Ingenieurbüro für Echtzeitprogrammierung

IEP

RT-Link

The easy way to the binary image

RT-Link

In industrial applications, often it is indispensable to use (E)EPROMs for storage and direct execution of software. RTOS-UH of the shelf provides tools to generate rommable code and to bind this code to ROM-images. For small systems, these tools allow to build ROM-images at justifiable effort. Nevertheless, preparing ROM-images for large systems can be a tedious chore.

ROM code

RT-Link provides an easy way to build ROM-image for large systems with multiple independent modules.

- unique relocation of modules compiled with the CODE=0 and VAR=0 option.
- automatic allocation of RAM- and ROM-space for each module. Binding to fixed ROM- as well as RAM-address is supported.
- building of compact image by gap-free allocation in RAM and ROM. There's no need for mutltiple compilation steps - **RT-Link** relocates the modules.
- manual splitting of S-Records becomes obsolet. **RT-Link** does automatic splitting of S-Records in code- and data area.

Cross Development

With **RT-Link**, building (E)EPROM-images is fast and free of errors, even during cross-development without access to the runtime target. **RT-Link** is available as native RTOS-UH executable as well as cross development tool for all cross development environments. One single symbol definition file (cross-reference file) contains all address information of a target system needed during cross-linking.

automatically

With **RT-Link**, the task to build ROM-images is easily automated and drastically shortened. All necessary object file are specified in a control file, and **RT-Link** takes care for all necessary steps to combine these files into a single ROM-image. **RT-Link** supports splitting of ROM images and allows to store image files either binary or as S-Record for further processing by (E)EPROM programming tools.

Control file

The control file of **RT-Link** contains not only the information about all necessary object files, but also the relevant information of the target configuration. Both automated RAM- and ROM-allocation as well as manual address pinpointing for single objects are supported. Aside from the ROM-image, **RT-Link** generates a map file with complete information about used memory areas and addresses of relocated symbols.

Modular

Changing source code typically leads to changes in the size needed for code or data. When linking manually, this changes have to be considered – either by reserving gaps for growth or by recompiling code to match a changed target address map. With **RT-Link**. this tedious chore belongs to the past – one single run of **RT-Link**, and a new ROM image is available. Especially during testing, turn around times are improved substantially.

Incremental

Readily tested object files can be combined to one single loadable S-Record. The number of files to transfer to the target is reduced. By resolving the internal references of linked modules, **RT-Link** even reduces the size of the files to be transferred.

By incremental linking the reuse of already linked modules is provided.

Indispensable

RT-Link is an indispensable tool for all programmers, who want to take profit of modular programming and have refrained from fine grained modularity because of the tedious chore of integrating modules to complete systems. In conjunction with a MAKE tool, just one command is necessary to build a new ROM image after changing the source. The developers can concentrate themselfs on contents and forget about the repetitive and error prone task of building ROM images.