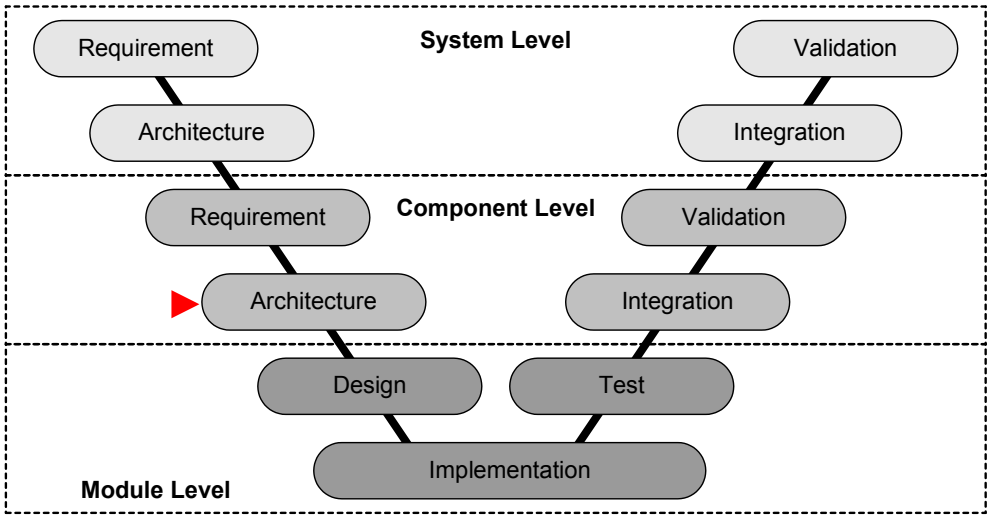
MEN Mikro Elektronik GmbH
Neuwieder Straße 3-7
90411 Nürnberg | Germany

Phone +49-911-99 33 5-0
Fax +49-911-99 33 5-901

info@men.de
www.men.de

Doc. Id.:	**14MH50C_SwArchSpec**	
Title:	**14MH50C Software Architecture Specification**	
Issue:	0.5	
Date of Issue:	2015-10-29	
Status:	in rework	
File name:	1517_DB-NetzAG_OpenETCS_MTCS/MEN-Products/14/14MH50C99/Specifications/work/14MH50C_SwArchSpec.docx	
Project:	1517 DB-NetzAG_OpenETCS_MTCS	
Author(s):	A. Jendrusch	
Reviewer:	[NAME]	
Quality Mgr.:	[NAME]	
Project Leader:	[NAME]	
Safety Mgr.:	N/A	



Im ausgedruckten Zustand NUR ZUR INFORMATION

Printed versions FOR INFORMATION ONLY

Please consider the environment before printing this document

List of Changes

Issue	Change Description	Author	Date of Issue
0.1	Created	F.Roschlapil	2015-08-20
0.4	submit	A.Jendrusch	2015-08-21
0.4	rework	A.Jendrusch	2015-10-29

Contents

Requirement Summary

1 Objective

1.1 Identification

This document is targeted at the openETCS project team members willing to understand the MEN Platform architecture in order to define how it will be integrated with the openETCS software and testing environment. It provides an overview of the MEN Platform, used as the base hardware platform for the demonstrator component of the openETCS project as part of the WP5.

1.2 Applicable Documents

1.2.1 Applicable Standards

Table 1. ***Applicable Standards***

Reference	Doc. Identification	Issue	Document Title

1.2.2 Applicable Specifications

The exact issue of the applicable documents is defined within the current issue of the applicable Hardware Configuration Index Document.

Table 2. ***Applicable Specifications***

Reference	Doc. Identification	Document Title

1.3 Referenced Documents

Table 3. ***Referenced Documents***

Reference	Doc. Identification	Issue	Document Title
MTCS		0.21	MTCS Prototype Application Note

1.4 Definitions

These definitions are provided for the terms which are used in this document. If a term is not defined here, it is possible that it is defined instead in the body of this document.

Table 4. ***Definitions***

Term	Definition
Application Specific Integrated Circuit (ASIC)	Integrated Circuits which are developed to implement a function, including, but not limited to: gate arrays, standard cell, and full custom components encompassing linear, digital, and mixed mode technologies.
Commercial Off-the-Shelf (COTS) Component	Component, integrated circuit or subsystem developed by a supplier for multiple customers, whose design and configuration is controlled by the supplier's or industry specification. Note: Examples of COTS components include resistors, capacitors, microprocessors, unprogrammed Field Programmable Gate Array and Erasable Programmable Logic Devices, other integrated circuit types and their implementable models, printed wiring assemblies and complete LRUs which are typically available as a catalogue item.
Complex Hardware Item or Complex Digital Device	All items that are not simple are considered to be complex.
New requirements	Additional requirement resulting from the hardware design processes, which may not be directly traceable to higher level requirements.
Integrated Circuit	- A circuit consisting of elements inseparably associated and formed on or within a single substrate to perform an electronic circuit function.
Process	A collection of activities performed in the development life cycle to produce a definable output or product.
Simple Hardware Item or Simple digital device	A hardware item is considered simple if a comprehensive combination of deterministic tests and analyses can ensure correct functional performance under all foreseeable operating conditions with no anomalous behaviour.
Software	Computer programs procedures, rules and associated documentation and data pertaining to the operation of a computer system.
Standard	A rule or basis of comparison used to provide both Guidance in and assessment of the performance of a given activity or the content of a specified data item.
System	A collection of hardware and software components organized to accomplish a specific function or set of functions.
System Architecture	The structure of the hardware and the software selected to implement the system requirements
System Requirements	Synonym for: User Requirements, Technical Specification, System Specification, Product Specification, Purchaser Technical Specification

Testing	The process of exercising a system or system component to verify that it satisfies specified requirements and to detect errors.
Traceability	The evidence of an association between items, such as between process outputs, between an output and its originating process, or between a requirement and its implementation.
Validation	The process of determining that the requirements are the correct requirements and that they are complete. The system life cycle process may use software requirements and New requirements in system validation.
Verification	The evaluation of the results of a process to ensure correctness and consistency with respect to the inputs and standards provided to that process.

1.5 Abbreviations

Table 5. ***Abbreviations***

Abbreviation	Description
ASIC	Application Specific Integrated Circuit
COTS	Commercial-Off-The-Shelf
CP	Control Processor
CPU	Central Processing Unit
DES	Design Item
DMI	Direct Media Interface
EN	European Norm
EVC	Electronic Vehicle Control
FUNC	Functional Item
IOP	I/O Processor
ISO	International Organization for Standardization
MEN	Mikro Elektronik GmbH Nürnberg
MTCS	Modular Train Control System
OS	Operating System
PROD	Product
QNX	QNX (Realtime OS)
SW	Software
TCP	Transmission Control Protocol
TCPI-IP	Transmission Control Protocol Internet Protocol

1.6 Conventions

Date format shall comply with ISO 8601 respective EN 28 601 extended formats: YYYY-MM-DD, e.g. 2005-12-24.

Bold type is used for emphasis.

Hexadecimal numbers are preceded by "0x", which is the usual C-language convention, and are printed in a monospace type, e.g. 0x00FFFF.

Signal names preceded by a slash ("/") indicate that this signal is either active low or that it becomes active at a falling edge.

Signal directions in signal mnemonics tables generally refer to the corresponding board or component, "in" meaning "to the board or component", "out" meaning "coming from it".

Vertical lines on the outer margin signal technical changes to the previous edition of the document.

Use of the words ‘shall’, “should”, ‘must’, ‘will’, ‘may’ and ‘is’ within the products of this standard shall be as follows:

The word ‘shall’ in a text expresses a mandatory requirement of the specification. Deviation from such a requirement is not permissible without formal agreement between the relevant parties.

The word ‘should’ in a text expresses a recommendation or advice on implementing such a requirement of the related specification. System Design expects such a requirement or advice to be followed unless good reasons are stated for not doing so.

The word ‘must’ in a text is used for legislative or regulatory requirements (e.g. health and safety) with which both the System Design and the Hardware Design shall comply. It is not used to express a requirement of the specification.

The word ‘will’ in a text expresses a provision or service by System Design or an intention by System Design in connection with a requirement of the specification. The development team may rely on such provisions, service or intention.

The word ‘may’ in a text expresses a permissible practice or action. It does not express a requirement of the specification.

The word ‘is’ in a text does not express a requirement of the specification. It describes a state or action performed not by the hardware.

Requirements shall comply with the following conventions:

<REQUIREMENT-ID> <CLASS> <BRIEF-DESC>

<REQUIREMENT-ID>:

MEN_<PROD>_SYSR_<xxxx> for System Requirements Specifications

MEN_<PROD>_SYSA_<xxxx> for System Architecture Specifications

MEN_<PROD>_SYST_<xxxx> for System Test Specifications

MEN_<PROD>_SR_<xxxx> for Software Requirements Specifications

MEN_<PROD>_SA_<xxxx> for Software Architecture Specifications

MEN_<PROD>_SD_<xxxx> for Software Design Specifications

MEN_<PROD>_ST_<xxxx> for Software Test Specifications

<CLASS>:

[FUNC]: A requirement that describes functional behaviour

[DES]: A requirement that describes how a specific function should be implemented. (This may include design constraints).

[PROCESS]: A requirement that demands a specific [development] process. Examples: Compliance to a development standard, tool usage, documentation, manufacturing.

For example:

MEN_XXXX_SA_1000 [FUNC] Example Requirement

If this is a New or New Requirement (not directly related to upper level requirement, but evolving from given OS), the requirement is followed by

New

Otherwise the upper level requirements are referenced:

MEN_A406_HR_1001 MEN_A406_HR_1002

2 Demonstrator

2.1 Overview

The Demonstrator consists of these three Hardware-Units:

Track Unit

- Balise transmission simulation
- Radio transmission simulation
- Train movement simulation

Onboard Unit

- EVC Kernel

DMI Unit

- DMI function

2.1.1 Track Unit

Hardware : Raspberry PI
OS : Linux raspberrypi 4.0.8-v7

The Track Unit simulates the route between Amsterdam and Utrecht.
At a defined cycle time (e.g. 10ms) the SCADE Model 'Amsterdam-Utrecht' is called with new Input Data. The resulting Output Data are transmitting via a TCP-IP Socket to the Onboard Unit.
A TCPI-IP socket Server is running at the Track Unit and accepts connection with the TCP-IP client at the Onboard Unit.
TCP/IP Port 1003 (EVC-to-TRACK)
TCP/IP Port 2003 (TRACK-to-EVC)

2.1.2 Onboard Unit

Hardware : MEN MTCS Platform
IOP OS : Yocto Linux
CP OS : QNX 6.5 SP1

MEN hardware for the openETCS demonstrator, later on referred as “MEN MTCS Platform”, in its simplest configuration, consists of a rack, a power supply, and a series of boards. Each board provides a specific set of functionalities and external interfaces. Different platform configurations allow to satisfy different interface requirements. Details are described at the ‘MTCS Prototype Application Note’.

The SCADE Model ‘EVC’ is running on one of the two CPs as a Thread.

Two other threads handle the socket communication to the Track Unit and to the DMI Unit.

The ‘EVC-Model’ is periodic called every 100ms with the Input Data Structure. If the EVC-Model returns, the communication threads to the DMI Unit and back to the Track Unit are signaled.

2.1.2.1 COM-1 Thread (Track-Side Thread)

This thread receives the Track-Data from the Track Unit (TCP-IP Port 2003) and signals to the ‘SCADE Thread’ that new Input Data are available.

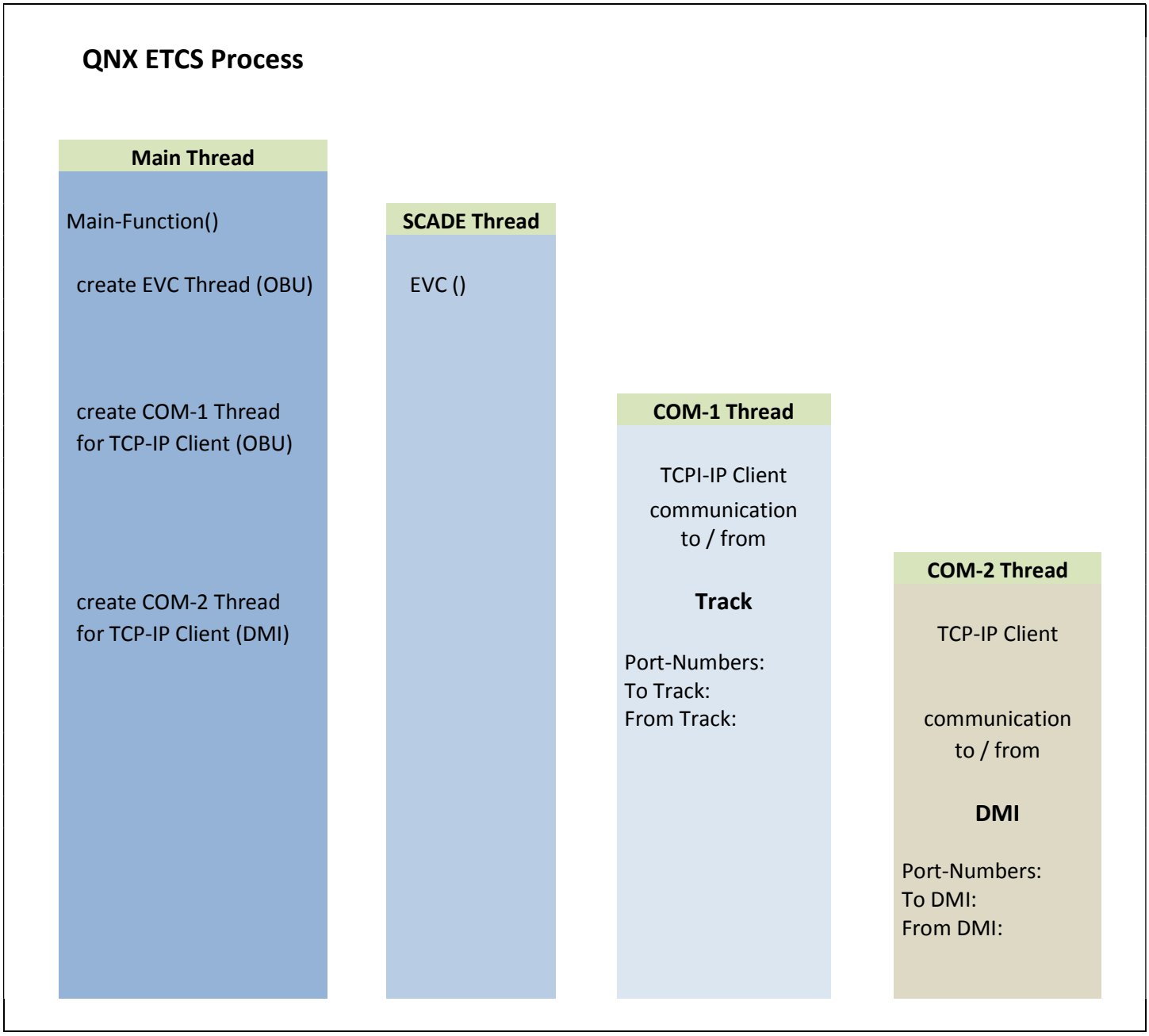
If the EVC Model returns this thread is signaled for transmitting data back to the Track.

2.1.2.2 COM-2 Thread (DMI-Side Thread)

This thread is signaled by the ‘SCADE Thread’ if the EVC-Model returns. The task is to transmit the Output Data to the DMI Unit (TCP-IP Port 1001).

Otherwise, If the COM-2 Thread receives data from the DMI Unit (TCP-IP Port 2001), it is necessary to signal new Input Data to the ‘SCADE Thread’.

2.1.2.3 SW Overview at the Onboard Unit



2.1.3 DMI Unit

Hardware : MEN DC13
OS : Windons 7 Embedded

A TCPI-IP socket server is running at the DMI Unit and is connected with the TCP-IP client at the Onboard Unit (TCP-IP Ports 1001(receive)/2001(send)).

This client receives the DMI Input Data from the Onboard Unit and sends back the SCADE Model Output Data.