Embedded GUIs with Qt

Objectives

- Installing Qt 5 and the needed components on an embedded platform
- Writing Qt applications
- Discovering Qt key components
- Understanding the proper use of threads in Qt applications

Labs are conducted on target boards, that can be:

Quad Cortex/A9-based "Sabre" boards from NXP, with Lauterbach JTAG probes.
Dual Cortex/A9-based "Panda ES" boards from Texas Instruments, with Lauterbach JTAG probes.
Atmel ARM9-based boards, with Lauterbach JTAG probes.

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Prerequisite

- Good knowledge of C++ language

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.

- In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

FIRST DAY

Introduction

- History
- Versions
- Licenses
- Components
- Qt Quick

Qt basics

- Hello world application
- Compiling and cross-compiling a Qt application
- The main mechanisms of Qt
  - MOC (Meta Object Compiler)
  - Main loop
  - Signals and slots
  - Introspection
  - Asynchronous calls
- Creation of a Qt application on Linux
  - Creation of Qt projects
  - Compiling
- Main classes
  - Base classes
  - Main widgets
  - Utilities
- The development tools for Qt
  - QT Designer
  - QtCreator
  - Qmake

Exercise: *Hello world application*

Qt widgets

- Basic widgets
  - Labels, buttons, …
- Layouts
- Dialogs
  - Custom dialogs
  - Standard dialogs

Exercise: *Writing an application combining various widgets*
Exercise: *Writing a custom dialog*

SECOND DAY

Threading

- Threading model
- Launching a worker thread
- Synchronization

- Queuing work to the GUI thread
- Timers

Exercise: *Writing a multi-threaded application*

Custom widget

- 2D drawing
- Handling mouse, touch screen and keyboard events

Exercise: *Moving an image on screen*

Model/view

- Model/View concept
- Model/View widget vs Standard Widget
- Standard models
- Writing a custom model
- Views

Exercise: *Using a QListView*

THIRD DAY

Install

- Low-level graphism
  - Frame buffer
  - Open GL ES and EGL
  - X Server
  - Wayland
- Qt platform plugins
  - EGLFS
  - LinuxFB
  - XCB (X server)
  - Wayland
- Low-level input subsystem
  - Input drivers
  - Tslib
  - Multi-touch protocol
- Configuring input in Qt
- Cross-compiling and installing Qt
  - Build system
  - Main options

Exercise: *cross-compiling and installing Qt5 on an embedded board*

Open GL ES 1.1 and 2.0

- Presentation of Open GL
  - Various Open GL versions
- Base concepts
  - Notion of state in OpenGL
  - Vertices and Triangles
  - Transformations
  - Drawing
  - Textures
  - Shaders
- OpenGL and OpenGL/ES
- Drawing in OpenGL
  - Vertices and index arrays
  - Drawing items
  - Projections

    - Viewpoints
- Transformation matrixes
    - Loading and initialization
    - Translations and rotations
    - Save and restore (Matrix stack)
- OpenGL shaders
    - OpenGL Shading Language (GLSL)
    - Vertex shaders
    - Fragment shaders
    - Applying transformations
- OpenGL rendering model
    - Surfaces
    - Rendering operators
    - Offscreen rendering
- OpenGL in Qt 5
    - Raw Opengl with Qt OpenGL module (QGLWidget class)
    - Qt wrappers above OpenGL ES of Qt GUI module (QOpenGLContext)

Exercise: *Rotating cube*

FOURTH DAY

Multimedia support

- Multimedia in Linux
    - Gstreamer
- Multimedia in Qt 5
    - QtMulimedia module
    - Audio, Video Camera, MediaPlayer and Radio

Exercise: *Playing a video file*

Qt Quick

- QML
    - Elements and properties
    - Javascript
- Main elements and properties
    - User input, state, model and views, …
- Interactions between C++ and QML

Exercise: *Hello world application with Qt Quick*

Renseignements pratiques

Inquiry : 4 days