Fault Tolerant Smart Transducer Interfaces for Safety-Critical Avionics Applications

Safwen Bouanen[1], Claude Thibeault[1], Yvon Savaria[2], José-Philippe Tremblay[2], Guchuan Zhu[2]

1. École de Technologie Supérieure
2. École Polytechnique de Montréal

Accepted in IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC), 2013

10/18/2013

Outline

1. Introduction
2. IEEE 1451
3. Interfaces design
4. The prototype
5. Test and validation
6. Conclusion

1. Introduction (1)

Current trends in the avionics domain

- Ever increasing number of functions/transducers
- Information flow increase
- Different communication protocols
- Diversity in the transducers market
- Migration to smart transducer interfaces
- Increasing design effort, cost and time

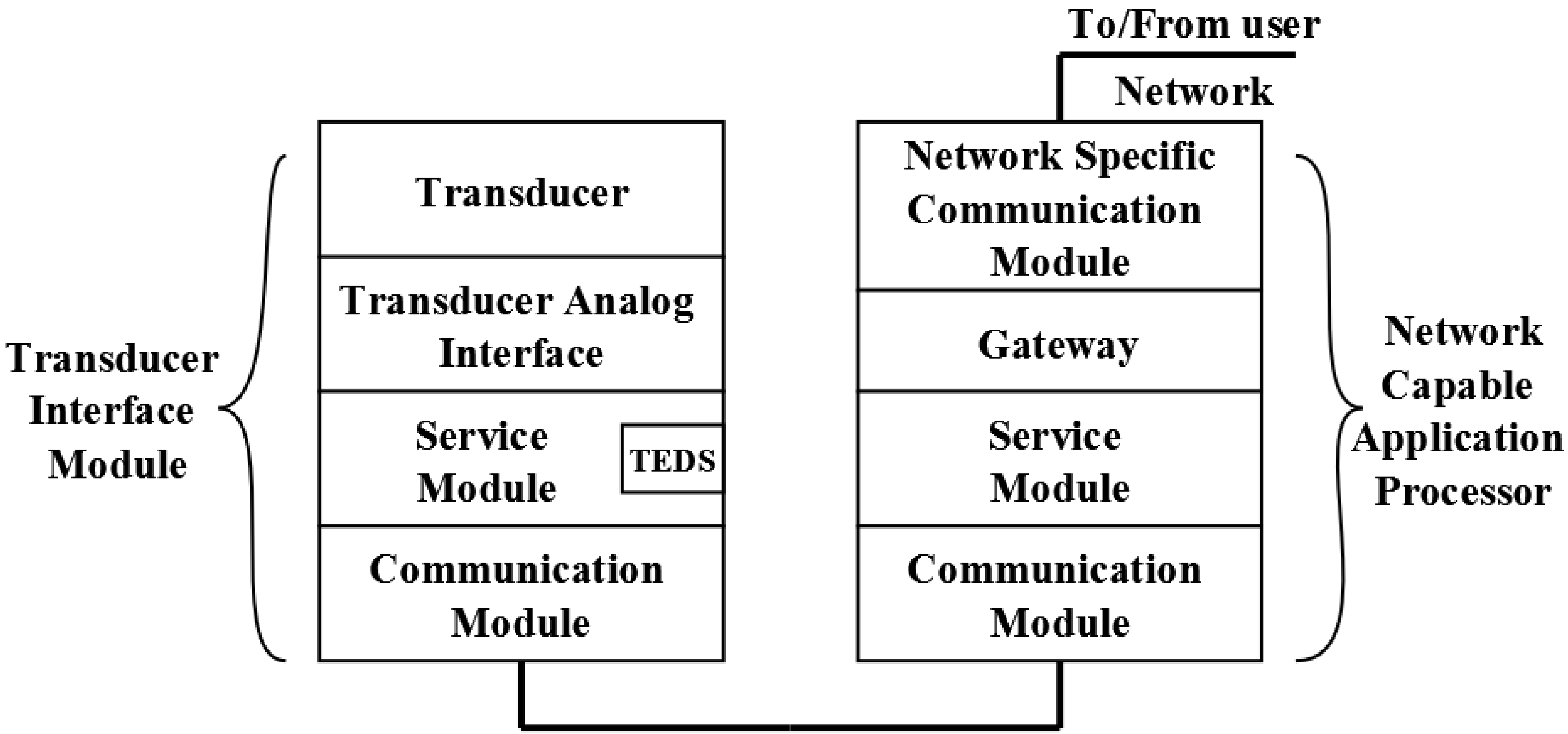
1. Introduction (2)

- Fault tolerance challenges
  - DIMA/IMA2G: I/O and computation close to transducers
  - Hardware redundancy complicates fault diagnosis
  - Strict reliability requirements
- Solutions: Transducer interfaces with:
  - Embedded fault tolerance mechanisms
  - Fail-safe capabilities
  - Graceful performance degradation

2. IEEE 1451 Standard (1)

- IEEE Standard for a Smart Transducer Interface for Sensors and Actuators (IEEE1451)
- Adoption Advantages
  - Increased compatibility
  - Independent of the selected network
  - Reduced design, installation and update effort
- Considered but not yet adopted by the avionics domain
- Fault tolerance aspects not yet fully addressed.

2. IEEE 1451 Standard (2)



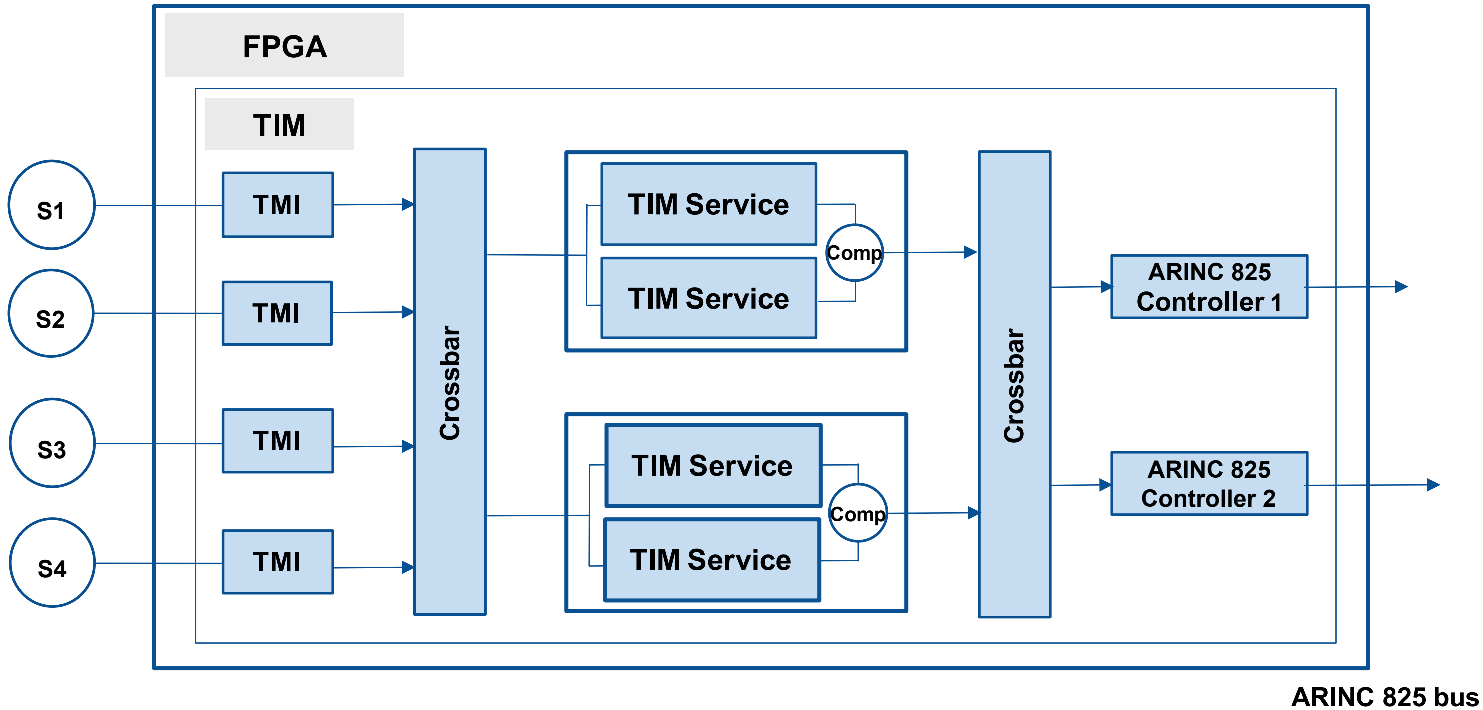
IEEE 1451 Reference Model

3. Interfaces Design
3.1 Requirements definition

- The interface must be able to deal with transient and permanent faults
- The interface should fail in a safe manner
- The interface should ensure graceful degradation
- MTBF smaller than 10^{-6}/ operating hours
- Unannunciated errors shall be less than 10^{-6}/ operating hours

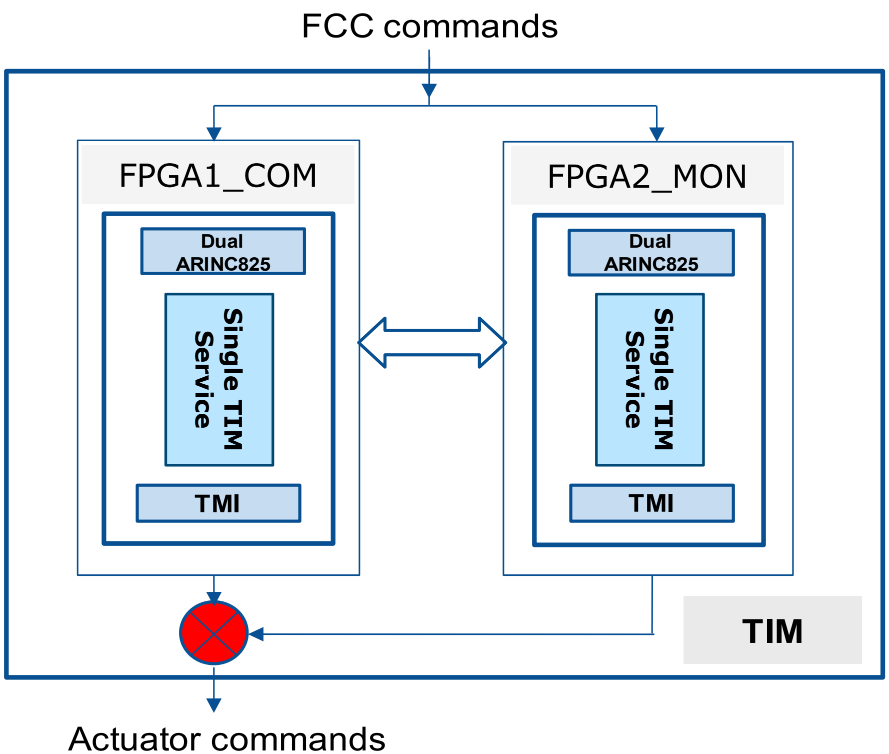
3. Interfaces Design

3.2 Single-chip Interface



- TIM-service cores aggregated in redundant lock-step pairs

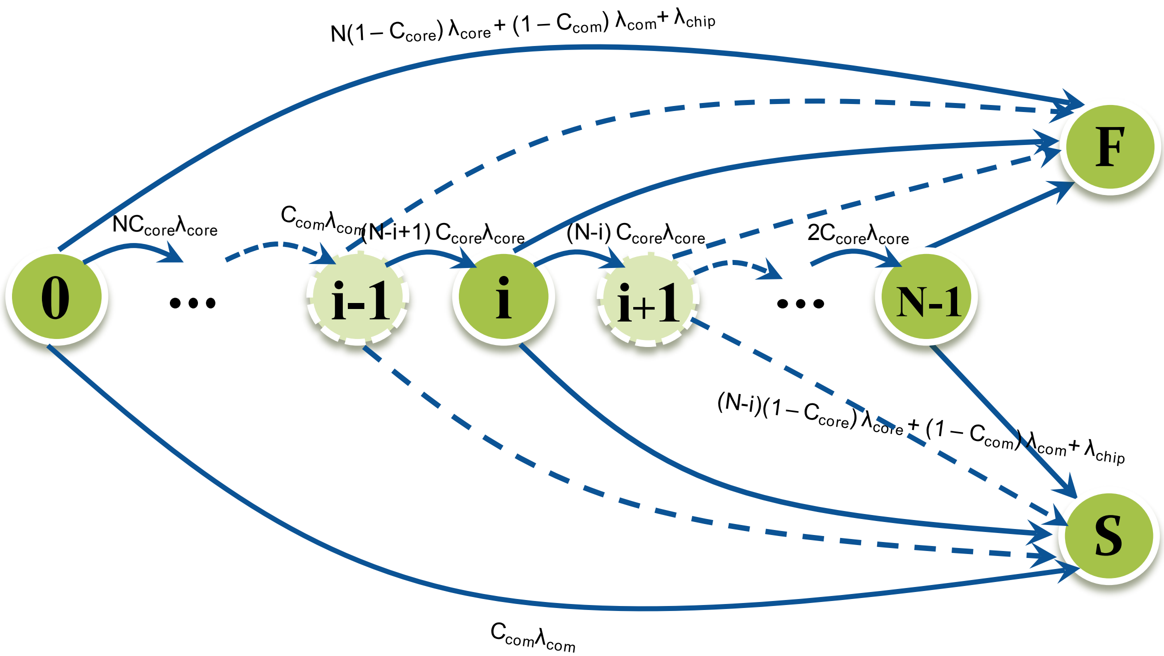
3. Interfaces Design
3.3 Dual-chip Interface



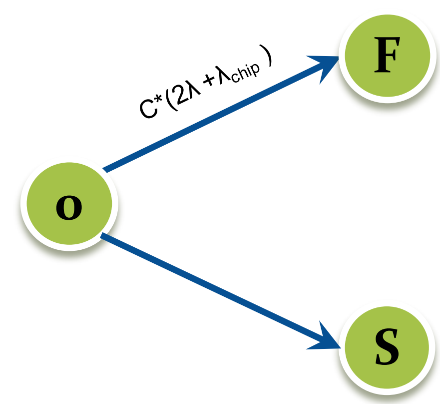
- Fault detection in ensured through the comparison of the outputs of the COM and MON lanes

3. Interfaces Design
3.4 Reliability and Safety Analysis

Markov Chains-based modeling

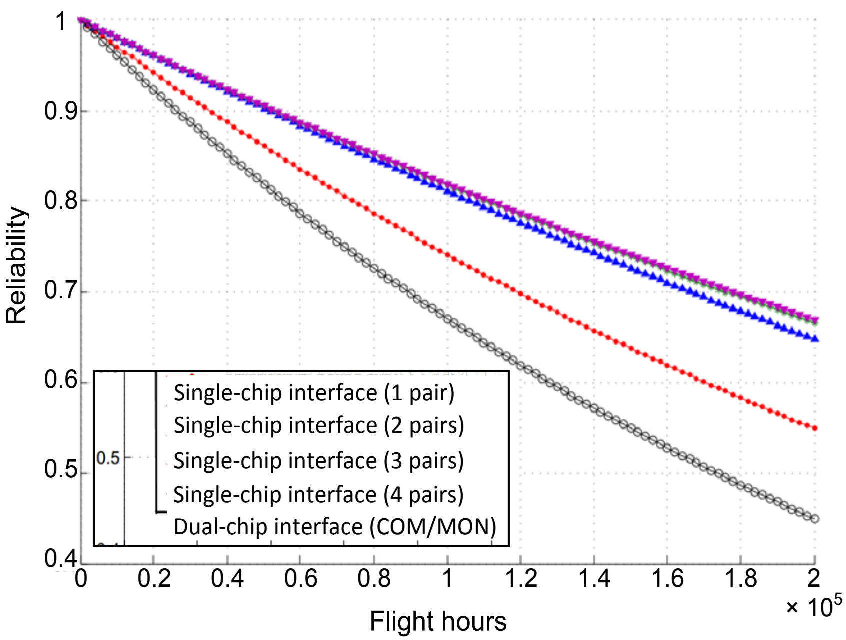


Single-chip model

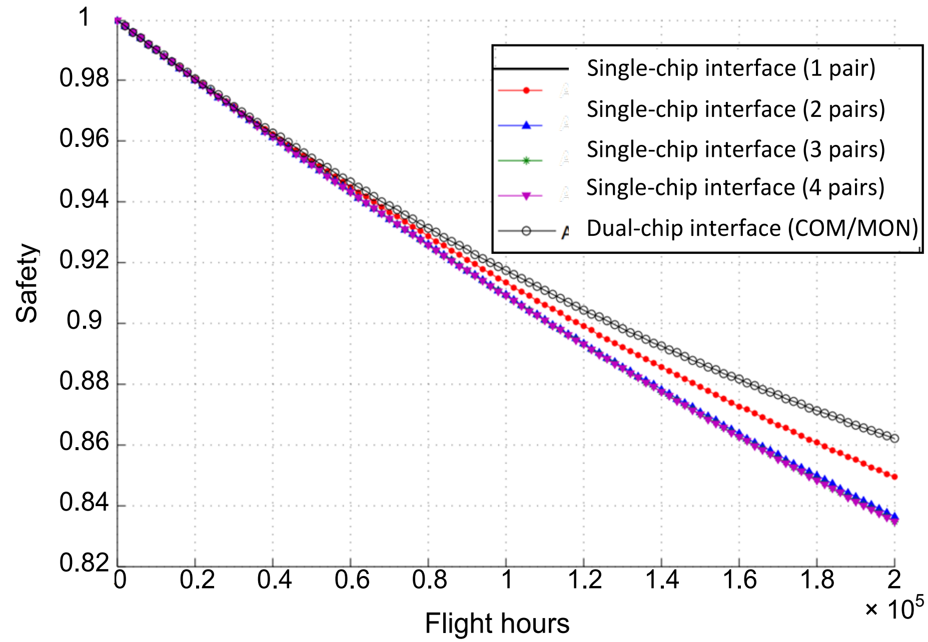


Dual-chip model

3. Interfaces Design
3.5 Analysis Results



Reliability



Safety

32nd DASC

4. The Prototype
4.1 Prototyping advantages

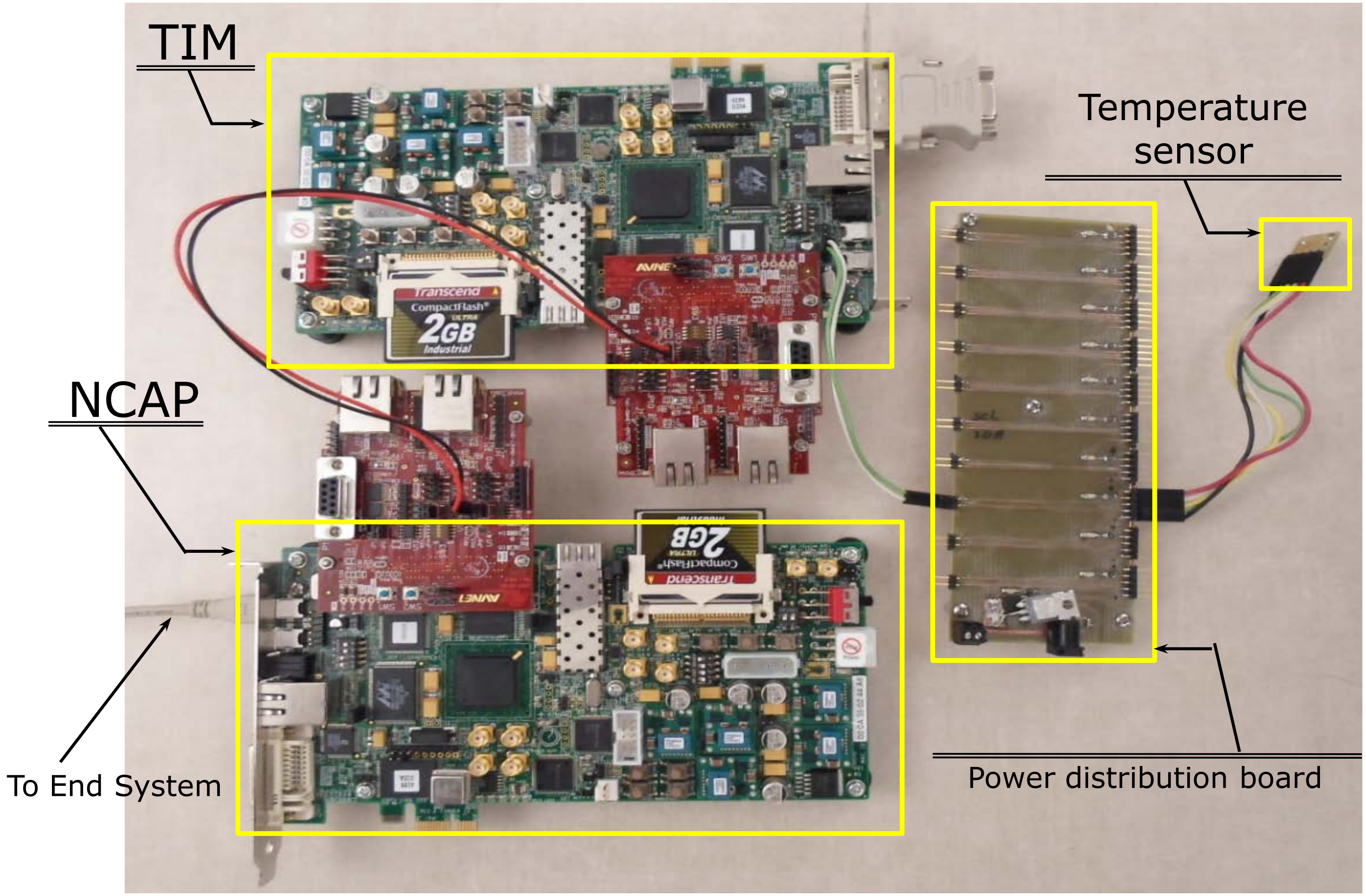
Prototyping allows:

- Identifying practical challenges and constraints
- Testing and benchmarking new algorithms
- Concept validation in early development stages
- Characterizing the design
- Exposing implementation challenges

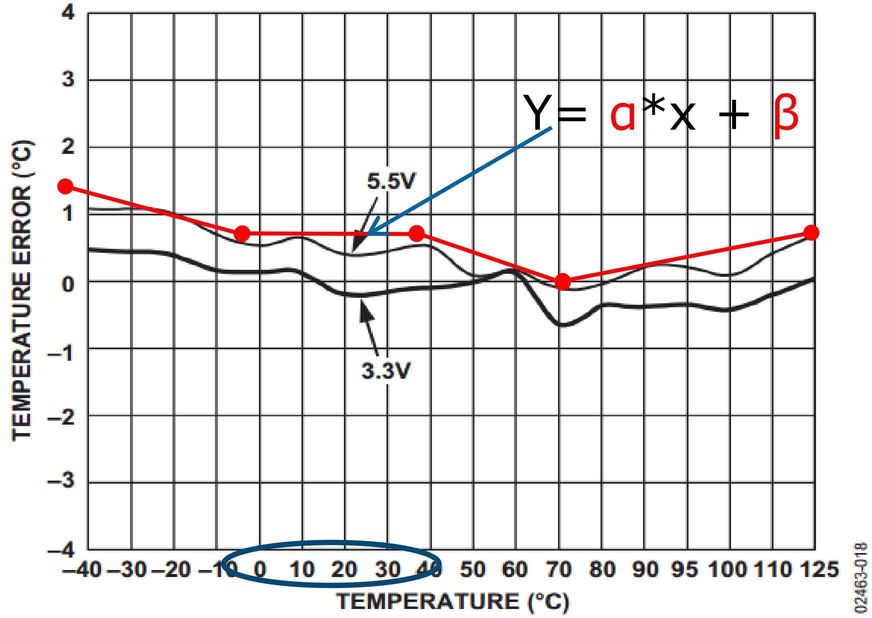
4. The Prototype
4.2 Prototype description (1)

- The prototype includes
  - 2 LX45T FPGA boards
  - 2 Mezzanine boards
  - Dual ARINC 825 bus
  - COTS sensors: AD7415 temperature sensors
- Configured to maintain
  - 1 Mbit/s throughput
  - Guaranty determinism

4. The Prototype
4.2 Prototype description (2)



4. The Prototype
4.3 TIM services



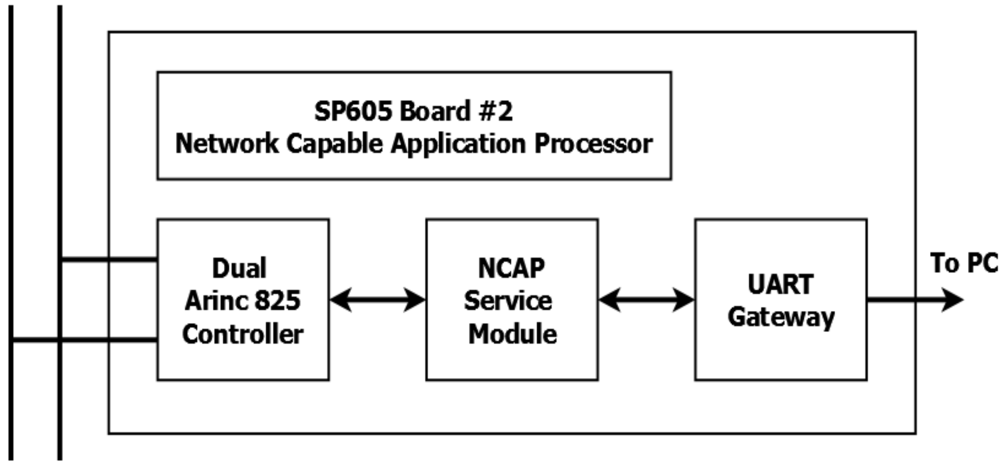
8. Typical Temperature Error @ 3.3 V and 5.5 V

- Example of a service offered by the TIM prototype
  - Data validation and correction
  - Coefficients are stored within the TEDS

32nd DASC

IEEE

4. The Prototype
4.4 NCAP



NCAP design

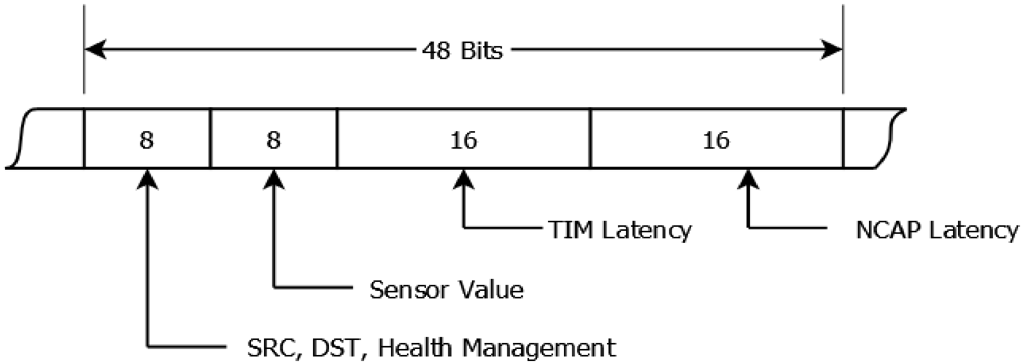
- The main functionalities of the NCAP are:
  - Data flow management (ARINC 825/UART-USB)
  - Data messages formatting (NCAP Service)
  - Latency measurement (NCAP Service)

5. Test and Validation
5.1 Fault Tolerance Validation

- Objectives:
  - Evaluate the interface capacity to detect and recover from faults
  - Evaluate messages latency in normal operation mode
  - Evaluate the impact of occurrence of faults on messages delays.
- Test procedure
  - Based on fault injection technique
  - Emulating the occurrence of faults in the TIM-service pairs.

5. Test and Validation
5.2 Latency Validation

- Latency Measurment :
  - NCAP_Latency is the delay for transferring messages through the ARINC 825 bus
  - Measured total latency = TIM_latency + NCAP_latency
  - The real latency is calculated by compensating the measured latency



Message structure

5. Test and Validation
5.3 Results

Sensor	Normal operation (ms)	1 faulty pair (ms)	2 faulty pairs (ms)	3 faulty pairs (ms)
S1	0.4	0.4	0.4	0.4
S2	0.5	0.6	0.6	0.6
S3	0.6	0.9	0.9	0.9
S4	0.7	1.1	1.1	1.1
S5	0.4	0.4	0.4	1.4
S6	0.5	0.5	0.6	1.6
S7	0.6	0.6	0.9	1.9
S8	0.7	0.7	1.1	2.1

Latency measures

– Fault tolerance mechanisms tested and validated

– The 2 ms message latency constraint is always satisfied

6. Conclusion

New smart transducers interfaces

- Two interface designs based on IEEE 1451
- Improved reliability and safety
- Validated through a hardware prototype (FPGAs,COTS …)
- Integrated into a dual ARINC 825 bus

Future work

- Implementation and validation of the dual-chip interface architecture

Thank you for your attention