

## 차세대 미디어 플레이어 위한 프로토콜 만들기- MTP (Media Transfer Protocol)

### 소개

By (Steve Kolokowsky, Cypress Semiconductor Corp. and Trevor Davis)

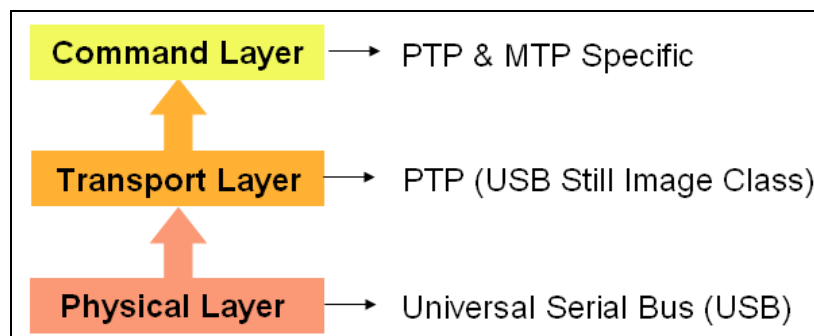
보안 디지털 콘텐츠의 이동은 그리 간단한 문제가 아니다. 사용자로부터 근본적으로 복잡함을 숨기는 기술 창출이 훨씬 어렵다. 디지털 오디오 및 비디오 콘텐츠 창출과 유통에 관여하고 있는 기업들은 현재 이러한 과제에 직면하고 있다. 차세대 핸드헬드 디바이스상에 보안기능과 기능이 풍부한 유저 인터페이스를 개발하려는 노력과 함께 마이크로소프트는 MTP (Media Transfer Protocol)이라는 하나의 새로운 프로토콜을 제안했다.

라이선스를 획득한 혹은 구입한 오디오 또는 비디오 콘텐츠의 보호권인 Digital Rights Management (DRM)은 핸드헬드 소비자 시장에서 아주 중요한 것이다. 디지털 콘텐츠 공급업체들은 영화와 같은 디지털 자료 발매를 여전히 꺼릴 것이다. 그러나, 올바른 프로토콜 보호와 함께 콘텐츠 공급업체들은 디지털 형식에서 가능한 그들의 지적 보호권을 편안히 만들 수 있을 것이다. 마이크로소프트는 MTP 가 바로 그런 프로토콜이라고 확신하고 있다.

### MTP 상세 내용

현대의 대부분의 프로토콜 스택들과 같이 MTP 는 레이어의 형식으로 묘사될 수 있으며, 다음 3 가지로 MTP 프로토콜을 세분화 했다; 1) Physical 2) Transport 3) Command. 이들 3 가지는 OSI 모델의 첫 3 가지 레이어와 동일한 것은 아니지만, 개념은 동일한 것이다. 흥미롭게도, physical 프로토콜은 스펙에서는 전혀 언급되지 않지만, 이것은 USB 일 것이다. MTP 를 위해 이용된 데이터 전송 레이어는 Picture Transfer Protocol (PTP) 스펙이다. 대부분의 commands 또한 PTP 의 확장 능력을 통해 확장된 PTP commands 이다.

그림 1.3 가지 MTP 프로토콜



## 1) Physical Layer

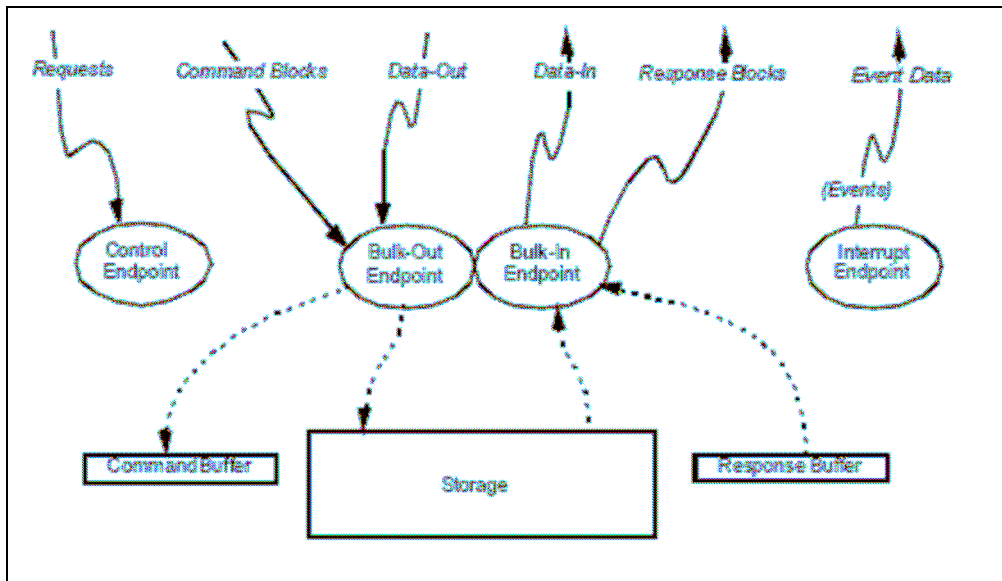
Note : PTP 와 MTP 가 physical layer 에서 동일한 관계로 이곳에서는 PTP 와 MTP 가 상호 교체할 수 있게 사용되었다. 명확히 하기 위해 이 섹션을 통해 MTP 를 사용할 것이다.

USB 디바이스는 엔드포인트(endpoint)를 통해 호스트와 커뮤니케이션 한다. USB 엔드포인트는 호스트와 함께 하나의 독자적인 커뮤니케이션 채널이다. 각 디바이스는 하나의 특수한 쌍방향 엔드포인트와 엔드포인트 0 로 레이블 된 컨트롤 엔드포인트를 가지는 것이 필요하다. 이 엔드포인트는 디바이스를 컨트롤 하고 디바이스의 특징을 결정하기 위해 호스트에 의해 사용된다. 다른 모든 엔드포인트는 독자적인 ordering 및 flow-control 을 가진 단방향 채널들이다. MTP 디바이스는 다음의 커뮤니케이션 채널을 제공하는 3 개의 엔드포인트를 가지고 있다.

- Data OUT of the host (bulk data endpoint)
- Data IN to the host (bulk In data endpoint)
- Commands (requests) OUT of the host (shares the control endpoint : EP 0)
- Events IN to the host (interrupt IN endpoints)

단어 IN 과 OUT 은 USB 에서 데이터 흐름의 방향을 나타내고 있다. USB 는 호스트 중심적인 프로토콜이다; IN 방향에서 어떠한 데이터의 흐름도 호스트로 흐르며, OUT 방향에서의 데이터 흐름은 디바이스로 간다.

그림 2. PTP / MTP 에서의 엔드포인트 사용 (source: PTP spec)

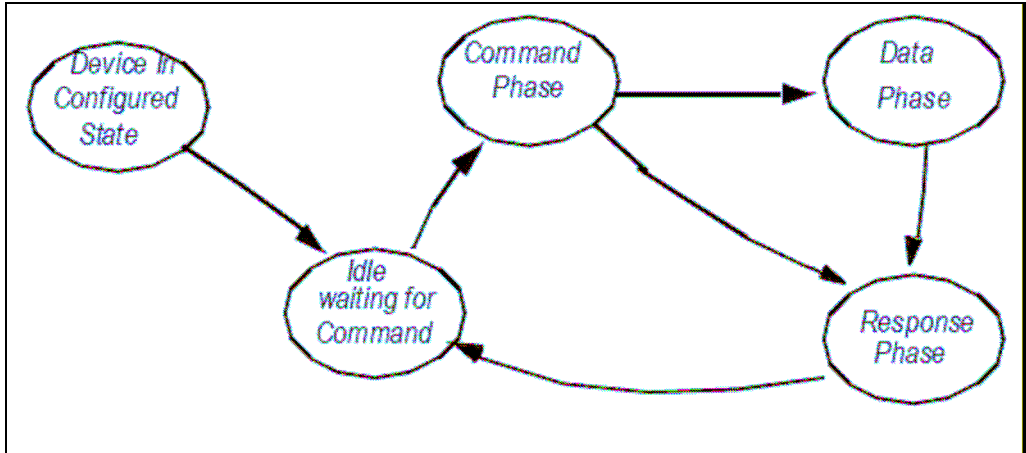


## 2) Transport Layer

MTP 의 데이터 전송 layer 는 PTP (USB Still Image Class)로부터 직접 취하게 된다. USB Still Image Class 는 commands 가 어떻게 PTP 상에 보내지는지, commands 에 어떻게 반응하는지, 전송을 어떻게 멈추는지, commands 가 어떻게 포맷되는지 등을 규명한다. 이 스펙에서는 다음 섹션에서 기술될 예정인 commands 를 기술하지 않는다.

USB transport state machine 은 command 단계, data 단계, 그리고 response 단계 등 3 가지 주요 단계를 가진다. 호스트는 항상 데이터 파이프를 통해 command 단계를 시작한다. Command 단계는 몇 가지 패킷을 (특히 이전의 USB 1.1 디바이스상에서) 지속할 수 있지만 규정된 모든 commands 는 64 bytes 이하이다.

그림 3: Data transport state machine (source: PTP spec)



Command 단계 동안, 호스트는 디바이스에 single command 를 보낼 것이다. 이들 commands 는 아래에서 보는 것처럼 항상 고정된 포맷을 가진 컨테이너에 보내진다.

Table 1 : Container structure (source PTP spec)

Byte Offset	Length (Bytes)	Field Name	Description
0	4	Container Length	이 field는 컨테이너에 많은 bytes를 비서명 정수로 암호화. Still Image Capture Device는 컨테이너의 크기를 결정하기 위해 이 field 이용
4	2	Container Type	이 field는 컨테이너의 타입 묘사: 0 <i>undefined</i> 1 <i>Command Block</i> 2 <i>Data Block</i> 3 <i>Response Block</i> 4 <i>Event Block</i>
6	2	Code	이 field는 PIMA 15740 OperationCode, Response Code, 혹은 EventCode 포함. Data Block은 Command Block으로 부터 OperationCode를 사용할 것이다
8	4	TransactionID	PIMA15740 operation 의 모든 단계와 관련한 host generated number
12	Command specific	Payload	이 field의 컨텐츠는 spec을 확장할 수 있기 때문에 operation 및 PIMA15740 operation 단계에 달려있다

만일 command 가 data 단계를 가진다면, IN 혹은 OUT data 단계는 command 단계를 따른다. 데이터는 또한 Command Blocks 을 위한 Container Type 1 대신 Container Type 2 와 함께 컨테이너에 보내진다. 마지막으로 디바이스는 command 의 결과를 나타내면서 response 로 보낸다. Response 코드는 command specific 이다.

### 3) Command

PTP 와 MTP 는 디바이스상에서 대부분의 것들을 Object 로 처리한다. 대부분의 commands 는 Objects 를 다루는 디바이스의 일상생활에 이용된다.

Object command	Purpose
GetNumObjects	Returns a count of the number of objects (songs, files, video clips) on the device
GetObjectHandles	Returns pointers to some or all of the objects
GetObjectInfo	Returns object type, size, format
GetObject	Returns object dataset
GetThumb	Returns summary (thumbnail, audio sample, video clip or frame)
DeleteObject	Bam! It's gone.
SendObjectInfo	Prepare to write object. Allocates space for the object.
SendObject	Host sends data to the device
SetObjectProtection	Write protect enable / disable

### 샘플 데이터

아래 CATC 추적은 호스트와 디바이스 간의 초기 커뮤니케이션의 일부를 보여주고 있다.

그림 4: MTP Data Traffic 의 예

Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	Param 1
35	OUT	2	Image	16	Command	OpenSession	0	1
Transaction	ENDP	Data						Data
230776	2	10 00 00 00 01 00 02 10 00 00 00 00 01 00 00						
Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	
36	IN	6	Image	12	Response	OK	0	
Transaction	ENDP	Data						
265064	6	0C 00 00 00 03 00 01 20 00 00 00 00						
Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	
37	OUT	2	Image	12	Command	GetStorageIDs	1	
Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	StorageID
38	IN	6	Image	20	Data	GetStorageIDs	1	65537
Transaction	ENDP	Data						
268204	6	0: 14 00 00 00 02 00 04 10 01 00 00 00 01 00 00 00 16: 01 00 01 00						
Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	
39	IN	6	Image	12	Response	OK	1	
Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	Param 1
40	OUT	2	Image	16	Command	GetStorageInfo	2	1
Transfer	Bulk	ENDP	Still	ConLen	ConType	Code	TransID	StorageType
41	IN	6	Image	78	Data	GetStorageInfo	2	Fixed RAM
AccCapab	MaxCapacity	FreeSpaceInBytes	FreeSpaceInImages	StorageDesc	Wc			
Read-Write	0xF8EC0006	0xF379AE326	Not implemented	PMC HardDisk				

호스트는 OUT 로 표시된 모든 전송을 시작하며, 디바이스는 모든 IN 트래픽을 보낸다. Transfers 는 Transactions 에 있는 데이터를 해독한다. 예를 들어, transfer 35 의 데이터는 transaction 230776 에 포함되어 있다.

Transfer 35 여기서 우리는 컨테이너가 실질적으로 이용되었다는 것을 볼 수 있다. Length field (0 x 10 / 16 decimal)는 Type (Command = 1)과 command code (OpenSession)에 이어 LSB 에서 MSB 로 전송되었다. MTP 전송 ID 는 접속에서 최초의 전송이라는 것을 나타내는 0 이다.

Transfer 36 이것은 호스트에 전송이 성공적으로 완료되었다는 것을 나타내면서 최초의 전송을 마감하고 있다. (Code = OK)

Transfer 37 호스트는 디바이스 (GetStorageIDs)에 있는 스토리지 요소들의 리스트를 요구하고 있다. 예를 들어, 확장 카드를 가진 Palm Pilot 는 아마도 메인 메모리와 확장 카드 등 2 개의 요소를 가지고 있을 것이다.

Transfer 38 이것은 Data Container 의 첫 사례이다. 디바이스는 0x1 (logical)과 0x1(physical) 로 매겨진 싱글 스토리지 요소를 가진 것에 반응한다. 이 command 는 다음 table 에서 완전히 해석되어 있다.

Transfer 39 OK code 와 함께 Transaction 1 마감

Transfer 40 다음 MTP 전송작업 시작

Byte Offset	Length (Bytes)	Field Name	Value
0	4	Container Length	0x14 / 20 Decimal
4	2	Container Type	2 = Data
6	2	Code	0x1004 = GetStorageIDs
8	4	TransactionID	1 = This is the second transaction in the session
12		Payload	In this command, the payload is an array of storageIDs. PTP/MTP arrays always start with a 32-bit number representing the size of the array.
12	4	NumElements	1 = Array size is one element
16	4	StorageID	0x00010001 = Physical storage element 1, logical storage element 1.

#### MTP vs PTP 의 차이점 이해

우리는 MTP 와 PTP 의 공통 부분을 기술하는데 많은 시간을 소비했지만, MTP 는 여러 측면에서 PTP 를 향상시킨다. 프로토콜의 일부 사소한 향상된 부분은 기능의 큰 변화를 가능하게 한다.



MTP 는 새로운 방식에서 objects 의 조합을 규명하는 새로운 objects 를 추가하며, 이는 어떠한 MP3 나 미디어 플레이어에도 필수적인 playlist 지원을 가능하게 해 준다. MTP 는 또한 캘린더, contact 및 그룹 objects 를 추가함으로써 Palm-type 디바이스 지원 기능을 추가한다. 물론, MTP 는 비디오 object 타입을 위한 지원 기능도 추가하고 있다.

MTP 는 Digital Rights Management (DRM) 상태, URL 을 포함한 DRM 특허를 추가함으로써 실행 기능을 보다 쉽게 지원할 수 있도록 만들어 준다.

MTP 는 또한 데이터 취급을 보다 손쉽게 만들어 주기 위한 확장기능을 더하고 있다. MTP 는 대형 아이템 (>4GB)의 전송을 허용하기 위해 PTP 전송을 확장하며, 서로 다른 USB 패킷으로 세분화 된 컨테이너와 데이터를 허용한다. MTP 는 패킷을 더욱 효율적으로 진행하기 위해 싸이프레스의 EZ-USB FX2LP 와 EZ-USB SX2 제품군)을 포함한 USB 칩을 할당한다.

### **Basic MTP vs Enhanced MTP**

마이크로소프트는 Windows 와 함께 MTP 디바이스 사용 경험을 향상시키기 위해 Basic MTP 구조에 더하여 일련의 추가적인 최적화 작업을 추가했다. Enhanced MTP 의 목적은 호스트가 MTP 를 타이트 하게 통합할 수 있도록 한 것이다. 대부분의 enhanced MTP 기능은 multiple objects 의 조절을 한번에 가능하게 하기 위한 것이다. 이로써 sync performance 는 향상될 수 있다.

### **법적인 문제들**

만일 당신이 마이크로소프트 웹 사이트에서 MTP 스펙을 다운로드 한다면, 당신은 스펙의 첫 3 페이지가 실질적으로 EULA (End User License Agreement)라는 것을 발견하게 될 것이다. 어떠한 MTP 개발을 시작하기 전에 라이선스를 읽기를 권유한다. EULA 의 일부 하이라이트는 다음과 같다;

1. **Basic MTP is available without paying a license fee:** “... Microsoft grants you the following limited, non-exclusive, world-wide, royalty-free, non-assignable, nontransferable, non-sublicenseable license” (emphasis added)
2. **“Enhanced Initiator” functions are not covered by this license.** “This Agreement does not grant you a license to build Solutions that implement any of the “Enhanced Initiator” features or functions described in the Specification”
3. **You must implement the whole spec:** “Your Implementations as incorporated into your Solutions must implement the Specification in its entirety”

### **결론**

궁극적으로 또 다른 전송 사양을 만들기 위한 유일한 이유는 엔드-유저이다. 컨슈머 마켓은 보다 간편하고 기능이 풍부한 핸드헬드 디바이스를 만들기 위한 압력이 증가하고 있으며, 콘텐츠 생성 업계는 그들이 발매하는 디지털 콘텐츠를 적절히 보호하려는 압력이 업계에 늘어나고 있다. 양자를 모두 만족시키기 위한 노력으로 마이크로소프트는 Media Transfer Protocol 을 제안했다. 설계자들이 기본 기술을 이해하는 것이 그들의 고객들이 궁극적으로 누릴 이점들을 이해하기 위한

핵심이다. 따라서, 새로운 프로토콜을 학습하는 것이 올해 해야 할 가장 재미있는 일은 아닐지라도 소비자들은 분명 감사하게 생각할 것이다.

To go beyond the scope of this article and start implementing your own MTP device, you will need several specifications:	
<u>Universal Serial Bus Specification Revision 2.0</u> <a href="http://www.usb.org/developers/docs/">http://www.usb.org/developers/docs/</a>	This is the basic USB specification. It governs the connectors, cables, electrical characteristics and basic communication structure of USB.
<u>USB Still Image Capture Device Definition</u> <a href="http://www.usb.org/developers/docs/">http://www.usb.org/developers/docs/</a>	This spec defines the transport layer for USB.
<u>PIMA 15740:2000: Photography – Electronic still picture imaging - Picture Transfer Protocol (PTP)</u> <a href="http://www.pima.net">www.pima.net</a>	PIMA merged with the DIG (Digital Imaging Group) of i3a (International Imaging Industry Association) in early 2001. The <a href="http://www.pima.net">www.pima.net</a> website is no longer associated with the DIG or PIMA. The PTP specs are available for a fee from <a href="http://www.i3a.org">www.i3a.org</a> . The fee is waived for .edu and .gov individuals. This spec describes the basic command set.
<u>Media Transfer Protocol</u> <a href="http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwm/html/mtp_spec.asp">http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwm/html/mtp_spec.asp</a>	Describes the commands added to PTP to create MTP

**References**

Media Transfer Protocol – Manders & Rosenbloom, Microsoft Corp. WinHEC 2004  
All specs listed above



Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134-1709  
Phone: 408-943-2600  
Fax: 408-943-4730  
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™, Programmable System-on-Chip™, and PSoC Express™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.