LXI
Reference Design

www.lxistandard.org
Introduction

• Peter Plazotta, CEO of TSEP
• About TSEP:
  ▪ 20 members staff (engineers, software developers, software developer trainees)
  ▪ Development of complete software solutions for Measuring Devices
  ▪ Development of testsystem for different companies, used world wide (Germany, Spain, China, Japan)
  ▪ Systemsoftware development for customers (Drivers etc.)
  ▪ LXI Member
Why a LXI Reference Design?
(From the LXI Consortium side)

• Lowers the boundary for vendors to use the standard
• Clarification of the standard
• Easier and wider distribution of the standard
• Shows vendors that the LXI Standard is ready to use
Why a LXI Reference Design? (From the vendors side)

- Reduces the implementation costs for the standard dramatically
- Lower costs and efforts for vendors if the standard changes
- Easier certificating for vendors
- Compatible with other vendors (i.e., LAN Messages)
Steps for LXI Reference Design and Implementation

1. Requirements
Steps for LXI Reference Design and Implementation

1. Requirements \( \Rightarrow \) done
Steps for LXI Reference Design and Implementation

1. Requirements ➔ done
2. Reference Design
Steps for LXI Reference Design and Implementation

1. Requirements \(\Rightarrow\) done
2. Reference Design \(\Rightarrow\) in progress
Steps for LXI Reference Design and Implementation

1. Requirements ➔ done
2. Reference Design ➔ in progress
3. Reference Implementation
Steps for LXI Reference Design and Implementation

1. Requirements ⇒ done
2. Reference Design ⇒ in progress
3. Reference Implementation ⇒ in progress
What will be included in the LXI Reference Design

- LXI Core 2011, including IPv6
- LXI Event Messaging
- LXI Event Logging
- LXI HiSlip
What will be excluded from the LXI Reference Design

- all modules depending on a vendor specific hardware
  - Clock Synchronisation
  - Wired Trigger Bus
  - Time Stamps

- reference hardware will not be provided
LXI Reference Design and IPv6

- IPv6 is included in the Reference Design
- IPv6 is not a standalone module
- IPv6 can be used optionally (compile time or runtime)
Requirements for the LXI Reference Design (LXI Standard)

- all rules and recommendations ➔ qualified
- unnecessary requirements ➔ skipped
- requirements which will be fulfilled just partly ➔ clarified
Requirements for the LXI Reference Design (non-functional)

- development environments
  - OS
  - compiler
  - tools
- usage of third-party software
- documentation of the Reference Design and source code
- version control system and the bug tracking
Requirements for the LXI Reference Design (low-end devices)

• keep in mind the low-end devices
• „lightweight“ C++
  – no templates
  – no complex inheritance
  – no exceptions
• avoid huge objects on the stack
• decrease the amount of dynamically allocated memory
Requirements for the LXI Reference Design (low-end devices)

- do not use modern C++ libraries, such as STL or Qt
- define all data sizes by constants
- avoid new C++ standard features
- support an embedded web server
Development Tools for the LXI Reference Design

- Sparx Tool „Enterprise Architect“
i. e. an UML design and business analysis
tool for object-oriented software systems
  – modeling
  – documenting
  – reverse engineering
  – building and maintaining

- customized output of the design results

- requirements included ⇒ traceability matrix
Development Tools for the LXI Reference Design
Development Environment for the LXI Reference Implementation

• Platforms
  – Windows 7 (64 bit)
  – Ubuntu Linux (i386, 64 bit)
  – Ubuntu Linux Arm (32 bit)

• Windows ⟷ Visual Studio 2013

• Linux ⟷ Eclipse, GNU C++ Compiler

• solutions for compiling will be provided
Development Environment for the LXI Reference Implementation

• Linux Arm development  ➔ on CubieTruck board

• Windows and Ubuntu (i386) development  ➔ in an Oracle Virtual Box

• source code documented for doxygen
Current Status:
We are on schedule!

- all prototyping (web server etc.) ➞ finished
- all basic classes ➞ designed and coded
- first modules ➞ designed
  – Core Framework, Network, Application Interface
- coding of the Core Framework ➞ started
- mDNS and Event Logging modules ➞ May
- HiSlip and Event Message modules ➞ June
Thank you!