

Control & Communication 상에서의 PLC 고장허용 통신 시스템 구현

임 완택*, 김 응석**, 양 해원*

*한양대학교 전자컴퓨터공학부, **한라대학교 전기전자제어공학부.

Implementation of PLC Fault Tolerance Communication System in Control & Communication Link

Wan - Taek Lim*, Eung - Seok Kim**, Hai - Won Yang*

*Dept.of Elec.Eng.Hanyang Univ. **Dept.of Elec.Eng.Halla Univ.

Abstract - In this paper, we introduced a fault tolerant control system with the aim of achieving higher degree of reliability for a PLC control system in the field network. The system reliability was evaluated by MTBF(Mean Time Between Failure). The design of the fault tolerant system through CC-Link of Mitsubishi's MELSEC network was presented. In addition, the PLC data is transmitted from the field network's PC to the host PC by TCP/IP Window socket.

스인 CC-Link는 하나의 Master PLC로 Local국, Remote I/O국, Remote Device국, Intelligent Device국을 제어할 수 있다. RS-485 통신을 하며, 버스형을 채용한다.

1. 서 론

최근 산업현장에서는 윈도우 기반 운영체제의 발전과 네트워크가 차지하는 비중이 증가하고 있다. 또한 생산현장에 설치된 자동화 관련 기기들 간의 실시간 데이터 통신을 실현하기 위해 PLC를 대상으로 한 PC 기반 분산제어시스템이 활용되고 있다[1]. PLC를 이용하면 작업환경을 개선하고 작업 능력과 생산성을 증대시킬 수 있다. 그러나 PC를 이용한 PLC 제어에는 수많은 고장 요소가 존재하며 PLC의 고장사고는 중대해해를 유발하게 된다. 이러한 중대해해를 예방하기 위하여 고장허용 제어시스템을 적용하면 고장이 발생하여도 연속적인 운전이 가능해지며, 전체 제어시스템 운용에 있어서 신뢰도를 향상시킬 수 있다[2][4].

본 논문에서는 이러한 고장허용 시스템을 구축하고자 한다. Remote I/O, Remote Device, 지능형 센서 및 PLC는 Master PLC와 CC-Link(Control & Communication Link) 프로토콜을 이용하여 연결한다. TCP/IP Socket 프로그램을 이용하여 MELSEC Net의 최하위 계층인 CC-Link의 Master PLC를 PC가 Client가 되어 제어한다. 최상위 계층인 Host의 PC가 Server가 되어 LAN을 통하여 Client에서 동작하는 PLC의 고장결과 및 데이터를 실시간으로 전송받는 알고리즘을 구성하여 전체 MELSEC Net의 분산제어시스템을 구현하였다.

2. MELSEC NETWORK 모델

2.1 MELSEC Net 구성

미쯔비시사의 FA(Factory Automation) MELSEC Net 시스템은 그림 1과 같다. MELSEC Net 시스템은 기존의 복수 네트워크를 고기능화, 고속화, 대용량화한 시스템이다. 통신 속도는 10MBPS로 고속 데이터 송신이 가능하며, 링크 전용 프로세서로 고속 링크 스캔이 가능하다. 그림 1에서 MELSEC Net은 Q시리즈 PLC를 Ethernet(10BASE5, 10BASE2)의 TCP/IP, UDP/IP 통신 프로토콜을 이용하여 PC나 워크스테이션과 접속한다. 전송로 형식은 루프형과 버스형을 채용한다. 구간거리는 광케이블 1km, 동축케이블 500m이고, 자국을 최대 64국까지 접속할 수 있다[3]. 필드 버

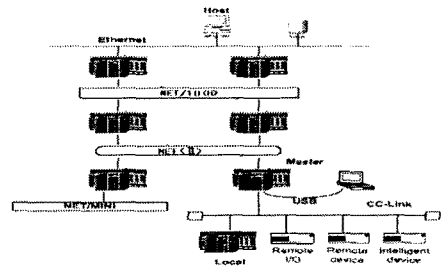


그림 1. 미쯔비시 FA 네트워크 구조

MELSEC Net은 그림 2에서 보는 바와 같이 리프레시 파라미터 데이터를 1모듈 당 64분할로 세분화하여 사용한다. 따라서 Q CPU는 시퀀스 프로그램에서 사용하지 않는 디바이스의 리프레시 처리를 삭제하고, 필요한 곳에서만 리프레시를 하여 리프레시 시간을 단축한다.

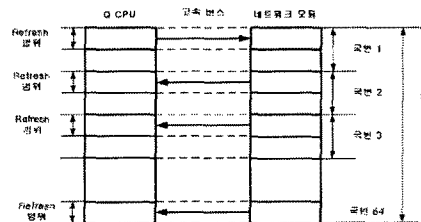


그림 2. Refresh 데이터 세분화 전송

2.2 DATA 전송

MELSEC Net의 Field Network Layer인 CC-Link는 분산 배치한 입출력 모듈, 인텔리전트 기능 모듈, 특수기능 모듈 등을 RS-485 전용 케이블로 연결하고, Master PLC CPU에서 위의 모듈을 제어하는 시스템이다. CC-Link는 컨베이어 라인이나 기계장치 등의 설비기기에 각 모듈을 분산하여 설치함으로써 시스템 전체의 배선을 절감할 수 있다. 또한 각 모듈이 취급하는 입출력 등의 ON/OFF 정보 및 수치 데이터를 고속으로 송수신을 할 수 있다. 이러한 이유로 PLC CPU를 복수로 접속함으로써 간단히 분산 시스템을 구축할 수 있다. CC-Link는 하나의 Master PLC에 전체 64개의 리모트 I/O, 리모트 디바이스, 로컬국, 인텔리전트 디바이스국을 접속할 수 있다. CC-Link는 Broadcast

Polling 프로토콜로 데이터를 송수신하게 된다.

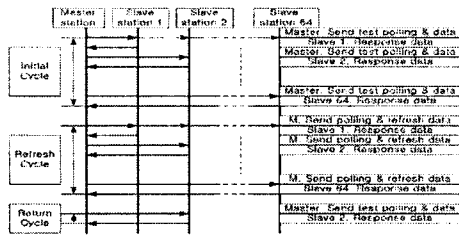


그림 3. Broadcast Polling Method

그림 3에서 보면, 시스템이 1회 시동하는 동안 Master국은 Slave국 1로 test polling data를 전송한다. Slave국 1은 그 응답으로 test-loop back data를 보낸다. 이 과정이 최대 64번의 Slave국까지 차례로 반복되고 이러한 과정을 Initial Cycle로 부른다. 그 다음 처음부터 다시 Master국은 Slave국 1로 실제 refresh data를 전송한다. Slave국 1은 그 응답으로 response data를 보낸다. 이러한 과정이 최대로 개설되어 있는 Slave국 수 만큼 거치며 이러한 과정을 Refresh Cycle로 부른다. 필드버스에서 데이터를 송수신하기 위해서는 위 두 과정만으로도 충분하다. 그러나 만일 어떠한 원인 불명의 오류가 존재할 시에는 시스템 다운이 발생되고, 이로 인해 원활한 데이터 전송을 하지 못하여 시스템의 신뢰도는 떨어지게 된다. 따라서 이러한 경우를 대비해서 만일 Slave국 2에서 고장발생시 Master국은 처음부터 다시 Initial Cycle을 수행하는 것이 아니라 고장이 발생한 Slave국 2에만 test polling data를 전송한다. 이때 Slave국 2로부터 응답이 있으면 시스템이 복원이 되고, 그렇지 않으면 Master국은 Slave국 2를 고장으로 간주한다. 결과적으로 Master PLC의 CPU는 송수신할 때 사용되는 버퍼메모리를 Slave국 2만 사용하지 않도록 메모리의 데이터를 삭제한다. 이러한 과정을 Return Cycle로 부른다.

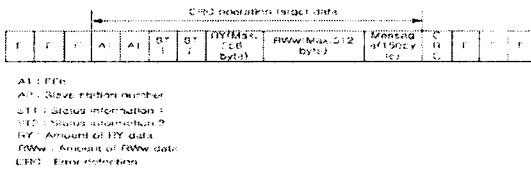


그림 4. Frame Format

그림 4는 Master 국과 리모트국간의 데이터 전송 프레임 형태의 예를 보인 것이다. RY는 리모트 국의 비트 데이터 출력을 말하고, RW는 워드데이터 출력을 말한다.

3. 고장허용 제어시스템 구현

3.1 PLC 제어시스템의 고장율

PLC 제어시스템의 신뢰도, 가용성(Availability), 안정성 항목은 언제나 명백하지 않다. 특히 신뢰도는 모든 측정에서 고장이 발생하지 않았다 하더라도 100(%) 신뢰도를 갖는다고는 볼 수 없다. 일반적으로 고장율(λ)이 신뢰도의 측도가 된다[4].

$$\text{고장율}(\lambda) = \frac{\text{고장개수}(n)}{\text{구성요소}(N_0) \times \text{사용시간}(t)} \quad (1)$$

3.2 신뢰도 함수

신뢰성 척도는 구성요소(N_0)와 사용시간(t)에서의

고장개수 그리고 사용시간(t)과 ($t+\Delta t$)시간의 고장개수(n)를 알면 위 식(1)에 의거하여 계산이 가능해진다. 일반적으로 신뢰도 함수는 다음과 같이 정의한다.

$$\text{신뢰도}(R(t)) = e^{-H(t)} \quad (2)$$

여기에서, $H(t) = \int_0^t \lambda(t)dt$ 이다. 따라서 고장율(λ)을 알면 신뢰도 함수를 구할 수 있다. 그러나 고장이 시간적으로 일정한 우발고장기간에서는 고장율(λ)이 시간에 따라 변하지 않는 상수로 볼 수 있기 때문에 신뢰도 함수는 다음과 같이 된다.

$$R(t) = e^{-\lambda t} \quad (3)$$

이와 같은 지수분포를 특징지어 주는 것은 λ , 즉 평균고장률이 되며 이 λ 의 역수 $\frac{1}{\lambda}$ 은 시간의 단위를 가지므로 고장발생시까지의 평균작동시간, 즉 평균수명(mean life)을 나타낸다. 이것을 MTTF(mean time to failure)라 부르며 고장시 수리를 하여 사용하는 기기의 경우에는 MTBF(mean time between failure)라고 한다.

3.3 PLC 제어시스템의 고장특성

PLC 제어시스템의 고장의 분포도는 그림 5와 같이 경험적으로 볼 때 PLC를 설치함에 있어 다음과 같이 개략적으로 분산시킬 수 있다.

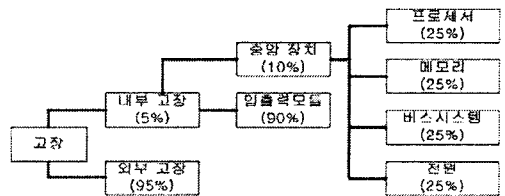


그림 5 PLC 설치에서의 고장발생 분포도

시스템의 고장은 PLC에서 5(%) 정도 발생되고, 센서나 단자대 혹은 케이블 등과 같은 외부에서 나머지 95(%)정도 고장이 발생한다. 그러나 PLC의 고장율은 적지만 실제 고장발생시 엄청난 손실을 가져올 수 있다.

3.4 PLC 고장허용 제어시스템

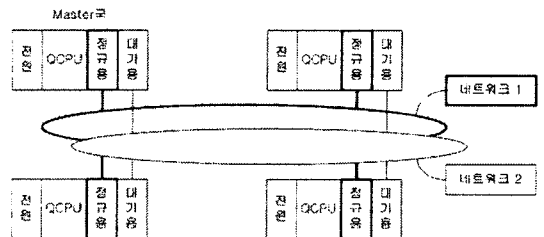


그림 6. 간이 이중화 네트워크 시스템

처음에는 Master PLC의 정규용 모듈(QJ71LP21)이 기능을 수행하다가 이것이 고장나면 대기용 모듈(QJ71BR11)이 그 기능을 받아서 계속 기능을 수행하도록 결합시켜서 간이 이중화 시스템을 구성할 수 있다. 이것을 이용하여 정규 네트워크가 다운되어도 대기 네

트위크와의 링크 데이터 리프레시로 전환하여 데이터 링크를 계속할 수 있다.

만일 Master PLC 정규용 모듈의 신뢰도 R_1 은 평균 고장율 λ_1 인 지수분포에 따르고, 대기용 모듈의 신뢰도 R_2 는 평균고장율 λ_2 에 따른다고 가정한다. 이때 시스템의 임무시간(mission time)이 T 라면, 이 시스템의 신뢰도는 다음과 같이 된다.

만일, $\lambda_1 \neq \lambda_2$ 인 경우에는

$$R_s = e^{-\lambda_1 T} + \int_0^T \lambda_1 e^{-\lambda_1 t} \cdot e^{-\lambda_2(T-t)} dt$$

$$= \frac{1}{\lambda_1 - \lambda_2} [\lambda_1 e^{-\lambda_2 T} - \lambda_2 e^{-\lambda_1 T}] \quad (4)$$

만일, $\lambda_1 = \lambda_2 = \lambda$ 인 경우에는

$$R_s = e^{-\lambda T} + \int_0^T \lambda e^{-\lambda t} \cdot e^{-\lambda(T-t)} dt$$

$$= e^{-\lambda T} (1 + \lambda T) \quad (5)$$

그리고 MTBF는 다음과 같이 된다.

$$MTBF_s = \int_0^\infty e^{-\lambda T} (1 + \lambda T) dT = \frac{2}{\lambda} \quad (6)$$

즉, MTBF는 모듈이 한 개인 경우의 2배가 된다. 신뢰도는 위 식(5)에서 알 수 있는 바와 같이 모듈이 한 개인 경우보다 $(1 + \lambda T)$ 배가 증가한다. 따라서 PLC 고장허용에 대한 이중화 시스템을 적용하면 전체 시스템의 신뢰도가 증가하게 된다. 신뢰도 증가는 시스템 고장발생시 시스템의 정지없이 연속운전이 가능함을 말한다.

4. TCP/IP WinSocket 시스템 구성

네트워크 프로그램 작성에는 여러 가지 레벨이 있을 수 있다. 그 가운데 TCP와 같은 트랜스포트 계층의 인터페이스를 이용하는 소켓 프로그램을 이용한다. TCP/IP의 소프트웨어적 기능을 담당하는 소켓은 소프트웨어로 작성된 통신 접속점이라고 할 수 있겠다. 소켓에 있어서 중요한 3가지의 요소를 생각할 수가 있는데, 프로토콜, 상호간의 IP주소, 상호간의 포트번호가 그것이다.

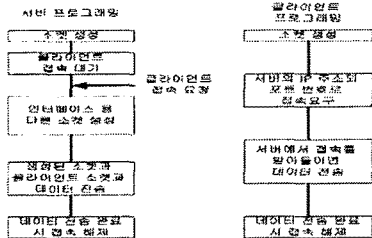


그림 7. 소켓 프로그램의 동작

그림 7은 서버와 클라이언트가 되는 두 PC사이를 TCP/IP 소켓으로 연결하고 데이터를 주고 받는 동작을 나타낸다. 소켓은 TCP 프로토콜을 사용하기 때문에 여러 도착하는 패킷의 순서에 신경 쓸 필요가 없다. 데이터를 받는 Host PC는 단순히 연속된 데이터로 인식하고 처리하면 된다. 이런 방식의 소켓은 많은 양의 데이터를 전송하거나 데이터에 대한 신뢰성이 요구될 때

사용한다. 서버는 소켓을 만들고 소켓에 이름을 주고 클라이언트가 소켓에 연결 요청이 들어오기를 기다린다. 서버가 요청을 감지하면 실제로 데이터를 전송 받는다.

```

#pragma once
#include <string>
using namespace std;

#define IPADDRESS "192.168.0.100" //클라이언트 서버의 IP 주소
#define PORT 2000 //클라이언트 서버의 포트 번호

class CData : public CObject //클라이언트 서버의 데이터
{
public:
    DECLARE_SERIAL(CData) //클라이언트 서버의 데이터
public:
    CData() {} //클라이언트 서버의 데이터 생성자
    CData(CData& data) {} //클라이언트 서버의 데이터 복사 생성자
public:
    BOOL m_bClose; //클라이언트 서버의 데이터 닫기 여부
    CString m_strPass; //클라이언트 서버의 데이터 비밀번호
public:
    void Init(); //클라이언트 서버의 데이터 초기화
public:
    virtual ~CData();
    virtual void Serialize(CArchive& ar) {} //클라이언트 서버의 데이터 직렬화
};

#pragma once
#pragma warning(disable:4786)
virtual void AssertValid() const;
virtual void Dump(CDumpContext& dc) const;
};
    
```

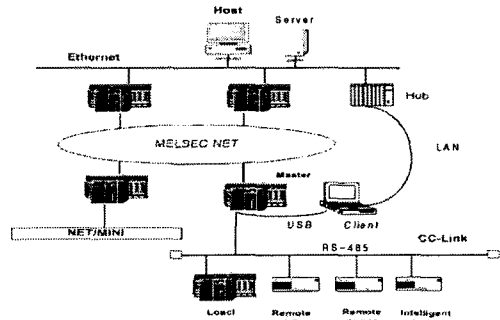


그림 8. 시스템 구성도

그림 8은 소켓 코드 일부와 소켓프로그램을 이용하여 시스템을 재구성한 것을 보이고 있다.

5. 결 론

본 논문에서는 고장허용 제어시스템을 위한 간이 이중화 시스템을 구성하였다. 따라서 고장을 허용하면서 데이터를 전송할 수 있도록 하였다. MTBF를 이용하여 본 논문에서 구현한 간이 이중화 시스템의 신뢰도가 향상되는 것을 입증하였다. 또한 TCP/IP 소켓프로그램을 구현하여 MELSEC Net의 최하위 필드 네트워크층과 최상위 Host간의 제어 및 감시를 가능하게 하였다. 본 논문에서 적용한 복수 네트워크의 구조가 간단하고, 배선이 절감되는 장점이 있다. 그러나 서버/클라이언트로 전송하기 위해서 네트워크 구조가 다소 복잡해지는 단점도 가지고 있다. 향후에는 이러한 문제를 해결할 수 있는 방안으로 TCP/IP 프로토콜과 무선통신을 연결시켜줄 수 있는 임베디드 리눅스 서버 구축에 관한 연구를 진행하고자 한다.

(참 고 문 헌)

- [1] Erich Sisa, "Communication Systems To Monitor And Control The Production Environment". Textile, Fiber, and Film Industry Technical Conference, IEEE Annual, pp. 8-7, 1997.
- [2] 손원기, 권오규, "이상허용제어에 관한 연구 동향" 대한전기학회지, 제48권, 4호, 1999.
- [3] Mitsubishi Programmable Logic Controller, QnA/Q4AR MELSECNET/10 Network System, manual.
- [4] 배도선, 전영록, 신뢰성 분석 대우학술총서 449 논저, 아르케, 1997. 7. 20.