



A Bluetooth Digital Camera Appliance

Alan McReynolds, John Recker, Jean Tourrilhes, Venky Krishnan

HP Laboratories
HPL-2003-265R1

Keyword(s):

Bluetooth; camera; upload

Abstract:

The digital camera's role in the photographic process is still largely limited to acquiring photographic images. A wireless link associated with software services on an access point allows a camera to finally supercede the "film roll" processing model. Adding remote services and a local cache of images and metadata can transform the camera into a full-fledged imaging appliance.

External Posting Date: March 21, 2013 [Fulltext] Approved for External Publication
Internal Posting Date: January 7, 2004 [Fulltext]

The camera as a photo appliance

With the exception of being able to immediately review pictures taken, digital cameras have yet to make a real break with the film roll model. Unloading the camera is still a manual step. The user is responsible for being conscious of how many shots can be stored in the camera's memory and needs to remember, at the correct time, to transfer the pictures. To be sure, many clever techniques have been employed to simplify the extraction of images from the camera: Memory cards, cables, disks, docking stations, and now wireless links. All of these systems depend on the user to initiate the transfer. Some current cameras do allow the user to specify "intents" for images currently stored on the camera that can direct third party processing of images as they are uploaded. In general, however, once an image has been removed from the camera, the camera no longer plays a role in the image handling process.

Our system, in contrast, assumes that a camera operates as a full-fledged peer in a photo-processing environment. Rather than requiring the user to manually unload the camera, the camera actively, and transparently to the user, attempts to communicate and synchronize its state with the environment.

As the opportunity arises, pictures are automatically delivered to the users designated location for data. Metadata, including intents, are synchronized as a part of the communication. Intents, such as sharing or auto print, are then processed in the background. Metadata retained on the camera, including cached and thumbnail images, allows the user to navigate and work with their photos independently of the background infrastructure. Metadata synchronization further allows the user to take advantage of album choices and organization applied at other stations (e.g. PC).

The local metadata contains a storage status for each picture. With this information, the camera can make full use of the local storage media by retaining a cache of uploaded pictures locally. When the camera needs space for a new picture it deletes the oldest (already uploaded) picture from memory. This allows the user to review, print, display or share more pictures, further enhancing the digital imaging experience.

Our wireless camera model is thus one of "hands off" background image and metadata synchronization. Our term for this is "Casual Upload". The camera is responsible for determining whether connectivity is needed and when it is most likely to be able to make a connection. The current implementation attempts to

establish a connection whenever new pictures are in memory. More sophisticated heuristics, involving environmental sensors, are the subject of a current invention disclosure. The key feature is that the connectivity occurs in the background without the need for direct user intervention.

Background

The Family Data Center [1] (FDC) provides an always-on local infrastructure for information appliances. The FDC can host applications as well as proxy communications with other Internet devices including other FDCs. Locally the FDC provides Bluetooth [4] and 802.11x wireless connectivity.

Several appliances have been built as part of the Family Data Center/HotSpot project, including: the “BuzzBox”, a streaming music box; and a photo/Bluetooth iPaq. The Bluetooth Camera is another example of an FDC appliance. This paper describes our prototype Bluetooth camera. It was designed as a demonstration platform for imaging appliance concepts.

Current HP camera models, including the HP 812, support “intents”. Essentially these are bits of metadata linked with the locally stored images that allow the photographer to indicate whether the image is to be printed or shared. We want to expand the concept of intents to include operations on images stored in the user’s image repository (FDC). The first step in that process is to create a mechanism for synchronizing camera metadata and images in the background. The background or “casual upload” model of camera connectivity is the primary focus of this document.

Connectivity choices

Three wireless choices are at the forefront today: 802.11b, Bluetooth and the GSM/3G cell infrastructure. A brief comparison:

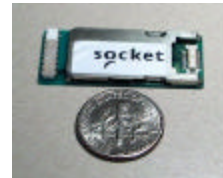
	Bluetooth	802.11b	GSM / 3G
Parts cost	\$5	\$10	???
Airtime cost	Free	Free	High (\$'s/MB)
Power	400mW peak	1W peak	High (???mW)
Throughput	500kb/s	5Mb/s	20-200kb/s
Connection options	Access Point / PC / Printer	Access Point/PC	Everywhere
Discovery Delay	~7s	~1s	0
Implementation Complexity	Medium	High	Medium

We chose Bluetooth for several reasons: Low cost, low power, simple connectivity, and interoperability with Bluetooth printers and desktops. The background data synchronization nature of our design does not need the speed and protocol options offered by 802.11b. The application would benefit greatly from the wide availability of GSM / 3G services, however the high price rules out this option. If cell service companies were to offer a deeply discounted rate (cents/MB) for background or off-peak data transfer, the ranking would change dramatically.

The hardware platform

We used an off-the-shelf HP 812 camera as our starting point. The camera uses a Motorola ColdFire CPU core included on the Agilent (formerly HP) Bedrock imaging chip. The manufacturing division provided us with the software development environment for the camera.

For communications we embedded a Socket Communications PNC Module with an integrated antenna in the camera body. Since the module is located inside the camera and the radio signals need to get through the case we ground off a portion of the camera's RFI shielding.



The BT module was powered from the camera's Bedrock CPU power supply. Thus it draws power whenever the camera is "on". There were no I/O lines free for use so power control could not be implemented. A production version would need to conserve power by switching power to the BT module. Control

over transmit power level would also be desirable, particularly in the direct print scenario where it is beneficial to limit communications to very nearby printers.

The docking station serial control lines were reused to provide connectivity between the CPU and BT module. The optimal serial port speed was found to be 998400 baud. This rate is divisible by both the camera and module clock systems with a clock offset error of .01%, well within the tolerance of the UARTs.

The software platform

The HP 812 uses the VxWorks operating system. Interfacing the Bluetooth (BT) module to the VxWorks OS required the integration of a BT communications stack. HPL has a license for the Extended Systems BT stack. The stack provides RFCOMM functionality and OBEX.

The first part of the effort was to port the Extended Systems stack to VxWorks. Although a preliminary port is provided with the stack, significant effort was required to optimize the existing serial port drivers for both speed and reliability. The original docking station control protocol operated at 9600 baud. With such a low rate, little effort had been put into serial port handling efficiency. The serial driver had fairly lengthy code paths for processing serial data as well as some accumulated legacy code from earlier camera designs. At nearly 1Mbaud, the overhead of handling characters consumed enough CPU cycles that the BT stack would lose bytes of incoming data. The serial BT protocol doesn't recover from data loss. By fully utilizing the FIFO, streamlining code and caching values returned from heavyweight calls, enough cycles were spared to allow the driver to operate. Some packets are still lost, most probably due to high priority interrupts coming from within the HP 812 camera application. The chosen solution was to improve the speed of recovery.

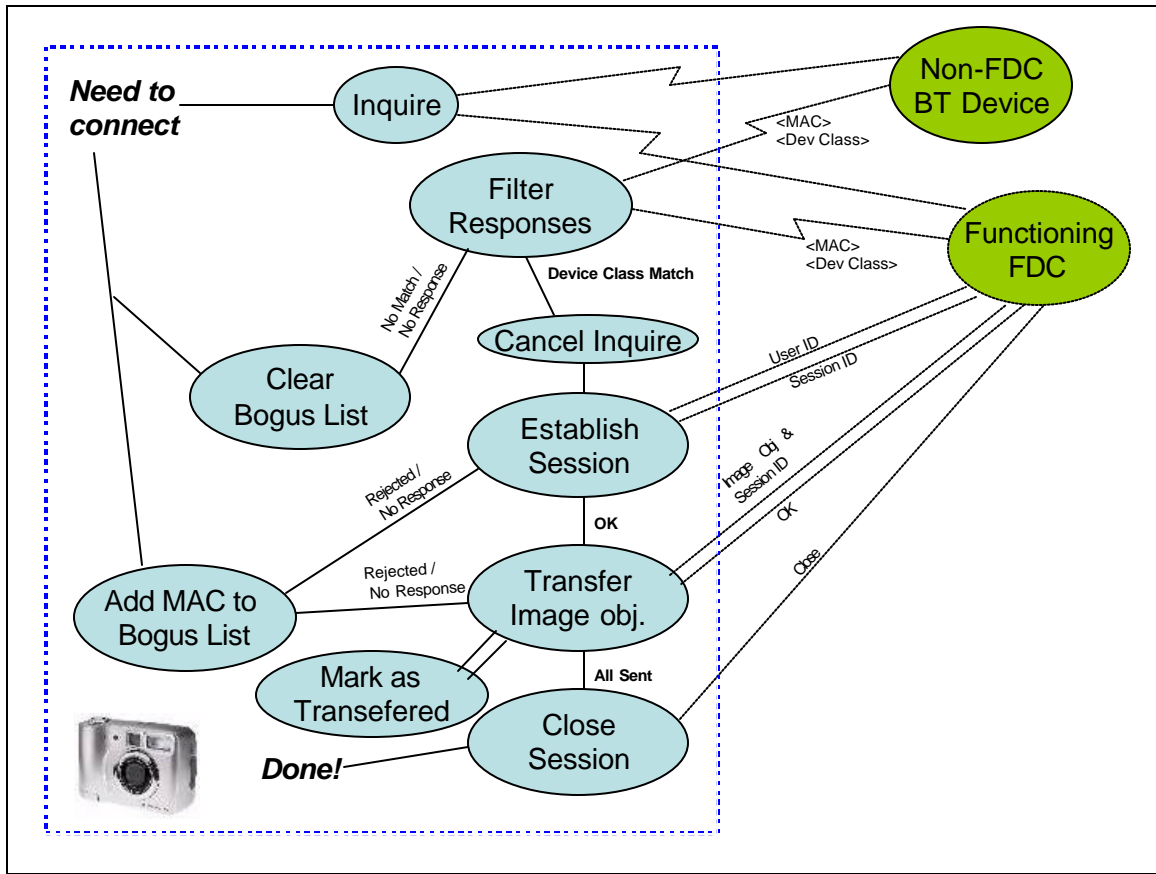


Fig. 1 Communications State Diagram

The camera is portable and so it may rapidly pass through the radio sphere of an FDC. To maximize the chance of successfully exchanging data objects it must rapidly identify and connect to suitable BT hosts. Once a new picture has been taken, the camera begins a periodic Bluetooth inquiry for a suitable upload connection. (Fig. 1) The camera searches for a hierarchy of connection options. First, the camera searches for a Bluetooth LAN access point. LAN access points are identified by Bluetooth in the device class information acquired as part of the discovery process. Next, the camera performs a Bluetooth Service Discovery Protocol (SDP) query to determine if the LAN access point supports the image data/metadata synchronization services, identified by a unique SDP UUID, supported by the FDC. As SDP queries require establishing a Bluetooth connection to a remote device and are thus relatively heavyweight operations, the device address of Bluetooth MAC IDs found to not meet the above criteria are stored in a table to speed subsequent searches and avoid looping on a defective device. Using this algorithm, once an acceptable access point is in range, the camera can typically initiates an image transfer in less than 7 seconds.

At a throughput of roughly 1MB/25s the camera can unload several pictures during the 100 seconds or so that it takes to walk across an FDC radio sphere. Since image delivery is a background process, completely unloading the camera is not important.

Communications Protocol

As taking pictures is an intermittent operation, an ongoing network connection is not needed. We thus chose to use the transaction oriented IRDA Object Exchange protocol[3] (OBEX) to synchronize images and metadata as individual objects. This same protocol used by many other Bluetooth devices including HP's Bluetooth enabled printers.

OBEX ⇔ HTTP Proxy

Though OBEX is the native protocol for the BT camera and also the BT printers. The FDC and other web enabled devices use HTTP to handle and forward transactions. One solution would be to enhance each target application with an alternate OBEX interface. That would be expensive and not very extensible.

Instead, an OBEX⇔ HTTP proxy was implemented in the FDC. The proxy we implemented has no application awareness. It does not alter the meaning of the requests, instead it translates each request so that it can be understood by the other side. The original CoolTown proxy code [5] was enhanced with BT OBEX support and support for posting binary data.

HTTP GET mappings		
HTTP GET operation	⇔	OBEX GET command
HTTP path/URL	⇔	OBEX Name header
HTTP Accept: header	⇔	in OBEX HTTP header
HTTP POST mappings		
HTTP POST operation	⇔	OBEX GET command
HTTP path/URL	⇔	OBEX Name header
HTTP Accept: header	⇔	in OBEX HTTP header
HTTP Content-Type: header	⇔	OBEX Type header
HTTP payload	⇔	OBEX Body header
HTTP Reply mappings		
HTTP status code	⇔	OBEX Response code
HTTP Content-Type: header	⇔	OBEX Type header
HTTP payload	⇔	OBEX Body header

Fig. 2 Proxy Protocol Mapping

The Proxy can interpret the OBEX Name header as a relative path or a fully qualified URL. If the OBEX Name header starts with the string "http://", the proxy assumes it is a URL and will query this URL directly. Otherwise, it will assume it is a relative path and will prepend "http://127.0.0.1:80/" to it (i.e. it will query the local web server for this path).

Image data transfer

Image upload is a three part operation. A connection with the FDC image server is established and a user profile is sent to the server to initiate a session. Currently this profile is stored in the camera file system, and is a placeholder for the robust end-to-end authentication system [2] demonstrated on other FDC prototypes. The FDC returns a session identifier, which is used the remaining transaction steps.

In the current implementation of the camera, the camera then sends each picture (along with the session ID) as separate OBEX GETs to the FDC. Upon successful object transfer the local copy of the picture is deleted. Future versions will maintain local thumbnails and synchronized metadata. The FDC authentication model would be implemented to ensure end-to-end connectivity and security. Finally, the FDC web service is notified to terminate the session.

Performance of Bluetooth component

Connection to FDC: ~7 seconds

Upload of picture: ~25seconds/MB of image data

Power consumption: ~120mA @3.3V max (BT module current only)

Transfer performance on the development platform is limited by the use of a UART data line between the camera and Bluetooth device. The interrupt handling overhead causes the module to reduce the over the air packet size. This in turn increases the amount of protocol overhead transmitted. Switching to a more efficient communication method with the module, for example USB, could result in a near doubling of throughput. Doubling the speed would not lead to a significant increase in power consumption. Or, in other words, the cumulative power required to transmit a megabyte will drop when the speed is increased.

Usage

User convenience is the chief benefit of the Casual Upload model. Aside from initially storing user identification in the camera, no user intervention is required. The user simply snaps pictures and stows the camera in their pocket or purse. Uploads occur unseen in the background, so the scheme is relatively insensitive to connection quality or speed. The user experience is that this style of digital camera, while otherwise normal, never runs out of "film". Through the use of caching, the user has greater control and access to their photos, directly from the camera.

The FDC (or other service provider) needs to keep metadata in order to organize and manage the stored images. By synchronizing this metadata and associated photo thumbnails with the camera one can add more appliance-like features. Using the camera interface one could control the printing, sharing and publishing of photos with third party devices and services. These services would be functional for all photos referenced in the metadata, not just those physically resident in the camera.

While one might not want to organize one's pictures using the camera's tiny screen, it might be desirable to, for example, use the camera to direct the printing of a photo collection to a local printer. Similarly, one might want to direct that pictures gathered on a trip be shared with others on the trip. Later when connectivity is available, the photos would then be sent not only to the user's home location, but also to the buddy list from the trip. No direct connection in space or time is required between the buddies.

References

- [1] [HPL-2002-134 HotSpot! -- A Service Delivery Environment for Nomadic Users System Architecture](#) - Das, Devaraj; Manjunath, Geetha; Krishnan, Venky; Reddy, Prakash
- [2] [HPL-2002-213 Authentication and Authorization of mobile clients in public data networks](#) - Reddy, Prakash; Krishnan, Venky; Zhang, Kan; Das, Devaraj
- [3] [OBEX Specification](#)
- [4] [Bluetooth Specification](#)
- [5] [CoolTown](#)