

Programmation temps réel pour le système d'exploitation (RTOS) QNX Neutrino®.

Pré-requis.

Le participant doit avoir une bonne connaissance de la programmation en langage C et des connaissances générales sur les systèmes d'exploitation. Une connaissance des systèmes embarqués ou des concepts de programmation temps réel est un atout.

Objectifs.

Le cours s'adresse à tous ceux qui développent des applications pour le système d'exploitation temps réel (RTOS – Real Time Operating System) QNX Neutrino. A la fin du cours, le participant aura acquis une connaissance des concepts réels de programmation d'applications pour QNX Neutrino avec de nombreux exemples pratiques écrits de programmes.

Les objectifs sont :

1. Familiariser le développeur avec les fonctionnalités, services et architecture du RTOS QNX Neutrino.
2. Appréhender et régler les cas concrets de problèmes liés aux développements d'applications temps réels et utiliser les techniques appropriées pour les résoudre.
3. Réaliser des exercices pratiques afin d'appliquer les concepts introduits durant le cours.

Thèmes.

Le cours est constitué des sections suivantes :

Architecture du système d'exploitation temps réel (RTOS) QNX Neutrino.

Le cours commence par une vue d'ensemble de QNX Neutrino qui met en avant les fonctionnalités clés, ses services et son architecture. Cette introduction sert de fils conducteur, feuille de route pour le reste du cours et permet de donner les bases des services que le RTOS fournit.

Environnement de Développement Intégré (IDE), fonctions de base.

Donne une vue d'ensemble nécessaire au programmeur qui utilise l'atelier logiciel de développement « *QNX Momentics* » pour QNX Neutrino et fournit le concept pour naviguer, configurer et personnaliser l'IDE selon les besoins et les préférences. Cette section met en œuvre aussi la gestion de projet, l'espace de travail et les modèles de connexions entre l'IDE et les cartes cibles et introduit les différentes façons de s'interfacer avec les cartes cibles.

Compilation et mise au point.

Cette section propose une introduction rapide aux fonctions d'édition de code, de compilation de code, d'exécution du code et de mise au point à partir de l'IDE qui permettra au participant de réaliser les exercices pratiques, tout au long de la formation.

Synchronisation de processus et de « Threads ».

De façon différente à la plupart des RTOS du marché, QNX Neutrino exécute les programmes dans des espaces mémoires protégés appelés processus. Cette section adresse comment un problème de design peut être reparti avec des processus séparés et comment les processus peuvent être démarrés et arrêtés. Elle définit aussi la notion de Thread, comment le démarrer, l'arrêter et comment synchroniser des accès à des ressources à travers les appels système POSIX de QNX Neutrino. Des exercices pratiques forment une partie importante de cette section.

Communication inter-processus (IPC).

Explorer les différentes méthodes que QNX Neutrino propose pour permettre aux processus d'échanger des informations et de les contrôler. L'accent est mis sur le passage de messages qui est un principe fondamentale de l'architecture de QNX Neutrino. Cette section examine aussi les autres méthodes de communication, telles que les impulsions, la mémoire partagée et les messages asynchrones.

Comparaison des méthodes IPC.

Rapide comparaison des avantages et désavantages à utiliser les différentes méthodes supportée par QNX et celles nécessaires aux besoins du système du participant.

Introduction à la programmation du matériel.

Introduction aux méthodes d'accès du matériel, incluant l'adressage des E/S (entrées/sorties) et de leurs mémoires, la gestion de la DMA et des interruptions.

Minuteur, horloge et temps échus.

Un système temps réel nécessite un séquenceur précis et rapide pour gérer des évènements périodiques. Cette section met en œuvre les fonctions du RTOS qui gèrent le(s) minuteur(s) temps réel, et les temps échus du noyau. Un exercice pratique illustre comment générer et gérer des évènements périodiques.

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

Cette section aborde comment réaliser une image avec les composants du système d'exploitation QNX, tels que le micro-noyau, le réseau, le gestionnaire de fichiers, les pilotes du matériel, les utilitaires et programmes, afin de produire une image du RTOS pour démarrer un système automatiquement. Elle montre aussi les différentes manières d'enregistrer l'image dans le système embarqué.

Introduction aux gestionnaires de ressources

Chaque processus peut être un gestionnaire de ressources, et les pilotes en sont généralement. Cette section montre les possibilités des gestionnaires de ressources, incluant les résolutions de nom de chemin, les formats des messages IPC et la structure générale d'un gestionnaire de ressource. Plusieurs exercices pratiques mettent en œuvre comment initialiser un gestionnaire de ressource et gérer les opérations d'écritures et de lectures.