

TP logiciel 1: Pilotage d'une carte I/O sous Qt



Objectifs :

- Utiliser Qt creator
- Développer un programme de mise en oeuvre d'une carte d'Entrées/Sorties piloter par l'interface USB en exploitant les données du constructeur.
- Utilisation de classes personnalisées qui dérivent des classes de base.
- Appel de fonctions spécialisées.
- Exploiter des documentations constructeurs pour la mise oeuvre de systèmes électroniques.

Travail à effectuer :

- Mettre en oeuvre cette nouvelle classe en créant un interface visuel agréable pour vérifier le fonctionnement de la carte P8055.

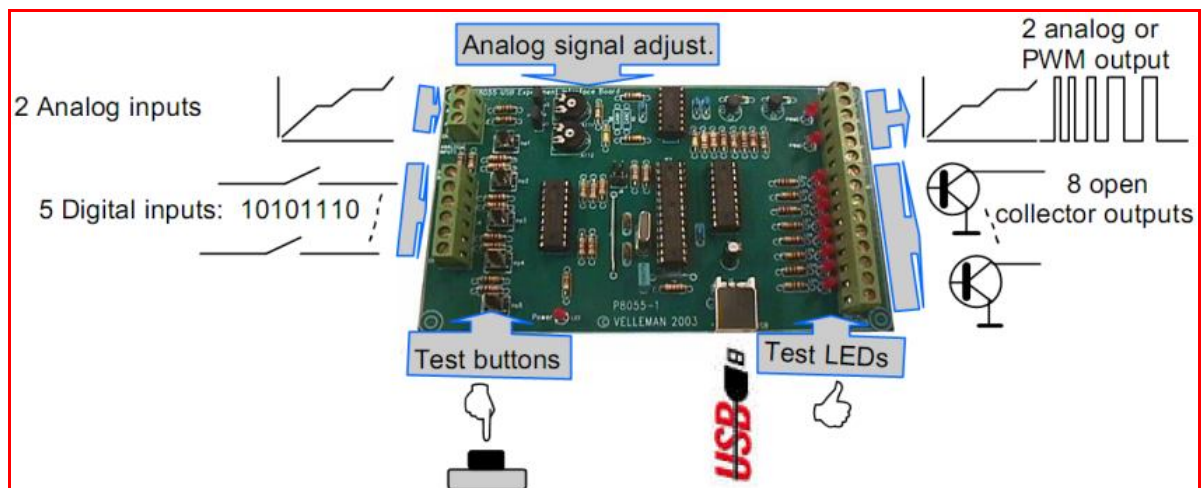


- Créer une application permettant la commande d'un chenillard 8 sorties par l'intermédiaire de la carte USB P8055.

Remarque: Commentez vos fichiers pour une meilleure compréhension et servez-vous au maximum des notices constructeurs pour la mise en oeuvre des éléments électroniques mis à disposition.

Présentation de la carte Interface USB d'experimentation - K8055 :

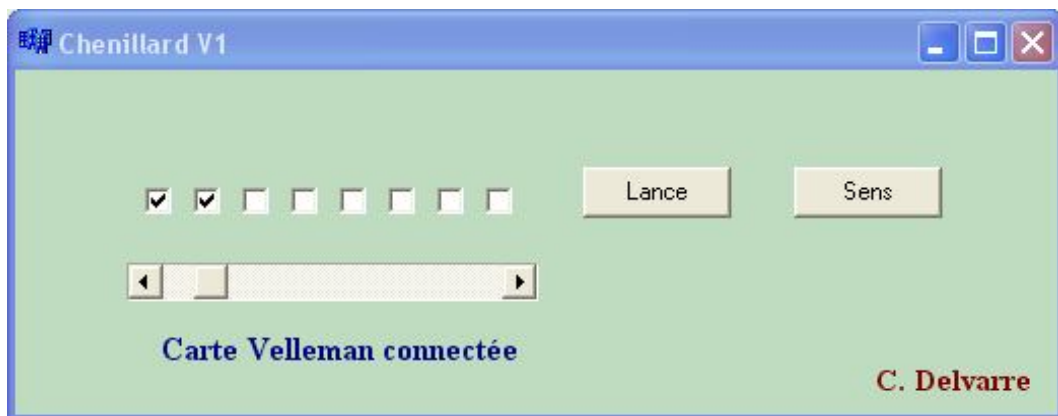
Le module VM110 ou K8055 de Velleman est une carte d'expérimentation USB fournissant des entrées/sorties pilotées depuis un ordinateur connecté à un PC. Toutes les routines de communication sont mémorisées dans une Dynamic Link Library (DLL) contrôlée par un programme Java.



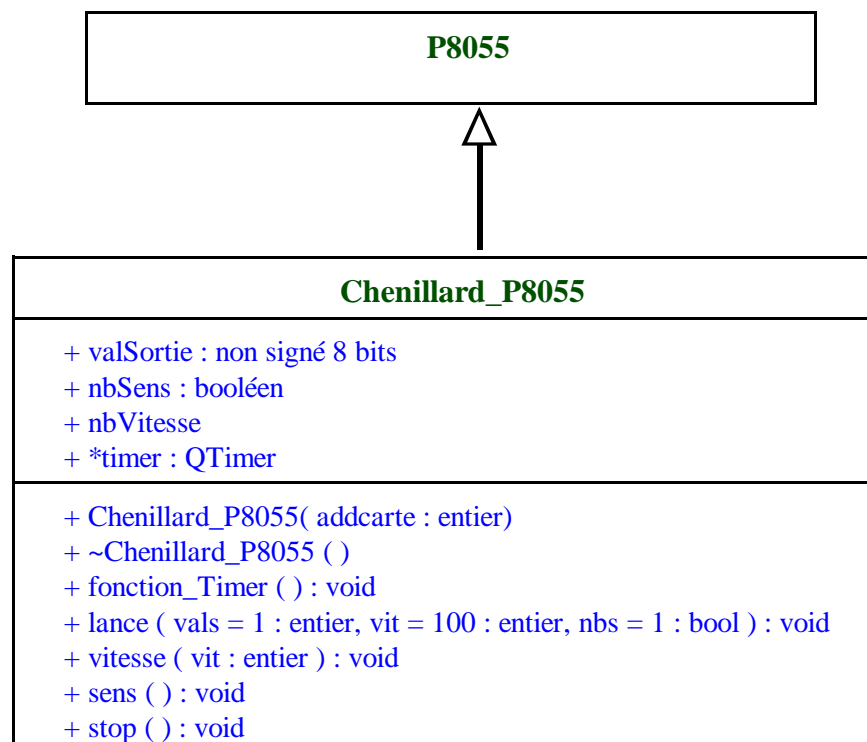
Travail demandé : Développer une Application Chenillard.

Créer une application qui permet de piloter successivement une ou plusieurs sorties de la carte Velleman et décale l'information dans un sens ou dans l'autre au choix de l'utilisateur avec une durée réglable (100 ms à 2 s environ).

Exemple de l'application de commande :



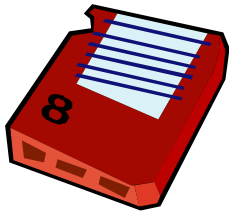
Pour cette application, on instancie un objet à partir d'une nouvelle classe "**chenillard_P8055**" qui dérive de la classe "**P8055**"



- créer sous Qt, la classe **chenillard_P8055** contenant les attributs suivants:

valsortie // valeur des sorties numériques
nbsens // valeur indiquant le sens de décalage
nbvitesse; // valeur vitesse de défilement
***timer ;** // objet **QTimer** définie dynamiquement

ainsi que les méthodes:



```
chenillard_P8055( int addcarte); // constructeur pour initialiser la carte  
~chenillard_P8055 ( ); // destructeur  
void fonction_timer( ); // appel automatique de la fonction Timer tous les “nbvitesse”  
millisecondes qui décale les informations en sortie en fonction du sens choisi.  
void lance(char vals=1, int vit=100, bool nbs=1); // déclenche le chenillard avec des  
paramètres par défaut modifiables (1 lampe, vitesse 100ms et sens croissant)  
void vitesse(int vit); // fixe en milliseconde la vitesse de décalage.  
void sens( ); // modifie le sens de décalage ( croissant ou décroissant)  
void stop ( ); // arrête le décalage automatique des sorties
```

puis développer votre nouvelle application pour obtenir un résultat identique à l'exemple.

