



## **TP 2 - Pilotage d'un afficheur LCD à partir d'une carte Usb I/O**

### **Objectifs :**

- Analyser la notice constructeur pour mettre en oeuvre le composants intelligent: afficheur LCD 2 lignes, 16 caractères.
- Concevoir une nouvelle classe ( LCD8055 ) propre au matériel et dérivée de la P8055 de la carte usb Velleman.
- Développer un programme à l'aide d'un objet conçu par la nouvelle classe spécialisée.
- Mettre en oeuvre un programme test sous l'environnement Qt creator.

### **But :**

- Piloter un module d'affichage alphanumérique à cristaux liquides en mode 4 bits par l'intermédiaire du module K8055 de Velleman connecté sur port USB du PC destiné à envoyer des messages à un afficheur LCD 2 lignes.

### **Critères d'évaluation:**

#### **Vous serez évalués sur les critères suivants:**

- \* Votre autonomie à résoudre les différents problèmes énoncés.
- \* Votre capacité à mettre en oeuvre le travail étudié.
- \* Votre capacité à utiliser le matériel mis à votre disposition.
- \* La rédaction de l'algorithme et du programme de mise en oeuvre.

### **Travail demandé**

Développer un programme en C++ qui permet d'afficher sur la première ligne du module à cristaux liquides le texte: **Ts SNIR 2** , sur la deuxième ligne du module la date, puis prévoir un bouton pour effacer le contenu de l'afficheur et un autre bouton qui transfère le contenu d'un message issu d'un objet TEdit.

#### **Afficheur alphanumérique à cristaux liquides:**

**notice constructeur: LCD\_2x16xfr.pdf et LCD\_2x16.pdf**

La partie visible d'un module d'affichage à cristaux liquides n'est seulement qu'un des composants utilisés pour le réaliser. La forme des caractères affichés a besoin d'être définie (générateur de caractères), le séquençement et la position de l'affichage doivent être contrôlés (base de temps) et les changements dans l'affichage doivent être mémorisés dans le module (Ram). Ce type de module à cristaux liquides a tous les circuits de contrôle et d'affichage réalisés dans un seul circuit intégré. Il peut fonctionner sous le contrôle d'un microprocesseur 4 bits ou 8 bits, pour afficher des caractères alphanumériques, des symboles et d'autres signes. Le module fournit à l'utilisateur un panneau d'affichage de caractères en matrices à point nécessitant un circuit d'interface simple.

### 1) Etude de la notice constructeur:

- a) Donner les principales caractéristiques de ce module (format, Alimentation, type de liaison, etc..:).
- b) Donner et analyser les différentes broches de commande.
- c) Relever les temps caractéristiques minimales pour la transmission de données.
- d) Déterminer les codes d'initialisation à envoyer au module pour effectuer:
  - une mise en service : transmission 4 bits, 2 lignes d'affichage, incrémentation à droite du curseur.
  - un effacement total
  - l'affichage des lettres **Ts SNIR 2**

### 2) Montage:

L'afficheur est câblé sur une carte d'essai, la résistance ajustable de 10Kohm permet l'ajustement du contraste.

L'ensemble est relié à la carte Velleman P8055 tel que :

<b>Afficheur LCD</b> <b>patte</b>	<b>VSS</b> <b>1</b>	<b>RS</b> <b>4</b>	<b>E</b> <b>6</b>	<b>D4</b> <b>11</b>	<b>D5</b> <b>12</b>	<b>D6</b> <b>13</b>	<b>D7</b> <b>14</b>
<b>Vellman K8055</b>	<b>Gnd</b>	<b>Q5</b>	<b>Q6</b>	<b>Q1</b>	<b>Q2</b>	<b>Q3</b>	<b>Q4</b>

- Alimenter le montage de l'afficheur LCD à une alimentation stabilisée +9V

### 3) Programme:

- a) Créer une nouvelle classe "**LCD8055**" qui dérive de la classe "**P8055**".

Pour cela , créer un projet sous QT, ajouter les fichiers .h et .cpp de la classe P8055 puis créer une nouvelle classe "**LCD8055**" qui dérive de **P8055** contenant les méthodes suivantes:

- en mode private :

**void valid\_E ( );** // fait passer le signal E à 1 puis le remet à 0 pour une validation de l'afficheur.

**void Envoie\_Octet( char octet, bool rs);** // envoie l'octet en 2 fois 4 bits avec le signal de validation E en tenant compte de l'entrée RS.

- en mode public:

**LCD8055(int addcarte);** // constructeur qui passe les paramètres au constructeur de la classe mère et qui initialise l'afficheur dans le mode 4bits, 2 lignes.

**~LCD8055 ( );** // destructeur

**void init\_LCD ( );** // initialise l'afficheur dans le mode 4bits, 2 lignes et effectue un effacement écran.

**void aff\_car (char val) ;** // affiche le caractère à l'emplacement du curseur de l'afficheur.

**void clear\_aff ( ) ;** // effectue un effacement de l'afficheur.

**void aff\_texte (char \*val) ;** // affiche le texte à partir de l'emplacement du curseur de l'afficheur.

**void pos\_curseur (bool ligne,int colonne);** positionne le curseur de l'afficheur en fonction des paramètres ligne (0:1<sup>er</sup> ligne - 1:2<sup>ème</sup> ligne) et colonne (0 à 15).

- b) Créer une application sous QT qui permet de tester toutes les méthodes de la classe "LCD8055".

