Proceedings of the 11th
KACC, October 1996

FFT 분석을 위한 SCPI(Standard Commands for Programmable Instrument) 명령어

SCPI(Standard Commands for Programmable Instrument) Commands for FFT Analysis

노승환

공주대학교 정보통신공학과(Tel, FAX: 0416-50-8623; E-mail: rosh@ice.kongju.ac.kr)

Abstracts SCPI(Standard Commands for Programmable Instrument) is a standard command sets designed for controlling various types of instruments. In order to control FFT(Fast Fourier Trnsform) analyzing device using SCPI it is required to support sweep measurement function. We defined SCPI command set for FFT analysis and developed parser of defined command set using lex(Lexical Analyzer Generator) and yacc(Yet Another Compiler Compiler). After developing FFT analyzing test was performed with that parser. Up to audible signal frequency the result of FFT analysis was accurate and that result was agree with that of conventional FFT analyzer. As a result it is proved that various types of instruments including sweep measurement instrument can be controlled with appropriate SCPI command sets. Also when developing new instruments the method used in this experiment will contribute to reducing the time required to develop the SCPI parser and increasing reliability.

Keywords Instrument, SCPI, Parser, lex, yacc

1. 서론

1970년대 후반에 이르러 계측시스템의 수동조작으로부터 컴퓨터가 모든 계측기기를 제어할 수 있도록 시스템화되고 표준화된 계측기기가 미래 계측산업의 핵심분야로 떠오르고 있다.[1][2][3]

이렇게 해서 탄생한 CAT(Computer Aided Test) 기술은 컴퓨터를 각종 계측기와 연결하여 자동으로 계측기기들을 제어할 수 있게 하였으며, IEEE488.1 과 IEEE488.2는 컴퓨터와 계측기기간 접합버스로 현재 가장 보편적으로 사용되고 있다.[4][5][6]

SCPI(Standard Commands for Programmable Instruments)는 IEEE 488.2의 범위를 능가하는 새로운 계측명령언어이며 매우 다양한 계측기기들을 수용할 수 있다. SCPI는 프로그래밍 관점에서 보면 같은 부류의 계측기기간에 또는 같은 기능을 갖는 계측기기간에 일관성을 추구한다.

SCPI 계측기기는 명령어와 파라미터를 수신하는데 있어서 매우 융통성이 많으므로 프로그램을 구성하기가 용이하다. 또한 계측기기는 제어기로 데이타 또는 상태정보를 되돌려 전송하며 SCPI에서는 이러한 응답 형식이 잘 정의 되어있어 계측기기의 데이타와 상태정보를 이해하기가 매우 용이하다.

본 연구에서는 UNIX 운영체제에서 제공되는 응용 프로그램 가운데 어휘분석생성기인 lex(Lexical Analyzer Generator)와 파서(parser) 발생기인 yacc(Yet Another Compiler Compiler)를 이용하여 FFT 분석에 필요한 SCPI 명령어 파서를 개발하고자 한다.

본 논문의 구성은 2장에서는 SCPI에 대해서 설명하였고 3장에서는 FFT 분석에 필요한 SCPI 명령어의 파서에 대하여 살펴본다. 4장에서는 오디오 신호의 FFT 분석에 필요한 SCPI 명령어들을 작성하고 이들을 앞에서 구현된 SCPI 파서의 입력으로 이용하여 FFT 분석실험을 실시한다. 마지막으로 5장에서는 4장에서 얻어진 결과에 대하여 고찰하고 추후 수행되어야 할 연구에 대하여 살펴본 후 결론을 맺는다.

2. SCPI 개요[1][3][4]

SCPI 명령어는 명령어 테이블로 구성되어 있으며 이 테이블은 KEYWORD, PARAMETER FORM 및 COMMENTS등 세부분으로 나뉘어진다. KEYWORD는 명령어의 이름을 나타내고 있으며 SCPI 명령어가 계층(hierarchy)구조를 갖기 때문에 한개 이상의 keyword로 구성되어 있다.

PARAMETER FORM 부분은 명령어 내에 포함 되어야 하는 수 또는 그 외의 파라미터 등을 나타낸다. 명령어는 SCPI에서 정의되는 파라미터 형태 또는 문자형태 이거나 이들 두개를 조합한 형태이어야 한다.

SCPI 계측기기는 구조화된 명령어 체계 때문에 <compound command program header>와 <compound query program header>를 파스(parse)해야 한다. 또한 모든 SCPI 계측기기는 IEEE 488.2에서 기본적인 명령어로 규정된 일반명령어(common command)를 모두 수용해야 한다. SCPI program header는 명령어를 구분하기 위해 사용되는 keyword이며 common-command header와 instrument-control header등 별개의 두개 형태로 구성되었다.

3. SCPI 파서 및 어휘분석기

3.1 FFT 분석을 위한 SCPI 명령어 정의

본 연구에서 사용된 FFT 분석을 위한 SCPI 분석 모델은 그림 1과 같다.

그림 1에서 INPut 블럭은 센서의 입력 포트의 특성을 제어하며 입력되는 오디오 신호의 감쇄정도를 조절한다.

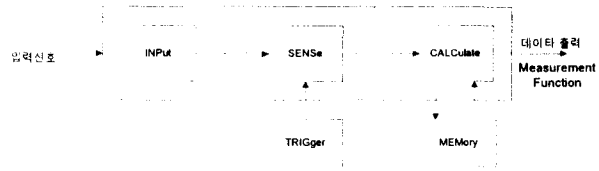

그림 1. FFT 분석 모델
Fig. 1. Model for FFT Analysis

SENSe 블럭에서는 입력되는 신호를 샘플하여 데이타를 수집하는 기능을 담당한다. 이 때 사용되는 명령어는 수집되는 데이타의 샘플율과 point수를 결정하며 이 들간에는 서로 상관관계를 갖는다. 또한 샘플된 데이타를 출력시키는 기능도 있으며 이 명령어에 따라 CALCulate 블럭에서 변환되기 전의 데이타를 볼 수 있다.

CALCulate 블럭에서는 수집된 데이타를 처리하는 기능을 담당한다. 이 기능에는 잡음을 제거하는 저주파 통과 필터링 과정, FFT 연산을 하는 과정 등을 포함한다.

그림 2에서는 FFT 분석을 위해 본 연구에서 구성한 SCPI 명령어 세트를 보여주고 있다.

```
*RST                          ; common command
*CLS                          ; common command
:SYSTem
      :VERSion?               ; return <NR2>              ex) 1990.0
                              ; query only
:INPut, 센서의 입력 포트의 특성을 제어한다.
      :ATTenuation  <numeric_value>
        :AUTO  <Boolean>|ONCE
        :STATe <Boolean>
:SENSe
      :DATA?                  ; query only

      :SWEep
        :POINts  <numeric_value>    ; TINTerval * POINts = TIME의
        :TIME<numeric_value>        ; 관계를 갖는다.
        :TINTerval  <numeric_value> ; point 수에는 제약을 받지 않는다.

:CALCulate
      :DATA?                  ; query only

      :FILTer
        [:GATE]
          :FREQuency
            :STATe  <Boolean>
            [:TYPE]:BPASs|NOTCh
            :STARt  <numeric_value>
            :STOP   <numeric_value>
            :SPAN   <numeric_value>
            :CENTer <numeric_value>
            :POINts  <numeric_value>
              :AUTO<Boolean>|ONCE ; OFF를 선택하면 sense된
                                  ; 데이터 수만큼 지정이 된다.

            :WINDow HAMMing       ; default
      :TRANsform
      :FREQuency
          :STATe     <Boolean>
        [:TYPE]      LPASs|BPASs
      :STARt         <numeric_value>
      :STOP          <numeric_value>
      :SPAN          <numeric_value>
      :CENTer        <numeric_value>
      :POINts        <numeric_value>
        AUTO         <Boolean>|ONCE
```

그림 2. FFT 분석을 위한 SCPI 명령어 세트
Fig. 2 Command Set for FFT Analysis

그림 2에서 *RST와 *CLS는 일반 명령어(common command)로써 모든 계측기기에서 기본적으로 수용해야 하는 명령어이다. *RST 명령어는 계측기기를 측정 또는 다른 기기동작을 위해 명령어를 기다리는 초기상태로 되돌리는 역할을 한다. *CLS는 디바이스내의 모든 상태 데이타 구조(status data structure)를 초기화한다.

가. SYSTem 서브시스템

SYSTem 서브시스템은 계측기능과는 직접적으로 관련이 없는 TIME, SECurity 및 VERSion? 등의 명령어가 포함된다.

나. INPut 서브시스템

INPut 서브시스템은 센서 입력포트의 특성을 제어한다. INPut의 기능은 입력되는 신호를 SENSe 블럭에서 데이타로 변환되기 전에 조정(conditioning)하는 기능을 담당한다. INPut 블럭의 기능은 필터링(filtering), 바이어싱(biasing), 주파수 변환 및 감쇄기능 등을 포함한다.

다. SENSe 서브시스템

SENSe 서브시스템은 여러 부분으로 분할된다. 각 부분은 디바이스에 의존하는 세팅을 제어하며 신호 자체는 다루지 않으며 데이타를 수집할 때 포인트 수, 간격 및 기간등을 결정하고 수집된 데이타를 접근할 수 있게 한다.

라. CALCulate 서브시스템

CALCulate 서브시스템은 논리적으로 SENSe 서브시스템과 버스 또는 디스플레이로 데이타를 출력시키는 중간에 위치한다. SENSe 서브시스템에 의해 수집된 데이타는 지정된 데로 CALCulate에 의해 변환되어 선택된 출력으로 전송한다. 결과적으로 새로운 데이타를 수집하면 CALCulate 서브시스템을 트리거 한다. 또한 CALCulate 서브시스템은 직접 명령어에 의해 수집된 데이타를 변환할 수도 있다. 그러므로 CALCulate의 구성을 달리해서 SENSe에 의해 새로운 데이타를 수집하지 않고 기존에 수집된 데이타를 이용해서 다른 결과를 계산해 낼 수도 있다.

3.2 파서의 구성

그림 3에서 FFT를 위한 SCPI 파싱트리(parsing tree)를 보여주고 있다.

우선 명령어가 주어지면 일반명령어(common command)인가 계측기기 명령어(instrument command)인가를 판단한다. 일반명령어인 경우에는 *RST와 *CLS가 있으며 이 명령에 해당하는 동작을 수행한다. 계측기기 명령어인 경우 반드시 수용해야 하는 명령어(mandated)인가 선택적인 명령어(optional)인가를 구분한다.

모든 기기에서 수용해야 하는 명령어로는 SYSTem 서브시스템이 있으며, 이에 해당하는 동작을 수행한 후에 새로운 명령어를 기다린다. 선택적 명령어인 경우 본 연구에서 수용하는 서브시스템은 INPut, SENSe 및 CALCulate등이 있다. 각 명령에서는 한 번에 한 개만의 서브시스템을 수용할 수 있으며 INPut 서브시스템인 경우 기기의 입력포트의 특성을 제어하는데 관련되는 동작을 수행한다.

다음 명령어가 SENSe 서브시스템인 경우 다시 DATA? 또는 SWEep인 경우로 구분된다. 일반적으로 SWEep 서브시스템 수행된 후에 DATA? 명령이 수행되며, SWEep 명령에서는 입력되는 신호를 시간영역에서 샘플하여 데이타로 변환하는데 필요한 파라미터 값들을 결정한다.

다음 CALCulate 서브시스템에서는 샘플된 데이타에 대해 필터 동작과 FFT 연산을 수행한다. CALCulate 서브시스템에는 DATA?, FILTer 및 TRANsform등이 있을 수 있으며 먼저 FILTer 서브시스템에서는 FIR(Finite Impulse Response) 필터에 필요한 파라미터 값들을 결정하는 동작을 수행하며 TRANsform

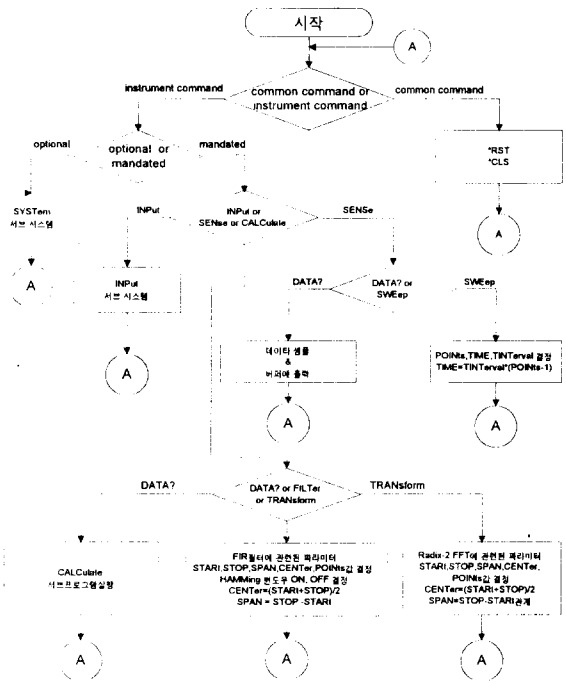

그림 3. FFT 명령어의 파싱 트리
Fig. 3 Parsing Tree for FFT Command

서브시스템에서는 데이타에 대해서 FFT 연산을 수행하는데 필요한 파라미터 값들을 결정하는 동작을 수행한다. 먼저 FILTer 서브시스템에서는 위의 각 파라미터 값들을 결정한 후에 윈도우를 실행할 것인가를 결정한다. 필터에 관한 파라미터 결정을 마치면 데이타를 시간영역에서 주파수 영역으로 변환시키기 위해 TRANsform 서브시스템에서 파라미터 값들을 결정한다. 이 값들은 따로 정해 주지 않을 경우에는 기본적으로 FILTer에서 사용한 값들과 동일한 값을 사용한다.

4. 실험 및 결과

4.1 실험

위의 과정을 통해 FFT 명령어 파서가 생성되었다. 생성된 파서는 C 언어로 되어 있으므로 이를 다시 컴파일하여 실행 파일을 얻을 수 있다.

생성된 파서의 동작을 확인하기 위해 SCPI 명령어로 구성된 명령어를 작성하였으며, 이 명령어를 생성된 파서의 입력으로 사용하여 FFT 분석 실험을 하였다. 구성된 명령어 세트의 한 예가 다음과 같다.

```
sense:sweep:points 1024
sense:sweep:tinterval 1e-4
sense:sweep:time?
sense:data?
calculate:filter:gate:frequency:state on
calculate:filter:gate:frequency:start 0
calculate:filter:gate:frequency:stop 2000
calculate:filter:gate:frequency:span?
calculate:filter:gate:frequency:center?
calculate:filter:gate:frequency:window hamming
calculate:transform:frequency:state on
calculate:data?
```

본 연구에서는 위와 같은 명령어 세트를 사용하여 샘플 point 수를 1024로 했을 때 입력신호를 달리해 가며 실험을 수행하여 데이타를 출력하였다. 필터는 모두 1kHz 저주파 통과 필터를 사용하였으며, 사용된 입력신호는 프로그램상에서 인위적으로 각 주파수에 해당하는 신호값들을 발생시켜 입력신호로 사용하였다.

4.2 결과

앞에서 설명한 명령어 세트를 이용하여 실험을 한 결과 다음과 같은 결과를 얻을 수 있다. 실험결과는 저주파 필터를 통과하지 않았을 때의 시간 영역과 주파수 영역의 신호를 각각 나타내고, 다시 같은 입력에 대해서 저주파 필터를 통과한 후에 시간영역과 주파수 영역의 신호를 각각 나타내었다.

실험에서는 입력신호가 200Hz, 500Hz, 1kHz와 4kHz의 혼합된 신호이며, point 수를 1024로 하였다. 센스된 데이타는 필터를 통과하지 않은 상태에서 더 많은 주파수 성분이 혼합된 결과 그림 4에서와 같이 시간영역에서 복잡함을 알 수 있다. 다음 그림 5에서는 filter를 OFF 한 상태에서 FFT를 수행한 결과이며, 저주파 통과 필터를 통과하지 않은 결과로 입력신호에서와 같이 모든 주파수 성분이 나타남을 알 수 있다. 그림 6은 같은 센스 데이타에 대해 필터를 통과한 후에 시간영역에서 데이타를 보여주고 있으며 고주파 성분의 신호가 제어되었음을 알 수 있다. 이를 다시 FFT 변환시킨 결과가 그림 7에 나타나 있다. 그림 7에서는 고주파 성분 4kHz의 신호는 완벽하게 제거되었음을 알 수 있으나, 본 실험에서 사용된 필터의 대역이 1kHz이므로 1kHz의 신호는 모두 제거되니 않은 상태로 남아 있고, 나머지 두 개의 신호는 필터의 영향을 전혀 받지 않았음을 알 수 있다.

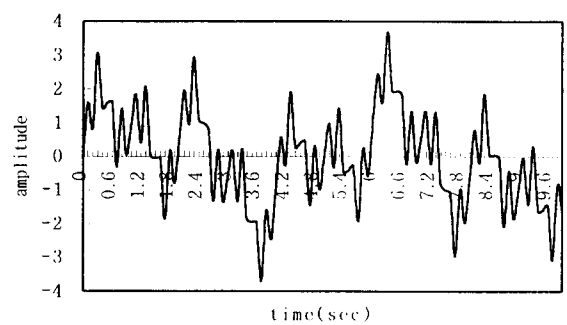

그림 4. 센스된 데이터(2)
Fig. 4 Sensed Data(2)

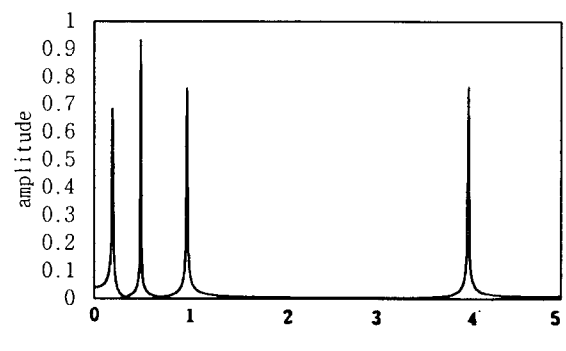

그림 5. 필터를 통과하지 않은 FFT(2)
Fig. 5 Filtered FFT(2)

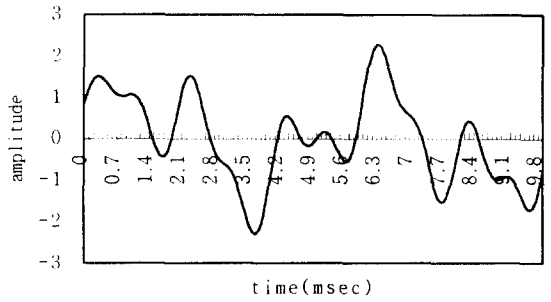

그림 6. 필터를 통과한 데이타(2)
Fig. 6 Filtered Data(2)

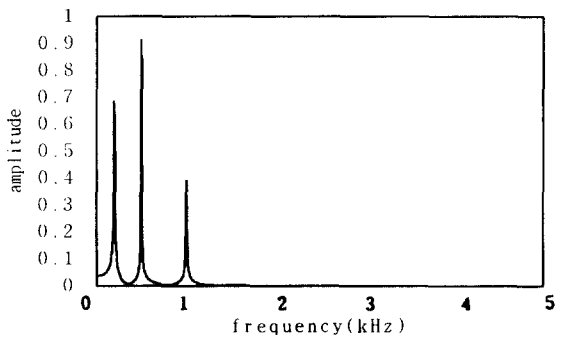

그림 7. 필터를 통과한 FFT(2)
Fig. 7 Filtered FFT(2)

5. 결론 및 고찰

현재 계측기기들이 통합화, 시스템화됨에 따라 컴퓨터를 이용하여 기기들을 제어하는 추세에 있고 이를 위하여 명령어 체계를 확립하는 과정이 진행중에 있다. 또한 새로운 계측기기들이 계속적으로 개발됨에 따라서 이러한 새로운 기기들을 제어하는 새로운 명령어들이 필요하게 되었고, 이를 위하여 본 연구에서는 FFT 분석을 위한 SCPI 명령어에 대하여 연구하였다.

FFT 분석과정은 단순히 입력된 신호의 양만을 측정하는 것에 비하면 매우 복잡한 과정을 거쳐야 한다. 그러므로 분석 과정을 이해하고 적절하게 제어한다는 것은 매우 어려운 일이다. 그러나 표준화된 명령어를 사용한다면 이러한 복잡한 과정이 체계적으로 수행될 수 있다.

이러한 명령어는 새로운 계측기기가 개발됨에 따라 동시에 명령어가 정의되고 이들 명령어를 해석하여 주어진 값에 분석과정을 수행해야 하는 설정도 동시에 개발되어야 한다. 그러므로 본 연구에서는 FFT 분석을 위한 SCPI 명령어 세트를 정의한 후, 설정를 개발하였다. 설정는 어휘분석기 생성기인 lex와 파서발생기인 yacc를 이용하고 여기에 사용자 프로그램을 추가한 C 원시 프로그램 형태로 발생되었다.

개발된 설정의 동작을 확인하기 위하여 다양한 입력신호에 대해 명령어에 따라 분석과정을 수행하였다. 분석한 결과 개발된 파서가 정상적으로 동작함을 알 수 있었다. 또한 본 연구에서는 개발된 명령어는 저주파 통과 필터만을 사용하도록 되어 있으나 이 외 다른 종류의 필터를 포함하고, 다양한 필터 알고리즘을 수용할 수 있다며 더욱 강력한 명령어 체계를 이룰 수 있을 것이다.

추후 연구되어야 할 사항은 앞에서 언급된 내용과 같이 기존의 명령어 및 파라미터에 더하여 더욱 다양한 알고리즘을 수용할 수 있는 명령어를 개발하는데 있으며, 또한 SCPI 명령어는 하드웨어의 종류에 의존하지 않는 표준화된 언어이므로 다양한 하드웨어 환경에 적합하게 동작하도록 실험이 수행되어야 할 것으로 본다.

참고문헌

[1] David A. Haworth, "An Architecture for Modular Instruments," *Tektronix*, 1988.

[2] John Novellina, "Now Bus Architecture Extend VXIbus," *Electronic Design*, Apr, 27, 1989.

[3] Steven H Leibson, "IEEE 488.2 Products are just now appearing," *EDN*, pp. 91-99, April 25, 1991.

[4] Narbert Laengrich, "Assuring Measurement Accuracy in IEEE 488 Based ATE System," *Test & Measurement World*, May, 1985.

[5] Hohn Purvis III, "The Personal Computer as Instrument Controller," *Test & Measurement World*, pp.42-51, Feb. 1986

[6] "*IEEE 488.2 - 1992, IEEE Standard codes, Formats, Protocols and Commom Command.*" IEEE, 1992.

[7] "*SCPI 1992, Volume 1 : Syntax & Style,*" SCPI Consortium, 1992.

[8] "*SCPI 1992, Volume 2 : Command Reference,*" SCPI Consortium, 1992.

[9] "*SCPI 1992, Volume 3 : Data Interchange Format,*" SCPI Consortium, 1992.