Durée: 8h

TP logiciel 14 :
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 :

- Etablir une nouvelle classe CarteP8055 avec des méthodes permettant des commandes simplifiées pour l'exploitation de la carte électronique d'entrées/sorties P8055 à partir des fichiers fournis par le constructeur.
- 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.

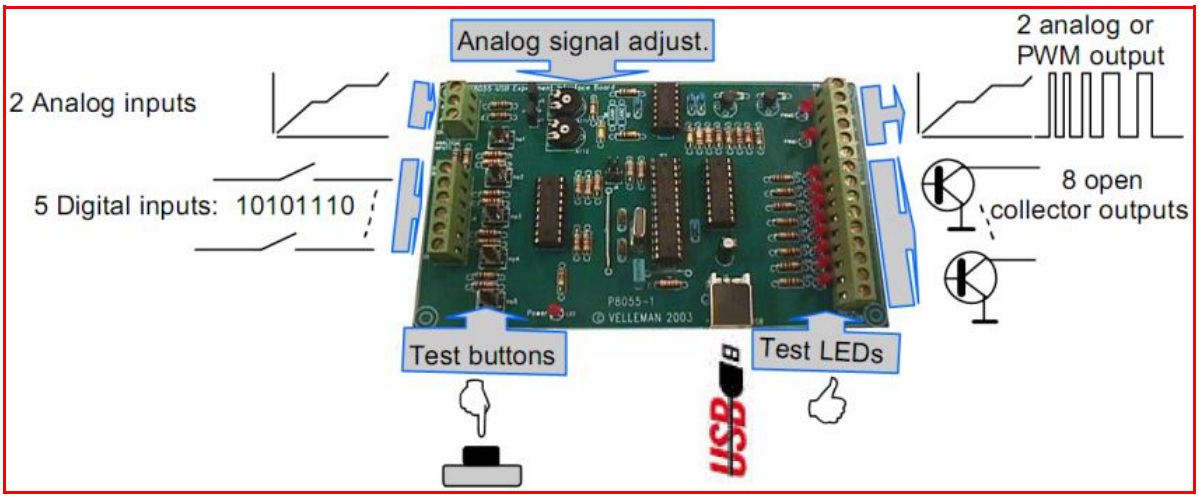
Remarque: Commentez vos fichiers pour une meilleure compréhension et servez-vous au maximum des notices constructeurs pour la mise en oeuvre des éléments électroniques mis à disposition.

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.

Elle possède :

- 5 entrées numériques TTL 5V (0 = terre, 1= ouvert) (l'appareil est pourvu de boutons de test : bouton relâché donne un niveau 1- bouton appuyé donne un niveau 0)
- 2 entrées analogiques (convertisseur 8 bits)
- 8 sorties numériques à collecteur ouvert (max 50V/100mA) (indication LED: Led allumé donne un niveau 0, Led éteinte donne un niveau 1)
- 2 sorties analogiques (8 bits également) avec conversion analogique et PWM
  0 à 5V, résistance de sortie 1K5
  PWM: 0 à 100% sorties à collecteur ouvert max 100mA / 40V (indications LED)
- 2 compteurs 16 bits d'impulsions sur entrée numérique (avec anti-rebond réglable)



Travail demandé :

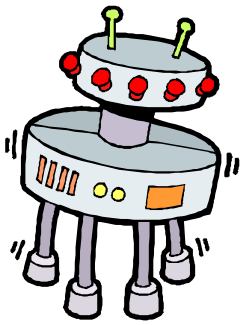
1) A partir de la documentation de la carte Interface USB d'experimentation, créer, sous Qt, la classe **'P8055'** (Fichiers de base pré-remplis : p8055.h et p8055.cpp)

P8055
+ ncarte : entier 8 bits + valSortie : entier non signée 8 bits + valEntree : entier non signée 8 bits
+ P8055 (addcarte : entier) + ~P8055 ( ) + P8055_present ( ) : entier 8 bits + setBit (val : entier non signé 8 bits) : void + clearBit(val : entier non signé 8 bits) : void + valOutput(val : entier non signé 8 bits) : void + getOutput( ) : entier non signé 8 bits + getInput ( ) : entier non signé 8 bits + getBitInput (val : entier non signé 8 bits) : bool

les variables :

ncarte // contenant l'adresse de la carte
valSortie // valeur des sorties numériques
valEntree; // valeur des entrées numériques

les méthodes:

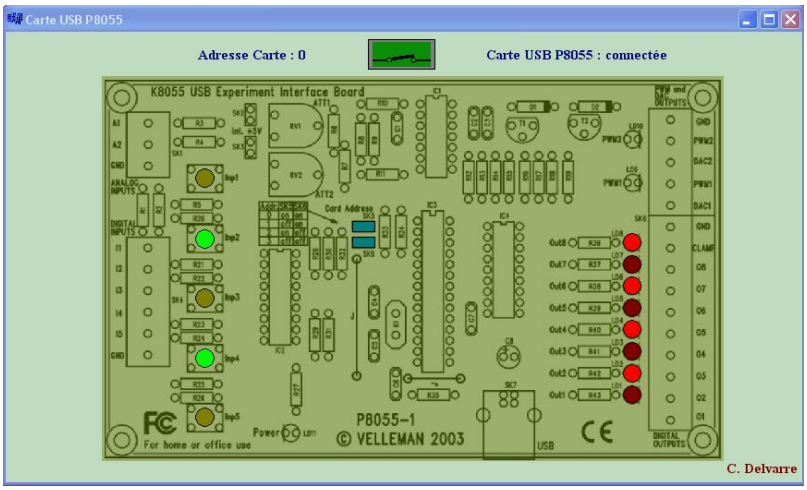


P8055(int addcarte); // constructeur pour initialiser la carte
~P8055( ); // destructeur pour fermer le lien
char P8055_present( ); // renvoie si la carte est connectée
void setBit(usigned char val); // mets un '1' la sortie numérique val
void clearBit(unsigned char val); // mets un '0' la sortie numérique val
void valOutput(unsigned char val); // envoie l'octet val sur les sorties
char getOutput( ) ; // retourne la valeur numérique des sorties
char getInput( ); // retourne la valeur numérique des entrées
bool getBitInput(unsigned char val); //retourne l'état de l'entrée définie par val.

Remarque : le constructeur fournit un fichier K8055D.dll. Cette librairie de lien dynamique (dll) sera chargée dans le constructeur de la classe P8055 à l'aide d'un objet Qlib pour exploiter les routines de communication de la carte K8055.

2) Dans une nouvelle application, vérifier toutes les méthodes de la classe P8055. Déclarer un objet **carte_usb** et visualiser les fonctionnalités de la carte interface Usb Entrées/Sorties numériques.

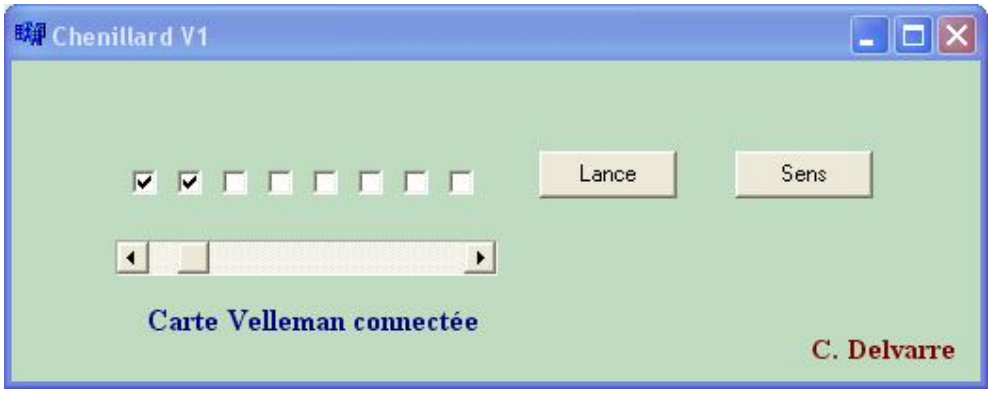
L'application visuelle est composée d'objets Qmenu, QLabel, etc... Vous pouvez développer votre application comme l'exemple proposé :



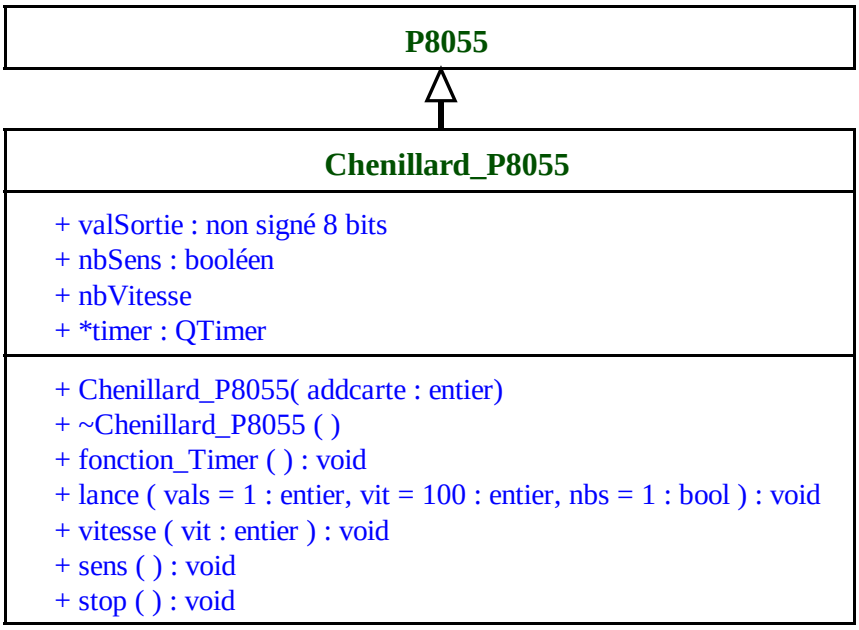
3) Application Chenillard :

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).

<u>Exemple de l'application de commande</u> :



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.