ICIC Express Letters
Part B: Applications
Volume **13**, Number **7**, July **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **681–688**

PLC FUNCTION BLOCK CREATION FOR MINIMIZING DATA LINK LIMITATION ON CC-LINK-BASED CONVEYOR HANDLING SYSTEM

Sawai Pongswatd, Patipak Niyomtamarat, Sart Kummool* and Teerawat Thepmanee

School of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang, Bangkok 10520, Thailand
{ sawai.po; 60601029; teerawat.th }@kmitl.ac.th; *Corresponding author: sart.ku@kmitl.ac.th

Received November 2021; accepted January 2022

Abstract. *In order to solve a facing network capacity problem when improving an existing conveyor handling system controlled by programmable logic controllers (PLCs) on master-slave control and communication link (CC-Link) network at a wooden pallet manufacturer, this article presents a software technique for short-term solution. For new requirements to upgrade a pallet sortation to be automated by installing new robotic handling unit, a slave PLC using an interface module modeled FX2N-32CCL (or slave PLC1) cannot receive all specified data of pallet production models from an existing master PLC using intelligent module modeled QJ61BT11N in one communication cycle due to limited data links. To overcome this limitation, a PLC function block creation based on data sorting approach is proposed. Two new function blocks named FB_SEND and FB_RECEIVE are created based on IEC61131-3 standard for adding to the existing ladder diagrams, which execute on the master PLC central processing unit (CPU) and slave PLC1 CPU, respectively. Experimental results verify that the created function blocks can provide data transfers correctly between the specified sources and destinations. The data received at the slave PLC1 CPU are similar to the data sent from the master PLC CPU. With the proposed solution, the desired conveyor handling system can be upgraded according to user's requirements.*
Keywords: CC-Link, Data link, Function block, Master-slave configuration, Programmable logic controller

1. **Introduction.** A programmable logic controller (PLC) is one of automation controllers offering cost-effective solutions for various industrial manufacturers. As a consequence of its real-time capabilities, the PLC plays an integral part in automatic monitoring and control systems [1,2] as well as mobile platform-based remote monitoring [3]. Nowadays, a variety of open communication protocols are available to enable interchange of PLC input and output data such as Modbus [4], process fieldbus (PROFIBUS) [5], and control and communication link (CC-Link) [6]. Moreover, with digital network communications benefits, a multi-PLC control system based on master-slave configuration can also be implemented [7]. Generally, a master PLC not only supervises production processes but also controls and monitors a set of slave PLCs that control production activities in the manufacturer. This implies that communications interface modules are significant components for providing successful data transfers among the master and slave PLCs connected to the network. Therefore, hardware sizing and selection of the right communication modules are required for successful engineering project execution.

During PLC project execution for upgrading an existing CC-Link based conveyor handling system of a wooden pallet manufacturer in Thailand, a serious problem of network capacity occurred on data links between the master PLC and one of slave PLCs used. In

DOI: 10.24507/icicelb.13.07.681

this article, a solution by using PLC programming to overcome this problem is presented. Two function blocks are created as new subroutine programs for adding the existing control programs. In this article, there are five sections including this introduction. Section 2 describes the studied conveyor handling system, which is required to improve by installing new robotic handling unit. Section 3 and Section 4 give details of the proposed function block creation and results of conducted experiments, respectively. Lastly, Section 5 provides the conclusions and possible future work.

2. **Studied CC-Link-Based Conveyor Handling System.** Figure 1 shows a diagram of the studied CC-Link-based conveyor handling system, which consists of five assemble units for pallet production with different models. The assembly units named ST1, ST2, ST3, ST4, and ST5 are individually controlled by the slave PLCs named PLC2, PLC3, PLC4, PLC5, and PLC6, respectively. The GT1455-QTBD touchscreen communicated with the slave PLC through RS232 interface is installed at each assembly unit for allowing an assembly operator to set specified operational parameters and perform motor jogging operation as well as for providing relevant information and production instruction that the operator needs to monitor. An operator workstation running human machine interface (HMI) 'Indusoft SCADA' software allows plant personnel not only to set desired pallet production models for all assembly units but also to monitor operating statuses and log important data and activities. The data transfers between the operator workstation and the slave PLCs are accomplished through the master PLC, which consists of the Q03UDE central processing unit (CPU) and the QJ61BT11N interface module. Each slave PLC consists of the FX3UC CPU and the FX2N-32CCL interface module. Based on CC-Link protocol, the QJ61BT11N and FX2N-32CCL modules function as the master station and remote device station, respectively. The FX2N-32CCL module can send (or receive) bit and word data to (or from) the QJ61BT11N module by utilizing cyclic transmission method. A transmission speed of the studied CC-Link network is set to 156 kbps. To increase productivity and reduce labor cost by upgrading the produced pallet sortation to be automated, a robotic handling unit controlled by the slave PLC1 is required for new installation. Moreover, the GT1455-QTBD touchscreen communicated with the slave PLC1 through RS232 interface is used to allow the operator to perform robot jogging operation, set desired storage units, and monitor defined parameters and operation sequences. The STGE1, STGE2, STGE3, STGE4, and STGE5 storage units are installed for storing finished pallets produced from the ST1, ST2, ST3, ST4, and ST5 assembly units, respectively. Figure 2 illustrates a data transfer diagram for upgrading the studied conveyor handling system. The function of software interface between the operator workstation and the master PLC is based on 'OPC' client-server architecture. The master PLC sends specified pallet models to the slave PLC2, PLC3, PLC4, PLC5, and PLC6 for production at the ST1, ST2, ST3, ST4, and ST5, respectively. Conversely, the actual statuses of sequential assembly operations as well as the number of finished pallets are sent from the slave PLC2-PLC6 to the master PLC. For the manufacturer's requirements of the new installation of the desired robotic handling unit, the specified pallet models for ongoing production at all five assembly units are transmitted from the master PLC to the slave PLC1 to automatically move finished pallets to the associated storage units. On the other hand, the actual statuses of robotic handling unit operations and the number of finished pallets stored in each storage unit are sent from the slave PLC1 to the master PLC. In addition, the data received by the master PLC can be also monitored on the operator workstation. A bit string is used to reflect the connection status of the CC-Link modules as well as the operating status of hardware and software components associated with the assembly and robotic handling units. For example, the 'active' state (1) reflects the 'Normal' or 'OK' operating status, while the 'inactive' state (0) reflects the 'Abnormal' or 'NG' operating status. The data size for storing the number

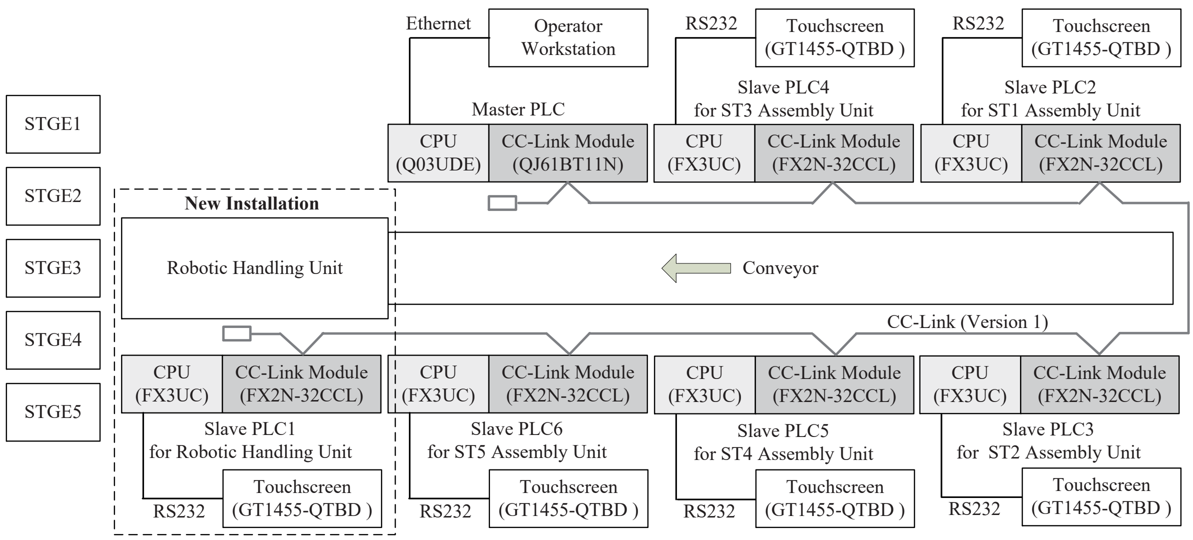


FIGURE 1. Studied conveyor handling system at a wooden pallet manufacturer

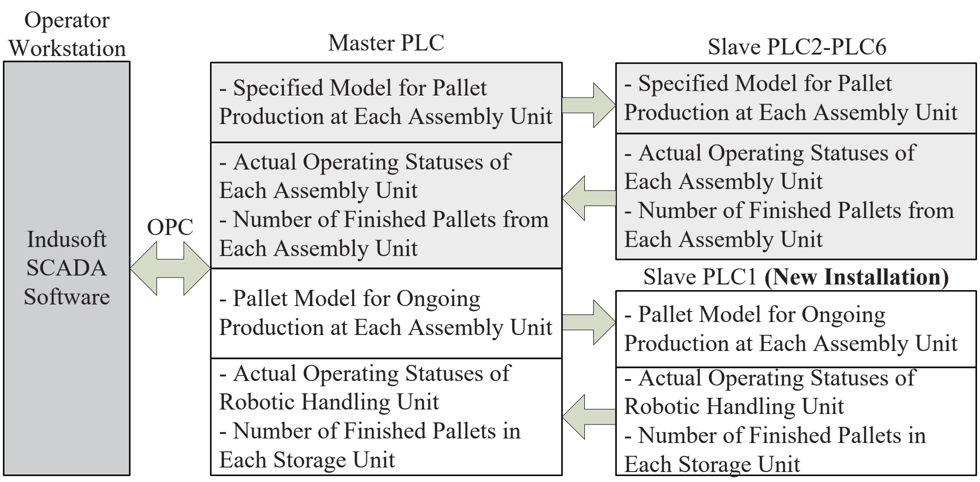


FIGURE 2. Data transfer diagram for upgrading the studied conveyor handling system

of finished pallets produced from each assembly unit is defined on 32 bits, while the data size for storing the number of finished pallets stored in each storage unit is defined on 4 bits. The pallet models are identified by utilizing American standard code for information interchange (ASCII) encoding in 4-word format, e.g., 3P5075G2 and 3P1007Z4.

Figure 3 shows a concept of data links between the master and slave PLCs for cyclic communications of numeric word data when setting the number of occupied stations for the FX2N-32CCL module to be equal to 1, four 'Write' remote registers (RWw0-RWw3) for receiving word data from the QJ61BT11N memory addresses in range of W1000-W1003 as well as four 'Read' remote registers (RWr0-RWr3) for sending word data to the QJ61BT11N memory addresses in range of W00-W03 are available. In case of setting the maximum number of occupied station for the FX2N-32CCL to be 4, sixteen 'Write' remote registers (RWw0-RWwF) and sixteen 'Read' remote registers (RWr0-RWrF) are available. However, for word data links between the master PLC and the slave PLC1, twenty 'Write' remote registers are required to receive all five pallet models for ongoing production at the ST1-ST5 assembly units. Therefore, a severe limitation on network capacity of the studied system occurs when installing the FX2N-32CCL module for the slave PLC1, which uses for controlling the desired robotic handling unit. This means that other appropriate

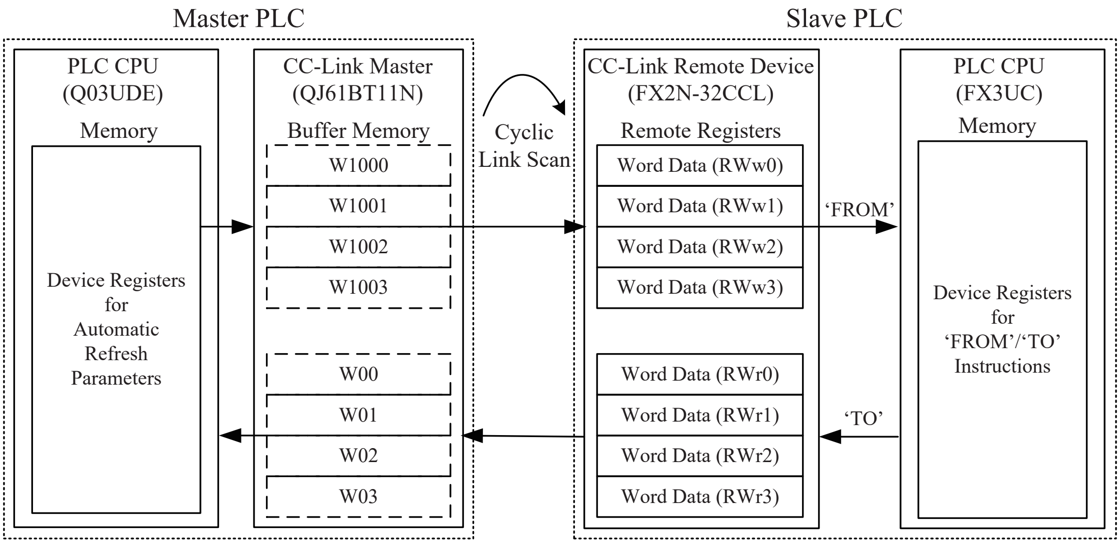


FIGURE 3. Concept of word data links between the master and slave PLCs

models of CC-Link remote devices are required to replace the FX2N-32CCL module used for the slave PLC1. However, the hardware replacement after finished installation has significant influence over the entire PLC control programming during engineering phase for the studied system. However, this replacement method may be a long-term solution for the manufacturer when revamping the studied handling system.

3. **Proposed PLC Function Block Creation.** To provide a short-term solution without bringing effects on existing control programs for the facing problem, the PLC function block creation based on data sorting approach is proposed. Two function blocks named FB_SEND and FB_RECEIVE are created as new subroutine programs to add the existing ladder diagrams running in the master PLC CPU and the slave PLC1 CPU, respectively, when setting the FX2N-32CCL module with single occupied station (see Figure 4). The repetitive executions of these created function blocks provide transmission of 20 words of five pallet models for ongoing production at the ST1-ST5 assembly units from the device registers in range of D00-D19 of the master PLC CPU to the device registers in range of D100-D119 of the slave PLC1 CPU. Table 1 gives the master PLC and slave PLC1 CPUs. For each data transmission cycle, the buffer memory address of W1000 of the QJ61BT11N module is specified for sending the 'Number_loop' parameter ($= 1, 2, \ldots, 7$) to indicate the number of times of the FB_SEND block executions in one 'FOR' instruction loop, while the buffer memory addresses in range of W1001-W1003 are specified for sending the 'DATA1'-'DATA3' parameters of three words of the pallet model. The data stored in the W1000-W1003 addresses of the QJ61BT11N module are transmitted to be recorded in the RWw0-RWw3 remote registers of the FX2N-32CCL module, respectively. Based on the 'From' instruction, these data are then moved to be processed by executing the FB_RECEIVE function block to transfer the 'DATA1'-'DATA3' to be stored in the specified device registers of the FX3UC CPU. Figure 5 shows a flowchart diagram for executing the FB_SEND and FB_RECEIVE function blocks, which are created to minimize the data link limitation between the master PLC and the slave PLC1. For complete transmission of all 20 words of five pallet production models, the number of repetitive executions of the FB_SEND block for running one 'FOR' instruction loop is set to 7. For example, based on the FB_SEND block executions, if 'Number_loop' is 1, then the data stored in the D00-D02 registers of the Master PLC CPU are moved to be recorded in the W1001-W1003 addresses of the QJ61BT11N module, respectively. After data transmission on the CC-Link network, the data in the RWw0 register of the FX2N-32CCL module are

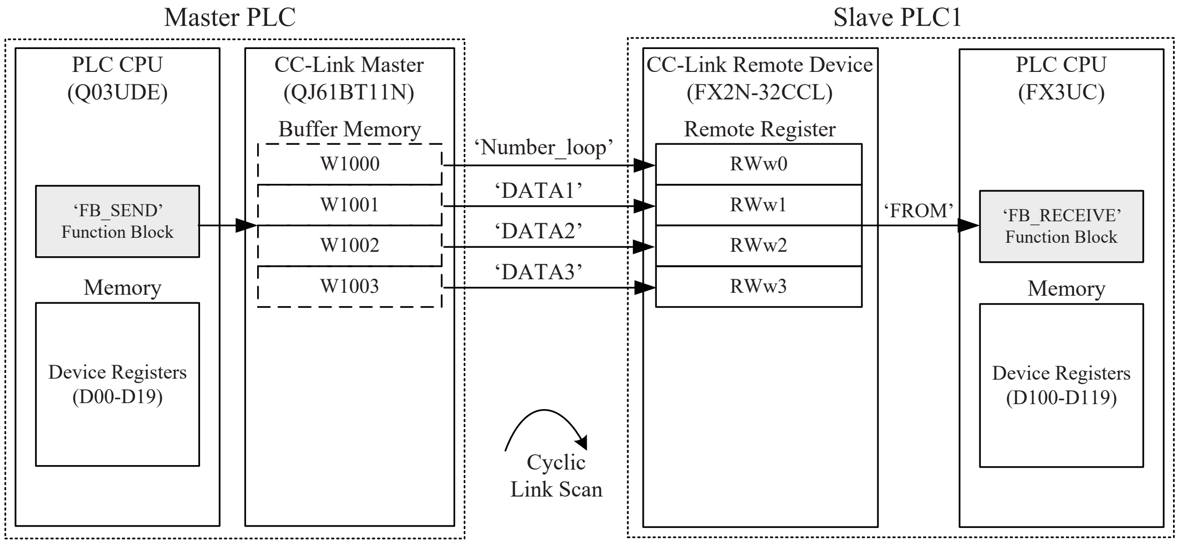


FIGURE 4. Concept of the proposed function block creation

TABLE 1. Device register assignment for the master PLC and slave PLC1 CPUs

Assembly unit	Device registers for recording pallet models for ongoing production	
	Master PLC CPU (source)	Slave PLC1 CPU (destination)
ST1	D00-D03	D100-D103
ST2	D04-D07	D104-D107
ST3	D08-D11	D108-D111
ST4	D12-D15	D112-D115
ST5	D16-D19	D116-D119

equal to 1, and the data in the RWw1-RWw3 are the same as that sent from the W1001-W1003 addresses, respectively. Based on the FB_RECEIVE block executions, the data in the D00-D03 registers in the slave PLC1 CPU are equal to the data in RWw0-RWw3 registers, respectively. In case of D00 = 1, then the data in the D100-D102 registers in the slave PLC1 CPU are the same as that sent from the D00-D02 registers in the master PLC CPU. Table 2 summarizes the descriptions of input and output parameters of two created function blocks, whereas Table 3 summarizes the specified sources and destinations for data transfers provided by executing the FB_SEND and FB_RECEIVE blocks for the 'Number_loop' values in range of 1-7.

4. **Experimental Results.** To verify the workability of two created function blocks, experiments for data transfers of pallet production models from the master PLC to the slave PLC1 were carried out. Figures 6 and 7 show the results for data transmission of five pallet models, which are 3P10007Z, 3P05324H, 3P05075A, 3P02011G, and 3P05075G for production at the ST1, ST2, ST3, ST4, and ST5, respectively. In the first step from conducted experiment, these specified pallet models were entered via the HMI screen at the operator workstation (see Figure 6(a)). In the second step, the data recorded in the D00-D19 (or D0-D19) device registers of the master PLC CPU and the D100-D119 device registers of the slave PLC1 CPU were captured from the MELSOFT GX Work2 program in online monitor mode (see Figures 7(a) and 7(b)). In the last step, the results shown on the touchscreen at the robotic handling unit were captured from the MELSOFT GT Designer3 program (see Figure 6(b)).

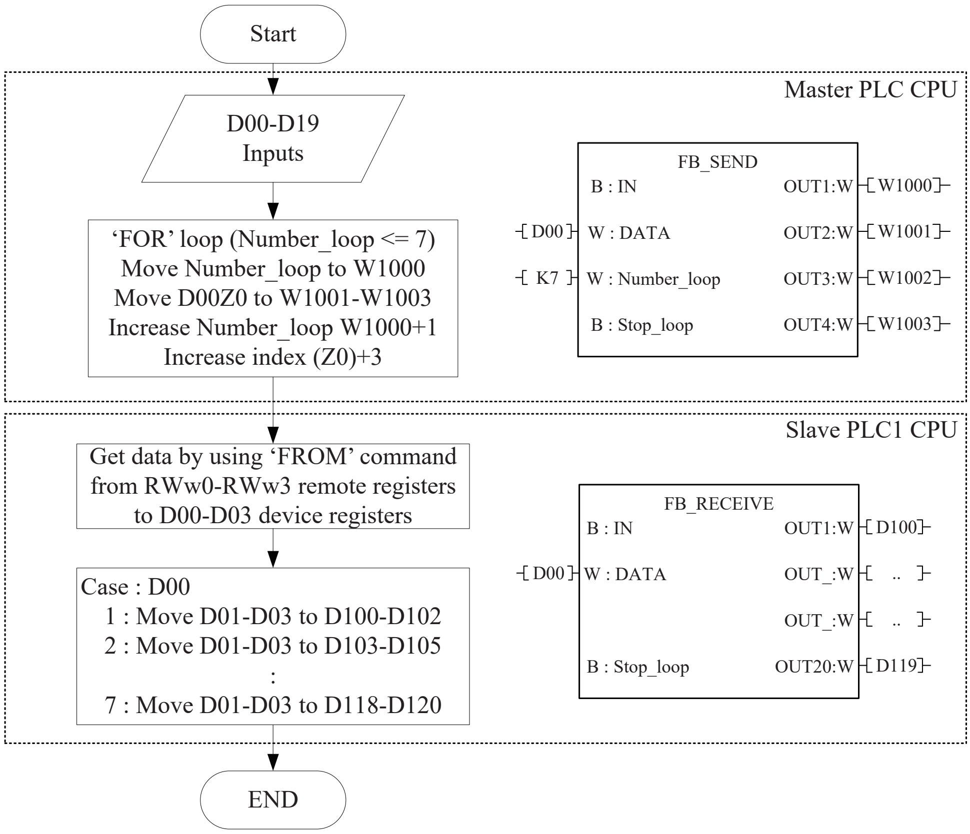


FIGURE 5. Flowchart diagram for executing two created PLC function blocks

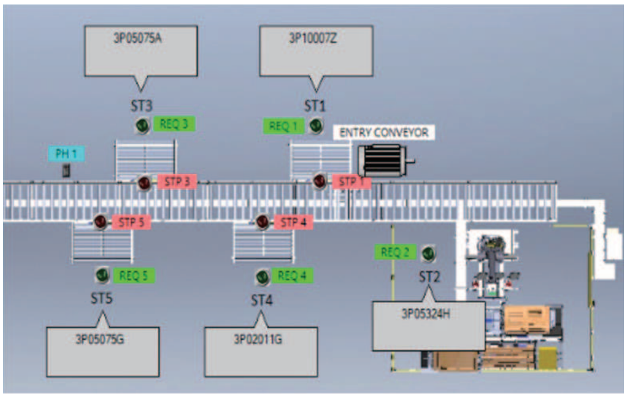
TABLE 2. Descriptions of input and output parameters of two created function blocks

Created block	Parameter	Description
FB_SEND	IN	Bit input for starting block executions
	DATA	Word input for the beginning of three device registers, which are required to send their data
	Number_loop	Word input for the number of repetition for running 'FOR' loop
	Stop_loop	Bit input for stopping block executions
	OUT1-OUT4	Word outputs for sending out to desired destination by using 'MOV' instruction
FB_RECEIVE	IN	Bit input for starting block executions
	DATA	Word input gotten from RWw0-RWw3 remote registers by using 'FROM' instruction to record in D00-D03 device registers
	Stop_loop	Bit input for stopping block executions
	OUT1-OUT20	Word outputs for transfer to desired specified device registers by using 'MOV' instruction

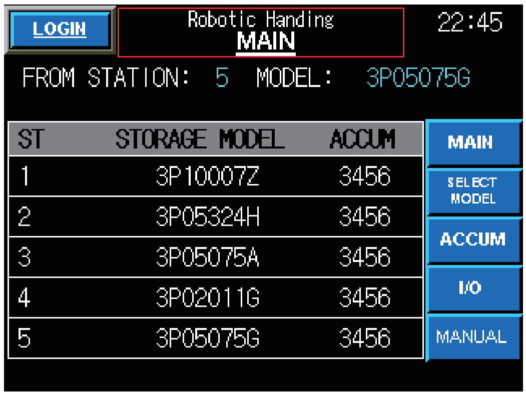
The desired pallet production models displayed in Figure 6(a) are the same as that in Figure 6(b). In addition, the similarities between the data recorded in the D00-D19 registers as shown in Figure 7(a) and the data recorded in the D100-D119 registers as shown in Figure 7(b) also support the specified sources and destinations for transmission as given in Table 3. The results of comparison not only between the data shown on Figures 6(a) and 6(b) but also between the data shown on Figures 7(a) and 7(b) confirm

TABLE 3. Specified sources and destinations for data transfers

Sources for the FB_SEND				Destinations for the FB_RECEIVE			
Number_loop (OUT1)	W1001 (OUT2)	W1002 (OUT3)	W1003 (OUT4)	D00 (RWw0)	D01 (RWw1)	D02 (RWw2)	D03 (RWw3)
1	D00	D01	D02	1	D100	D101	D102
2	D03	D04	D05	2	D103	D104	D105
3	D06	D07	D08	3	D106	D107	D108
4	D09	D10	D11	4	D109	D110	D111
5	D12	D13	D14	5	D112	D113	D114
6	D15	D16	D17	6	D115	D116	D117
7	D18	D19		7	D118	D119	

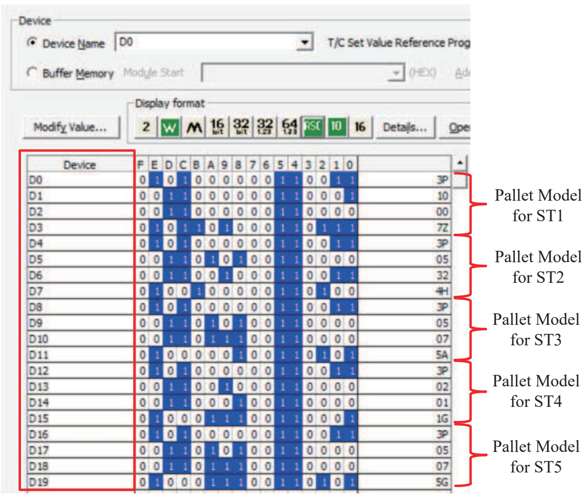


(a) HMI screen at the operator workstation

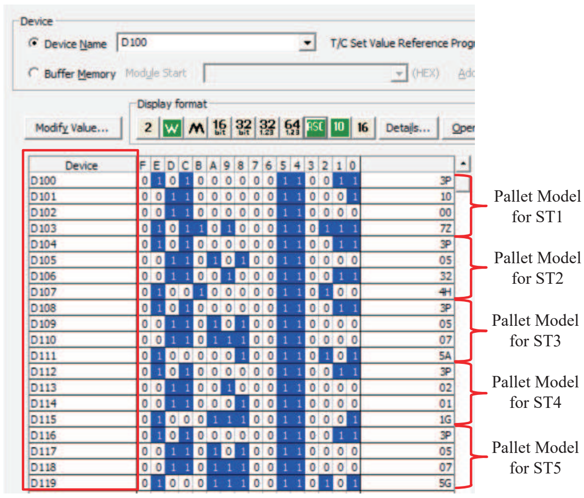


(b) Touchscreen at the handling unit

FIGURE 6. Test results from monitoring the operator workstation and the handling unit



(a) Master PLC CPU (source)



(b) Slave PLC1 CPU (destination)

FIGURE 7. Test results recorded in the registers of the master PLC and slave PLC1 CPUs

that the created FB_SEND and FB_RECEIVE blocks can function correctly. Thus, the network capacity problem due to data link limitation can be solved with the proposed function block creation.

5. **Conclusions.** A short-term solution for solving the network capacity problem of the studied CC-Link-based conveyor handling system for wooden pallet production has been proposed. The function block creation based on data sorting approach for providing data transfers of required pallet production models from the master PLC to the interested slave PLC, used for controlling the new robotic handling unit for automatic finished pallet sortation, has been described. Experimental results used to confirm the workability of two created PLC function blocks has been also presented. Based on the created function blocks, the data link limitation can be minimized. Applying the function block creation concept to other slave PLCs in the studied system as well as other industrial automation systems for providing overall equipment effectiveness (OEE) data collections is the future work.

REFERENCES

[1] J.-H. Guo, K.-L. Su and B.-Y. Li, Development of a PLC based robot arm, *ICIC Express Letters, Part B: Applications*, vol.6, no.3, pp.637-642, 2015.

[2] S. Pongswatd, K. Smerpitak and S. Kummool, HMI implementation using radar chart for dual-loop system, *ICIC Express Letters*, vol.13, no.8, pp.735-742, 2019.

[3] A. Z. C. Magalhães, A. B. Lugli and T. C. Pimenta, Study and implementation of a real industrial application involving concepts of Industry 4.0, *ICIC Express Letters, Part B: Applications*, vol.11, no.7, pp.623-630, 2020.

[4] L. Chanka, K. Smerpitak, P. Julsereewong and A. Julsereewong, Integration of weighing indicator with proprietary commands into PLC-based system using Modbus protocol, *Proc. of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, pp.265-270, 2017.

[5] F. R. Silva, A. B. Lugli, J. P. C. Henriques and M. M. D. Santos, Automation integrated using PROFIBUS and supervisory system, *International Journal of Innovative Computing, Information and Control*, vol.12, no.6, pp.2027-2039, 2016.

[6] K. Fang and Y. Tian, Touch screen monitoring system based on CC-Link, *Proc. of 2013 International Conference on Computational and Information Sciences*, Shiyang, China, pp.1893-1896, 2013.

[7] K. Smerpitak, W. Jearnpanitpong, A. Julsereewong and T. Thepmanee, Multi-PLC control system based on wireless bridge/base stations for work-in-process movements in corrugated box manufacturer, *Proc. of 2018 18th International Conference on Control, Automation and Systems*, Gang Won, Korea, pp.1175-1180, 2018.