

# ENTRE-VOIES

Mai-Juin 2015  
N° 194

**Périodique d'information du R.C.F.E.B.** Editeur responsable : Philippe CAMUS, Neuville 43, 4987 Stoumont. – les articles publiés n'engagent que leurs auteurs, sauf mention, ils sont la propriété du C.F.E.B. asbl, ils peuvent être reproduits dans un but non lucratif à condition de citer la source et les auteurs, et d'envoyer une copie à l'éditeur. Téléphone : Jean-Claude SIMAR, président 087 - 33 89 56 Email : jean-claude.simar@skynet.be Web : <http://www.rcfeb.be> Compte bancaire : BE 34 751-2028029-90.

## Le mot du Président :

Bonjour à toutes et à tous.

Quand vous lirez ces quelques lignes, nous aurons déjà derrière nous notre voyage annuel au salon du modélisme de Dortmund.

Belle réussite, nous étions 44 participants que je remercie pour le respect des horaires établis. J'espère que chacun y aura trouvé son petit bonheur !

Egalement terminée, l'Assemblée Générale de notre club du 24 avril.

A méditer, l'année prochaine sera une année de réélection d'un nouveau comité et je vous demande de réfléchir afin de poser votre candidature.

Prochain défi, l'organisation de la Porte Ouverte à l'occasion de notre 60ème anniversaire qui se tiendra le 15/16/22 et 23 août soit 4 jours.

Le comité compte sur votre présence afin de faire de cette manifestation un succès. Des listes de participation seront affichées au club. Les bonnes volontés sont les bienvenues pour préparer les réseaux.

Nous sommes toujours demandeurs de membres qui accepteraient d'animer une soirée sur un sujet particulier ou tout simplement la présentation d'un DVD.

Nous espérons votre présence active à nos séances.

Pour le Comité

 **Le Président. J-C Simar**

## Les cartes Arduino

Une carte « Arduino » est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur qui peut être programmé pour analyser et produire des signaux électriques de manière à effectuer des tâches diverses comme la domotique (contrôle des appareils domestiques, éclairage, chauffage), le pilotage d'un robot, etc...



C'est une plate-forme basée sur une interface entrée/sortie simple,

L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour communiquer avec les autres circuits, la carte « Uno » possède 13 entrées numériques et 6 entrées analogiques. Les connexions sont établies au travers de connecteurs femelles HE14 situés sur le dessus de la carte, Divers modules peuvent s'empiler sur « Arduino »

La programmation est une application JAVA, libre et multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers d'une liaison USB.

Le langage de programmation est le C++ compilé avec avr-g++ et lié à la bibliothèque de développement « Arduino » pour utilisation de la carte et de ses entrées/sorties.

Il me semble que l'utilisation de cette carte peut rendre pas mal de service pour l'automatisation et les décors des trains miniatures...

A mon avis, l'intérêt de cette carte est aussi la multitude de modules pratiquement prêt à l'emploi qui se raccorde à la carte et sont d'une simplicité et d'une efficacité

remarquable et ce pour un prix très compétitif : exemples, les relais, les petits moteurs pas à pas, les servomoteurs...

Voici quelques exemples :

- La commande d'aiguillages avec un servo-moteur...
- La réalisation d'animations lumineuses
- Un passage à niveau non gardé et ses signaux

Passons directement à un exemple pratique qui vous fera mieux comprendre l'intérêt de cette carte.

Première partie, la conception.

Réalisation d'un passage à niveau non gardé.  
Un train dans un sens de circulation.

1. les signaux fonctionnent en voie libre (clignotant central blanc)
2. détection de l'arrivée du train,
3. signaux en mode voie occupée (deux feux rouges qui clignotent alternativement.)
4. détection du passage du train après le passage à niveau.
5. signal en mode normal après passage du train

Plus tard, nous allons prévoir la détection indépendamment du sens du train.

### **Réalisation des signaux lumineux.**

Programme et mise en place sur le réseau.

Ce petit montage va nous faire comprendre comme raccorder les leds et les faire clignoter.

Comment placer et raccorder des photorésistances et lire leurs valeurs libres ou recouvertes.

Attribuer une valeur à des variables suivant les informations transmises par les LDR.

Ecrire le programme, le compiler et le transmettre à la carte « Arduino »

Matériel :

- 2 photorésistances (LDR) et deux résistances de 10 k ohms
- 4 leds rouge et les résistances (330 ohms, « Arduino » travaille en 5 volts)
- 2 leds blanche et les résistances 330 ohms.
- Carte « Arduino UNO » plaquette d'essais et câbles de connexion.

Placer les composants sur une plaque d'essais.

Placer la photorésistance au début de la plaque d'essais, une patte sur le positif commun.

Placer l'autre patte dans 1ère partie de la ligne avec un câble de connexion et une patte de la résistance de 10 k ohm.

Placer la seconde patte de la résistance sur le négatif commun.

Placer ensuite une led rouge, une patte dans le négatif commun et l'autre dans la ligne.

Placer ensuite une résistance de 330 ohms de la ligne 1 à la ligne 2 puis sur cette ligne 2 un câble de connexion.

Faites de même avec la led blanche puis la led rouge.

Placer ensuite à la fin de la plaquette la seconde photo résistance avec la première comme modèle.

Placer les câbles des photorésistances aux broches A0 et A1 partie analogique

Placer les câbles des leds sur les broches 3 4 5 partie digitale.

Placer les câbles + et - sur la partie 5V et GND.

### **La Programmation**

Action :

Un train passe au-dessus de la première photo résistance (LDR) et change la valeur de celle-ci.

Suite à ce changement la led blanche du signal cède la place aux leds rouges qui clignotent en alternance.

Quand le train passe au-dessus de la seconde photo résistance qui se trouve après le passage à niveau à une distance qui sera établie suivant la longueur du plus long convoi, la valeur de celle-ci change et provoque l'arrêt des leds rouges et l'allumage de la led blanche qui clignote également, mais plus lentement que les leds rouges et signale que le passage est libre.

### **Le langage de programmation :**

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (langage source) qui seront directement converties en langage machine grâce à un compilateur. L'exécution d'un programme « Arduino » s'effectue de manière séquentielle, cad que les instructions sont exécutées les unes à la suite des autres.

### **La structure d'un programme :**

Un programme « Arduino » comporte 3 parties :

1. La partie déclaration des variables (optionnelle)

2. La partie initialisation et configuration des entrées/sorties : la fonction setup()
3. La partie principale qui s'exécute en boucle : la fonction loop().

Dans chaque parties sont utilisées différentes instructions issues de la syntaxe du langage « Arduino »

Déclaration des variables leds et photorésistances en intégrer avec leur attache sur la carte « Arduino »

Déclarations des variables qui prennent la valeur des LDR éclairées ou masquées.

Déclaration des variables de travail qui seront la base du fonctionnement de ce module.

Mais plutôt que de décrire cela, réalisons le programme sans oublier de commenter les actions, tenants et aboutissants qui permettront de comprendre et de modifier ce programme.

```
//programme pour signal lumineux de passage à niveau non gardé,
```

```
int sensor1 = A0;           // LDR1 en analogique A0
int sensor2 = A1;           // LDR2 en analogique A1
int sensorvalue1 = 1;      // variable qui sera
//initialisée suivant la valeur de la LDR1
int sensorvalue2 = 1;      // variable qui sera
//initialisée suivant la valeur de la LDR2
int ledr1 = 3;              // broche led rouge 1
int ledr2 = 4;              // broche led blanche
int ledb = 5;               // broche led rouge 2
int TPT1 = 0;               // déclaration de variable de travail
int TPT2 = 0;
```

```
void setup()
{
  // place les broches 3,4,5 en sortie
  pinMode (ledr1,OUTPUT);
  pinMode (ledr2,OUTPUT);
  pinMode (ledb,OUTPUT);
  // pour permettre la lecture des valeurs des LDR
  Serial.begin (9600);
}
```

```
void loop()
{
  // lecture de la valeur des LDR 1 puis LDR 2
  sensorvalue1 = analogRead(sensor1);
  sensorvalue2 = analogRead(sensor2);

  // attribution d'une valeur aux variables de travail
  if (sensorvalue2<=50)
  { TPT1 = 0; }
  if (sensorvalue1<=50)
  { TPT1 = 1; }

  // lecture pour définir les valeurs de travail
  Serial.print( " sensorv1 = ");
  Serial.println( sensorvalue1);
  Serial.print ( " sensorv2 = ");
  Serial.println(sensorvalue2);
  Serial.print(" TPT1 = ");
  Serial.println( TPT1 );
  delay(1500); // délais pour la lecture des variables
```

```
if (TPT1 == 1) // programme réel
{
  // place la led rouge 1 en haut, le led s'allume
  digitalWrite ( ledr1,HIGH);

  // place les autre leds en "bas", elles sont éteintes
  digitalWrite ( ledr2,LOW);
  digitalWrite ( ledb ,LOW);
  delay(500);//délai pour la fréquence de clignotement

  digitalWrite ( ledr1,LOW);
  digitalWrite ( ledr2,HIGH);
  digitalWrite (ledb,LOW);
  //dans cette partie, seule les leds rouge travaillent;
  nous sommes dans la partie passage de train

  delay(500);// délai d'une 1/2 seconde
}

if (TPT1 == 0)
{
  digitalWrite(ledb,HIGH);
  // Nous sommes dans la partie du passage
  // à niveau libre, seule la led blanche travaille

  digitalWrite (ledr1,LOW);
  // et clignote deux fois moins vite que les rouges

  digitalWrite (ledr2,LOW); //
  delay(1000); // délai de 1 seconde ( 1000 milis )

  digitalWrite (ledb,LOW);
  delay(1000);
  digitalWrite(ledb,HIGH);
}
}
```

## La réalisation.

Placer la LDR de détection d'arrivée du train,

Placer la LDR a +/- 30cm du passage pour laisser le temps aux usagers de la route de réaliser l'arrivée prochaine d'un train.

Placer les signaux à droite de la route (suivant le sens de circulation) et raccorder les fils,

Placer le seconde LDR 20 cm plus loin que la longueur du plus long train qui pourrait passer sur cette partie du réseau,

Faire les raccordements sur la carte « Arduino »  
Raccorder la carte au PC avec un câble USB

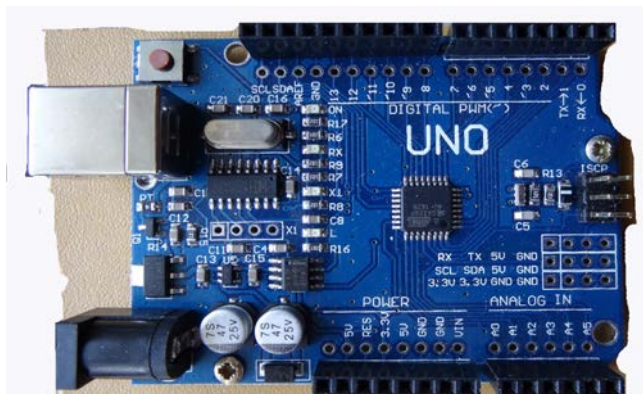
Ouvrir l'IDE « Arduino » et charger le programme,

Transférer le programme sur la carte et ouvrir le lecteur du programme

Faire un essai et éventuellement changer les affectations des variables LDR en fonction de la clarté sur le réseau,

Quand tout va bien, enregistrer le programme sur le PC ou sur une clé USB.

Retirer la liaison carte PC et alimenter votre carte par une l'autre entrée d'alimentation (de 8 à 12 volts redressés).



 **Louis Allaerts**

### Liste des bourses et autres manifestations pour les mois de Mai et Juin 2015.

#### MAI

- 09/5 St Ghislain de 10 à 17 H
- 10/5 Remouchamps (Aywaille) , de 9 à 13H
- 16 et 17/5 Expo-bourse Enghien, de 10 à 18h
- 17/5 Woluwe St Lambert de 7h30 à 13h30
- 23 et 24/5 Portes ouvertes Flawinne (Namur), de 10 à 18h Rue Joseph Gaillard.
- 31/5 Hoeselt de 9 à 13h

#### JUIN

- 7/6 Namur  
place de l'école des Cadets, 4 de 9 à 13h
- 21/6 Woluwe St Lambert

### Souvenirs de vacances.

Le chemin de fer de la Baie de Somme.



Situé dans le nord de la France sur la Côte Picarde, le Chemin de Fer de la Baie de Somme exploite depuis 1971 le Réseau des Bains de Mer créé en 1887.

Tout au long de ses 27 kilomètres de parcours, vous découvrirez la diversité des paysages, de la faune et de la flore d'un environnement bien préservé où le chemin de fer s'intègre en permanence.

Entre terre et mer, la Baie de Somme vous époustoufflera au rythme nonchalant de ses trains historiques à vapeur...



 **Philippe Camus.**