

소규모 운영체제에서의 스트림데이터 처리기법

김진덕

동의대학교

The Processing Method of Stream Data in the Small-size Operating System

Jin-Deog Kim

Donggeui University

E-mail : jdk@deu.ac.kr

요약

스트림 데이터들은 매우 용량이 크며, 짧은 보고 주기, 비동기적인 보고 시간을 가진다. 이러한 시스템의 질의는 최신의 값을 검색하는 현재질의, 과거 특정시점의 값을 검색하는 스냅샷 질의, 과거부터 현재까지의 값들을 검색하는 슬라이딩 윈도우 질의 등이 있다. 이 논문에서는 소규모 운영체제에서 파일 구조화된 데이터베이스를 이용하여 스트림 데이터들을 효율적으로 관리하는 기법을 제안하고자한다. 그리고 스트림 데이터에 대한 슬라이딩 윈도우 질의를 포함한 다양한 질의를 수용하는 질의 모델을 제안한다. 제안한 기법들을 원자로 안전 운전원 모듈에 적용하여 그 유용성을 검증하고자한다. 운전원 모듈(COM)은 QNX 운영체제를 기반으로 하며, 파일 데이터베이스는 QNX COM의 적은 저장장치, 낮은 계산 능력을 감안하여 델타버전과 버퍼링 등의 방법을 도입한다.

ABSTRACT

Stream data need a efficient data management with high reliability and real time processing. The characteristics of these data are a large volume, a short report interval and asynchronous report time. The typical queries of these systems consist of the current query to search the latest signal value, the snapshot query to search the signal value of a past time, the historical query to search the signal value of a past time to current. This paper proposes the efficient method to manage the above signals by using a file structured database in QNX operating systems. The query model to accommodate various query for stream data is proposed. The proposed methods are applied to reactive protection system to verify their usefulness. The COM(Cabinet Operator Module) based on the QNX employs file database that adopts a delta version and a buffering method for the resource limit of a small storage and a low computing power.

키워드

스트림 데이터, 원자로 안전 보호 데이터, 슬라이드 윈도우 질의

1. 서론

데이터들의 지속적인 유지와 다양한 분석으로 기초로 하는 기존의 DBMS와는 달리, 최근 빈번한 변경을 요구하는 실시간 데이터의 효율적인 관리와 새로운 질의 처리 기법이 요구되는 응용이 부각되고 있다. 원자로 보호 계통 운전원 모듈 또한 그 대표적인 예이다. 기 개발된 원자로 보호 계통은 핵중기 공급계통의 상태를 감시하는 기능을 제공하며, 계측채널의 신호를 수집하고 설정치를 비교하는 각종 주변 기기 및 제어스위치로 구

성된 캐비닛운전원모듈(COM), 개시회로 등으로 구성되어 있다. 이러한 원자로 데이터는 실시간으로 끊임없이 새로운 값이 도달하는 데이터 스트림(Data Stream)이다.

이러한 스트림 데이터를 데이터 전체를 저장하는 것은 공간 효율성이 매우 떨어지며, 질의 응답 속도 또한 보장할 수 없다. 그리고 스트림 데이터는 소규모 특화된 운영체제에서 운용되는 경우가 많다. 따라서 높은 공간 효율성을 보장하는 스트림 데이터 관리 기법과 슬라이딩 윈도우 질의를

원활하게 수용하는 기법이 필요하다. 그러나 지금까지의 원자로 보호계통 운전원 모듈은 각 외부 장치로 부터의 현재 신호를 점검하여 동작상태를 검증하는 방식이었다.

따라서 본 논문에서는 원자로 보호계통의 구성 모듈로부터 입력되는 신호 정보들을 소규모 QNX[1] 운영체제에서 실시간으로 데이터베이스화하고, 현재와 과거 신호 및 이력 신호 정보를 검색할 수 있는 질의 모델을 설계하고자 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 스트림 데이터와 원자로 보호 계통에 대한 관련 연구를 살펴보고, 제 3장에서는 이 논문에서 설계하고자 하는 DB 시스템을 기술하며, 제 4장에서는 슬라이딩 윈도우 기반 질의 모델 등에 대해 자세히 설명하며, 제 5장에서 결론을 내린다.

II. 관련 연구

데이터 스트림 처리에 관한 다양한 연구가 진행되어 왔다[4, 5, 7]. Telegraph[4]는 관계기반 질의어를 이용하여 슬라이딩 윈도우 질의를 제공한다. Niagara[5]는 스트림 데이터에 대해 가변적인 질의를 효과적으로 처리하기 위한 적응적 질의 엔진을 제공한다.

그리고 Niagara와 Telegraph에서는 질의 색인을 활용하여 연속질의 성능을 높이고자 하였다. 이 색인은 질의 대상 속성의 간격을 보유하고 있으며, 센서 노드로부터 이벤트가 발생할 때 이진 탐색을 기반으로 질의의 결과 값을 도출한다.

관련 연구 [7]에서는 최근 활성화되고 있는 RFID 미들웨어 시스템에서 연속질을 원활하게 처리하기 위한 색인을 제시하였으며, 색인 구성에 질의를 기반으로 하였다.

이상과 같이 데이터 스트림에 대한 다양한 질의어 모델이 제시되었지만, 지금까지 QNX와 같은 소규모 특수 목적 운영체제하의 데이터스트림에 관한 효율적인 관리 및 슬라이딩 윈도우 질의에 관한 연구는 연구된 바 없다.

기존의 원자로 보호계통 시스템은 핵증기공급 계통의 상태를 감시하여 정해진 안전계통 설정치에 도달하면 원자로를 정지시키는 기능을 제공하며, 계측채널의 신호를 수집하고 설정치를 비교하는 비교논리프로세서(BP), 다중 채널의 트립신호를 Voting하는 동시논리프로세서(CP), 시험 및 유지보수를 관장하는 자동시험 및 연계프로세서(ATIP), 운전원화면 및 제어스위치로 구성된 캐비닛운전원모듈(COM), 개시회로 등으로 구성되어 있다. 또한 이들 각 구성장치들은 Profibus-FMS 프로토콜[2,3]을 이용하여 통신한다. 운전원 모듈의 운영체제로는 QNX[1]를 사용하고 있다. Profibus는 실시간 처리 요구조건을 만족한다.

특히 국내의 자동차 회사에서는 QNX Neutrino를 자사의 텔레매틱스를 위한 운영체제로 채택하고 있다. QNX는 1980년 IBM-PC용의

상용 실시간운영체제로 시작하여 20년 동안 PC용 상용제어시스템의 분야에서 발전해오고 있다.

III. 실시간 신호 관리 기법

3.1 COM DB 시스템 구성도

COM 모듈은 BP, CP, ATIP와 같은 주변 기기로부터 시험신호를 받아들여 각 장치의 상태를 조작자에게 보고하는 역할을 수행한다. 또한 조작자가 필요한 정보를 주변기기에 보내는 역할 또한 수행한다. 그러나 지금까지 시스템은 끊임없이 변경되며 누적되는 신호에 대한 체계적인 관리가 없어 현재 질의만 가능하고, 스냅샷 질의 및 슬라이딩 윈도우 질의는 불가능하였다.

이에 본 논문에서는 그림 1과 같이 주 운전원 모듈인 COM과 외부 장치 간에 스트림 데이터베이스 시스템을 두어 실시간으로 변경되는 신호들을 체계적으로 관리하며, 질의에 응답할 수 있는 환경을 제공한다.

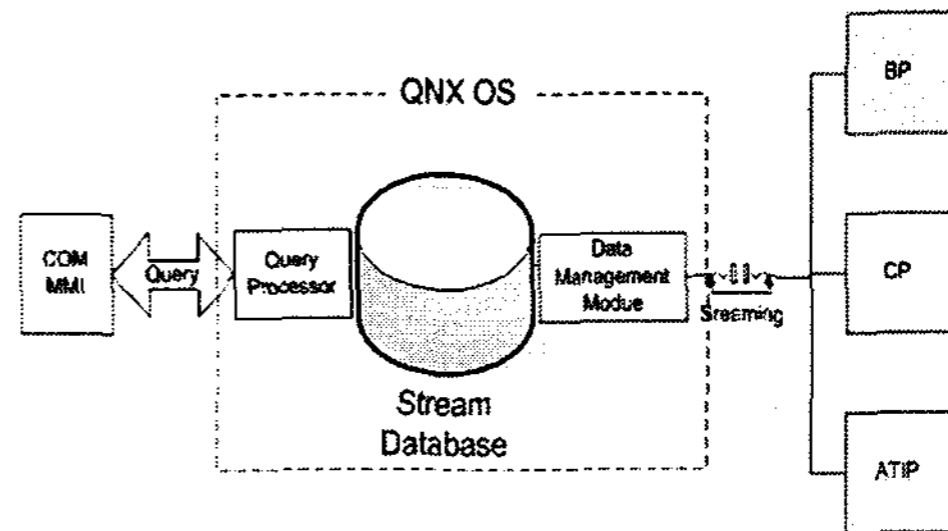


그림 1. COM DB 시스템 구성도

스트림 데이터베이스 시스템은 소규모 모바일 QNX 운영체제 하에서 구축되며, 각 장치로부터 입력되는 신호는 약 10,465개이며, 출력되는 신호는 200개이다. 스트림 데이터베이스가 유지 및 관리하는 신호와 관리 요건 등은 다음과 같다.

- 신호의 대부분은 Historical Data로 유지되어야 하며, 일부 신호는 특정 값이 감지되는 즉시 조작자에게 통보되어야 한다.
- COM 장비에서 이들 데이터를 수집할 때 얼마나 자주 수집하여야 하는지, 즉 각 신호별 또는 장치별 주기가 설정하며, Event Report Window에 출력될 신호는 별도로 처리한다.

스트림 데이터베이스 시스템은 QNX의 리소스 한계(적은 메모리, 낮은 연산능력)와 질의에 적합한 파일 구조 데이터베이스를 설계하고자 한다.

그림 1의 스트림 데이터는 주변 장치들로부터 Profibus 카드를 통해 수행된다[2,3]. 각 Profibus에는 여러 개의 OD(Object Dictionary)가 있으며, 각 OD는 여러 개의 Address와 bit로 세분된다. 각 OD는 230바이트의 데이터영역이 있으며, Configuration 파일에 따라 각 Slot이 2, 4, 8 바이트로 구성할 수 있다.

이 논문에서는 원자로 보호 계통의 신호가 2바이트 정수, 2바이트 실수, 1비트 Bool값으로 구성되므로 각 OD는 2byte 단위의 Address로 구성하였으며, 저장 용량의 압축을 위해 각 비트를 최대한 활용하여 16개의 Bool 신호를 하나의 Slot에 저장할 수 있도록 하였다[6].

3.2 파일 DB 구축

진술한 Profibus 카드의 Configuration을 토대로 획득한 각 외부 장치의 신호를 파일 구조 기반 DB에 저장하고 관리한다. 원자로 보호계통 신호는 개수가 매우 많고, 빈번히 변하는 특성을 가지며, 때로는 특정 신호 값이 거의 변하지 않고, 주기에 따라 변하는 특성 또한 갖고 있다. 따라서 이 논문에서는 델타버전 기법을 이용한다. 주기적으로 전체 신호가 포함된 Master File를 구축하고, 일정 주기별로 이전 데이터와의 차이 값만을 수용하는 Delta 파일을 만들어 QNX의 작은 저장 구조에 적합하도록 하였다.

각 외부 장치로부터 입력되는 신호 값이 이전 값과 같은 경우 무시되며, 값이 변화하는 값을 저장한다. 델타버전은 전체 신호 중 동일한 신호에 대한 값이 두 번째 변경될 때 새로운 버전을 생성한다. 그림 2는 이를 도식화한 것이다.

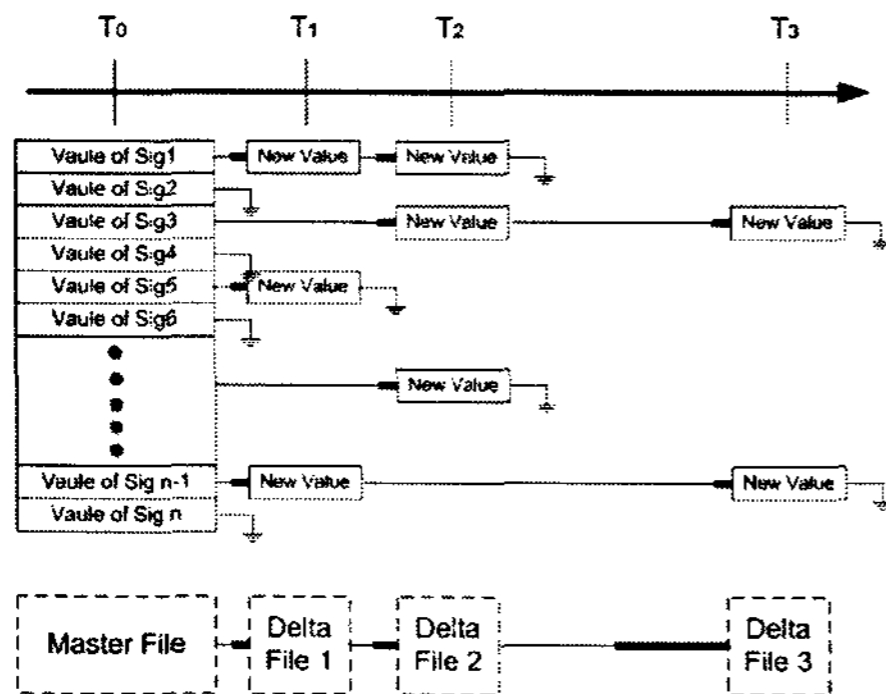


그림 2. 델타 파일 생성

그림 2에서 초기에 Master 파일이 생성되며, Delta 파일 1과 2는 3개의 변경된 값만을 갖고, Delta 파일 3은 3개의 변경된 값만을 갖는다. 각 파일은 생성된 시간을 유효 신호의 개수를 포함한다. 또한 신호가 도착하는 시간이 매우 가변적이고, 값의 변화 또한 비주기적으로 변화하므로 델타파일의 생성 시간 또한 일정하지 않다.

이 논문에서는 저장 공간의 효율성 및 질의 속도의 향상을 위해 각 파일은 논리적으로만 존재하며, Master 파일과 여러 개의 Delta 파일이 하나의 물리적인 파일 내에 포함되도록 하였다.

또한 관리의 용이함을 도모하기 위해 하나의 물리적인 파일의 크기가 일정 규모가 될 때까지 델타파일을 생성한다. 따라서 신호의 값이 거의 변하지 않거나, 신호의 전달 주기가 길 경우 하나의 물리적인 파일은 매우 긴 시간을 포함한다.

IV. 질의 모델

일반적으로 스트림데이터에 대한 질의는 일정 기간 동안 또는 일정 횟수에 걸쳐 연속적으로 수행되며, 새로운 데이터가 도착할 때 마다 즉시 반영하는 특징이 있다. 이러한 질의를 연속질의(continuous query)라 한다. 원자로 안전 보호 계통의 데이터들은 신호 값들을 현재부터 과거 특정 신호까지 연속적으로 모니터링하는 경우가 많다.

질의의 경계는 슬라이딩 윈도우를 사용하여 지정한다. 이 논문에서는 윈도우의 사이즈의 범위를 지정하는 방법, 검색하고자하는 신호의 개수, 질의생존 기간, 질의 연산 방법 등에 따라 다양한 질의를 수행할 수 있다.

그림 3에서 윈도우 사이즈(T_{start} , T_{end})를 기준으로 질의는 크게 다음과 같은 5가지로 분류할 수 있음을 보여준다. 이 때 T_{now} 는 현재시점으로서 시간이 흐름에 따라 T_{now} 도 변하게 된다.

- 1) 과거의 특정 시점의 신호값을 검색하는 Snapshot 질의. 이 때 T_{start} 와 T_{end} 는 같음.
 - 2) 과거의 특정시점 T_{start} 부터 T_{end} 까지의 신호값을 검색하는 Past History 질의
 - 3) 과거의 고정된 특정시점 T_{start} 부터 T_{now} 까지의 신호값을 검색하는 Current History with Closed Past 질의
 - 4) 과거의 변화하는 시점 T_{start} 로부터 T_{now} 까지의 신호값을 검색하는 Current History with Open Past 질의
 - 5) T_{now} 시점의 신호값을 검색하는 Current 질의
- 진술한 윈도우사이즈는 시간으로 정의한 것이지만, (T_{end} , #of data)와 같이 최종시간과 데이터의 개수로 정의할 수도 있다. 예를 들어 Snapshot 질의는 (T_{end} , 1)로 표현된다.

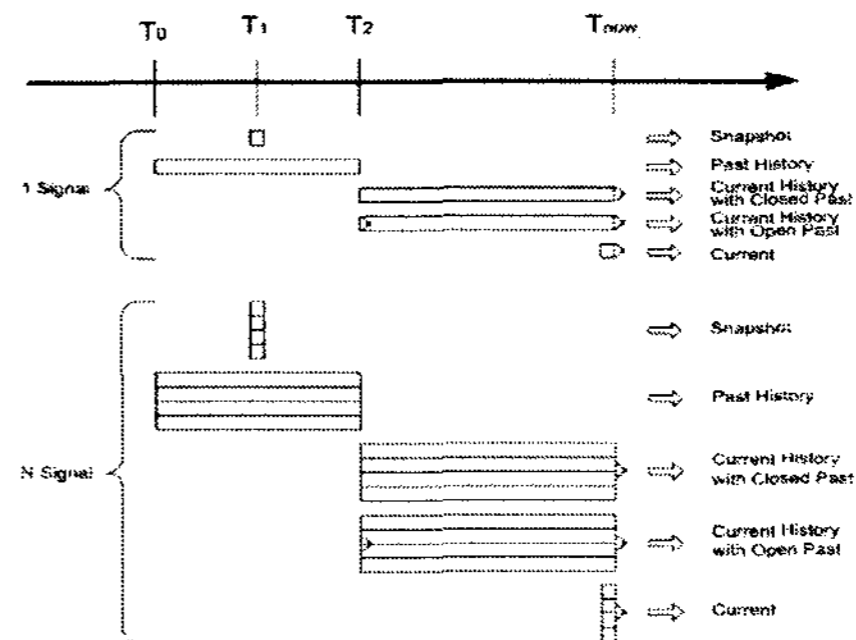


그림 3. 질의 모델

그리고 한 개의 신호값을 검색하느냐, 여러 개의 신호값을 검색하느냐도 구분할 수 있다. 질의 생존 기간은 연속질의가 수행되는 시간을 의미하는 것으로서 알람이벤트는 1회, PhAB에 의한 검색은 윈도우 생존기간, TextOut은 사용자의 지정

시간 등으로 구분된다.

질의 연산 방법은 기본 값과 MIN, MAX, AVG와 같은 집합 함수를 적용할 수 있으며, 신호값의 여과(filtering)을 위해 Predicate를 사용할 수 있다.

그림 4와 5는 상기의 질의 모델 중 Current 슬라이드 윈도우 질의와 Past History 질의의 결과를 각각 보여 주는 것이다. 그림 4는 PhAB를 통해 Current 슬라이드 윈도우 질의의 결과를 보여 주는 것으로서 질의 결과가 실시간 변화되는 신호값을 즉시에 반영한다. 질의 결과 임계치를 벗어난 신호를 다른 색으로 표현하고 있다. 실시간 신호값을 보다 빨리 처리하기 위해 이 논문에서는 버퍼를 두어 최근 값을 메모리에서 관리하고 있다.

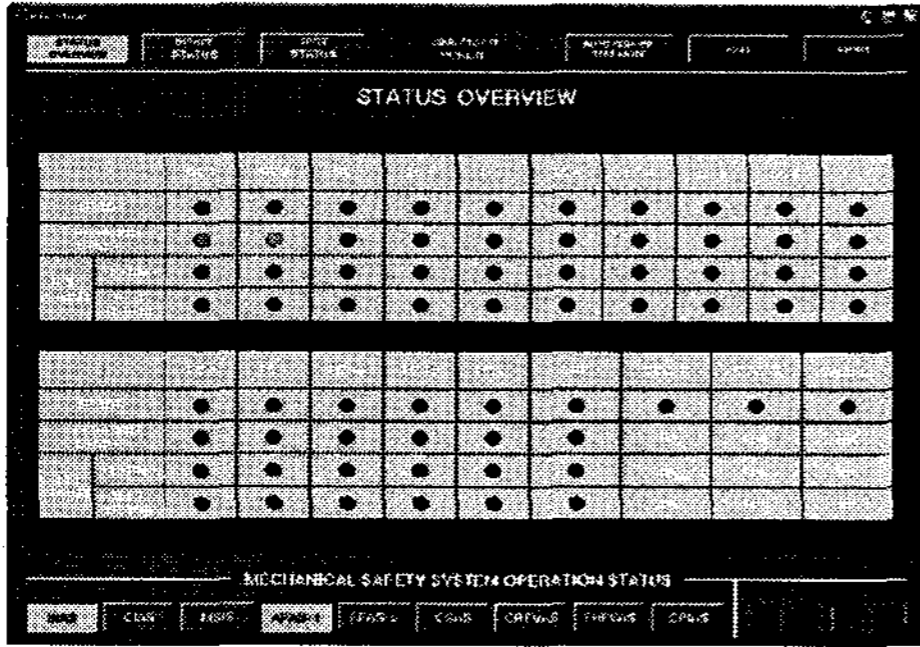


그림 4. Current 질의 결과

그림 5는 Past History의 결과로서 과거의 일정 기간 동안의 여러 개의 신호값들을 SAMBA를 통해 웹페이지로 보여주고 있다. 결과로서 버전별로 축약된 정보를 다시 복원시키며, 각 버전의 저장 시간을 보여준다.

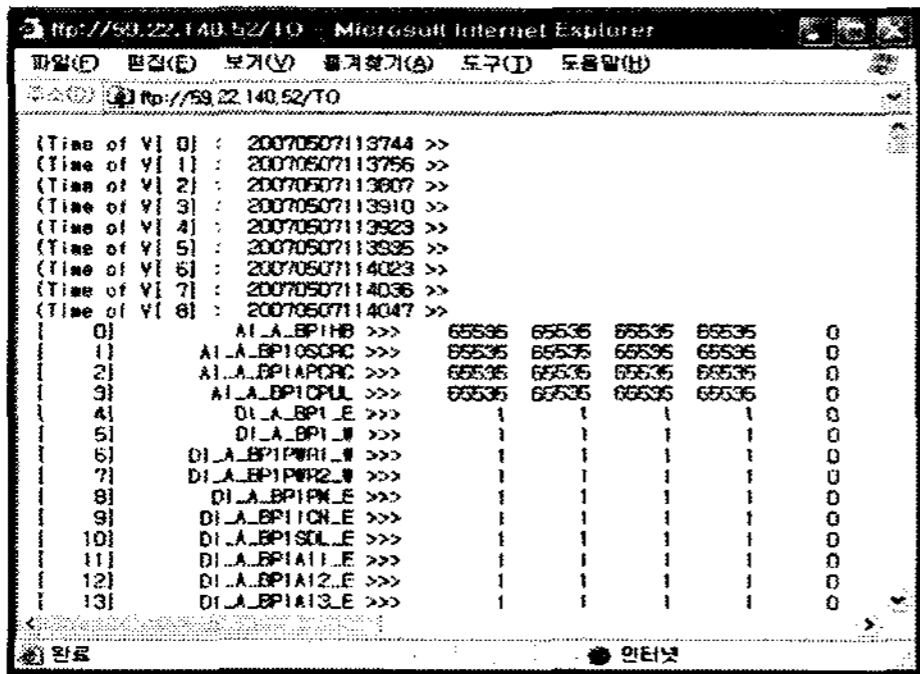


그림 5. Past History 질의 결과

이상과 같은 슬라이딩 윈도우 질의를 조합하여 다음과 같은 질의 모델을 생성할 수 있다.

```
SELECT      Value | Aggregate_Function
FROM        { SignalID }+
WHERE       Predicate
```

WINDOW T_{start}, T_{end}
TIME Interval

- Aggregate_Funtion := max|min|avg
- Predicate := <, <=, =, >=, >, AND, OR
- Tstart := T₁(closed), T₁+(open)
- Tend := T₁(closed), T_{now}(open)
- Interval := 1, PhAB, TEXT_OUT

V. 결 론

이 논문에서는 소규모 운영체제에서 파일 구조화된 데이터베이스를 이용한 스트림 데이터를 효율적으로 관리하는 기법과 스트림 데이터에 적용 가능한 슬라이딩 윈도우 질의를 설계하였다. 제안한 기법들을 원자로 안전 운전원 모듈에 적용하여 그 유용성을 검증하였다.

특히, 안전 운전원 모듈의 QNX 운영체제가 적은 저장장치와 낮은 계산 능력을 고려하여 델타 버전과 버퍼링 기법을 도입하여 해결하였다.

이와 같은 연구 결과는 QNX를 널리 사용하는 국내의 텔레메틱스/스마트홈 산업에서도 기술의 활용도가 높을 것으로 판단된다. 또한 다양한 센서데이터를 관리하고 질의를 제공하는 소규모 임베디드 시스템에 효과적으로 적용할 수 있을 것으로 판단한다.

참고문헌

- [1] QNX Software Systems, <http://www.qnx.com/>
- [2] PROFIBUS International Open Solutions for the World of Automation, <http://www.profibus.com/>
- [3] "PROFIBUS Specification," Normative Parts of PROFIBUS - FMS, -DP, -PA according to the European Standard EN 50 170 Volume 2, Edition 1.0, March, 1998
- [4] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V Raman, F Reiss, M Shah, "TelegraphCQ: Continuous Dataflow Processing for an Uncertain World"
- [5] J Chen, D. J. Dewitt, F Tian, Y Wang, "NiagaraCQ : A Scalable Continuous Query system for Internet Databases"
- [6] 김진덕, 진교홍, 이성진, 정해원, "빈번한 변경이 요구되는 데이터의 효율적인 실시간 처리 기법", 한국해양정보통신학회 추계종합학술대회, 10권 1호, 2006
- [7] 박재관, 홍봉희, 반재훈, "RFID 스트리밍 데이터 처리를 위한 연속 질의의 변환 기법", 정보처리학회논문지 D, 14-D권, pp. 273-284, 2007