

# μC/OS-II 기반의 S3C2440A 프로세서로의 실시간 시리얼 통신 모듈 포팅

박성주\*, 오삼권\*

\*호서대학교 컴퓨터공학부 컴퓨터공학전공

e-mail : [psjbear@hanmail.net](mailto:psjbear@hanmail.net) , [ohsk@office.hoseo.ac.kr](mailto:ohsk@office.hoseo.ac.kr)

## Porting a Real-time Serial Communication Module for S3C2440A Processor with μC/OS-II

Sung Joo Park\*, Sam Kweon Oh\*

\*Dept. of Computer Engineering, Hoseo University

### 요 약

시리얼 통신은 다양한 분야의 임베디드 시스템에서 사용되는 통신 시스템의 일종이다. μC/OS-II 는 실시간 임베디드 시스템에 사용되는 실시간 운영체제이다. S3C2440A 는 이러한 임베디드 시스템에 사용되는 마이크로 프로세서로 칩에 집적 된 UART 를 통하여 시리얼 통신을 지원한다. 이 논문에서는 시리얼 통신 중 μC/OS-II 기반의 실시간 시스템을 위한 시리얼 통신의 설계 와 S3C2440A 로의 포팅을 한다.

### 1. 서론

실시간 시스템에서는 실시간성 보장을 위해 OS 뿐만 아니라 주변장치 및 응용프로그램도 실시간성을 만족 하도록 설계 되고 구현 되어 한다. 다수의 시스템이 네트워크에 의해 연결되어 상호 연동되는 환경에서는 각 시스템 사이의 통신 또한 매우 중요한 요소이다. 이런 시스템 환경에서는 통신도 시스템의 상의 제약에 따라 시간적 제약조건을 만족하여야 한다.

본 논문에서는 실시간 시스템에 탑재되는 운영체제인 μC/OS-II 기반의 S3C2440A 마이크로프로세서를 위한 실시간 시리얼 통신 모듈을 포트 한다.

본 논문의 구성은 다음과 같다. 2 장은 μC/OS-II[1] 와 S3C2440A[2]에 대해 기술한다. 3 장은 통신 모듈의 디자인 및 포팅 방법에 대해 설명한다. 마지막으로 4 장은 결론 및 성능 개선을 위한 방향을 논한다.

### 2. 관련연구

#### 2.1. μC/OS-II

μC/OS-II 는 micrium 사에서 만들어진 실시간 운영체제로 다음과 같은 특징을 가진다.[1]

- > 소스코드 형태로 제공

- > 풍부한 이식성
- > 임베디드 시스템으로의 내장성
- > OS 의 기능을 조정 가능한 유연성
- > 우선순위 기반의 선점형 멀티태스킹 커널
- > 확정적인 실행시간
- > 태스크별 독립 스택
- > 다양한 OS 서비스
- > 중첩 가능 인터럽트
- > 입증된 안정성
- > 운영체제 차원의 주변장치 지원이 없는 단점

μC/OS-II 는 운영체제 상에서 별도의 I/O 장치에 대한 지원이 없기 때문에 시리얼 통신을 위한 모듈을 별도로 작성하여 탑재 해야 한다.

#### 2.1 μC/OS-II 의 Interrupt Service Routine

인터럽트는 비동기적 인 이벤트가 발생한 것을 CPU 에게 알려주는 하드웨어 메커니즘이다. ISR 이란 인터럽트 서비스 루틴의 약어로 시스템에 발생한 인터럽트를 처리해주는 프로그램 루틴을 말한다. μC/OS-II 에서의 ISR 구조는 다음과 같다.[1]

```

YourISR:
모든 CPU 레지스터 저장;
OSIntEnter() 호출 또는 OSIntNesting 직접증가;
If (OSIntNesting == 1) {
    OSTCBCur->OSTCBSkPtr = SP;
}
인터럽트를 발생시킨 장치 클리어;
인터럽트 재활성화(옵션);
실제 ISR 을 위한 코드 실행;
OSIntExit() 호출;
모든 CPU 레지스터 복구;
인터럽트 복귀 명령 실행;
    
```

μC/OS-II 에서는 위와 같은 방식으로 인터럽트가 처리 된다. 상기 표기된 부분 중 “실제 ISR 을 위한 코드 실행” 부분을 제외한 부분은 CPU 포트에 속하는 부분이다. 이 포트에서 작성되는 부분이 “실제 ISR 을 위한 코드 실행” 부분이다.

**2.2. S3C2440A 마이크로프로세서와 인터럽트 컨트롤러**

삼성전자(주)에서 제작된 32 비트 마이크로 프로세서인 S3C2440A 는 영국 ARM 사에서 디자인 된 32 비트 프로세서 코어 ARM920T 를 사용한 SoC(System On Chip)으로 초소형의 칩 하나에 여러 기능을 집적한 고성능 프로세서이다. 이 마이크로 프로세서는 최대 500Mhz 의 동작 속도로 강력한 처리 능력을 가지며 다양한 소형 임베디드 시스템에 사용되기 위한 저전력 소모를 장점으로 가진다.

S3C2440A 는 마이크로 프로세서에 집적된 주변장치를 제어하는데 사용되는 레지스터들을 메모리 특정 주소 번지에 대응 시키는 방식을 채택하고 있다. 따라서 주변장치를 제어하고자 하면 해당되는 레지스터의 메모리 주소에 I/O 작업을 수행하면 된다. 이 방식은 C 언어에 의한 손쉬운 I/O 프로그래밍을 가능하게 한다.

이 프로세서에 집적된 기능중의 하나인 인터럽트 컨트롤러는 60 개의 인터럽트 소스를 처리 할 수 있으며, 각 인터럽트는 고유의 우선순위에 따라 처리가 된다. 이 우선순위는 프로그램에 의해 제어가 가능하며 우선순위 로테이션 기능을 통해 시스템 동작 중에 인터럽트의 우선순위를 동적으로 변경할 수 있다.

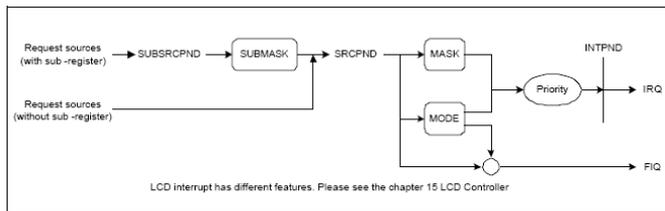


Figure 14-1. Interrupt Process Diagram

<그림 1> 인터럽트 처리 다이어그램 [2]

<그림 1>은 S3C2440A 에서의 인터럽트 처리 다이어그램으로 인터럽트 발생시부터 CPU 에 요청이 도달하기 까지의 과정이 나타나 있다. 일단 인터럽트가 발생하면 해당 소스의 펜딩(pending, 인터럽트 요청이 있음) 레지스터가 1 로 세트되고 이후 마스크 레지스터에 의해 인터럽트 발생여부를 결정한다. S3C2440A 에

서는 마스크 레지스터가 1 로 세트 되어 있을 경우 인터럽트가 발생하지 않는다. 반대로 마스크 레지스터의 해당 비트가 0 인 경우는 인터럽트가 발생할 수 있다.

Interrupt Controller					
SRCPND	0X4A000000	←	W	R/W	Interrupt request status
INTMOD	0X4A000004			W	Interrupt mode control
INTMSK	0X4A000008			R/W	Interrupt mask control
PRIORITY	0X4A00000C			W	IRQ priority control
INTPND	0X4A000010			R/W	Interrupt request status
INTOFFSET	0X4A000014			R	Interrupt request source offset
SUBSRCPND	0X4A000018			R/W	Sub source pending
INTSUBMSK	0X4A00001C			R/W	Interrupt sub mask

<표 1> 인터럽트 컨트롤러 레지스터 [2]

<표 1>은 인터럽트 컨트롤러 제어에 사용되는 레지스터들과 메모리상의 주소를 나타낸다. SRCPND, INTPND, SUBSRCPND 는 인터럽트 발생 여부를 알려주는 펜딩 레지스터이며 인터럽트 발생후 펜딩 레지스터를 직접 클리어 하지 않으면 그 상태를 그대로 유지한다. SRCMSK, INTMSK, INTSUBMSK 는 해당 인터럽트 소스에 대한 인터럽트 요청의 허가 여부를 결정하는 마스크 레지스터이다. INTMOD 레지스터는 인터럽트의 IRQ 혹은 FIQ(Fast Interrupt Request)의 동작 모드를 결정한다. PRIORITY 는 각 인터럽트 소스의 우선순위와 관련된 설정을 지시한다. INTOFFSET 은 INTPND 레지스터의 어떤 인터럽트 요청이 들어왔는지를 10 진 값으로 가진다.

**2.3 .S3C2440A 의 UART**

UART(Universal Asynchronous Receiver/Transmitter)는 컴퓨터의 바이트 데이터를 이진 비트열로 상호간에 변환 시켜주는 장치이다. S3C2440A 마이크로 프로세서는 총 3 개의 UART(채널 0 ~ 3)을 가지고 있다. 각 UART 는 IrDA(적외선 통신) 모드, DMA(Direct Memory Access), 인터럽트 동작모드를 지원하며, 각 UART 는 64 바이트의 FIFO(First-In First Out)방식의 버퍼를 가지고 있다. 그리고 하드웨어 흐름제어 기능과 최대 115.2Kbps 의 속도를 지원한다. 각 UART 에는 독립적으로 보오-레이트(baud-rate) 생성기를 가지고 있어 채널 별로 서로 다른 속도 및 설정을 가질 수 있다.

<표 2>는 S3C2440A 의 UART 제어를 위한 주요 레지스터 및 그에 해당하는 메모리 주소를 보여준다. UBRDIV 와 ULCON, UCON, UFCON, UMCON 레지스터 는 UART 를 제어하기 위한 설정 레지스터들이다. UTRSTAT, UERSTAT, UFSTAT, UMSTAT 은 현재 UART 의 상태를 나타내며 UTXH 와 URXH 는 송수신을 위한 버퍼 레지스터이다.

UART					
ULCON0	0x50000000	←	W	R/W	UART 0 line control
UCON0	0x50000004				UART 0 control
UFCON0	0x50000008				UART 0 FIFO control
UMCON0	0x5000000C				UART 0 modem control
UTRSTAT0	0x50000010			R	UART 0 Tx/Rx status
UERSTAT0	0x50000014				UART 0 Rx error status
UFSTAT0	0x50000018				UART 0 FIFO status
UMSTAT0	0x5000001C				UART 0 modem status
UTXH0	0x50000023	0x50000020	B	W	UART 0 transmission hold
URXH0	0x50000027	0x50000024		R	UART 0 receive buffer
UBRDIV0	0x50000028	←	W	R/W	UART 0 baud rate divisor
ULCON1	0x50004000				UART 1 line control

<표 2> UART 레지스터 [2]

2.4. (주)CLabSys LN2440SBC

(주)CLabSys 에서 제작한 소형 멀티미디어 시스템을 위한 개발용 보드로 삼성전자의 S3C2440A 프로세서, 64MB 의 SDRAM, 32MB 의 NAND Flash, 이더넷 컨트롤러, 2 차 저장장치인 CF/SD/MMC, USB 연결을 위한 USB 커넥터, 1 개의 UART RS-232C 포트 와 LCD 연결을 위한 커넥터를 제공한다. 이 보드의 마이크로 프로세서는 기본적으로 400MHz 의 속도로 동작한다.

이 보드는 포트 된 시리얼 통신 모듈을 구현하여 검증하기 위해 사용하였다.

3. 실시간 시리얼 통신 모듈

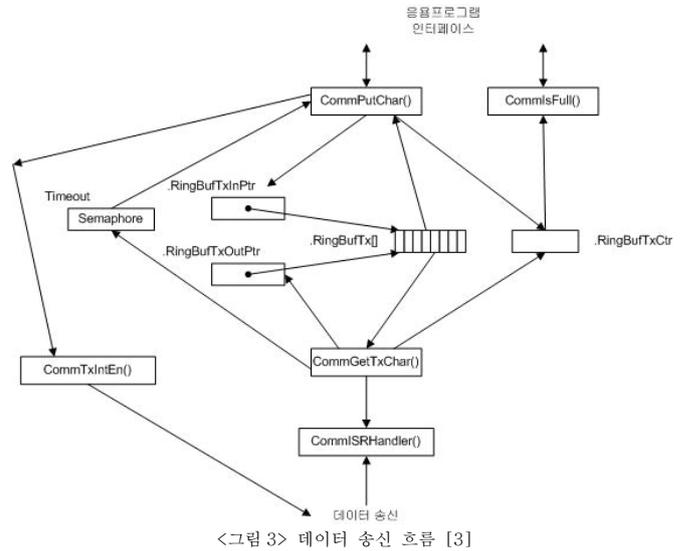
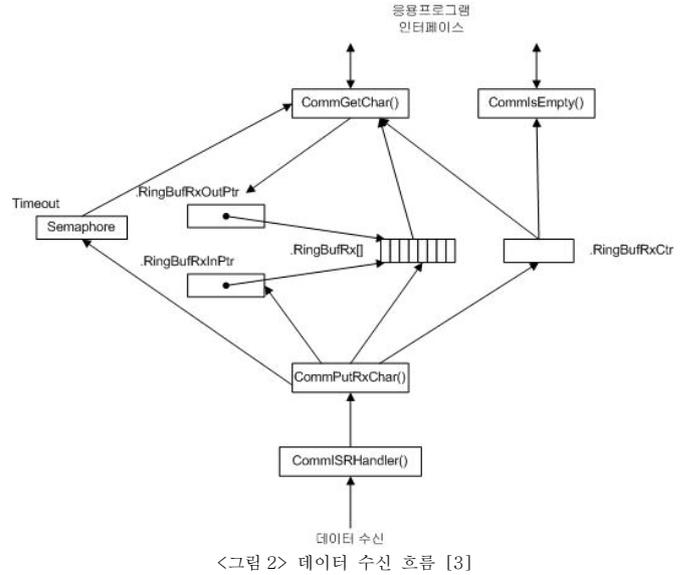
기본적인 모듈 디자인은 Jean J. Labrosse 의 Embedd Systems Building Blocks 의 시리얼 통신 모듈[3]을 기반으로 하였다.

송수신 모듈의 구조 및 동작은 <그림 3> 및 <그림 4>와 같다.

구조는 크게 두 가지 부분으로 구성 된다. 하나는 하드웨어에 비의존적인 응용프로그램으로의 인터페이스 함수이고, 다른 하나는 하드웨어에 의존적인 포트 부분이다. 두 부분 사이에 일시적인 데이터 저장을 위한 링버퍼(Rinf Buffer)는 전자의 부분에 속해 있다. 하드웨어에 무관한 부분은 바이트 단위의 송수신을 위한 함수들을 제공한다. 하드웨어에 의존적인 포트 부분은 하드웨어 제어를 위한 기능을 제공한다.

먼저 데이터 수신시에는 ISR 에 의해 내부의 링버퍼에 일괄적으로 저장을 하며 이 버퍼에 저장된 데이터를 태스크에서 꺼내어 가는 방식으로 동작한다. 송신시는 먼저 데이터를 링버퍼에 기록하고 송신 인터럽트를 활성화 시켜 UART 를 통해 데이터를 송신 한다. 이때 S3C2440A 의 UART 는 두 가지의 인터럽트 동작 방식을 지원하는데 레벨(Level)방식과 펄스(Pulse)방식 두 가지 이다. 이 포트에서는 레벨 방식을 사용 하였다. 펄스 방식에서는 송신 버퍼가 비어있다고 하여 송신 인터럽트가 지속적으로 발생 하지 않기에 레벨 방식을 취하여 송신 버퍼가 비어있는 동안은 항상 인터럽트가 발생 하여 인터럽트 활성화 즉시 인터럽트 요청이 발생 될 수 있도록 하는 것이다. 인터럽트가 발생하여 펜딩 된다 하더라도 마스크 레지스터에 의하여 인터럽트 활성화 여부를 결정 할 수 있으므로 불필요한 인터럽트 발생을 없앨 수 있다.

μC/OS-II 는 인터럽트의 중첩 기능을 지원하기 때문에 인터럽트 응답 속도의 향상은 피할 수 있으나 이런 인터럽트 중첩으로 인해 발생하는 하위 우선순위 인터럽트 처리 지연은 실시간 시스템에서는 문제의 소지가 있다. 따라서 이 포트에서는 인터럽트의 중첩을 허용하지 않도록 되어있다.



통신 포트는 2 개의 세마포어(Semaphore)를 가지고 있다. 세마포어는 수신용과 송신용 버퍼의 상태를 알려주기 위한 목적으로 존재한다. 또한 세마포어를 사용함으로써 태스크 레벨 코드는 수신 또는 송신되는 데이터를 최대한 빠르게 처리할 수 있으며 응용프로그램은 수신 버퍼에 데이터가 수신되는 것을 폴링(polling)할 필요가 없어진다. 마찬가지로 송신버퍼가 가득 차 있는 경우 응용프로그램은 서스펜드(Suspend)된다. 따라서 시리얼 포트에 데이터를 보낼 때 송신버퍼가 비어 있는지 확인할 필요는 없다.

세마포어를 사용함에 있어 제한시간(timeout)을 지정할 수 있기 때문에 송수신시의 최대 대기 시간을 조정 할 수 있다. 이러한 특성으로 인해 실시간 시스템에서의 시간제약을 만족 시킬 수 있다.

이 포트는 ARM 의 메모리 I/O 방식의 특성상 어셈블리어 사용 없이 C 언어 만으로도 장치 제어가 가능한 장점으로 인해 코드 또한 C 언어 만으로 작성 되었다.

데이터의 송수신은 바이트 단위로 이루어지며 CommGetChar() 함수와 CommPutChar() 함수를 통해

이루어 진다.

이 논문에서 포팅 한 함수는 다음과 같다.

(1) CommCfgPort()

보오-레이트, 데이터비트수, 패리티, 정지비트수의 설정을 하는 함수이다. 보오-레이트 설정은 다음 식에 의한 계산 결과를 UBRDIV 레지스터에 기록한다.

$$UBRDIV = (\text{int})(\text{UART Clock/baud rate} * 16) - 1 [2]$$

데이터비트수, 패리티, 정지비트수는 UCON, ULCON, UFCON, UMCON 레지스터에 기록하여 설정한다..[2]

(2) CommISRHandler()

실제 데이터의 송수신을 처리하는 ISR 이다.

ISR 에서의 처리 시간을 줄이기 위한 설계가 되어 있다. ISR 은 링버퍼에 수신 데이터를 쓰거나 링버퍼의 송신 데이터만을 송신함으로써 태스크에서 나머지 부분을 처리 하도록 한다. 태스크로의 데이터 이동은 응용프로그램이 호출한 함수 내에서 처리된다.

(3) CommRxFlush()

수신 버퍼를 비우는 함수다.

(4) CommRxIntEx()

CommRxIntDis()

수신 인터럽트를 조작하는 함수다.

(5) CommTxIntEn()

CommTxIntDis()

송신 인터럽트를 조작하는 함수다.

수신 및 송신 인터럽트는 이 함수들을 직접 조작하여 제어할 수 있다.

(6) CommSetIntVect()

CommRclIntVect()

새로운 ISR 을 지정한 위치에 설치하거나 원래의 ISR 로 복원하는 함수다.

(7) UCOS\_VectGet()

μC/OS-II 에서 현재 설정된 ISR 의 주소를 가져온다

이 함수는 통신 모듈이 아닌 μC/OS-II 의 ARM 포트 상에 구현 되었다.

#### 4. 결론 및 향후 개선 방향

이 포트를 통해 실시간 통신을 위한 본 모듈을 다른 시스템에도 채용할 수 있고, 모듈의 구조적 특징인 하드웨어 혹은 소프트웨어에 의존적인 부분이 분리된 구조로 다른 시리얼 통신 하드웨어로의 포팅이 용이하다.

하지만 현재의 응용프로그램 인터페이스 함수는 바이트 단위의 송수신을 하기 때문에 매번 바이트 송수신시마다 ISR-태스크간의 문맥전환이 발생하게 된다. 이는 시간에 지대한 영향을 받는 실시간 시스템에 있어서는 상당한 오버헤드를 유발할 수 있는 요소로 작용한다. 따라서 차후 연구에서는 응용프로그램에 의해 선택적으로 여러 바이트를 한번에 송수신 할 수 있도록 하여 시스템에 걸리는 오버헤드를 최소화 할 수 있는 모듈의 개발이 필요하다.

#### 참고문헌

- [1] Jean J. Labrosse, MicroC/OS-II 실시간 커널 2 판, 에이콘, 2005
- [2] Samsung, S3C2440A User's Manual, Samsung, 2004
- [3] Jean J. Labrosse, Embedded Systems Building Blocks, 에이콘, 2001