

이중 계측기 자동 제어를 위한 SCPI기반 명령어 시퀀스 생성에 관한 연구

유홍연, 임권섭, 윤광수, 전은경, 오승훈

한국전자통신연구원

{keister, kaide.lim, yks0604, ekjeon, osh93}@etri.re.kr

A Study on SCPI-based Commands Sequence Generation for Automatic Control of Heterogeneous Instruments

Hong Yeon Yu, Kwon Seop Lim, Kwang Su Yun, Eun Kyoung Jeon, Seung Hun Oh

Electronics and Telecommunications Research Institute

요 약

본 논문은 제조(전자 부품/모듈/패키징) 공정으로 생산된 제품의 특성화, 유효성 검사 및 적합성 테스트를 위해 작업자가 자동화된 공정 시스템 제어 프로그램을 신속하게 재현할 수 있는 SCPI 기반 제어 명령어 라이브러리 생성 방법과 이를 활용한 이중 계측기 제어 명령어 시퀀스 생성 방법을 제안한다.

I. 서 론

제조업의 지능형 스마트 팩토리는 IIoT(Industry IoT) 및 AI 기술을 접목하여 다양한 제조 설비를 통해 제품의 조립, 가공 등의 생산 공정부터 다양한 계측기를 통해 생산 제품의 성능평가를 수행하는 측정 공정까지 모든 과정을 사용자의 개입을 최소화할 수 있도록 자동화하고 이를 디지털로 처리할 수 있는 시스템을 구현함으로써 실현된다.

측정 공정 자동화 시스템의 구성은 공정 조건에 따른 제어 명령어를 다수의 계측기에 전송하고 그 결과 데이터를 수집 할 수 있는 제어용 컴퓨터와 GPIB, RS-232, VXIbus, LAN 등 물리적 통신 인터페이스를 가지는 하나 이상의 계측기, 측정 대상 DUT(device under test) 유닛 및 제어용 컴퓨터에서 실행되는 원격제어 응용 프로그램으로 구성된다.

일반적으로 원격제어 응용 프로그램은 SCPI(Standard Commands for Programmable Instrumentation)[1] 기반 계측기 제어 명령어를 C, C++ 등 컴퓨터 프로그래밍 언어에 연결하여 개발하거나 LabVIEW, MATLAB, Agilent VEE와 같은 상용 응용 프로그램을 이용하여 측정 공정 자동화 시스템을 구축하고 있다.

이러한 원격제어 응용 프로그램 기반 측정 공정 자동화 시스템은 반복적 측정, 장기간 측정, 데이터 수집/분석 등 오퍼레이터가 다수의 계측기 앞에서 수작업으로 여러 가지 작업을 수행할 경우 적은 비용으로 효과적인 측정 공정이 가능하도록 도와준다.

그러나 SCPI 명령어 기반 원격제어 응용 프로그램 개발은 다양한 측정 조건 설정의 복잡성과 미리 정의된 측정 절차에 따른 명령어 시퀀스를 구성할 수 있는 전문적인 프로그래밍 능력과 개발 시간을 요구하고 있다. 반면, 상용 원격제어 응용 프로그램은 직관적인 사용자 인터페이스 등 다양한 기능을 제공하지만, 프로그램을 활용할 수 있는 전문적인 지식이 필요하고 이를 교육하는데 많은 시간이 소요된다. 또한, 전문적인 프로그램 활용 능력을 보유한 오퍼레이터가 공정에서 제외되고, 측정 조건이 일부 변경될 경우 원격제어 프로그램을 재설계하여 구축하는데 많은 시간과 비용이 필요하며, 이러한 SCPI 명령어 기반 측정 공정을 단순화할 수 있는

솔루션이 요구된다.

본 논문에서는 상기 문제점을 해결하고자 계측기 조작 설명서를 통해 측정 공정을 수행할 수 있는 능력만을 가진 현장 오퍼레이터가 간편하게 SCPI 명령어 기반 측정 공정 제어 절차를 설계할 수 있는 도구를 제안한다. 그리고 기존[2] 논문에서 제시한 측정 공정 자동화 시스템에 제조사로부터 제공된 계측기 프로그램 매뉴얼의 제어 명령어를 라이브러리화하여 시스템에 활용할 수 있는 방법을 제시한다.

II. 본론

가. SCPI 명령어 개요

표준 SCPI는 IEEE-488.2 표준 코드 및 형식을 기반으로 물리적 통신 계층을 통해 컴퓨터와 계측기 간의 제어 메시지를 ASCII 문자열로 전송할 수 있는 공통 인터페이스 언어로 다양한 물리적 통신 인터페이스를 지원한다.

표준 SCPI 명령어는 계층화된 구조로 명령어를 정의하고 있으며, 하나 이상의 명령어 keyword, 명령어의 상태를 설정하는 parameter 및 사용자 명령어 argument로 제어 명령어를 구성할 수 있으며, 그림 1과 같이 구성된다.

그림 1에서 루트 계층의 "MEASure"는 1레벨 명령 키워드이고, "VOLTage", "CURRent"는 2레벨 키워드이며 "DC?", "AC?"는 3레벨 키워드를 나타낸다.

```
:MEASure
:VOLTage
:DC?
:AC?
:CURRent
:DC?
:AC?
...
```

그림 1. SCPI 구문 구조[1]

예를 들어 계측기로부터 현재 DC Voltage 측정 값을 질의할 경우 'MEASure:VOLTage:DC?'와 같이 명령어를 구성하여 제어 컴퓨터에서 계측기로 메시지를 전송한다.

나. 이중 계측기 명령어 라이브러리 생성

본 논문에서는 'RST', 'CLS' 등과 같이 계측기 시스템의 현재 상태를 질의하거나 변경 요청하는 공통 명령어(common command), 특정 신호 발생 등을 위한 설정 명령어(set operation command), 현재 측정된 계측값의 질의 명령어(query operation command), 제조사 고유 기능으로(데이터 저장 경로 설정 등) 제공되는 시스템 설정 명령어(instrument configuration command)로 분류하여 계측기 명령어를 그림 2와 같이 디지털 파일 형태로 라이브러리화 한다.

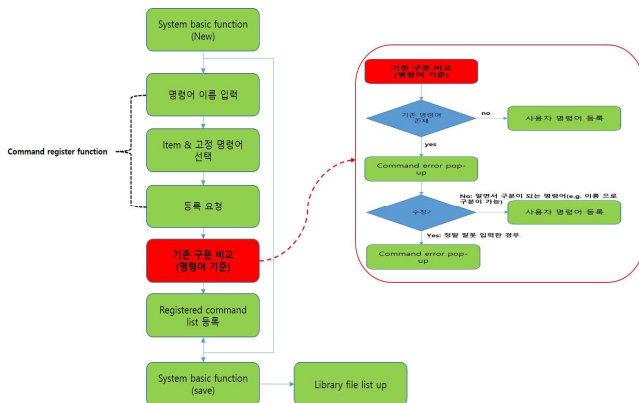


그림 2. SCPI 분류 기준에 따른 제조사 계측기 제어 명령어 라이브러리 생성 절차

그리고 각 제어 명령어에서 사용자가 설정할 수 있는 parameter와 argument의 종류를 그림 3과 같이 분류하여 라이브러리화함으로써 공정 절차에 따른 자동 제어 명령어 시퀀스 생성시 사용자 입력 값 설정을 단순화 해주고 사용자가 생성한 명령어의 오류를 자동으로 검증할 수 있도록 도와준다.

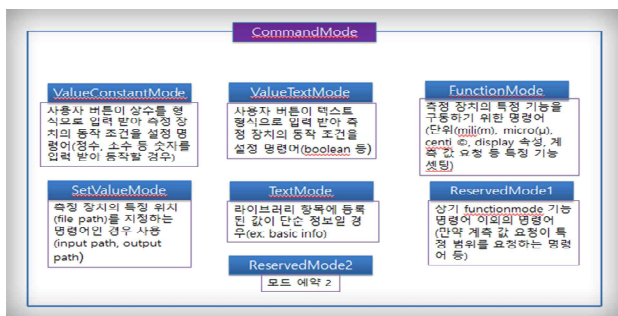


그림 3. 제어 명령어 라이브러리의 사용자 설정 모드 분류 항목

다. 라이브러리 기반 측정 공정 자동화 시스템 구현

그림 4는 다양한 계측기의 제어 명령어를 라이브러리로 생성하여 기존[2] 측정 공정 자동화 시스템 적용하기 위한 구성을 나타낸 것으로, 제어 명령어 라이브러리는 CIA 단말에 계측기(MEU)의 물리적 통신 인터페이스와 함께 동시 등록되고, 제어 컴퓨터 역할을 수행하는 UMD에서 오퍼레이터가 제어 시퀀스를 생성하고 측정된 데이터를 실시간으로 모니터링 및 수집 기능을 제공하도록 구성된다.

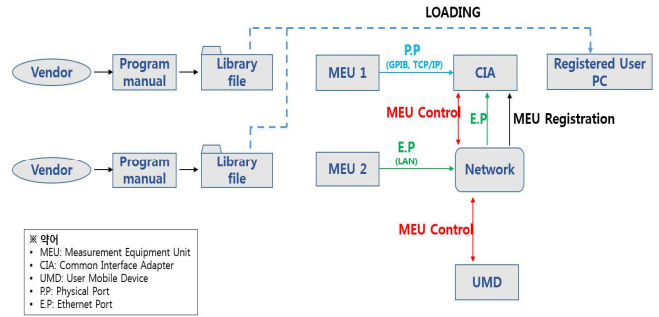


그림 4. 라이브러리 기반 측정 공정 자동화 시스템 구성

오퍼레이터의 공정 절차에 따른 제어 명령어 시퀀스 생성은 그림 5와 같이 계측기 라이브러리에서 설정하고자 하는 명령어를 순차적으로 선택하여 배열하고, 반복적인 계측이 필요한 경우 반복 조건(초기값, 증감값)을 설정함으로써 자동으로 제어 명령어 시퀀스를 생성한다.

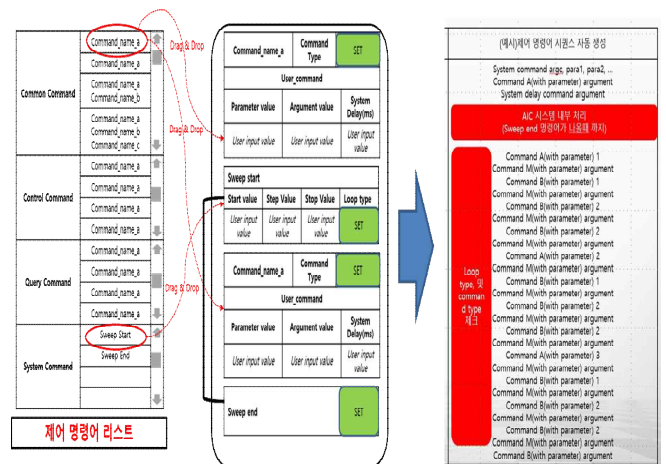


그림 5. 제어 명령어 시퀀스 생성 예시

III. 결론

본 논문에서는 다품종 전자 부품/모듈/패키지 측정 공정에 활용할 수 있도록 다양한 제조사 계측기의 SCPI 제어 명령어를 5종류로 분류하여 디지털 파일 형태로 라이브러리화하여 측정 공정 자동화 시스템을 구현함으로써 전문 오퍼레이터의 개입 없이 신속/정확하게 측정 공정을 수행할 수 있었다. 또한, 기존 수작업으로 인한 사용자 오류를 줄이고 측정 공정을 관리하는데 소요되는 많은 시간을 절약할 수 있다.

ACKNOWLEDGMENT

본 연구는 한국전자통신연구원 연구운영지원사업의 일환으로 수행되었음.(21ZK1100, 호남권 지역산업 기반 ICT 융합 기술 고도화 지원사업)

참고 문헌

- [1] Wikipedia, "Standard Commands for Programmable Instruments," 2020. (https://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instruments)
- [2] 유홍연, 박지원, 손동훈, 임권섭, "IIoT기반 측정공정 자동화 시스템 연구개발," CICS 2020, pp. 143-145.