

USB 통신을 이용한 89S51 Flash Memory Writer 대한 연구

이 덕 형, 이 영 일, 홍 선 기
호서대학교 정보제어공학과

A study of USB Communication using 89S51 Flash Memory Writer

Duck-Hyoung Lee, Young-il Lee, Sun-Ki Hong
Department of Information Control Engineering Hoseo University

Abstract - 기존의 89S51의 Flash Memory에 데이터를 저장하기 위해서 패러럴 포트를 사용하였다. 하지만 패러럴 포트를 이용한 방법은 많은 단점을 갖고 있는데, 그 중에 하나의 포트에 하나의 디바이스에 접속 할 수 없기 때문에 여러 디바이스를 접속하기 위해서는 포트 수를 증가시켜야 한다는 문제점이 있다. PC는 패러럴 포트를 1~2개 정도만 갖고 있어서 확장을 하기가 여의치 않다. 이에 따라 패러럴 포트의 단점을 보완하고자 한다. 이러한 문제를 해결하고 보완 할 수 있는 USB 통신을 이용해 Micro-Controller인 89S51에 내장된 Flash Memory에 데이터를 저장 하려고 한다.

며, address 378h의 bit0에서 bit7까지로 8개의 data bit로 구성되어 졌다. Status Address는 pin10에서 pin13, pin15이며, address 379h의 bit3에서 bit7까지 구성되어 졌다. Control address는 pin1, pin14, pin16, pin17이며, address 37ah의 bit0에서 bit3까지로 구성되어 졌다. 89S51에 Parallel Port를 연결하여 Flash Memory에 데이터를 저장하기 위해서는 SPI통신을 이용한다. SPI 인터페이스는 SCK, MOSI, MISO으로 구성되어 진다.

1. 서 론

기존의 개인용 컴퓨터의 주변 인터페이스로는 크게 serial port(RC-232C)와 parallel port를 들 수 있다. 이 장치들은 Micro-Controller의 메모리에 펌웨어 프로그램을 저장 하는데 사용되어 졌다. 어느 것이나 전송속도가 느리다는 점과 하나의 포트에 하나의 디바이스에 접속 할 수 없어서 여러 장치들을 연결하기 위해서는 새로운 포트를 추가시켜야 하는 문제점이 있었다. USB는 다른 인터페이스처럼 설치를 위해 Driver가 필요치 않고, 한 개의 주변기기의 설정을 변경하기 위해서 컴퓨터를 재부팅 할 필요도 없으며 시스템을 사용하는 중에 주변기기를 연결하거나 제거하여도 문제가 되지 않는다. 그러한 예로 PC에 조이스틱을 연결하고 게임을 한 후 다른 작업을 할 경우 게임을 하기위해 사용한 조이스틱을 제거한 후 그 자리에 키보드나 프린터를 꽂기만 하여도 다른 작업을 하는데 아무런 영향 없이 새로운 작업을 실행 할 수 있다.

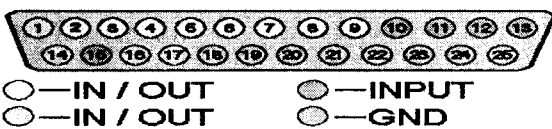
USB 통신을 이용해 Micro-controller인 89S51의 Flash Memory에 데이터를 저장시키기 위해서는 PC의 USB 포트와 Micro-Controller가 서로 통신을 하기위한 USB 포맷 구조를 맞춰주는 칩인 PDIUSB12라는 칩을 사용하였다. 또한 USB 전송방식 중에서 데이터 손실이 적은 Bulk방식을 사용하였고, Visual C++을 이용해서 89S51의 Flash Memory에 데이터를 저장 및 읽어 올 수 있도록 프로그래밍 하였다.

2. 본 론

2.1 Parallel Port를 이용한 방법

현재의 컴퓨터의 인터페이스는 모니터와 키보드 그리고 데이터 저장을 위한 디스크 드라이브, 프린터 등 각종 입출력 장치들로 구성되어 있다. 이 장치들은 서로 유기적으로 연결됨으로써 DATA를 전송 및 송신할 수 있다. 그러나 이렇게 데이터를 전송 및 송신을 위해서는 각 장치들은 내부적 특성이 달라서 서로 간에 일정한 약속이 전제되어야만 하는데 이것을 포맷 구조라고 한다. 데이터를 서로 원활하게 전송 및 송신하기위해서 연결시켜주는 부분이 반드시 필요하게 된다.

이러한 부분을 인터페이스라고 부르는데 이에는 1개의 전송로를 이용하여 순차적으로 전송하는 직렬 인터페이스와 여러 개의 전송로를 이용하여 동시에 전송하는 병렬 인터페이스가 있다. 가장 일반적인 직렬 인터페이스는 RS-232C 인터페이스인데 이는 미국 전자공업협회(Electronic Industries Association, EIA)가 규정한 직렬 데이터 규격으로서 공중통신회선을 이용하여 데이터를 전송 및 송신 할 수 있도록 되어있다. 모든 컴퓨터에는 하나 이상의 프린터 포트가 있을 것이다. 이것이 바로 PARALLEL PORT인데 이 포트는 일반적으로 프린터 전용포트로 사용되므로 흔히들 프린터 포트라고 부르고 있다. Parallel Port의 구조에 대해서 알아보면 [그림 1]과 같다.



<그림 1> Parallel Port의 구조

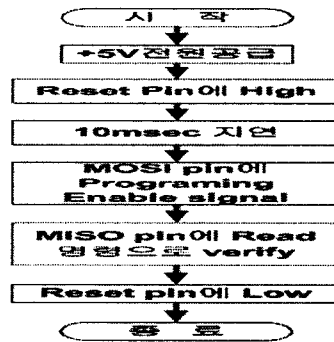
Parallel Port의 Data Address는 PC에서 pin2에서 pin9까지 할당되어 있으

Table 11. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	101x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx xxxx (Address)	xxxx xxxx (Address)	xxxx xxxx (Data)	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx xxxx (Address)	xxxx xxxx (Address)	xxxx xxxx (Data)	Write data to Program memory in the byte mode
Write Lock Bits**	1010 1100	1110 00 00	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1)
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxxx 00 00	Read back current status of the lock bits (a programmed lock bit reads back as a '1')
Read Signature Bytes	0010 1000	xxxx xxxx (Address)	xxxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx xxxx (Address)	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxxx xxxx (Address)	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

<그림 2> 89S51의 Serial 명령어

89S51의 Flash Memory에 데이터를 저장하려면, <그림 2>의 Serial Programming Instruction Set에 의해서 verify하여야 한다.



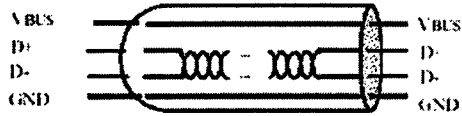
<그림 3> flash memory의 데이터 저장 과정

Flash Memory에 데이터를 저장하기 위해서는 <그림 3>의 Flash Memory의 데이터 저장 과정에 의해서 장치를 구성하고 프로그래밍 해야 한다. 하지만 PC에서는 Parallel Port가 하나 또는 둘 밖에 없어서 여러 장치를 연결 할 수 없다. 또 다른 장치를 Parallel Port에 연결하기 위해서는 기존에 사용하던 장치를 제거하고 새로운 장치를 Parallel Port에 연결하고 이 장치를 인식 하기위해서 Driver 파일을 설치해줘야만 한다. Driver 파일을 설치하고 난후 이 장치에 연결된 새로운 장치를 사용하기 위해서는 재부팅을 실행해야지만 새로운 장치를 사용할 수 있다. 이러한 문제점을 보완하기위해서 USB 통신을 이용한 방법으로 89S51의 flash memory에 데이터를 저장하는 방법을 구현하였다.

2.2 USB Communication을 이용한 방법

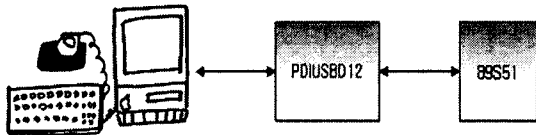
USB는 CTI (Computer Telephony Integration)산업의 성장을 위하여 상호 접속 (Interconnection)을 위한 간단한 방식을 제공하기 위하여 개발된 보편적 버스 규격이라고 할 수 있다. 진정한 플러그 앤 플레이 (PnP)를 위한 PC 주변장치의 Bus 규격으로서, 새로운 주변기기가 접속되었을 때 재부팅

이나 셋업 과정 없이 자동인식으로 최대 127개의 장치를 연결할 수 있을 뿐더러 데이터 전송속도도 빠르게 향상된 것이다. 현재의 버전 USB V1.1은 2개의 속도모드를 지원한다. 2개의 속도모드로는 12Mbit/s의 전송속도를 갖는 Full Speed Mode와 1.5Mbit/s의 전송속도를 갖는 Low Speed Mode이다. USB는 4선 케이블 방식을 사용하는데 2개의 선은 전원(5V와 GND)으로 사용되며, 다른 2개의 선은 차동 데이터로 사용된다.



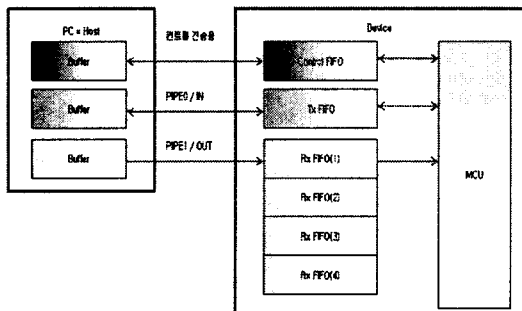
<그림 4> USB 4선 Cable의 구조

표준적인 USB 주변장치는 많은 Endpoint를 갖는다. 또한 USB는 PC에서 다른 Endpoint로 Pipe라는 데이터 전송법이 사용 되고 데이터 전송속도를 증가시키려면, 주변장치는 한 번에 하나 이상의 Pipe를 소비하기도 한다. PC와 주변장치의 사이에는 4개의 기본적인 전송안내 방식이 있다. Control 전송 방식은 주변장치로 전송제어신호의 송신에 대부분 사용된다. 높은 우선순위와 내장된 오류정정 보호가 있고 대부분 초기화 정보의 전송에 사용되며 저속도 전송의 범용으로 사용될 수 있다. 또한 모든 USB 장치는 제어전송을 지원한다. Bulk 전송 방식은 시간독립적인 저장장치의 대량의 데이터 전송에 대부분 사용되고 친숙한 프린터, 디스크 드라이버 등에 쓰이고 버스에서 낮은 우선순위를 가진다. Interrupt 전송 방식은 즉시 혹은 주기적으로 PC에 데이터를 보내야 하는 마우스나 키보드 같은 저속의 주변장치에 대부분 사용된다. Isochronous 전송 방식은 확정된 데이터의 전송이 많은 주변장치(예: 사운드카드)에 이용되며 에러 보정이 포함되지 않고 시스템은 약간의 데이터를 잃을 수 있다.



<그림 5> PC에서 89S51로 데이터 전송 방법

USB에서 89S51로 데이터를 빠른 속도로 전송하기 위해서 Full Speed 모드로 12Mbit/s로 전송하도록 하였고 데이터를 잃지 않고 안전성을 고려하여 Bulk 전송 방식을 채택하였다. USB에서는 Parallel Port와는 달리 PDIUSB12라는 USB 포맷구조를 맞춰주는 IC를 부가적으로 더 연결해줘야 한다. PC에서 89S51로 데이터를 전송하는 과정은 <그림 5>와 같은 형태로 전송된다.

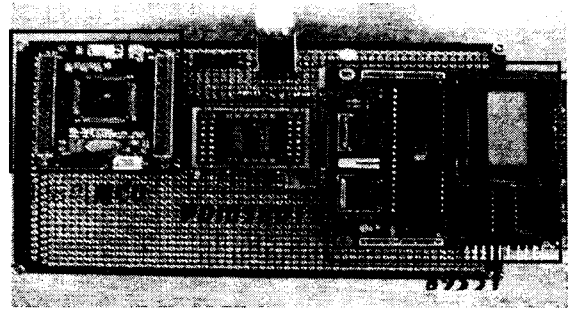


<그림 6> USB Data 전송 구조

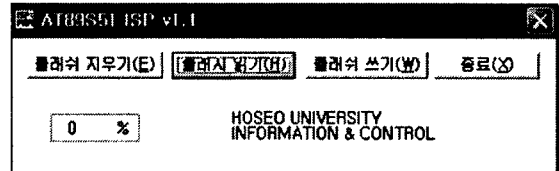
USB가 데이터를 89S51에 전송 할 때는 <그림 6>와 같은 구조로 보내어지게 되는데 데이터 전송은 항상 PC(Host)가 89S51(Device)로 처리하는 형태로 이루어진다.

3. 결 과

USB의 포맷 구조를 맞춰주는 PDIUSB12라는 IC를 사용하기 위해서는 별도의 Micro-Controller로 PDIUSB12가 동작할 수 있도록 핸들링이 필요하다. <그림 7>은 전체적인 회로의 사진이다.



<그림 7> 89S51 flash memory Writer 회로



<그림 8> 89S51의 어플리케이션의 구조

89S51로 데이터를 전송하기위해서 Visual C++에서 프로그래밍 하였고, 만들어지는 품의 구조는 <그림 8>와 같은 구조로 되었으며 Flash Memory에 데이터를 지우기, 읽기, 쓰기위하여 각각의 아이콘을 만들어 주었고, 현재 진행 상태를 표시하기위해서 상태 표시 또한 만들어서 표시하였다.

4. 결 론

Parallel Port를 이용한 방법과 USB 통신을 이용한 방법 2가지로 89S51의 Flash Memory에 데이터를 전송하는 것을 구현해 보았는데, Parallel Port를 이용하는 경우에는 PC에서는 Parallel Port가 하나 또는 둘 밖에 없어서 여러 장치를 연결 할 수 없다는 점과 또 다른 장치를 Parallel Port에 연결하기 위해서는 기존에 사용하던 장치를 제거하고 새로운 장치를 바로 사용할 수 없는 단점이 있었고, USB 통신을 이용하여 데이터를 저장하는 경우에는 12Mbit/s라는 매우 빠른 속도로 데이터를 보낼 수 있었으며, 또한 USB 전송 방식 중에 하나인 Bulk 전송 방식으로 데이터를 전송케 하여 데이터의 안전성 또한 보장되었다.

[참 고 문 헌]

- [1] 양승현, 김기순, "마이크로프로세서 설계 및 제작 8051", 엔트미디어, 2003
- [2] 양승현, 이석원, "비주얼 C++을 이용한 프린트포트 제어", 동일출판사, 2002
- [3] Philips Semiconductors, "PDIUSB12", 2001
- [4] USB.org, "Universal Serial Bus Specification Revision 1.1", 1998
- [5] Atmel, "AT89S51", 2003