Microcontroller Networking

for Embedded

Measurement & Control Applications

M C N F

MicroController Networking Framework

- Summary -

Dr.-Ing. Andreas Eppinger

2012-12-17 Rev 1.0.0

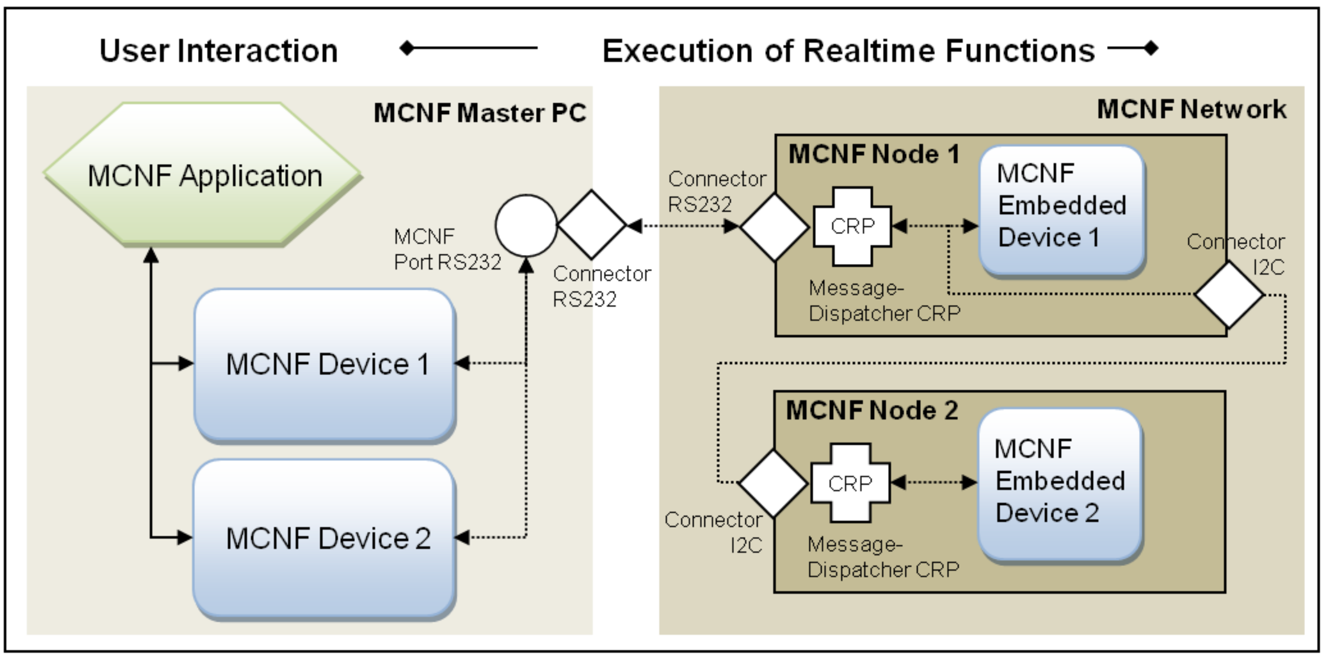
MCNF Summary

Many measurement and control applications combine embedded microprocessor systems with user interfaces running on a PC. The microprocessors execute real time critical functions while the PC acts as a master to control configuration, user interaction, data management and scripting.

The “**M**icro **C**ontroller **N**etworking **F**ramework” (**MCNF**) is a software framework that supports this architecture of a “master”, typically a PC, controlling a number of applications executed in a heterogeneous network. “Heterogeneous” in this context means a network with a mix of different network standards, e.g. TCP/IP, RS232, I^2C or SPI.

Logically, MCNF applications consist of two parts: A user control “Application”and a number of “Devices”. Devices provide a set of functions which they are able to execute. We can think of them as a model and representation of function code implemented in an embedded device and executed on the embedded system.

Networking between these components is a key element: The embedded system is typically built upon a network of microprocessors which is connected to the master (-PC) via communication channels.



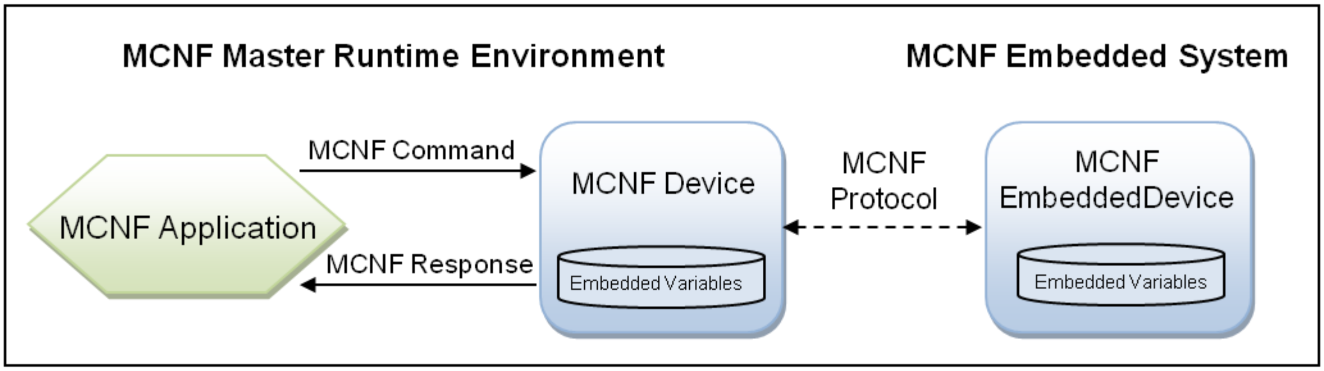
A “MCNF Network” is built from “MCNF Nodes” and its input- / output-“Connectors”. Nodes represent microcontrollers and the connectors are the communication channels. Nodes interact through links between the connectors. The (PC-) master node’s communication channels are called “Ports” and they have connectors as well. In addition to these networking components, the nodes also contain embedded devices providing functions executable on the respective node. Message Dispatchers in the nodes handle the MCNF protocol and relay messages to embedded devices on a node or to gateway connectors.

The basic idea of MCNF is to keep the application logic independent from the network.
This allows applications to be executed on different network configurations without the need to change the application code.

An executable MCNF system configuration maps the application's device models to the "real" embedded devices provided by the network's nodes.

During execution, applications call device functions which create messages, send them over the network to the respective embedded devices, execute the embedded function code and process the responses.

In order to build and execute applications, we have to consider the two basic parts:
The (PC-) master's runtime environment and the embedded system.



MCNF defines a small kernel to be implemented on each node in the embedded system. This kernel software provides communication functions and handles the MCNF protocol. Application specific embedded device functionality has to be defined within the framework provided by the kernel. A set of predefined standard functions can be used to manage application data: Reading and writing, saving to EEPROM and loading from EEPROM of "embedded variables" is supported.

After implementing the MCNF kernel and the application specific embedded device functions, the user would define node-, device- and user controls-objects within the context of the master's MCNF runtime environment which is based on the .NET platform. The devices provide the defined functionality to the MCNF applications via an asynchronous command - response interface. Pre-defined standard commands and responses handle the manipulation of embedded variables.

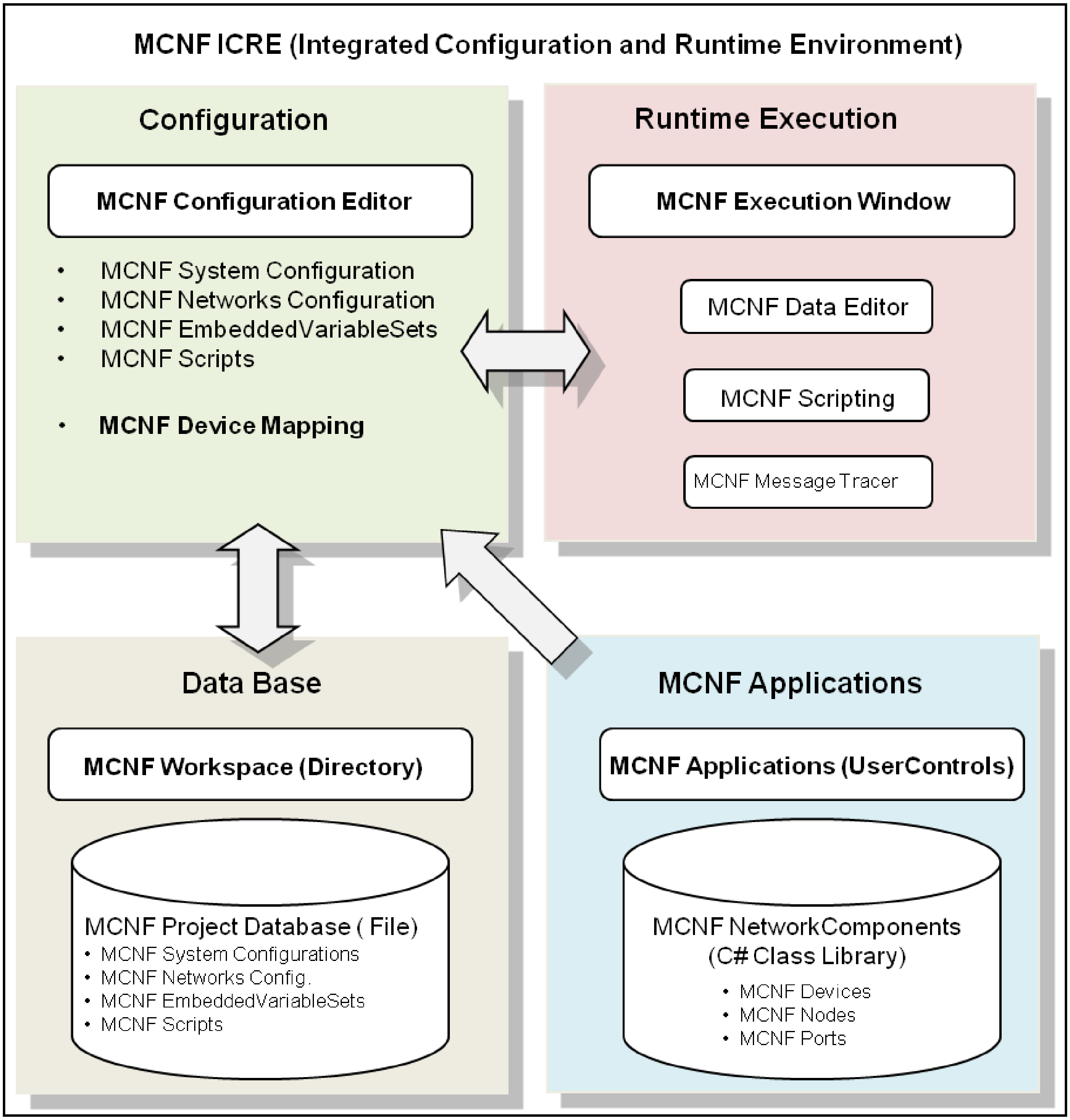
The master's MCNF framework is basically built around the "MCNF_SystemLib" class library, which contains base classes for all components and implements the framework mechanisms.

The elements defined by the user would then be used to create a MCNF network definition and a MCNF system definition. In the MCNF system definition, the user combines application user controls with a network definition and maps application devices (device models) to embedded devices.
These configuration tasks are accomplished using the "MCNF_ConfigurationEditor" which is part of the MCNF's "Integrated Configuration and Runtime Environment" (MCNF_ICRE). The MCNF_ConfigurationEditor allows the user to define and manage Projects, System Configurations, Network Configurations incl. Network Parameter Settings and export/import elements between projects.

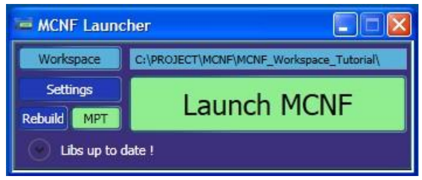
From the MCNF_ConfigurationEditor the user can launch the system execution either in a simulation mode or connected to the real embedded system.

When execution a MCNF system, the application's user controls are shown in the "MCNF_Execution" window. They can be arranged freely within the window.

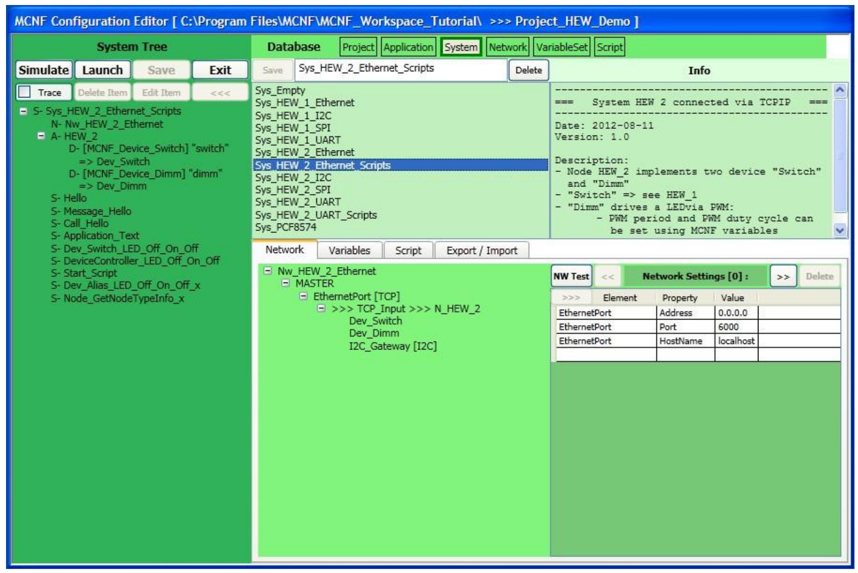
The MCNF_Execution window also provides access to several support functions like (Online-) Data Editor, Message Tracer and Scripting.



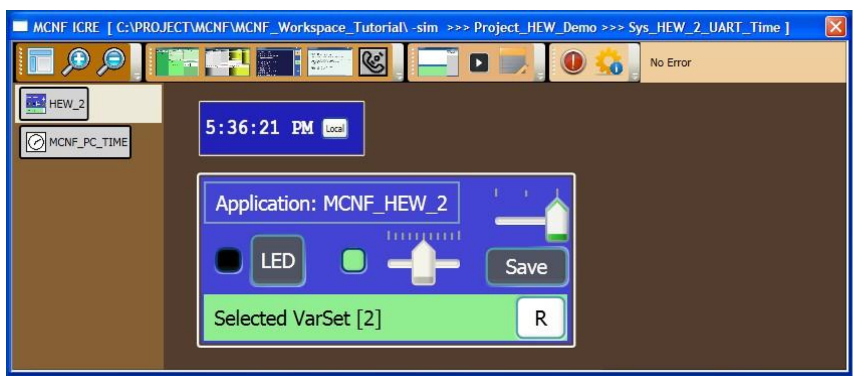
MCNF Launcher



System and Network Configuration Editor



Execution Window with 2 Applications



Data Editor: **Scripting Window:** **Message Tracer:**

Tools: