

인텔 RUPI44를 이용한 data수집용 고속 통신모듈의 개발에 관한 연구

유 영 호* · 하 주 식**

A Study on the Development of High Speed Communication Module for Data Aquisition System Using INTEL RUPI44

Yu Yung - Ho, Ha Ju - Sik

Abstract

Recent automation system is bigger and bigger in the size by adopting the Distributed Computer Control System. In this paper high speed communication module over 375 KBPS is developed by using Intel RUPI44 and DPRam to bulid the Distributed Computer Control System(DCCS) with Personal Computer not a Engineering Workstation. Using developed communication module, the DCCS can be built by only inserting developed module to the PC and the Local controller composed by Intel MCS96 system. The performance of developed module was proved through the experiments.

1. 서 론

최근의 자동화는 그 규모적인 면에 있어서 어느 한 기계를 자동으로 감시제어하는 소규모 자동화에서 점차 공정자동화, 공장자동화 내지는 단지자동화로 그 규모가 거대화 되어가고 있고, 질적인 면에서는 국소부분의 최적제어에서 제품의 주문에서부터 생산,운반에 이르기 까지를 고려한 종합적인 최적제어로 제어대상의 안목이 거시적으로 변하고 있으며, 하드웨어적인 측면에서는 개인용컴퓨터와 마이크로프로세서

* 한국해양대학교 이공대학 제어계측공학과 교수

** 한국해양대학교 해사대학 기관공학과 교수

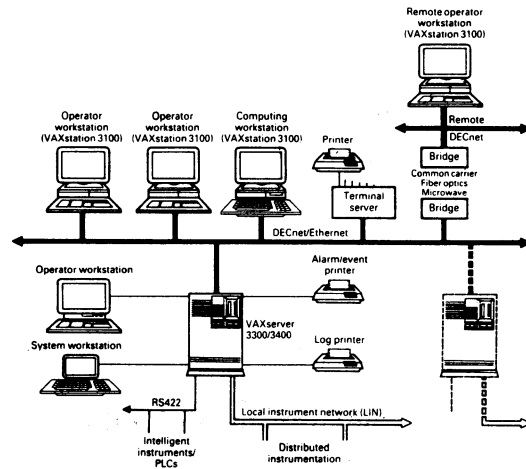


Fig. 1.1 Maxi - Vis IV supervisory computer system.

제어하여 제어한 결과를 Coordinating Controller에 전달한다. Coordinating Controller는 전달된 자료를 분석하여 Local Controller의 상호 관계 및 공장 전체의 목표에 맞추어 새롭게 Local Controller에 지시하므로써 공장 전체가 일관된 목표를 지향하도록 한다.

이러한 분산제어시스템을 구축하려면 Coordinating Controller는 Local Controller가 관장하는 모든 System을 직접 실시간으로 제어 및 감시한 자료를 바탕으로 분석하여야 하므로 두 Controller간에는 고속으로 데이터를 주고 받을 수 있는 통신 모듈이 필요하다.

Fig. 1.1은 TCS 6000 Integrated System의 구성도로서, Distributed Process Control 을 위하여 VMS하에서 동작하는 DEC VAX Server 3300/3400을 근간으로 하고 Operating과 computing을 위하여서 VAX station 3100을 사용하고 있음을 보여 주고 있다. Operating Work Station 에서 조작하면 VAX Server 3300/3400은 RS422를 통하여 Local Controller 인 Intelligent Instrument 또는 Programable Logic Controller(PLC)을 제어하고 제어된 결과는 RS422를 통하여 VAX Server에 수집되며 Operating Workstation에 Display된다.

이제 까지의 분산제어시스템은 주로 Workstation을 중심으로 구성하였으나 Workstation을 능가할 수는 없다 할지라도 PC를 중심으로 Fig 1.1과 같은 저가의 분산제어시스템을 구축하여 공장 자동화를 한다면 여러가지 어려운 여건에 있는 우리의 실정에 있어서 상당히 의미가 있다고 생각된다.

따라서 본 연구에서는 DOS System으로 동작하는 IBM486 호환 PC를 중심으로 분산제어시스템을 구축하기위한 첫 단계로 PC와 하부 Intelligent Instrument 또는 PLC에 data를 주고 받을 수 있는 통신모듈을 개발하는 것을 목적으로 한다. 이러한 분산제어시스템을 구축하기위하여서는 다음과 같은 조건을 만족하도록 설계한다.

1. IBM PC호환기종에는 모두 이식이 가능하도록 가능하면 PC를 개조하지 않아야 한다.
2. PLC등은 고가이므로 Local Controller는 micro processor system을 사용하여 제어할려는 시스

의 발달로 하부레벨(Local Controller)에서는 One Chip Micro processor를 비롯한 8Bit 내지 16Bit Microprocessor 또는 PLC(Programable Logic Controller)를 중간레벨(Coordinating Controller)이상에서는 개인용 컴퓨터 또는 EWS(Engineering Work Station)를 사용하고 이들 시스템간의 통신은 LAN등의 통신망을 사용하는 분산제어시스템을 구축하는 것이 대부분이다.

분산제어시스템에서는 여러 대의 Local Controller가 지정된 제어동작을 하는데 있어 제어의 목표치를 비롯한 여러가지 Parameter를 Coordinating Controller로 부터 지시받아 지시대로

템에 따라 다양한 기능을 용이하게 부여하며, hardware의 수정없이 모든 시스템에 사용할 수 있도록 범용으로 설계한다.

3. Local Controller가 상부 시스템과 통신때문에 실시간 제어가 방해 받지 않도록 한다.

4. Local Controller의 수가 많아지더라도 Coordinating Controller에서는 실시간에 가깝게 제어/감시 할 수 있도록 통신속도는 375KBPS 이상되도록한다.

5. Local Controller 상호간의 serial 통신이 가능하도록한다.

6. Local Controller의 확장이 용이해야하며,확장으로 인하여 부가적인 hardware가 필요치 않아야한다.

이상의 조건을 만족할 수 있도록 하기위하여 Local Controller로서는 제어전용의 16bit Embedded Controller인 인텔의 MCS96으로 구성하고 Coordinating Controller와의 통신을 위해서는 별도의 통신 전용 프로세서인 인텔의 RUPI(Remote Universal Peripheral Interface)44을 사용하여 제어와 통신을 분리시키므로서 통신으로 인하여 제어프로세서가 제어의 지연을 초래하는 일이 없도록하여 확장성을 용이하게 한다.

2. Local Controller의 구성

Local Controller는 분산제어시스템에서의 위치와 기능 및 구성을 간략히 도시하면 Fig. 2.1, Fig. 2.2와 같다. Fig.2.2에서와 같이 Local Controller의 기능은 Plant와 직결되어 Plant를 감시 제어하기 위하여서는 Analog 입출력 장치인 ADC(Analog Digital Converter), DAC(Digital Analog Converter)을, Digital 입출력을 위하여 DI(Digital Input), DO(Digital Output)을, Local Operating을 위하여 Local Operator Input와 Display Unit를, event제어를 위하여 Interrupt System을 갖추어

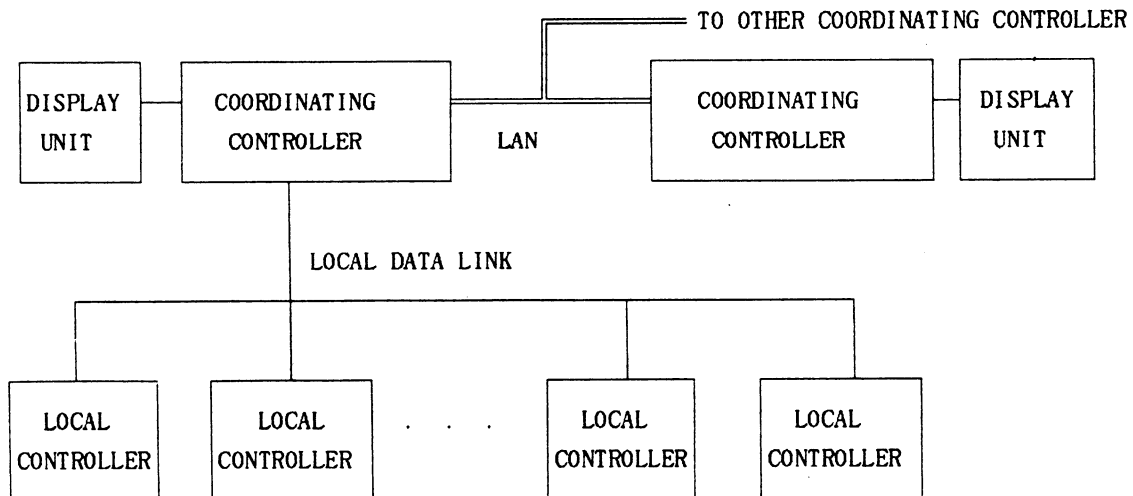


Fig. 2.1 A Typical Distributed Computer Network.

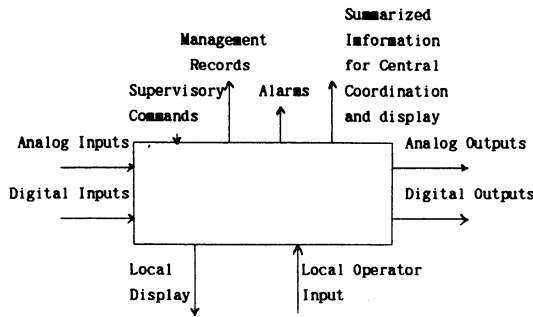


Fig. 2.2 The Main Functions of Local Controller.

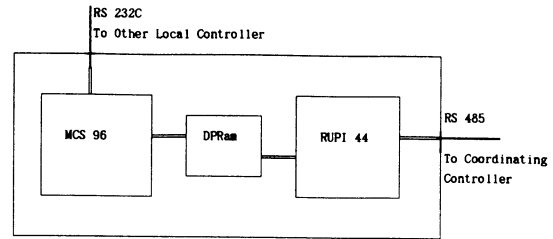


Fig. 2.3 Constitution of Local Controller.

을 위하여 통신시스템이 구축되어야한다.

본 연구에서는 Local Controller의 기능을 제어부와 통신부로 나누고 제어부는 제어전용 프로세서인 MCS96을, 통신부는 통신 전용 프로세서인 RUPI44을 사용하였으며 Fig. 2.3은 이의 블록도로서 제어부와 통신부는 Dual Port Ram(DPRam)으로 연결되어 상부시스템인 Coordinating Controller로 부터의 data와 Local Controller에서 상부시스템으로 줄 data 모두가 DPRam에 있으므로 이 RAM에 있는 data를 사용하기만 하면 되도록 하였다.

야한다. 또한 상위 시스템인 Coordinating Controller와의 통신과 Local Controller와의 통신

2.1 MCS96 Micro Processor 의 개요

Fig. 2.4는 MCS 96계열의 기능도로서 Analo Input로서 10 BIT AD Convertor을 8 Channel, Analo Output로서 PWM출력 1 Channel, DI/O Port로서 8 Bit 2 Port, 21개의 Interrupt Source로 8가지의 Interrupt Type가 있으며, 특히 고속 입출력장치인 HSI, HSO가 있어 언제 어떠한 입력이 있었는지 또는 어떠한 출력을 할 것인지를 program할 수 있다. 또한 Serial Port가 있어 통신이 가능하다. 또한 자체의 8K Byte의 내부 EPROM에 password를 걸 수 있어 porgram의 보안이 가능하며, 16 Bit Microprocessor이므로 64K Byte의 외부 RAM을 가질 수 있다. 뿐만아니라 CPU가 halt되었을 경우 S/W를 reset하도록하는 Watch Dog Timer기능이 있으며 blackout이 발생하였을 때 중요한 data을 기억해 복전이 되었을 때 계속적으로 동작할 수 있도록 16 Byte의 불휘발성 RAM을 가지고 있어 산업에 적용할 때 고려하여야 할 문제중 많은 부분을 hardware적으로 보완해 주고 있다.

2.2 RUPI44(8044) Micro Processor 와 SDLC의 개요

Remote Universal Peripheral Interface 44 Micro Processor은 Intel 계의 8Bit One Chip Micro Controller인 8051을 근간으로 하여 IBM의 Synchronous Data Link Control(SDLC)과 High Level Data Link Control(HDLC) Protocol 을 지원하는 SIU(Serial Interface Unit)을 부가한 통신전용 Micro Processor로서 MCS 8051과의 관계는 Fig. 2.5와 같다.

8 Bit Micro Controller 8051의 구성은 FIg. 2.6과 같이 64K Byte의 Program Memory .64 K Byte Data Memory공간을 가질수 있으며 8 Bit Port 를 4개, 두개의 우선순위를 지정할 수 있는 6

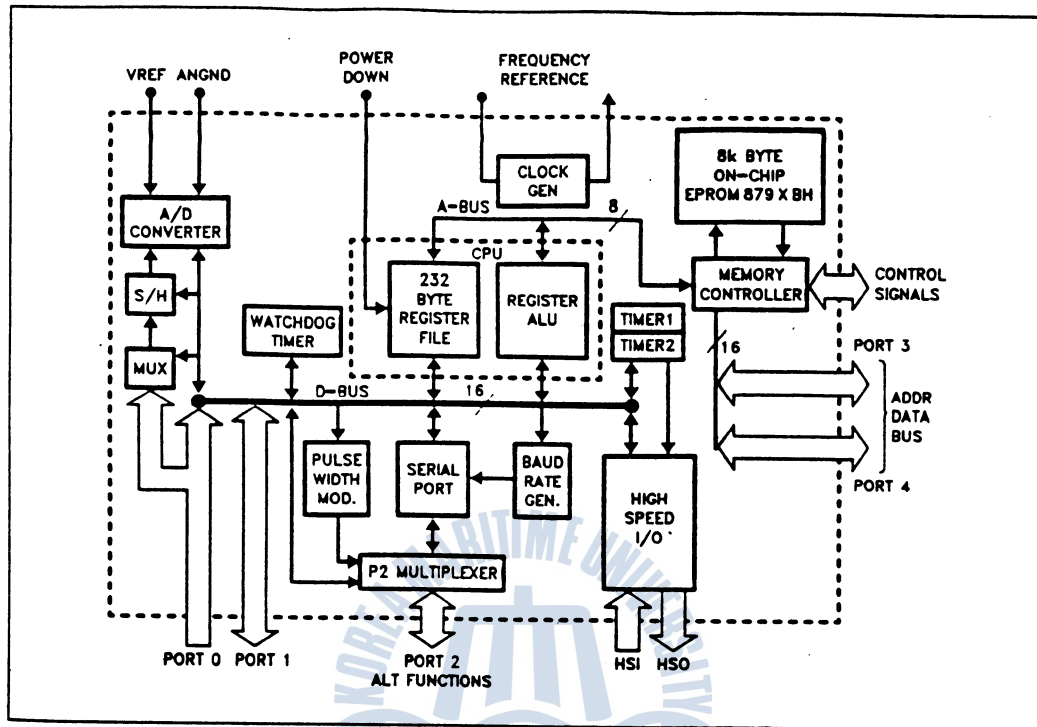


Fig. 2.4 Block Diagram of MSC96.

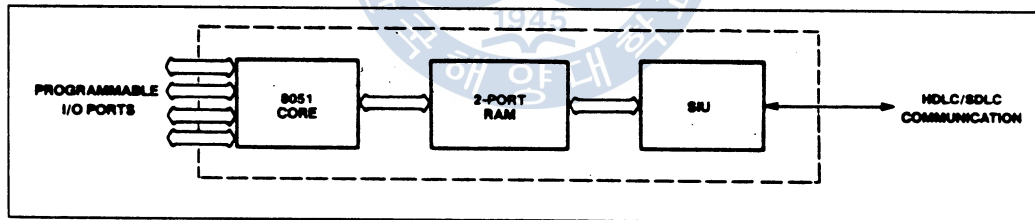


Fig. 2.5 RUP1 44 Dual Controller Architecture.

Source/5 Vector Interrupt, 2개의 외부 Interrupt, Full Duplex UART, 2개의 Counter/Timer, 128 Byte 의 내부 Data RAM이 있는 제어전용 Micro Processor이다.

MCS 8051과 SIU을 합친 RUP1 8044를 간단히 블록도로 표시하면 Fig. 2.7과 같으며 Fig. 2.8은 SIU의 블록도이다.

SIU는 BIT Processor(BIP)과 Byte Processor(BYP)로 나누어져 있으며 이들은 각기 다음과 같은 기능을 나누어 담당하므로써 가능한한 CPU를 관여시키지 않으면서 효율적인 통신이 이루어지도록한다.

2.2.1 Bit Processor(BIP)와 Byte Processor(BYP)

Fig. 2.8에서 BIP는 Serial Port Pin과 SIU Bus 사이에서 Serial Data Stream으로 data를 보내거나 이들 Data Stream으로 부터 data를 받는 일을 한다. BIP의 주요한 기능은 다음과 같다.

- a. Shifting

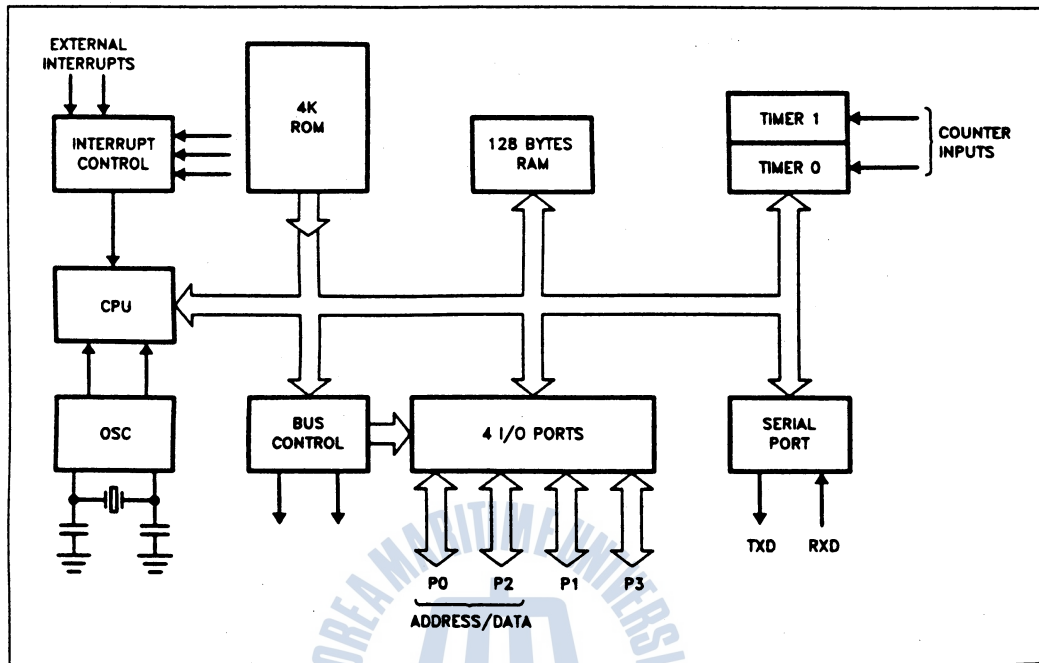


Fig. 2.6 Block Diagram of the 8051 Core.

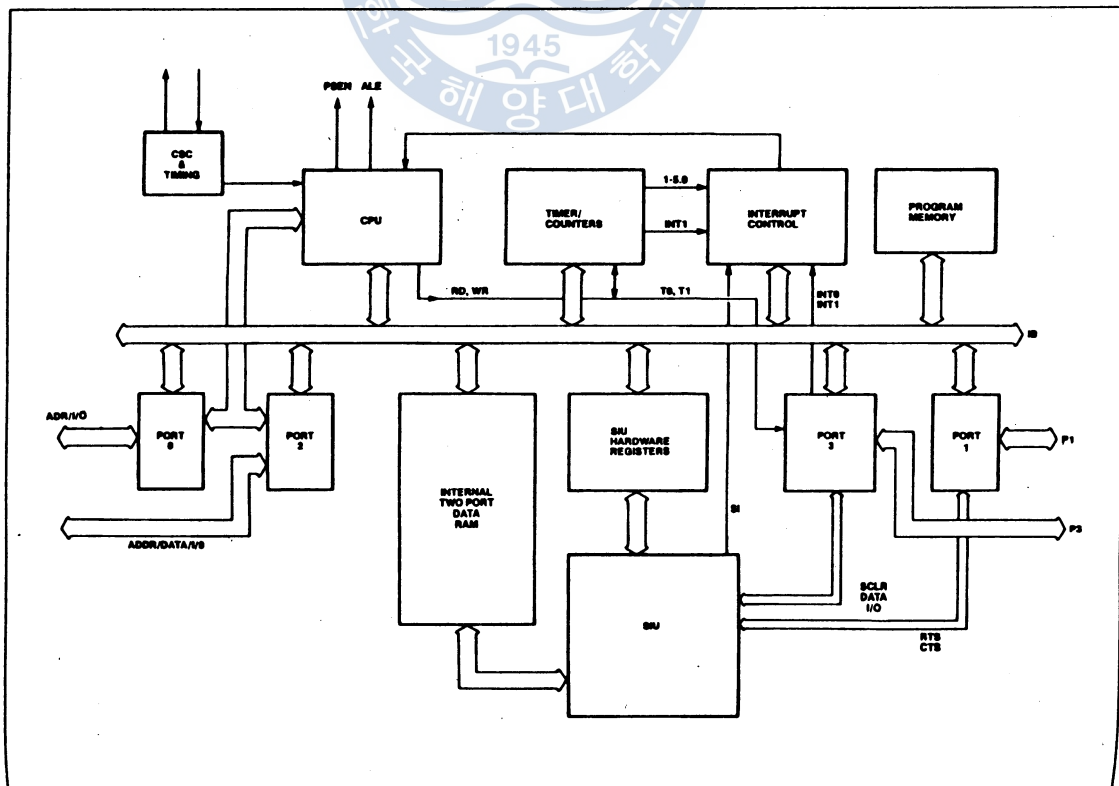


Fig. 2.7 RUPI 8044 Block Diagram.

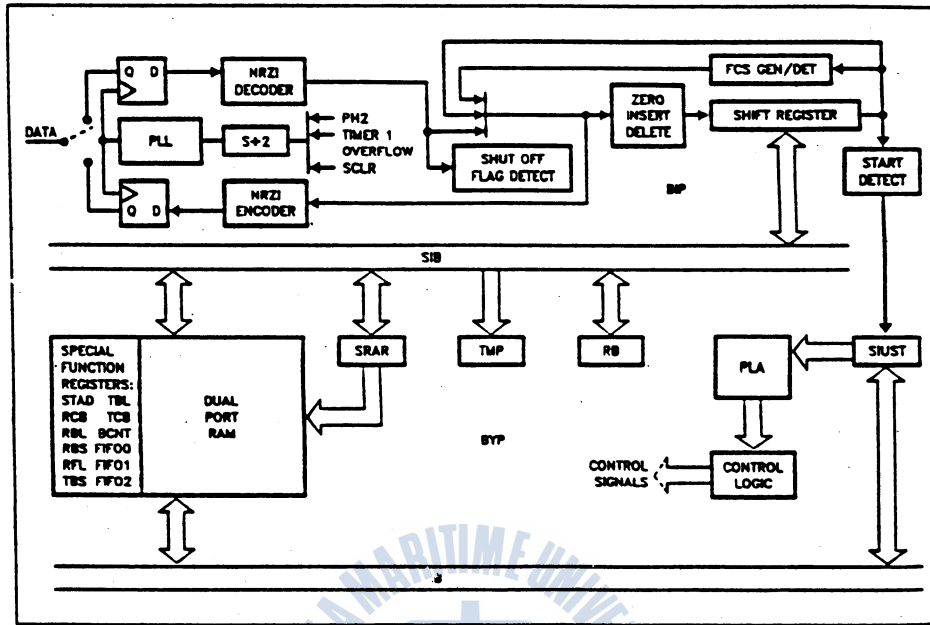


Fig. 2.8 SIU Block Diagram.

- b. NRZI(Non Return to Zero Inverted) encoding/decoding
- c. Zero Insertion/Deletion
- d. FCS(Field Checking Sequence) generation/checking

수신할 때 오류의 확률을 줄이면서 보다 빠른 속도로 전송하기 위한 방법은 NRZL(Non Return to Zero Level), NRZI(Non Return to Zero Inverted), RZ(Return to Zero), Biphas 등 이 있다.

NRZL은 Bit 1, 0을 전압레벨 L1, L2에 할당하여 전송하는 동안 0이 되지 않는 경우로서 Mark을 - 전압, Space을 + 전압으로 하여 1과 0이 변할 때만 천이가 일어나도록하여 영문자 S(1010011B)을 전송 하면 Fig. 2.9와 같이 된다.

NRZI는 Mark와 Space가 Bit 값 자체가 아니라 천이가 있으면 Mark를, 천이가 없으면 Space를 말하는 것으로 영문자 S을 NRZI로 전송할 경우를 그리면 Fig. 2.10과 같다. NRZI에는 NRZM(Non Return to Zero Mark: 천이가 있으면 Mark을 의미)와 NRZS(Non Return to Zero Space: 천이가 있으면 Space을 의미)가 있으며 전압의 값이 Bit의 값이 아니고 천이가 있느냐, 없느냐로 Bit가 정해지므로 NRZL보다 잡음에 강한 장점이 있다.

NRZ의 단점은 NRZL의 경우에는 1 또는 0이 길어지면(NRZS의 경우에는 1이 길어질 때, NRZI의 경우에는 0이 길어 질 때) 신호선은 일정한 전압값을 오래동안 가지게 되어 수신 또는 송신

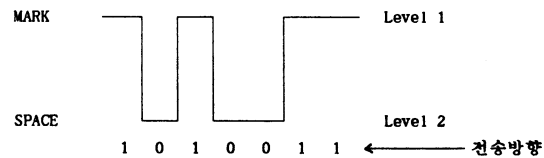


Fig. 2.9 Non Return to zero Level.

시에 동기를 잃어버리게 된다. 이 문제는 Bit의 중간에서 천이가 이루어지도록하여 UART가 천이되는 값에서 부터 Bit동기를 유지하도록 하면 된다. 이것이 RZ Encoding 기법이다. RZ는 Zero와 Nonzero의 값을 가지고 Space는 Zero의 값(천이가 아님)을, Mark는 1 Bit을 송신하는 동안 2번 천이한다. 처음

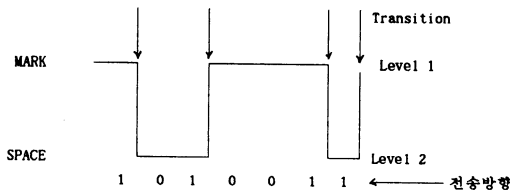


Fig. 2.10 Non Return to Zero Inverted.

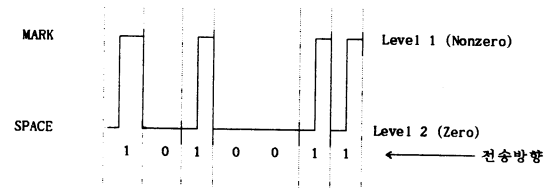


Fig. 2.11 Return to Zero.

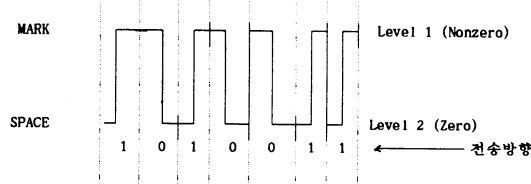


Fig. 2.12 Biphas.

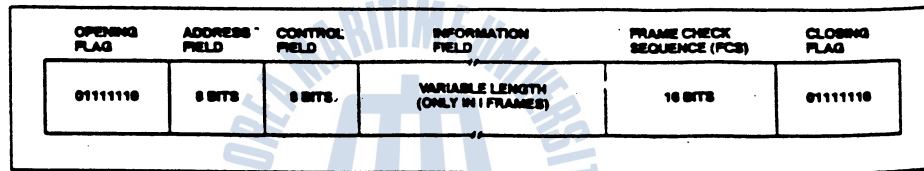


Fig. 2.13 Frame Format OF SDLC.

천이는 Mark bit의 시작에서 다음 천이는 Mark bit의 중간에서 발생하고 그다음은 반은 Zero로 된다. 영문자 S을 RZ로 송신하면 Fig. 2.11과 같다.

RZ는 Mark때만 2번 천이가 있는 데 반하여 고속 통신에 많이 사용되는 Manchester Encoding이라고 하는 Biphas는 모든 Bit에 천이를 일으킨다. 즉 Mark는 High에서 Low로 Space는 그 반대로 하여 수신측에서 양쪽의 Clocking 정보를 얻어 Data을 추출할 수 있도록 하고 송신과 동기되도록한다. 영문자 S을 Biphas로 송신하면 Fig. 2.12와 같다.

Data 통신에서는 Data의 투명성을 확보하기위해 문자가 아닌 Arbitrary Data을 전송할 필요가 있다. HDLC/SDLC에서는 Data의 투명성을 확보하기위해 Zero Bit Insertion을 행한다. 즉 Data Stream의 FLAG(Open Flag와 Closing Flag : 01111110(7EH))을 제외하고 어떤 data라도 연속한 1이 5개 이상이면 0을 삽입한다. 수신측에서는 Open Flag을 수신하고 난뒤 부터는 수신측의 hardware에 의해 5개의 연속 1을 수신한 다음의 0은 삭제되어 원래의 Data를 복귀한다.

Fig. 2.8에서 수신되는 Data Bit Stream 은 NRZI Decoder에 의해 decoding되어 ZID을 통과하면서 원래의 data로 복귀시키며, Flag 또는 GA(Go Ahead)을 감지한다. 원래되로 복귀된 data는 Shift Register(SR)에 의해 Serial to Parallel로 변환되면 BIP는 BYP로 하여금 Byte을 처리하도록하고 다음 Byte을 SR에 load한다. 또한 serial data는 SR을 통해 16Bit FCS GEN/CHK Register에 의해 CRC을 검사한다.

BYP는 SR로 부터 받은 Address와 Control Byte를 적합한 Register에 옮기고 SR의 내용이 Information Byte이면 3 Byte FIFO을 통해 RBS(Receive Buffer Address) Register가 가지고 있는 RAM번지에 data을 이동시킨다.

data을 송신할 때는 BYP는 data을 RB(RAM Buffer)에 이동시킴으로서 BIP가 이를 처리토록한다.

BIP는 data을 SR을 통하여 Serial Data Stream으로 변환시키고 ZI, NRZI Encoding하여 Serial Port에서 전송케한다. 마지막으로 FCS GEN/CHK의 내용과 Closing Flag을 전송한다.

2.2.2 SDLC(Synchronous Data Link Control) Protocol

SDLC는 IBM 표준 통신 Protocol로서 비동기통신 Protocol에 비해 data 투명성이 간단하고 Message Level 로 동작하며, FCS에의해 신뢰성이 높으며 많은 byte의 data을 한 Frame으로 하여 보낼 수 있다는 것이다. Fig. 2.13은 SDLC의 Frame Format이다. SDLC의 Frame Format는 잘 서술된 형태의 Format로서 Control Field 나 Information Field를 결정하기 위해 비동기식에서와 같이 CPU가 수신한 문자의 의미를 해석할 필요가 없을 뿐만아니라 Frame의 Identification Number을 송신함으로서 통신의 신뢰성을 향상시킬 수 있다.

HDLC/SDLC의 Network은 Point to Point(Half Duplex), Multipoint(Half Duplex)와 Loop형으로 구성할 수 있으나, 본 연구에서는 분산제어시스템을 구성하기위해 Coordinating Controller의 data을 Local Controller에 전달하고 Local Controller의 제어결과를 상위레벨의 Coordinating Controller에 전달하여 display하면 되므로 Multipoint(Half Duplex)로 구성하기로 한다.이를 도시하면 Fig. 2.14와 같다.

Fig. 2.14에서 Primary는 Coordinating Controller의 통신모듈(Master)이고 Secondary는 각

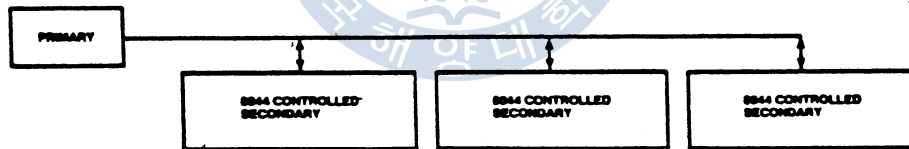


Fig. 2.14 Multipoint Half Duplex Network.

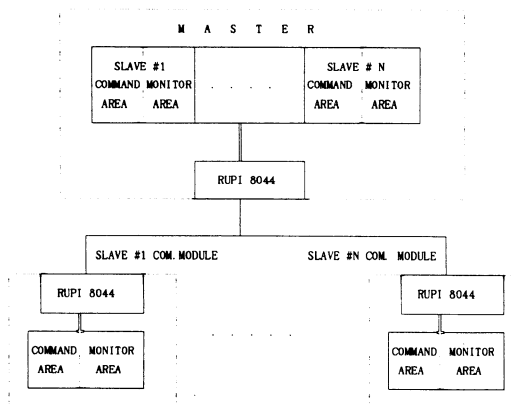


Fig. 2.15 Memory Configuration of Proposed System.

Local Controller의 통신모듈(Slave)이다.

본 연구에서는 Master의 DPRam을 각 Slave에 해당되는 영역을 할당하고 각 영역을 Master에서 Slave로 지시하는 영역(Command Area)과 각 Slave의 제어 결과를 Master로 보고할 영역(Monitor Area)로 나누며, 각 Slave도 같은 방법으로 영역을 나누어 Slave는 Command Area의 data을 Reference로 하여 제어하고 결과를 Monitor Area에 쓰는 작업만하고 각 Area의 송,수신은 RUPI 44가 전담함으로서 통신으로 인하여 실시간 제어

에 방해가 되지 않도록 구성하였다. Fig. 2.15는 본 연구에서 구성한 시스템의 Memory 구성이다.

3. 개발된 시스템의 Hardware와 Software

제안된 시스템은 PC로서 분산제어시스템을 구축하는 것이므로 사용하고 있는 PC에 Master Card를 PC의 Slot에 꽂으면 간단히 구축되어야 한다. 따라서 현재 사용하고 있는 PC의 Memory위에 DPRam이 존재해야하므로 사용되는 PC의 Memory크기(4, 16 MByte 등)를 감안하여 간단히 Jumper만 바꾸어 범용성을 기하였다. 또한 RS485를 사용하여 최대 32개의 Slave를 접속할 수 있고 최대거리 1.2 Km까지 사용할 수 있어 공장자동화에는 이상적이라 할 수 있으며 전송선도 Twisted Pair로서 배선도 용이하다.

Table 1은 각 방식의 사양을 비교한것으로 RS-422과 RS-485는 기본적으로 동일하며 단지 RS-422은 Point to Point의 경우에, RS-485는 Bus방식으로 Network형성이 가능하다.

DPRam을 통하여 제어전용 Microprocessor와 연결되므로 RUPI44는 Fig. 3.1과 같이 송수신할 수

Table 1. Electrical Characteristic Driver/Receiver.

동 작		RS - 232C	RS - 422A	RS - 423A
		불 평 형	평 형	불 평 형
Driver	최대 Cable 길이	15m	1.2km	600m
	최대 Data Rate	20K Baud	10M Baud	300k Baud
	무부하 출력(절대치)	25V이하	출력간 6V이하	6V이하
	부하출력(절대치)	5-15V	출력간 2V이상	3.6V이상
	Power Off시 출력저항	300Ω이상	출력에 6 - 0.25V을 걸어 100μA이하	출력에 -6 - +6V을 걸어 100μA이하
	단락시출력전류(절대치)	500mA이하	150mA이하	150mA이하
	Slew Rate	30 V/μs이하	-	-
Receiver	입력저항	3k-7kΩ	4kΩ이상	4kΩ이상
	Threshold	-3- +3V	-0.2- +0.2V	-0.2- +0.2V
	입력최대전압	±25V	±12	±12

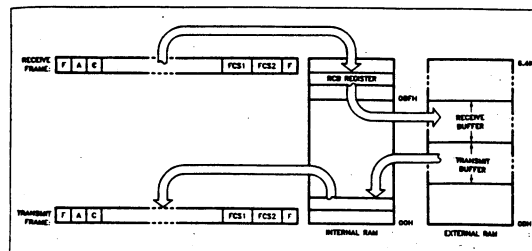


Fig. 3.1 Transmission/Reception Data Flow Using External RAM.

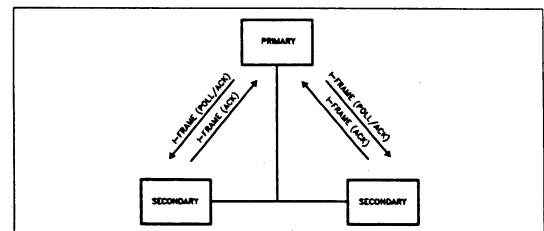


Fig. 3.2 Polling Sequence Between the Master and Slave.

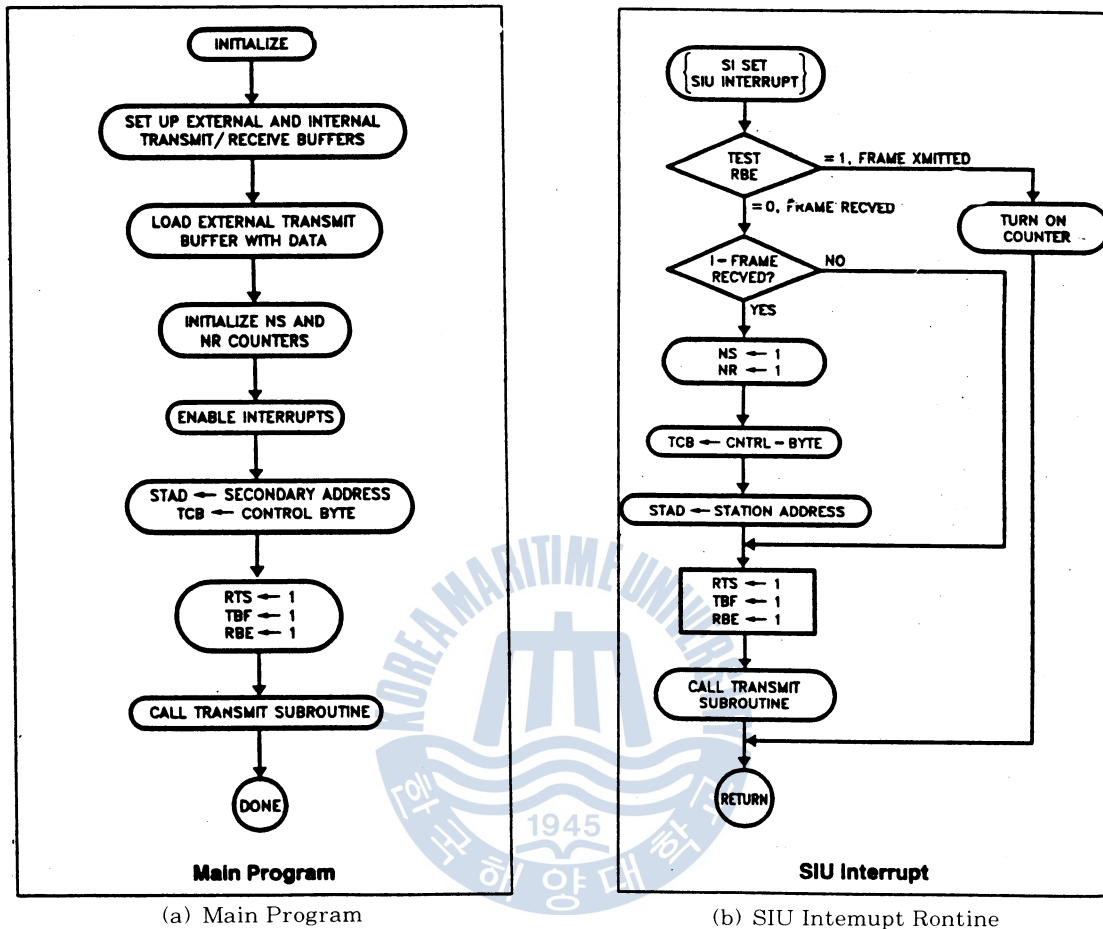
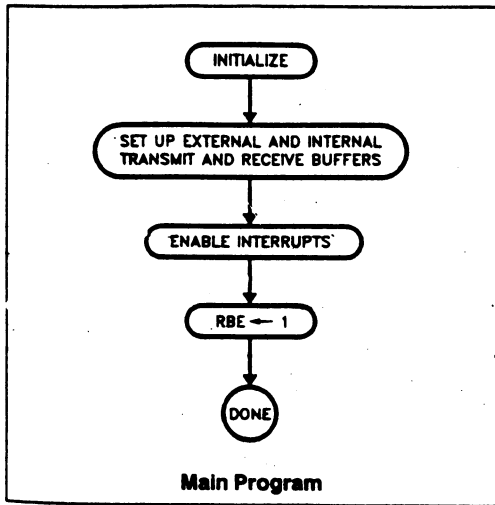


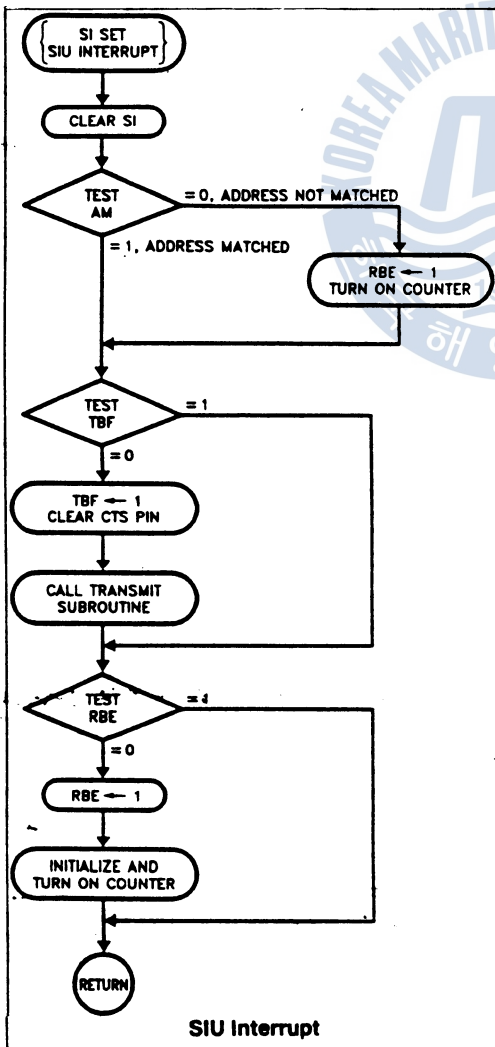
Fig. 3.3 Master Station Flow Char.

있도록 설계한다. 먼저 Master에서 해당 Slave로 data을 송신하고 해당 Slave가 data을 수신후 Master로 Slave가 송신한다. Master는 Slave로 부터 data을 수신한 후 다음 Slave로 data을 송신하여 관장하고 있는 Local Controller모두에게 순차적으로 data을 송수신한다.

먼저 Master가 data을 송신할 때 2 Byte의 Preframe을 발하므로 이것에 의해 Slave는 Timer Interrupt 가 걸려 수신 Routine에 들어가 Address을 받고 자기 Address와 같으면 ACK하여 계속 수신하고 자기 Address가 아니면 Abort한다. Control Field와 I Frame을 다 수신하면 ACK하고 ACK가 없으면 재송신한다. 송신을 완료한 Master는 수신준비를 하고 Preframe을 기다린다. 한편 수신을 완료한 Slave는 Master에 송신하기위해 Preframe을 보내고 Master는 Timer interrupt Routine에서 수신 Mode가 되어 계속 수신한다. Slave가 Master로 부터 ACK을 받으면 송신 Routine을 빠져 나와 수신 Mode가되어 다음 Preframe을 기다린다. 수신을 완료한 Master는 SI Interrupt Routine에서 다음 Slave 에 송신하기 위한 절차에 들어간다. 이 절차를 간단히 도시하면 Fig. 3.2와 같고 Fig 3.3, 3.4.는 Master와 Slave의 Main Program, SI Interrupt Routine 이며 Fig. 3.5와 3.6은 송신과 Slave 수신의 Flowchart이다.



(a) Main Program



(b) SIU Interrupt Routine

Fig. 3.4 Slave Station Flow Chart

4. 실험 및 고찰

Photo 1과 2는 PC의 Slot에 꽂히는 Master와 Local Controller에 장착되는 Slave Module이다.

실험에서는 Master와 Slave가 서로 주고 받는 data수를 각각 128 Byte(2 Byte 정수로하면 각각 64개의 정수 Data가 된다)로 하고 2개의 Slave로 Network을 형성하여 송수신 실험을 하고 송수신 상태를 Oscilloscope로 관찰하였다.

Photo 3은 Master와 Slave #1의 송신상태이고 Photo 4는 Slave#1과 Slave#2의 송신 상태이다.

Photo. 3에서 위의 것이 Master이고 아래가 Slave 1의 Graph이다. 모두 Low에 있는 동안이 송신기간이고 High에 있는 기간이 수신 및 대기

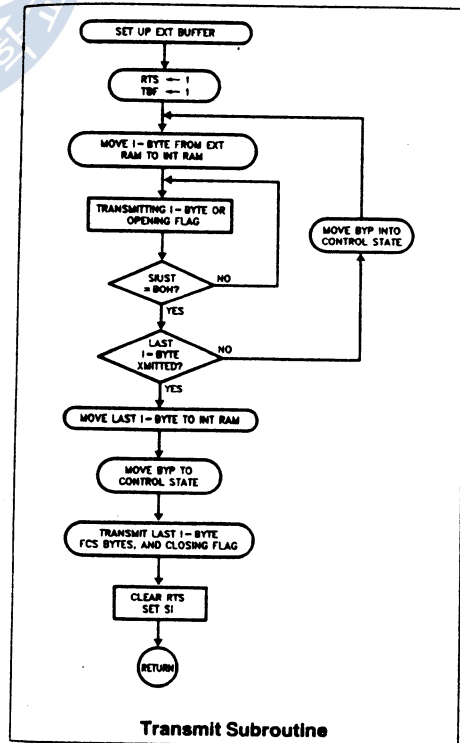


Fig. 3.5 Transmit Flow Chart.

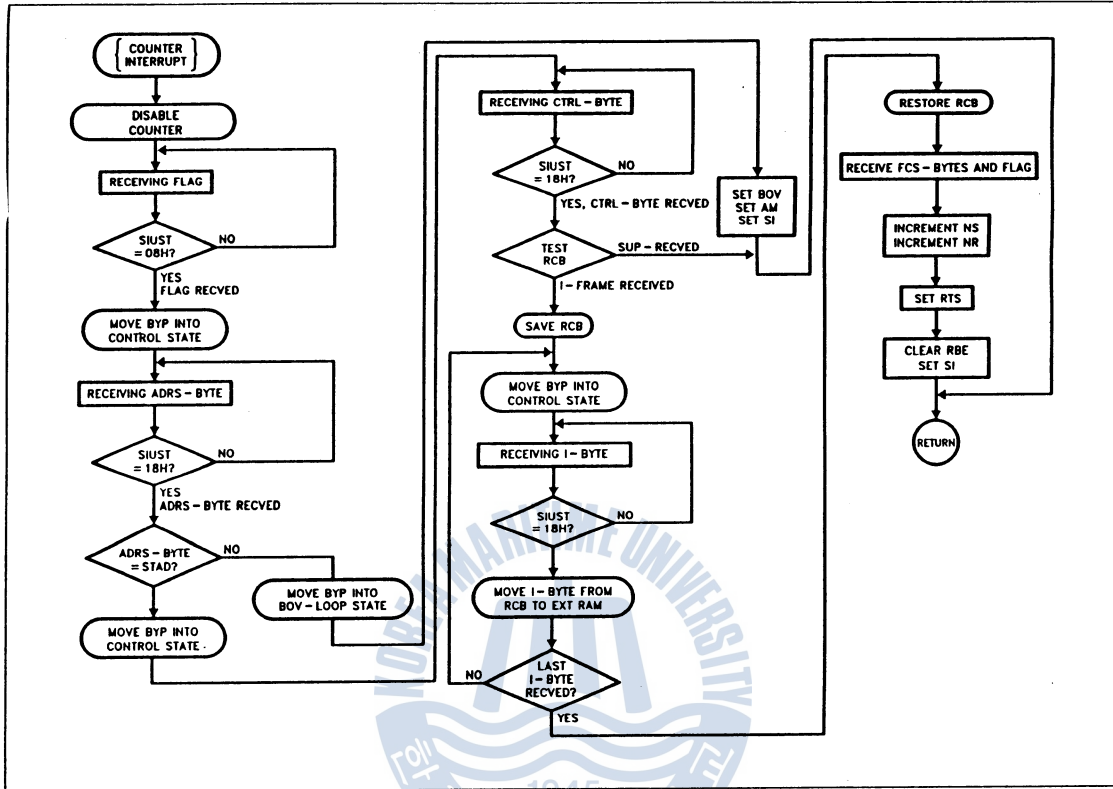


Fig. 3.6 Slave Receive Flow Chart.

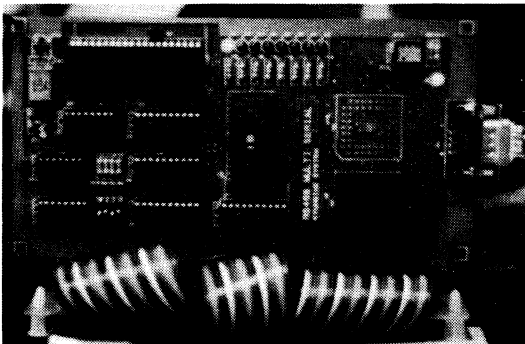


Photo. 1. Photograph of Master Module.

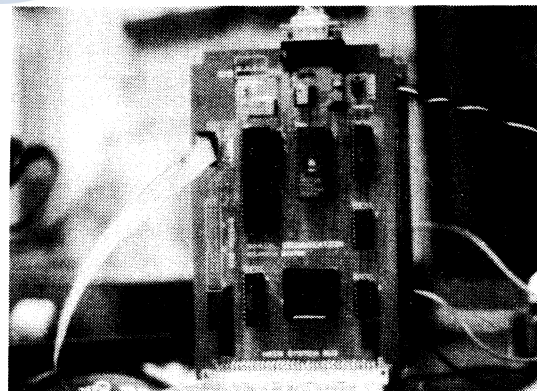


Photo. 2. Photograph of Slave Module.

기간. 다음 Sequence 준비기간이다. Master의 송신 직후 Slave가 송신하는 것을 알 수 있다.

Photo. 4에서 위의 것이 Slave 1, 아래가 Slave 2의 것이고 모두 Low에서 송신한다.

사진에 보는 바와 같이 Slave가 2 Set이므로 다른 Slave의 대기기간 중간에서 또 다른 Slave의 송신이 이루어지고 있음을 알 수 있다. 두 사진에서 송신기간의 차이가 발생하는 것은 ACK을 받지 못하여 재

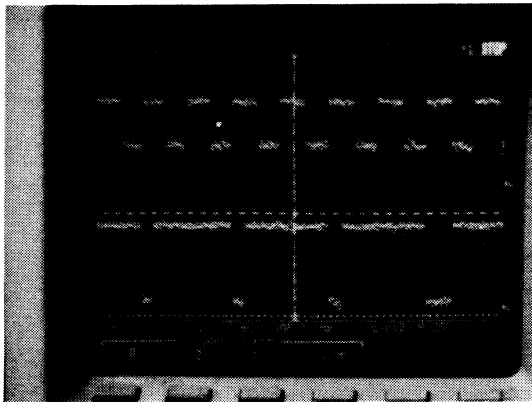


Photo. 3. Transmission Status Between Master and #1 Slave.

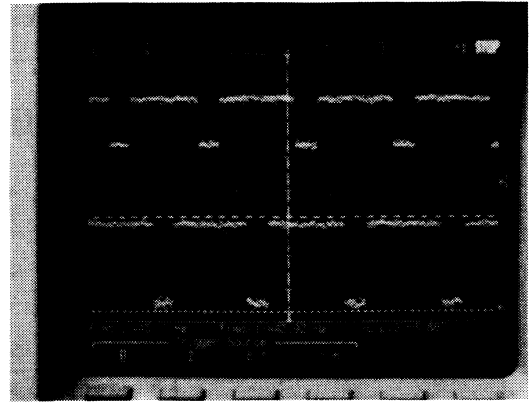


Photo. 4. Transmission Status Between #1 Slave and #2 Slave.

송신이 이루어 지기 때문에 생각 된다.

실험에서 375 KHz의 동기 Clock을 사용하여 장기간 원활한 동작을 관찰하여 375 Kbps의 송수신이 가능함을 확인 하였다.

5. 결 론

인텔의 통신전용 Microprocessor RUPI 8044을 이용하여 375Kbps으로 통신할 수 있는 SDLC Data통신망을 구축하고 실험을 통하여 성능을 확인하였으며, 전송된 data을 PC로 확인 한 결과 이상없이 통신할 수 있음을 입증하였다.

이로서 Local Controller로서 제어전용의 MCS 96 Microprocessor와 통신 전용의 RUPI 8044 Microprocessor 을 장착하고 Coordinating Controller로서 범용 PC을 사용한 저가의 분산제어시스템의 구축가능성을 확인하였다.

앞으로 연구하여야 할 과제로서 실제 공장과 같은 환경이 열악한 환경에서 실험을 행하여 오동작없이 강건하게 동작할 수 있는지 여부를 확인하고 시스템을 보완하는 문제가 남아 있다고 하겠다.

참고문헌

- 1) J.R. Leigh, "Applied Digital Control, theory, design and implementation 2nd ED.", 1992 Prentice Hall
- 2) Larry Hughes, "Data Communications", 1992, McGraw Hill
- 3) 宮崎誠一 著, "データ 轉送技術實用ノウハウのすべて", 1991 CQ出版
- 4) "トランジスタ技術 SPECIAL No.8" CQ出版, 1992
- 5) INTEL, "16BIT Embedded Controllers", 1990
- 6) INTEL, "EMBEDDED APPLICATION", 1991
- 7) INTEL, "8 BIT EMBEDDED CONTROLLERS" 1991