Designing for SDIO

By Leonard Ott, Chief Technical Officer Socket Communications Today's system designers are faced with many possible I/O functions that can be included in their design. These range from wireless options such as WLAN, Bluetooth or wideband technologies, to data collection options such as Barcode scanning or RFID readers. Committing to any one of these technologies upfront can constitute a major expense in time and dollars for an unsure payback.
One way to manage the many options is to build in a plug or slot that can take many different I/O functions using existing standards to allow the system designer to take advantage of a wider variety of I/O peripherals. However, care must be taken in choosing an interface that allows flexibility but is still reliable, robust and economical. Physical size must also be taken into account as devices continue to get smaller.
In 1999, three flash memory companies-SanDisk, Toshiba, Matsushita Electric--got together to define a new memory card format to address the size and power requirements of modern handheld devices. Out of this partnership the Secure Digital Association (SDA) was formed and has since grown to over 700 companies. In 2001, IO capability was added to the SDA specifications. Since then, a variety of IO Products have been shipped, including WLAN, Bluetooth, GPS, Camera, Barcode scanners and many others.
SD is ideally suited for handheld devices in many ways, the most obvious due to its small size, 1.4mm X 24mm X 32mm, only slightly bigger than a postage stamp.

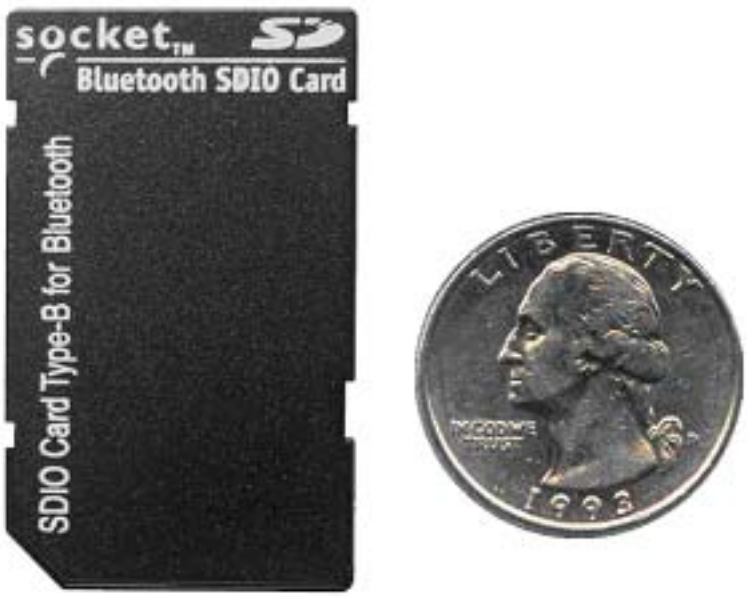

Figure 1: Small size of SDIO card
However, it is worth noting that this size can be too small for many I/O functions to be stuffed inside, thus the specification allows for the SDIO card to stick out of the

slot and, after mandated keep out areas, to grow in any direction. A well designed handheld device will have allowances for SDIO cards that can be extended; such as placing the SDIO slot at or near the 'top' of the handheld and no permanent cover over the slot.

Another important criteria for any plug in technology is the "cost" of the slot, and one way that SD/SDIO keeps the cost of the slot down is by only requiring a small number of connector pins. SD/SDIO cards are built with 9 connectors on the card, which support three different bus modes: SPI, 1 bit SD and 4 bit SD. SPI and 1 bit SD modes require only 6 active connections, including power and ground. SD 4 Bit mode adds three more data lines for four times the performance; however, not all SDIO cards support 4 bit mode, but most SDIO Host controller chips do support it, so the cost is almost negligible.

Multiple slots can be supported either in a bus topology or in a star topology. Most multi-slot systems will use a star topology, as bus topologies are subject to electrical issues when hot swapping, extra overhead for selecting the appropriate card, and are forced to the lowest common denominator regarding performance issues, such as clock speeds, bus widths and voltage selection.

SPI mode is the simplest mode of operation supported by SD/SDIO and can be implemented completely in software for a very low cost; however, the performance will be low. Most designs will utilize an existing SPI controller included in the host processor for the low level SPI and only do the protocol portion of the SD SPI interface in software. This can allow for a theoretical transfer rate of 20Mb per second if the maximum clock rate of 20MHz is used. Systems requiring higher performance and robustness should use the preferred SD1/4 bit interface with a 25MHz clock rate, allowing for theoretical through put of 100Mb per second (25Mb in 1 bit mode). Both SPI and SD modes require CRC checking on all data, but SD mode adds more error retry capabilities.

Many current processors include a full SD Host Controller built into the chip, including many variants of the Intel XScale. Adding an appropriate connector is all that is required for these processors to add hardware support for SD Memory and IO cards.

The major hardware requirement for a slot that supports SD memory for SDIO is interrupt handling. An SPI or 1 bit slot supports interrupts by monitoring one of the otherwise unused 4 bit data lines. This can generally be done with any GPIO input on the host processor. A 4 bit implementation must look at the Data 1 line either when the bus is idle (not transferring data) or between data blocks.

Every SDIO card contains up to 8 logical sections: Function 0 is the common SDIO control section, and Functions 1-7 are the actual IO functions. An SDIO card must be designed to utilize Function 0 and Function 1. Functions 2-7 are optional. Each Function, including Function 0, must have a Card Information Structure (CIS). The CIS is based on the CIS as defined by the PCMCIA and includes information on the voltage and power requirements of each Function, the functionality provided by each Function, and information about the manufacturer, including an ID number assigned to the manufacturer by the PCMCIA or JEIDA. In addition to the CIS, each Function has a set of registers that provide additional information about the card and the Functions on the card, including a pointer to the CIS information on the card. These registers include specific SD bus information for the card, such as clock speeds supported, bus widths supported (1 bit or 4 bit), interrupt control and Function enable bits.

A major component of an SD/SDIO implementation is the software to drive the bus, commonly referred to as the Bus Controller Software. Every standard, plug and play hot-swapping bus includes some form of bus controller software and SD/SDIO is no exception. The bus controller software handles the low level bus initialization, basic data transfer, card detection and initialization and, most importantly, determining the correct card function driver to load. For an SDIO bus, there are two ways the correct function driver can be determined. The first way is if the card Function is one of the defined application Functions, which at this time are Bluetooth (two types), PHS, GPS and Camera. For each of these Function types register mappings and protocols have been defined to allow a standard driver to be written.
If the card function does not fit into one of these categories, the card's CIS can be used to generate a unique Plug and Play ID. There is no standard method for generating a unique Plug and Play ID, each operating system (or bus driver) is free to calculate an ID as required. Usually, the CIS data is used to generate a unique CRC value. Since the CIS information includes vendor and product IDs and function types, this works quite well.
As expected with so few connection pins, the SD/SDIO bus is essentially a serial interface, with 4 bit mode providing some optimization. Additionally, all transactions over the SD/SDIO bus are done in a command/response format. An SDIO command to read or write a single byte, such as a status register consists of 48 bits, including the data, and the response is 48 bits, including the data. Thus, there is a significant overhead for doing single byte transactions. SDIO products that have low data rates, such as GPS receivers and Barcode scanners, have mapped a 16550 UART onto the SDIO bus, necessitating many single byte read and write operations. However, at the low data rates these devices communicate at, this is not a problem. Devices requiring higher data rates, such as Bluetooth receivers and Wireless LAN cards, have been optimized such that large packets can be sent and received, thus minimizing the command/response overhead and taking advantage of the block oriented nature of the SD/SDIO interface.
In conclusion, SD/SDIO slots can provide handheld devices with a small, low power slot that can provide additional memory storage and a wide array of IO devices, with plenty of performance for today's technologies.

Glossary of Terms

CIS - Card Information Structure
JEIDA - Japan Electronic Industry Development Association
PCMCIA - Personal Computer Memory Card Interface Association
SDA - Secure Digital Association
SD - Secure Digital
SDIO - Secure Digital I/O
UART - Universal Asynchronous Receiver/Transmitter