[redacted]

Through this course, the attendee will become familiar with RVDS compiler, assembler, linker and simulator

Objectives

- Through this course, the attendee will become familiar with RVDS compiler, assembler and linker.
- The course explains the subtleties of the scatter file.
- A lot of tips are provided which contribute to optimize ARM code execution time and / or ARM code compacity.
- Practical exercices have been developed to highlight the theoretical aspects.

Labs are run under RVDS4.0
A more detailed course description is available on request at formation@ac6-formation.com

Prerequisites

- Knowledge of C language.

Course Environment

- Theoretical course
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

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 by quizzes offered at the end of various sections to verify that 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

EMBEDDED SOFTWARE DEVELOPMENT WITH RVDS

- Embedded development process
- Application startup

- Placing code, data, stack and heap in the memory map, scatterloading
- Tailoring the C library to your target
- Reset and initialisation
- Placing a minimal vector table
- Further memory map considerations, 8-byte stack alignment in handlers
- Building and debugging your image
- Long branch veneers

C/C++ COMPILER HINTS AND TIPS

- ARM compiler optimisations, tail-call optimization, inlining of functions
- Mixing C/C++ and assembly
- Coding with ARM compiler
- Measuring stack usage
- Unaligned accesses
- Local and global data issues, alignment of structures
- Further optimisations, linker feedback

Renseignements pratiques

Inquiry : 1 day