

---

# 블루투스 임베디드 시스템에 적용 가능한 직렬 포트 인터페이스 설계

문상국

목원대학교 전자공학과

Design of a Serial Port Interface Suitable for Bluetooth Embedded Systems

Sangook Moon

Mokwon University, Department of Electronic Engineering

E-mail : smoon@mokwon.ac.kr

## 요 약

본 연구에서는 임베디드 시스템, 특히 블루투스 베이스밴드에서 사용이 가능한 고속 직렬 포트 인터페이스를 설계하였다. 인터페이스는 ARM 프로세서를 응용할 수 있는 AMBA APB에 호환될 수 있도록 설계하였으며, 8비트 형태로 외부 디바이스와 코프로세서 간 데이터와 명령을 전송할 수 있다. 오류 정정을 위하여, CRC를 적용하였고 멀티미디어 카드를 위한 인터페이스도 제공하였다. 설계한 직렬 포트 인터페이스는 자동합성하여 P&R을 수행하였다. 결과물은 Altera FPGA로 구현하였으며 25MHz에서 정상동작하였다.

## ABSTRACT

In this contribution, we designed a serial port interface (SPI) suitable for embedded systems, especially for Bluetooth baseband. Proposed architecture is compatible for the APB bus in AMBA bus architecture. The 8-bit design of the SPI module is in charge of transferring the data and the instructions between the external devices and the coprocessors. We adopted the cyclic redundancy check method for the error correction. Also, we provided the interface for multimedia cards. The designed SPI module was automatically synthesized, placed, and routed. Implementation was performed through the Altera FPGA and well operated at 25MHz clock frequency.

## 키워드

임베디드시스템, 베이스밴드, GPIO

## 1. 서 론

블루투스와 같은 임베디드시스템에서 내부 코프로세서와 외부 디바이스와 데이터 전송이 필요한데, 이 때 필요한 것이 시리얼 포트 인터페이스이다. 이는 고속 데이터 전송을 위해 구현하는 것으로 임베디드시스템마다 용도와 목적에 따라 디자인이 다양하다. 대부분 마이크로프로세서에서 데이터라인이 따로 있는 시리얼 통신이라든지 spi통신을 사용하는 이유는 한라인에 비트를 백업할 수 있는 기능으로 마이크로프로세서의 포트 수를 줄이기 위함이며, 또한 제어하기도 편리하다는 점이 있다.

SPI는 외부 디바이스와의 통신을 위하여 필요한 클럭과 동기화된 고속 시리얼 포트이다. SPI는 바이트를 통신 단위로 사용하며 모든 명령, 응답과 데이터를 바이트 단위로 구성한다. 코프로세서와 외부 디바이스 간의 모든 통신은 코프로세서가 주도하며, 시리얼 데이터 통신은 특정한 선택 신호에 의해 시작되며 종료된다. 외부 디바이스와 코프로세서간 통신은 CRC (cyclic redundancy check)에 의해 보호되지만, 외부 디바이스는 non-protected 모드를 지원하여 신뢰할 수 있는 데이터를 보낼 수 있기 때문에 하드웨어나 펌웨어 방식의 CRC를 배제할 수 있어야 한다. 멀티미디어카드와의 통신도 지원해야 하는데, 이는 non-protected 모드에서

초기화되며 코프로세서는 소프트웨어적으로 CRC 제어가 가능하도록 구현되어야 한다.

본 논문에서는 고성능 SPI를 FPGA로 구현하여 칩 레벨의 포스트-레이아웃 레벨에서 시뮬레이션 하여 정상적인 동작을 검증하였다.

## II. 블루투스 베이스밴드

그림 1은 블루투스 베이스밴드의 일반적인 블록 다이어그램이다. 베이스밴드의 세부 블록은 아래와 같이 세분화된다.

- RF 인터페이스
- LMP 인터페이스
- 홉 선택 제어기
- ACL & SCO 링크 제어기
- HEC, FEC 제어기
- 액세스코드 상관기
- 데이터 화이트닝
- 암호화
- 인증
- 오디오 인터페이스
- 클럭 복원과 동기화 모듈
- 채널 제어기
- RX/TX 버퍼와 레지스터

그림 오른쪽에는 RF 모듈이 존재하여서 2.4 GHz ISM 밴드의 주파수를 변조하는 역할을 수행하고, 이로 받은 데이터들은 아래 부분의 타이밍 복원 회로에 의해 1MHz의 샘플링 형식에 맞는 데이터의 스트림으로 입력된다. 입력된 데이터들은 먼저 로우패스 필터를 통과하여 노이즈를 제거하면서 신호의 수행 사이클에 충실히 번역되어 베이스밴드 내로 전달된다. 이 때 64비트 블록 단위로 전송되는 데이터는 데이터의 싱크 검색기에 의해 신호의 문자열이 '1010' 또는 '0101' 인지를 감별하여 그것이 데이터 패킷의 시작인지를 인지한다. 성공적으로 인지되어 전달된 데이터는 일정한 블록 형태의 병렬 데이터로 변환되고 변환된 데이터는 헤더와 페이로드 부분으로 나뉘어 각각의 에러 정정 블록으로 전달된다. 헤더 에러 검출 블록에서는 스펙에 정의된 헤더의 에러를 검출하는 알고리즘을 이용하여 헤더의 에러를 검출하여 이후를 진행하고, 페이로드 부분은 화이트닝/디화이트닝 블록에 의해 스크램블링 되었던 데이터들이 의미있는 값을 가지면서 FEC (Forward Error Check)의 세가지 모드 (1/3, 2/3, 3/3)에 의하여 에러를 다

시 검출하고, [1][2] 간단한 에러 복구 메커니즘에 의해 복구된 데이터는 헤더와 페이로드로 분리되어 베이스밴드 내 메모리에 데이터 블록으로 각각 저장된다. 통신 패킷의 종류는 SCO (Synchronous Connection Oriented)와 ACL (Asynchronous Connection Oriented) 두 종류로 나뉜다. SCO 패킷은 슬롯을 할당하여 주로 음성통신과 같은 어플리케이션에 사용되고, ACL은 유연성을 가지는 데이터 패킷들을 전송하는데 주로 사용된다 [3].

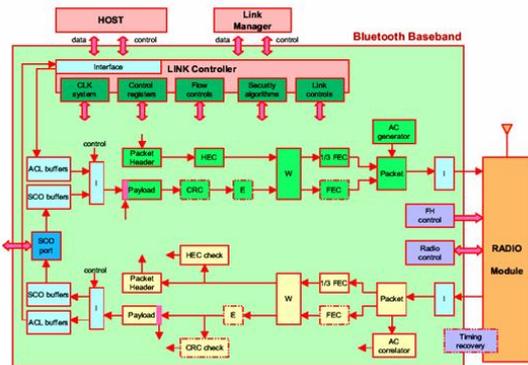


그림 1. 블루투스 베이스밴드 블록 다이어그램  
Fig. 1. Bluetooth baseband block diagram

## III. SPI 모듈

SPI와 블루투스 베이스밴드 내의 코프로세서와의 통신을 위한 신호는 4가지로 구성되며, 이는 표 1에 보이는 것과 같다.

표 1. SPI와 베이스밴드 간 통신 신호  
Fig. 2. Signal description between Baseband and SPI

Name	Type	Description
MOSI	O	Host to card data signal
MISO	I	Card to host data signal
SPICLK	O	Host to card clock signal
CS	O	Host to card chip select signal

SPI와 코프로세서 간 통신을 위해서는 제어 레지스터를 두어 레지스터에 설정된 명령 코드에 따라 하드웨어가 동작하는 방식으로 구현된다. SPI 제어기에는 SPI 제어 레지스터, 상태 레지스터, 교환을 위한 카운터, 8비트 TX 데이터 버퍼, 8비트 RX 데이터 버퍼, 테스트 레지스터, 그리고 GPIO와

의 핀 공유를 위한 신호를 저장하는 레지스터를 갖도록 설계하였고, 각 레지스터의 이름과 데이터 폭, 기본값을 표 2에 보인다.

표 2. SPI 제어 레지스터 요약  
Fig. 2. SPI control registers summary

Name	Width	R/W	Default
SPICR	8	R/W	0x20
SPISR	8	R	0x00
XCHCOUNTER	10	R/W	0x00
Txdatabuffer	8	W	0x00
Rxdatabuffer	8	R	0x00
Testregister1	8		0x00
Testregister2	8		0x00
ResetReg	8	R/W	0x00
TIC	8	R/W	0x00
SPI_ON	1	R/W	0x01

#### IV. SPI 구현과 동작

설계한 SPI 모듈의 동작은 다음과 같다.

코프로세서가 TX 버퍼에 데이터 시퀀스를 써 넣으면, FIFO의 내용은 TX 시프트 레지스터로 로드되며 한 바이트씩 시리얼 방식으로 시프트된다. TX FIFO의 모든 내용이 TX 시프트 레지스터로 전송되면 제어 모듈은 코프로세서로의 인터럽트를 요청하여 차후의 데이터 전송을 가능하도록 준비 작업을 한다. 직렬 입력 데이터는 RX 시프트 레지스터로 전송된다. 8비트가 모두 시프트되면, RX 시프트 레지스터의 내용은 RX FIFO에 복사된다. RX FIFO가 꽉 차게 되면, 제어 모듈은 코프로세서에 인터럽트를 요청하는 신호를 보내게 되며, 코프로세서는 RX FIFO의 내용을 읽어 인터럽트 서비스 루틴을 수행하게 된다. 타이밍과 제어 신호를 제어하는 신호는 SPICLK인데, 이 특수한 클럭 신호는 변동이 가능하도록 설계하였다.

SPI 제어 블록에서 사용하는 프로토콜은 명령과 응답 방식으로 설계하였다. 코프로세서가 SPI를 통해 멀티미디어카드로 특정한 명령을 요청하였을 경우 멀티미디어카드는 SPI를 통하여 코프로세서에 응답을 전달한다. 응답은 가변적인 길이를 가질 수 있다. 그림 2는 코프로세서 호스트와 멀티미디어 카드와의 명령과 응답을 통신하는 타이밍을 보인다. 그림 3에는 데이터 읽기, 그림 4에는 데이터 쓰기 타이밍 다이어그램을 각각 보인다. 약자로 사용된 H는 로직 1, L은 로직 0의 값을 나타내며, X

와 Z는 don't care, high-impedance를 각각 나타낸다.

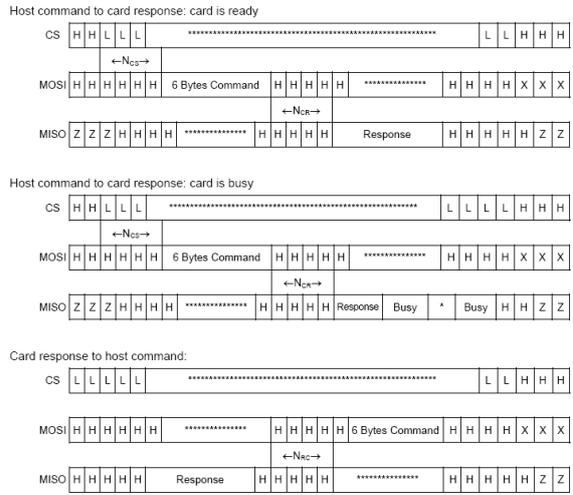


그림 2. SPI 명령-응답 타이밍도

Fig. 2. SPI command-response timing diagram

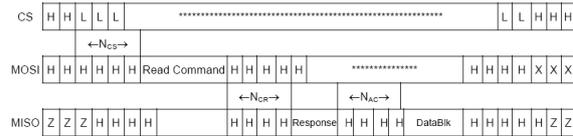


그림 3. SPI 데이터 읽기 타이밍도

Fig. 3. SPI data read timing diagram

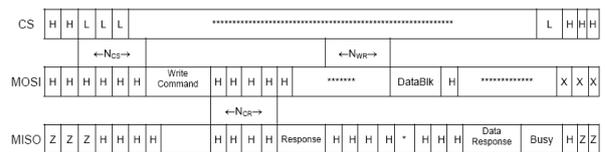


그림 4. SPI 데이터 쓰기 타이밍도

Fig. 4. SPI data write timing diagram

#### V. 결론

SPI와 코프로세서 간 통신을 위해서는 제어 레지스터를 두어 레지스터에 설정된 명령 코드에 따라 하드웨어가 동작하는 방식으로 구현된다. 본 논문에서 설계한 SPI 제어기에는 SPI 제어 레지스터, 상태 레지스터, 교환을 위한 카운터, 8비트 TX 데이터 버퍼, 8비트 RX 데이터 버퍼, 테스트 레지스터, 그리고 GPIO와의 핀 공유를 위한 신호를 저

장하는 레지스터를 갖도록 설계하였다.

본 연구에서는 임베디드 시스템, 특히 블루투스 베이스밴드에서 사용이 가능한 시리얼 포트 인터페이스를 설계하였다. 제안하는 구조는 AMBA 버스구조의 APB 저전력 버스에 호환되도록 설계하였다. 설계한 SPI 모듈은 자동합성하여 P&R을 수행하였다. 결과물은 Altera FPGA로 구현하였으며 25MHz에서 정상동작하였다.

### 참고문헌

- [1] Das, A. et al., "Adaptive link-level error recovery mechanisms in Bluetooth," Personal Wireless Communications, 2000 IEEE International Conference on , 17-20 Dec. 2000 pp. :85~89
- [2] Cheol-Hee Park et. al., "Design and implementation of error control algorithms for Bluetooth system: open-loop and closed-loop algorithms," Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on , 13-15 June 2000, pp. 302~303
- [3] <http://www.bluetooth.com>