

## The Designing Bus for Nuclear Safety Class Controller

Dongil Lee, Myeongkyun Lee and Donghwa Yun (PONUTech) Kwangki Ryoo (HNU)



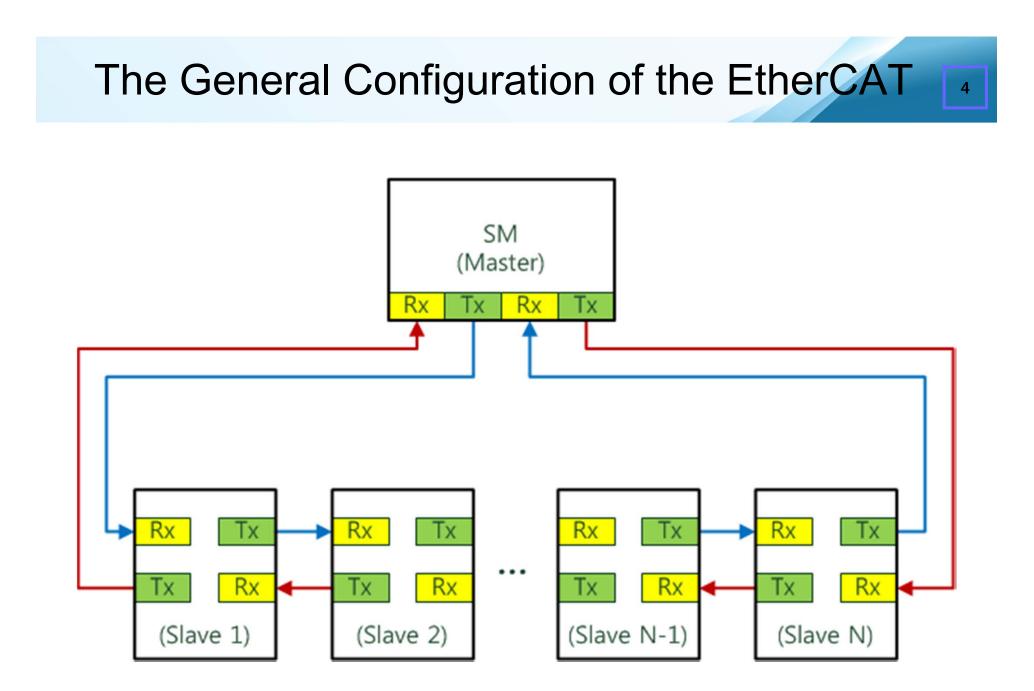
# CONTENTS

- Introduction
- The General Configuration of the EtherCAT
- Ethernet Fame with EtherCAT Data
- Controller
- Verification of performance
- Conclusion
- Reference

#### Introduction

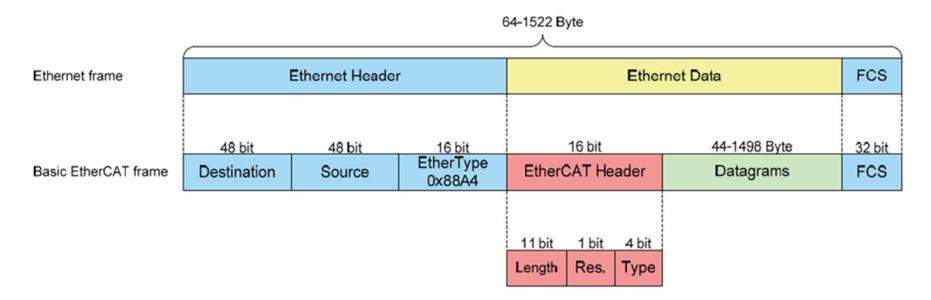
- Existing nuclear safety class controller typically used the Parallel Link. The increase of a small data communication in a result of increase in the number of I/O contacts demands to high-speed bus but it could not use some unstable problem that is switching noise, reference voltage unclear, magnetic coupling, FEXT-induced jitter and etc. in the parallel link.
- However to be shown to solve uses the E-Bus of EtherCAT widely used industrial site.
- The EtherCAT supports almost Line, tree or star topology.
- The On-the-fly processing way which can read and write data when the frame passes through in the network and the using the FMMU (Fieldbus Memory Management Unit) mapping in the 4GB (Giga Byte) space base on the hardware can process easily for a packet to each slave data.







### Ethernet Fame with EtherCAT Data



5

#### • EtherCAT Frame Header

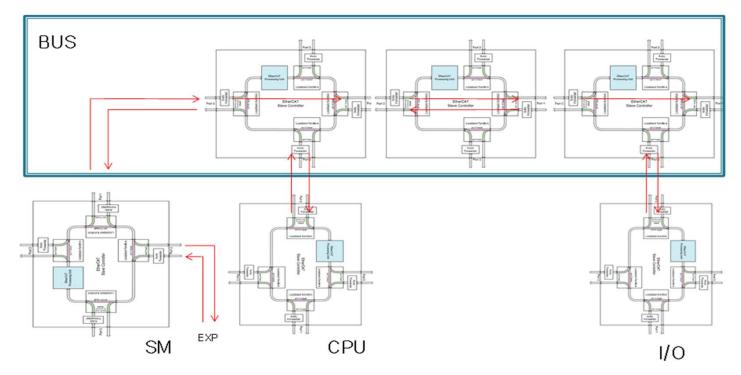
- Length
  - 11bit; Length of the EtherCAT datagrams (excl. FCS)
- Reserved
  - 1bit
- Туре
  - 4bit; Protocol type. Only EtherCAT commands (Type = 0x1) are supported by EtherCAT Slave Controller(ESC)s.



#### Controller

#### • Structure

In case of a EtherCAT Chip by ASIC, it has four (4) ports zero (0), one (1) ports are used as the default and auto-forward orders are zero (0) → three (3) → one (1) → two (2) port. In addition, each module shall install a chip in order to configure an "On-The-Fly" format a feature of EtherCAT and each slot shall setup a chip for Hot-Swap function in the backplane. And it shall implement a using three (3) of four (4) ports.





#### Controller

• Shape

**PONU**Tect

 The pin of backplane was saved and was possible designed to high speed data transaction because to design an E-bus using a LVDS line. 7



### Verification of performance

• Read information for each module after connected a computer through a RS-232.

8

• Information about the module, as shown in the following figure was confirmed by serial communication.

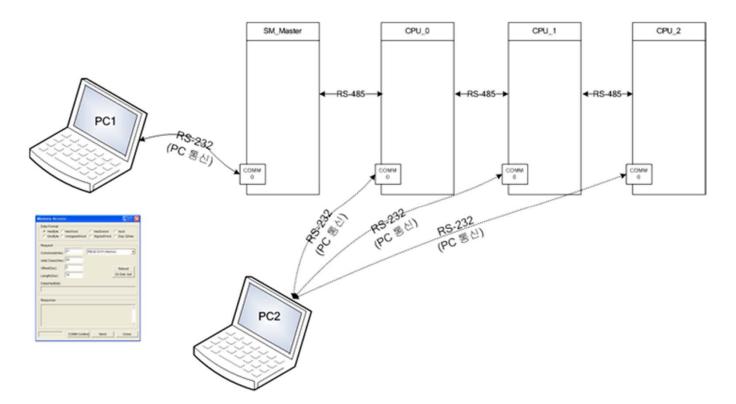
Memory Access	
Data Format C HexByte C HexWord C HexDword C Ascii C DecByte C UnsignedWord C SignedWord C Dsp 32Hex C BitByte C BitWord	
Request	
Commnad(Hex) 21 READ DATA Memory	
Addr,Class(Hex) 0006	
Offset(Dec) 0 Reboot	
Length(Dec) 16 Cir Deb. buf.	
Data(HexByte)	
Response	· · · · · ·
02 03 04 04 04 00 06 06 06 06 06 00 00 05 05 05 00 00 00 00 <u>~</u>	0 0 0 0 0
4/4 C COMM Contine Send Close	
Min,Max(Word)	



### Verification of performance

 In order to confirm to performance of bus communication such as the figure is constructed. All CPU and SM (System Manager) module connected computers and then confirmed an operation to read and to write of a data in CPU.

9





#### Verification of performance

• Each CPU in the data transmission was a constant pattern SM normally received in the data could confirm that it has. Data can also be read normally the normal bus communication was confirmed.

SM Data Write (CPU0 영역)	SM Data Write (CPU1 영역)	SM Data Write (CPU2 영역)	
Memory Access	Data Format     HexDword     Ascii       Deta Format     HexDword     Ascii       Declefke     UnsignedWord     SignedWord       BitByte     BitWord     SignedWord       Request     Commnad(Hex)     Image: SignedWord       Commad(Hex)     Image: SignedWord     Reboot       Offset(Dec)     Image: SignedWord     Cir Deb. but       DetaHeadedword     Image: SignedWord     Cir Deb. but       DetaHeadedword     Image: SignedWord     SignedWord       Offset(Dec)     Image: SignedWord     Cir Deb. but       DetaHeadedword     Image: SignedWord     SignedWord       Image: SignedWord     Image: SignedWord     SignedWord	Methody Access  Data Format  Checkyte  Hex/Word  Hex/Word  Access  Commad(Hex)  Addr,Class(Hex)  Distree  Commad(Hex)  Request  Commad(Hex)  Commad(Hex)  Request  Commad(Hex)  Request  Commad(Hex)  Report  Reboot  Cir Deb. buf.  Distree  Distree Distree  Distree Distree Distree Distree Distree Distree Distree Distree Distree Distree D	
×	CPU0 Data Read	CPU1 Data Read	CPU2 Data Read
T/T         C         Colose           Min,Max(Word)         Init         Close	Memory Accous       Image: X         Data Format       Head/word       Ascill         C BetByte       Head/word       SignedWord       Dsp 32Hex         BitByte       BitByte       BitByte       Dsp 32Hex         Commad(Hex)       21       READ DATA Memory       Image: X         Addr.Class(Hex)       0020       Reboot       Clir Deb. buf         Data(Hex)f(ee)       16       Clir Deb. buf       Data(Hex)f(ee)         Data(Hex)f(ee)       16       Clir Deb. buf       Data(Hex)f(ee)         Data(Hex)f(ee)       16       Clir Deb. buf       Data(Hex)f(ee)         Data(Hex)f(ee)       00 00 00 00 00 00 00 00 00 00 00 00 00	Windowy Addcess       Image: State Sta	Minday Add ess         Data Format         Charles Format         Commad(Heig)         21         Request         Commad(Heig)         Offset(Dec)         0         Data(Heid)(bei)         16         Charles Heightes         91 92 93 94         Response         91 92 93 94         Bible C       COMM Contine         Send       Close         Min,Max(Word)       Init

#### Conclusion

- Data of the many contacts transmission was confirmed that there is no problem by means of using bus that a nuclear safety class controller was used by the E-bus of Widely used in EtherCAT of industrial communication network.
- The transmission of quantity will have many limitations because Commercial ASIC chips 8KByte Data transfer only. In the future a FPGA implementation of EtherCAT is required in order to transfer more DATA.



#### References

- [1] BECKHOFF, ET1100 Hardware Data Sheet v1.8, 2010
- [2] EtherCAT Technology Group, EtherCAT Communication Presentation, 2011





# **THANK YOU**

**PONU-Tech Co., Ltd.**