Serial 전송라인에서 Multi-Protocol 통신의 구현 연구

A study of multi protocol communication on single serial interface

이재철*, 고대식*

Jae-Cheol Lee*, Dae-Sik Ko*

요 약

Serial 통신에는 RS-232C 통신과 같은 Un-balanced 방식과 RS-485/422통신과 같은 Balanced 통신 방식이 있다. 이 가운데 RS-485 통신의 경우는 통신의 거리가 길고 Balanced 방식에 의한 전송방식이기 때문에 RS-232에 비하여 데이터의 신뢰성이 매우 우수하므로 산업현장에 많이 사용되고 있다. 통상적으로는 하나의 통신선 위에서는 하나의 프로토콜을 이용하여 통신을 구현하는 것이 일반적인 방법이다. 본 연구에서는 RS-485 serial 전송의 Physical layer 위에 여러 가지 종류의 장비나 기기를 동시 연결하여 사용할 수 있도록, 다수의 통신 protocol을 구현하는 방법을 실현하여 보고 성능을 검증하여 보고자 한다.

Abstract

The RS-232C (Unbalanced serial) and RS-485 (balanced serial) are standard in serial communication. RS-485 uses a differential electrical signal, as opposed to unbalanced signals referenced to ground with the RS-232. Differential transmission, which uses two lines each for transmit and receive signals, results in greater noise immunity and longer distances as compared to the RS-232. The greater noise immunity and distance are big advantages in industrial environments. In general, one protocol on one serial interface is very normal application. In this study, we implemented multi protocols on one serial RS-485 interface and examine the performance and result with hanging multi equipments.

Key words : Protocol, RS-485, RS-232C, Physical layer, MODBUS, Task schedule

I. 서 론

1-1. Un-balanced Line driver

RS-232C 표준 serial 통신방식이 대표적인 Un-balanced line driver 방식으로, 모든 전송신호의 기준점을 전송시스템의 Ground 신호로 하여 Connector로 송신신호를 내 보내고 수신 측에서는 역시 전송시스템의 Ground 신호를 기준으로 입력된 신호를 받아들이는 신호 송수신 방식으로 두 시스템간의 신호의 기준인 Ground를 서로 연결하게 된다. 그림1에 Unbalance Line driver interface를 보인다.[3]

* 목원대학교 대학원 IT공학과 (College of IT engineering, Mok-won University)
. 제1저자 (First author) : 이재철
· 투고일자 : 2008년 9월 18일
· 심사(수정)일자 : 2008년 9월 19일 (수정일자 : 2008년 10월 20일)
· 게재일자 : 2008년 10월 30일

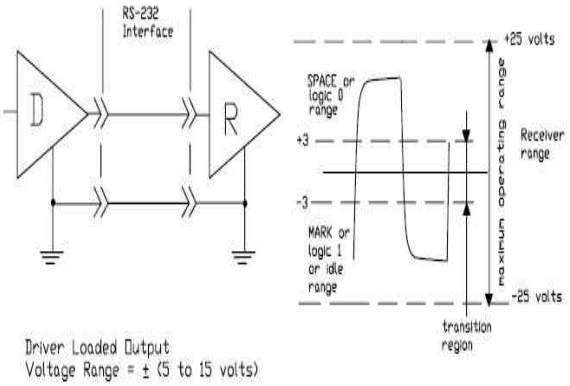

그림1, Unbalanced Line driver interface
Fig1. Unbalanced Line Driver interface

RS-232C 수신 장치는 통상적으로 +3에서 +12V사이, -3V에서 -12V사이의 레벨에서 동작을 한다. IDLE 상태는 Negative voltage에서 유지하고 신호를 반복할 때는 Negative와 Positive 전압의 사이를 반복하면서 출력한다. (전압의 레벨은 -5V - +5V사이를 사용하는 5V 레벨과 -15V와 +15V 사이를 사용하는 15V 레벨이 있다)

1-2. Balanced line Driver

Balanced Line Driver 방식의 대표적인 것으로 RS-485와 RS-422 방식을 들 수 있다. 그림2에서 보듯이 Balanced line driver 방식에서는 전송신호를 만들 때 시스템의 Ground를 기준점으로 하지 않고 Driver의 출력 반전/비 반전 신호의 두 pair를 이용한다.[3]

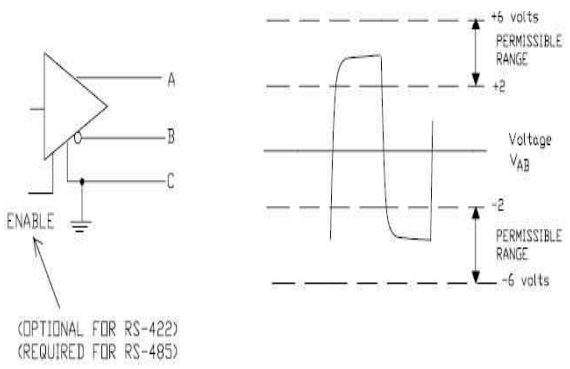

그림2. Balanced Line driver interface
Fig2. Balanced Line driver interface

Driver의 비 반전 신호를 출력하는 신호를 A, 반전 신호의 출력을 B로 표현하며 이들 사이의 신호레벨은 Ground를 기준으로 Positive로 2~6V 사이를, Negative로 -2 ~ -6V 사이를 번갈아 반복하면서 신호를 전송한다. 시스템의 기준전압인 Ground 전위를 기준으로 A와 B입력 각각은 같은 전위를 유지 하므로 Balanced 방식이라고 한다.

Ground(C)는 시스템 간의 안정을 위해 필요에 따라 연결을 할 수 있지만 데이터 전송라인 신호의 Logic 상태를 결정하지는 않는다. RS-485의 경우는 양방향 반 이중 통신방식으로 IDLE 상태인 경우에는 출력의 상태를 OFF하여 Tri-state를 만들어야 같은 신호 선에 연결된 다른 기기들이 신호를 전송할 수 있으므로 ENABLE 신호가 필수이다. 본 연구에서는 서로 다른 여러 종류의 protocol을 사용하는 기기들을 RS-485 Line에 연결하여 사용하는 방법을 연구한다.

II. 본 론

연구에서 사용할 Protocol은 MODBUS-RTU, MODBUS-ASCII 그리고 RS-485기본 Format으로 3가지의 Protocol이 하나의 RS-485 Serial에서 실행될 수 있도록 구현을 해 보기로 한다.

2-1. MODBUS Protocol

MODICON사에서 개발된 공통 사용언어의 하나로 MODICON controller 장치를 통신 네트워크의 형태에 상관없이 인식하고 사용할 수 있는 메시지의 구조를 정의한 OSI layer의 Layer 2인 Data link layer에 해당하는 protocol이다 [1][2]

표준 MODBUS는 RS-232C / RS-485 호환 Serial interface상에서 사용되어지며 상호간의 통신은 Master-slave 기법으로 통신한다. 즉 한편에서 요구를 하고(Master) 다른 한편에서는 응답을 하는 것과 같은(Slave) Unicast 방식으로, Master는 주소필드를 사용하여 개별 Slave를 지정할 수 있고 또는 Broadcasting 방식으로 모든 Slave에게 통신을 개시할 수도 있다. 지정된 Slave 하나만 일대일 통신할 수 있으며 Broadcasting의 경우는 응답의 절차는 없다. 아

래의 그림3은 MODBUS에서 사용되는 Query(요구) / 응답 Message Format을 보인 것이다.[1][2]

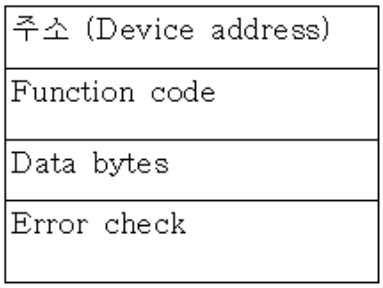

그림3. MODBUS Message 포맷
Fig3. Message format of MODBUS

표준 MODBUS 는 ASCII mode와 RTU mode 두 가지의 전송모드를 지원 한다.

1 RTU mode (Remote Terminal Unit)

Binary 형태로 메시지를 전송하는 방법으로 ASCII 모드에 비하여 두 배 정도의 빠른 속도로 메시지를 전송한다는 장점이 있다. 단 각 메시지는 반드시 한번의 stream으로 모두 보내져야 한다.

- Coding system : 8 bit binary
- Bit per byte : 1 start, 8 data bit(LSB first), Parity 1 bit(odd/even), stop 1
- 에러검출 : CRC16 (Cyclic redundancy check) 방식
- Framing (최대 256 byte)

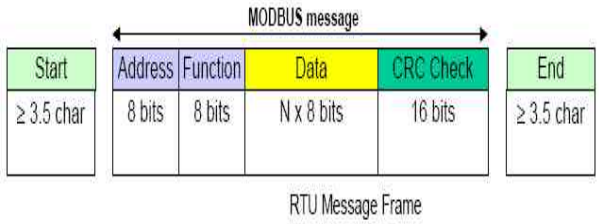

그림4. MODBUS-RTU frame
Fig4. MODBUS-RTU frame

Frame은 최소 3.5char 시간 동안의 silent time으로 시작되며 처음 필드로 Device를 지정하는 Address가 따라 나온다. Network에 연결된 기기들은 자신의 Address를 기다리다가 수신한 값이address와 같은지 확인 후 같으면 자신의 수신 frame로 간주하고 이하의 Function code, data field, CRC field, end delimiter등을 처리한다.

2 ASCII mode

전송될 모든 MODBUS의 데이터는 8비트 한 바이트의 단위로 각 바이트 정보는 두 개의 ASCII 문자로 변환하여 전송된다. 이 모드는 에러 발생이 없이 문자들 사이에 1초까지의 Time interval을 허용한다는 장점이 있다. 즉 discontinue하게 전송이 가능하다. 단 같은 baud rate에서 RTU mode에 비하여 약 2배의 전송속도를 필요로 한다.

- Coding system : 16진수, 각 digit값 0-9, A-F 값은 ASCII 문자의 값
- Bit : 1 start, 7 data bits(LSB first), Parity 1 bit (odd/even), stop 1 bit
- 에러 책크 : LRC (Longitudinal Redundancy Check) 방식
- Framing (최대 513 char)

그림5. MODBUS-ASCII Frame
Fig5. MODBUS-ASCII Frame

Colon(:)으로 frame이 시작되며 CR/LF 두 바이트로 종료를 알리게 된다. Address field부터 데이터 필드까지의 데이터는 ASCII code문자 0-9 (30-39), A-F (3A-3F) 까지만 허용한다. 즉 모든 8 bit의 데이터는 두 개의 ASCII char로 변환되어 전송되어져야 한다. Network에 연결된 기기들은 Colon (:) char를 기다리고 :이 수신되면 Address 필드의 두 char를 읽어서 자신의 address와 같은지 확인 후 같으면 자신의 수신 frame로 간주하고 이하의Function code, data field, LRC field, end delimiter등을 처리한다.

Char들 사이는 1초까지의 시간적 interval 을 허용하지만 그 이상은 에러로 처리한다.

III. 실험방법 및 system 구현

3-1. 실험 시스템의 구성

아래 그림6은 2wire RS-485 통신라인에 여러 종류의 장치를 연결한 구성도 이다. Board에 연결된 Components들은 실시간 처리를 하고, WI값의 무게를 읽어서 Graphic panel에 표시하며 Graphic panel에 설정된 파라메터 값을 읽어서 ST board의 제어에 사용하는 시스템을 설계 하였다.

RS-485 BUS 상에 기본 Format으로 통신을 하는 Component 제어를 위한 ST board와, MODBUS-RTU Format으로 통신을 하는 Graphic panel, 무게 측정 위해 MODBUS-ASCII Format으로 통신을 하는 Weighing Indicator를 연결한다. 좌측의 board는 Master로 동작을 하고 나머지 ST Board 및 Graphic panel, Weighing Indicator는 Slave mode로 동작 한다 [6]

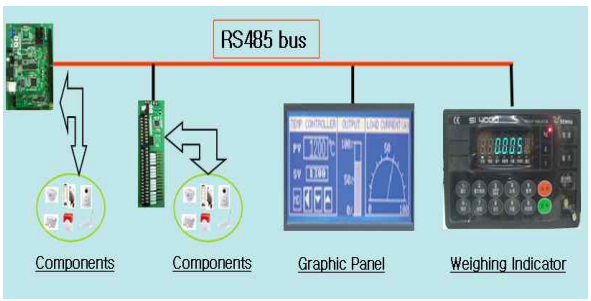

그림6. 실험 시스템의 구성
Fig6. Configuration of E.U.T.

ST (RS-485 제어 Board): Atmega128 8 bit processor

RS-485 통신 속도 : 38400bps, Data 8 bits, No parity, 1 stop bit

Graphic Panel : GP-2480 (AUTONICs), MODBUS-RTU protocol 사용

Weighing Indicator : SI-4000 (세화 CNM), MODBUS-ASCII protocol 사용

1 소프트웨어 설계 고려

첫째로 RS-485 2wire 방식에서의 Data Collision 고려하여 Master는 TX 종료 후 즉시 Tri-state로 이동하여 slave가 송신할 수 있도록 해야 한다.

둘째로 Slave의 응답시간과 response time을 계산해야 한다. 오직 하나의 마스터와 다수의 slave를 시용하는 Master / slave system 구현한다.

.2 실행 TASK의 결정

WI read(17.4msec 소요) : 200msec 마다 무게를 측정하는 TASK

WI write(3msec 소요) : 영점조정, 가끔

GP read(20msec 소요) : 400msec 마다

GP write(10msec 소요) : 200msec 마다

ST r/w(10msec 소요) : 50msec 마다

3 각 Task의 실행소요시간 (38400bps)

각각의 작업을 TASK로 구분하여 Software처리를 계산하여 보면

ST write Command : 10 byte (30usec x 10x10) = 3msec

ST Read Response : 10 byte (30usec x 10x10) = 3msec

WI Read Command : 8 byte (30usec x 8x10) = 2.4msec ; WI format 참조

WI Write Command : 8 byte (30usec x 8x10) = 2.4msec ; 영점조정

WI Read Response : 24 byte (30usec x 24x10) = 7.2msec

GP Read Command : 8 byte (30usec x 8x10) = 2.4msec ; Parameter read

GP Read Response : 35byte (30usec x 45x10) = 13.5msec ; Parameter 값

GP Write Command : 21byte (30usec x 21x10) = 6.3msec ; 운전화면 표시

GP Write Response : 8 byte (30usec x 8x10) = 2.4msec

4 TASK scheduling

아래 그림7은 Time tick에 따른 schedule 되어진 Task의 배열을 보인 것이다. Time tick이 짧을수록 실시간 성이 높고 빠른 처리를 한다는 의미를 가진다.

T000	T050	T100	T150	T200	T250	T300	T350	T400	T450	T500	T550	T600	T650
ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST
ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST
WIR				WIR				WIR				WIR	
	GPW				GPW				GPW				GPW
		GPR								GPR			
28	20	30	10	28	20	10	10	28	20	30	10	28	20msec
13	10	14	07	13	10	7	7	13	10	14	7	13	10msec

그림7. Task scheduling
Fig7. Task scheduling of EUT

Time tick마다 해당된 Task를 수행하는 방법으로 시험을 구현하여 보았다. 위의 표에서 보듯이 50msec의 Time tick 마다 지정된 Task를 수행하는데 Task 의 실행시간이 모두 50msec 이내에서 종료될 수 있게 되었다. WIR(Read fr. weighing indicator) task는 200msec 마다 행하고, GPW (Write to Graphic panel)은 200msec 마다, GPR(Read graphic panel)은 400msec마다, 마지막으로 ST(RS-485 board read/write)는 매 50msec마다 수행하게 하였다.

매 Time마다 실행 소요시간을 보면 Baud rate가 38400bps일 때와 115,200bps의 두 경우를 비교하여 보면 충분히 50msec내에서 처리를 하고 있는 모습을 보였고 Baud rate을 115,200bps로 하면 최대 소요시간이 14msec로, Time tick을 20msec 정도로 구현을 하여도 문제가 없음을 나타내고 있다.

3-2 실험결과

아래 그림8는 3개의 Protocol이 연속으로 데이터를 주고받는 파형을 관찰 한 것이다. (RS-485 Format, Graphic panel의 ModBus_RTU, Weighing Indicator의 ModBus_ASCII 통신)

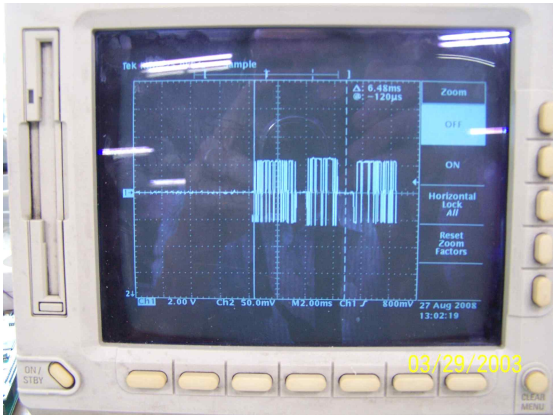

그림8. RS-485 Multi 신호파형
Fig8. Waveform of Multi-Protocols

아래의 그림9 파형은 Balanced serial 통신의 파형을 자세히 보인 것이다.

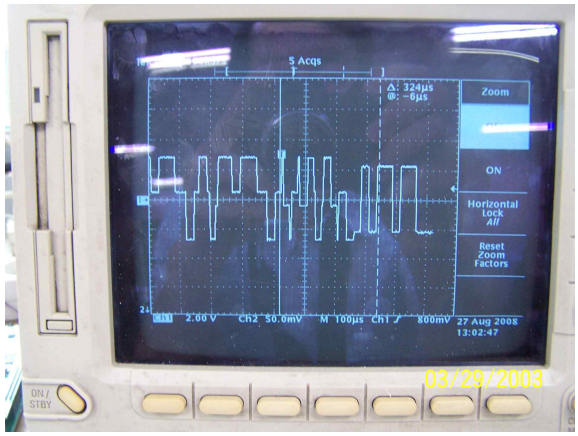

그림9. RS-485 신호파형
Fig9. Waveform of RS485 signal

1 실제 측정된 시간

WI read : command 8bytes (2.4msec) + Wait (1-5msec) + response 24bytes (4msec+3msec+3msec) = 17.4msec (현재중량을 읽는데 소요시간)

GP read : Command 8byte (5msec) + response 47bytes (15msec) = 20msec (설정 값 Read)

GP write : Command 21bytes (8msec) + ack (2msec) = 10msec (운전화면표시)

ST Read/Write = command (3msec) + response (3ms) = 10msec (DI/DO R/W)

2.2 Monitoring data (PPA 5.02 for WinNT)

아래 그림10은 실제로 RS-485 통신선로에서

Protocol을 주고받는 모습을 Monitoring 한 모습을 보인 것 이다.

첫 번째는 ModBus RTU의 Protocol을 보인 것으로 Graphic panel의 Data를 읽고 쓰는 모습을 Hexa값으로 읽은 것이다. [4]

MODBUS-RTU (GP read / write)

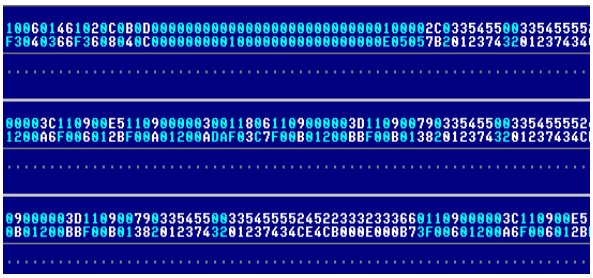

그림10. Code monitor (MODBUS RTU)
Fig10. MODBUS RTU code

아래 그림11은 ModBus ASCII 통신을 ASCII 통신으로 명령을 주고받는 것을 보인 것이다. 무게를 측정하라는 RCWT 명령어와 이에 대한 응답으로 무게를 응답하는 모습이다. MODBUS-ASCII (WI write / read) [5]

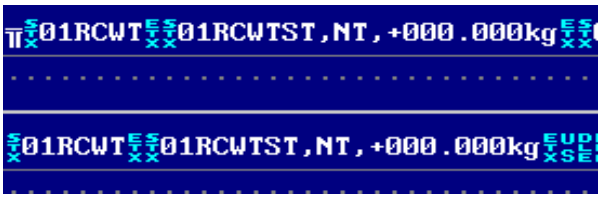

그림11. Code monitor (MODBUS ASCII)
Fig11. MODBUS ASCII code

IV. 결 론

본 연구에서는 RS-485 Balanced 방식의 Serial 통신 Physical layer위에 Multi-Protocol을 구현하는 실험을 하여 보았다. 사용된 Protocol에는 RS-485의 기본 Format 방식과 MODBUS-RTU, MODBUS-ASCII를 동시에 구현하여 하나의 Physical layer위에 서로 다는 Protocol을 가지는 다수의 장비를 연결하는 시험이다. 실시간 처리를 요구하는 ST task 경우는 매 50msec 마다 실행을 행하게 하고, 실시간성이 다소 필요하지 않은 Graphic panel의 read는 400msec, write 동작은 200msec 주기마다 실행하게 하였다. 또한 무게를 측정하는 Task역시 200msec 마다 수행하여 본 연구의 실험으로 사용하였다. 실제로 실험결과 원활한 동작을 이루었고 적용에도 문제가 없었다. 하나의 serial line 상에서 3가지의 protocol을 시간을 분할하여 수행하여 봄으로써 다양한 기기를 동시에 사용하는 성과를 이루었다.

참 고 문 헌

[1] MODBUS-IDA.ORG, "MODBUS over serial line specification", V1.02, 2006

[2] MODBUS-IDA.ORG, "MODBUS application protocol specification", V1.1b, 2006

[3] B&B electronic, "RS-422 and RS-485 protocol", 2006

[4] SEHWA C&M, "Digital weighing Indicator SI-4000" V1.25, 2008

[5] Autonics, "Graphic panel 2480 universal communication", 2005

[6] 박찬원., “로드셀을 이용한 전기식 지시저울의 기술개발 동향과 전망”, 2006

이 재 철 (李在喆)

1995년 2월:방송대 전산학과 학사
1998년 2월:성균관대 정보공학 석사
2006년 3월:목원대 IT 공학과 박사과정
2008년:현 동양 P&A 연구소장
1997년-2005년:아이앤티(주) 연구소장
1985년-1997년:LG전자(주) 선임연구원
관심분야:자동화 시스템, 통신시스템

고 대 식(高大植)

1982년:경희대 전자공학 학사
1987년:경희대 전자공학 석사
1991년:경희대 전자공학 박사
1994년~1995년:UCSB Post-Doc
2001년~2003년:목원대 학술정보처장
1989년~현재:목원대 전자공학과 교수
2007년~현재:목원대 공학교육혁신센터 장
관심분야 : 인터넷 멀티미디어 통신, U-city, SSD