IEICE TRANS. INF. & SYST., VOL.E102–D, NO.1 JANUARY 2019

PAPER

Design of High-Speed Easy-to-Expand CC-Link Parallel Communication Module Based on R-IN32M3

Yeong-Mo YEON[†,††], *Nonmember and* **Seung-Hee KIM**[††a)], *Member*

SUMMARY The CC-Link proposed by the Mitsubishi Electric Company is an industrial network used exclusively in most industries. However, the probabilities of data loss and interference with equipment control increase if the transmission time is greater than the link scan time of 381μs. The link scan time can be reduced by designing the CC-Link module as an external microprocessor (MPU) interface of R-IN32M3; however, it then suffers from expandability issues. Thus, in this paper, we propose a new CC-Link module utilizing R-IN32M3 to improve the expandability. In our designed CC-Link module, we devise a dual-port RAM (DPRAM) function in an external I/O module, which enables parallel communication between the DPRAM and the external MPU. Our experiment with the implemented CC-Link prototype demonstrates that our CC-Link design improves the communication speed owing to the parallel communication between DPRAM and external MPU, and expandability of remote I/O. Our design achieves miniaturization of the CC-Link module, wiring reduction, and an approximately 30% reduction in the link scan time. Furthermore, because we utilize both the Renesas R-IN32M3 and Xilinx XC95144XL chips widely used in diverse application areas, the designed CC-Link module reduces the investment cost. The proposed design is expected to significantly contribute to the utilization of the programmable logic controller memory and I/O expansion for factory automation and improvement of the investment efficiency in the flat panel display industry.

key words: *CC-Link, dual-port RAM (DPRAM), remote I/O, complex programmable logic device (CPLD), R-IN32M3*

1. Introduction

A control and communication Link (CC-Link) is an industrial open network that enables communication among devices developed by numerous manufacturers [1]. In detail, CC-Link is a system which connects remote I/O function modules, intelligent function modules, specific function modules, and so on using private line and manages these modules through the programmable logic controller (PLC) CPU. Using this CC-Link, various equipment controls for factory automation are generally linked to the fieldbus industrial network via a PLC, which is a universal controller used for automatic control and monitoring of a machine [2].

CC-Link can simultaneously control and process data, and international standards for a high-speed fieldbus network, specifically the International Electrotechnical Commission (IEC) 61158 and IEC 61784, have been established in 2014. Recently, CC-Link has been widely used in the domestic flat panel display (FPD) industry and other industries because of its high communication speed of 10 Mbps, maximum transmission distance of 100 m, and its capacity that can connect up to 64 remote I/O stations [3].

A conceptual architecture of a CC-Link is illustrated in Fig. 1. In this figure, CC-Link is used to connect master station and slave stations. The master station is the centralized control for the data link system, whereas the slave station is controlled by the master station. Each slave station can be a remote I/O, sensor transformer, a human machine interface (HMI), a robot or one of various types of controllers.

The expandable CC-Link field network must be developed to flexibly respond to industrial equipment fieldbuses. The term “expandable”, means that several modules or devices of the slave stations can be communicate with the master station via the CC-link. However, the dedicated chips of Mitsubishi, Hilscher, and Renesas have vendor-specific limitations in terms of expandability.

The LSI MFP3N, the dedicated communication chip of Mitsubishi, specifically requires a host microcontroller (MCU). Its user manual must be purchased, and the company has a quarterly pricing policy, which makes commercialization very difficult. Meanwhile, developing user applications with the Hilscher netX chip is difficult and time intensive. Thus, a loadable firmware (LFW) is used, and the interface is employed through dual-port memory communication requiring a host MCU. Managing product releases also becomes difficult when the LFW is updated. On the other hand, Renesas’ R-IN32M3 chip with a built-

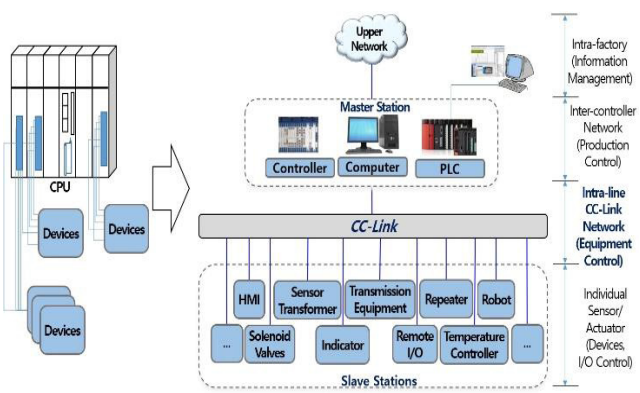

Fig. 1 Conceptual architecture of CC-Link

Manuscript received March 21, 2018.
Manuscript revised August 24, 2018.
Manuscript publicized October 9, 2018.

[†]The author is with METIS, 69–83 Eungwon 1 gil, Mokcheon-eup, Dongnam-gu, Cheonan City, Chungnam Province, 31224 Republic of Korea.
[††]The authors are with Korea University of Technology and Education, 1600 Chungjeol-ro, Byeongcheon-myeon, Dongnam-gu, Cheonan City, Chungnam Province, 31243 Republic of Korea.
a) E-mail: sh.kim@koreatech.ac.kr

DOI: 10.1587/transinf.2018EDP7101

in hardware real-time OS is an application-specific standard product (ASSP) consisting of a slave-only communication large-scale integration (LSI) of the field network and ARM Cortex-M3. We do not need to buy a separate license, so we can easily create designs using the above-mentioned these chips. Moreover, it does not require an additional MCU, and various communication protocols are supported. However, because the above products occupy two stations during input and output processes, they cannot use the memory areas of input (4bytes), output (4bytes), RWw (8bytes) and RWr (8bytes) simultaneously. As a result, the above products require more resources as the number of slave station increases. The reliability of communication significantly decreases on account of a 90% or higher data loss and a communication delay of 6 ms or longer when the I/O is expanded through the universal asynchronous receiver/transmitter (UART). This is based on a UART communication speed of 115,200 bps and a PLC master communication speed of 10 Mbps [4], [5].

As mentioned above, the previous CC-Link modules have constraints in terms of the board type and expandability based on the 1 : 1 mapping of the interfaces between the IC and external devices of slave stations. This is a severe restriction for the application scope of the PLC expands. To alleviate the above-mentioned problems, we propose a design for an expandable CC-Link slave module herein. As an expandable module, the remote I/O module is designed to support multiple remote I/O devices that enable the selection of multiple inputs and outputs. For verifying the feasibility of the design, we implemented it in the Renesas R-IN32M3 and Xilinx XC95144XL chips.

The contributions of our work are summarized as follows.

- We guarantee the expandability, usability and compatibility among the heterogeneous chips using external interface architecture.
- By adopting a parallel function, the communication speed of I/O devices is improved.
- We can add the slave devices non-sequentially.
- The number of physical communication and power cable lines for connection is decreased.

The paper is organized as follows: In Sect. 2, we explain module and link scan time measurement for the CC-Link communication. In Sect. 3, we propose the design of an expandable CC-Link parallel communication module using R-IN32M3 in detail. Section 4 verifies the communication performance of the new CC-Link module using the prototype that implements the proposed design. Finally, we describe results and their implications in Sect. 5.

2. Background

In this section, we describe the characteristics of the CC-Link communication module. In addition, we discuss in detail the measurement equation of the link scan time for CC-Link communication applied to measure the performance and reliability of the proposed design.

Table 1 Characteristics of ICs produced by different companied [6]

Vendor	IC Name	Max Stations	Characteristics
Mitsubishi	MFP2N	1	•Software development is not necessary •Only one station is supported; no expandability •Difficult to build a development environment
	MFP3N	1-4	•Devices can be developed without considering the protocol •MCU is required for external parallel communication •Difficult to build a development environment
Hilscher	NetX Series	1-4	•Profile for remote devices is provided •Simple control with a host interface by a dual port memory •Difficult to develop an ARM9 user application •A separate MCU is needed to use a host interface by a dual port memory
Renesas	R-IN32M3	1-4	•ASSP chip with embedded communication LSI and Cortex-Me •No separate CPU is necessary; various communication protocols are supported

2.1 CC-Link Communication Module

Both the Hilscher and HMS chips provide a board-type CC-Link communication module and a dual-port RAM (DPRAM) parallel interface for external communication. The DPRAM interface design for remote I/O has had a decisive impact on performance. The interface of the existing module is designed inside the module for fast data communication. However, this design lacked expandability because it only supported one-to-one communication with external devices. Table 1 presents the characteristics of different ICs. Some of the currently developed chips can occupy up to four stations. However, in practice, when a maximum of four stations is used, a total of 256 I/O ports having 128 inputs and 128 outputs required for normal data communication have not yet been developed. This is because permanent I/O modules can communicate one-to-one only with permanent devices.

As for the expandability of the module, Joo [7] suggested the RS-485 protocol and controller area network (CAN) communication in the field programmable gate array (FPGA), which is a semiconductor logical device containing internal lines enabling design and programming as another kind of various remote I/O configurations. Therefore, it is different from the expandable type, and the communication interface is not specified. D. Zhang [8] designed a high-speed parallel interface based on ARM & FPGA to solve the meta stability problem of asynchronous data using

Table 2 Standard transmission rates

Transmission rate	156K	625K	2.5M	5M	10M
BT	51.2	12.8	3.2	1.6	0.8

Table 3 NI value per final station number

Final station number	1~8	9~16	17~24	25~32	33~40	41~48	49~56	57~64
NI	8	16	24	32	40	48	56	64

the on-chip DPRAM in FPGA. Such work indicates that our approach for exploiting expandable CC-Link parallel communication in designing the module is feasible.

In this study, we used an R-IN32M3 chip for CC-link communication and a Xilinx XC95144XL chip for slave stations. The R-IN32M3 chip is used to resolve the data loss problem and speed delay of parallel communication, and the Xilinx XC95144XL chip is used in the I/O control for the slave station. In other words, this chip is designed to enable the addition of slaves as well as data upload and download thus enabling high-speed processing [9]–[11]. XC95144XL-10TQ100C from Xilinx is selected for the parallel interface, DPRAM design, and I/O control. The complex programmable logic device (CPLD) XC95144XL chip is easy to develop and consists of programmable logical blocks and buses that connect them [12]. The CPLD module in the system has 144 macro control units, 3200 logic gates, a 5 ns pin-to-pin logic delay, and a 178 MHz system frequency [13]. In addition, the binary file can be downloaded even with no EEPROM [14]. We particularly select the CPLD chip herein because of its advantage of operating immediately when the system reboots [13].

2.2 Link Scan Time Measurement for CC-Link Communication

The CC-Link communication link scan time consists of a cyclic transmission (periodic communication in the same network) between the CC-Link master station, where the data link system is controlled, and the remote station. The latter, which is controlled by the master station, is generally a remote I/O station and the unique voice of remote devices. The link scan time (LS) of the remote I/O network mode is calculated by the conventional method using Eq. (1) [15] as follows:

$$LS(\mu s) = ① BT\{27 + (② NI \times 4.8) + (③ N \times 30) + (④ ni \times 4.8)\} + ⑤ ST + ⑥ RT + ⑦ F \quad (1)$$

① *BT*: Transmission rate (*bps*), as given in Table 2.
② *NI*: Last station number (It must be a multiple of 8, given in Table 3)
③ *N*: Number of connected stations excluding reserved stations
④ *ni*: Total number of occupied stations of a, b and c types excluding reserved stations
a: Total number of occupied stations for remote I/O stations.
b: Total number of occupied stations for remote device stations.
c: Total number of occupied stations for local stations, standby master, and intelligent device stations
⑤ *ST*: Scan Time, Constant
$ST = 250 + (ni\ x\ 10)$
⑥ *RT*: Retransmission processing time only when a transient request is made
$RT = \alpha + \beta \times (number\ of\ detected\ faulty\ stations - 1)$
α: Retry processing time of the first station
$\alpha = BT \times (778.5 + (13.2 + (NI \times 4.8)) \times 3)$
β: Retry processing time of the second and subsequent stations
$\beta = BT \times (778.5 + P \times 3)$,
where P is 10.8
⑦ *F*: Return processing time only when there is a fault station (i.e.; a module that cannot link data by power off, etc.; it returns to a normal state and accesses the internet)
$F = BT \times 243.1 + ST$

3. Design and Implementation

In this section, we first present the principal concepts of the proposed design and then to demonstrate the feasibility of our proposed device, we designed the prototype using the R-IN32M3 and XC95144XL module design, protocol design, sequence diagram of the MCU, and programming of the CPLD module.

3.1 Principal Concepts of the Proposal Design

CC-Link has a memory area for input, output, RWw, and RWr in each occupied station. However, a module wherein we can use the total memory area at the same time has not been developed yet. Therefore, if we configure remote inputs and outputs using existing modules, these can be used with two occupied stations. This is important cause for the memory shortage of the master station during expansion. For solving these problems, we propose an expandable CC-Link slave module with external interface architecture. In detail, we designed not only remote I/O module that enable the choice of multiple inputs and outputs but also a rotary switch supporting the giving of a slave ID for each I/O module.

Figure 2 shows an example of the proposed expandable remote I/O conceptual design, where the IN/OUT function is deployed with 32 points per module, and R-IN32M3 is implemented to provide the DPRAM interface function. Furthermore, this design provides parallel communication, a special feature in implementing the DPRAM function with a CPLD chip for easy expansion.

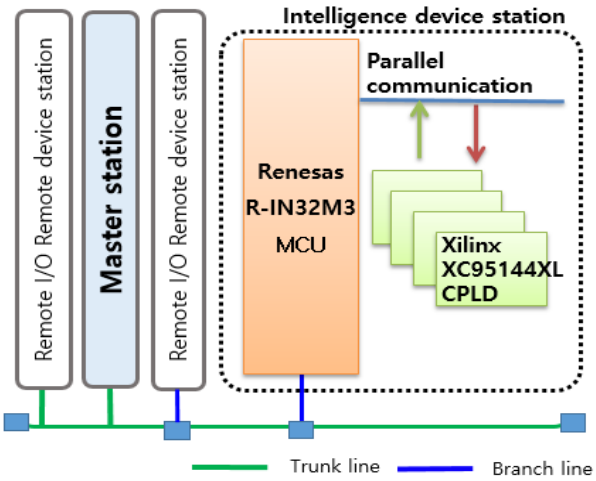

Fig. 2 Proposed conceptual design (block diagram)

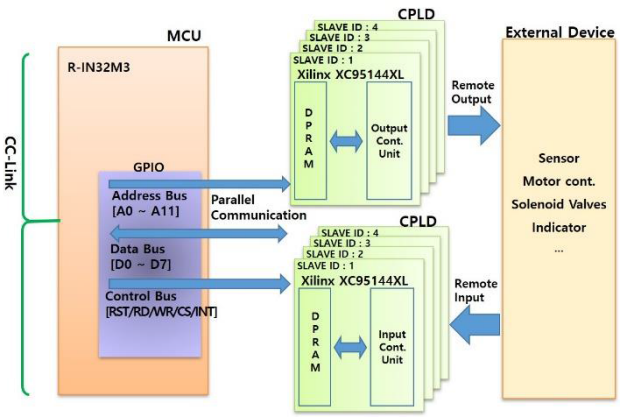

Fig. 3 Circuit diagram of proposed module

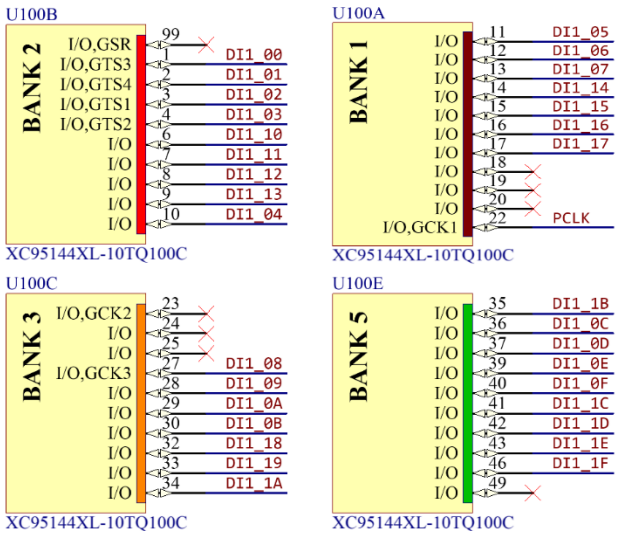

Fig. 4 32-point control pin for the external I/O control of XC95144XL

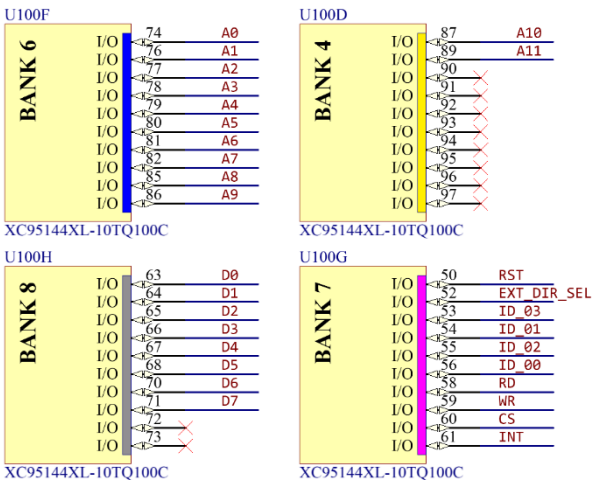

Fig. 5 Parallel interface and ID setting pin

3.2 R-IN32M3 and XC95144XL Module Design

The PCB module for the R-IN32M3 chip was designed with four or more layers for the hardware test design with the BGA324 pin, offering the advantages of development cost savings and easy maintenance. The peripheral circuits of the MCU were configured with a reset circuit, X-Tal circuit, serial flash for booting, and JTAG for debugging in an on-board form to minimize the PCB module size.

In addition, the LED, station number setting, station count setting, speed setting, and dedicated communication line were designed to be connected to the external device for the implementation of the DPRAM interface and the CC-Link function as shown in Fig. 3. In other words, the proposed product design can transmit and control in real time the RX (16 bytes), RY (16 bytes), RWw (32 bytes), and RWr (32 bytes) of the CC-Link memory to change and expand to a flexible remote I/O and remote device for industrial sites.

The DPRAM interface cannot be used for purposes other than parallel communication if only the address data, write, and read pins for the external MPU interface of the R-IN32M3 chip are linked to the external connected terminal. Thus, the utilization of the alternate function of the I/O port has improved by connecting even general GPIOs to the parallel communication line.

The remote I/O circuit configuration in Fig. 4 consists of a circuit for the 32-point input and output control of I/O signals. The configuration in Fig. 5 comprises a rotary switch for the address setting, 12 address pins (i.e. A0~A11) for the interface between R-IN32M3 and DPRAM, eight data pins (i.e. D0~D7) and control pins (i.e. RST, RD, WR, CS, and INT) [16].

3.3 Protocol Design

Figures 6 and 7 outline the protocol for an accurate data exchange in the communication of the R-IN32M3 (i.e. MCU) and XC95144XL (i.e. CPLD) modules. The operation sequence of the protocol for the data input in Fig. 6 can be summarized as follows:

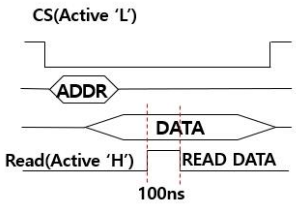

Fig. 6 Internal data reading timing from DPRAM

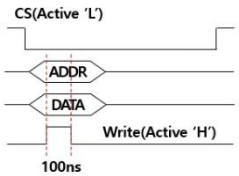

Fig. 7 Internal data writing timing to DPRAM

① The R-IN32M3 module sends a CS signal to the XC95144XL module as a low active signal.
② The R-IN32M3 module sends the address number to be read in the address bus line to the XC95144XL module.
③ The R-IN32M3 module finally sends the RD signal to the XC95144XL module.
④ The XC95144XL module sends data to the R-IN32M3 module through the data bus line.
⑤ The R-IN32M3 module returns the CS signal to a high state when the data reception is completed.

Conversely, the protocol operation sequence to the output data in Fig. 7 can be outlined as follows:

① The R-IN32M3 module sends a CS signal to the XC95144XL module as a low active signal.
② The R-IN32M3 module sends the address number to be written on the address bus line to the XC95144XL module.
③ The R-IN32M3 module sends the data value to be written in the data bus line to the XC95144XL module.
④ The R-IN32M3 module sends a WD signal to be written on the XC95144XL module.
⑤ The R-IN32M3 module restores the CS signal to the high state when the data transmission is complete.

The address for data transmission to the XC95144XL module multi-connected from the R-IN32M3 module was configured as 12-bits, shown in Fig. 8. The α 4-bits were used for comparison depending on the XC95144XL module function. The β 4-bits determined the XC95144XL module sequence. Finally, the γ 4-bits were defined as the address for accessing the internal memory of the XC95144XL module.

3.4 Flow of MCU

Figure 9 shows the flowchart of main () function running on the R-IN32M3 chip. The main task begins as soon as power

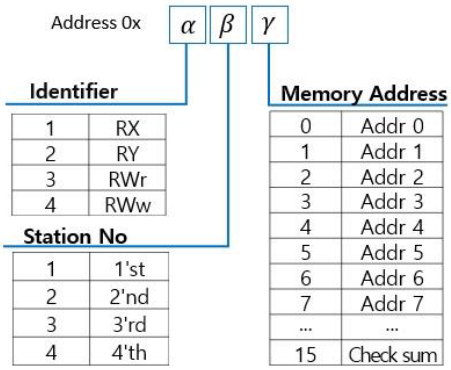

Fig. 8 Address definition

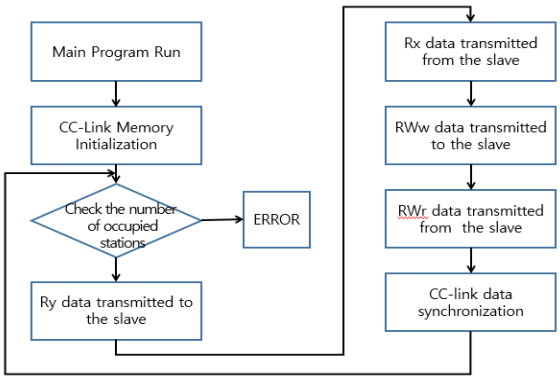

Fig. 9 R-IN32M3 main function flowchart

is supplied. At this time, the number of occupied stations is set by the rotary switch input value for initializing and expanding the register for the CC-Link function. The RY data transmission to the XC95144XL module, RX data reception, RWw data transmission, and RWr data reception are then performed after the communication connection with the CC-Link master. Finally, the data are synchronized with the CC-Link master, and the process is repeated for the number of occupied stations. The cycle for reading the RX, RY, RWr, and RWw data in four stations was programmed to allow the main task to read and write register values in real time and minimize data loss and delay during synchronization with the CC-Link master.

3.5 Programming of CPLD Module

The CPLD XC95144XL module was developed using the VHDL language in the ISE v14.7 [17] environment. In detail, the XC95144XL module was implemented as follows: the CS signal is first checked; the DPRAM data are output as a data signal if the read signal is Rising Edge; and the input data are then sent to R-IN32M3.

```
always(@posedge Read)
begin
if (CS == 0)
```

```
  begin
  DATA[7:0] = DPRAM[ADDR]
  end
end
```

The input data are saved in the DPRAM of the XC95144XL module and are processed as a data output if the write signal is Rising Edge.

```
always(@posedge Write)
begin
if (CS == 0) then
  begin
  DPRAM[ADDR] = DATA[7:0]
  end
end
```

4. Performance Verification Using Prototypes

In this section, we discuss empirical experiments using prototypes that were implemented reflecting the design explained in Sect. 3.

4.1 Test Environment and Scenario

We downloaded the R-IN32MC-E3 source from Renesas' website. In addition, we constructed a development and test environment to compile the IAR Embedded Workbench for ARM v7.60.

Figure 10 shows the constructed test environment. The data signals between the R-IN32M3 module and XC95144XL module were measured. The same environment in Fig. 11 was constructed with Q12HCPU for the CPU and QJ61BT11N for the master to evaluate the data loss and delay. The master parameters were set to 10 Mbps and the occupation of four stations. The GPIO that could simultaneously output signals with the CS, WD, and RD signals of the DPRAM interface of R-IN32M3 with the MSOX3054A oscilloscope was set with the debugger pin. The measurement was then conducted.

The green dashed box in Fig. 11 is the developed prototype tested with four input modules and four output modules connected to it. The testing instrument was configured with the GOT1000 Touch LCD, Q61P POWER, Q12HCPU, and QJ61BT11N master module.

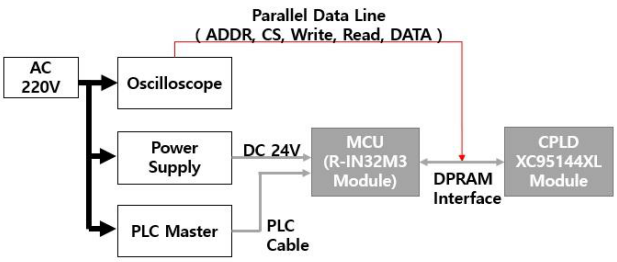

Fig. 10 Construction of the test environment

The data input and output directions for the input bit data RX, output bit data RY, input word data RWr, and output word data RWw were defined by identifying the address for each data memory map. The time required for the DPRAM communication was measured by adding a 1-byte check sum in each section to improve reliability.

Our experiments were performed on a GUI that had been designed to allow individual data changes and enable detailed testing. RX, RY, RWr, and RWw functions were verified on the touch monitor using the CC-Link master test equipment.

If the results of the measurements obtained after performing the experiment are less than LS, then it would mean that both modules are successful in terms of technical performance and expandability of scale.

4.2 Experimental Results

We used the oscilloscope to measure the elapsed time of the RY as output bit data and that of the RX as input bit data. The communication times of stations 1–4 for RY was measured by setting a spare pin of R-IN32M3 as the debugger terminal. As shown in Fig. 12, the total elapsed time for all data through the parallel communication used in the GPIO of R-IN32M3 and the total data transmission and reception

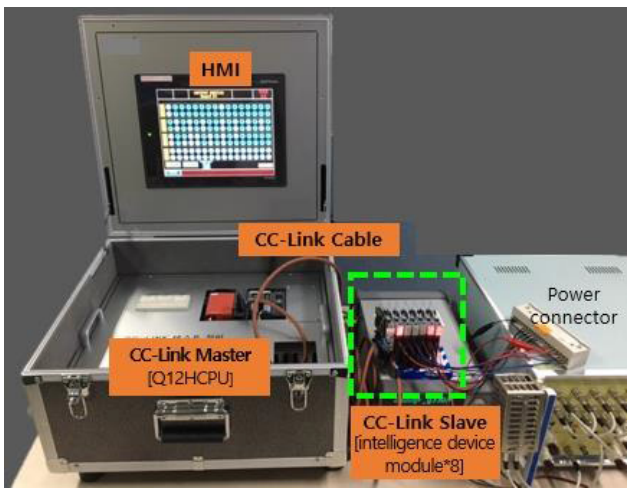

Fig. 11 Test environment

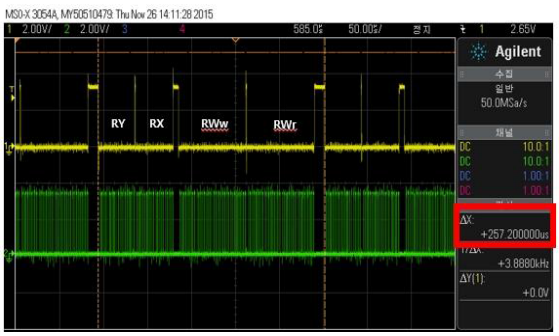

Fig. 12 Total elapsed time in one frame (257μs)

Table 4 Maximum mean time of internal communication from one to four expandable configurations

Device No.	Data Stations	RY (4)	RX (4)	RWw (4)	RWr (4)	TOTAL (μs)
1	TIMING	42.2	48.6	77.6	90	258
2	TIMING	42.1	48.7	78	89.2	258
3	TIMING	42.2	48.7	77.5	89	257
4	TIMING	42	48	78	89	257

time was 257μs.

In Table 4, we represented the maximum mean time of internal communication from one to four expandable configurations of each 4-CPLD module per device.

Summarizing the above-mentioned verification results, the same operation characteristics were maintained, regardless of the test data, as given in Table 4, because it was designed to operate according to the internal clock. Examining the data in more detail, the total transmission and reception were completed within 257μs through 258μs when the signals for each data section were measured by replacing the control memory data of the CPU task with the GPIO pin and implementing parallel communication.

We verified the accuracy of our experiment using the measurement for the link scan time of CC-Link communication in Sect. 2.2. When the transmission rate is 10Mbps, assuming that there is no faulty station, data retransmission is no longer required. Using Eq. (1), the link scan time of the network mode is calculated as follows:

$$LS\ [\mu s] = BT\{27+(NI\times 4.8)+(N\times 30)+(ni\times 4.8)\} + ST + RT + F$$

As the transmission rate (BT) is 10 Mbps, it is 0.8 bps according to Table 2. Because the number of connected modules (NI) is the result of one to four final stations, we compute for eight connected stations which are multiples of eight (Refer to Table 3). The number of connected modules (N) is one and the number of occupied stations (ni) is four.

$$ST = 250 + (ni \times 10) = 250 + (4 \times 10) = 290$$

If there is no error, RT and F both become 0. Therefore, we may get the value of LS as follows:

$$LS = 0.8\{27+(8\times 4.8)+(1\times 30)+(4\times 4.8)\}+290+0+0 = 381.68\mu s$$

External communication must be possible in the PLC communication cycle for lossless data communication. Therefore, for an expandable remote I/O transmission, R-IN32M3 must enable the transmission of 48 bytes and the reception of the same value, 48 bytes, but at a minimum rate of 381.68 μs. This experimental result confirms that lossless communication is possible in the PLC link scan time (381.68 μs) even with the GPIO. Moreover, the speed can be improved by at least 30%. The technical performance and expandability of the proposed design were verified through tests with prototypes. The PCB module for the R-IN32M3 chip was designed with four or more layers for the hardware test design with the BGA324 pin, offering the advantages of development cost savings and easy maintenance.

4.3 Implications and Discussion

Industrial expert recognizes how difficult it is to expand and retrofit add-ons (a component or accessory) providing new features into an existing product. In other words, it is necessary to add many sensor modules to various I/O types at the time of expansion, and every time one sensor module is added, one PLC input card can be added to it. In the worst case, when the base module needs to be expanded, the base model itself must be replaced, which causes a major problem in terms of the cost and time consumed in wiring [18]. The Mitsubishi product lacks expandability and has become the pioneer in raising the need for research in this area. Nevertheless, the gateway communication modules produced by Hilscher and HMS also have problems, such as data delays and losses.

Using the R-IN32M3 development module, the communication of the GPIO with the external MCU in parallel communication is possible. Furthermore, it can be used as UART, SPI, I2C, etc. when the function is changed to an alternate one.

The experiments proved that these problems could be resolved through a combination of the recently released R-IN32M3 module from Renesas and the XC95144XL module when the proposed design was applied.

The module developed in this study makes it possible to simplify the installation of various kinds of modules and parts in the construction of complex equipment such as conveyor lines and machinery, thus saving wiring, time, and costs of the whole system because the price of Renesas R-IN32M3 is less than $30. Thus, the proposed design has a competitive edge in terms of cost compared to Hilscher's module-type COMX 10CA-CCS and IC-type NEXT50 or Mitsubishi's MFP2N. In addition, with intelligent devices such as the RS-232C interface module, data communication can be performed at high speeds; thus, it is very simple and useful for extending the transmission distance when constructing a control facility requiring high-complexity real-time service. Ultimately, from the perspective of a company that needs to produce more modules, it can be expected to improve the productivity of corporations by helping achieve stability and scalability with low additional costs.

5. Conclusion

CC-Link products are exclusively supplied by Mitsubishi Electric; however, these products have a configuration limited to factory automation. New communication modules that can offer higher investment efficiency in the FPD industry as a whole and enable the use of the master PLC memory and flexible I/O expansion for factory automation should be developed to address this problem. Therefore,

we proposed a universal CC-Link parallel communication design in this study through a GPIO that allows the use of an alternate function with the Renesas R-IN32M3 and Xilinx XC95144XL chips, whose supply has recently been increasing. The proposed design enables not only a random expansion and the use of a remote I/O station, but also a parallel communication between the DPRAM and external MPU through optimization of the DPRAM interface. In our experiment using the implemented prototype, the link scan time of our design is 257 μs, which is 124.68 μs faster compared to the conventional design. Moreover, the proposed CC-Link module can use the current chips as they are and save wiring, time, and costs. Therefore, the proposed design is expected to significantly improve the general investment efficiency; likewise, CC-Link module can also offer a performance improvement effect while using the Renesas R-IN32M3 and Xilinx XC95144XL chips as they are, as the proposed module will maximize the use of PLC memories and I/O expansion in factory automation and can be applied to various complex products.

This study was conducted based on parallel communication. Hence, to further increase applicability, a multilayer design may be unavoidable because of the many patterns in the PCB design. Further research on new hardware designs is required to minimize the number of pins in serial communication consisting of ENABLE, LATCH, DO, DI, and CLK pins.

References

[1] http://en.wikipedia.org, accessed Jan. 2018
[2] H.J. Kim, K.C. Lee, and S. Lee, "Development of DSP based EtherCAT slave controller for performance evaluation of industrial robot control," Conference Proc. Korean Society of Precision Engineering (KSPE), pp.455–456, May 2013.
[3] Y.M. Lee and K.S. Shu Hsu, "Using CC-Link industrial communication on the implementation of motor HMI monitoring in lab practice," Proc. 2nd International Conference on Innovation, Communication and Engineering (ICICE), pp.357–360, Oct. 2013.
[4] T. Gomes, P. Garcia, S. Pinto, J. Monteiro, and A. Tavares, "Bringing Hardware Multithreading to the Real-time Domain," IEEE Embedded Syst. Lett., vol.8, no.1, pp.2–5, March 2016.
[5] J.J. Labrosse, "White paper: Hardware-accelerated rtos: c/os-iii hw-rtos and the r-in32m3," http://ieeexplore.ieee.org/document/7287750/, accessed Aug. 8, 2017.
[6] CC-Link Partner Association, "Development guide of CC-Link-compatible products," CC-Link Partner Association, Korea, pp.23–24, June 2014.
[7] J.H. Joo, "The implementation of flexible digital IO module appling FPGA technology," Conference Proc. Korean Institute of Electrical Engineers, pp.344–345, Nov. 2010.
[8] Z. Daode, Y. Pan, and X. Hu, "Design of high-speed parallel data interface based on ARM & FPGA," JCP 7.3, 804-809, 2012.
[9] D.-S. Kim, H.-J. Chung, and Y.-H. Lee, "Design of general peripheral interface using serial link," Journal of Korea Institute of Information, Electronics, and Communication Technology, vol.4, no.1, pp.68–75, March 2011.
[10] T. Song, M. Song, and Y. Chung, "A study on the high-speed communication interface with virtual modem," The Institute of Electronics Engineers of Korea — Semiconductor and Devices, vol.44, no.1, pp.84–89, Jan. 2007.
[11] J.-T. Hong, D.-H. Park, and Y.-H. Yu, "A Study of NMEA 2000 Protocol Application for Ship Electrical Power Converter Monitoring System," Journal of the Korean Society of Marine Engineering, vol.35, no.2, pp.288–294, March 2011.
[12] B.J. Kim, J.C. Park, J.G. Shon, B.K. Lim, and H.J. Jeon, "The development of CPLD controller for reducing harmonics of 3 phase diode rectifier," Journal of the Korean Institute of Illuminating and Electrical Installation Engineers, vol.14, no.3, pp.43–48, May 2000.
[13] J.J. Rodriguez-Andina, M.J. Moure, and M.D. Valdes, "Features, Design Tools, and Application Domains of FPGAs," IEEE Trans. Ind. Electron., vol.54, no.4, pp.1810–1823, 2007.
[14] K. Liping, Z. Zhefeng, and X. Gang, "The Hardware Design of Dual-mode Wireless Video Surveillance System Based on DM6437," 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing, Wuhan, Hubei, pp.546–549, 2010.
[15] Mitsubishi Electric Corporation, "Melsec-q cc-Link system master/local module user's manual," Mitsubishi Electric Corporation, Japan, SH (NA)-080394E-N, pp.369–370, March 2015.
[16] S. Szachowalow, G. Jablonski, and D. Makowski, "Software for data acquisition AMC module with PCI express interface," 16th Int. Conf. Mixed Design of Integrated Circuits & Systems, 2009, MIXDES'09, pp.145–148, June 2009.
[17] Xilinx press releases and device data sheets, http://www.xilinx.com, accessed March 13, 2017.
[18] D.K. Kwak, "A study on development of disaster prevention automation system on IT using One-chip Type PLC," The Transactions of Korean Institute of Power Electronics, vol.16, no.4, pp.97–104, April 2011.

Yeong-Mo Yeon received a bachelor's degree at Chungju National University. He studied in the IT convergence SW engineering department of Korea University of Technology and Education. Currently, he is working at METIS as the research director. His major interest areas consist of computer and sensor networking.

Seung-Hee Kim received a bachelor's degree at Dongguk University, a master's degree at Yonsei University, and doctoral degrees in Industry Information Systems from the IT Policy School at Seoul National University of Science & Technology. Currently, she is an assistant professor at Korea University of Technology and Education. Her major interest areas consist of embedded systems, software quality engineering, and optimization.