

CANopen 표준 기반 멀티 엔코더 관리 시스템의 구현

안 효 성* · 김 태 현**
* 서울시립대학교 기계정보공학과

Implementation of Multi-encoder Management System based on CANopen Protocol

Hyosung Ahn* and Taehyoun Kim**

* Dept. of Mechanical and Information Engineering, Univ. of Seoul

(Received October 4, 2016 ; Revised January 9, 2017 ; Accepted January 19, 2017)

Key Words: CANopen, Smart Encoder(스마트 엔코더), Encoder Management System(엔코더 관리 시스템), Open-source Software(오픈 소스 소프트웨어)

초록: 최근 컴퓨터 및 통신기술의 적극적인 결합에 따른 생산 시스템의 자동화 및 지능화가 급속하게 진행됨에 따라, 자동화 시스템의 핵심 요소 중 하나인 스마트 센서를 내장한 필드 장치들의 수도 급격하게 증가하고 있으며 이들을 통합 관리할 필요성도 높아지고 있다. 본 논문에서는 엔코더의 위치 데이터 센싱 기능과 CANopen 프로토콜을 결합한 스마트 엔코더 구조를 제안하고, 복수 개의 엔코더를 CAN 네트워크를 통해 동시에 관리, 모니터링할 수 있는 시스템 구조를 설계, 구현하였다. 구현한 시스템의 성능과 기능적 동작은 상용 엔코더와의 비교 실험과 CANopen 호환성 테스트를 이용하여 검증하였다.

Abstract: Recently, with the adoption of modern computing and communication technologies, manufacturing systems have become more autonomous and intelligent. Thus, as the number of field devices with smart sensors also increase, the need for an integrated management of such devices becomes essential. This paper proposes a smart encoder architecture that integrates the position sensing function with CANopen connectivity. In addition, an integrated system is proposed to simultaneously control and monitor multiple encoders over the Controller Area Network (CAN) fieldbus network. We evaluated the performance and functionalities of the proposed system by comparative experiments with commercial CANopen smart encoders using a CANopen conformance test.

- 기호설명 -

0x(숫자) : 16 진수 표기법

1. 서론

최근 전통적인 기계 산업 분야에 컴퓨터 및 통신기술을 적극적으로 결합하는 움직임이 활발해짐에 따라 생산 시스템의 자동화 및 지능화 또한 급속하게 진행되고 있다. 이러한 추세에 따라 다양한 응용 조건 및 응용 환경에 쉽게 대응이 가능한 스마트 센서에 대한 수요도 크게 증가하고

있다.⁽¹⁾ 스마트 센서는 일반적으로 데이터 센싱 및 내부 처리, 센싱한 정보의 외부 전송에 필요한 센서 및 회로 인터페이스를 하나의 패키지로 구성한 형태로, 기존의 센서 기능 외에도 장치 설정, 상태 관리, 에러 알림 등 사용자에게 추가적인 기능을 제공한다.⁽²⁾ 특히 높은 수준의 지능화가 진행됨에 따라 이러한 스마트 센서를 내장한 필드 장치들의 수도 급격하게 증가하고 있으며 이들을 통합 관리할 필요성도 높아지고 있다.

통합 관리 대상이 되는 스마트 센서의 수가 늘어날수록 장치간 연결 방식에서 기존의 점대점(point-to-point) 방식은 다양한 제조사 간의 호환성 보장, 확장성 등 유지보수 측면에서 한계를

† Corresponding Author, thkim@uos.ac.kr
© 2017 The Korean Society of Mechanical Engineers

가지게 되었으며, 이러한 문제를 해결하기 위해 버스 형태의 공유 전송매체를 이용한 표준 필드 버스의 사용이 일반화되기 시작하였다. CAN (Controller Area Network)⁽³⁾은 대표적인 산업용 필드버스 표준 기술 중 하나로, CAN 네트워크를 기반으로 하는 상위 응용을 정의하는 개방형 표준인 CANopen⁽⁴⁾ 프로토콜 등을 통하여 상위 수준의 장치간 호환성을 추가로 보장하고 있다.

엔코더는 물체의 직선 또는 회전 변위를 측정하는 센서로, 다양한 형태의 로봇과 CNC 제어기 등 정밀한 위치 기반 제어가 필요한 산업 응용에서 필수적으로 사용된다.⁽⁵⁻⁷⁾ 특히 다관절 로봇의 경우 다축 간 동기화를 위해 각 축의 현재 위치를 측정하는 엔코더 센싱 정보를 동시에 취득하고 네트워크를 통해 제어기로 전송하는 기능의 활용도가 꾸준히 높아질 것으로 예상된다. CANopen 표준에서는 엔코더에 대한 장치 프로파일인 CiA 406 표준을 통해 엔코더의 주요 기능 및 동작을 정의한다.

본 연구에서는 엔코더의 센싱 기능과 CANopen 프로토콜을 결합한 스마트 엔코더 구조를 제안하고, 플랫폼을 구성하는 멀티 축에 부착된 각각의 엔코더를 CAN 네트워크를 통해 동시에 관리, 모니터링할 수 있는 시스템 구조를 설계, 구현하였다. CANopen 표준 중 스마트 엔코더 운용에 필수적인 프로파일인 CiA 301⁽⁸⁾과 CiA 406⁽⁹⁾에 대해서는 오픈 소스 기반 CANopen 프로토콜 스택인 CanFestival⁽¹⁰⁾을 확장, 구현하였으며 실제 산업체에서 상용 개발 중인 엔코더 플랫폼 상에서 엔코더 센싱 로직과 통합하였다. 또한, 복수 개의 CANopen 호환 엔코더를 동시에 관리할 수 있는 시스템의 동작 실증을 위해 CANopen 마스터 장치와 외부 PC 상에서 동작하는 모니터링 소프트웨어, 이들 간의 통신을 위한 자체 프로토콜 등의 설계, 구현 내용을 제시한다. 최종적으로 구현된 시스템의 성능과 기능적 동작은 2종의 상용 엔코더와의 비교 실험을 통해 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 배경에 대해 간략하게 소개하고, 3장에서는 CANopen 지원 스마트 엔코더와 관리 및 모니터링 시스템에 대한 설계 및 구현 내용을 기술한다. 4장에서는 CANopen 지원 엔코더에 대하여 테스트 시나리오, 테스트 베드 상에서의 동작, CANopen Conformance Test Tool 수행 결과를 통

해 개발한 내용을 검증하고, 마지막으로 5장에서 결론을 맺는다.

2. 연구 배경

CANopen 표준은 CAN 네트워크로 연결된 장치간의 상위 수준 동작 규약 및 호환성 보장을 위한 내용을 정의하고 있으며, Fig. 1과 같은 계층 구조 형태의 참조 모델을 가진다. CANopen에서 정의하는 세부 응용 계층은 프로파일(profile) 개념을 사용하여 정의되는데, 기능적 특성에 따라 크게 통신 프로파일과 장치 프로파일로 구분된다. CiA 301 표준에 정의된 통신 프로파일은 CANopen 장치의 공통적인 통신 인터페이스로, 제공하는 서비스에 따라 Table 1에 제시한 다양한 통신 객체를 정의하고 있다. 장치 프로파일은 각 장치에서 제공하는 응용 서비스 및 세부 동작

Table 1 CANopen communication objects

| 항목 | 기능 |
|------------------------------|---|
| NMT (Network management) | <ul style="list-style-type: none"> CANopen 상태머신 관리 Node 상태 확인(Heartbeat, Node guarding) |
| SYNC (Synchronization) | <ul style="list-style-type: none"> PDO 데이터 동기화 메커니즘 객체 사전 설정에 따라 SYNC producer / consumer로 나누어짐 |
| PDO (Process data object) | <ul style="list-style-type: none"> 실시간 프로세스 데이터 전송 요청/응답 과정을 거치지 않아 전송 오버헤드가 작음 |
| SDO (Service data object) | <ul style="list-style-type: none"> 객체 사전 읽기 및 쓰기 요청/응답 방식으로 전송이 일어나 PDO에 비해 오버헤드가 큼 |
| EMCY (Emergency) | <ul style="list-style-type: none"> 에러 정보 전달 |

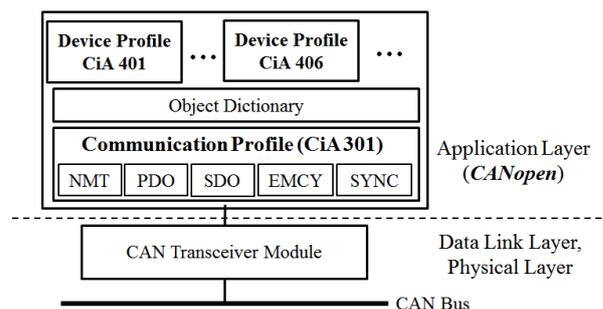


Fig. 1 CANopen profile reference model

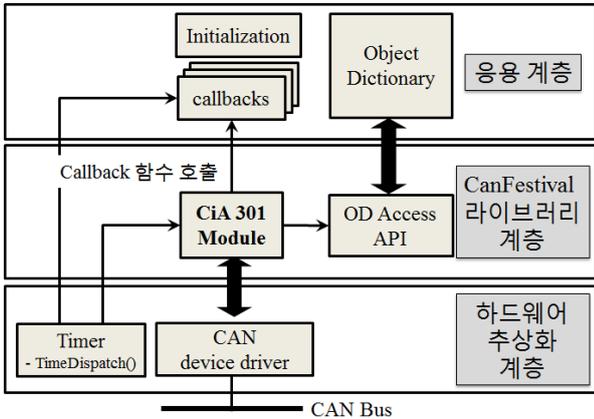


Fig. 2 CanFestival software architecture

에 대해 CiA 4XX 표준으로 정의한다. 엔코더의 경우 CiA 406 표준에서 센서에서 읽은 위치 데이터를 소프트웨어적으로 가공하여 출력하는 기능과 소프트웨어적 작업 공간 설정, CAM 등 엔코더의 고급 기능에 대해 정의한다.

CANopen 표준을 장치에 실장하기 위해서는 CANopen 통신 프로토콜 스택의 구현이 필수적이다. 최근에는 잘 정의된 오픈 소스 솔루션을 활용하여 개발 주기를 단축하는 시도들이 활발하게 있으며, 대표적인 오픈 소스 형태의 CANopen 프로토콜 스택 구현으로 CanFestival이 있다. CanFestival은 ANSI C 표준을 준수하여 작성되었고 하드웨어 의존적인 부분들을 모듈 형태로 분리한 하드웨어 추상화 계층(이하 HAL: Hardware Abstraction Layer)을 제공함에 따라 다양한 컨트롤러 플랫폼으로 이식이 용이한 장점을 가진다.

CanFestival은 Fig. 2와 같이 응용 계층, CanFestival 라이브러리 계층, HAL 등 세 부분으로 구성된다. 응용 계층은 장치 초기화 및 CANopen 스택의 주요 변수 초기화 함수와 응용에 정의된 각종 콜백 함수, 장치의 속성을 나타내는 객체사전(OD: Object Dictionary) 등을 정의한다. CanFestival 라이브러리 계층은 CiA 301 통신 프로파일 처리에 관련된 API 모듈과 응용에서 구성한 객체사전의 엔트리들을 접근하는 API 모듈로 구성된다. 하드웨어 의존적인 부분을 담당하는 HAL에는 주기적 데이터 전송을 위한 타이머 처리부와 CAN 장치의 데이터 송수신을 위한 장치 관리자(device driver) 함수들이 존재한다.

본 연구진은 이미 선행 연구⁽¹¹⁾에서 CanFestival 구현을 확장하여 엔코더를 위한 장치 프로파일인 CiA 406 모듈을 구현하고 더미 데이터를 이용한

Table 2 H/W and HAL configuration for CANopen encoder

| 항 목 | 주요 내용 |
|-----------|---|
| CPU clock | 80 MHz(40 MIPS) |
| CAN 모듈 설정 | 전송속도 지원: 20 kbps, 50 kbps, 250 kbps, 500 kbps, 1Mbps |
| Timer 설정 | Timer 4를 사용하도록 setTimer() 구현 |
| Interrupt | CAN Rx 인터럽트 처리: canDispatch() 호출 Timer 인터럽트: TimeDispatch() 호출 |
| DMA 설정 | DMA 채널 0: CAN Tx 버퍼로 설정 DMA 채널 1: CAN Rx 버퍼로 설정 |

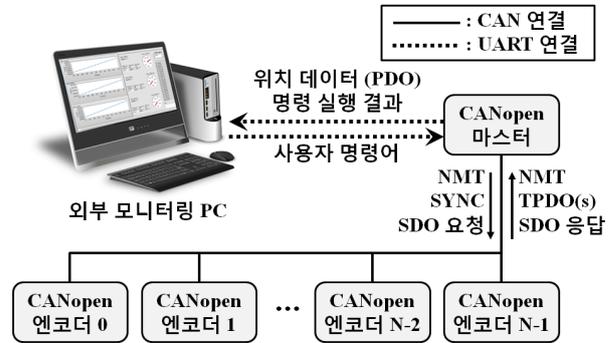


Fig. 3 Architecture of multi-encoder management system

기본 기능 확인을 하였으나, 실제 엔코더 장치에 실장한 상태에서의 위치 센싱 모듈과의 통합 검증은 이루어지지 않았다.

3. CANopen 표준 멀티 엔코더 관리 시스템의 설계 및 구현

3장에서는 CANopen 기반 멀티 엔코더 관리 시스템의 구조와 시스템을 구성하는 CANopen 엔코더, CANopen 마스터 장치와 외부 관리/모니터링 소프트웨어의 설계 및 구현 과정을 기술한다.

3.1 CANopen 엔코더 통합 관리 시스템 구조

본 연구에서 제안하는 CANopen 표준에 기반한 멀티 엔코더 동시 관리 시스템의 구조는 Fig. 3과 같이 외부 모니터링 PC, CANopen 마스터, 복수개의 CANopen 엔코더로 구성된다.

CANopen 마스터는 각 통신 객체를 통해 CANopen 엔코더의 설정 사항을 관리하며, 주기적으로 센싱되는 위치 데이터 수집 작업을 동기화한다. 외부 PC에서 동작하는 엔코더 관리 프로그램은 CANopen 마스터와 통신하여 동기화된 위

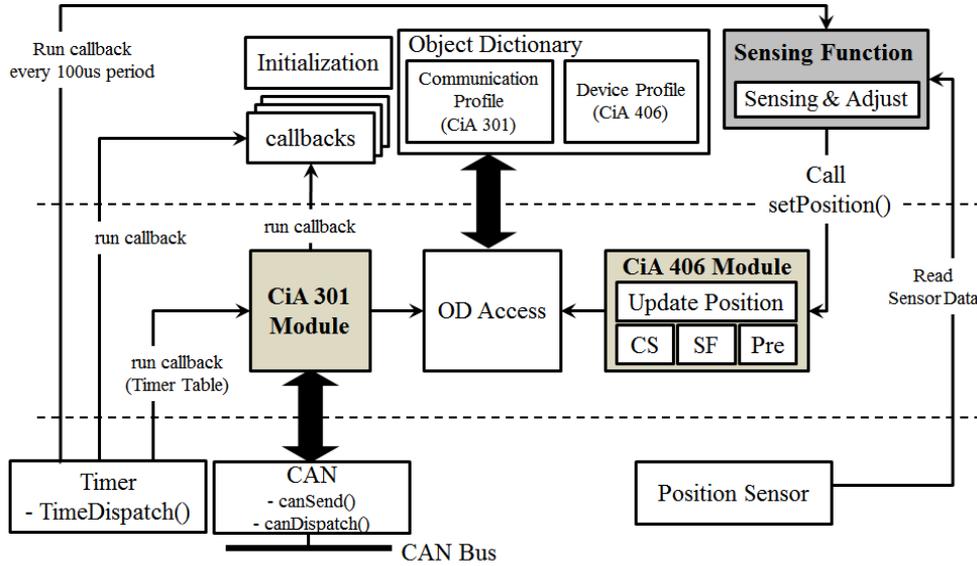


Fig. 4 Software architecture of CANopen encoder

치 데이터를 사용자에게 전달하고, 엔코더 관리 기능을 지원하기 위해 CANopen 마스터에 사용자 명령을 전달한다.

3.2 CANopen 호환 엔코더의 설계 및 구현

실제 엔코더 장치에서 CANopen 기능을 지원하기 위한 구현 내용은 크게 CANopen 프로토콜 계층 구현, 위치 데이터를 CAN 네트워크를 통해 주기적으로 CAN 마스터 장치로 전송하기 위해 위치 정보 센싱 모듈과 CAN 데이터 전송부 간의 연동 인터페이스 구현, 엔코더 장치를 위한 객체 사전 엔트리 구성 등 세 가지로 나눌 수 있다.

먼저 CANopen 프로토콜 구현은 선행연구⁽¹¹⁾에서 기본 통신 프로파일인 CiA 301 모듈과 엔코더 장치 프로파일인 CiA 406 모듈에 대해 CanFestival을 확장, 구현하여 이미 기본 기능을 대부분 검증한 상태이므로 실제 엔코더 로직이 동작하는 마이크로컨트롤러 하드웨어에 의존성이 있는 HAL 포팅 작업을 주로 진행하였다. 본 연구에서 사용한 엔코더는 마이크로칩 사의 dsPIC33EV⁽¹²⁾ 마이크로컨트롤러를 기본 플랫폼으로 사용하며, 타겟 플랫폼 상에서의 구체적인 하드웨어 설정 및 포팅 항목은 Table 2와 같다.

먼저 오실레이터 클럭에 따라 CAN 모듈의 전송 속도를 설정하고, CiA 301 모듈에서 각종 메시지 전송 시 호출하는 `canSend` 함수를 구현하여 CAN Tx 버퍼에 삽입된 메시지를 전송하도록 하였다. 타이머는 16bit의 해상도를 가진 Timer 4를

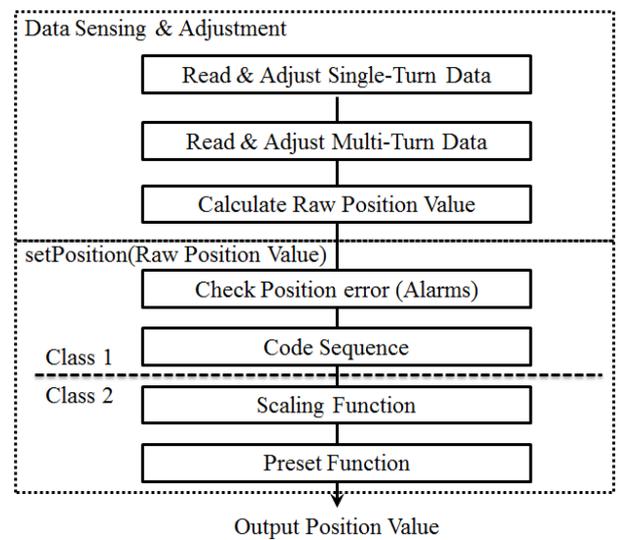


Fig. 5 Functional flow of position sensing and adjustment

사용했으며, 마찬가지로 오실레이터 클럭에 따라 타이머 틱이 적절한 시간으로 변환되도록 하는 매크로를 지정하였다. 또한, CAN 메시지 수신과 타이머 인터럽트 발생 시 인터럽트 핸들러에서 각각 `canDispatch` 함수와 `TimeDispatch` 함수를 호출하도록 하여 적절한 동작을 하도록 하였다. 마지막으로, 본 연구에서 사용한 마이크로칩 사의 dsPIC33 계열 하드웨어는 DMA(Directly Memory Access) 모듈을 사용하여 데이터 전송에 따른 CPU 부하 경감이 가능하므로, 두 개의 DMA 채널을 각기 CAN 송신 및 수신 버퍼로 지정하여

Table 3 OD entries for CANopen encoder

| 인덱스/서브인덱스 | 내용 | 값 |
|--------------|-------------------------------|-----------------|
| 0x1005 / 0x0 | SYNC COB-ID | 0x80 |
| 0x1200 / 0x1 | Rx SDO COB-ID | 0x600 + Node-ID |
| 0x1200 / 0x2 | Tx SDO COB-ID | 0x580 + Node-ID |
| 0x1801 / 0x1 | PDO2 COB-ID | 0x280 + Node-ID |
| 0x1801 / 0x2 | PDO2 transmission type | 0x1(SYNC 기반) |
| 0x1a01 / 0x1 | PDO2 mapping (position value) | 0x60040020 |

메시지 송수신 중 다른 작업을 수행할 수 있도록 구현하였다.

CANopen 엔코더의 소프트웨어 구조는 Fig. 4와 같이 Fig. 2에 제시된 CanFestival 기본 소프트웨어 구조에 100us 주기로 위치 센서에서 데이터를 읽어서 위치를 계산한 뒤 setPosition 함수를 통해 CiA 406 모듈에 전달하는 인터페이스 함수가 추가된 형태이다. 구체적으로 인터페이스 함수가 센서에서 데이터를 읽어서 CiA 406 모듈에 전달하는 과정은 Fig. 5와 같다. 먼저 센서에서 Single-Turn 데이터를 읽은 뒤, 하드웨어적으로 반전되어 있는 데이터를 보수화한다. 그리고 센서에서 Multi-Turn 데이터를 읽은 뒤, 이를 소프트웨어적으로 16bit로 확장한다. 전체 위치 데이터는 Multi-Turn 데이터를 17bit만큼 왼쪽 시프트 시킨 후, Single-Turn 데이터와 OR 연산을 적용하고, setPosition() 함수의 인자로 CiA 406 모듈에 전달된다. CiA 406 모듈은 Alarm 기능을 통해 위치 데이터의 에러 여부를 표시하고, 장치 설정에 따라 가공한 위치 데이터를 최종적으로 객체 사전 엔트리 0x6004 / 0x0 번지에 저장한다.

CANopen 표준에서는 CANopen 장치를 최초 CAN 네트워크에 추가했을 때 바로 사용이 가능하도록 각 통신 객체에서 사용할 아이디를 미리 정의한다. 따라서 기본 설정으로 미리 정의된 값을 사용하면 추가적인 작업 없이 바로 장치 사용이 가능하다. CANopen 엔코더를 위한 주요 기본 설정 내용은 Table 3에 제시한 것과 같이 SDO 통신을 위한 통신 객체 아이디 설정과 SYNC 메시지 수신 시 위치 데이터 동기화를 위한 PDO 설정 사항 등이다.

3.3 CANopen 마스터와 외부 관리 프로그램 구현
멀티 엔코더 동시 관리 시스템에서 CANopen

Table 4 OD entries for CANopen master

| 인덱스/서브인덱스 | 내용 | 값 |
|--------------|--------------------|-----------------------|
| 0x1005 / 0x0 | SYNC COB-ID | 0x80 |
| 0x1006 / 0x0 | SYNC Cycle time | 0x2710 (10 ms로 설정) |
| 0x1280 / 0x1 | Transmit SDO | 0x600+Node-ID |
| 0x1280 / 0x2 | Receive SDO | 0x580+Node-ID |
| 0x1280 / 0x3 | SDO server Node-ID | Node-ID |
| 0x1400 / 0x1 | PDO1 COB-ID | 0x280+Node-ID |
| 0x1600 / 0x1 | PDO1 mapping | 0x20000020 |
| 0x2000 / 0x0 | Slave position 정보 | - |

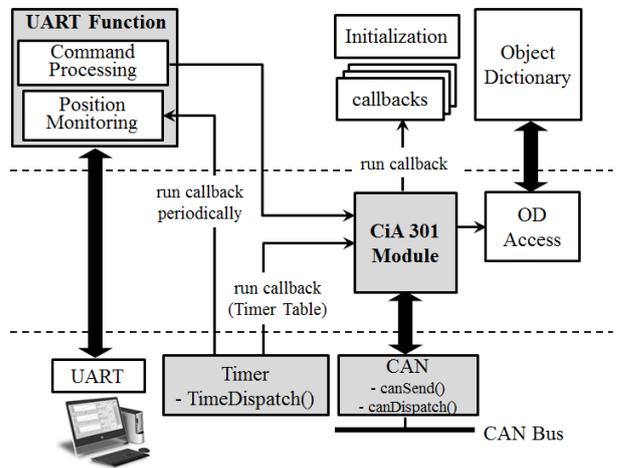


Fig. 6 Software architecture of CANopen master

마스터는 복수 개의 CANopen 엔코더의 설정 사항 관리와 주기적인 위치 데이터 수집을 담당한다. 본 연구에서 구현한 CANopen 마스터 장치는 기본적으로 CANopen 엔코더와 동일한 dsPIC33 계열 마이크로컨트롤러를 타겟 플랫폼으로 사용하므로 3.2절에서 이미 제시한 CanFestival 포팅 과정을 따라 CANopen 프로토콜 계층을 구현하였으며, 매 10 ms 주기로 SYNC 메시지를 생성하여 CANopen 엔코더의 위치 데이터 정보를 취합하도록 하였다. 또한, CANopen 마스터 장치는 외부 PC에서 GUI를 통해 사용자가 네트워크 관리와 각 엔코더 설정을 위해 전달하는 명령어를 파싱하여 CANopen 엔코더로 전송하고, 명령 실행 결과를 다시 외부 관리 PC의 GUI를 통해 사용자에게 알려 주는 중계 역할도 담당한다.

CANopen 마스터 장치의 내부 소프트웨어 구조는 Fig. 6과 같다. 엔코더와 달리 별도의 장치 프로파일이 필요 없으므로 CiA 301 모듈을 중심으로 외부 관리 PC와 통신을 위한 UART 데이터

Table 5 Types of user control commands

| 명령어 | Arg 1 | Arg 2 | Arg 3 | Arg 4 | 동작 |
|----------|-------|-------|-------|-------|---------|
| getstate | 대상 | | | | 상태머신 확인 |
| setstate | 대상 | 상태 | | | 상태머신 설정 |
| read | 대상 | 주소 | 타입 | | 객체사전 읽기 |
| write | 대상 | 주소 | 타입 | 값 | 객체사전 쓰기 |

※ 대상: m(마스터)/s+node-id(슬레이브).
 상태: start, reset, stop, pre-operational

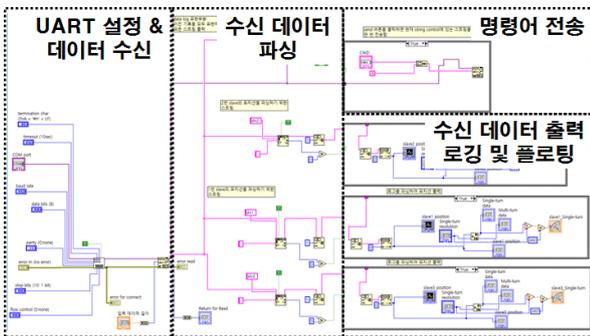


Fig. 7 LabVIEW monitoring program

처리 모듈과 사용자가 내린 명령어 파싱 모듈, 엔코더로부터 수집한 위치 정보 등을 외부 관리 PC로 전송하기 위한 처리 모듈을 포함한다.

먼저 CANopen 마스터에서는 각 CANopen 엔코더와의 통신을 위해 Table 4에 제시된 것과 같이 SDO, PDO, SYNC 통신 객체 아이디와 SYNC 메시지 생성 주기(us 단위), PDO 저장 주소 등의 객체 사전 항목들을 설정해야 한다. 표의 객체 엔트리에 쓰는 값에 해당하는 내용에서 “Node-ID”를 더하는 부분은 각 엔코더별로 해당 엔코더(슬레이브)의 식별자를 표시해 구분하기 위한 내용이다.

외부 모니터링 PC 상에서는 CANopen 마스터를 통한 CANopen 엔코더들의 동작 파라미터 설정과 현재 위치 정보 수집을 사용자 친화적인 GUI를 통해 수행할 수 있도록 Fig. 7과 같은 구조를 가진 LabVIEW 프로그램을 구현하였다. 외부 관리 프로그램은 UART 설정 및 데이터 수신 부분과 수신 데이터를 파싱하는 부분, 파싱한 데이터를 로깅 혹은 그래프 형태로 출력하는 부분, 사용자 입력 명령어를 CANopen 마스터에 전송하는 부분으로 구성하였다. 관리 프로그램과 CANopen 마스터 사이의 통신은 UART를 통해 Table 5와 같이 간단한 사용자 정의 프로토콜을

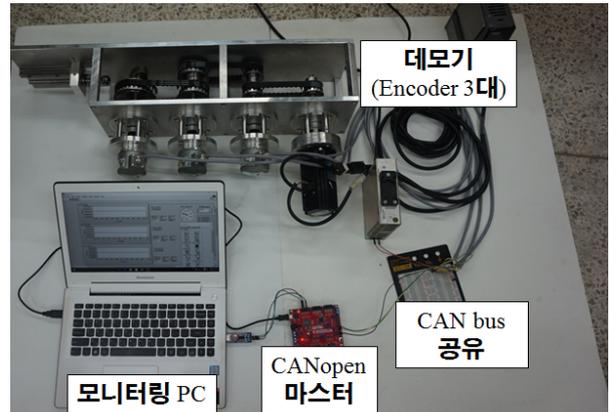


Fig. 8 Multi-encoder management system testbed

사용하도록 하였으며, UART 전송 속도는 9600 ~ 115200 bps 등 다양한 속도를 지원하되 기본적으로 CANopen 마스터의 설정에 따르도록 하였다.

4. 멀티 엔코더 관리 시스템을 이용한 CANopen 엔코더의 동작 검증

4.1 검증 환경

제안된 멀티 엔코더 시스템의 동작과 CANopen 엔코더의 기본 동작 검증을 위한 테스트베드 환경은 Fig. 8과 같이 세 대의 CANopen 엔코더와 한 대의 모터 드라이브로 구성된 데모기와 CANopen 마스터, 모니터링 PC로 구성하였다. 데모기의 엔코더들은 모터가 회전하면 함께 회전하도록 밴드로 이어져 있는데, 각각 축의 기어 비를 다르게 설정하여 엔코더 구성 위치에 따라 다른 회전 변위를 가지도록 하였다. CANopen 장치간 CAN 통신 속도는 1Mbps로 설정하였으며, 모니터링 PC와 CANopen 마스터간 UART 연결의 통신 속도는 115200 bps로 설정하였다.

4.2 검증 시나리오와 결과

본 연구에서 개발한 시스템의 기본 동작 검증 과정은 다음과 같다. 먼저 상용 CANopen 엔코더와의 동작 비교 테스트를 위해 Fig. 8에 제시된 데모기의 엔코더 중 일부를 Kubler사와 Posital사의 상용 엔코더^(13,14)로 대체한 상태에서 동작시 상태머신의 변화와 적절한 통신 객체의 교환이 이루어지는 여부에 대한 확인 테스트를 위해 Table 6의 테스트 시나리오와 같이 CANopen 엔코더의 부팅부터 실제 Operational 상태에 진입하여 위치 데이터를 전송할 때까지의 과정을 확인하였으며 개발한 CANopen 엔코더와 상용 엔코더

Table 6 Test scenario for CANopen stack test

| 순서 | 구분 | 시나리오 | 정상 동작시 실행 결과 |
|----|-----|-----------------------------|---|
| 1 | | Slave Boot-up | • Pre-OP 상태로 자동 진입 |
| 2 | NMT | 상태 천이 | • 지정한 상태로 상태 천이 (Reset, Stop, OP) |
| 3 | | OD 엔트리 0x1005 내용 확인 | • SYNC 메시지 내의 • COB-ID 0x80 확인 |
| 4 | SDO | OD 엔트리 0x1017 번지에 0x03E8 쓰기 | • Heartbeat 생성 주기를 1초로 설정 |
| 5 | NMT | Heartbeat 메시지 확인 | • 매초마다 Heartbeat 메시지 발생 여부 • 장치 상태 OP 유지 |
| 6 | PDO | SYNC 메시지 | • SYNC 메시지 수신 후 TPDO2로 위치 정보 전송 |

2종 모두 Table 6에 제시된 정상 동작 시 기대되는 실행 결과를 정확하게 보임을 확인하였다.

테스트 시나리오에 기반한 기본 동작 테스트 외에도 CANopen 표준 관리그룹인 CiA에서 제공하는 CANopen Conformance Test Tool(이하 CTT)을 사용하여 통신 스택에 대한 추가 검증을 진행하였다. CTT에서는 통신 프로파일 CiA 301에 대한 테스트만을 지원하며, 주기적으로 센서 데이터를 읽어서 위치 데이터를 갱신하는 엔코더 기능이 켜져 있을 경우 테스트 결과를 실패로 간주하는 항목들이 있어 엔코더 기능을 비활성화 한 상태로 테스트를 실시하였다. 구체적인 테스트 과정은 CiA 310 표준⁽¹⁵⁾에 따라 카테고리별로 나누어져 있으며, CANopen 엔코더 동작에 있어 가장 중요한 항목은 SDO, PDO, NMT에 대한 항목들이다. 검증 과정에서 3.2절에서 기술한 것과 같이 CanFestival을 기반으로 구현한 내용 중 비정상 동작 상황에서의 에러 처리가 미비한 항목들이 많이 발견되어 관련 항목을 위한 추가 구현을 진행하였다. 최종 검증 결과, Table 7에 제시된 것과 같이 전체 97개 테스트 항목에 대해 상용 엔코더 대비 기능적으로 우수한 결과를 얻을 수 있었다.

기능적인 검증 외에도 CAN 네트워크를 통한 위치 전송 동작의 동기화 성능도 검증하였다. 비교 대상으로 Kubler사의 상용 엔코더에 개발 엔코더와 동일한 설정을 적용하여 SYNC 메시지에 따라 TPDO 2에서 0x6004 / 0x0의 위치 데이터를 전송하도록 하였으며, 오실로스코프를 사용

Table 7 Conformance test results

| 항목 | 결과 | 개발품 | P사 | K사 |
|-------------------|-------|-----|----|----|
| SDO (30개) | Pass | 23 | 19 | 19 |
| | Warn. | 0 | 0 | 0 |
| | Error | 0 | 0 | 0 |
| PDO (33개) | Pass | 7 | 11 | 11 |
| | Warn. | 25 | 25 | 25 |
| | Error | 0 | 0 | 0 |
| EMCY (6개) | Pass | 8 | 8 | 8 |
| | Warn. | 3 | 2 | 3 |
| | Error | 2 | 2 | 0 |
| Error Ctrl. (20개) | Pass | 0 | 1 | 2 |
| | Warn. | 1 | 1 | 1 |
| | Error | 0 | 0 | 0 |
| States (8개) | Pass | 19 | 16 | 18 |
| | Warn. | 0 | 0 | 0 |
| | Error | 0 | 3 | 2 |
| Total (97개) | Pass | 1 | 1 | 0 |
| | Warn. | 7 | 8 | 7 |
| | Error | 1 | 0 | 0 |
| Total (97개) | Pass | 0 | 0 | 1 |
| | Warn. | 0 | 0 | 0 |
| | Error | 0 | 0 | 0 |
| Total (97개) | Pass | 77 | 70 | 72 |
| | Warn. | 3 | 2 | 0 |
| | Error | 0 | 4 | 5 |
| Total (97개) | Pass | 17 | 21 | 20 |
| | Warn. | | | |
| | Error | | | |

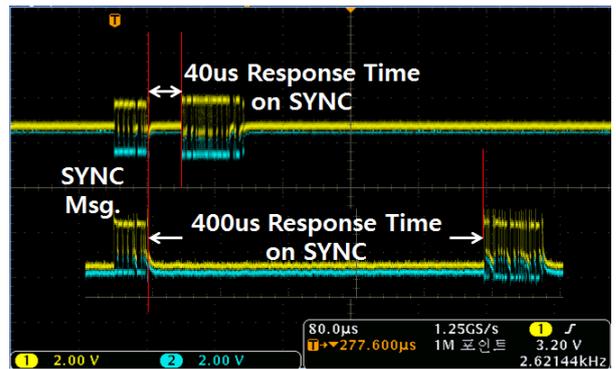


Fig. 9 Response times of CANopen encoders

하여 CAN 메시지 전송을 확인하였다. Fig. 9와 같이 두 엔코더 모두 SYNC 메시지에 따라 위치 데이터를 정상적으로 전송하며, SYNC 메시지 수신 이후 PDO 메시지가 전송되기까지의 응답시간이 상용 제품에서 훨씬 긴 것을 확인할 수 있었다. 응답시간이 짧으면 더 빠른 주기의 동기화가 가능하며, 모션 제어와 같이 정밀한 피드백 제어가 필요한 응용에도 적용이 가능하다는 장점이 있다. Fig. 10은 10 ms 주기로 위치 데이터 동기

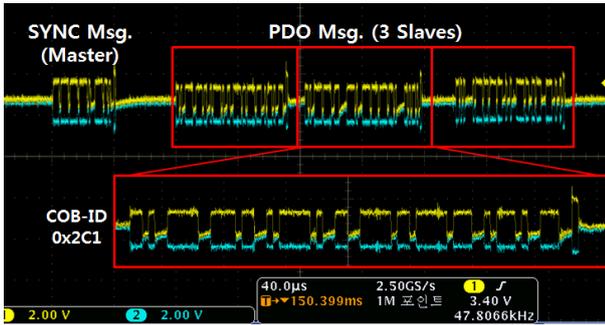


Fig. 10 Synchronization on 10ms period

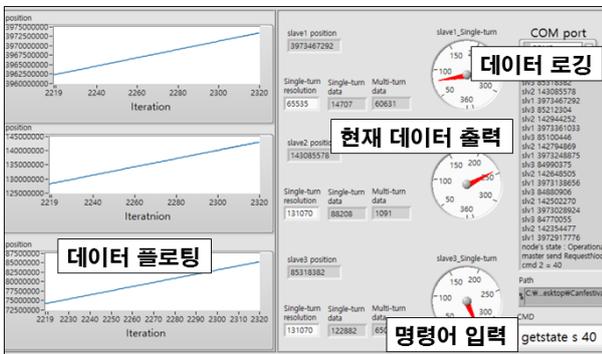


Fig. 11 Snapshot of multi-encoder control and monitoring GUI

화가 진행되는 상황에서 CANopen 마스터의 SYNC 메시지에 따라 데모기에 장착된 세 대의 CANopen 엔코더가 센싱한 위치 정보를 PDO 메시지를 통해 정확하게 전송함을 나타낸다.

마지막으로 외부 관리 소프트웨어의 기능을 검증하였다. 모터 드라이브를 일정 속도(120 rpm)로 회전시켰을 때 사용자 인터페이스에서 Fig. 11과 같이 사용자 정의 프로토콜을 활용한 엔코더 제어 정보 설정 기능과 모니터링 기능이 잘 작동하는 것을 확인할 수 있다. 모니터링 프로그램은 CANopen 마스터와 주고받은 데이터에 대한 로그와 현재 위치 데이터 및 누적된 위치 데이터에 대한 그래프와 360도 기준 현재 각도에 대한 정보를 제공한다. 또한 Table 5에서 제시한 명령어 입력 기능을 이용해 동작 도중 각 엔코더의 상태 확인 및 설정 변경 또한 잘 동작하는 것을 확인하였다.

5. 결론

본 연구에서는 오픈 소스 CANopen 프로토콜 스택인 CanFestival을 기반으로 엔코더 센싱 로직을 통합한 CANopen 표준 호환 엔코더 소프트웨

어 구조를 제안하고, 복수개의 CANopen 엔코더에 대한 모니터링 및 관리 기능을 제공하는 시스템 구조를 설계, 구현하였다.

구현 내용은 상용 제품을 대조군으로 테스트 시나리오, 응답성 비교, CANopen Conformance Test를 사용해 검증하였다. 검증 과정에서 미구현된 CiA 301 일부 항목을 CanFestival에 추가 구현하였으며, 검증 결과에서 기존 상용 제품 대비 기능 지원 및 성능 측면에서 우수한 동작을 보임을 확인하였다.

후기

본 연구는 2015년도 산업통상자원부의 재원으로 한국에너지기술평가원(KETEP)의 지원을 받아 수행한 연구과제입니다.(No. 20132020101970)

참고문헌 (References)

- (1) Hunter, G. W., Stetter, J. R., Hesketh, P. J. and Liu, C. C., 2010, "Smart Sensor Systems," *The Electrochemical Society Interface*, Vol. 19, No. 4, pp. 29~34.
- (2) Meijer, G., Pertijs, M. and Makinwa, K., 2014, "Smart Sensor Systems: Emerging Technologies and Applications," *John Willey & Sons Ltd, Hoboken*, pp. 1~2.
- (3) Bosch, R., 1991, "CAN Specification Version 2.0."
- (4) CAN in Automation(CiA): CANopen, <http://www.cancia.org/index.php?id=canopen>.
- (5) Lee, E-S. and Kim, J-G., 2005, "Measurement and Analysis for Positioning Control Characteristics using Encoder Signal of NC Machine Controller," *Trans. Korean Soc. Mech. Eng. A*, Vol. 29, No. 2, pp. 311~317.
- (6) Ham, H., H., Hong, S., H., Song, J., Baek, J. and Ryu, J., 2012, "Localization of a Tracked Robot Based on Fuzzy Fusion of Wheel Odometry and Visual Odometry in Indoor and Outdoor Environments," *Trans. Korean Soc. Mech. Eng. A*, Vol. 36, No. 6, pp. 629~635.
- (7) Lim, J. S and Lee, Y. J., 2015, "Detection of Absolute Position of Robot Joint Using Incremental Encoders," *Journal of Institute of Control, Robotics and Systems*, Vol. 21, No. 6, pp. 577~582.
- (8) CiA, 2011, "CANopen Application Layer and Communication Profile Version 4.2.0."
- (9) CiA, 2006, "CANopen Device Profile for Encoders Version 3.2.0."

- (10) CanFestival, <http://canfestival.org>.
- (11) Hwang, H., Ahn, H., Kim, S. and Kim, T., 2015, "Implementation of CiA 406 Device Profile for CANopen Compatible Encoders," *Trans. Korean Soc. Mech. Eng. A*, Vol. 39, No. 12, pp. 1287~1295.
- (12) Microchip, dsPIC33EV256GM106 Reference Manual, <http://www.microchip.com/wwwproducts/en/dsPIC33EV256GM106>.
- (13) Fritz Kubler GmbH, 2014, "Technical Manual Absolute Singleturn/Multiturn Encoder F58XX CANopen."
- (14) POSITAL GmbH, 2013, "Absolute Ixarc Rotary Encoder with CANopen Interface User Manual."
- (15) CiA, 2009, "CiA 310 Draft Standard Proposal: Conformance Test Plan Version 1.1.0."