

## ARM920T의 Real-Time 제어적용을 위한 성능 측정 및 비교

김택기\*, 박상혁\*\*, 임재식\*\*\*, 이영일\*\*\*\*

\* MDS 테크놀로지 (TEL: 010-2378-6983, E-mail: cmos80@naver.com)

\*\* 서울산업대학교 제어계측공학과 (TEL: 010-9955-9516, E-mail: camio98@naver.com)

\*\*\* 서울산업대학교 나노아йти공학과 (TEL: 019-625-7882, E-mail: jaesiklim@gmail.com)

\*\*\*\* 서울산업대학교 제어계측공학과 (TEL: 011-9518-5370, E-mail: yilee@snut.ac.kr)

### The Performance Measurement and Comparison for Real-Time Control Application of ARM920T

Taek Ki Kim\*, Sang Hyuk Park\*\*, Jaesik Lim\*\*\*, Young Il Lee\*\*\*\*

\* MDS Technology

\*\* Department of Control Instrumentation Engineering, Seoul National University of Technology

\*\*\* Department of Nano Technology & Information Technology Engineering, Seoul National University of Technology

\*\*\*\* Department of Control Instrumentation Engineering, Seoul National University of Technology

**Abstract** - 본 논문에서는 ARM 프로세서의 제어기 구현 성능을 Firmware 환경 및 임베디드 OS 환경에서 실험, 조사하였다. 이산형 PID제어기를 S3C2410 칩을 사용하여 구현하였고 Cache 사용여부, 클럭주파수 등의 변화에 따라 테스트를 시행하면서 성능을 비교 분석하였다. WinCE 환경에서 실시간 제어기를 구현하기 위한 방법을 제안하고 ARM을 사용한 PID 제어기의 최대 샘플링 주파수를 실험을 통해서 밝혔다.

**Abstract** -In this paper, we investigate the ability of the ARM processor to implement an industrial controller with or without embedded operating system. Discrete-time PID controllers are implemented and tested under various settings e.g. cache on/off, different clock frequencies using S3C2410X chip. A method of real-time application of discrete-time PID controller in WinCE environment is proposed. Based on the test result, we provide the maximum sampling frequencies of PID controller using ARM processor.

## 1. 서 론

1980년대 중반이후 등장한 DSP(Digital Signal Processor)는 그때까지 소프트웨어적으로 불가능하다고 여겨졌던 연산을 현실화하는 계기를 마련하였다.[1] 제어 시스템에서 DSP가 많이 사용되는 이유는 DSP가 가지는 강력한 연산 능력과 속도 때문이다. 하지만 가격이 비싸고, 하드웨어 설계가 복잡하다는 단점을 가지고 있다.

최근 산업 전반에 걸쳐 두루 사용되고 있는 ARM 프로세서에서 DSP 만큼의 연산 능력을 기대하는 것은 어렵지만, 높은 동작 주파수와 캐시(cache) 및 배럴시프트 레지스터(barrel shift register)의 사용으로 상당히 높은 수준의 성능을 발휘할 수 있다.[2][3] 또한 OS가 포팅 되었을 경우, 멀티 태스크(multi task)를 지원하기 때문에 좀 더 쉽고 다양한 형태의 어플리케이션 구현이 가능하다는 장점을 가지고 있다.

이에 본 논문에서는 ARM 프로세서의 펌웨어(firmware) 환경 및 임베디드 OS 환경에서의 제어기 구현 능력에 대하여 분석해 보았다. 이를 위하여 ARM 920T 기반의 상용 ARM 프로세서를 선택하였고, 프로세서에 내장된 캐시를 이용하여 몇 가지 실험을 수행하였다. 먼저, 프로세서의 동작 주파수 별로 캐시를 사용하였을 때와 사용하지 않았을 때의 ARM 프로세서 성능을 측정 및 비교해 보았고, 동일한 조건 내에서 이산 PID 제어 프로그램을 적용하여 제어 루틴의 처리 성능을 측정 및 비교해 보았다. 또한 OS(Operating System) 기반 시스템에서 제어연산 처리 성능을 분석하기 위하여 임베디드(embedded) 시스템 OS인 WinCE(Ver 4.2)가 포팅 된 상태에서 동일한 이산 PID 제어 프로그램의 처리 성능을 측정 및 비교해 보았다.

이를 통하여 ARM 프로세서에서의 제어 루틴 처리 성능을 분석하였고, ARM 프로세서를 이용한 제어기 구현 시, 구현 가능한 샘플링(sampling) 주파수와 적용 가능한 제어시스템을 제시하였다.

## 2. Firmware 성능 분석

### 2.1 Hardware spec'

펌웨어 성능 분석을 위하여 S3C2410X 프로세서를 사용한 개발보드를

를 선택하였으며, 개발보드의 주요사양은 다음과 같다.

- CPU : S3C2410X
- NOR Flash : SST39VF160 × 1 (2MB)  
메모리 접근시간 320 nsec
- NAND Flash : K9F5608U0B × 1 (32MB)
- SDRAM : K4S281632D × 2 (64MB)  
메모리 접근시간 180 nsec

### 2.2 Cache 사용 효과

앞서도 밝혔듯이 ARM920T가 가지고 있는 캐시는 코어 내부에 존재하기 때문에 정확한 프로그램 실행 시간을 예측하기 어렵다. 따라서 본 논문에서는 S3C2410X가 가지고 있는 캐시의 성능을 측정하기 위하여 간단한 프로그램을 플래시에 다운로드한 후 캐시를 사용하였을 때와 사용하지 않았을 때의 프로그램 실행 시간을 측정하여 비교하는 방법을 사용하였다.

실험은 루프문을 1000번 실행시킬 때, 1000번 루프마다 GPIO 포트 로 출력을 내보내 실행시간을 동작 주파수 별로 오실로스코프를 이용하여 측정하였다. 표 2는 주파수 별로 측정된 실행시간을 비교한 것이다.

표 2. 동작 주파수별 루프 실행시간 비교

HCLK/HCLK/PCLK (MHz)	Non Cache (usec)	Cache Available (usec)
50 / 50 / 50	5600	320
100 / 50 / 50	5600	160
100 / 100 / 50	2800	80
200 / 100 / 50	2800	80

표 2의 비교 결과에서 보듯이 캐시를 사용하였을 때와 사용하지 않았을 때의 실행시간이 평균적으로 35배 정도의 차이가 있음을 알 수 있다. 최대 동작 주파수가 203 MHz인 고성능 프로세서지만, 이는 코어의 최대 동작 주파수이며, 캐시를 사용하지 않을 경우, 명령어 및 데이터를 메모리로부터 읽어 와야 하기 때문에, 프로그램의 실행시간은 길어지게 되고, 메모리의 속도에 따라 프로그램의 실행시간이 달라질 수 있다. 하지만, 캐시를 사용할 경우 메모리에 존재하게 되는 명령어 및 데이터를 캐시에 저장하여 사용하게 되고, 이는 메모리로부터가 아닌 ARM920T의 코어 내부에서 직접 명령어 및 데이터에 접근하게 되므로 상당한 성능 향상 효과를 볼 수 있음을 알 수 있다.

### 2.3 PID 제어 루틴 수행 능력

S3C2410X에서 PID 제어연산의 처리 성능을 측정 및 비교하기 위하여 간단한 이산 PID 제어 프로그램[5]을 작성하여 앞서 3.1절에서와 같은 실험을 수행하였다. 여기에서 플랜트 출력(measured variable)은 S3C2410X에 내장된 10-bit A/D converter를 사용하여 입력을 받았고, A/D 변환 주파수는 1 MHz로 설정하여 사용하였다. 그리고 설정치(set point)는 상수로 두고 사용하였고, 제어기 출력은 실제 기구물을 연결

하지 않고, 특정 메모리 영역을 지정하여 그곳에 저장하도록 하였다. 그리고 타이머 인터럽트를 이용하여, 매 인터럽트 발생 시 PID 제어 코드를 실행시키고, 서비스 루틴의 처음과 마지막에 GPIO 포트로 출력을 내보내, PID 코드의 실행시간을 동작 주파수 별로 오실로스코프를 이용하여 측정하였다.

표 3. 동작 주파수별 PID 루트 실행시간 비교

HCLK/HCLK/PCLK (MHz)	Non Cache (usec)	Cache Available (usec)
50 / 50 / 50	850	80
100 / 50 / 50	820	80
100 / 100 / 50	420-440	46-48
200 / 100 / 50	390-430	44-46

표 3의 비교 결과에서 보듯이 캐시를 사용하였을 때와 사용하지 않았을 때의 실행시간이 평균적으로 10배 정도의 차이가 있음을 알 수 있다. 앞서 실험의 비교에서와 같이 35배 이상의 성능 차이를 나타내지는 않았지만, 10배 정도의 비교가 가능한 성능 차이를 나타내었다. 앞서 실험에서의 비교와 이렇게 차이를 보이는 것은 프로그램 내에서의 A/D converter 접근 및 변환 시간과 메모리 접근 등의 이유로 좀 더 시간을 필요로 하기 때문으로 보인다.[2]

실험 결과에서 보듯이 펌웨어 환경에서 캐시를 사용할 경우, 이산 PID 제어 코드에 대하여 최대 46usec 정도의 성능을 나타내었으며, 이는 약 20KHz 정도의 샘플링 주파수를 가지고 동작할 수 있음을 의미한다. 또한 ARM 프로세서의 캐시를 사용하였을 경우 상당한 성능 향상 효과가 있음을 알 수 있다.

### 3. WinCE 기반 Real-Time 수행 능력 분석

실험을 위하여 S3C2410X 개발 보드에 임베디드 시스템 OS인 WinCE(Ver 4.2)를 포팅 하였으며, 동작 주파수는 프로세서의 최대 동작 주파수를 사용했다.

그리고 OS 기반 시스템에서 PID 제어기의 성능을 측정 및 비교하기 위하여 펌웨어 환경에서 실험 시 사용되었던 것과 동일한 이산 PID 제어 프로그램을 이용하여 실험을 수행하였다. 표 5는 측정된 PID 제어 프로그램의 제어 코드의 실행시간을 나타낸 것이며, 실험 결과는 WinCE 어플리케이션상의 타이머 쓰레드(thread) 베이스에서 실행된 것이다.

표 5. WinCE 타이머 쓰레드를 사용한 PID 코드

HCLK/HCLK/PCLK (MHz)	Operation Time (usec)
203 / 101.5 / 50.7	140

WinCE에서는 타이머 쓰레드를 위한 타이머를 msec 단위로만 제공하고 있다. 즉, 최대 샘플링 주파수가 1KHz임을 의미한다. 표 5의 실험결과 또한 그림 2에서 보는바와 같이 타이머 쓰레드를 이용하여 1msec 단위로 샘플링 주기를 설정하고, 샘플링 주기 내에서 PID 코드의 실행시간을 측정한 것이다.

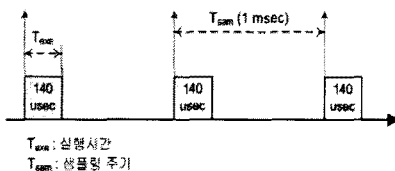


그림 2. WinCE의 타이머 쓰레드를 이용했을 경우의 PID 코드 실행시간과 샘플링 주기

실험 결과에서 보듯이 WinCE 환경에서 이산 PID 제어 코드의 실행시간은 약 140usec 정도였으며, 이는 약 7KHz 정도의 샘플링 주파수를 가질 수 있음을 알 수 있다.

제어기의 샘플링 주기를 1msec 이하로 구현하기 위해서는 그림 3과 같이 디바이스 드라이버(device driver)에서 nsec 단위로 설정이 가능한 ARM 자체의 타이머 인터럽트를 설정하여 인터럽트가 발생될 때마다, WinCE의 시스템 핸들러(handler)를 이용하여 PID 제어 코드를 실행하도록 하면 된다.

또한 PID 제어 코드를 ARM의 자체 타이머 인터럽트와 함께 디바이스

이스 드라이버에 구현하고, 별도의 쓰레드를 통해 샘플링 된 데이터를 핸들링 하도록 할 경우, 더 빠른 샘플링 주파수를 구현할 수 있다.

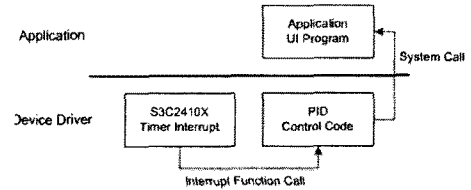


그림 3. 디바이스 드라이버를 사용한 PID 제어 코드 핸들링

이 경우 PID 제어 코드의 실행시간은 표 6의 결과와 같이 타이머 쓰레드를 사용하는 경우보다 짧게 나타났다.

표 6. 디바이스 드라이버를 사용한 PID 제어 코드 실행시간

HCLK/HCLK/PCLK (MHz)	Operation Time (usec)
203 / 101.5 / 50.7	42-44

하지만, WinCE 환경에서  $T_{sam}$  시간을  $T_{exe}$  시간과 동일하게 설정할 수는 없다. WinCE 환경에서는 TFT-LCD를 통해 UI(User Interface)를 제공하는 경우가 많고,  $T_{sam}$  시간을  $T_{exe}$  시간과 동일하게 설정할 경우, 시스템 자원을 독점하게 되고, 이는 WinCE의 다른 기능이 동작하지 못하게 되는 문제가 발생할 수 있기 때문이다. 이러한 문제를 해결하기 위해서는  $T_{sam}$  시간을  $T_{exe}$  시간보다 길게 설정하여야 한다.

실제 WinCE 환경에서 테스트하였을 때,  $T_{sam}$  시간이 최소 500usec 이상이 되어야 하는 것을 확인할 수 있었다.  $T_{sam}$  시간이 500usec 이하일 경우에는 WinCE의 다른 기능들이 동작하지 않거나 혹은 오동작을 일으키는 것을 확인할 수 있었다. 이는 실제 WinCE 환경에서는 2KHz 정도의 샘플링 주파수를 가질 수 있음을 의미한다고 생각할 수 있다.

### 4. 결 론

본 논문에서는 ARM 프로세서의 펌웨어 환경 및 OS 시스템 환경에서의 제어기 구현 능력에 대하여 분석해 보았다. 이렇게 측정된 ARM 프로세서에서의 이산 PID 제어기 샘플링 주파수를 통해 볼 때, 펌웨어 환경에서는 약 20KHz 이하의 샘플링 주파수를 가지는 제어 시스템을 ARM 프로세서를 이용하여 구현하는 것이 가능할 것으로 생각된다.

그리고 WinCE 환경에서는 프로그램의 구성에 따라 최소 1KHz에서부터 최대 약 2KHz까지의 샘플링 주파수를 가지는 제어 시스템을 구현하는 것이 가능할 것으로 생각된다. 그러나 앞서 4장에서 밝힌  $T_{sam}$  시간은 동작을 위한 최소한의 시간을 밝힌 것이므로 실제 시스템에서는 그 이상의 시간이 필요할 수도 있다. 하지만, WinCE라는 OS를 사용함으로써 프로그램의 유연성(flexibility)이 증가하기 때문에 제어 시스템의 요구 성능에 따라 다양한 형태의 제어 시스템을 구현하는 것이 가능할 것으로 생각된다.

### [참 고 문 헌]

- [1] 석출기, "DSP TMS320C3X의 특성 및 제어기술", 전력전자학회지 제9권 제2호, 2004.4
- [2] "ARM920T Technical Reference Manual(Rev 1)", ARM, 2000/2001
- [3] Andrew N. Sloss, Dominic Symes and Chris Wright, "ARM System Developer's Guide 1/e", Elsevier Inc., 2004
- [4] "S3C2410X 32-Bit RISC Microprocessor User's Manual Revision 1.2", Samsung Electronics, 2003
- [5] Karl J. Astrom, Bjorn Wittenmark, "Computer-Controlled Systems Theory and Design 3/e", Prentice Hall, 1997