Semiconductors

AN10034_2

ISP1261 USB OTG Bridge Controller and SEOC Protocol

Rev. 02.01 —30 March 2004 Application note

Document information

Info	Content
Keywords	ISP1261, seoc, software emulated otg controller, bridge
Abstract	The Universal Serial Bus (USB) was designed as an external bus system centered around a personal computer (PC.) In a standard USB system, there is only one USB host, and one or more USB peripherals and USB hubs. USB uses four standard transfer types—control, bulk, interrupt and isochronous—to communicate between the host and a peripheral. Based on the USB bulk transfer mechanism, the Philips USB On-The-Go (OTG) bridge controller ISP1261 uses a patent-pending software and hardware solution, which transforms virtually any USB peripheral into a full-fledged USB OTG dual-role controller. An OTG controller can assume the role of a USB host or a USB peripheral. This paper explains the solution – the Philips Software Emulated OTG Controller (SEOC).

PHILIPS

Revision history

Rev	Date	Description
2.0	March 2004	Removed mention of ISP1262; Updated term “interface device” to “peripheral”.
1.0	October 2003	First version

Contact information

For additional information, please visit: *http://www.semiconductors.philips.com/*
For sales office addresses, please send an email to: *sales.addresses@www.semiconductors.philips.com*

This is a legal agreement between you (either an individual or an entity) and Philips Semiconductors. By accepting this product, you indicate your agreement to the disclaimer specified as follows:

DISCLAIMER

PRODUCT IS DEEMED ACCEPTED BY RECIPIENT. THE PRODUCT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, PHILIPS SEMICONDUCTORS FURTHER DISCLAIMS ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANT ABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THE PRODUCT AND DOCUMENTATION REMAINS WITH THE RECIPIENT. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL PHILIPS SEMICONDUCTORS OR ITS SUPPLIERS BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, DIRECT, INDIRECT, SPECIAL, PUNITIVE, OR OTHER DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THIS AGREEMENT OR THE USE OF OR INABILITY TO USE THE PRODUCT, EVEN IF PHILIPS SEMICONDUCTORS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Contents

FIGURES

1. Introduction

Many mobile computing devices in the market such as the digital still camera (DSC), the mobile handset, the personal digital assistant (PDA) and the MP3 player have USB peripheral hardware built in. A mobile computing device uses the USB peripheral hardware to communicate with a PC to upload or download data, synchronize e-mail messages and contact information. Without built-in USB host hardware, a mobile computing device cannot communicate with other USB peripherals. ISP1261 allows USB host hardware to be added ***externally***, and therefore possibly upgrade numerous mobile computing devices from being peripheral-only to dual-role host and peripheral without any change to the existing hardware.

The following diagram illustrates a PDA with an ISP1261-equipped external dongle connecting to a USB peripheral, namely, a hard disk.

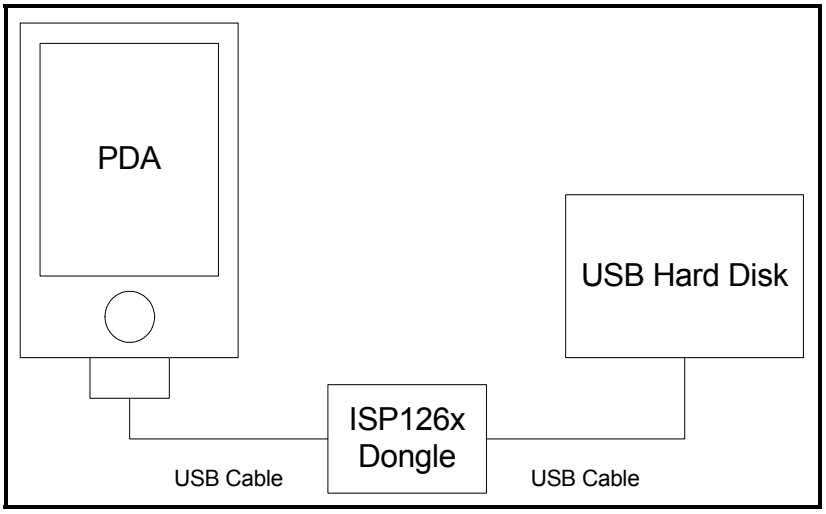

Figure 1-1: PDA with an ISP1261-equipped external dongle, connected to a USB hard disk.

The hardware scheme is very simple. The ISP1261 has two USB cable connection: one to the PDA and one to the USB hard disk. Both the PDA and the USB hard disk are strictly USB peripherals. But with the ISP1261, the PDA can assume the role of a USB host and gain access to the data stored in the USB hard disk! This is possible due to three key technologies:

- Philips Advanced Host Controller (PAHC)
- SEOC Protocol
- Virtual Hardware Abstraction Layer (VHAL) software

1.1. Key Technology 1: Philips Advanced Host Controller (PAHC)

PAHC is the core of the embedded USB host controller part of the Philips ISP1362. It is a transfer-based controller, with hardware responsible for most of the USB flow-control activities. The important features of the PAHC are:

- Auto-retry on Not-acknowledge (NAK)
- Start-of-frame (SOF) generation by hardware
- CRC generation and checking

- Toggle-bit tracking
- Error detection

1.2. Key Technology 2: SEOC Protocol

Software Emulated OTG Controller (SEOC) Protocol is a set of transfer methods based on the standard USB bulk transfer mechanism. This protocol allows a USB peripheral to access a set of hardware (registers and memory) that is located remotely though the USB connection. The following figure illustrates this structure:

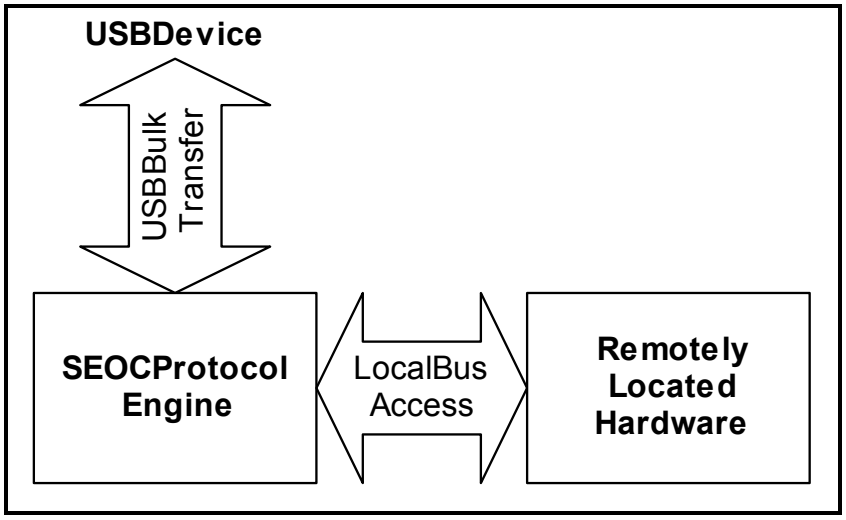

Figure 1-2: SEOC Protocol.

SEOC Protocol Engine consists of a PAHC and a Finite State Machine (FSM) executing the SEOC Protocol. The SEOC Protocol Engine sends an IN-Token to the USB peripheral continuously. When the USB peripheral needs to access the remote hardware, it replies with a DATA packet. The SEOC Protocol Engine examines this packet to find out the kind of access required, and proceeds to access the remote hardware accordingly. For 'READ' operations, the SEOC Protocol Engine sends an OUT-DATA containing the result of the reading operation to the USB peripheral.

1.3. Key Technology 3: Virtual Hardware Abstraction Layer (VHAL)

Virtual Hardware Abstraction Layer (VHAL) is a software layer required for a USB peripheral to take on the role of a USB host. The VHAL maps the remotely located hardware as virtual local hardware, using the SEOC protocol running on the USB peripheral hardware. The following diagram shows the VHAL providing a 'virtual connection' to the remote hardware using the existing USB peripheral hardware infrastructure. The remote hardware, in this case, is an OTG Controller.

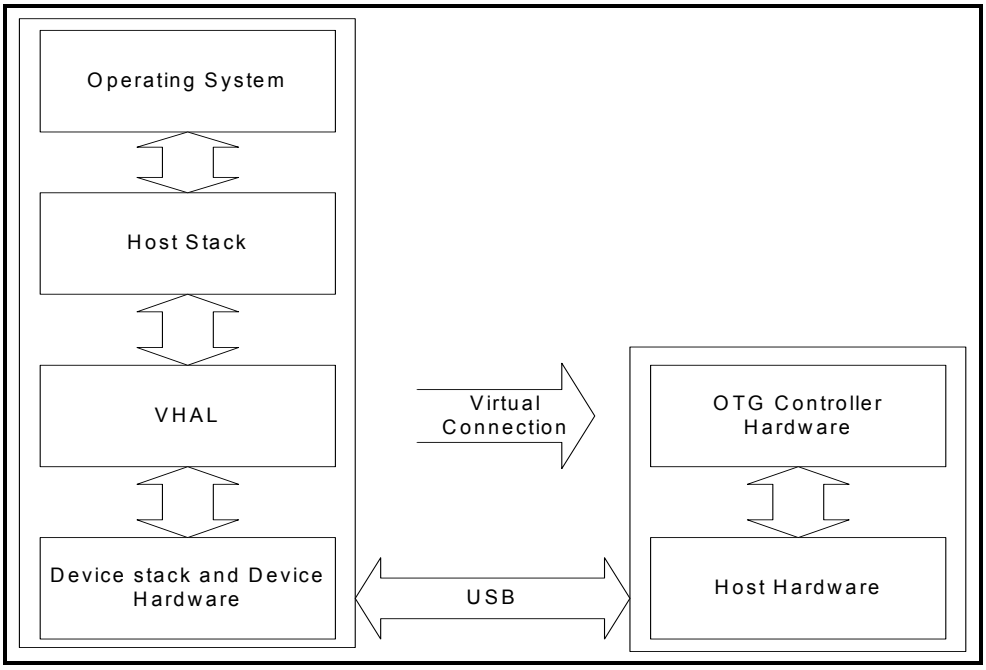

Figure 1-3: Virtual Hardware Abstraction Layer.

With the VHAL providing full access to the remotely located OTG hardware, the USB host stack in the pseudo host can function as if an OTG controller is built-in.

2. ISP1261

ISP1261 implements the SEOC Protocol entirely in dedicated hardware. A simplified block diagram of ISP1261 is shown as follows:

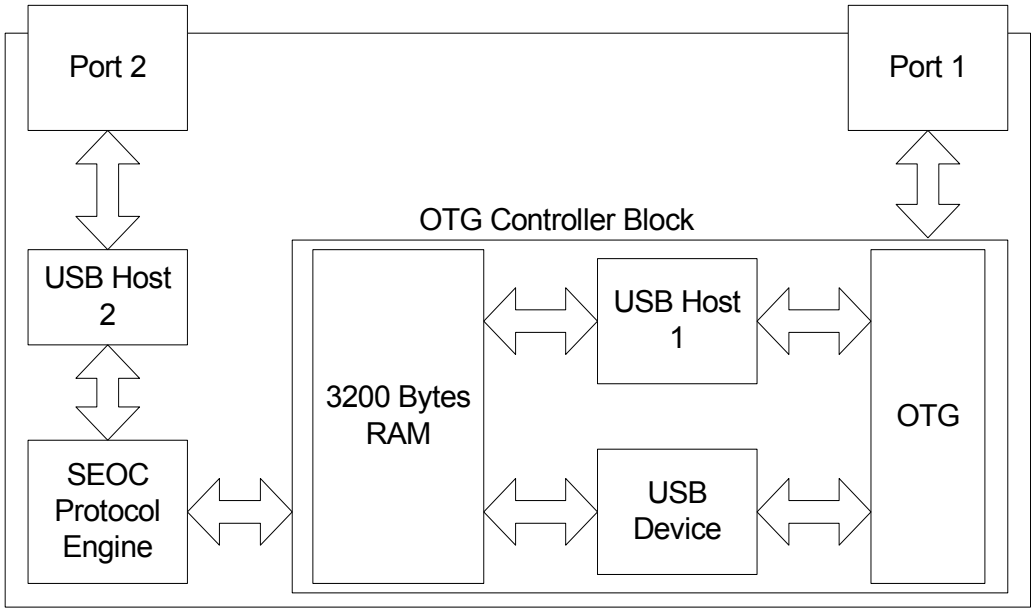

Figure 2-1: ISP1261: Simplified Block Diagram.

There is a complete OTG Controller block in the ISP1261. Connect Port 2 to a mobile computing device that already has a USB peripheral built in but that would now like to assume the role of a USB host. Let us call such a mobile computing device 'pseudo host'. Connect Port 1 to a USB peripheral that the pseudo host wants to access. USB Host 2, together with the SEOC Protocol Engine, forms the heart of the ISP1261. It performs the SEOC Protocol operations without the need of any firmware or user intervention, as all intelligence is built into the hardware.

3. Enumeration and Polling

ISP1261 SEOC Protocol Engine operates primarily in two states: Enumeration state and Polling state. On powering up, ISP1261 enters the Enumeration state. The operation during the Enumeration state is shown in the following flow chart:

The steps in enumeration can be summarized as

- USB Bus Reset
- Set Address (to 2)
- Get Vendor Command (SEOC Specific)
- Set Configuration

"Get Vendor Command" is a vendor-specific request. This request allows the pseudo host to confirm that the connected USB host is indeed ISP1261. The pseudo host uses a reply to the "Get Vendor Command" to inform the ISP1261 of the endpoints it has allocated for SEOC Protocol operation. During the Enumeration state, any error will cause the ISP1261 to return to the reset state and repeat the enumeration process.

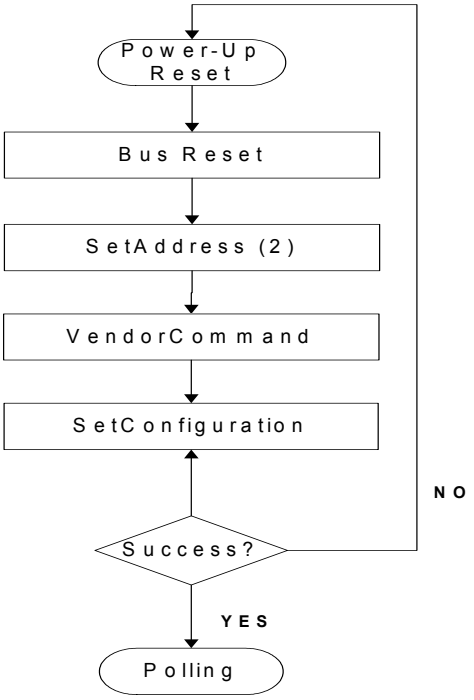

Figure 3-1: Enumeration State flowchart.

If the enumeration is successful, ISP1261 enters the Polling state. The following is a flow chart of the Polling State:

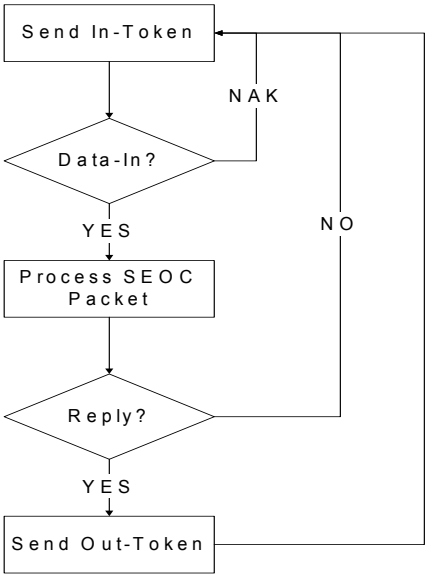

Figure 3-2: Polling State flowchart.

As shown in the flow chart, the ISP1261 continuously issues an IN-Token to the pseudo host. The pseudo host replies with an SEOC packet whenever it wishes to access the OTG hardware and with NAKs when it does not. The PAHC can issue an IN-Token at an interval of 8 microseconds when the pseudo host is NAKing.

4. SEOC Packet

An SEOC packet is a data structure that consists of:

- A 4-byte header
- 0 to 60 bytes payload

The SEOC packets allows the pseudo host to execute four basic accesses:

- Register Read
- Register Write
- Memory Read
- Memory Write

A read operation requires a transfer of two packets:

- Pseudo host sends a “Read Command” to ISP1261
- ISP1261 replies with the result of the read operation

A write operation requires just one packet, where the pseudo host sends a “Write Command” together with data to be written.

5. Interrupt Emulation in SEOC Protocol

Using a high-frequency polling scheme, the SEOC protocol allows the pseudo host to gain full access to the remotely located hardware.

It would be extremely inefficient, however, if the USB host stack were to rely on polling for all operations. A more efficient stack would require an interrupt signal. An interrupt signal allows the host stack to go into idle mode most of the time, and wake up only when the host hardware sends the interrupt signal, indicating that some processing is required. ISP1261 uses an interrupt generation scheme to emulate an interrupt path that would otherwise be impossible, as there is no direct connection between the OTG hardware and the microprocessor of the pseudo host. This interrupt emulation scheme is illustrated in the following diagram:

Figure 5-1: Interrupt Emulation in SEOC Protocol.

The pseudo host must first configure one of its endpoints to generate an interrupt signal upon receiving a data packet. The number of this endpoint is communicated to the ISP1261 during the enumeration process. When there is a pending interrupt on the OTG hardware, the SEOC Protocol Engine constructs a packet with the relevant interrupt information and sends it to the predetermined endpoint on the pseudo host. This would generate a local interrupt signal, which would invoke the appropriate Interrupt Service Routine (ISR).

6. Power Management in SEOC Protocol

ISP1261 can be put into a deep suspend mode where the SEOC Protocol Engine stops sending an IN-Token to the pseudo host, and the OTG hardware in ISP1261 goes into suspend state. Power consumption is minimal during this state of operation.

ISP1261 can be awoken from this state by:

- Pseudo host (bus resume on USB)
- OTG events such as Session Request Protocol (SRP)
- Host events such as disconnection or connection
- Peripheral events such as connection or bus resume

7. Efficiency of SEOC Protocol

Pseudo host relies on an additional protocol layer to gain access to the OTG hardware. The header and flow control required by the protocol takes a toll on the available bandwidth. The final available bandwidth depends on the amount of optimization on the stack as well as the bandwidth of the original USB peripheral hardware.

The SEOC Protocol is most efficient in stacks in which register access is minimized and the memory accessed is packed into minimal number of packets. The SEOC Protocol incurs an overhead of approximately 30% on the overall available bandwidth.

8. System Requirement

ISP1261 is designed to work with the vast majority of USB peripheral hardware, with a very basic requirement as shown in the following list:

- Control-In and Control-Out endpoint
- Bulk-In endpoint of 64 bytes
- Bulk-Out endpoint of 64 bytes
- Bulk-Out endpoint of 16 bytes (Optional)

The firmware or operating system of the platform should be upgradable, as additional software (VHAL and USB stack, class drivers) must be added.

The USB peripheral hardware can be using a built-in USB Transceiver, or it can be designed to work with an external USB transceiver such as the Philips ISP110x series. ISP1261 is designed to work with these two categories of USB peripheral hardware.

- Uses the USB D+/ D– as its interface to the USB peripheral hardware.
- Allows four modes of transceiver interfaces most commonly used. The mode is programmable by pins.

9. Applications

ISP1261 is particularly suitable for highly integrated computing devices with built-in USB peripheral hardware. It can be added onto a printed circuit board (PCB) with minimal changes, or added externally with no changes to the existing PCB, in the form of a dongle.