu de la roue selon un rapport de réduction 17 : 1, c'est-à-dire qu'il faut 17 tours de l'axe du moteur pour 1 tour de roue. Le pignon, en matière plastique type nylon, doit être bien graissé.

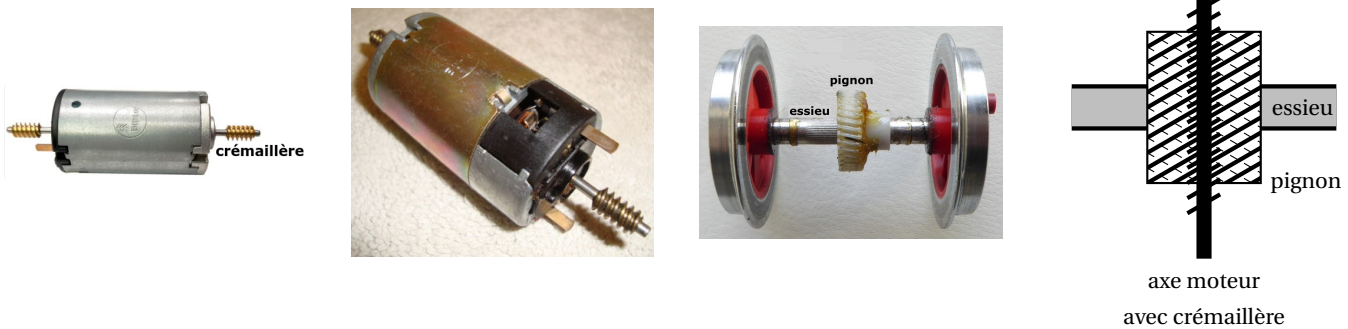


FIGURE 11 – Système pignon-crémaillère associant le moteur aux roues de la locomotive

## 5.3–Fabrication de la centrale DCC

### 5.3.1–Le booster LMD18200

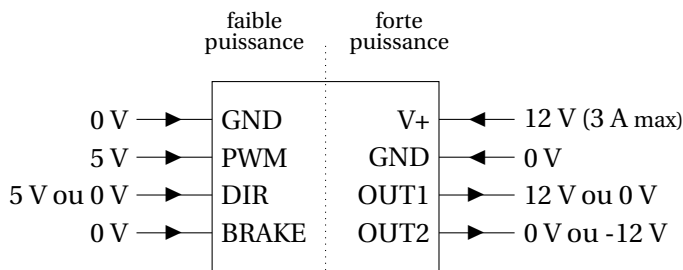
Les broches PWM, BRAKE et DIR peuvent être portées soit au potentiel 5 V soit au potentiel 0 V. On les relie à une carte Arduino. Il s'agit de la partie commande du booster. Elle fonctionne avec une très faible puissance électrique.

La partie dédiée à une puissance plus élevée fonctionne pour un générateur fournissant une tension supérieure à 11 V (jusqu'à 55 V) entre les broches V+ et GND. L'intensité du courant ne doit pas dépasser 3 A. On pourra donc alimenter plusieurs locomotives.

La table de vérité montre que si le potentiel de DIR est un signal carré entre 0 et 5 V de fréquence  $f$ , alors on obtient entre OUT1 et OUT2 une tension carrée entre -12 V et +12 V de même fréquence (si le générateur fournit une tension de 12 V). Le signal reste propre même pour une période de 1  $\mu$ s.

En conséquence, ce module est bien adapté, en intensité, en tension et en fréquence pour réaliser une centrale DCC.

Attention! En cas de court-circuit accidentel il faut prévoir une immédiate coupure du courant.



PWM	DIR	BRAKE	$V_{OUT1} - V_{OUT2}$
1	1	0	12 V
1	0	0	-12 V
0		0	
1	1	1	
0		1	

## 5.4–Fabrication du décodeur de locomotive

### 5.4.1–Fonctionnement d’un décodeur du commerce

Les adresses d’un décodeur du commerce sont préprogrammées en usine. Ces adresses peuvent être modifiées. En général l’adresse d’une locomotive est 3. Si l’on dispose de 2 locomotives, pour qu’elles soient gérées indépendamment, il faut pouvoir modifier l’adresse de l’une des deux.

On peut modifier une adresse avec une carte Arduino.

### 5.4.2–Principe de gestion du moteur de la locomotive

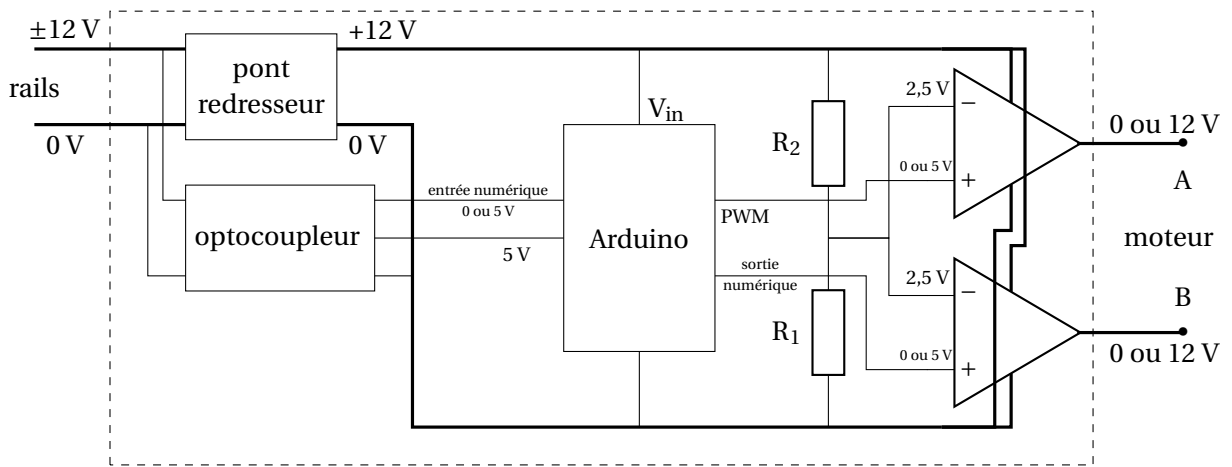


FIGURE 12 – Principe de fonctionnement d’un décodeur

La centrale DCC génère une tension alternative carrée de -12 V et +12 V. Elle doit pouvoir débiter un courant de l’ordre de 2 A pour tracter 2 trains sur les rails.

Le circuit de forte puissance (en trait fort sur la figure) : la tension alternative est d’abord redressée à l’aide d’un pont de 4 diodes. À la sortie du redresseur on doit obtenir une tension quasi-continue d’un peu moins de 12 V en raison des tensions de seuil de 2 diodes en série pour chaque alternance.

Cette tension alimente aussi 2 amplificateurs opérationnels dont les sorties sont branchées au moteur de la locomotive. L’ordre de grandeur de l’intensité dans le moteur est de 1 A. Les amplificateurs opérationnels sont donc des amplificateurs opérationnels de puissance. La tension entre les 2 bornes du moteur peut donc être :

$$U_{AB} = (+12 - 0) = 12 \text{ V} \quad U_{AB} = (+12 - 12) = 0 \quad U_{AB} = (0 - 12) = -12 \text{ V} \quad U_{AB} = (0 - 0) = 0 \text{ V}$$

Le signe de  $U_{AB}$  donne le sens de parcours de la locomotive sur le rail.

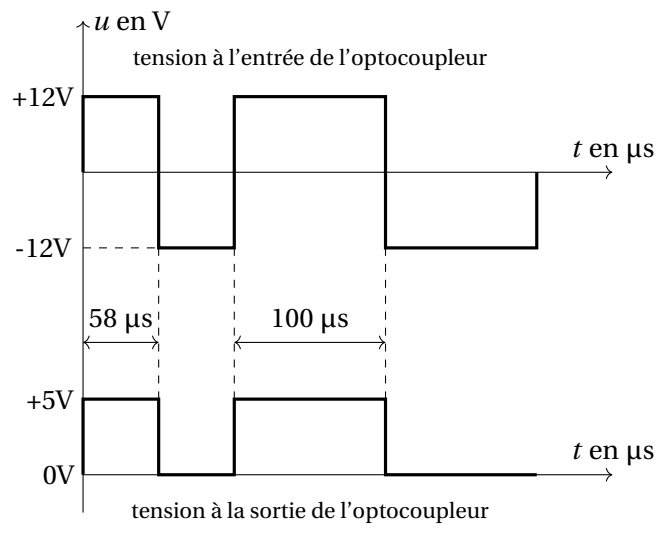
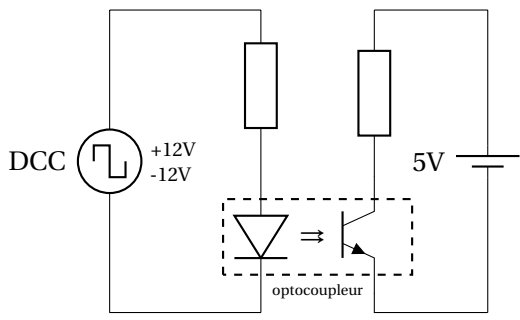
Le circuit de faible puissance (en trait fin sur la figure) :

La tension continue de 12 V alimente une carte Arduino par la broche  $V_{in}$  dont la tension recommandée est de 7 à 12 V. En interne cette tension est régulée en 5 V.

Cette tension alimente aussi un pont diviseur de tension par l’intermédiaire de 2 résistances. Pour obtenir les 2,5 V aux entrées des 2 amplificateurs opérationnels, il faut :

$$\frac{R_1}{R_1 + R_2} = \frac{2,5}{12}$$

Enfin, la tension alternative carrée de -12 V et +12 V est reliée à l’entrée d’un optocoupleur. La figure suivante montre la tension en sortie en fonction de la tension d’entrée :



## -Table des figures

1	câble dans un champ magnétique	3
2	règle des 3 doigts	3
3	moteur à courant continu	3
4	potentiel de chaque rail	4
5	bit 1	5
6	bit 0	5
7	Exemple de trame spécifique envoyée par la centrale DCC	5
8	Trame idle	6
9	circuit	8
10	Caractéristique $u = f(i)$ du moteur de locomotive LGB	12
11	Système pignon-crémaillère associant le moteur aux roues de la locomotive	12
12	Principe de fonctionnement d'un décodeur	13