CCIPC - CANopen Controller IP Core

Microelectronics Presentation Days

Noordvijk March 30th - April 1st, 2010

Pietro Tosi – CAEN AURELIA SPACE

Topics

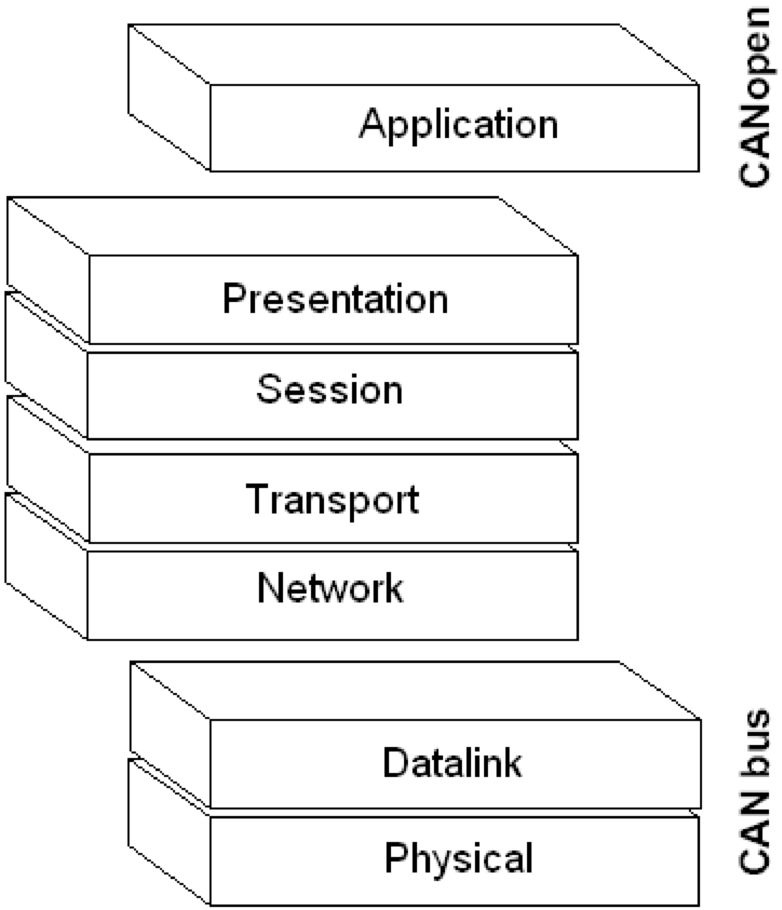
- CCIPC project
- CANopen Protocol overview
- IP core Architecture
- Implementation (preliminary) results
- Conclusions

CCIPC Project

- Development of a new IP core implementing CANbus + CANopen features
- Commissioned by Thales Alenia Italia to CAEN AURELIA SPACE
- Developed in the frame of the ExoMars Programme funded by the European Space Agency
- To be utilized in the ExoMars Entry Descending and Landing Demonstrator and Rover Modules
- To be included in the ESA IP portfolio

- CAN bus is a serial standard link defined for automotive used also in automation and space fields.
- CAN bus defined to save wiring into vehicles allows safe and robust communication on a simple twisted pair
- CAN bus defines the Physical, MAC and Logic layers of the OSI communication stack.
- CANopen defines different sub-protocols covering the Network, Transport and Application layers on top of CAN bus.
- CAN bus + CANopen offer a cheaper alternative to MIL1553 as avionic bus

CAN and ISO/OSI

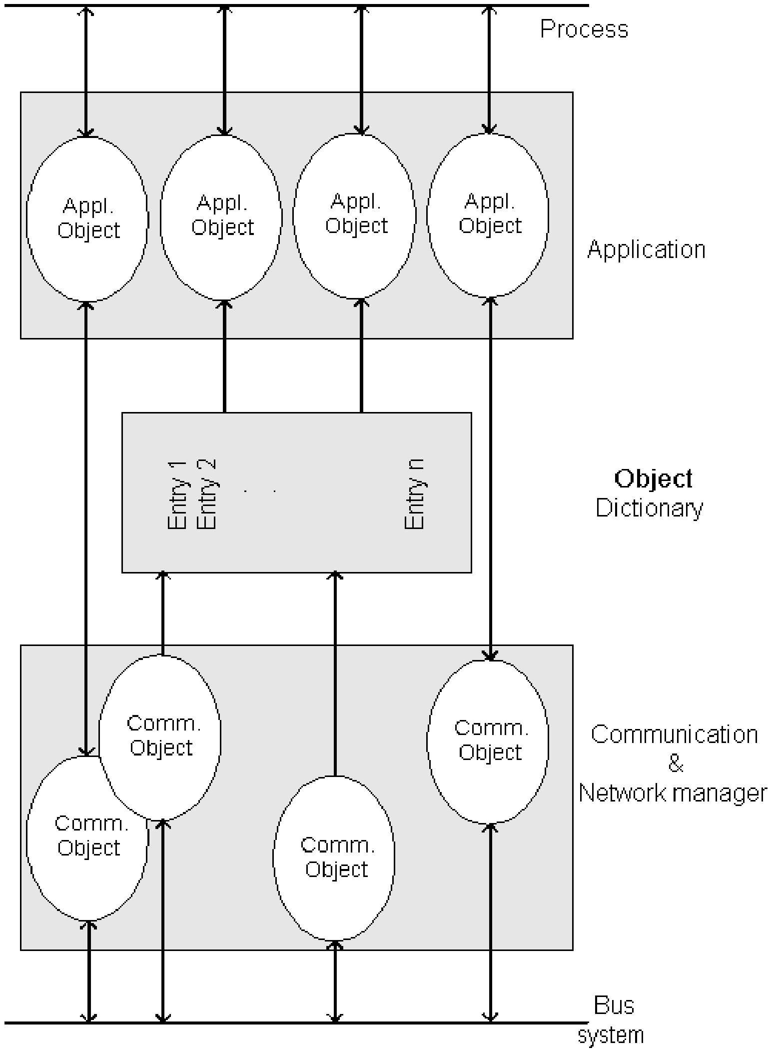


CANopen
Application layer:
- filters and decodes messages

CAN bus
Datalink layer
- checks packet
- detects and signals transmission error

Physical layer:
- transfers data bits between different nodes

Application Layer



Application
- describes the characteristic and functionality of the device

Object Dictionary
- represents the interface towards the application

Communication
- defines services to receive and transmit objects on the bus
- Network manager controls the device status

Network Management

- Node Oriented
- Follows a master-slave structures
- Executes NMT services :
  - Module Control service
  - Error Control service
  - Bootup service

Network Management

Module Control service

- NMT master controls the state of the NMT slaves:
  - Initialise Node
  - Start/Stop Node
  - Reset Node

Error Control service

- Heartbeat mechanism is used:
  - Heartbeat producers cyclically transmit message on the net
  - Heartbeat consumers listen for this message
  - If heartbeat cycle fails, CCIPC switches from current to redundant bus

Bootup service

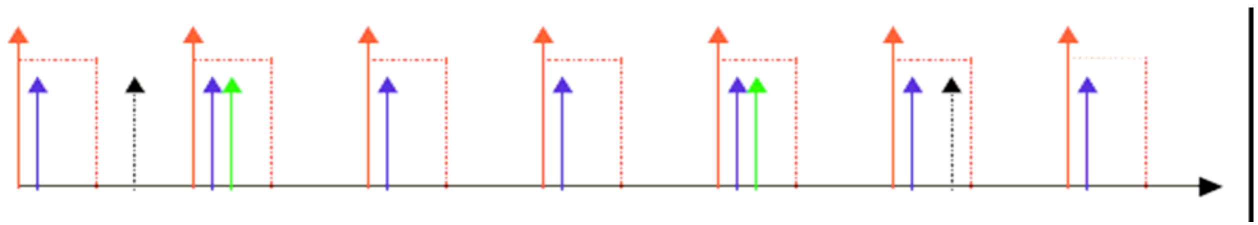
- NMT slave indicates that it is initialised

Process Data Object (PDO)

- Used to perform real-time data transfer
- Provides the interface to the Application Objects
- Two types of PDO:
  - RPDO – receive PDO
  - TPDO – transmit PDO
- PDOs are described by
  - communication parameters – describe the communication capabilities of the PDO
  - mapping parameters – contain information about the device variables

Process Data Object (PDO)

- Transmission mode
  - Synchronous: PDO are transmitted/received after reception of the SYNCH object
  - Asynchronous: PDO are transmitted/received immediately



SYNC object
PDO synchronous - cyclic
PDO synchronous - acyclic
PDO asynchronous

Service Data Object (SDO)

- Used to perform access on the Object Dictionary entries
- Used to transfer multiple data sets
- Client/server model
- SDO transfer types:
  - Segmented – SDO is transferred as a sequency of segments
  - Expedited – if data set is up to 4 bytes
- Three basic services:
  - SDO download – client downloads data to server
  - SDO upload - client uploads data from server
  - SDO abort – client/server terminates due to error

CANopen Object Dictionary CCIPC

- OD is the core of any CANopen node
- OD is an ordered LUT containing all communication parameters and application objects
- Each mapped object can be accessed by 16-bit Index and 8-bit Sub-Index

Index (hex)	Object
0000	not used
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufacturer Specific Complex Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardised Device Profile Area
A000-BFFF	Standardised Interface Profile Area
C000-FFFF	Reserved for further use

Index	Sub-Index	Field in PDO Communication Parameter Record	Data Type
0020h	0h	number of supported entries in the record	UNSIGNED8
	1h	COB-ID	UNSIGNED32
	2h	transmission type	UNSIGNED8
	3h	inhibit time	UNSIGNED16
	4h	reserved	UNSIGNED8
	5h	event timer	UNSIGNED16

- OD area containing all data objects
  - Manufacturer Specific – contains the manufacturer specific functionality data object
  - Standardised Device – contains all data objects common to a class of device

Index	SubIndex	Name	Type	Attr	Default
2A00h		ADC configuration	Record		
	0	Number of entries	U8	RO	7
	1	Number of channels	U8	RW	
	2	Conversion rate	U8	RW	0
	3	Input range	U8	RW	0
	4	Unipolar/Bipolar	U8	RW	0
	5	Power save mode	U8	RW	0
	6	Offset	U32	RW	
	7	Gain	U32	RW	

ADC configuration example

CCIPC implements a CANopen Slave node

CCIPC CANopen Supported Services:

- Network Management protocol
- Heartbeat consumer-producer with error detection mechanism
- T/R PDO transfer
- SDO transfer

CCIPC Additional Service:

- Bus Redundancy management

CCIPC shall be configurable for the following features

- Number of NodeID
- Number of T/R PDO
- Number and type of Application Objects
- Host interface type
- Object Dictionary indexing

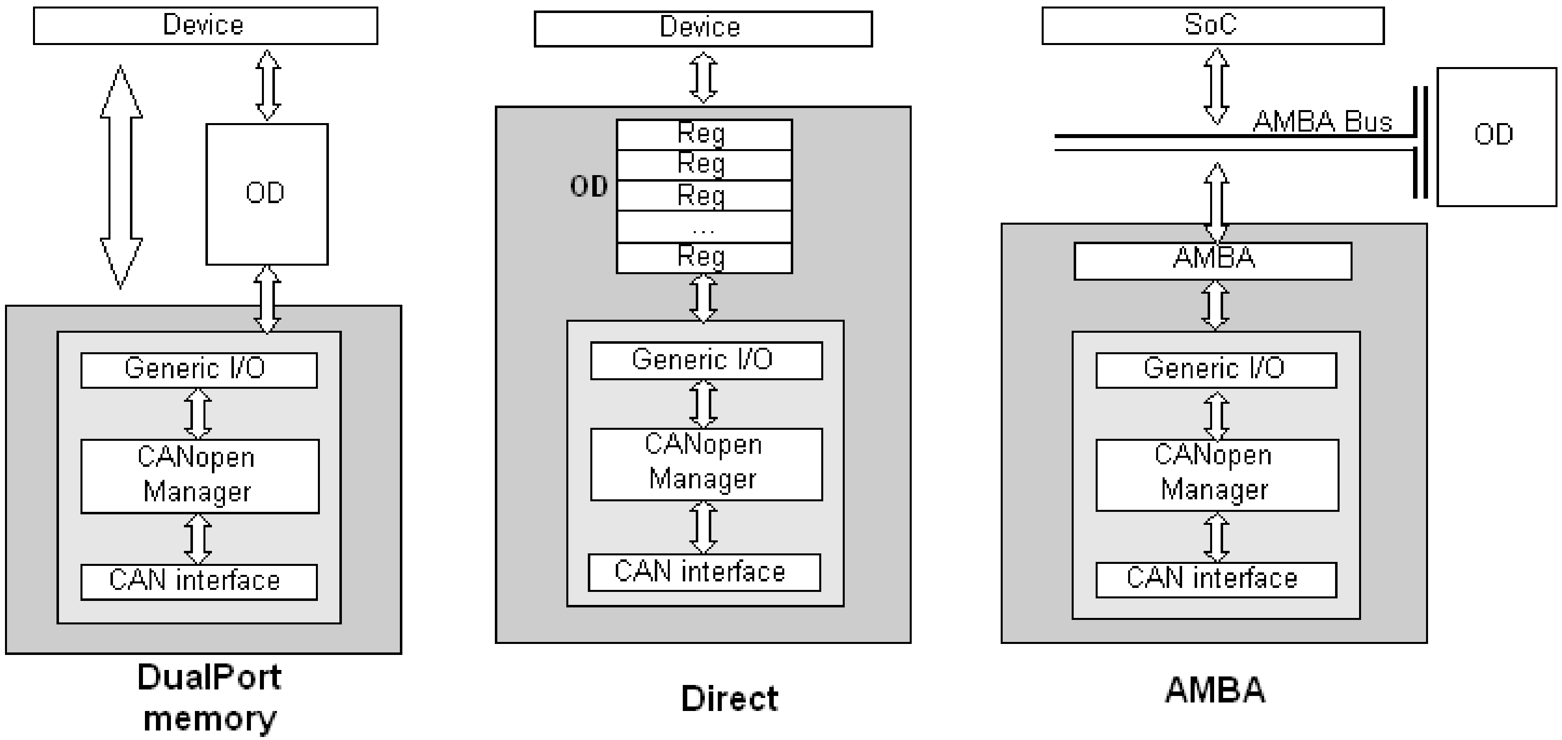
A Perl-Tk GUI shall be provided to lead the user into core instance customization

- CCIPC occupies one or more CanOpen NodeID

Base NodeID	Mask size	ID-Mask	Node ID count	Node ID range
BBB BBBB	0	111 1111	1	BBB.BBBB only
BBB BBB0	1	111 1110	2	BBB.BBB0 - BBB.BBB1
BBB BB00	2	111 1100	4	BBB.BB00 - BBB.BB11
BBB B000	3	111 1000	8	BBB.B000 - BBB.B111
BBB 0000	4	111 0000	16	BBB.0000 - BBB.1111
BB0 0000	5	110 0000	32	BB0.0000 - BB1.1111
B00 0000	6	100 0000	64	B00.0000 - B11.1111
000 0001 (1)	7	000 0000	127	000.0001 - 111.1111
(1) Node ID 0 is reserved				

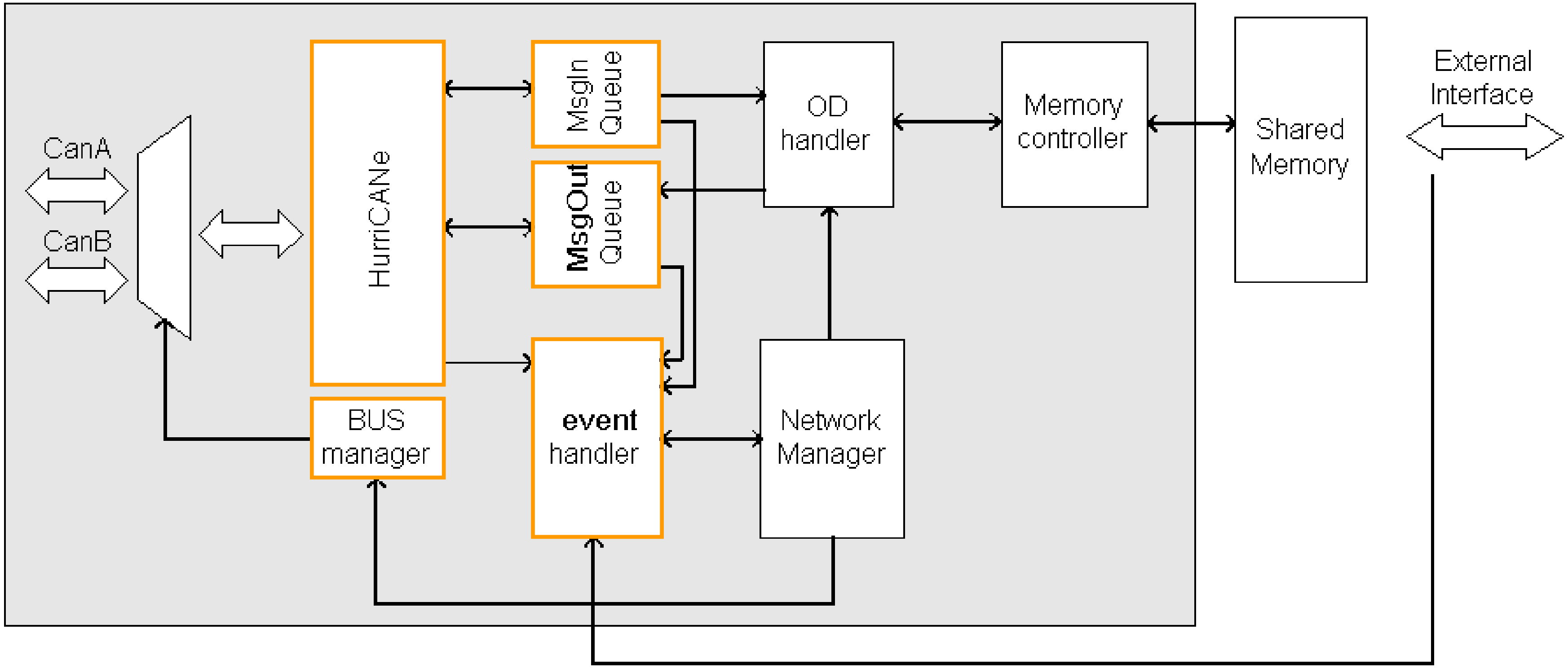
$$NodeIDrange = [BaseNodeID\ ;\ BaseNodeID + 2^{MaskSize} - 1]$$

Host Interface



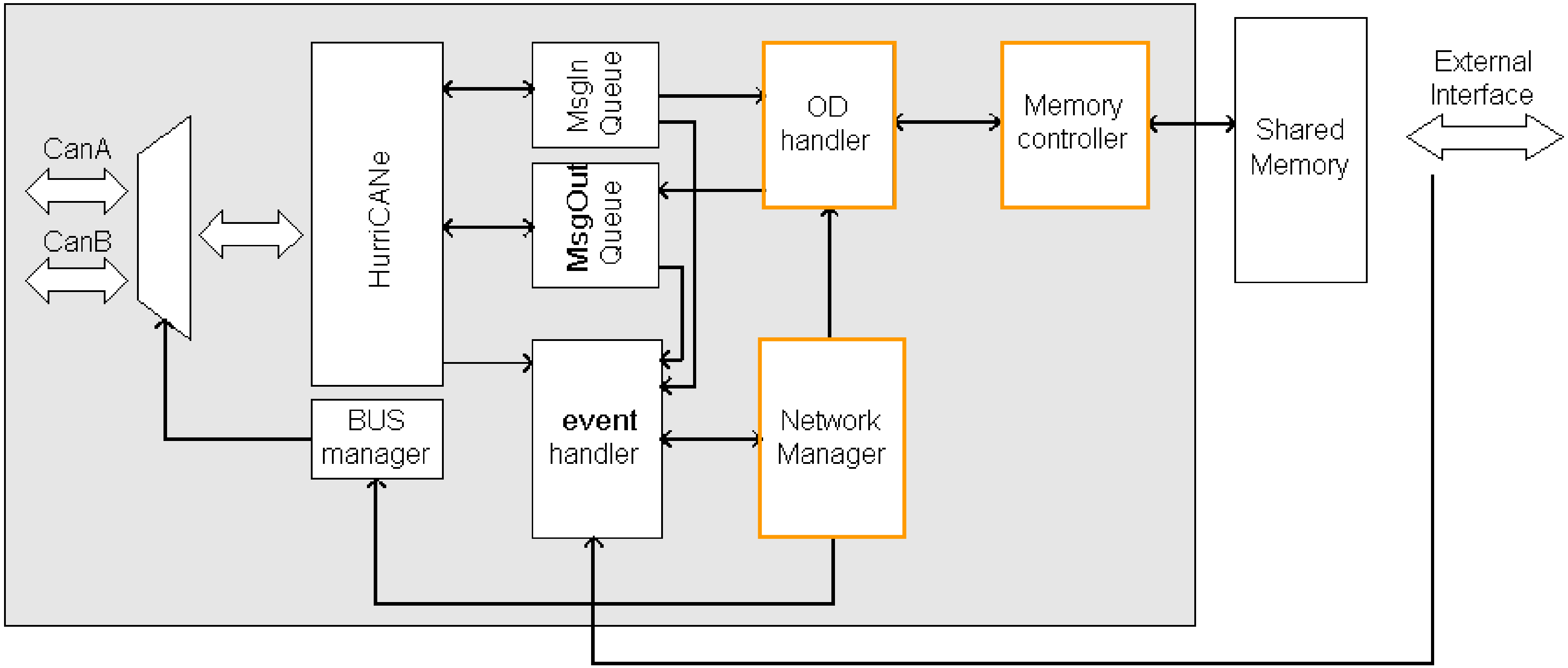
DualPort memory → OD entries accessible using simple memory interface
Direct → OD entries directly mapped into Core and connected to external device
AMBA → Complex SoC oriented with Master/Slave interface

IP Core Architecture



- BUS manager – Implements Bus redundancy features
- CAN interface – HurriCANe for Tx and Rx CAN messages
- MsgIn/MsgOut Queue – Queue of in/out message
- Event handler – Enables the different CanOpen functionality

IP Core Architecture



- OD handler – Manages OD interface and CanOpen services
- Network Manager – Manager of core boot operation and status
- Memory controller – Controls the access to the OD area

Synthesis and fitting results CCIPC

- Resource occupation
  - Close to 100% of RTAX250
  - Around 25% of RTAX1000

Block	Resource distribution (%)
Hurricane + interface	40
OD handler	31
Network Manager	6
Memory Controller	11
Others	12

- Timing reports
  - Requested frequency : 16 MHz
  - Estimated frequency : 21 MHz

Memory Resource Occupation CCIPC

Estimation of memory resources budget on RTAX1000

$NodeID_{count}$	MaskID	PDO parameters area			Device Profile area		
		CBR(1)	Byte		CBR	Byte	
			Data	EDAC		Data(2)	EDAC
8	111 1000	8	3.2k	0.8k	28	11k	3k
4	111 1100	4	1.6k	0.4k	32	12K	4k
2	111 1110	2	0.8k	0.2k	34	13k	4k
1	111 1111	1	0.4k	0.1k	35	13k	4k

(1) CBR – 512x9 core block RAM in Actel RTAX device

(2) Effective payload data depends on indexing implementation

Conclusion

- CCIPC + HurriCANe (CANopen + CAN bus) shall offer an alternative to MIL1553 in future space missions

- The IP core is designed following the ECSS CAN BUS Working Group standardization activities and therefore it will be compliant with the incoming ECSS-E-50-15C.

- IP core is expected for Q3 2010
- Engineering samples for Exomars in 2011
- The first flight is expected in 2016 with the EXOMARS Entry Descending and Landing Demonstrator Module