



Received: 2024/11/13  
Revised: 2024/11/21  
Accepted: 2024/12/07  
Published: 2024/12/31

**\*Corresponding Author:**

**Joo-Hun Kim**  
Technology & Research Institute, KTE Co., Ltd.  
95-12, Noksansaneop-daero, Gangseo-gu, Busan  
46752, Korea  
Fax: +82-51-600-3552  
E-mail: jhkim4@kte.co.kr

**Abstract**

본 논문에서는 RTOS 기반 PLC를 활용하여 기존의 PC와 PLC를 활용한 제어 시스템과 동일 기능을 가지는 제어 시스템의 하드웨어 설계 및 구현에 대해 다루었다. 또한 소형화를 비롯한 기존 제어 시스템의 개선사항들에 대한 적극 수용을 중점적으로 보여준다. 이를 통해 높은 신뢰성을 보이고 실시간 데이터 처리가 가능한 제어 시스템의 하드웨어를 설계 및 구현한다.

In this paper, we deal with the hardware design and implementation of a control system using RTOS-based PLC which has the same function as a control system using an existing PC and PLC. It also focuses on miniaturization of control system along with active acceptance of the improvement requirements of existing control system. Through this, the hardware of a control system that shows high reliability and can process real-time data is designed and implemented.

**Keywords**

하드웨어 설계(Hardware Design),  
소형화(Miniaturization),  
제어 시스템(Control System),  
프로그래머블 로직 컨트롤러  
(Programmable Logic Controller),  
실시간 운영체제(Real-time Operating System)

# RTOS 기반 PLC를 활용한 소형화된 제어 시스템의 하드웨어 설계 및 구현

## Hardware Design and Implementation of a Miniaturized Control System Using RTOS-based PLC

김주현<sup>1\*</sup>, 이수진<sup>2</sup>, 남수민<sup>3</sup>

<sup>1</sup>(주)KTE 기술연구소 책임연구원

<sup>2</sup>(주)KTE 기술연구소 선임연구원

<sup>3</sup>(주)KTE 기술연구소 연구원

Joo-Hun Kim<sup>1\*</sup>, Su-Jin Lee<sup>2</sup>, SuMin Nam<sup>3</sup>

<sup>1</sup>Senior manager, Technology & Research Institute, KTE Co.,Ltd.

<sup>2</sup>Manager, Technology & Research Institute, KTE Co.,Ltd.

<sup>3</sup>Assistant, Technology & Research Institute, KTE Co.,Ltd.

### 1. 서론

현대 산업의 자동화 분야에서는 공간의 제약이 있는 환경에서 기존의 시스템과 동일 기능을 유지하면서 에너지 효율성을 높일 수 있는 소형 제어 시스템의 필요성이 높아지고 있다.

제어 시스템의 소형화를 위해 고려할 것은 제어 시스템을 구성하는 주요 구성 자재와 각 자재의 역할이다. 이러한 구성 요소들의 분석, 통합이 소형화를 위해 중요한 과정이다. 일반적인 해양 선박의 제어 시스템은 PLC(programmable logic controller)와 PC 외에도 통신 허브, 유선 및 무선 통신 장치 등의 다양한 구성 요소로 이루어져 있다.

본 논문에서는 RTOS(real-time based operating system) 기반 PLC를 사용하여 위에서 설명한 바와 같이 제어 시스템의 소형화에 대한 방법을 연구한다. RTOS는 실시간으로 작업을 완료할 수 있도록 보장하는 운영체제로, embedded system에서 높은 신뢰성과 실시간 데이터 처리 능력을 제공한다. 본 논문은 RTOS 기반 PLC를 활용한 제어 시스템을 기존 PLC와 PC를 활용한 제어 시스템에 대비하여 동일 기능을 가지는지 여부를 확인하고자 하였다.

본 연구는 기존 제어 시스템 대비 소형화된 제어 시스템의 하드웨어를 설계하고 구현하는 점에 중점을 두었으며 기존 제어 시스

템과의 비교, 분석을 통해 제어 시스템의 소형화 유용성을 입증하는 것을 목표로 한다.

## 2. RTOS

### 2.1 RTOS의 정의

RTOS는 요구되는 작업이 일정한 시간 내에 완료되도록 보장해 주는 운영 체제로, 실시간 시스템의 필요성에서 시작되었다. 주로 embedded system에서 RTOS는 일반적인 OS와 비교하여 시간 관리에 차이가 있다. 시간 관리, 다시 말해 real-time은 사용 가능 자원이 한정된 상황에서 작업 수행이 요구되는 경우 일정한 시간 내에 작업을 모두 처리하고 결과를 내는 것을 말한다. 즉, 중요한 작업을 하는데 지연 없이 작업을 처리하고 완료 시간을 일정하게 유지하도록 보장해 주는 것이다.

### 2.2 RTOS의 핵심 기능

RTOS의 핵심 기능은 크게 8가지로 나눌 수 있다. 결정론적 타이밍(deterministic timing), 멀티태스킹(multitasking), 태스크 우선순위(task priority), 인터럽트 처리(interrupt handling), 자원 관리(resource management), 실시간 스케줄링(real-time scheduling), 타이머 및 클럭 관리(timers and clock management), 통신 기능(inter-task communication)이다.

결정론적 타이밍은 RTOS의 기본적인 기능으로 요구되는 작업이 일정한 시간 내에 완료되도록 보장하며 예측이 가능한 응답 시간을 제공한다는 특징이 있다. 멀티태스킹은 여러 작업을 동시에 수행하는 기능

이며 이를 통해 CPU가 가진 자원의 효율적인 분배가 가능하다. 태스크 우선순위는 작업을 처리하는 과정에서 중요 작업의 우선 처리가 보장되는 기능으로 우선순위를 설정함으로써 긴급 작업의 우선 처리가 가능하다. 인터럽트 처리는 발생하게 되는 외부 이벤트에 즉각 반응하는 기능으로 효율적이고 즉각적인 인터럽트 관리를 통해 요구되는 작업을 효율적으로 처리할 수 있게 한다.

자원 관리는 메모리, CPU와 같은 보유 자원의 효율적인 사용을 위한 기능으로, 이 기능을 통해 자원의 할당 및 해제 그리고 자원 충돌을 방지할 수 있다. 실시간 스케줄링은 정적·동적 스케줄링 알고리즘을 사용하여 태스크에서 요구하는 시간 조건을 충족시키는 기능이다. 타이머 및 클럭 관리는 주기적 작업 실행과 타임아웃 관리를 통해 이벤트의 정확한 시간 측정이나 이벤트의 예약과 같은 기능을 수행한다. 마지막으로 통신 기능은 태스크 간 데이터의 교환 및 동기화를 위한 기능으로, 메시지 큐, 이벤트 플래그, 세마포어 등을 사용한다.

### 2.3 RTOS의 구조

Fig. 1은 RTOS의 구조에서 task의 상태를 나타낸 다이어그램이다. Task는 embedded system에서 하나의 목적을 위해 응용 프로그램을 논리적으로 나누는 개념이다. 목적을 수행하는 데 여러 기능을 동시에 수행할 필요성에 의해 기능 블록의 모듈화 및 CPU의 효율적인 사용 등의 목적으로 multitasking이 발생하게 되었다.

Task status는 Fig. 1에 나타난 바와 같이 크게 dormant, ready, running, ISR, waiting으로 나눌 수 있다. Dormant의 경우는 수면 상태로, task가 메

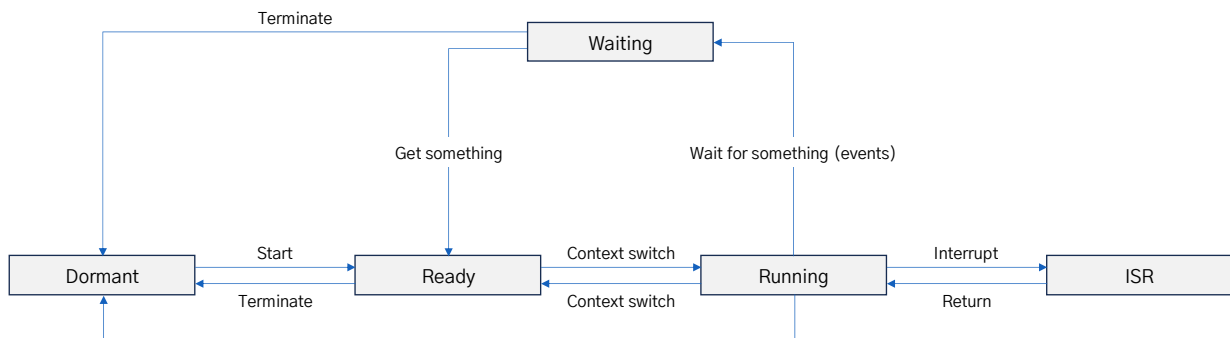


Fig. 1. Diagram of task status

모리에는 존재하지만 수행될 수 있는 상태가 아님을 나타낸다. Ready의 경우 task의 준비 상태로 수행은 가능하지만, 현재 수행되는 task보다 우선순위가 낮은 상태를 나타낸다. Running은 실행 상태로 CPU를 실제로 사용하여 태스크가 수행 중임을 나타낸다. Delay는 지연 상태로 일정 시간 동안 작업이 중지됨을 뜻한다. Waiting은 이벤트에 대한 task의 대기 상태로, task가 이벤트 발생을 기다리고 있는 상태이다. 마지막으로 ISR(interrupted)은 중단 상태로 CPU가 인터럽트 서비스를 수행하는 경우의 상태를 나타낸다.

### 3. 제어 시스템의 하드웨어 구성

#### 3.1 기존 PLC 활용 제어 시스템

PLC를 활용한 일반적인 제어 시스템의 하드웨어 구성은 Fig. 2에 나타난 바와 같이 크게 PLC 본체(main unit), PC, 전원 공급 장치(power supply), 입력 장치(input devices), 출력 장치(output devices) 등의 장치로 이루어져 있다.

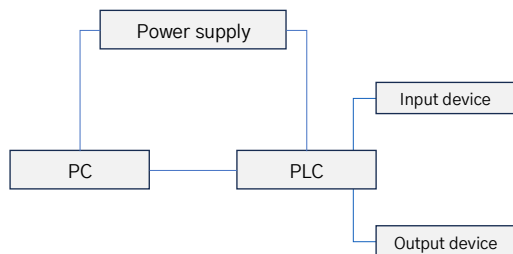


Fig. 2. PLC+PC based control system

PLC 본체는 기본적으로 CPU(중앙처리장치)를 포함한다. CPU는 PLC의 두뇌 역할로, 모든 제어 작업과 논리 연산을 수행하는 프로그램 메모리와 작업 메모리가 포함되어 있다. PC는 PLC와 통신이 연결되어 PLC의 데이터를 로깅하는 역할을 한다. Power supply는 PLC 및 PC와 같이 제어 시스템의 장치들에 안정적인 전원을 공급한다. 입력 장치와 출력 장치는 각각 디지털, 아날로그로 나뉜다.

이와 같은 기존 제어 시스템의 구성을 소형화된 제어 시스템에 그대로 적용하기에는 여러 한계점이 발생한다. 기존 PLC의 경우 PLC 본체 외에도 데이터 로깅을 위해 통신선에 의한 PC와의 연결이 필요하

다. 이처럼 구성을 위해서는 여러 추가 장치가 공간을 많이 차지하는 단점이 있으며, 이는 제어 시스템의 소형화가 어렵다는 뜻이다.

제어 시스템 소형화를 위해서는 기존의 구성보다 축소된 구성으로 동일 기능을 사용할 수 있는 시스템의 구축이 필요하다. 이를 위해 사용한 것이 RTOS 기반 PLC이다.

#### 3.2 RTOS 기반 PLC 활용 제어 시스템

Fig. 3는 RTOS 기반 PLC를 활용한 제어 시스템의 하드웨어 구성을 나타낸 것으로, PLC 본체, 전원 공급 장치, 입력 장치, 출력 장치 등의 장치로 이루어져 있다.

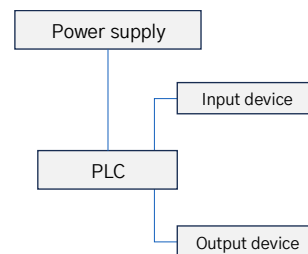


Fig. 3. RTOS-based control system

RTOS 기반 PLC이기 때문에, 데이터 로깅이 PLC 만으로도 가능하여 PC가 구성에서 빠질 수 있었으며, 이를 통해 기존 제어 시스템 구성에 포함되어 있던 PC와 PLC의 통신 연결을 위한 통신선이 제외될 수 있었다. 이를 통해 제어 시스템 자체의 물리적인 크기뿐만 아니라 전력 소모량도 줄일 수 있다. 더불어 기존 제어 시스템에서 각 기능을 위해 추가된 장비들이 RTOS 기반 PLC로 대체되는 것이 유지 보수에 있어 유리한 것을 알 수 있다. 이는 곧 PLC가 제어에 집중할 수 있음을 의미하며, 또한 시스템 부하를 줄일 수 있으므로 제어 시스템의 관리가 용이함을 의미한다.

### 4. 제어 시스템의 하드웨어 구현

#### 4.1 RTOS 기반 PLC의 선정

RTOS 기반 PLC는 Mitsubishi MelsecQ Series, Beckhoff Embedded PC Series, Bachmann PLC 등

이 있으며, 위의 제품들 중 본 논문에서는 Bachmann PLC의 중급 Level 제품인 MX220을 선택하여 제어 시스템의 하드웨어를 구현하였다.

#### 4.2 제어 시스템의 하드웨어 구성

Fig. 4는 RTOS 기반 PLC를 활용한 제어 시스템의 하드웨어 구성 모습이다.



Fig. 4. Control system with MX220 PLC

### 5. 제어 시스템 검증

PC와 PLC를 활용한 기존의 제어 시스템과 비교하여, 본 논문을 통해 구성·구현한 RTOS 기반 PLC를 활용한 소형화된 제어 시스템이 기존의 제어 시스템과 동일 기능을 가지는지, 소형화의 유용성이 있는지 검증하기 위해 각 구성의 크기 및 중량과 소비전력, PLC의 전반적인 로직 구현, 데이터 로깅 총 3가지를 통해 확인하였다. Fig. 5는 기존의 PLC와 PC를 활용한 제어 시스템의 하드웨어 구성 모습이다.



Fig. 5. Control system with S7-1500 and C6920

#### 5.1 크기와 중량, 소비전력

제어 시스템 구성의 크기와 소비전력 비교는 공통 구성 요소인 power supply와 입/출력 장치를 제외하고 기존의 제어 시스템에 포함되는 PLC와 PC 그리고 소형화된 제어 시스템에 포함되는 PLC를 비교하는 방식으로 수행하였다.

Table 1은 기존의 PLC와 PC를 활용한 제어 시스템의 각 구성요소별 크기와 무게 그리고 소비전력에 대해 정리한 표이다.

Table 1. Size, weight and power consumption of control system using PLC+PC

PLC	Size (mm)	Weight (g)	Current consumption (W)
PC (C6920)	121×235×65	1,900	33.6
PLC (S7-1500)	210×147×129	2,094	36.0

Table 2는 RTOS 기반 PLC를 활용한 제어 시스템의 각 구성요소별 크기와 중량 그리고 소비전력에 대해 정리한 표이다. Table 1과 Table 2의 비교를 통해 RTOS 기반 PLC를 활용한 제어 시스템이 PLC와 PC를 활용한 기존 제어 시스템보다도 공간, 무게, 소비전력 측면에서 유리한 것을 확인할 수 있다.

Table 2. Size, weight and power consumption of control system using RTOS-based PLC

PLC	Size (mm)	Weight (g)	Current consumption (W)
PLC (MX220)	114×121×57	700	7.4

#### 5.2 로직 구현

PLC의 전반적인 로직 구현을 확인하기 위해 기존 제어 시스템에서 PC를 통해 PLC에 프로그래밍한 로직에 대해 RTOS 기반 PLC를 활용한 제어 시스템에서 동일한 구현이 가능한지 확인하였다. Fig. 6는 기존 PLC와 PC를 활용한 제어 시스템에서 구현한 로직의 모습이다.

□ Network 128: Title:

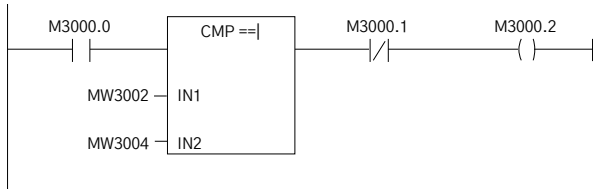


Fig. 6. Logic in PLC+PC based control system

Fig. 7은 RTOS 기반 PLC를 활용한 제어 시스템에서 기존의 제어 시스템에서의 로직을 동일하게 구현한 모습이다.

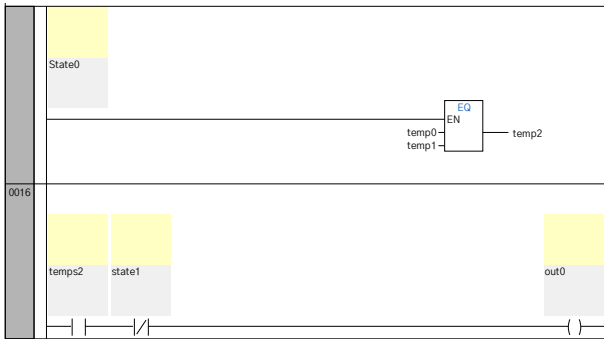


Fig. 7. Logic in RTOS-based PLC control system

Fig. 6과 Fig. 7의 비교를 통해 기존 제어 시스템에서 구현이 가능한 로직을 RTOS 기반 PLC를 활용한 제어 시스템에서도 동일하게 구현할 수 있음을 알 수 있다.

### 5.3 데이터 로깅

데이터 로깅의 경우 PLC와 PC를 활용한 기존 제어 시스템은 PC를 통해 데이터 로깅을 하고, RTOS 기반 PLC를 활용한 제어 시스템의 경우 PLC를 통해 데이터 로깅을 하여 확인했다.

Fig. 8은 PC와 PLC를 활용한 기존의 제어 시스템 PLC의 데이터 로깅에 대한 자료이다. PLC 측에서 데이터를 CSV 파일 형식으로 저장한 후, 연결된 PC가 해당 파일에 액세스하여 해당 데이터를 수집한다.

Fig. 9은 RTOS 기반 PLC를 활용한 제어 시스템의 데이터 로깅에 대한 자료이다. PLC 측에서 데이터를 CSV 파일 형식으로 저장하며, 파일 액세스 또한 PLC 측에서 가능하다.

A1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	TIME																
2	6:00:00	OFF	Normal	0.002	0	0.001	0.001	0.001	0.001	0.001	0.001	0	0	0.002	0.002	0.002	0.002
3	6:00:01	OFF	Normal	0	0.002	0	0.002	0.001	0.001	0.001	0.001	0	0	0.002	0.003	0.002	0.002
4	6:00:02	OFF	Normal	0.001	0.001	0.002	0.001	0.001	0.001	0.001	0.002	0	0	0.002	0.002	0.002	0.002
5	6:00:03	OFF	Normal	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.002	0	0	0.001	0.002	0.002	0.004
6	6:00:04	OFF	Normal	0	0.001	0.001	0.001	0	0.002	0.001	0.002	0	0	0.002	0.004	0.002	0.002
7	6:00:05	OFF	Normal	0.001	0.001	0.001	0.002	0.001	0.001	0.001	0.002	0	0	0.001	0.002	0.002	0.003
8	6:00:06	OFF	Normal	0.002	0	0.001	0.002	0.001	0.001	0.001	0.002	0	0	0.001	0.002	0.003	0.003
9	6:00:07	OFF	Normal	0.001	0.001	0	0.002	0.001	0.002	0.001	0.002	0	0	0.002	0.002	0.002	0.002
10	6:00:08	OFF	Normal	0.001	0.001	0.002	0.001	0.002	0.001	0.001	0.001	0	0	0.002	0.003	0.002	0.002
11	6:00:09	OFF	Normal	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.001	0	0	0.002	0.002	0.002
12	6:00:10	OFF	Normal	0	0.001	0.001	0.002	0	0.001	0.002	0.002	0	0	0.002	0.002	0.001	0.002
13	6:00:11	OFF	Normal	0	0.002	0.001	0.001	0.002	0.002	0.001	0.001	0	0	0.001	0.002	0.003	0.003
14	6:00:12	OFF	Normal	0	0.001	0.001	0	0.001	0.001	0.001	0.001	0	0	0.002	0.003	0.002	0.002
15	6:00:13	OFF	Normal	0.001	0.001	0	0.002	0	0.001	0.001	0.001	0	0	0.002	0.002	0.002	0.002
16	6:00:14	OFF	Normal	0	0.001	0.001	0.002	0	0.001	0.001	0.001	0	0	0.001	0.002	0.003	0.004
17	6:00:15	OFF	Normal	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0.001	0	0	0.002	0.003	0.002	0.002
18	6:00:16	OFF	Normal	0.002	0.002	0.001	0.001	0.002	0.001	0.001	0.001	0	0	0	0.002	0.002	0.003
19	6:00:17	OFF	Normal	0	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0	0	0.002	0.003	0.003	0.003
20	6:00:18	OFF	Normal	0.001	0	0.002	0.001	0.001	0.001	0.001	0.001	0	0	0.002	0.002	0.001	0.001
21	6:00:19	OFF	Normal	0.001	0.002	0	0.002	0.001	0.002	0	0.001	0	0	0.001	0.002	0.002	0.003
22	6:00:20	OFF	Normal	0	0.001	0.001	0.001	0.001	0.001	0.001	0	0	0	0.002	0.002	0.002	0.002
23	6:00:21	OFF	Normal	0.001	0.001	0.001	0.001	0	0.001	0.001	0.002	0	0	0.002	0.002	0.002	0.002
24	6:00:22	OFF	Normal	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.002	0	0	0	0.002	0.002	0.002
25	6:00:23	OFF	Normal	0	0.001	0	0.001	0.001	0.002	0.001	0.002	0	0	0	0.002	0.002	0.002
26	6:00:24	OFF	Normal	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.002	0	0	0.001	0.002	0.003	0.003
27	6:00:25	OFF	Normal	0.001	0.001	0.001	0	0.001	0.001	0.002	0.001	0	0	0.001	0.002	0.002	0.003

Fig. 8. Data logging in PLC+PC based control system

A1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	TIME																
2	6:00:00	-0.002	0.001	-0.001	0.001	0	0	0	0	0	0	0	0	0.001	0	0	-0.002
3	6:00:01	-0.001	-0.001	-0.002	0.002	-0.002	0	0	0.002	0	0	0	-0.001	0	-0.001	-0.001	0
4	6:00:02	-0.002	0.001	-0.001	0.001	-0.001	0	0	0	0	0	0	0	0.001	0	0	-0.002
5	6:00:03	-0.001	0.001	-0.001	0	0	0	-0.001	0.001	0	0.001	0	0	0	0	0	-0.001
6	6:00:04	-0.001	0.001	-0.001	0	-0.001	0	0	0.001	0	0	-0.001	0	0	0	0	0.002
7	6:00:05	-0.002	0.002	0	0.002	-0.001	0.001	0	-0.001	-0.002	0	0	0	0.001	-0.001	0	-0.001
8	6:00:06	-0.002	0	-0.001	0	-0.001	0	0	0.001	0	0	0	0	-0.001	-0.002	0	0
9	6:00:07	0	0	-0.001	0	-0.001	0	-0.001	0	-0.001	0	0	0	0	-0.001	0.001	-0.001
10	6:00:08	-0.002	0.001	0	0.001	0	0	0	0	0	0	-0.001	0	0	0	0	-0.001
11	6:00:09	0	-0.001	-0.002	0.001	-0.001	0	0.001	0.001	0	0	-0.001	0	-0.002	0.001	0	0.001
12	6:00:10	-0.002	0.001	-0.001	0.002	0	0	-0.001	0	0	0	0	0	0.001	0	0	0.001
13	6:00:11	-0.001	0.001	0	0.001	0	0	0.001	0.001	0.002	0.001	-0.001	0	0	0.002	0.002	0.001
14	6:00:12	-0.001	0	-0.002	0	-0.001	0.001	0	0.002	0.001	0	0	0	0	-0.001	0	0.001
15	6:00:13	-0.001	0.001	0	0.001	0	0	0	-0.001	-0.001	0	0	0	0.001	0	0	-0.001
16	6:00:14	-0.002	0	-0.001	0.001	-0.001	0.001	0	0.001	-0.001	0	0	0	-0.001	0	0	-0.001
17	6:00:15	-0.001	0	-0.001	0.001	0	0	0	0	0	0	-0.001	0.001	0	0	0	-0.001
18	6:00:16	-0.001	0.001	-0.001	0.002	0	0	0	0	-0.001	0	0	0	0.001	0	0.001	-0.001
19	6:00:17	-0.001	0	-0.001	0.001	-0.001	0	0	0.001	0	0.001	0	0.001	0	-0.001	-0.002	0
20	6:00:18	0	0.001	-0.001	0	-0.001	0	-0.001	0.001	-0.001	0	0	0.001	0	0.001	0	-0.001
21	6:00:19	0	0.001	0	0.001	-0.001	0	0	0	0	0	0	0	0	0	0	-0.001
22	6:00:20	-0.002	0	-0.002	0	-0.001	0.001	0.001	0.001	0	0.001	0	0	0	-0.001	0	0
23	6:00:21	-0.001	0.001	0	0.001	0	0	-0.002	0.001	-0.001	-0.001	0	0.001	0	0	0	-0.002
24	6:00:22	-0.001	0.001	-0.001	0.001	-0.001	0.001	0	0.001	-0.001	0.001	0	0	0	-0.001	0	-0.001
25	6:00:23	-0.002	0	-0.001	0	-0.001	0	0	0	0	0	0	0	0	-0.001	0	-0.001
26	6:00:24	-0.001	0.002	-0.001	0.001	-0.001	0	-0.002	0	-0.001	0	0.001	0	-0.001	0.001	-0.001	0
27	6:00:25	-0.001	0	-0.002	0.001	-0.001	0	0	0.001	-0.001	0	-0.001	0	-0.001	0	0	0.001

Fig. 9. Data logging in RTOS-based PLC control system

Fig. 8과 Fig. 9의 비교를 통해 기존 PC와 PLC를 활용한 제어 시스템에서의 데이터 로깅 기능을 RTOS 기반 PLC를 활용한 제어 시스템에서 동일하게 사용할 수 있는 것을 알 수 있다. 또한 Fig. 5와 Fig. 6에서 보듯이, 기존 제어 시스템에서는 데이터 로깅을 위해 PLC에 PC를 연결해야 했던 점에 비해 RTOS 기반 PLC를 활용한 제어 시스템의 경우는 PC 없이 PLC만으로 데이터 로깅이 가능하여 동일 기능을 수행하는 것을 확인하였다.

## 6. 결론

본 논문에서는 RTOS 기반 PLC를 활용한 제어 시스템의 하드웨어 구성 및 구현을 통해 기존의 PLC와 PC를 활용한 제어 시스템을 소형화하는 방법을 제안하였다.

RTOS 기반 PLC를 활용하여 제어 시스템의 하드웨어를 구성할 경우, 물리적인 크기 측면에서 기존 제어 시스템의 소형화가 가능하며, 사용하는 RTOS 기반 PLC에 따라 기존 PLC와 PC를 활용한 제어 시스템과 동일 기능을 가진 제어 시스템의 구현이 가능



한 것을 확인할 수 있었다. 또한 동일 기능의 구현이 가능하므로 구성 장비의 축소 또한 가능하고, 그에 따라 유지 보수 측면에서 기존의 PC와 PLC를 활용한 제어 시스템보다 RTOS 기반 PLC를 활용한 제어 시스템이 유리한 것을 알 수 있었다. 그리고 데이터 로그 기능의 검증 과정을 통해 RTOS 기반 PLC를 활용하여 제어 시스템을 구성하면 통신 네트워크를 축소할 수 있는 것을 확인했다.

본 논문에서는 하드웨어의 구성 및 구현을 통한 기존 제어 시스템과의 성능과 기능 비교를 기준으로 연구를 진행했지만, 향후 소프트웨어 구성 및 구현을 통해 기존 제어 시스템의 소프트웨어 개선사항 연구를 진행할 계획이다. 본 논문에서의 연구와 향후 추가 연구를 통해 효율적이고 효과적인 제어 시스템을 구성, 구현할 수 있을 것으로 예상된다.

## 참고문헌

- [1] Je Wung Kim, Sung-Soo Lim, "System Model-driven Conversion from PLC-based Systems to RTOS-based Systems," *Journal of the Korea Society of Computer and Information*, Vol. 14, No. 3, 2009, pp. 13-26.
- [2] Ingyu Choi, Changki Jeong, Byungju Ahn, "The Integration of Hardware Using PLC for Turbine Control system," *Journal of the Korea Institute of Electrical Engineers*, Vol. 2002, No. 7, 2002, pp. 2203-2205.
- [3] Sang-Hak Oh, Joong-Soo Chung, "Design of The Embedded Network Software Using RTOS," *Journal of the Korea Information Processing Society*, Vol. 39, No. 2, 2012, pp. 231-233.
- [4] Chang-Gyu Bak, "Design of Lightweight RTOS for MCU," *Journal of the Korea Institute of Maritime Information & Communication Sciences*, Vol. 15, No. 6, 2011, pp. 1301-1306.