

석유화학 공정에 적용하기 위한 산업프로토콜 개선방법 연구

A Study on the Method of Industrial Protocol Improvement for Application in Petrochemistry Process

심현*, 문지훈**, 이규수***, 오재철****

Hyun Sim, Ji-Hoon Moon, Kyu-Su Lee, Jae-Cheol Oh

Abstract - The research designed model which was integrated Cache technical for improvement of modbus protocol. The mode bus it leads and it does not solve in existing to shorten the tank data computing time. It designe the algorithm which discriminates the shift time of the petrochemistry process tank in following. It is an objective of dissertation.

key word : petrochemistry, industrial protocol, Modbus protocol

1. 서론

국내 산업현장에서 공장시스템들을 효율적이고 유기적으로 연결시키기 위해서 산업네트워크인 산업용 프로토콜을 활용한다. 그중에서도 국내 굴지의 석유화학 공정에서는 Modbus Protocol을 사용하고 있다. 하지만 이에 대한 연구가 국내에서는 매우 열악하여 제대로 된 정보를 찾기 힘들다. 이에 국내 산업환경에서 활용하는 모드버스 프로토콜을 개선하여 이를 현장에 적용하는 연구를 하게되었다. 본 연구는 모드버스 프로토콜 개선을 위해서 캐쉬 기법을 적용하고 이를 설계하였다. 모드버스 캐쉬를 통해서 기존에 해결하지 못했던 탱크 데이터 처리 시간을 단축시키고, 추후에 석유화학 공정 탱크의 교체시기를 판별하는 알고리즘을 설계하고자 하는 것이 본 논문의 목적이다.

2. Modbus Protocol

modbus 프로토콜은 공장에서 사용되어지는 RS-232, RS-485 통신 기기와의 통신에 사용되어진다. Modbus 프로토콜의 역할은 다른 기기들과의 통신을 위한 Interface 역할을 담당한다. 즉, 컴퓨터와 대화를 하기 위해서운영체제를 사용 하듯이 산

업 제어용 기계와 컴퓨터와 통신을 하기 위해서는 modbus 통신을 필수적이다.

Modbus는 매우 간단한 프로토콜 구조를 가지고 있다. 프로토콜 형식으로는 Device Address, Function Code, Data, CRC-16으로 구성되어 있다. Device Address는 각각의 산업용 장비의 Device Address를 나타낸다. 이는 우리가 인터넷을 사용하기 위해서 고유한 IP Address를 가지는 것과 흡사하다. 만약 우리가 고유의 IP Address를 가지고 있지않다면, 인터넷 상에서 다른 사람들과의 통신을 불가능 할 것이다. 따라서 Device Address도 다른 기계와의 통신을 할 수 있는 Interface를 나타낸다. 두 번째로 Function Code는 현재 Device Address로부터 데이터를 읽을 것인지 아니면 쓰기를 할 것인지를 나타낸다. 우리는 인터넷에서 수많은 데이터를 저장 하고나, 볼 수 있다. 이와 같이 modbus 프로토콜에서도 데이터를 읽을 것인지 아니면 저장 할 것인지를 알려 주어야 한다. Data 부분은 실제 데이터가 들어가는 부분이다. 그리고 마지막으로 CRC-16은 에러 검사를 하는루틴이다. 만약 우리가 E-Mail를 보내는데, 송신한 내용이 변경되어 수신 측에 전달되지 않아야 할 것이다. 따라서 송신한 내용과 수신한 내용을 같은지를 검사가 이루어지는게 E-Mail 통신 코드 검사일 것이다. 따라서 modbus 프로토콜에서도 송신한 내용과 수신한 내용이 일치하는지를 검사하는 CRC-16 코드를 사용한다. [Figure 1]는 modbus 프로토콜 형식을 나타낸다.

저자 소개

- * 심 현 : 順天大學 컴퓨터學科 博士課程
- ** 문지훈 : 順天大學 컴퓨터學科 博士課程
- *** 이규수 : 順天大學 컴퓨터學科 碩士課程
- **** 오재철 : 順天大學 컴퓨터學科 教授·工博

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4	8 BITS	8 BITS	n x 8 BITS	16 BITS	T1-T2-T3-T4

[Figure 1] modbus 프로토콜 형식

modbus 통신에서는 RS-232, RS-485 통신을 사용하므로, 통신 프로토콜 데이터는 UART를 통하여 전달된다. UART로 들어오는 데이터에 대하여 modbus 프로토콜 형식으로 만들어 소프트웨어에 전달된다면 좀 더 향상된 처리를 수행할 수 있다. [Figure 2]은 일반적인 modbus 통신을 나타낸다.



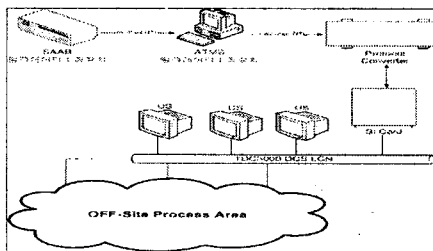
[Figure 2] 일반적 modbus 통신

2. Modbus Cache 설계

우선 석유화학공단에서 적용하여 사용하고 있는 Modbus Protocol을 분석하고 여기에 Cache기법을 적용하여 개선된 Modbus Cache Protocol을 제작할 것이다. 설계순서는 첫째, Modbus Protocol 분석을 하고, 둘째 UART의 데이터 처리 기법, 셋째 탱크 데이터 처리를 위한 RS-232 인터럽트 설계, 넷째 탱크캐쉬 스케줄링 설계를 한다.

2.1 modbus RTU에 의한 Tank 정보 분석

기존 Protocol Converter는 modbus RTU에 의해서 ATMS에서 들어오는 정보를 읽어 들이게 된다. SAAB를 통하여 현재 탱크 데이터 값을 얻은 후, ATMS로 탱크 데이터를 전달하게 된다. [Figure 3]는 탱크 데이터의 전달 과정을 보여준다.



[Figure 3] 탱크 데이터의 전달 과정

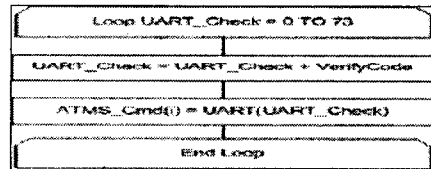
[Figure 3]는 탱크 데이터의 전달 과정을 나타내는데 우선, SAAB에 의해서 탱크 데이터를 ATMS로 보내게 된다. SAAB는 modbus master 형식으로 데이터를 ATMS로 보내게 되면이 장치는 modbus RTU를 이용하여 각 정보를 전달하게 된다. 그림에서의 Protocol Converter는 전달 받은 패킷 데이터

를 받아 modbus 프로토콜로 변환하여 SI 장치로 넘겨주게 된다. SI 장치의 역할은 DCS와의 통신을 담당하게 된다. DCS는 modbus 프로토콜 형식의 데이터 통신만 가능하므로 Protocol Converter에서 RS-232 정보를 modbus 프로토콜 형식으로 변경하게 된다. 이렇게 modbus 프로토콜 형식으로 변경된 정보는 DCS LCN에 의해서 각 US에 탱크 데이터를 전달하게 된다. 탱크 데이터는 73 바이트로 고정된 길이로 구성되어 있다. 그렇지만 실제 사용되는 탱크 데이터는 [Table 1]과 같다.

[Table 1] 실제 사용되는 탱크 데이터

필요탱크데이터	패킷 위치
탱크 이름	ATMS_Cmd[5] ~ ATMS_Cmd[6]
온도	ATMS_Cmd[36] ~ ATMS_Cmd[40]
높이	ATMS_Cmd[30] ~ ATMS_Cmd[34]

[Table 1]은 실제 사용되는 탱크 데이터를 나타낸다. 73 바이트에서 STX, ETX를 제외한 71 바이트에서 실제로 사용되는 바이트는 11 바이트이다. 그렇지만 현재 Protocol Converter는 73바이트를 모두 받아 메모리에 저장한 다음 데이터를 처리하게 된다. 하지만 실제 사용되는 패킷만 메모리에 저장한다면 메모리의 효율적인 사용이 가능하며, 따라서 시스템 성능 향상을 기대할 수 있다. [Figure 4]은 UART의 데이터 처리 기법을 나타낸다.



[Figure 4] UART의 데이터 처리 기법

[Figure 4]은 UART의 데이터 처리 기법을 나타내는데 기존 시스템은 ATMS의 전체 데이터를 메모리에 저장한 반면 변경된 modbus cache 기능을 하는 시스템은 UART에서 받은 데이터를 모두 저장하는 것이 아니라 VerifyCode 즉, 실제 사용되는 패킷 데이터에 한하여 ATMS_Cmd 메모리에 저장을 한다. 따라서 기존 시스템에 비해서 UART로부터 받은 데이터 처리도 향상됨을 알 수 있다.

2.2 탱크 데이터처리를 위한 RS-232인터럽트설계

현재 Protocol Converter의 경우 ATMS에서 데이터를 받을 때 73 바이트의 데이터를 받아 들이게 된다. 또한 에러 처리를 위해서 STX, 실제 데이터, ETX가 들어올 경우 하나의 패킷으로 판단하여 데이터를 처리하게 된다. [Figure 5]는 탱크 데이터의 정상적인 데이터 구성을 나타낸다.

STX	탱크 데이터	ETX
-----	--------	-----

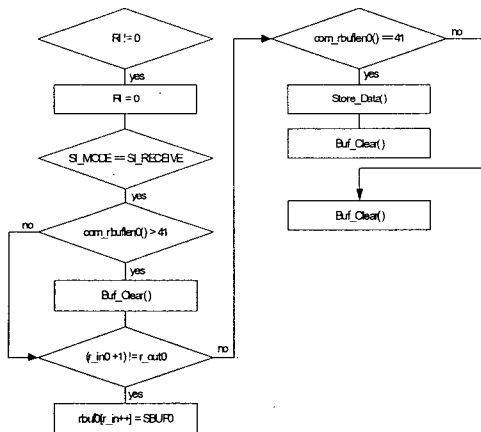
[Figure 5] 탱크 데이터의 정상적인 데이터 구성

[Figure 5]는 탱크 데이터의 정상적인 데이터 구성을 나타내며 STX는 패킷의 시작, ETX는 패킷의 마지막 정보임을 나타낸다. 탱크 데이터는 71 바이트가 들어오게 된다. 하지만 실제 탱크에서 오는 데이터 71 바이트를 모두 사용하지 않는다. 탱크에서 필요로 하는 데이터는 탱크명, 온도, 높이만 필요하다. 따라서 73 바이트를 모두 받지 않은 경우라도 탱크에 필요한 데이터를 모두 받았을 경우라면 DCS로 데이터 전송이 가능할 것이다. 따라서 ATMS에서 전달받은 데이터에서 탱크명, 온도, 높이의 패킷 위치만 안다면 71 바이트의 탱크 데이터를 받지 못하였을 경우에도 정상적인 데이터 처리가 가능하다. [Table 2]는 필요한 탱크 데이터 및 패킷 위치를 나타낸다.

[Table 2] 필요한 탱크 데이터 및 패킷 위치

설명	바이트 수
탱크명	2
온도	4
높이	5

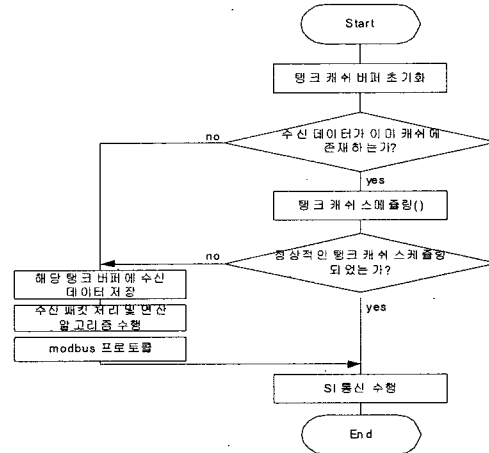
[Table 2]는 필요한 탱크 데이터 및 패킷 위치를 나타내는데, 실제 필요한 데이터를 계산하면 11 바이트만 받으면 된다는 것을 알 수가 있다. 따라서 ATMS에서 오류가 발생하여 전체 데이터를 받지 못하고 40바이트만 받은 경우라도 정상적인 데이터 처리가 가능함을 알 수 있다. [Figure 6]는 RS-232 인터럽트 설계 순서도를 나타낸다.



[Figure 6] RS-232 인터럽트 설계 순서도

[Figure 6]는 RS-232 인터럽트 설계 순서도를 나타내는데 ATMS에서 73 바이트를 받은 것이 아니라 41 바이트만 받아서 데이터를 처리하게 된다. 만약 ATMS에서 42에서 데이터가 끊어 졌다면 이는 문제에서 DCS에 전달이 가능할 것이다. 만약 RI에 데이터가 들어온 경우라면, RI = 0으로 하여 송신된 데이터를 처리하였음을 나타낸다. SI_MODE가 SI_RECEIVE인 경우라면 com_rbufen()을 검사하는데, 이 함수는 송신된 데이터 바이트를 기록하는 함수이다. 41 보다 클 경우라면 원하는 탱크 데이

터가 모두 들어왔음을 나타낸다. (r_in0 + 1) != r_out0은 저장할 데이터가 존재할 경우 rbuf0에 SBUF0을 저장하여 수신된 데이터를 저장한다. com_rbufen이 41가 같은 경우 데이터를 메모리에 저장한 후 Clear 함수를 호출하여 인터럽트 관련 변수를 초기화 한다.



[Figure 7] 탱크캐시 스케줄링

[Figure 7]은 실제 제작될 탱크캐시 스케줄링의 플로우 설계이다. 위의 설계에 따라서 프로토콜을 제작하였으며 이를 이용하여 프로그램상에서 시뮬레이션을 실시하였다.

3. 결론

본 논문에서 설계한 Modbus Cache Protocol은 프로그램상 시뮬레이션에서는 성능이 개선되는 효과를 보였다. 하지만 실제로 산업현장에 적용하는 경우에는 얼마만큼의 효과를 얻을 수 있을지는 좀더 연구를 해 봐야 할 것이다. 그러나 본 연구에서 얻을 수 있는 의미는 석유화학공장에서 주로 사용하는 Modbus Protocol에 대한 연구가 제대로 이루어지지 않아 많은 외화를 낭비하고 있는 환경에서 제대로 된 프로토콜의 분석과 활용방법을 제시하여 국가 경제적 이익에 이바지할 수 있는 기술이 되는 것이라 하겠다.

참고문헌

1. Liang,W.,Zhang,L.,Wang,Z.,Fan,J. Run Hua Yu Mi Feng/Lubrication Engineering (5), pp.74-77, 2006 .
2. Tyco Electronics UK Limited, Rs485&Modbus Protocol Guide, Feb, 2007.
3. Acromag Incorporated, Introduction To Modbus, 2002.
4. MODBUS, OPC force decision Sheble, N. 2005 InTech 52 (10), pp. 110