

Activité : mise en oeuvre d'un périphérique USB sous Linux

Thierry Vaira <tvaira@free.fr>

Table des matières

Activité : mise en oeuvre d'un périphérique USB sous Linux	1
Cahier des charges	1
Pré-requis	1
Objectifs	1
Exigences	2
Ressources documentaires	2
Voir aussi	2
Ressources d'environnement	2
Matériels disponibles	3
Compétences et Savoir-faire	3

Site : tvaira.free.fr

Activité : mise en oeuvre d'un périphérique USB sous Linux

Cahier des charges

Vous intervenez en équipe en tant que technicien IR dans une entreprise spécialisée dans les systèmes numériques. Vous venez de recevoir le périphérique USB correspondant à vos besoins et vous devez :

- identifier les informations techniques nécessaire à sa prise en charge
- intégrer le nouveau matériel au système
- rendre opérationnel sa mise en oeuvre
- réaliser la recette de son installation
- développer un module logiciel minimal et le documenter
- rendre la solution opérationnelle et la documenter

Le module logiciel sera développé en équipe et permettra l'utilisation de **Doxygen** (un logiciel de génération de documentation automatique).

Pré-requis

- connaissance des commandes de base du système GNU/Linux
- connaissances en programmation C/C++ et en programmation orienté objet (POO)

Objectifs

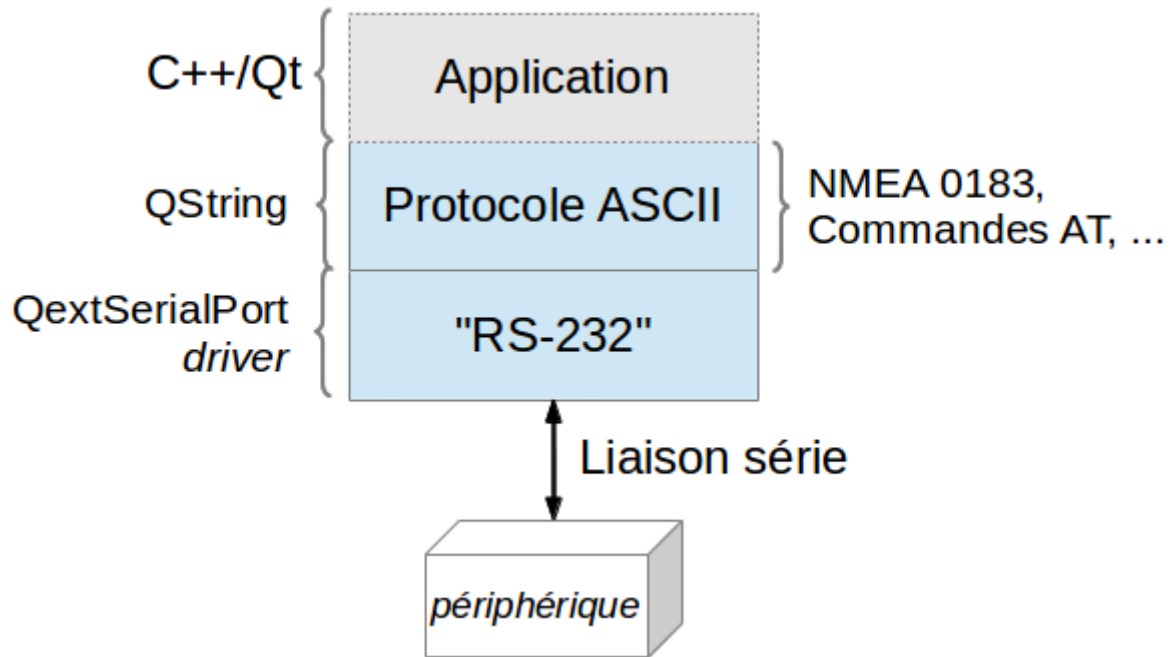
Vous devez développer un module logiciel (Application) qui soit capable de communiquer avec un périphérique via une liaison USB.

Ces périphériques utilisent en réalité une transmission série avec un convertisseur USB <-> RS-232 (les circuits les plus répandus sont pl2303, FTDI FT232, ...). Certains adaptateurs ajoutent un circuit MAX232 pour mettre en forme des signaux conformes au standard RS-232. Au final, ils sont "vus" logiquement comme des ports séries virtuels.

La prise en charge du périphérique est assurée par le système d'exploitation via un pilote de périphérique (*driver*). Au niveau programmation, vous utiliserez les services d'une classe Qt (`QextSerialPort` pour Qt4 ou `QSerialPort` pour Qt5) pour communiquer avec le périphérique.

Ce type de périphérique utilise un protocole de communication de type ASCII pour échanger des trames. Il existe des protocoles propriétaires et/ou standardisés. Certains sont très répandus : NMEA 0183 (GPS, station météo, ...), commandes AT (modem, XBee , Bluetooth, ...).

Pour décoder les trames (vérification, extraction des données, ...), on utilisera les services de la classe `QString` de Qt.



Exigences

- équipe de 3 étudiants maximum
- assurer un compte-rendu oral et être capable de répondre à des questions techniques
- livrer un programme de test écrit en C++ qui recevra en argument le nom du fichier spécial de périphérique (sinon on utilisera le nom par défaut) et qui utilisera les services de la classe `QextSerialPort` pour gérer le périphérique USB à votre charge. On affichera dans la console une réception de données.
- livrer une application qui possèdera au moins 2 classes. Pour les fonctionnalités, voir avec le client.
- livrer une documentation (au format PDF et HTML) du programme et qui intégrera aussi un manuel d'installation et une fiche de recette complétée
- utilisation de Subversion obligatoire

Ressources documentaires

- [Mise en oeuvre d'un périphérique USB sous Linux](#)
- [Aide à la mise en oeuvre d'un périphérique USB sous Linux](#)
- [Mise en oeuvre d'un port série sous Qt](#)
- [Documentation du code](#)
- [La norme NMEA 0183](#)
- [Les commandes AT](#)
- [Conversions de coordonnées GPS](#)
- [Questionnaire](#)

Voir aussi

- [Mise en oeuvre du panneau lumineux Mc Crypte 590996](#)
- [Mise en oeuvre d'un périphérique USB sous Windows](#)

Ressources d'environnement

- Système d'exploitation : GNU/Linux (Ubuntu 12.04.5 LTS)
- Environnement de développement : Geany 1.24.1

- Compilateur : GNU g++/gcc version 4.6.3
- Chaîne de fabrication : qmake 2.01a et GNU make 3.81
- API : Qt 4.8.1
- Bibliothèque : [QextSerialPort](#).

Matériels disponibles

1. [Columbus V-800 GPS Receiver](#)
2. [Adafruit Ultimate GPS](#)
3. [Module GPS Libelium](#)
4. [CV7-V-USB](#)
5. [SIM5218](#)

Compétences et Savoir-faire

- A1. Rechercher et/ou exploiter des documents ...
 - T1.2 Extraire et synthétiser l'information pertinente.
 - C1.1 Rechercher et structurer des informations techniques.
 - SF01. Déterminer les critères de recherche et identifier les informations disponibles.
- A7. Réaliser ou mettre en oeuvre et valider une solution.
 - T7.2 Produire un prototype logiciel et/ou matériel.
 - C4.1 Câbler et/ou intégrer un matériel.
 - SF35. Réaliser la mise en situation et interconnecter du matériel.
 - C4.2 Adapter et/ou configurer un matériel.
 - SF36. Rendre opérationnel un matériel.
 - C4.3 Installer et configurer une chaîne de développement
 - SF38. Fournir un environnement de développement opérationnel.
 - C4.4 Développer un module logiciel.
 - SF42. S'assurer de la conformité du logiciel avec le cahier des charges.
 - T7.3 Valider le prototype.
 - C4.5 Tester et valider un module logiciel et matériel.
 - SF44. Faire la recette d'un module.
 - C4.6 : intégrer un module logiciel
 - SF46. Rendre opérationnel le système ou le sous-système.
- A9. Elaborer et/ou mettre en oeuvre le processus de réception ...
 - T9.2 Installer un système ou un service.
 - C5.1 Préparer la solution et le plan d'action.
 - SF49. Le système est opérationnel et documenté.
 - C5.2 Mettre en oeuvre une solution matérielle/logicielle en situation.
 - SF50. Rendre la solution opérationnelle et la documenter.