About IrDA

Software Protocol

(the following text was taken from the IrDA Web Site: www.irda.org.)

Table of Contents

IrDA Protocol Stack

IrDA's stated purpose is ”to create an interoperable, low cost, low power, half duplex serial data interconnection standard that supports a walk–up, point–to–point user model that is adaptable to a wide range of appliances and devices”. IrDC will soon offer lower costs than current wire/cable/connector setups thus becoming widespread. If it works over a wire/cable, it will be capable of working over the IrDA infrared data link.

The IrDA protocol stack provides hardware and software architecture guidelines for designing an IrDA compliant system. Figure 1 shows the different layers in the stack. As a minimum, compliance to IrPHYS, IrLAP, and IrLMP is required. Optional sections of the protocol include the Higher Speed Extensions (1.15 & 4 Mbit/s), IrCOMM, TinyTP & PnP.

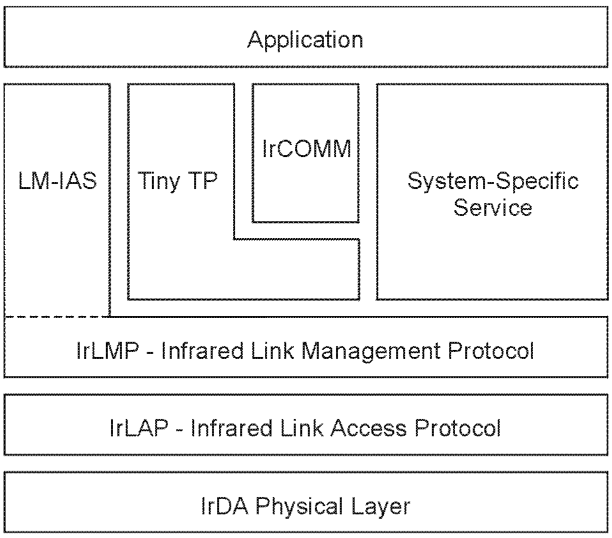


Figure 1. IrDA Protocol Stack

IrDA Link Access Protocol (IrLAP)

IrLAP, is derived from an existing asynchronous data communications standard (an adaptation of HDLC). It provides guidelines for link access in which the software searches for other machines to connect to (sniffing), recognizes other machines (discovery), resolves addressing conflicts, initiates connection and information exchange, and manages disconnection. During data transfer, IrLAP is responsible for providing reliable error detection, retransmission, and flow control. While the discovery and address conflict resolution procedures are somewhat unique to IrLAP, the link initialization/ shutdown, connection startup, disconnection and information transfer procedures all resemble similar operations in HDLC protocols and are adapted to the IrDA serial infrared environment.

A link operates essentially as follows:

A device (primary) will want to connect to another device (either by automatic detection via the discovery and sniffing capability of IrLAP, or via direct user request). A data link involves at least one primary (commanding) station and one or more secondary (responding) stations, whose roles IrLAP has the responsibility of managing. The primary station has responsibility for the data link. All transmissions over a data link go to or from the primary station and can be point–to–point or point–to–multi–point. There is always one and only one primary station while all other stations must be secondary stations.

After obeying the media access rules the primary will send connection request information at 9600 bit/s to the other device, this data will include information such as its address and its other capabilities such as data rate, etc. The responding device will assume the secondary role and, after obeying the media access rules, return information that contains its address and capabilities. The primary and secondary will then change the data rate and other link parameters to the common set defined by the capabilities described in the information transfer. The primary will then send data to the secondary confirming the link data rate and capabilities. The two devices are now connected and the data is transferred between primary and secondary under the complete control of the primary. Rules are defined which ensure that the secondary and primary are both able to efficiently transfer data.

Any station that is capable can contend to play the primary station role. The role of primary is determined dynamically when the link connection is established and continues until the connection is closed (the exception is that there is a method provided for a primary and secondary on a point to point link to exchange roles without closing the connection).

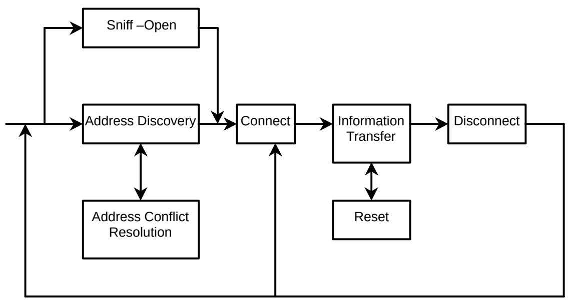


Figure 2. IrLAP Operation Procedures

IrDA Infrared Link Management Protocol (IrLMP)

Once a connection is established, IrLMP manages the available functions and applications between the communicating devices. IrLMP determines the capabilities of the connected devices (discovery) and manages negotiation of link parameters and correct data transfer (link control). The link management framework also provides a means for a device to establish coexisting access to an IR connection between multiple devices (multiplexing).

Discovery occurs when two devices first encounter each other. Each service and each protocol on a device will have registered with the link management. The information registered includes both standard and protocol specific information. An application can query the capabilities of devices within range.

Once an application on one device has determined which service or protocol it wishes to use, it requests the link control to use the protocol. The link management framework allows multiplexing of application or transport protocols on the same link connection at the same time.

Multifunctional devices may have requirements for the IR link to support concurrent activities. While the IrDA Link Access Protocol, IrLAP, provides a single connection between a given pair of IrDA compliant devices, it provides no means for multiple application components to share coexisting access to that connection.

A useful feature set of capabilities would have the mobile user wishing his handheld/PDA or note–book computer to synchronize systems, check/send/receive electronic mail, reconcile scheduling & address book (PIM) data, and also initiate deferred printing when it makes contact with the owner's desktop personal computer (PC) and network gateway. Each activity may be controlled by distinct application components within each device (mobile unit and PC) and each component has a separate requirement to locate its peer component and establish the relevant connection.

IrDA Link Management Protocol, IrLMP, is a mandatory component of IrDA compliant devices and addresses both the previous needs mentioned.

- Connection multiplexing services provided by IrLMP Multiplexer (LM–MUX).
- Location of peer application components supported by IrLMP Information Access Service (LM–IAS)

IrLMP Multiplexer (LM–MUX) provides multiplexed channels on top of an IrLAP connection. Each of these so called Link Service Access Point or LSAP–connections (distinguished from IrLAP connections) carries a logical separated information stream. The multiplexer operates in two modes. In multiplexed mode, the multiplexer will accept data from any of its clients (transport entities or directly bound applications). In exclusive mode a single LSAP–connection claims access to the IR medium.

IrLMP Information Access Service (LM–IAS) is a direct client of LM–MUX and operates in a 'client server' manner. Application components that wish to make themselves accessible to peer components in other devices 'describe' themselves by creating an object whose attributes carry the essential parameters required to establish a connection between the two peers. An application component seeking a peer application component inspects the objects within the remote LM–IAS information base. Assuming a suitable object exists, the connection parameters and the mechanism are retrieved, which is accessible at the top of the LM–IAS.

MUX is used to enumerate the devices available. The resulting list of devices may be sorted/prioritized on the basis of the information hints supplied in discovery responses.

IrDA Physical Layer (IrPHY)

The Physical Layer provides guidelines for the physical connection of equipment using IR and defines the specifications which infrared transceivers must be able to meet. The specifications of the Physical Layer were already described in the relevant chapter before.

IrDA Flow Control Mechanism (Tiny TP)

Whilst IrLAP provides a flow–control mechanism between peer IrLAP entities, the introduction of multiplexed channels above IrLAP by IrLMP LM–MUX introduces a problem. Reliance on IrLAP to provide flow-control for a multiplexed channel can result in dead-locks if the consumption of data from one multiplexed channel is dependent on data flowing in an adjacent multiplexed channel. Conversely, if inbound data on a multiplexed channel cannot be consumed and the underlying IrLAP connection cannot be flow–controlled off due to the possibility of deadlock, inbound data (freshly arrived or buffered) must be discarded in the event of buffer exhaustion. Sadly this reduces the reliable delivery service provided by IrLAP to a best effort delivery service provided by IrLMP LM–MUX (when multiple multiplexed channels are in operation).

There are at least two possible solutions for restoring a reliable delivery service above IrLMP LM–MUX.

- Provide a per application stream flow–control mechanism above LM–MUX between peer application entities. This ensures that there is always sufficient buffer space available to accommodate in–bound application data.
- Provide a per application stream retransmission mechanism above LM–MUX that recovers from the loss of data that arises if inbound buffering become exhausted.

The Tiny TP protocol provides:

- Independently flow controlled transport connections
- Segmentation and re–assembly

IrDA Serial & Parallel Port Emulation (IrCOMM)

IrCOMM defines the emulation of Serial and Parallel ports over the IrLMP/IrLAP protocol stack. The motivation for IrCOMM comes from the many printing and communication applications which use standard communication APIs to talk to other devices via serial and parallel ports. By making IrDA protocols accessible via these APIs, many existing applications including printing can run over an IrDA infrared link without change. This intent to support so–called legacy applications is the basis for IrCOMM. New applications are encouraged to take better advantage of IrDA protocols by using their capabilities directly.

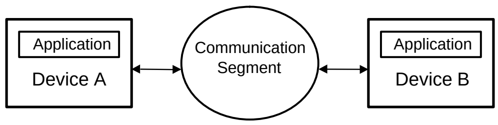
Emulating COMM ports raises a number of questions, starting with what kinds of ports will be emulated. IrCOMM emulates RS–232 (EIA/TIA–232–E) serial ports, and Centronics parallel ports like those found on most personal computers. The four service types used to emulate these ports are the core of this specification. Before discussing service types, however, there are some basic differences between wired and IrDA communications to consider.

Wired communications methods can send **streams of information in both directions at once,** because there are multiple wires (some to send data on, some to receive data from). With infrared, there is the equivalent of only one wire (the IR path through the air). This has the following implications:

- IrDA protocols send **packets one way at a time**. If a device tried to send data and listen for data at the same time, it would ”hear” itself and not the device it wants to communicate with. The way IrDA devices achieve two way communications is to take turns, also known as ”turning the link around”. This happens at least every 500 ms, and can be made more frequent as necessary. This latency makes it impossible to perfectly emulate the wired COMM environment – very timing sensitive operations will be disrupted. Fortunately, many communication tasks are not so sensitive, and can therefore use IrCOMM.
- All of the information carried on multiple wires must be carried on the single IR “wire”. This is accomplished by subdividing the packets into data and control parts. In this way a logical data channel and control channel are created, and the various wires can be emulated.

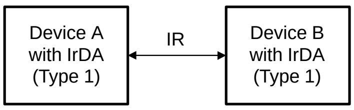
On a different level, IrCOMM is intended for legacy applications, applications that know about serial or parallel ports but know nothing about IrDA protocols. IrDA protocols, however, have very different procedures and APIs from wired COMMs. Suppose, for example, a word processing application wants to print via IR using IrDA protocols, an application must first ”discover” the printer (locate a printer in IR–space), then check the printer’s IAS to find information needed to connect. Since the word processing application (a legacy application) knows nothing about this, IrCOMM maps these operations into normal COMM operations so that it is completely transparent.

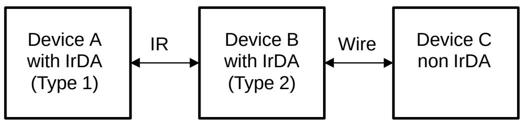
For the purposes of IrCOMM a complete communication path involves two applications running on different devices (the communication endpoints) with a communication segment between them. The communication segment may consist solely of IR or IR connections to a network. The figure below shows the complete communication path.



IrCOMM is intended to cover applications that make use of the serial and parallel ports of the devices in which they reside. In the simple case, the communication segment is an IR link from one device to another (direct connect). In the case where the communication segment is a network, IR is used for the path between the device and a networking connection device like a modem. Modems communicate through the network using wire, radio or IR. IrCOMM is only concerned with the connection between devices in the direct connect case or between the device and a modem in the network case. There are other configurations that IrCOMM can support such as modules that communicate via IR on one side and provide a wired interface on the other side. These devices are not really modems but offer a similar service and thus, are not explicitly discussed.

Basically two device types exist that IrCOMM must accommodate. Type 1 devices are the communication end points such as computers and printers. Type 2 devices are those that are part of the communication segment such as modems. Though IrCOMM does not make a distinction between these two device types in the protocol, accommodating both types of devices impacts the IrCOMM protocol. The figures below illustrate how the two IrCOMM device types fit into communication paths.





The information transferred between two IrCOMM entities has been defined to support both type 1 and type 2 devices. Some information is only needed by type 2 devices while other information is intended to be used by both. In the protocol no distinction is made between type 1 and type 2 therefore, it is up to the IrCOMM implementor to determine if the information passed in the IrCOMM protocol is of use to the implementation. Since the devices do not know the type of the other device in the communication path, they must pass all the information specified by the protocol of which they have knowledge.

IrCOMM emulates serial and parallel ports. However, printing and communications applications use communication ports in a variety of ways. To address this need, IrCOMM provides four service types or classes: 3–Wire raw, 3–Wire, 9–Wire, and Centronics. The service types fall into 2 camps, which are called raw and cooked; the differences hinge on whether a control channel is supplied and the type of flow control used. 3–Wire raw provides a data channel only, and uses IrLAP flow control. The ”cooked” service types (3–Wire, 9–Wire, and Centronics) support a control channel, and employ Tiny TP flow control.

3–Wire raw – This service type can be used for serial or parallel emulation when a single exclusive connection is acceptable, and only the data circuits need to be emulated (no non–data circuits in a corresponding wired setting carry any information). The name of this service comes from the notion of emulating the minimum three RS–232 circuits required for full duplex communications. The circuits are shown below.

CCITT	Signal Description	Comment
102	Signal Common	This circuit is not needed for IR but is shown because it is one of the circuits that drove the definition of the name.
103	Transmitted Data (TD)	This connection carries data transmitted by the DTE
104	Received Data (RD)	This connection carries data received by the DTE

Here are the main attributes of the 3–Wire Raw service class:

- **Only one non–IAS IrLMP connection can be open if 3–Wire raw is used** – all other connections must be closed before it can be established, and others must wait until the raw connection is closed before they can connect. This is because 3–Wire raw uses the flow control features of IrLAP, which can result in a deadlock condition if more than one non–IAS connection is open.
- **Minimal Implementation.** All the IrCOMM data is sent directly via IrLMP in IrLMP packets. All data that follows the IrLMP Mux bytes in an IrLMP packet is IrCOMM data, (i.e., it is the information that would travel over the data line(s) of a wired interface). No control channel is available to communicate information about the state of other leads (e.g., RTS/CTS), software flow control settings, and the like. A service which employs 3–Wire raw must be able to do without that information. The link is merely a raw channel for the movement of data.
- **This service can be used to emulate both serial and parallel ports.** This may seem counter–intuitive (who has ever heard of a 3–Wire parallel port?), but if you remove the non–data circuits (which 3–Wire raw does not emulate), serial and parallel are equivalent, i.e., just streams of data.

IrLPT is an IrDA service in use on commercially available printing devices. It is equivalent to 3–Wire raw in functionality, but is slightly different in how it uses the IAS.

3–Wire – Like 3–Wire raw, the name of this service comes from the minimum three RS–232 circuits required for full duplex communications. Like 3–Wire raw, it is intended for both serial and parallel ports. However, there are the following important differences to be noted:

- 3–Wire service class makes use of Tiny TP flow control, so that it may coexist with other connections that employ higher level (not IrLAP) flow control (including other cooked IrCOMM connections). It is not limited like 3–Wire raw to a single IrLMP connection.
- 3–Wire service class supports a control channel for sending information like data format. The control channel mechanism is described in the chapter titled Frame Formats and the Control Channel.
- Because of the need for flow control and the use of the control channel, the 3–Wire service type uses a more elaborate frame format.

9–Wire – The name of this service class comes from the notion of emulating the 9 circuits of an RS–232 interface which are part of a standard IBM compatible PC. Unlike the previous services it is true to its name; 9–Wire emulates serial ports only. Three of the circuits are the same as described in the 3–Wire service classes. The other six are listed below.

CCITT	Signal Description
105	Request to Send (RTS)
106	Clear to Send (CTS)
107	Data Set Ready (DSR)
108	Data Terminal Ready (DTR)
109	Data Channel Received line signal detector (RLSD), aka Carrier Detect (CD)
125	Calling indicator, aka Ring Indicator (RI)

Some attributes of this service are listed below.

- Like 3–Wire, it uses the Tiny TP flow control mechanism. It also uses the same control channel mechanism for sending information like data format.
- The control channel is used to send the states of the other RS–232 leads as they change.

Centronics – This service is intended to emulate the function of a standard Centronics interface. This service is for parallel ports only. Some attributes of this service are listed below.

- It uses the Tiny TP flow control mechanism.
- It uses the same control channel mechanism used in 3–Wire to send the status/changes of the additional circuits.

IrDA Object Exchange Protocol (IrOBEX)

One of the most basic and desirable uses of the IrDA infrared communication protocols is simply to send an arbitrary ‘thing‘, or data object, from one device to another, and to make it easy for both application developers and users to do so. We refer to this as object exchange (uncapitalized), and it is the subject of the protocol IrOBEX (for IrDA Object Exchange, OBEX for short). OBEX is a compact, efficient, binary protocol that enables a wide range of devices to exchange data in a simple and spontaneous manner. OBEX is being defined by members of the Infrared Data Association to interconnect the full range of devices that support IrDA protocols. It is not, however, limited to use in an IrDA environment.

OBEX performs a function similar to HTTP, a major protocol underlying the World Wide Web. However, OBEX works for the many very useful devices that cannot afford the substantial resources required for an HTTP server, and it also targets devices with different usage models from the Web. OBEX is enough like HTTP to serve as a compact final hop to a device ”not quite” on the Web.

A major use of OBEX is a ”Squirt” or ”Slurp” application, allowing rapid and ubiquitous communications among portable devices or in dynamic environments. For instance, a laptop user squirts a file to another laptop or PDA; an industrial computer slurps status and diagnostic information from a piece of factory floor machinery; a digital camera squirts its pictures into a film development kiosk, or if lost can be queried (slurped) for the electronic business card of its owner. However, OBEX is not limited to quick connect–transfer–disconnect scenarios – it also allows sessions in which transfers take place over a period of time, maintaining the connection even when it is idle.

PCs, pagers, PDAs, phones, auto-tellers, information kiosks, calculators, data collection devices, watches, home electronics, industrial machinery, medical instruments, automobiles, pizza ovens, and office equipment are all candidates for using OBEX. To support this wide variety of platforms, OBEX is designed to transfer flexibly defined ”objects”; for example, files, diagnostic information, electronic business cards, bank account balances, electrocardiogram strips, or itemized receipts at the grocery store. ”Object” has no lofty technical meaning here; it is intended to convey flexibility in what information can be transferred. OBEX can also be used for Command and Control functions – directives to TVs,VCRs, overhead projectors, computers and machinery.

OBEX consists of two major parts: a model for representing objects (and information that describes the objects), and a session protocol to provide a structure for the ”conversation” between devices. OBEX is designed to fulfill the following major goals:

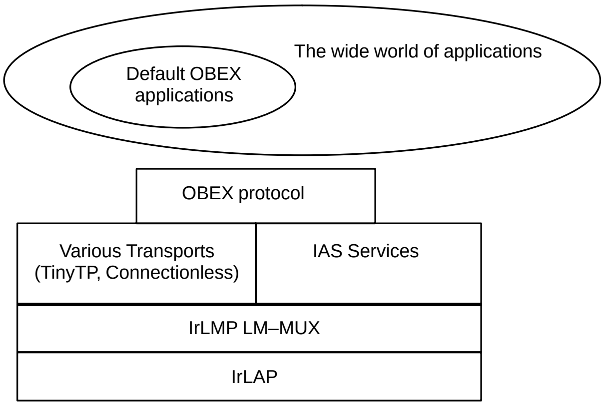


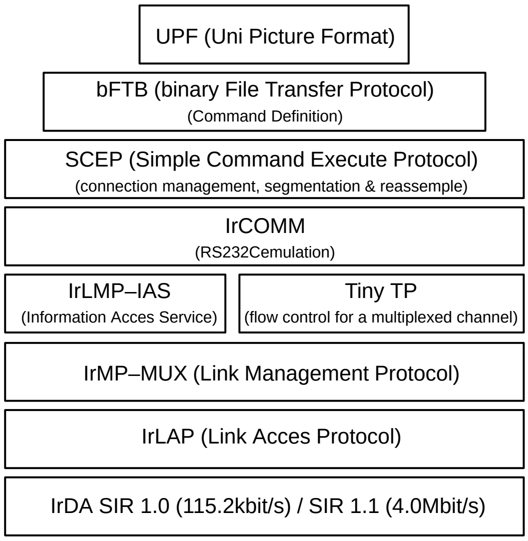
Figure 3. OBEX in the IrDA architecture

- Application friendly – provide the key tools for rapid development of applications
- Compact – minimum strain on resources of small devices
- Cross platform
- Flexible data handling, including data typing and support for standardized types – this will allow simpler devices for the user through more intelligent handling of data inside.
- Maps easily into Internet data transfer protocols
- Extensible – provides growth path to future needs like security, compression, and other extended features without burdening more constrained implementations.
- Debuggable

IrDA Infrared Transfer Picture (IrTran–P)

IrTran–P(Infrared Transfer Picture) is an image communication scheme for a digital camera based on the Infrared Communication Standard specification created by IrDA. The IrTran–P specification should generally be used together with the IrDA standard specifications.

IrTran–P is placed on the upper layer of IrSIR, IrLAP, IrLMP, TinyTP and IrCOMM which is already established as IrDA standard specifications. SCEP (Simple Command Execute Protocol) and a bFTP (Binary File Transfer Protocol) are necessary for exchanging an image between devices and mutually exchanging properties of the devices. An image format (file) called UPF (Uni Picture Format) is exchanged on such an entity (UPF is the image format out of the category of IrDA, and can be found in the appendix of the IrTran–P specification). IrTran–P is a generic name given to all of these components.



SCEP establishes a session on IrCOMM and provides a transparent session which notifies an upper layer of a command. The procedure of SCEP is developed by a lower layer making use of an advantage that an IrDA protocol is “error free”, as a high speed session layer matching the IrDA protocol.

As is apparent from its name, bFTP provides a service for transferring a binary file. The bFTP assumes a virtual file system together with a communication protocol. The bFTP has an aspect which enables it to be easily implemented, because it assumes such a simple file system that will allow “a binary file to be stored with its name”.

Moreover, bFTP is characterized by a query function which allows it to query functions and properties of a device and the image format available in the theme of this section, i.e., the image transfer. This query function simplifies the user interface of a digital camera, and allows the most suitable data of an image to be transferred between the digital cameras or printers facing each other. In addition, this function makes it possible for the user to transfer, communicate or print suitable image data regardless of the difference in platform or model just by “selecting a photograph to be sent and pushing a transmission button”.

As mentioned earlier, UPF is the standard of an image format not included in the category of the IrDA standard. The IrDA standards were originally provided for defining and standardizing a protocol in connection with infrared communications. Therefore, it is out of the scope to define the contents of an image transfer. However, in order to ensure mutual connectivity as an application of a digital camera, it is required to decide an image format so that image data sent via infrared communication is reliably reproduced. Therefore, in advocating IrTran–P as a standard to IrDA, the specific contents of an image format of IrTran–P are defined and described in an appendix.

UPF is an image file format based on the JPEG base line. JFIF, which is a JPEG file, makes an image of various color forms available and employs a high level of compression scheme. For this reason, JFIF may be regarded as the industry standard of an image file format today. Since JPEG is a format enabling a variety of color forms, a compromise is required to some extent in order to realize the standard at a low cost, for example adopting only a part of the format as the standard. In UPF, among the formats included in the base line of JPEG, the format which allows the devices at least to reliably display and mutually transfer an image is defined as an indispensable one, and others are regarded as an option. For more details, please refer to the sections of UPF in the later part of this document.

It is characteristic of a digital camera that all the data accompanying a photograph taken by a digital camera, such as a photo–taking date/time and the orientation (direction) of an image and other additional data, cannot be covered by the data within the JPEG format. In view of such a background, UPF is designed so that data is separated and stored on its own header arranged in the file without changing the image data scheme of JPEG Base Line at all. In addition, the header has expandability and allows a vendor–unique function to be added thereto. This makes it possible to separate the data necessary for a digital video camera from the data necessary for display and expansion of an image. This is advantageous in that the existing JPEG techniques can be used where they stand and do not have to be changed. In a compact device like a digital camera, when using existing hardware or software, e.g. in the case where an algorithm of JPEG compression/expansion or the like is performed by hardware or is fixedly used as firm ware, it is undesirable to change JPEG itself.

As a further advanced step, UPF is designed so that additional data on an ambiguous point within the data of JPEG scheme is arranged in the header part. The additional data includes factors such as white level, black level and color–difference signal, necessary for reproducing an image with correct brightness and color.

Although the format of a digital camera is being examined by various organizations, a conclusive decision has not yet been made. In most cases, an arrangement such as a new addition of a tag to JPEG or the like has been proposed. However, it will take a long time to reach a conclusion satisfactory to all the companies concerned. The approach of making the best use of an existing standard, wherein the data necessary for a digital camera is separated and added so as to assure expandability, is more realistic than the approach of waiting for a new standard to be defined and finalized.

Though UPF is defined as an appendix, it is indispensable in order that the IrTran–P standard is able to support an image format of a UPF scheme.

In IrTran–P, a sender starts an operation which transfers picture data from a digital camera.

- **Operation by User**

With the use of ”selection of a specific picture” and a ”transmission button” the user puts the digital camera of the sender into a transmission state.

It is assumed that the device of a receiver is always in a receiving state or put info a picture data receiving state by a ”reception button”.

- **Establishment of Session by SCEP**

The digital camera of the sender carries out a discovery procedure using IrDA protocols and performs a connection for the physical to IrCOMM layers of IrDA protocols in accordance with IrDA protocols. When an IrDA transmission path is established, SCEP makes a ”session establishment request” from the sender to the receiver of a digital camera, printer or PC. If the receiver is implemented with SCEP, it must make a response of either ”session established” or ”session establishment rejected”.

- **Query Operation by bFTP (Query function)**

When a session by SCEP is established, the digital camera of the sender issues a Query request in order to recognize the picture processing functions of the receiver. The information mutually exchanged by the Query request includes the transmittable / receivable picture size, the picture compression format and the basic picture size of the device. Since this information is exchanged before transfer of picture data, the picture data can be transferred in ”the most reasonable format” between devices of different platforms.

In IrTran–P, a ”mandatory format” is defined among the picture data formats of both sides, whereby a picture can be reliably exchanged between devices of different grades or manufacturers.

Furthermore, the power supply condition of the device, the receivable data capacity etc. should be queried. This makes it possible to deal with the applications of a portable system.

- **Transfer of Picture Data by bFTP**

Transfer of picture data is started once the most appropriate picture format for both the sender and the receiver has been determined by Query. SCEP performs the data transfer at a high transmission rate by making use of IrDA protocols. After the file transfer is completed, next picture data may be subsequently transmitted, or a session may be disconnected by SCEP. (Accordingly, even a simple model can transmit more than one picture in succession.)

Completion of Session by SCEP

When the picture transfer has been completed, the digital camera of the sender disconnects a session by SCEP. Thereafter, a disconnection request is issued for IrCOMM and lower layers of IrDA protocols and the picture transfer operation has been completed.

Appendix

List of Terms and Abbreviations:	
IrDA	Infrared Data Association
IrLAP	IrDA Link Access Protocol
IrLMP	IrDALink Management Protocol
IrPHY	IrDA Physical Layer
Ir TRAN–P	IrDA Transfer Picture
IrCOMM	IrDA ”Serial & Parallel Port Emulation over Ir”
IrOBEX	IrDA Object Exchange Protocol
Tiny TP	a Flow control mechanism for use with IrLMP
LM–MUX	IrLMP Multiplexer
LM–IAS	IrLMP Information Access Service
SCEP	Simple Command Execute Protocol
BFTP	binary File Transfer Protocol
UPF	Uni Picture Format
SIR	Serial Infrared
FIR	Fast Infrared