



Q1 - Interfaces graphiques embarquées avec Qt

Interfaces graphiques embarquées avec Qt

Objectifs

- Installer Qt 5 et les composants nécessaires sur une plateforme embarquée
- Écrire des applications Qt
- Découvrir les composants clés de Qt
- Comprendre le bon usage des threads dans les applications Qt

Environnement du cours

- Support de cours imprimé (en anglais)
- Un PC Linux pour deux stagiaires.
- Une plateforme cible (i.MX6 Sabre ou PandaBoard ES) pour deux stagiaires

Prérequis

- Bonne connaissance du langage C++

Audience visée

- Tout ingénieur ou technicien en systèmes embarqués possédant les prérequis ci-dessus.

Plan du cours

Premier jour

Introduction

- Historique
- Versions
- Licences
- Composants
- Qt Quick

Les bases de Qt

- Application Hello world
- Compiler et compiler de façon croisée une application Qt
- Les mécanismes principaux de Qt
 - MOC (Meta Object Compiler)
 - Boucle principale
 - Signaux et slots
 - Introspection
 - Appels asynchrones
- Création d'une application Qt sous Linux
 - Création de projets Qt
 - Compilation
- Classes principales

- Classes de base
- Principaux widgets
- Utilitaires
- Les outils de développement pour Qt
 - QT Designer
 - QtCreator
 - Qmake

Les widgets Qt

- Widgets de base
 - Labels, boutons, ...
- Layouts
- Boîtes de dialogue
 - Boîtes de dialogue personnalisées
 - Boîtes de dialogue standard

Deuxième jour

Programmation multithread

- Modèle de threads
- Lancer un thread de travail
- Synchronisation
- Confier du travail au thread graphique
- Timers

Widget personnalisé

- Dessin 2D
- Traitement des événements souris, écran tactile et clavier

Modèle/vue

- Le concept modèle/vue
- Widget modèle/vue et widget standard
- Modèles standard
- Écrire un modèle personnalisé
- Les vues

Troisième jour

Installation

- Graphisme de bas niveau
 - Frame buffer
 - Open GL ES et EGL
 - Serveur X
 - Wayland
- Plugins de plateforme Qt
 - EGLFS
 - LinuxFB
 - XCB (serveur X)
 - Wayland
- Sous-système d'entrée de bas niveau
 - Drivers d'entrée
 - Tslib

- Protocole multi-touch
- Configurer les entrées dans Qt
- Compiler de façon croisée et installer Qt
 - Système de build
 - Principales options

Open GL ES 1.1 et 2.0

- Présentation d'Open GL
 - Les différentes versions d'Open GL
- Concepts de base
 - La notion d'état dans OpenGL
 - Vertices et triangles
 - Transformations
 - Dessin
 - Textures
 - Shaders
- OpenGL et OpenGL/ES
- Dessiner en OpenGL
 - Tableaux de vertices et d'indices
 - Dessiner des éléments
 - Projections
 - Points de vue
- Matrices de transformation
 - Chargement et initialisation
 - Translations et rotations
 - Sauvegarde et restauration (pile de matrices)
- Les shaders OpenGL
 - OpenGL Shading Language (GLSL)
 - Vertex shaders
 - Fragment shaders
 - Appliquer des transformations
- Le modèle de rendu OpenGL
 - Surfaces
 - Opérateurs de rendu
 - Rendu hors écran
- OpenGL dans Qt 5
 - OpenGL brut avec le module Qt OpenGL (classe QGLWidget)
 - Encapsulations Qt au-dessus d'OpenGL ES du module Qt GUI (QOpenGLContext)

Quatrième jour

Support multimédia

- Le multimédia sous Linux
 - Gstreamer
- Le multimédia dans Qt 5
 - Module QtMultimedia
 - Audio, Video Camera, MediaPlayer et Radio

Qt Quick

- QML
 - Éléments et propriétés
 - Javascript
- Principaux éléments et propriétés
 - Saisie utilisateur, état, modèles et vues, ...

- Interactions entre C++ et QML