

## FORMATION QNX les essentiels

Programmation temps réel avec QNX Neutrino et prise en main des outils “QNX Momentics IDE”, introduction à la programmation matérielle, traitement multicœurs et partitionnement adaptatif.

**DUREE:** 32 heures (4.5 Jours).

### OBJECTIFS

L'objectif de ce cours est d'inculquer aux participants les concepts de la programmation d'application temps réel sous QNX Neutrino et d'apprendre à utiliser l'environnement de développement intégré QNX Momentics pour développer des applications, accéder au matériel, déboguer des problèmes complexes et analyser et optimiser les performances des systèmes.

A l'issue de la formation, les participants seront capables d'appliquer leur nouveau savoir-faire pour la recherche et la résolution de problèmes dans leurs applications QNX, qu'il s'agisse de dysfonctionnements ou de problèmes de performance ou d'accès du matériel.

Ce cours est un condensé de modules tirés de 3 formations différentes de QNX (Programmation temps réel QNX Neutrino, Déboguer analyser les performances avec l'IDE Momentics , Développement d'un ressource manager / pilote sous QNX Neutrino).

La formation comprend de nombreux exercices pratiques.

### PREREQUIS

Obligatoire : Connaissance du langage de programmation C. Ordinateur portable.

Souhaitable : Notions d'utilisation d'outil IDE (« Integrated Development Environment ») ou d'atelier logiciel de développement similaire, de développement croisé, programmation pour environnement temps réel.

### COMPETENCES PROFESSIONNELLES VISEES

Développer une application QNX et respecter les méthodes de programmation, des synchronisations Inter-processus (IPC), du « *message passing* » de QNX pour l'utilisation dans un environnement temps réels. Comprendre l'ordonnanceur du RTOS (« *Realtime OS* ») de QNX. Editer et compiler des codes et déboguer une application C à l'aide de divers outils et techniques dans l'environnement IDE (« *Integrated Development Environment* »). Déterminer la cause des problèmes de performance d'une application à l'aide de divers outils et techniques de trace. Déterminer la cause des problèmes de performance d'un système multiprocessus (multithread) et multicœurs. Savoir utiliser le bon outil de débogage ou de profilage selon le type précis de problème. Apprendre à développer un gestionnaire de ressource / pilote pour programmer le matériel et ses horloges et créer une image QNX.

## PROGRAMME

- **Architecture du système d'exploitation temps réel QNX Neutrino :**
- **Prise en main de l'Environnement de Développement Intégré (IDE).**
- **Synchronisation de processus et de « Threads ».**
- **Communication interprocessus (IPC).**
- **Minuteur, horloge et temps échus.**
- **Introduction à la programmation du matériel.**
- **Introduction aux gestionnaires de ressources**
- **Profiler une application.**
- **Couverture de code**
- **Débogue de problèmes mémoire.**
- **Profiler le Système.**
- **Construction d'une image QNX Neutrino, amorcer un démarrage.**
- **Traitement Multiprocesseurs Symétrique (SMP) et lié à un CPU (BMP)**
- **Partitionnement Adaptatif (APS).**

### PROGRAMME QNX « Essentials »

#### ***Architecture du système d'exploitation temps réel QNX Neutrino :***

Revue des fonctionnalités du micronoyau QNX Neutrino et de son gestionnaire de processus, détail de l'adressage et de la protection mémoire, notion de processus et de « threads », du séquenceur et de sa gestion du temps. Communication interprocessus et synchronisations, gestionnaire de ressources et notions d'objets partagés.

#### ***Prise en main de l'Environnement de Développement Intégré (IDE) :***

Introduction à « QNX Momentics » pour QNX Neutrino, navigation, configuration et personnalisation de l'IDE. Notions de gestion de projet, d'espace de travail, connexions aux cartes cibles. Exercice d'édition de code, de construction d'application et de compilation. Exécution et débogue sur une cible, point d'arrêt et examen des données.

#### ***Synchronisation de processus et de « Threads » :***

Définition d'un thread, ses priorités, ses algorithmes de séquençement, lancement et arrêt d'un thread, synchronisation et accès aux ressources, fonctions POSIX.

#### ***Communication interprocessus (IPC).***

Introduction aux méthodes de communication et de synchronisation entre processus, échange d'informations et contrôle : utilisation des messages synchrones. et asynchrone, des impulsions, et de la mémoire partagée.

#### ***Minuteur, horloge et temps échus.***

Notion de déterminisme temporel, introduction à la gestion des horloges et du temps (horloge périodique, tic d'horloge, marquage de temps, temps échus et performance), influence de l'architecture matérielle.

#### ***Introduction à la programmation du matériel.***

Méthodes d'accès au matériel, adressage des E/S et de leurs mémoires, gestion des DMA et des interruptions.

**Introduction aux gestionnaires de ressources**

Notions de gestionnaire de ressources : résolution de noms de chemin, format des messages IPC et structure générale d'un gestionnaire de ressource.

**Profiler une application.**

Utilisation de l'IDE pour profiler un processus simple, analyse de la consommation de temps CPU, identification des fonctions chronophages et optimisation du code, utilisation du profileur en mode post-mortem.

**Couverture de code**

Déterminer l'efficacité d'une suite de tests nécessite du travail utilisation des outils de l'IDE pour déterminer le code source réellement exécuté par le test.

**Débogue de problèmes mémoire.**

Introduction aux problèmes de corruption mémoire, de fuites et de consommation de mémoire excessive. Prise en main des outils d'analyse fournis par l'IDE.

**Profiler le Système.**

Introduction au profileur du système, configuration de l'outil, création et analyse des « logs » à travers l'IDE, affichage via le profileur, insertion de traces/données dans un log.

**Construction d'une image QNX Neutrino, amorcer un démarrage.**

Configuration et amorçage, du micro-noyau, puis du réseau, du gestionnaire de fichiers, des pilotes, des utilitaires et de l'applicatif. Génération d'une image du RTOS avec démarrage automatique. Introduction aux différentes méthodes de boot des systèmes embarqués.

**Traitement Multiprocesseurs Symétrique (SMP) et lié (BMP)**

Utilisation de l'IDE pour profiler un processus simple, analyse de la consommation de temps CPU, identification des fonctions chronophages et optimisation du code, utilisation du profileur en mode post-mortem. Notions et mesure des migrations de thread d'un cœur à l'autre, notion d'affinité Thread-cœur. Notion de programmation multicœurs.

**Partitionnement Adaptatif.**

Utilisation du partitionnement adaptatif sous QNX neutrino : notions et mise en œuvre, Caractéristiques du « thread scheduler », protection des threads critiques. Considération d'architecture, précaution d'utilisation avec les systèmes multicœurs. Utilisation des outils Momentics (avec le noyau instrumenté) pour l'optimisation des budgets CPU.