```
////////////////////////////////////////////////////
////////////////////////////////////////////////////
//////////////////Moniteur DCC//////////////////////
///////////////Développé par AERONEF////////////////
////////////////////////////////////////////////////
///////////Copyright @ Le Site Ferroviaire///////////
////////////////////////////////////////////////////
//////////////////Novembre 2018//////////////////////
////////////////////////////////////////////////////
////////////////////////////////////////////////////
volatile byte demi_bitun = 0;//Variable de comptage d'un demi_byte à 1
volatile byte demi_bitzero = 0;//Variable de comptage d'un demi_byte à 0

volatile byte preambule;//Variable de comptage du Préambule
volatile boolean validpreambule = 1;//Valide la construction du préambule

volatile int tramedcc = 0;//Valide la construction de la trame DCC
volatile byte tramedcccomplete [55] = {0};//Tableau d'enregistrement de la Trame DCC

volatile int i;//Variable de comptage du tableau

////////////////////////////////////////////////
////////////////////////////////////////////////
void setup()
{
Serial.begin (1000000);//Vitesse transmision du moniteur série

TCNT0 = 0;//Mise à 0 du timer 0

TCCR0B = 0b00000011;//Défini le rapport de division du Timer 0 par 64

EICRA = 0b00000001;//CHANGE INT0
EIMSK |= (1<<0);//Autorise INT0

}
////////////////////////////////////////////////
////////////////////////////////////////////////
void loop()
{
}
////////////////////////////////////////////////
///////////Traitement de l'interruption//////////
////////////////////////////////////////////////
ISR (INT0_vect)
//void dureeimpulsion()//Déclenché par CHANGE
{
if (TCNT0 < 19)//Si TIMER0 : durée < 76µs alors c'est un bit 1
 {
 TCNT0 = 0;//Raz compteur interruption

 demi_bitun = demi_bitun + 1 ;//Incrémente demi_bitun
 demi_bitzero = 0;//Raz pour éviter un demi_zero suivi d'un demi_un

 if (demi_bitun == 2)//Valide si deux demi_un se suivent
 {demi_bitun = 0;
```

```
  if (validpreambule == 1)//Si à 1 permet la construction du préambule
  {bitun();}

  if (tramedcc == 1)//Si à 1 permet la construction de la trame DCC
  {debuttramedccun();}

  }//Fin du Valide si deux demi_un se suivent
  }//Fin du if (TCNT0 < 19)//Si durée < 76µs alors c'est un bit 1
   //(16000000/64) = 4µs * 19 = 76µs


else //Si timer TC0 > 19 alors c'est un bit 0
  {
  TCNT0 = 0;//Raz compteur interruption

  demi_bitzero = demi_bitzero + 1 ;//Incrémente demi_bizéro
  demi_bitun = 0;//Raz pour éviter un demi_un suivi d'un demi_zero

  if (demi_bitzero == 2)
  {demi_bitzero = 0;

  if (validpreambule == 1)//Si à 1 permet la construction du préambule
  {bitzero();}

  if (tramedcc == 1)//Si à 1 permet la construction de la trame DCC
  {debuttramedcczero();}

  }//Fin du Valide si deux demi_bitzéro se suivent
  }//Fin du if (TCNT0 < 19)//Si durée > 76µs alors c'est un bit 0
   //(16000000/64) = 4µs * 19 = 76µs

}//Fin du void dureeimpulsion()//Déclenché par CHANGE
//////////////////////////////////////////////////
///////Fin du Traitement de l'interruption//////////
//////////////////////////////////////////////////
void bitun()
{
 if (validpreambule == 1)//Si à 1 permet la construction du préambule
   {
   preambule = preambule + 1 ;//Augmente le compteur du preambule
    }//Fin du If
}//Fin du Void bitun()
//////////////////////////////////////////////////
//////////////////////////////////////////////////
//////////////////////////////////////////////////
void bitzero()
{
  if (preambule < 12 && validpreambule == 1)
  //Si validpreambule = 1 permet la construction du préambule
    {
    preambule = 0;
    //Remet la variable préambule à zéro : il faut au moins 12 bit à un consécutifs
    }

  if (preambule > 11 && validpreambule == 1)//Le préambule est validé
    {
```

```
   tramedcc = 1;//L'enregistrement de la trame DCC est possible
   validpreambule = 0;//Préambule contient le nombre de 1
   }

}//Fin du Void bitzero()
////////////////////////////////////////////////
///////////Construction de la trame DCC/////////////
////////////////////////////////////////////////
void debuttramedccun()//Construction de la trame DCC bit 1
{if (i<55)
 {tramedcccomplete[i] = 1;//Enregistrement des bits dans le tableau de la trame DCC
  i = i + 1;
 }
 else
 {
  nbreoctet();//Envoie le Void nbreoctet lorsque le tableau est complet
 }

}//Fin du void octetun()//Recomposition de l'octet 1
////////////////////////////////////////////////
////////////////////////////////////////////////
////////////////////////////////////////////////
void debuttramedcczero()//Construction de la trame DCC bit 0
{if (i<55)
 {tramedcccomplete[i] = 0;//Enregistrement des bits dans le tableau de la trame DCC
  i = i + 1;
 }

}//Fin du void octetzero1()//Recomposition de l'octet 1
////////////////////////////////////////////////
////////////////////////////////////////////////
////////////////////////////////////////////////
void nbreoctet()//Détermine le nombre d'octet valide
{
//Arrêt de l'Interruptions INT0
EIMSK &=~ (1<<0);//detachInterrupt (digitalPinToInterrupt(interruptPin), dureeimpulsion, CHANGE);

if (tramedcccomplete[27] == 1)
{affichagetrame3octets();}

else if (tramedcccomplete[36] == 1)
{affichagetrame4octets();}

else if (tramedcccomplete[45] == 1)
{affichagetrame5octets();}

else if (tramedcccomplete[54] == 1)
{affichagetrame6octets();}

//Retour à l'état initiale des variables
tramedcc = 0;//Construction trame DCC impossible

preambule = 0;//Variable préambule à 0

i = 0;//Compteur variable tableau à 0
```

```
validpreambule = 1;//Construction Préambule possible

//Reprise de l'Interruptions INT0
EIMSK |= (1<<0);

}
////////////////////////////////////////////////
////////////////////////////////////////////////
////////////////////////////////////////////////
void affichagetrame3octets()//Affichage de la trame 3 octets
{
Serial.print (preambule);//Affichage le contenu de la variable Préambule
Serial.print (" ");//Intervalle

Serial.print (tramedcccomplete[0]);
Serial.print (" ");

//Affichage octet 1
for (volatile byte j=1;j<=8;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[9]);
Serial.print (" ");

//Affichage octet 2
for (volatile byte j=10;j<=17;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[18]);
Serial.print (" ");

//Affichage octet 3
for (volatile byte j=19;j<=26;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.println (tramedcccomplete[27]);

}
////////////////////////////////////////////////
////////////////////////////////////////////////
////////////////////////////////////////////////
void affichagetrame4octets()//Affichage de la trame 4 octets
{
Serial.print (preambule);
Serial.print (" ");

Serial.print (tramedcccomplete[0]);
Serial.print (" ");

for (volatile byte j=1;j<=8;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");
```

```
Serial.print (tramedcccomplete[9]);
Serial.print (" ");

for (volatile byte j=10;j<=17;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[18]);
Serial.print (" ");

for (volatile byte j=19;j<=26;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[27]);
Serial.print (" ");

for (volatile byte j=28;j<=35;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.println (tramedcccomplete[36]);

}
////////////////////////////////////////////////
////////////////////////////////////////////////
////////////////////////////////////////////////
void affichagetrame5octets()//Affichage de la trame 5 octets
{
Serial.print (preambule);
Serial.print (" ");

Serial.print (tramedcccomplete[0]);
Serial.print (" ");

for (volatile byte j=1;j<=8;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[9]);
Serial.print (" ");

for (volatile byte j=10;j<=17;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[18]);
Serial.print (" ");

for (volatile byte j=19;j<=26;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[27]);
Serial.print (" ");
```

```
for (volatile byte j=28;j<=35;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[36]);
Serial.print (" ");

for (volatile byte j=37;j<=44;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.println (tramedcccomplete[45]);

}
////////////////////////////////////////////////
////////////////////////////////////////////////
////////////////////////////////////////////////
void affichagetrame6octets()//Affichage de la trame 6 octets
{
Serial.print (preambule);
Serial.print (" ");

Serial.print (tramedcccomplete[0]);
Serial.print (" ");

for (volatile byte j=1;j<=8;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[9]);
Serial.print (" ");

for (volatile byte j=10;j<=17;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[18]);
Serial.print (" ");

for (volatile byte j=19;j<=26;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.print (tramedcccomplete[27]);
Serial.print (" ");

for (volatile byte j=28;j<=35;j++)
{Serial.print (tramedcccomplete[j]);}

Serial.print (" ");

Serial.print (tramedcccomplete[36]);
Serial.print (" ");

for (volatile byte j=37;j<=44;j++)
{Serial.print (tramedcccomplete[j]);}
```

```
Serial.print (" ");

Serial.print (tramedcccomplete[45]);
Serial.print (" ");

for (volatile byte j=46;j<=53;j++)
{Serial.print (tramedcccomplete[j]);}
Serial.print (" ");

Serial.println (tramedcccomplete[54]);

}
```