Eyuel Zewdu Teferi
David Schumacher
Souha Kouki
STMicroelectronics

MIPI® I3C℠ Application and Validation Models for IoT Sensor Nodes
MIPI® I3C℠ Application and Validation Models for IoT Sensor Nodes

• I3C: Overview
  – Evolution of MIPI I3C Interface specification
  – I3C Features and Uses

• I3C application to IoT Sensor Nodes
  – Sensor Node sub-systems
  – Communication among Sensor Node sub-systems
  – Using I3C for Sensor Node board communication

• I3C CTS and SysML as enablers toward an efficient validation
  – Why manage IoT systems complexity?
  – How to manage IoT systems complexity?
  – IBI use case validation management example

• Conclusions and Best Practices
MIPI I3C Overview
Evolution of MIPI I3C Interface specification

I3C v1.0
- First Specification
- Released in 2016
- 2 wires interface
- IBI for interrupt management
- Hot-Join
- 1.2V – 3.3V support

I3C v1.1
- New I3C specification
- More features added
- Released in 2019
- Multi-lane configuration
- Structured IBI notification
- Group Addressing
- In-Band Reset function

I3C v1.1.1
- Updates on v1.1
- Released in 2021
- Virtual devices use
- Passive Hot-Join
- Group management
I3C Features and Uses (1/3)

- Multi-lane configuration (introduced in v1.1)
  - Allows the I3C bus to extend the Data lines to two or 4 data lanes
  - This helps in increasing the data throughput using the same base clock
  - Dual-Lane and Quad-lane increase the effective data rate by 2x and 4x respectively
- Structured IBI notification (introduced in v1.1)
  - IBI can be accompanied by Mandatory Data Byte (MDB)
  - By defining a structured MDB format the efficiency of IBI usage can be increased
  - These updates include the pending read notification and timing control information
I3C Features and Uses (2/3)

• Group Addressing (introduced in v1.1)
  – I3C is already has a broadcast and unicast communications
  – Group addressing allows devices to be assigned a Group address effectively creating a multicast system
  – The multicast is advantageous in sending data to a group of devices on the bus
  – Helps in grouping devices with different capabilities into a group i.e., multi-lane
  – Revised V1.1.1 allows groups to have a dedicated multilane configuration

• In-Band Reset function
  – This pattern-based reset function is aimed at making the I3C a true two wire interface
  – This will help in avoiding the use of a dedicated reset line
  – It has a configurable option where the reset can be escalated from a peripheral reset to whole chip reset
  – Useful in avoiding deadlocks on the bus due to an unresponsive device
I3C Features and Uses (3/3)

- **Secondary Controller (introduced in v1.0 and updated in v1.1)**
  - I3C is a true multi-drop interface
  - Built-in support for multiple controllers
  - Conflict avoidance and bus control management are already part of the protocol
  - Devices that are capable of being in control can request and give back control

- **Low power consumption**
  - I3C is a protocol that has been designed to support different voltage ranges
    - The standardized electrical behaviors can be adopted based on the design requirement
      - 1.2V – 3.3V
  - Low power applications can take advantage of these I3C features
    - High speed communication with the same base clock achieved through
      - HDR modes
      - Multilane communication
    - Low Voltage communications
    - Ideal for battery operated devices
I3C Application to IoT Sensor Nodes
IoT Sensor Nodes

• Internet of Things (IoT) is a network that is made of physical objects that are capable of being connected to a network to make measurements or to execute given tasks.

• Critical elements of an IoT system are:
  – The communication element that manages device connectivity (**Internet**)
  – The sensing and processing system (**Things**)

• Sensor Nodes are basic elements of an IoT device that comprise the sensing and the connection element.
A sensor node is an IoT device that is made of different components divided into subsystems.

A sensor node consists the following subsystems:

- **Sensing**: continuous small scale data generation
- **Processing**: central processing based on configuration
- **Communication (wired/wireless)**: for connected devices
- **Power supply**: battery subsystem for remote devices
- **Memory**: data storage for later processing or backup to Memory
- **Security**: all devices need to be secure from intrusion
Communication among Sensor Node sub-systems

- A sensor board communication could be divided into two types based on the amount of data and frequency of communication

**High Activity Low Throughput bus**
- H.A.L.T bus
- Short but frequent communications.
  - Sensing subsystem
  - Actuating subsystem
- Configuration of system
  - Power management
  - Subsystem configuration
- Frequent interrupts from sensing elements

**Low Activity High Throughput bus**
- L.A.H.T bus
- Large data that require high throughput
  - Security Subsystem
  - Memory Subsystem
- Low latency communication
  - Communication subsystem
  - Memory Subsystem
- Periodic communication requests
Using I3C for Sensor Node board communication

Bus configuration Option 1

- Use two buses for the two types of communication
  - Use a secondary controller to manage the high activity bus
  - L.A.H.T bus could use these I3C features for high data rate
    - Multi-Lane
    - HDR modes
  - H.A.L.T bus requires
    - Single lane due to continuous IBI
    - Secondary controller as a Sensor Hub to collect data from sensors
Using I3C for Sensor Node board communication

Bus configuration Option 2

- Take advantage of Group addressing
- Two groups with dedicated features per group
  - Group 1: H.A.L.T
    - Group of devices that use the single lane features
    - Devices in this group can send IBI
  - Group 2: L.A.H.T
    - Grouped to use multilane configuration for higher throughput
    - Devices in this group may not require the IBI capability
Sensor Node reference board communication using I3C
I3C CTS and SysML as enablers toward an efficient validation
Why manage I3C IoT systems complexity?

• Systems are:
  – More and more complex,
  – More and more connected,
  – More and more critical,
  – More and more secured,
  – More and more standardized

• Project management needs:
  – Effective collaboration
  – Good understanding of (sub-)systems
  – Best anticipation as possible
  – Effective design, development and verification
Why manage I3C IoT systems complexity?

• Targets:
  – To build a global and shared vision
  – To identify needs and constraints
  – To synchronize all stakeholders + constraints between themselves
  – To be efficient with trade-offs and optimizations

• In a nutshell:
  – By setting of adequate activities to design, develop, and verify a system while satisfying all stakeholders
Why manage I3C IoT systems complexity?

Conceptualization  
Design  
Development  
Verification

Influence on project cost

High  
Low

With shared constraints  
Without shared constraints

Project lifecycle

© 2021 MIPI Alliance, Inc.
Why manage I3C IoT systems complexity?

• Digital communication interfaces = A choking point of the global performance of the system
• Sensor node example with I3C interface as subsystem:
  – 2 modes + HDR sub-modes
  – Backward compatibility
  – Multi-role
  – I3C Built-in features

⇒ I3C = Powerful system enabler
How to manage IoT systems complexity?

• One interesting path: By building the model of the system:
How to manage IoT systems complexity?

Inputs

- Requirements model
  - Use cases diagrams
  - Requirement diagrams
  - Test cases diagrams

- Architecture model
  - Packages diagrams
  - (...)

- Dynamic model

Traceability

Allocation

Detect issues earlier

© 2021 MIPI Alliance, Inc.
How to manage IoT systems complexity?

System model

Behavior diagrams

Requirement diagrams

Parametric diagrams

Structure diagrams

Use case

State machine

Activity

Sequence

Package

Blocks Definition

Internal Blocks Def.
Example of I3C Sensor Node

- By considering system in its environment,
- By exploring all interface capabilities
- By linking and crossing all requirements:
  - Technical,
  - Functional,
  - Standards,
  - Validation,
  - ...
Example of I3C Sensor Node

• High level system description
  – Nominal scenario
  – Alternate scenario
  – Error scenario

• (Sub) Systems stakeholders identification
**Link to I3C use cases**

- High level dependance identification,
- Stowage link with other sub-system
- Model of MIPI I3C standard:
  - Cross view,
  - No “hidden” constraints,
  - Items identified earlier
Link to CTS use cases

- Identification of dependances or constraints by dynamic links
- Example: I3C standard and CTS model help to not forgot items to implement
Focus on demand

Capabilities to dive into:
- Standard
- CTS
- Project
Conclusions and Best Practices

- Advanced features allow easy adoption of MIPI I3C
- MIPI I3C: A scalable and flexible solution for IoT Sensor Node
- CTS as a starting point for I3C protocol validation
- CTS as an enabler to model your complex system validation
- Advanced systems validation model can be built on top of CTS
• MIPI® I3C® Specifications
  – https://www.mipi.org/specifications/i3c-sensor-specification

• MIPI® I3C® System Integrator App Note

• MIPI® I3C® FAQ
  – https://www.mipi.org/resources/I3C-frequently-asked-questions

• MIPI® I3C® CTS
  – https://www.mipi.org/faq-category/conformance-testing

• STMicroelectronics MEMS Sensor products
THANK YOU!