An Introduction to IrDA Control

Robert Stuart, IrDA Product Manager
Presented at Portable Design East on August 31, 1998

This paper is an overview of the IrDA* Control system technology. The areas covered will be the Physical Layer (PHY), Media Access Control Layer (MAC) and the Logical Link Control Bridge Layer (LLC).

NOTE: *The Infrared Data Association: IrDA was established in 1993 to set and support hardware and software standards, which create infrared communications links. IrDA standards support a broad range of computing, communications, and consumer devices. International in scope, IrDA is a non-profit corporation head quartered in Walnut Creek, California, and led by a Board of Directors, which represents a voting membership of more than 160 corporate members worldwide. As a leading high technology standards association, IrDA is committed to developing and promoting infrared standards for the hardware, software, systems, components, peripherals, communications, and consumer markets.

IrDA can be contacted at: PO Box 3883, Walnut Creek, CA, 94598; Web: www.irda.org, Phone: 925-943-6546, Fax: 925-943-5600.

INTRODUCTION

IrDA Control technology differs from the classical IrDA Data technology in several key characteristics. IrDA Data is a peer-to-peer file-oriented data transmission system. The link range was specifically designed for a one-meter range to meet a variety of requirements.

IrDA Control is a command and control architecture for communication with wireless peripheral devices such as mice, keyboards, gamepads and joysticks. This system is specifically oriented towards control data packets, and is not intended to pass files. The purpose is to pass short control packets between the host device and the remote input devices.

SYSTEM OVERVIEW

The IrDA Control system is a polled-host topology. The host device polls up to eight peripheral devices in an ordered sequence, providing service requests and handling the peripheral device responses.

The host may be a Personal Computer (PC) with peripheral devices such as a mouse and keyboard. Once the system boots up, the remote (wireless) keyboard and mouse will operate with the host PC in the same manner as a wired keyboard and mouse. The PC system drivers acknowledge the wireless mouse and keyboard, and they will work in addition to the normal mouse and keyboard, if desired.

When the peripheral devices are brought into operation, the system performs an enumeration sequence so that the host knows the peripheral device, what type of device it is and how it is expected to act. Once enumerated successfully, the device will then be bound to the host when it is to be used. Up to eight peripheral devices may be held in the device enumeration list and up to four devices actively bound and communicating with the host at one time.

If a mouse is operating, and then is not used for a few seconds, the binding will be dropped and the enumeration still held. When the mouse is again used, the system will rebind it and accept inputs from it. If the mouse remains idle and another idle device needs service, if it has previously been enumerated, it will be bound and service will be provided as long as overall system requirements are not exceeded.

The IrDA Control system has an operating range of about seven meters, on average. Peripheral devices may be used in a short-range environment, or at longer ranges such as sitting on the couch in a family room at home.

The use of IrDA Control is not limited to the PC environment. It will work as effectively with Set Top Boxes and other consumer devices and will lead to new interactive remote devices for use with these products.

The system layers covered in this paper are shown in Figure 1.

The applications and access layer software reside on top of the physical layer. These will be described from the bottom up.

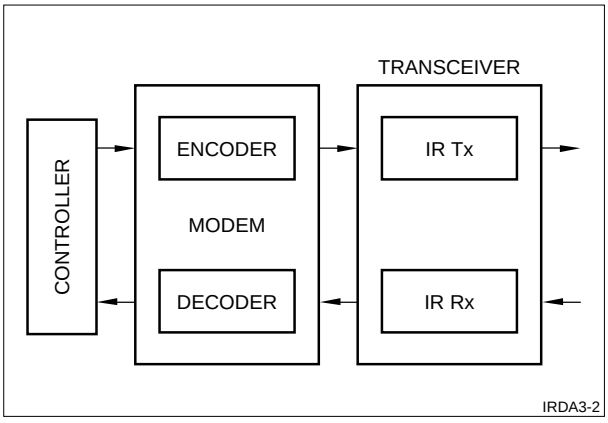

Figure 1. System Layers

PHYSICAL LAYER (PHY)

The IrDA Control system uses a PHY that is different from the earlier data-oriented IrDA 1.0 and 1.1 standards. IrDA Control uses a 16 Pulse Sequence Modulation (16 PSM) format. Each data bit encapsulates a 1.5 MHz subcarrier frequency. The overall payload capability for the system is 75 kbps*.

NOTE: *Refer to the complete IrDA Control 1.0 specification, Copyright Infrared Data Association.

The IrDA Control specification defines the transmission speeds, modulation schemes, infrared wavelengths of the optical signals emitted by the transmitter and those signals received by the receiver.

The specification does not mandate the actual signals in the encoder/decoder process or the internal signals in the IR transceiver.

The data transmission process is handled by optical transceiver devices that incorporate both the transmission Light Emitting Diode (LED) and the Photodetector (PD) circuits and amplifiers.

An encoder and decoder reside in the bit-stream path and handle data coding and the modulation process. Data is passed from the controlling device to the encoder/decoder and then on the transceiver. The 1.5 MHz subcarrier process and the coding of the transmission symbols were chosen to minimize possible interference with other transmission systems.

The basic flow of information shown in Figure 2 works for both the host and peripheral side of the system. In both cases, when actions are to be completed, the controller makes a decision and sends data out through the infrared link to the other device.

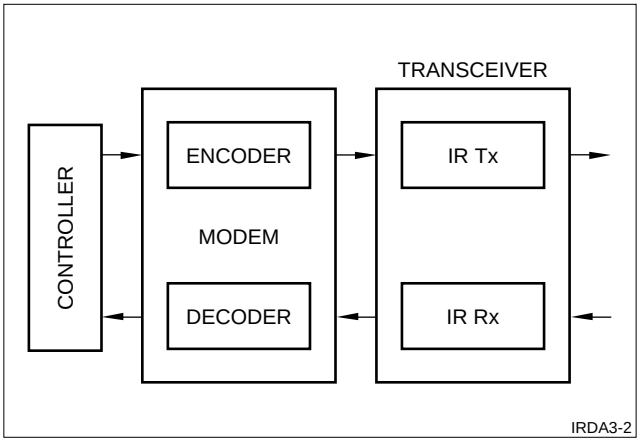

Figure 2. PHY Layer Block Diagram

In the case of a mouse, position or button press information is held in the microcontroller. When polled by the host, the mouse will respond, informing the host that it has information to send. The host will then request the information and the mouse will send it. The mouse controller will pass the data to the Modem function, which will handle all coding and modulation details. The transceiver performs the electrical to optical translation between systems.

The encoder automatically formats the data stream in the 16 PSM scheme for transmission.

A time defined as ‘symbol time (Dt)’ is equally divided into eight slots defined as ‘chips’, and a pulse is allowed only during two or four of those chip periods. Each chip time (Ct) is given by the following equation:

$Ct = Dt \div 8$

Information is transmitted according to the pulse pattern of the sequence. Unique sets of four bits correspond to a specific symbol value. The Data Bit Sets of the 16 PSM symbols are shown in Table 1. The waveforms that have legal pulse sequences are defined as 16PSM Data Symbols, or simply as Symbols.

In the 16 PSM scheme, four bits of information can be transmitted within a single symbol time. Accordingly, there are 16 waveforms defined as 16 PSM Data Symbols. Each unique set of four bits corresponds to one of 16 symbol values, and is defined as a Data Bit Set (DBS).

Table 1. 16 PSM Data Symbol Representation

DATA VALUE (HEX)	DATA BIT SET (DBS)	16 PSM DATA SYMBOL
0x0	0 0 0 0	1 0 1 0 0 0 0 0
0x1	0 0 0 1	0 1 0 1 0 0 0 0
0x2	0 0 1 0	0 0 1 0 1 0 0 0
0x3	0 0 1 1	0 0 0 1 0 1 0 0
0x4	0 1 0 0	0 0 0 0 1 0 1 0
0x5	0 1 0 1	0 0 0 0 0 1 0 1
0x6	0 1 1 0	1 0 0 0 0 0 1 0
0x7	0 1 1 1	0 1 0 0 0 0 0 1
0x8	1 0 0 0	1 1 1 1 0 0 0 0
0x9	1 0 0 1	0 1 1 1 1 0 0 0
0xA	1 0 1 0	0 0 1 1 1 1 0 0
0xB	1 0 1 1	0 0 0 1 1 1 1 0
0xC	1 1 0 0	0 0 0 0 1 1 1 1
0xD	1 1 0 1	1 0 0 0 0 1 1 1
0xE	1 1 1 0	1 0 1 0 0 1 0 1
0xF	1 1 1 1	1 1 1 0 0 0 0 1

The encoder will place each bit of the DBS into one of the eight chip times in each symbol, as explained above. The encoder will also insert the 1.5 MHz subcarrier into each bit envelope in preparation for transmission of the symbol.

For the Data Bit Set corresponding to the hex value in the left column, the encoder will format the 16 PSM data shown on the right. This data symbol will then be transmitted over the optical link.

The system packet structure is shown in Figure 3. Two types of packets are used in the IrDA Control system: short packets and long packets.

Each packet consists of six fields: The Automatic Gain Control (AGC); Preamble (PRE); Start Flag(STA or STL); MAC frame; Cyclic Redundancy Check (CRC, either CRC-8 or CRC-16); and Stop Flag (STO).

Data transmission starts with the leftmost bit in each field. The AGC field is used to set the AGC level in the receiver.

The Preamble is used to attain clock synchronization. The Start Flag is used for symbol synchronization. The MAC frame is passed, the CRC is sent and the Stop Flag sent to end the transmission.

The 1.5 MHz subcarrier pulses in the data bits are transmitted for a logical ‘1’ and are not transmitted for a logical ‘0’.

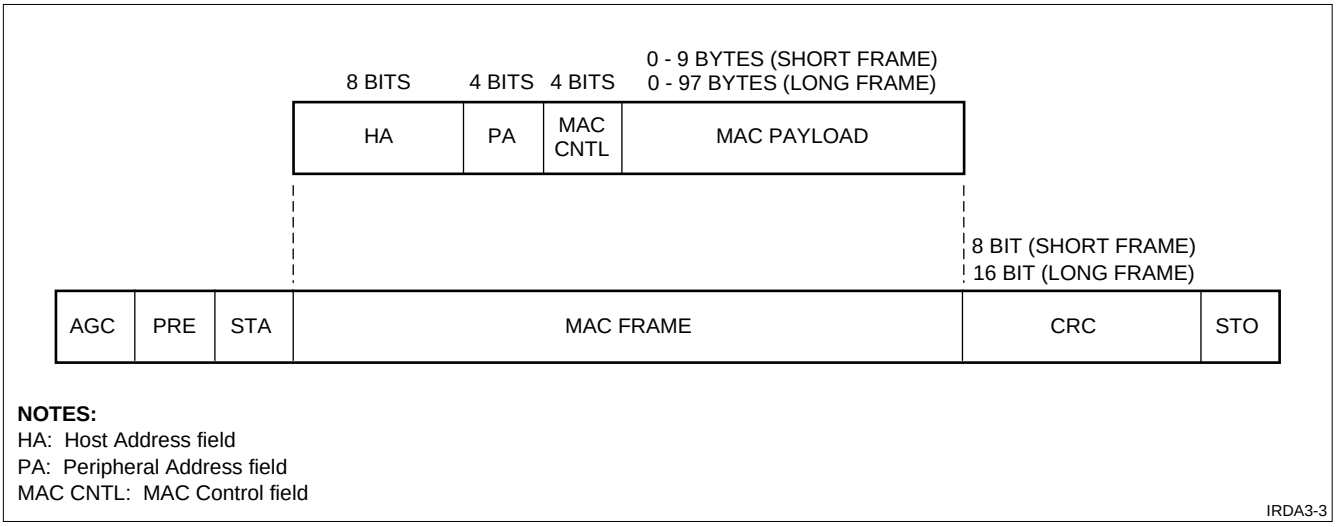

Figure 3. MAC Frame Structure

The diagram in Figure 4 shows an example of data value 0xF5 is being sent out. The lower half of the data byte is sent first, so the value F5 is actually transmitted as 5F.

The Subcarrier Emission Pulse Chip (SEPC) is the string of ten 1.5 MHz subcarrier pulses that are inserted into each logical '1' data bit of the eight chip times in the transmitted 16 PSM data Symbol.

When the data stream is detected at the receiver, the 1.5 MHz subcarrier pulses will be extracted and the coded Data Symbol will be forwarded to the decoder. The Data Bit Set associated with each 16 PSM Data Symbol will be extracted and passed to the MAC layer. The system will recognize the control symbols such as Start and Stop and translate them appropriately.

The system specifications call for a minimum five-meter operating range. Various distances and operating angles between host and peripherals are described in the full specification. The basic operating environment calls for a ±30° angle from the transceiver in the horizontal plane and ±15° in the vertical plane.

Detailed specifics of the ranges, angles and operating conditions are in the complete IrDA Control specification.*

NOTE: *Refer to the complete IrDA Control 1.0 specification, Copyright Infrared Data Association.

MEDIA ACCESS CONTROL LAYER (MAC)

The IrDA Control system consists of hosts and peripherals between which infrared communication takes place. The host manages its communications with multiple peripherals on a time division basis, using polled-response handshakes.

The host polls all of the bound peripherals to determine which items need to be serviced. The peripherals respond to the poll from the host, and do not initiate transmission. The peripheral devices do not transmit unless they are given response permission.

The only exception is when the host is asleep and a peripheral initiates a wakeup call for service. Then the host steps back into the polling sequence and looks for devices to service. If there is no transmission between the host and any peripheral for a set time, then the host will again enter sleep mode.

Generally, hosts do not communicate with each other, however there could be times when they need to do so, if there are multiple hosts in a room. Usually if multiple hosts are present, they detect each other and dither their transmissions to reduce the possibility for interference.

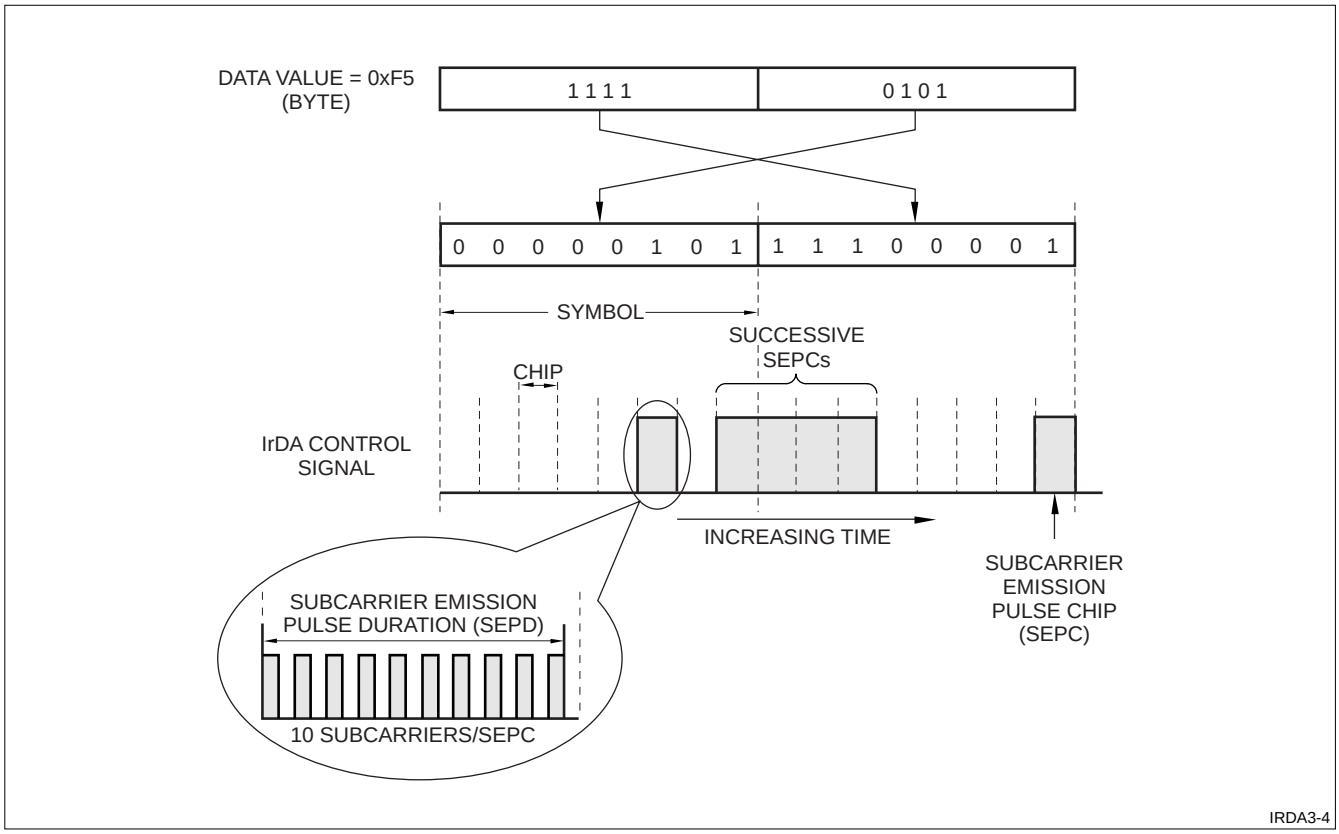

Figure 4. Data Transmission Coding Example

Each device has an address and identifier that clearly identifies hosts and peripherals. An 8-bit host address (HADD) and a 16-bit host ID (HostID) identify a host. A host address may be set at the factory, or be determined while the host is set up.

A peripheral is identified by a 32-bit physical ID (PFID). A host and a peripheral have to exchange address/ID information as part of a process called enumeration.

A logical 4-bit peripheral address (PADD) is uniquely assigned to each peripheral by the host to establish 'active' communication. This procedure is a part of a process called binding, which is performed when an enumerated peripheral requests communication with the host. The ID numbers are used only in the beginning of a communication to identify the devices. After the identification, hosts/peripherals are identified only by their address.

The requirements for IrDA Control communication vary depending on the application. In order to comply with various application requirements, three operational modes are offered for a host.

Mode-0: Sleep Mode

This is a 'Low resource usage' mode to minimize power consumption when a host and its peripherals do not need to communicate. This is also the default mode for each host.

Mode-1: Normal Mode

This is the normal operational mode of the host. This mode supports peripherals that may have different bandwidth requirements. Peripherals supported include devices that must be handled within certain time limits [Critical Latency peripheral (CL)], like joysticks and game pads.

Peripherals that normally do not have critical latency requirements [Non-critical Latency peripheral (NCL)], like Remote Control units are also supported. Keyboards and mice could be handled as NCL or CL peripherals under this mode. A CL peripheral is able to support CL polling rate.

An NCL peripheral is not able to support CL polling rate and is always polled at the NCL polling rate. A host must guarantee that a CL peripheral is polled every 13.8 ms.

Mode-2: IrDA-coexistence mode

This operating mode is available to allow coexistence of IrDA SIR version 1.1 data communication and IrDA Control communication.*

NOTE: *Refer to the complete IrDA Data specification version 1.1 or 1.2 for detailed information on SIR and FIR modes of operation.

The host may move between any of the three modes listed above. It is not required that all hosts support all three modes.

When enumerated, the peripheral identifies the type of service that it requires, so that the host knows whether CL or NCL support is to be used. Critical Latency devices have priority over Non Critical Latency devices, which may not be serviced within the 13.8 ms cycle, depending on system resources and how they are being used. In some cases they may unbind and wait for a service slot.

If four CL devices such as joysticks are actively engaged in a game, then the NCL devices may not be poled for an extended period of time. They do remain enumerated and known to the host. Should the play of the game slow such that the CL activity decreases then the NCL devices can rebind and be serviced. An example of this is to stop the play and enter some text or player names or other similar activity.

FRAMES

Two types of MAC frames are defined based on the maximum MAC payload data length that can be transmitted by a host or a peripheral. One is a short frame and the other is a long frame. A short frame can accommodate up to 9 bytes of MAC payload data and must be transmitted with the STS flag, STO flag and CRC-8. These are shown in Figure 5.

A long frame can accommodate up to 97 bytes of MAC payload data and must be transmitted with the STL flag, STO flag and CRC-16. Long frames are suitable for larger data exchanges.

Host devices and peripheral devices may always use short frames. Host devices may use long frames in Mode-1 only. Peripheral devices may use long frames only when responding to a polling packet from a host device whose long frame enable bit is set to '1', which occurs when the host is in Mode-1. A host device and a peripheral device are prohibited from both using a long frame in the same polling procedure (in the polling frame from a host as well as the responding frame from a peripheral).

In this case it is also possible that the NCL polling cycle may be stretched if several NCL devices are exchanging long frames. Once the activity is finished, the normal polling cycle will be resumed.

The basic polling cycle for the IrDA Control system is defined as 13.8 ms. Up to four CL peripherals can be polled with short frames within this cycle time. The basic polling cycle time is dependent on the minimum interval between inputs from a peripheral input device, such as a joystick or gamepad. These devices have the most critical response time. Keyboards and mice are more flexible with regards to actual response time.

A Non Critical Latency device is not guaranteed a poll within the 13.8 ms time. The entire polling cycle time is defined as the time period in which all bound peripherals can be polled by a host. The host has to manage all of the peripherals so that the entire polling cycle time does not exceed 69 ms.

The possibility exists that cases may arise when the cycle time is shorter than the time required servicing all of the items in the list. The host will try to service all of the devices on the next poll cycle. A peripheral device that misses one or two poll cycles will not immediately be unbound. The Peripheral must not acknowledge approximately 100 polls before the host drops the binding.

With this information in mind, long frames are only applicable for transmission when CL devices are not bound on the system, or their service requirements do not restrict the system from servicing long frames.

The MAC frame field structure is shown in Figure 3. The Host Address and Peripheral Address fields and the number of bits associated with each are shown.

The MAC control field has a variety of functions. It is used to communicate packet direction, bind timer restarted, long frame enable, device hailing and polling requests.

Enumeration is the procedure in which a host and a peripheral recognize (discover) each other to enable communication between them. The host identifies the peripheral using the peripheral physical identifier (PFID) and the peripheral identifies the host using a host address. The PFID and the HADD are exchanged during the enumeration procedure.

Enumeration is basically the process where the host adds the peripheral to the list of items that it 'knows'.

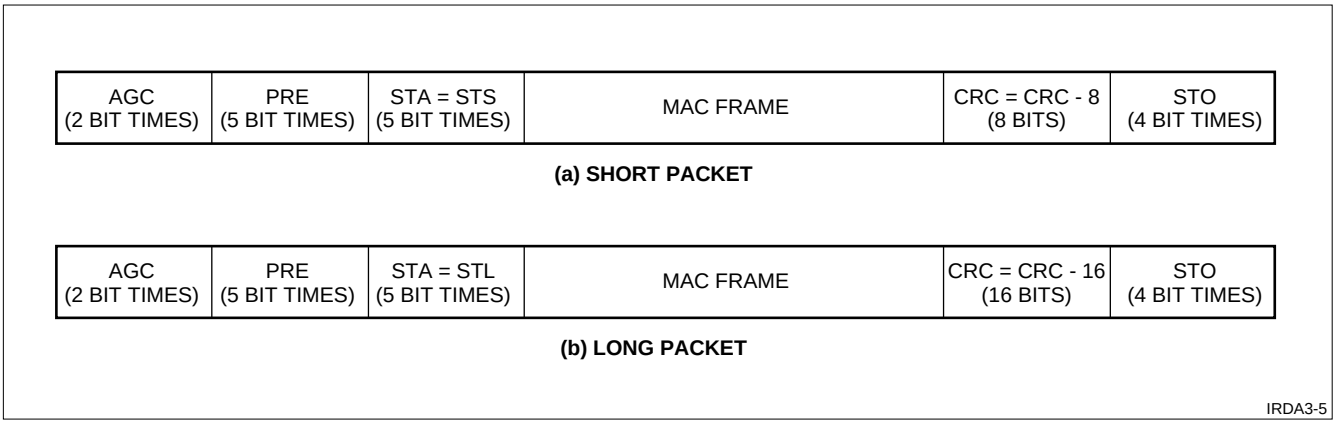

Figure 5. System Packet Structure

An IrDA Control peripheral must be enumerated (and bound) with a host before it can exchange data with the host side application layer. A peripheral that has not been enumerated must not perform any communication other than the enumeration procedure. The host ignores a hailing response received from any peripheral to which it has not enumerated.

Special mechanisms may be required on IrDA Control devices to initiate the enumeration procedure, such as a button located on the device. The enumeration procedure uses short frames only and is carried out in the following steps. An example of this is if a new device asks for service when the host is not hailing for new devices, but servicing devices already enumerated and bound.

Peripheral address '0xF' is used in the process of enumeration. During enumeration, the host polls using a peripheral address '0xF'. Unenumerated peripherals are allowed to respond to host polls with PADD of '0xF' only.

The host issues an enumeration hail with the 'hailing' bit set to '1', and a peripheral address 0xF. This host poll frame includes information about the host (Host ID and Host Info).

After storing the HADD, HostID and Host Info data, a peripheral that desires enumeration responds to the hail frame with a frame including its PFID and information about itself, Peripheral Info.

The Peripheral Information tells the host whether the peripheral is a critical latency peripheral (i.e., the peripheral supports the CL polling rate) or not, as well as whether the peripheral has the ability to send or receive long frames. Other information that can be sent as part of the Peripheral Information is a Device Descriptor, Configuration Descriptor and other fields that are used to tell the host more about the peripheral and what it is expected to do. This information can be used to tell the host which device driver is to be used.

The host, which has received the response frame, stores the PFID and Peripheral Information. Then in the next polling cycle, it responds to the peripheral with a frame including the received PFID.

Once the enumeration process is completed, the PFID will be added to the enumeration list in the host. Any item that has been enumerated will be in the list, up to a total of eight items. When additional items are enumerated, the least active device will be dropped from the list. As peripherals are brought into use, the list will be updated for those devices that are in use and have been recently used.

The enumeration procedure may fail due to multiple peripherals responding to the same hail. After responding to the enumeration hail, the peripheral should receive a response from the host with PFID. If a peripheral does not receive the above packet from the host within 69 ms after a request, the peripheral recognizes the failure and goes back to responding with a frame that includes its PFID and a random back-off value between 0 to 7.

If the random back-off value is 0, this peripheral will send a response frame in the next hailing cycle. If the random back-off value is 7, this peripheral will ignore 7 hailing frames and can send a response frame in the eighth hailing cycle.

The full detail of all possible modes and various conditions are in the IrDA Control specification. *

NOTE: *Refer to the complete IrDA Control 1.0 specification, Copyright Infrared Data Association.

The process in which a host dynamically recognizes that an enumerated peripheral needs to be added to the active device-polling loop is called 'Binding'. When bound, the host will include the peripheral device in the active polling cycle and issue poll requests to the device on a cyclic basis. To bind, a process similar to the enumeration sequence is used.

When bound, the peripheral will respond to host polls indicating that it has data for the host. The host will then ask for the data.

When a bound peripheral does not respond to polling for a certain time period, the host recognizes that the peripheral does not need further communication and drops it from the active polling list. This process is called 'Unbinding.' An unbound peripheral is still enumerated and can be picked up into the polling cycle at any time.

When the device has been unbound and sits idle, the peripheral will go into a sleep state where power consumption is very low, typically 1 μA. If we use the mouse as an example, it goes to sleep once it sits idle for more than a few seconds. When asleep, it will awaken when moved or one of the buttons is pushed. Then it will respond to a hailing poll, or will send a wake frame to the host if it is asleep.

The complete details of all cases for binding and unbinding are in the complete specification.*

NOTE: *Refer to the complete IrDA Control 1.0 specification, Copyright Infrared Data Association.

PROTOCOL STACK

The IrDA Protocol stack resides on top of the MAC layer and services the Human Input Device (HID) definition device LLC, the HA LLC and the future device LLCs (yet to be defined). This stack is shown in Figure 6.

At the present time, the HID stack is the most complete and is used in conjunction with standard Universal Serial Bus (USB) definition HID devices.

The stack is basically the same for both the host and peripheral side. In the case of the peripheral, only one of the three application stack LLC columns would be used to service the inputs from the device. In the case of a special function device, more than one column may be present, however the complexity of the peripheral device increases drastically.

In the case of the host, the HID drivers are compatible with USB logical devices. The USB Host version interfaces will implement a Common Class driver that links in with Windows 98* USB drivers to provide an interface to the PC system drivers. The Common Class driver definition allows for multiple devices to work through the single device endpoint in the system, without requiring multiple Serial Interface Engines in the host interface device.

NOTE: *Windows 98 is a trademark of Microsoft Corporation.

When the USB host recognizes a bound peripheral, the device driver information will be passed up to the operating system to identify the bound device. The host machine, when operating with Windows 98, will then ask for the device description and the host will pass this request out to the peripheral.

The peripheral will respond with its peripheral information, which will be passed up to the operating system. The operating system will then load the appropriate driver to service the peripheral.

When a host device is embedded in a system, which has its own operating software, the same process is followed. In this case there may be a variety of drivers to support multiple types of peripheral devices. The operating system will need to provide the same service capabilities, however the USB interface may not be present. If the USB interface is present, then the system software should support the standard USB services.

An example of such an implementation is a Set Top Box (STB) used in the family room. The addition of a keyboard and mouse for use with an Electronic Program Guide (EPG) may be a desired feature set. The STB may have a packaged operating system or use its own software. In either case, all of the AMC and LLC functions must be supported to provide service to the peripheral devices. The STB acts as the host and polls the room.

In a STB application, a forced enumeration may be a desired function. An example is the use a specific keyboard and mouse combination product. The user would not want to enumerate every device that came into view, such as another keyboard or a series of game controllers and joysticks. The user may prefer to tell the STB when a new device should be enumerated instead of hailing the world at large. In either case, the same enumeration and bind process as described above would apply.

The operating system of a STB is likely to talk directly with the IrDA Control host controller and not rely on a USB interface due to the additional cost. It also does not make fiscal sense to add another interface only to speak to an embedded IC. However, depending on feature set, the STB may include a USB port on the rear, so that an IrDA Control interface can be added at a later time, depending on user functions and feature set. In this case, a standard USB host interface could be plugged into the port and function as part of the overall system. The STB operating system would need to support USB services. In STBs that incorporate Windows CE*, some USB support may be provided as part of the operating system in the future, and adding an IrDA Control feature would not be difficult.

NOTE: *Windows CE is a trademark of Microsoft Corporation.

The Home Appliance application column is still being defined. The Future device application column provides for future devices not yet developed or defined.

LOGICAL LINK CONTROL (LLC)

The Logical Link Control Layer provides resources for reliable communication of data between the MAC layer and the application, as shown in Figure 6. IrDA Control has a goal of providing components, and a protocol, that allow it to be used in a wide range of devices, with great variation in resource and cost requirements.

The LLC provides the link layer resources used by IrDA Control devices, regardless of what higher level protocol may be used. It enables reliability through the use of lightweight protocol controlling frames.

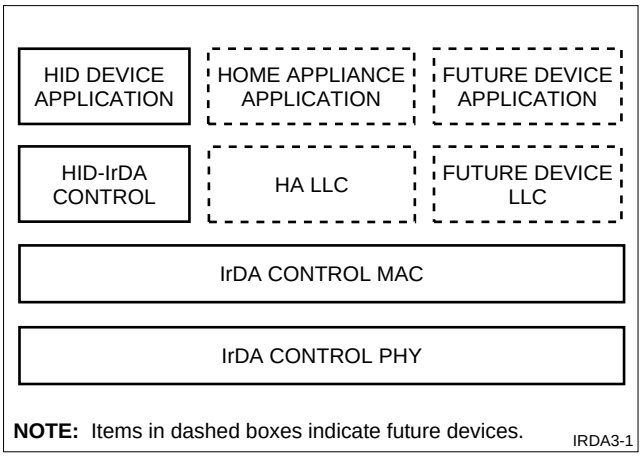

Figure 6. Protocol Stack

The LLC layer specifies only simple methods for acknowledgment of delivery. Therefore, it might happen that the LLC Layer by itself couldn't honor an application that requires strictly reliable data communication. The upper layers should implement error correction functions, re-transmission functions, and so on, when assurance of reliable communication is necessary. In some cases, such as USB-HID, much of the Link Layer actually resides with the Host Operating System, and the IrDA Control LLC layer is used as a bridge to and from the MAC layer.

The LLC Layer is not utilized during enumeration and binding procedures. The functions of LLC Layer are:

Information Features

- Send Commands
- Receive Requests
- Send Data
- Receive Data

Reliability Features

- Prevent duplicate frames.
- Acknowledgment of delivery based on single frame transmission (ACK).
- Re-transmission function responding to NAK or ignore.
- Provide notice of unsupported features or inability to handle a request at this time.

The LLC field in the MAC frame has control and payload sections that are used to communicate information between layers. Mode and status fields pass bits indicating the packet type and operation to be conducted.

END-USER PRODUCTS

The end goal of the IrDA Control system is to allow developers to create host and peripheral products that are basically interchangeable. A mouse designed for one system should work on all systems, regardless of the manufacturer.

In like manner, any host that supports a mouse, regardless if it is on a PC operating system or an embedded controller product running in a consumer product should recognize an IrDA Control mouse.

Part of the descriptor fields in the peripheral product is an explanation of what it is, so that the correct driver is loaded. Game controllers is another category could experience quick growth, as there is already a large number and variety of products on the market. Drivers for most of these are common on systems or are easily loaded. A correctly implemented IrDA Control device appears transparent to the overall operating system.

One device that is envisioned for the Home Automation LLC path is intelligent remote controls for consumer products. Instead of six remotes on the coffee table, a single remote that could talk to every device in your home. It could be able to download the EPG from the STB and program the VCR and other products. If it has a touch-sensitive Liquid Crystal Display, it could be used by anyone. The size and format of the display could then be customized to meet user needs.

SUMMARY

IrDA Control is a polled host system that supports a variety of cordless input peripherals. It is a control input oriented system that eliminates the wires required for standard input devices. It is appropriate for applications where the user prefers to eliminate the tether and step back from the product in use.

IrDA Control input products are also appropriate in applications where the user prefers to eliminate the clutter of cables in a short-range application.

This paper outlined the hardware approach for the IrDA Control system as well as the Media Access Control and Logical Link Control layers. The MAC controls the software coordination between the hardware and software for system operation. The LLC layer arbitrates the link and tries to maintain an orderly flow of information.

The IrDA Control system can be implemented into a wide variety of products. It is currently PC-oriented, however it does not have to be. Anything that you can envision is a possible application, when you are ready to remove the cable for control input devices.

LIFE SUPPORT POLICY

SHARP components should not be used in medical devices with life support functions or in safety equipment (or similiar applications where component failure would result in loss of life or physical harm) without the written approval of an officer of the SHARP Corporation.

LIMITED WARRANTY

SHARP warrants to its Customer that the Products will be free from defects in material and workmanship under normal use and service for a period of one year from the date of invoice. Customer's exclusive remedy for breach of this warranty is that SHARP will either (i) repair or replace, at its option, any Product which fails during the warranty period because of such defect (if Customer promptly reported the failure to SHARP in writing) or, (ii) if SHARP is unable to repair or replace, refund the purchase price of the Product upon its return to SHARP. This warranty does not apply to any Product which has been subjected to misuse, abnormal service or handling, or which has been altered or modified in design or construction, or which has been serviced or repaired by anyone other than Sharp. The warranties set forth herein are in lieu of, and exclusive of, all other warranties, express or implied. ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR USE AND FITNESS FOR A PARTICULAR PURPOSE, ARE SPECIFICALLY EXCLUDED. In no event will Sharp be liable, or in any way responsible, for any incidental or consequential economic or property damage.

The above warranty is also extended to Customers of Sharp authorized distributors with the following exception: reports of failures of Products during the warranty period and return of Products that were purchased from an authorized distributor must be made through the distributor. In case Sharp is unable to repair or replace such Products, refunds will be issued to the distributor in the amount of distributor cost.

SHARP reserves the right to make changes in specifications at any time and without notice. SHARP does not assume any responsibility for the use of any circuitry described; no circuit patent licenses are implied.

SHARP®

NORTH AMERICA

SHARP Microelectronics
of the Americas
5700 NW Pacific Rim Blvd., M/S 20
Camas, WA 98607, U.S.A.
Phone: (360) 834-2500
Telex: 49608472 (SHARPCAM)
Facsimile: (360) 834-8903
http://www.sharpsma.com

EUROPE

SHARP Electronics (Europe) GmbH
Microelectronics Division
Sonninstraße 3
20097 Hamburg, Germany
Phone: (49) 40 2376-2286
Facsimile: (49) 40 2376-2232
http://www.sharpmed.com

ASIA

SHARP Corporation
Integrated Circuits Group
2613-1 Ichinomoto-Cho
Tenri-City, Nara, 632, Japan
Phone: (07436) 5-1321
Telex: LABOMETA-B J63428
Facsimile: (07436) 5-1532