

오픈소스 하드웨어와 MATLAB을 이용한 고속모니터링 시스템 개발

Development of a High Speed Monitoring System Using Opensource Hardware and MATLAB

조 봉 언¹, 이 영 삼^{1*}
(Bongeon Jo¹ and Young-Sam Lee^{1*})

¹Department of Electrical Engineering, Inha University

Abstract: In this paper, we deal with the development of a high speed monitoring system using opensource hardware and MATLAB/Simulink. We use the Arduino Due as opensource hardware. The proposed high speed monitoring system has four features. First, it does not affect the operation of the user application running on a micro-controller. The proposed system performs realtime monitoring using an algorithm that minimizes the computational load. Second, data channels can have different data types and update periods. Consequently, flexible data monitoring is possible. Third, the proposed system uses MATLAB/Simulink for monitoring. As a result, users can utilize all the powerful functions supported by MATLAB to process the monitored data. Finally, the algorithm of the proposed high speed monitoring system can be applied to any micro-controller with serial communication peripheral. Through experiments, we verify the performance of the developed monitoring system. It is shown that the data loss does not happen through the basic experiment. Through realtime monitoring experiment for the sensorless drive system of the BLDC motor, it is illustrated that the proposed system has its strength in high speed monitoring of data changing very fast such as current or back EMF signals.

Keywords: high speed monitoring system, data monitoring, USB data acquisition

I. 서론

임베디드 시스템은 가전제품, 통신 장비, 항공, 자동차, 제어, 로봇, 게임, 의료 등의 다양한 분야에서 활용되고 있다. 최근에는 인공지능, 자율형자동차와 같은 4차 산업혁명의 영향을 받은 분야의 임베디드 시스템 시장의 규모가 증가하고 있다. 임베디드 프로그램의 내부 정보와 변수를 모니터링하는 것은 시스템의 설계와 제작에 있어 개발자의 직관을 높여주고 시스템의 동작에 나타나는 오류를 해결하는데 도움을 준다. 데이터를 모니터링하는 방법은 여러 가지가 존재 한다. 여분의 I/O를 조작하여 핀의 on/off를 측정하거나 LCD와 같은 제한된 인터페이스를 이용하여 데이터를 확인할 수 있다[1]. 최근 마이크로컨트롤러는 소형화 되고 특정한 용도로 발전하고 있어 데이터 모니터링을 위해 필요한 여분의 핀이 존재하지 않고 모니터링 데이터가 저장되지 않는 단점이 있다. JTAG(Joint Test Action Group)과 같은 on-chip 에뮬레이터나 ICE(In-Circuit Emulator)와 같은 장비를 이용하여 데이터를 확인해볼 수 있다[2,3]. 이와 같은 장비는 브레이크 포인트를 설정하여 한 시점에서의 하드웨어와 소프트웨어의 상태를 확인할 수 있다. 이러한 에뮬레이터는 고가의 장비이고 마이크로컨트롤러의 종류에

따라 전용의 에뮬레이터가 필요하다. 또한 마이크로컨트롤러의 프로그램이 멈추어야 메모리에 접근이 가능하기 때문에 어플리케이션에서 사용 중인 통신에서 timeout에러, 모터의 과전류와 같은 기계시스템의 오동작이 발생할 수 있다. 본 논문에서는 어플리케이션의 동작에 영향을 받지 않고 모니터링하는 정보의 기록, 저장이 가능하고 그래프로 디스플레이가 가능한 모니터링 시스템을 개발하고자 한다.

데이터 수집을 위한 DAQ(Data Acquisition)시스템은 전압, 전류, 온도 등과 같은 전기 또는 물리적 현상을 측정하고 수집한다. DAQ 시스템은 전기, 전자, 화학, 의료, 농업, 환경 등 다양한 산업에서 사용되고 있다. 마이크로컨트롤러를 이용한 DAQ 시스템의 개발이 활발히 이루어지고 있다. 마이크로컨트롤러와 USB를 이용해 심전도 신호를 측정하거나 Zigbee 모듈을 이용하여 운동량, 맥박, 피부전도도와 같은 생체신호를 수집하는 DAQ 시스템이 연구되었다[4,5]. 대기 온도, 풍향, 습도와 같은 날씨를 측정하기 위한 DAQ 시스템[6] 그리고 농업을 위한 대지의 습도, 온도, 대기온도, 물의 상태를 수집하는 시스템[7]과 같은 환경의 정보를 수집하기 위한 연구들이 진행되었다. 사물인터넷의 발전으로 다양한 스마트 시설물의 정보를 수집하고 원격제어를

*Corresponding Author

Manuscript received October 28, 2019; revised November 13, 2019; accepted November 24, 2019

조봉언: 인하대학교 전기공학과(jbw0803@gmail.com, ORCID[®] 0000-0001-5802-0133)

이영삼: 인하대학교 전기공학과(lys@inha.ac.kr, ORCID[®] 0000-0003-0665-1464)

※ 이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2017R1D1A1B03029578).

위한 연구들이 진행되고 있다[8,9]. 위의 DAQ 시스템들은 특정한 목적을 위하여 개발되었으며 데이터 취득을 위한 센서가 정해져있다. 진동을 측정하기 위한 DAQ 시스템[10]과 원격접속을 통한 DC모터 제어연구[11]에서는 NI사의 DAQ보드를 통한 데이터 모니터링을 수행한다. LabView 프로그램을 이용하여 모니터링하는 데이터의 저장과 그래픽 디스플레이를 지원한다. 하지만 PC의 PCI슬롯에 별도의 DAQ보드를 장착하여야 하는 단점이 있다. 본 논문에서는 임베디드 시스템에 적용하여 데이터를 수집할 수 있는 고속모니터링 시스템을 제안한다. 제안하는 고속모니터링 시스템은 별도의 장치 없이 마이크로컨트롤러에서 동작하고 다양한 센서를 사용할 수 있으며 임베디드 시스템의 내부의 변수와 같은 정보를 모니터링할 수 있다.

본 논문은 다음과 같이 구성된다. 먼저 II절에서는 고속모니터링 시스템을 구현하기 위한 프로토콜과 알고리즘을 제안하고 제안하는 시스템을 구현한다. 다음으로 III절에서는 오픈소스하드웨어와 MATLAB/Simulink를 이용한 실험을 통해 제안하는 시스템을 검증한다. 마지막으로 IV절에서 결론을 제시한다.

II. 제안하는 고속모니터링 시스템

본 논문에서는 마이크로컨트롤러에서 수행중인 프로그램의 내부정보를 확인할 수 있는 고속모니터링시스템을 제안한다. 제안하는 고속모니터링 시스템은 다음과 같은 특징을 가진다.

첫 번째로, 제안하는 고속모니터링 시스템은 마이크로컨트롤러에서 수행중인 프로그램의 동작에 간섭을 일으키지 않고 적은 시스템자원을 활용하여 내부 정보를 모니터링할 수 있게 한다. 마이크로컨트롤러의 프로그램이 동작하고 남은 여분의 자원과 소개되는 프로토콜과 알고리즘을 이용하여 모니터링 시스템에 발생하는 오버헤드를 줄인다.

두 번째로, 모니터링하는 채널의 데이터는 서로 다른 자료형과 업데이트 주기를 가질 수 있다. 각 채널에 할당된 변수에 맞추어 표 1과 같은 자료형을 가질 수 있다. 이는 C 언어 변수의 자료형을 기반으로 한다. 모니터링하고자 하는 센서 측정값, 제어시스템의 제어량과 같이 모니터링이 필요한 다양한 변수를 자료형 변환 없이 모니터링할 수 있다. 또한 각 채널의 데이터는 가장 빠른 채널의 주기의 정수배에 해당하는 시간의 업데이트 주기를 가질 수 있다.

표 1. 제안하는 고속모니터링 시스템이 지원하는 자료형. Table 1. Data types supported by a proposed high speed monitoring system.

정수형 변수	8 bit	boolean
	8 bit	unsigned/signed char
	16 bit	unsigned/signed short
	32 bit	unsigned/signed int
실수형 변수	32 bit	float
	64 bit	double

세 번째로, MATLAB/Simulink를 이용하여 전송된 데이터를 모니터링한다. 'From Target'이라는 이름의 데이터를 수신하는 사용자 정의블록을 C-code S-function을 이용하여 구현하고 Simulink 환경에서 전송된 데이터를 처리한다. 이는 전송된 데이터를 Simulink와 MATLAB의 강력한 데이터 처리 능력을 이용하여 손쉽게 저장, 후처리, 디스플레이 할 수 있다.

마지막으로 다양한 임베디드 시스템에 적용할 수 있다. 본 논문에서 제안하는 통신 프로토콜을 만족시키면 마이크로컨트롤러의 종류와 상관없이 고속모니터링 시스템을 사용할 수 있다.

1. 통신 프로토콜

본 논문에서 제안하는 고속모니터링 시스템은 여러 채널의 데이터를 수집하고, 각 채널은 각자의 자료형과 주기를 가진다. 이를 위해 다음과 같은 통신 프로토콜을 제안한다.

모니터링하고자 하는 채널의 데이터는 그림 1과 같이 구성된다. 그림 1에서 나타내는 채널은 32bit 자료형의 데이터가 할당되어 있다. 그림 1의 (a)는 바이너리 데이터로 구성된 데이터이고 base16 인코딩을 이용하여 (b)와 같은 형태로 변경한다. base16 인코딩은 이진데이터를 16개의 ASCII문자로 인코딩하는 방법이다. 인코딩이 끝난 채널의 데이터는 채널과 자료형 정보가 더해져 기존의 데이터 크기에 비해 2배 커진 크기를 가진다. 제안하는 고속모니터링 시스템의 특징 중 하나인 수행중인 프로그램의 간섭과 더불어 데이터 전송 중에 생길 수 있는 손실 그리고 동기화가 어긋나는 것과 같은 오류를 방지하기 위해 base16 인코딩 방식을 사용한다.

2. 제안하는 고속모니터링 시스템의 알고리즘

제안하는 고속모니터링 시스템은 그림 2와 같은 구조를 가진다. 마이크로컨트롤러는 본래의 시스템이 수행해야할 프로그램을 실행하면서 모니터링하고자하는 채널의 데이터들을 수집하여 그림 1과 같은 프로토콜에 맞추어 저장한다. 저장된 데이터는 일정한 시간에 맞추어 USB 통신을 통해 PC에 전송된다. PC에서는 본 논문에서 제안한 알고리즘을 기반으로 구현한 'From Target'이라는 Simulink용 사용자 정의 블록을 이용하여 인코딩된 데이터를 저장하고 디코딩하여 모니터링이 가능한 데이터로 처리한다.

마이크로컨트롤러에서 한 주기마다 저장되는 데이터는 그림 3의 (a)와 같은 형태로 저장된다. 이때 각 채널의 데이터는 그림 1의 (b)와 같은 형태를 가진다. 처음과 끝에

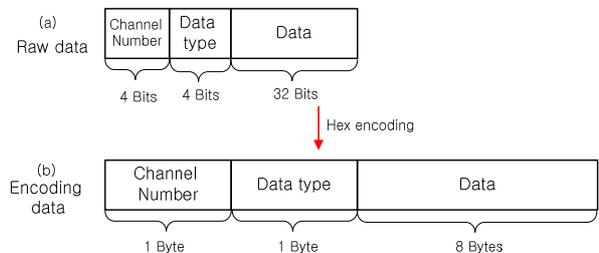


그림 1. 32bit 변수가 할당된 채널의 데이터.

Fig. 1. Data of channel to which 32-bit data type variable is assigned.

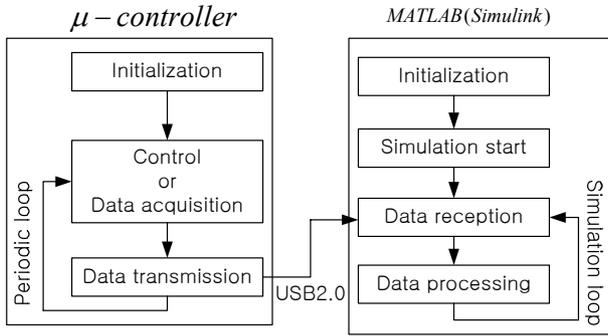


그림 2. 제안된 고속모니터링 시스템의 개념도.

Fig. 2. The concept diagram of the proposed high speed monitoring system.

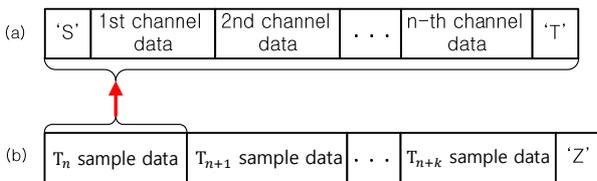


그림 3. USB 통신을 위한 데이터 패킷.

Fig. 3. Data packet for USB communication.

들어가는 'S'와 'T'는 각 주기마다 전송할 데이터에서 처음과 끝을 나타내는 문자로 base 16 인코딩에서 사용하지 않는 문자이다. 그림 3의 (b)는 USB 통신을 통해 PC로 전송되는 데이터의 패킷을 나타낸다. USB 통신의 특성상 일정 시간마다 데이터를 전송할 수 있기 때문에 여러 주기의 데이터를 모아서 하나의 데이터 패킷을 구성한 후 한 번에 전송한다. 전송할 데이터패킷의 끝을 나타내기 위한 문자로 'Z'를 사용한다.

데이터 수신을 위한 Simulink용 사용자 정의블록 'From Target'의 내부 알고리즘은 그림 4와 같이 나타낼 수 있다. Simulink에서 모델을 생성하여 실행하면 처음에 초기화 과정을 거친다. 초기화과정에는 USB 통신을 사용하기 위한 준비작업을 수행하고 모니터링 데이터의 주기를 설정한다. 그 이후 그림 3의 (b)의 형태로 인코딩되어 전송되는 데이터를 받아 저장한다. 데이터 패킷의 마지막으로 설정한 'Z'를 받으면 저장된 데이터를 디코딩하여 모니터링하는 채널에 업데이트를 해준다. 모든 데이터를 디코딩한 다음 처음으로 돌아가 인코딩된 데이터를 받을 준비를 한다.

3. 마이크로 컨트롤러 펌웨어

제안하는 고속모니터링 시스템을 구현하기 위하여 마이크로컨트롤러 ATSAM3X8E를 탑재한 Arduino Due를 사용한다. 오픈소스 하드웨어인 Arduino Due는 ARM Cortex-M3 마이크로컨트롤러를 탑재한 보드로 84 MHz의 clock으로 동작한다. 54개의 digital I/O 핀과 더불어 12개의 PWM핀, 16개의 아날로그 입력, 2개의 DAC의 주변장치를 가지고 있다. 또한 통신을 위한 USART 핀과 high-speed USB 포트가 있다.

제안하는 고속모니터링시스템의 마이크로 컨트롤러 펌웨어는 초기화, 데이터 인코딩, 통신으로 크게 세 부분으로

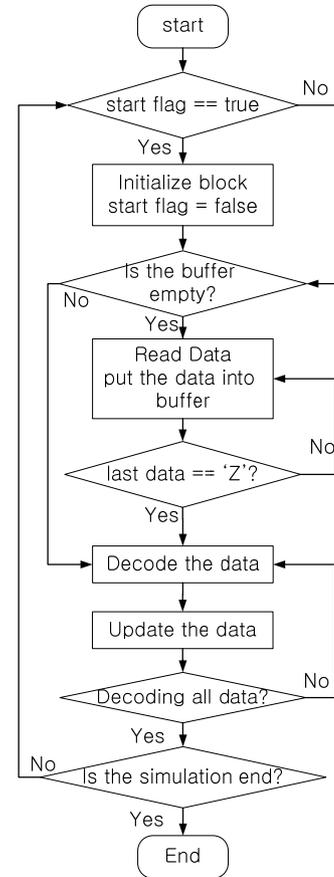


그림 4. 사용자정의블록의 순서도.

Fig. 4. The flowchart of the user-defined block.

구성된다. 첫 번째, 초기화 부분은 수집하려는 데이터의 채널 개수와 채널에 할당된 변수의 자료형을 설정한다. 내장된 USB의 통신 속도와 인코딩된 데이터의 크기를 고려하여 몇 번의 샘플링 시간이 지난 후 전송할 것인지를 결정한다. 두 번째, 데이터 인코딩 부분은 일정한 주기마다 통신 프로토콜에 맞추어 모니터링하고자 하는 데이터를 그림 3의 (a)와 같이 형태로 변환하여 저장한다. 마지막으로 통신과정에서는 버퍼에 저장된 데이터를 그림 3의 (b)와 같이 마지막 구분 문자 'Z'를 포함하여 전송한다.

4. MATLAB 소프트웨어

마이크로컨트롤러에서 전송한 데이터는 그림 4의 알고리즘을 통해 구현된 Simulink용 사용자정의 블록인 'From Target' 통해 수신한다. 이때 'From Target' 블록은 USB통신을 처리할 수 있게 하기 위해 m-file S-function이 아닌 C-code S-function을 이용하여 구현한다. 그림 5는 사용자정의 블록을 이용하여 만든 Simulink 모델이다. 블록파라미터를 변경하여 채널을 설정할 수 있고 'Sample time'으로 나타난 주기는 대화상자에 값을 지정하는 방법으로 변경할 수 있다.

III. 제안하는 고속모니터링 시스템 실험

본 논문에서 제안하는 고속모니터링 시스템을 검증하기 위한 실험을 한다. 성능을 검증하기 위하여 Simulink에서 제공하는 블록을 이용한 실험과 결과를 비교하고자 한다.

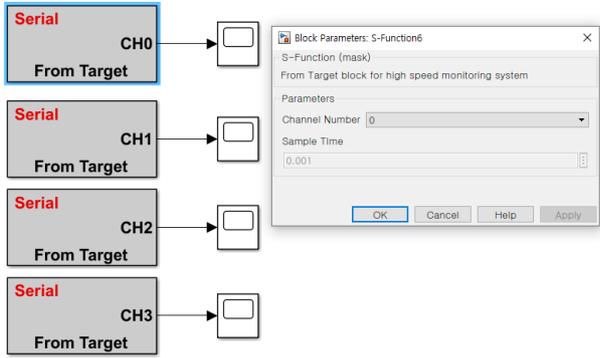


그림 5. 고속모니터링 시스템을 위해 구성한 Simulink 모델.
Fig. 5. Simulink model constructed for high speed monitoring system.

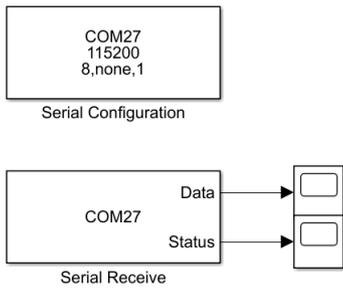


그림 6. ‘Instrument Control Toolbox’의 블록을 이용한 데이터 모니터링 모델.
Fig. 6. The data monitoring model using a block supported by ‘Instrument Control Toolbox’

Simulink의 ‘Instrument Control Toolbox’에서 제공하는 블록을 이용하여 그림 6과 같은 Simulink 모델을 설계한다. 각 블록의 파라미터는 그림 7과 같이 설정한다. 그림 7의 (a)는 ‘Serial Configuration’ 블록의 파라미터를 설정을 나타낸다. 그림 7의 (b)는 ‘Serial Receive’ 블록의 설정을 나타낸다. ‘Instrument Control Toolbox’을 이용한 모니터링 실험의 설정은 표 2와 같다. ‘Instrument Control Toolbox’에서 제공하는 블록을 이용할 경우 복수의 데이터를 수신할 경우 데이터의 유형과 샘플링 주기가 모두 동일해야 한다는 제약이 있다. ‘Serial receive’ 블록은 블록킹모드를 설정할 수 있다. 블록킹모드는 Simulink의 실행주기마다 ‘Header’와 ‘Terminator’를 수신했는지 폴링방식으로 데이터를 확인하는 모드이다. 블록킹모드를 사용하지 않으면 ‘Header’와 ‘Terminator’의 수신여부를 ‘Status’를 이용해 나타낸다. ‘Header’와 ‘Terminator’가 수신되었으면 ‘Data’에 새로운 값을 출력하고 ‘Status’에는 1을 출력한다. ‘Header’와 ‘Terminator’가 수신되지 않았으면 ‘Data’를 이전 출력값 혹은 특정한 값으로 출력하고 ‘Status’에 0을 출력한다. 실험에서는 블록킹모드를 사용하지 않는다.

그림 8은 ‘Instrument Control Toolbox’의 블록을 이용하여 데이터를 수신한 결과이다. Matlab 명령어 ‘tic/toc’을 이용하여 데이터 수신에 소요된 시간을 측정하면 16.91초이다. 비어있는 Simulink 모델을 이용하여 모델의 초기화와 종료에 소요되는 시간은 약 1.2초로 측정되었다. 마이크로컨트롤러

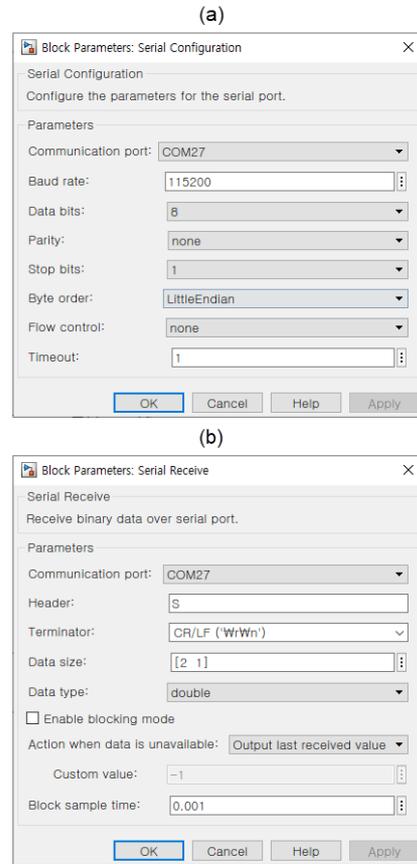


그림 7. (a) 블록 파라미터 : Serial Configuration.
(b) 블록 파라미터 : Serial Receive.
Fig. 7. (a) Block parameters : Serial Configuration.
(b) Block parameters : Serial Receive.

표 2. 모니터링 실험을 위한 파라미터 설정.
Table 2. Parameter setting for monitoring experiment.

설정	값
통신포트	COM27
통신방법	High speed USB
바이트순서	Little Endian
Timeout	1 [sec]
Header	'S'
Terminator	'\r\n'
데이터 종류	double
데이터 크기	2
블록킹 모드	X
샘플링 시간	0.001 [sec]
데이터 수신 시간	10 [sec]

에서 10초 동안 전송한 데이터를 수신하여 디스플레이 하는데 걸리는 시간이 약 15.7초이다. 그림 8의 (a)은 수신된 데이터를 출력하는 그래프로 빨간색 그래프는 1번 데이터로 1 ms 마다 0.001씩 증가한다. 파란색 점선 그래프는 2번

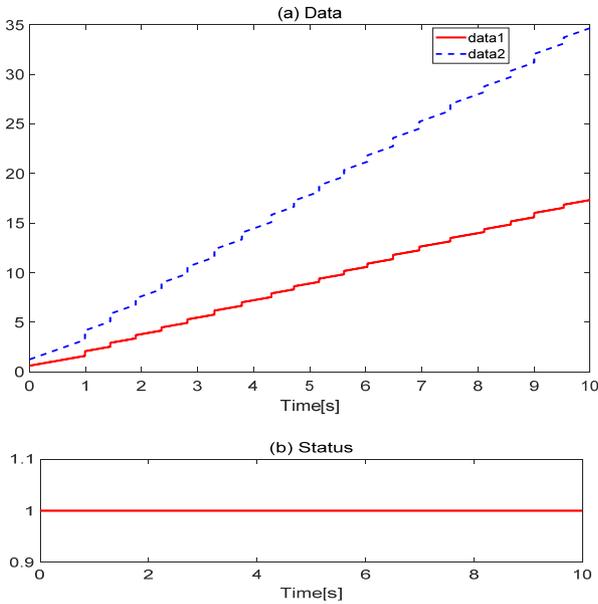


그림 8. ‘Instrument Control Toolbox’의 블록을 이용한 실험결과 : (a) Data, (b) Status.

Fig. 8. The experimental result using a block supported by ‘Instrument Control Toolbox’ : (a) Data, (b) Status.

데이터로 1 ms 마다 0.002씩 증가한다. 첫 번째 데이터를 기준으로 0.990초 시점에 수신된 값이 1.619에서 2.094로 증가한다. 0.475의 차이는 475개의 데이터가 수신되지 못했음을 나타낸다. 첫 번째 누락이 일어난 뒤 실험이 종료될 때까지 19번의 데이터 누락이 일어난다. 데이터의 누락이 나타날 때 300~700개의 데이터가 누락되어 수신된다. 그림 8의 (b)는 ‘Status’를 나타내는 그래프로 실험이 끝날 때 까지 1을 출력한다. 이는 모든 시점에서 ‘Header’와 ‘Terminator’가 수신되었음을 나타낸다.

본 논문에서 제안하는 고속모니터링 시스템을 구현한 그림 5의 모델을 이용한 실험을 한다. 총 4개의 데이터를 모니터링한다. 채널 0번과 1번 데이터는 double형이며 샘플링 주기는 각자 1 KHz, 200 Hz이다. 두 데이터는 1 ms마다 0.001씩 증가한다. 채널 2번과 3번 데이터는 signed char, unsigned char형이며 샘플링 주기는 100 Hz이다. 두 데이터는 10 ms 마다 1씩 증가한다. 마찬가지로 Simulink에서 수신에 사용되는 시간을 측정하기 위해 ‘tic/toc’ 명령어를 사용하면 11.4초가 측정된다. 마찬가지로 비어있는 모델에서 측정된 시간을 빼면 그림 5와 같이 제안하는 고속모니터링 시스템을 이용하여 마이크로컨트롤러에서 10초 동안 전송한 데이터를 수신하는데 소요된 시간은 약 10.2초이다. 제안된 고속모니터링 시스템을 이용하여 데이터를 수신한 시간과 실제 실험에 소모된 시간이 약 0.2초 차이가 나는데 이건 초기화 과정과 종료과정에서 소모되는 시간으로 볼 수 있다.

두 개의 double형 데이터를 모니터링한 결과는 그림 9에서 나타난다. 빨간색 실선 그래프는 채널 0번을 나타내고 파란색 점선은 채널 1번을 나타낸다. (a)는 전체 실험시간을 나타낸다. 0초에서 10초에서 시작한 데이터들은 10초에서

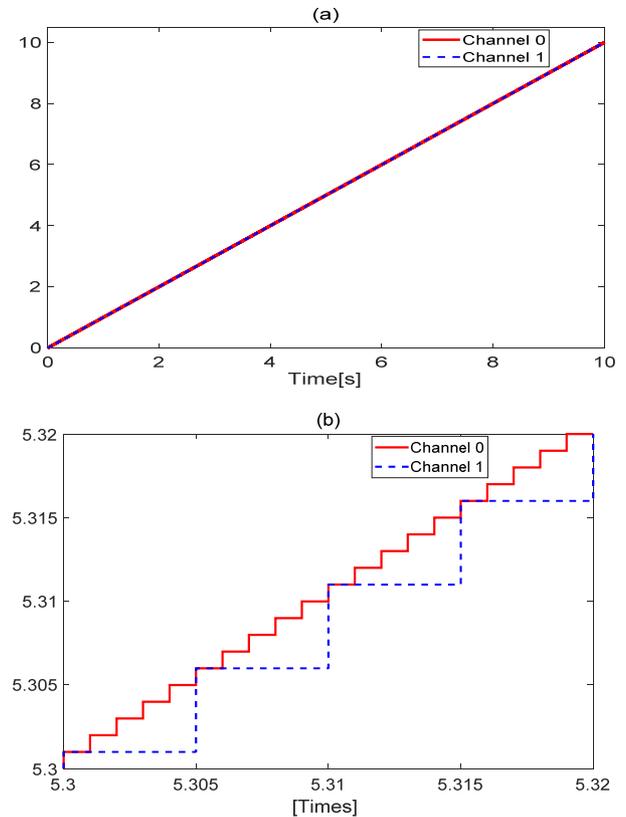


그림 9. 제안하는 고속모니터링 시스템의 실험결과 : 실수형 데이터.

Fig. 9. The experimental result of the proposed high speed monitoring system : real data.

10이 되며 실험이 종료된다. 두 채널의 데이터는 동일하며 샘플링 시간만 서로 다르기 때문에 그래프가 겹쳐 보인다. 데이터측정이 종료될 때 까지 누락된 데이터 없이 모두 측정된 것을 확인할 수 있다. (b)는 5.3초부터 5.32초까지의 시간을 확대한 그래프이다. 빨간색 그래프인 채널 0번은 1 KHz의 샘플링시간을 가지고 있기 때문에 1 ms마다 0.001씩 증가하는 것을 확인할 수 있다. 파란색 점선 그래프는 채널 1번으로 200 Hz의 업데이트 주기를 가지며 데이터의 크기가 5 ms 마다 0.005씩 증가하는 것을 확인할 수 있다.

채널 2와 채널 3의 실험결과를 나타낸 그림 10에서 빨간색 실선 그래프는 채널 2를 나타내고 파란색 점선은 채널 3을 나타낸다. 채널 2는 signed char 형으로 -128~127의 값을 가지며 채널 3은 unsigned char형을 0~255의 값을 가진다. 두 채널의 데이터는 초당 100씩 증가되는 것을 확인할 수 있으며 오버플로우가 일어나는 것을 확인할 수 있다. 채널 2의 경우 127에서 1이 증가하여 -128이 되고 채널 3의 경우에는 255에서 0이 된다.

Simulink의 ‘Instrument Control Toolbox’에서 제공하는 블록을 이용한 데이터모니터링은 같은 유형의 데이터만 수신할 수 있다. 그림 6의 Simulink 모델을 이용한 실험을 통해 1ms마다 double형 데이터 2개를 수신하는데 데이터의 연속성이 깨지면서 정확한 데이터를 수신하지 못하였다. 또한 10초간 데이터모니터링하는 데 약 16초의 시간이 소모되어

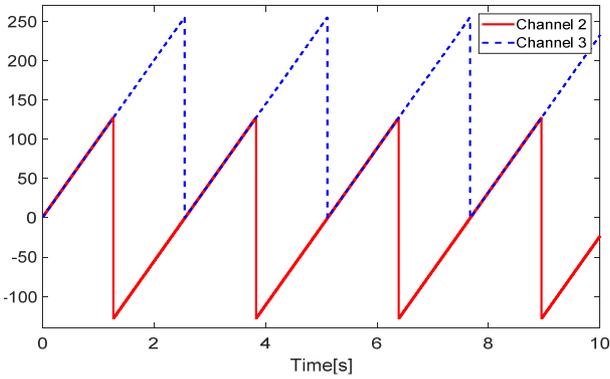


그림 10. 제안하는 고속모니터링 시스템의 실험결과 : 정수형 데이터.

Fig. 10. The experimental result of the proposed high speed monitoring system : integer data.

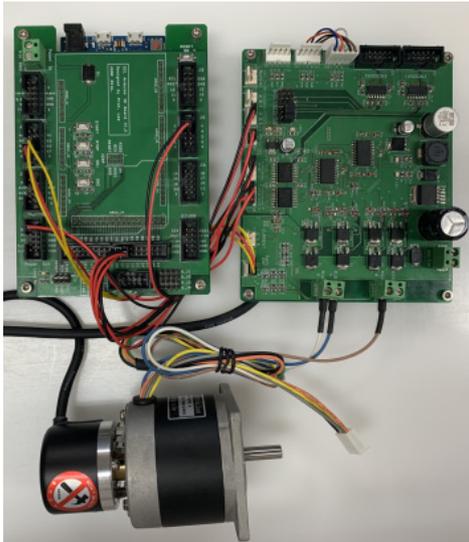


그림 11. BLDC 모터 구동 시스템.

Fig. 11. BLDC motor drive system.

실시간 데이터모니터링이 불가능하다. 제안하는 고속모니터링 시스템은 그림 5의 Simulink 모델을 이용하여 2개의 double형 데이터와 2개의 8bit 정수형 데이터를 모니터링이 하였다. 각 데이터의 샘플링 주파수는 1 KHz, 200 Hz 그리고 100 Hz로 여러 샘플링시간을 가지고 있다. 그림 9와 그림 10을 통해 모니터링하는 데이터의 누락 없이 전부 수신하는 것을 볼 수 있다. 또한 10초간 실험하는 동안 전송된 데이터를 수신하는데 걸리는 시간이 약 10.2초로 Simulink의 시작과 종료에 소요되는 시간을 제외하면 실시간 모니터링이 됨을 확인할 수 있다.

다음은 BLDC모터의 센서리스 구동에 제안하는 고속모니터링 시스템을 적용하는 실험을 수행한다. 실험을 위해 그림 11과 같은 BLDC 모터구동 시스템을 이용한다. Arduino Due를 이용하여 BLDC 센서리스 제어와 속도제어 그리고 고속모니터링 시스템을 적용한다. 실험에 사용한 인버터는 자체 제작한 인버터로 총 4개의 하프 브릿지를 가지고 있어 1개의 BLDC 모터를 구동하거나 2개의 DC모터를 구동

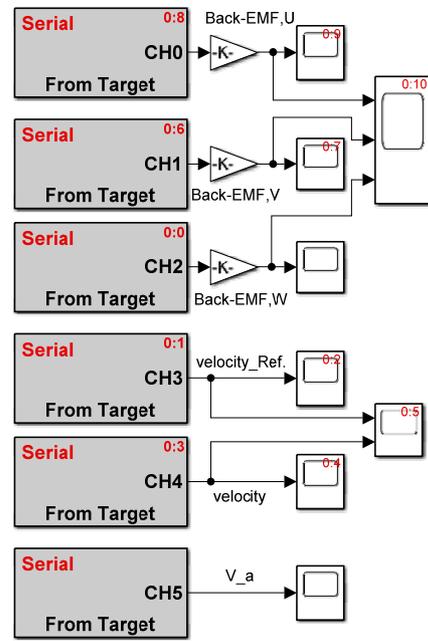


그림 12. BLDC 모터 구동 시스템의 모니터링을 위한 Simulink 모델.

Fig. 12. Simulink model for high speed monitoring of BLDC motor drive system.

할 수 있는 인버터이다. 실험에 사용한 BLDC모터는 (주)디엔지위드사의 BL5057이다. 이 모터는 50W의 출력을 가지고 정격전압은 DC 24V이다. 속도제어에 사용한 엔코더는 Autonics 사의 E40H8으로 1000PPR의 분해능을 가지고 있는 로터리 엔코더이다. BLDC 모터의 센서리스 제어를 위해 역기전력을 측정하여 회전자의 위치 정보를 얻는 기법을 사용한다[12,13]. 속도제어는 PID제어기를 이용한다.

BLDC모터가 구동될 때 발생하는 역기전력이나 전류파형은 변화가 매우 빨라 고속의 샘플링이 필요하다. 보통의 관련분야 연구, 개발인력들은 고속의 data를 관측하기 위해 관측하고자 하는 값을 DAC (Digital to Analog Converter)를 이용해 아날로그로 출력한 후 오실로스코프로 관측하거나 기록하는 방법을 사용하였다[14,15]. 하지만 이 방법을 통한 관측은 DAC를 거치는 동안 잡음의 유입을 동반하고 장비의 수준이 낮을 경우 데이터의 기록이 원활하지 않은 경우가 많다.

그림 13은 모니터링을 통해 측정된 BLDC 모터의 역기전력 파형을 나타낸다. (a)는 U상의 역기전력, (b)는 V상의 역기전력 그리고 (c)는 W상의 역기전력이다. 역기전력을 측정하기 위해 Arduino Due의 ADC(Analog to Digital Converter)기능을 이용한다. 측정된 역기전력은 디지털 값을 가지게 되므로 이를 원래의 값으로 모니터링하기 위하여 그림 12와 같이 Simulink의 gain 블록을 이용한 적절한 스케일링을 한다. 모니터링하는 역기전력 파형은 20KHz의 샘플링주파수를 가진다.

그림 14는 속도제어를 위한 속도지령과 BLDC 모터의 속도를 나타낸 그래프이다. PID제어기로 이루어진 속도제어 시스템은 1 KHz로 동작한다. 따라서 모니터링된 속도지령과 BLDC 모터의 속도는 1 KHz의 샘플링주파수를 가진다. 빨간색 실선은 기준속도를 나타내고 굵기가 굵은 파란색

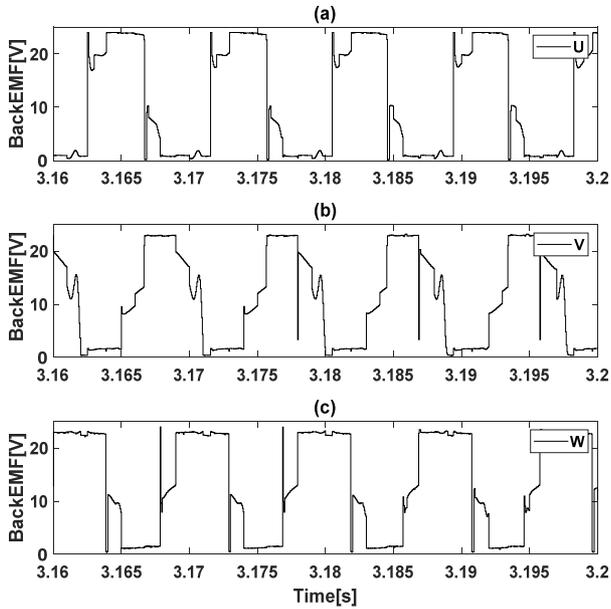


그림 13. BLDC 모터 센서리스 구동 실험 결과 - 역기전력.

Fig. 13. The experimental result of the BLDC motor sensorless control - back-electromotive force.

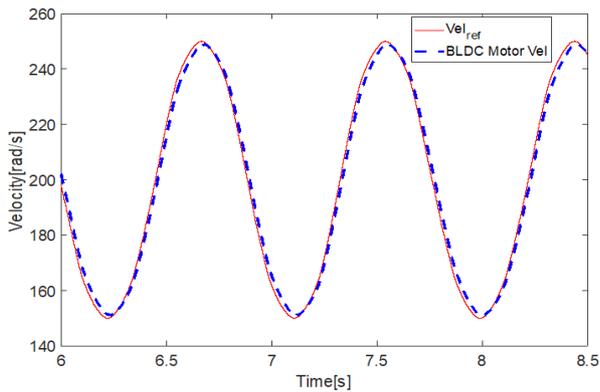


그림 14. BLDC 모터 속도제어 결과.

Fig. 14. The experimental result of the BLDC motor speed control.

점선은 BLDC모터의 회전속도를 나타낸다. BLDC모터가 기준속도를 추종하여 속도제어가 수행되고 있음을 보여준다.

그림 15는 PID제어를 거쳐 BLDC모터에 인가되는 전압을 나타내는 그래프이다. 마찬가지로 1KHz의 샘플링 주파수를 가진다.

BLDC 모터의 센서리스 제어를 수행하고 있는 마이크로컨트롤러에 영향을 끼치지 않고 내부의 여러 데이터를 확인할 수 있는 고속모니터링 시스템을 확인하였다. 그림 12의 채널 0, 채널 1, 채널 2번 데이터와 같이 측정된 데이터를 Simulink의 블록을 통해 실제 물리량과 같이 스케일링하여 확인할 수 있다. 실시간으로 업데이트되는 특징은 제어시스템 개발 시 제어기 설계자가 제어시스템의 응답양상을 실시간으로 확인할 수 있게 함으로써 설계자의 직관향상에 크게 기여할 수 있다. 또한 그림 15의 제어입력과 같이 시스템에 적용되는 물리량을 확인하여 시스템의 손상을 방지하는데 활용 가능하다.

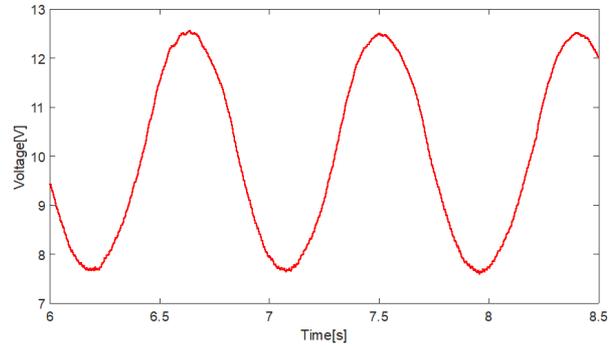


그림 15. BLDC 모터에 인가되는 전압.

Fig. 15. The voltage applied to the BLDC motor.

IV. 결론

본 논문에서는 마이크로컨트롤러에서 수행중인 어플리케이션의 내부 변수 값의 변화를 실시간으로 확인할 수 있는 고속모니터링 시스템을 제안하였다. 제안된 시스템은 마이크로컨트롤러에서 동작중인 어플리케이션에 대한 간섭 없이 내부 정보를 모니터링할 수 있도록 설계되었다. 고속모니터링 시스템을 위한 통신 프로토콜을 정의하고 마이크로컨트롤러의 펌웨어와 MATLAB/Simulink 소프트웨어의 알고리즘을 제안하였다. 이를 통해 제안하는 고속모니터링 시스템은 서로 다른 자료형의 변수를 모니터링할 수 있고 각 채널의 샘플링 주기를 다르게 설정할 수 있다. 또한 MATLAB/Simulink를 이용한 모니터링으로 손쉬운 데이터연동이 가능하다. 본 논문에서 제시한 프로토콜 생성과 알고리즘은 다양한 마이크로컨트롤러에 적용하여 고속모니터링 시스템을 구현할 수 있어 높은 이식성을 가지고 있다.

ARM Cortex-M3 CPU를 탑재한 오픈소스하드웨어 Arduino Due와 Simulink의 사용자정의 블록을 이용하여 고속모니터링 시스템을 구현하였다. 변수의 증가를 두 개의 샘플링 주파수로 모니터링하는 실험을 통해 제안한 고속모니터링 시스템의 유용성을 검증하였다. 그리고 BLDC 모터의 센서리스 속도 제어 시스템에 적용하여 마이크로컨트롤러의 어플리케이션이 동작하는 가운데 모니터링하고자 하는 변수들을 측정하는 실험을 수행하였다. 실험을 통해 20 KHz로 측정되는 BLDC 모터의 역기전력을 모니터링하고 1 msec의 주기를 가지는 속도제어 시스템의 고속모니터링에 적용할 수 있음을 확인하였다. 제안되는 모니터링 시스템을 통해 유형과 샘플링 주기가 다른 데이터의 관측과 기록이 가능하다. 또한 BLDC모터의 역기전력신호와 같이 신호의 변화가 매우 빨라 고속 샘플링이 필요한 경우에 특히 강점을 가진다. 제안되는 시스템을 이용할 경우 임베디드 어플리케이션에서 발생하는 현상을 효율적으로 관측, 기록, 분석할 수 있고 이를 통해 개발하는 어플리케이션의 성능을 평가하고 개선할 수 있다. 또한 뛰어난 실시간 성능은 제어시스템의 개발과정에서 제어기 설계자의 직관향상에 크게 기여할 수 있다.

REFERENCES

- [1] P. H. Kulkarni, P. D. Kute, and V. N. More, "IoT based data processing for automated industrial meter reader using raspberry Pi," *Proc. of International Conference on Internet of Things and Application*, pp. 107-111, Jan. 2016.
- [2] M. C. Cheol, Y. J. Kim, S. E. Ha, Y. K. Jeon, and C. D. Lim, "Implementation of a real-time tracing tool for remote debugging of SoC programs," *The KIPS Transactions : Part A (in Korean)*, vol. 12-A, no. 7, pp. 583-588, Dec. 2005.
- [3] H. G. Lee, Y. S. Han, and S. U. Kim, "Debugging environment via USB-JTAG interface for EISC embedded system," *The KIPS Transactions : Part A (in Korean)*, vol. 17-A, no. 3, pp. 153-158, Jun. 2010.
- [4] M. S. Jang, S. H. Eom, P. S. Kim, and E. H. Lee, "Design and implementation of mobile terminal with multi bio-signal measurement for u-Healthcare system," *Journal of Rehabilitation Welfare Engineering & Assistive Technology (in Korean)*, vol. 7, no. 2, pp. 27-34, Dec. 2013.
- [5] E. J. Hwang, M. J. Park, D. Y. Kim, J. H. Yoo, and S. C. Kim, "The design of biosignal acquisition board using USB," *Information and Control Symposium (in Korean)*, pp. 135-136, Apr. 2008.
- [6] J. T. Devaraju, K. R. Suhas, H. K. Mohana, and V. A. Patil, "Wireless portable microcontroller based weather monitoring station," *Measurement*, vol. 76, pp. 189-200, Dec. 2015.
- [7] D. K. Fisher and H. Kebede, "A low-cost microcontroller-based system to monitor crop temperature and water status," *Computers and Electronics in Agriculture*, vol. 74, pp. 168-173, Oct. 2010.
- [8] H. S. Kim and C. G. Lee, "Development of the home control system base on USB," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 12, no. 4, pp. 405-410, Apr. 2006.
- [9] T. Y. Kim and Y. S. Lee, "Development of a remote control and monitoring system using raspberry Pi and its verification through HILS," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 23, no. 11, pp. 927-934, Nov. 2017.
- [10] A. Gani and M. J. E. Salami, "A LabVIEW based data acquisition system for vibration monitoring and analysis," *Student Conference on Research and Development, Shah Alam, Malaysiavol*, pp. 62-65, 2002.
- [11] K. Yeung and J. Huang, "Development of a remote-access laboratory: a dc motor control experiment," *Computers in Industry*, vol. 52, no. 3, pp. 305-311, Dec. 2003.
- [12] K. Iizuka, H. Uzuhashi, H. Kano, T. Endo, and K. Mohri, "Microcomputer control for sensorless brushless motor," *IEEE Trans. Industry Applications*, vol. IA-21, no. 4, May 1985.
- [13] S. H. Kim, *Motor control - DC, AC, BLDC*, 2nd Ed., BookDu, Seoul, 2017.
- [14] B. E. Jo and Y. S. Lee, "Development of a hardware-in-the-Loop simulator for BLDC motor driving using micro-controller," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 24, no. 12, pp. 1101-1110, Dec. 2018.
- [15] W. J. Jin, D. O. Kim, J. Y. Hwang, and G. H. Lee, "A study on torque control strategy under phase fault of permanent magnet synchronous motor drive system," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 25, no. 1, pp. 1-9, Jan. 2019

조 봉 언

제어 · 로봇 · 시스템학회 논문지, 제24권 제12호 참조

이 영 삼

제어 · 로봇 · 시스템학회 논문지, 제15권 제4호 참조