Implementing an IrDA Control Peripheral

Robert Stuart, IrDA Product Manager

Presented at Portable Design 1999 on February 3, 1999

This application note provides an overview of the IrDA Control system, with a focus on the development of a peripheral device.

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

The development of a peripheral such as a mouse and keyboard will be examined here, with emphasis on the mouse. Many aspects of a keyboard will be similar.

When the peripheral devices are brought into operation, the IrDA Control system performs an enumeration process 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 previously enumerated idle device needs service, it will be bound and service will be provided as long as overall system requirements are not exceeded. More than one device can be bound and in service at one time.

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 peripheral devices discussed here are assumed to be operating up the USB HID stack shown. The same principles apply to the Home Appliance (HA) vertical application stack and the Future Device Application stack. The difference between these three application stacks is the method in which the messages are passed up to the operating system.

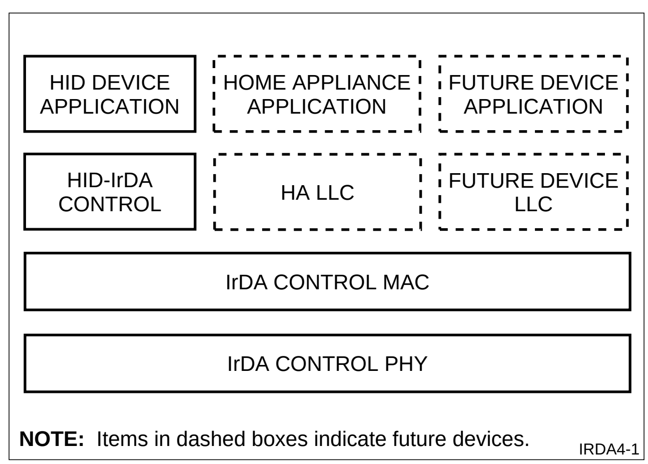

Figure 1. System Layers

A more in depth of explanation of the system layers and features is provided in “An Introduction to IrDA Control”.[1] Complete details for system timing and priorities are included in the IrDA Control specification.[2] In the case of the devices discussed here, general application issues will be explained as they occur.

NOTES:
1. “An Introduction to IrDA Control”, Robert Stuart, August 31, 1998. See www.sharpsma.com
2. IrDA Control Specification, Copyright Infrared Data Association. See www.irda.org/standards/specifications.

THE SYSTEM

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.

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 Peripheral ID (PFID). A host and a peripheral have to exchange address/ID information (HADD/HostID and PFID) 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.

When enumerated, the peripheral identifies the type of service that it requires, so that the host knows whether Critical Latency (CL) or Non-Critical Latency (NCL) support is to be used. Critical Latency devices have priority over Non Critical Latency devices, which may not always 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 polled for one or two cycles. 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 2.

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. It is prohibited that both a host device and a peripheral device use 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 the cycle time is shorter than the time required for 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. Standard keyboard and mouse devices must not acknowledge polls for five seconds before the host drops the binding. Gaming devices have up to 30 seconds before being unbound.

The MAC frame field structure is shown in Figure 2. 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 restart, Long frame enable, device hailing and polling requests.

AGC (2 BIT TIMES)	PRE (5 BIT TIMES)	STA = STS (5 BIT TIMES)	MAC FRAME	CRC = CRC - 8 (8 BITS)	STO (4 BIT TIMES)

(a) SHORT PACKET

AGC (2 BIT TIMES)	PRE (5 BIT TIMES)	STA = STL (5 BIT TIMES)	MAC FRAME	CRC = CRC - 16 (16 BITS)	STO (4 BIT TIMES)

(b) LONG PACKET

IRDA4-2

Figure 2. System Packet Structure

ENUMERATION

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 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'.

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. For example, trying to enumerate a new device during an active session of play is inconvenient and should be delayed until a pause in the play. In many cases a new device should not be added to a system unless permission is granted.

Peripheral address '0xF' is used in the process of enumeration. During enumeration, the host polls using a peripheral address '0xF'. Peripherals that have not been enumerated 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 Information).

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

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.

The host, which has received the response frame, stores the PFID and Peripheral Information. 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. The keyboard and mouse will always be retained, as they are important input devices for the overall system. As peripherals are brought into use, the list will be updated for those devices that are in use and have been recently used.

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

NOTE: *IrDA Control Specification, Copyright Infrared Data Association. See www.irda.org/standards/specifications.

BINDING

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 assign a soft 4-bit Peripheral Address and 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 less than 10 µA. This process also puts the optical transceiver in a power-down state as well.

If we use the mouse as an example, it goes to sleep once it sits idle for more than a five 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 host will again assign a PADD during this bound session. The peripheral device may be serviced in a different slot than the position it was originally assigned.

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

NOTE: *IrDA Control Specification, Copyright Infrared Data Association. See www.irda.org/standards/specifications.

PERIPHERAL CHARACTERISTICS

The IrDA Control specification places requirements on a peripheral device to provide a certain level of capabilities and resources.

Each peripheral device has a Peripheral ID Field. The device must be able to generate and store a 32-bit Physical ID (PFID). This ID consists of a 30-bit pre-assigned or pseudo-random number and a 2-bit field to signify the type of device.

In the PFID field, bits D0 - D29 are the pre-assigned or pseudo-random bit fields. Bit D30 is the Mouse Device Flag, and bit D31 is the Keyboard Device Flag. Each bit is set to '0' if it is not the stated device, and is set to a logical '1' if it is the stated device. In the case of a multifunction device, such as an integrated keyboard and pointing device, both fields may actually be valid.

The peripheral device also has a Peripheral Info field that is 16 bits in length.

This field indicates the capabilities of the peripheral device to the Host during the enumeration process. The format for the Peripheral Information field is summarized below. If the capability is available, the bit value is set to '1'. Reserved bits in this field must be set to '0'.

Table 1. Peripheral Info Field Summary

BIT	DESCRIPTION
D0	Upper Layer HID-IrDA protocol support
D1	Upper Layer HA-LLC protocol support
D2 - D3	Reserved
D4	Support for long frames from the host
D5	Support for long frames to the host
D6	Maximum polling rate: NCL/CL
D7 - D15	Reserved

Each peripheral device also needs to be able to store a host-generated 4-bit Peripheral Address and an 8-bit Host Address, as well as a 16-bit Host ID. This information is used to correctly identify the host that the device is enumerated with. The PADD is used to correctly interpret the poll that is directed to the device.

During enumeration the host and peripheral device will exchange ID and address and device information. The host is required to retain peripheral device information in its own memory, and will retain this information until replaced by another device.

The long frame mode identified for bits D4 and D5 above allow the transfer of a longer-than-normal packet. This feature allows for NCL devices to pass information of greater content.

An example is for a peripheral device to upload a new character set, key map or other information.

IMPLEMENTATION

Using a mouse as an example, the development block diagram is shown in Figure 3. The optical signals are handled by the IrDA Control transceiver, which directly interfaces with the Peripheral Engine (PE). All of the IrDA Control Media Access Control (MAC) and logical Link Control (LLC) functions are within the PE, which operates independently.

The internal mouse microprocessor communicates with the PE directly. The rotary encoder input data for the mouse position, and button press information is collected by the microprocessor.

Other status information such as the state of the battery and identification bits is configured at the microprocessor. The data from the encoder and status inputs are held in the microprocessor until the correct time to send them to the PE.

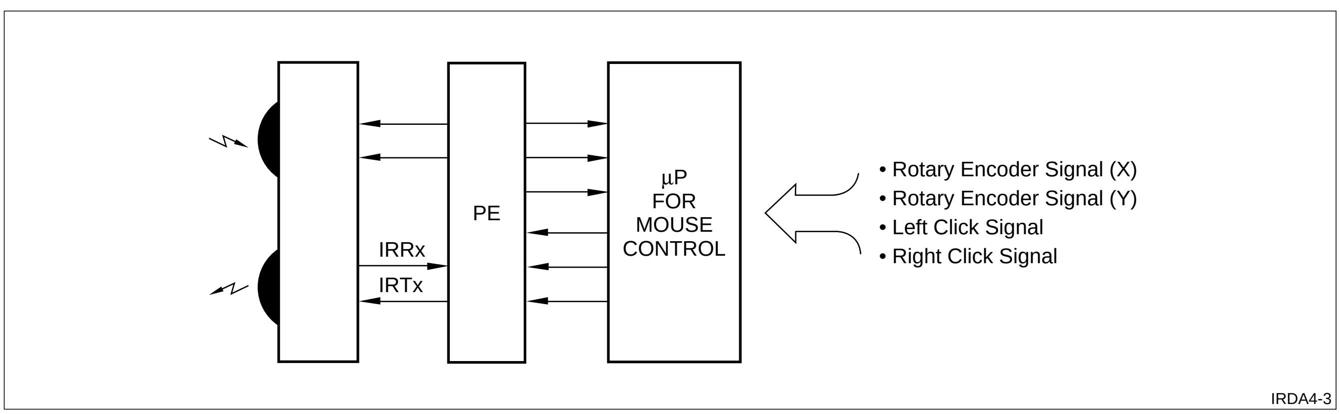

Figure 3. Mouse Block Diagram

MOUSE BLOCK DIAGRAM

When the PE has received a message via the optical interface, it sets the MSG line high, and the microprocessor will issue the Get_Message command to read out the message. The microprocessor provides the data clock to read data out of the PE and to write data into the PE.

The microprocessor monitors the RDY line for flow control between the microprocessor and the PE. The program running on the microprocessor monitors the input lines and manages the communications between the two devices. When the external inputs require action, it signals to the PE and handshakes the data to the PE.

The microprocessor manages the activity of the mouse. The PE handles all of the infrared data transactions as well as all of the encoding and decoding functions required by the link. All of the required MAC and LLC functions as part of IrDA Control transactions are also handled automatically in the PE.

The IrDA Control specification identifies what the required activity and conditions between the infrared ports, and does not dictate how these transactions are handled in the implemented system. It does state how the system must operate, but some flexibility is allowed in the implementation as long as all required functions are supported.

Additionally, it is the goal of the specification to have all IrDA Control devices work interchangeably without specific setup requirements between devices from each manufacturer.

In this basic implementation, the PE is fairly smart, however it depends on the microprocessor to take the necessary actions to convert mouse actions into data that can be sent.

The microprocessor issues a series of basic commands to the PE to set its status and to tell the system what the mouse is doing. The PE knows how to encode the mouse data into IrDA Control format and handle all of the communication process steps necessary.

For example, when the mouse is first powered or the battery is changed, a Reset should be asserted by the microprocessor to initialize internal functions. If for some reason the mouse is not able to communicate with the host in an intelligible manner, another reset may be required to correct the situation.

When the mouse has become idle for more than five seconds, the PE automatically shuts down to conserve power. This forces a maximum battery conservation mode. Once the mouse is moved or a button pressed, the microprocessor can wake it up and return it to operation with any command.

The microprocessor can also load Peripheral Information data into the PE for its use during the session. When first reset, the data describing the device and the service it needs is loaded. This is the 16-bit field previously described.

The PE has memory to retain the 32-bit PFID and the 16-bit Peripheral Information. All other data is loaded into the PE when requested from the Host side. An example is when the Host requests the Peripheral Information fields that identify what the mouse is by name.

The microprocessor has a variety of commands at its disposal for setting and reading information from the PE. The microprocessor can command the PE to reset its condition, send data, get message, set the peripheral information and set the operation mode. A variety of other commands are also available to the microprocessor.

A variety of commands are also available to the PE when it receives messages from the IR link. It will signal to the microprocessor that data has been received, pass peripheral information to the microprocessor, indicate a change in the IrDA Control system status and to abort a function in process.

Interrupts that are serviced due to moving the mouse or pressing a button cause the microprocessor to send commands to the PE. The rotary encoder outputs from the X and Y position inputs and digitized and held as data values that will be sent to the PE to be passed up to host system. The microprocessor will send the appropriate commands to the PE in order to communicate the necessary information.

The host system knows the type of peripheral and the type of input signals it will provide. For example, the host knows the type of inputs expected from a mouse, or a similar pointing device.

ENDPOINTS

When developed to be compatible with the Universal Serial Bus (USB) system, endpoints are the final destination of communication flow between the host and peripheral devices in general. IrDA Control devices use the same endpoint concept as USB, but it is slightly different than the physical endpoints used in USB.

The IrDA Control endpoint is mapped as one of the elements of the Logical Link Control layer control field, and is handled by the HID LLC layer. The Pipe is originally a concept idea referring to the logical communication channel in USB and is also used in IrDA Control. Four endpoints are supported:

- Endpoint 0x0 is the Control Pipe, and is required for all devices. Host Commands and Device Requests are sent via this Pipe.
- Endpoint 0x1 is an IN Data Pipe and is optional. Data from a device to a host uses this pipe type.
- Endpoint 0x2 is an OUT Data Pipe and is optional. Data from a host to a device uses this pipe type.
- Endpoint 0x3 can be an IN or OUT Data Pipe and is optional.

Multiple endpoints may be used on a single device and allow for multiple devices to be implemented in a single product. An example is a composite keyboard that includes a pointing device. The host system can write to either endpoint on the composite device. For example, the keyboard can be assigned at Endpoint (EP) 1, and the pointing device functions at EP3. The data field to assign the Endpoint is available in send data commands.

PRODUCT PACKAGING

Many products that use infrared communication locate their reception component behind dark plastic. This often fits in with the overall design and coloring of the product, but has an intentional design purpose as well.

The photodiode that is used in optical receivers is sensitive to a variety of light wavelengths, not only the wavelength intended for reception. Visible light has many component factors and sources in a room. The dark plastic used in product faceplates acts as a high-pass filter, reducing the amount of visible light and other wavelengths landing on the photodiode and raising the internal noise currents.

The reduction of unwanted wavelengths provides for a quieter and more sensitive receiver. The wavelength used for IrDA Control communications is 880 nm. The characteristic of any plastic used in a final product should keep this in mind and not attenuate this wavelength.

A simple material for test purposes is to obtain a roll of 100 ASA color film. Pull the film all of the way out of the can and expose it to outdoor light for 30 seconds or so.

Have the film developed, but first inform the person operating the developer what you are doing so that your intent is understood, and the material is processed correctly.

The developed negative stock should be dark and difficult to see through. Use of one or more layers will provide a variable attenuation filter that you can tailor for experimental purposes.

OTHER PERIPHERAL DEVICES

The keyboard and mouse described here are intended as examples and are not the only devices that may be implemented in an IrDA Control system.

Many other types of devices not described can be conceived and designed. A gamepad and be developed that performs as well as a wired unit. Such a product would identify itself as a critical latency device during enumeration.

A smart remote control device could also be developed to take advantage of the long packet mode to upload the correct programming information for a new television or VCR. The Electronic Program Guide (EPG) for a set-top box could be operated from the smart remote instead of using an on-screen display, interrupting the program.

SUMMARY

The IrDA Control specification provides an environment for new cordless input devices to simplify current home and office applications.

The connectors for many devices are located in an inconvenient position. Use of IrDA Control provides for cordless devices that can communicate with the front of a product without the need for inconvenient cords or extension cables, and the associated storage clutter.

Many of the implementation factors are soon to be provided in standard products, simplifying development of cordless products.

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

Reference Code SMA99056