E*phone Functional Specification and Feature Roadmap

Henning Schulzrinne

July 13, 2000

1 Introduction

This document describes the feature set for versions 2.1 and later of the e*phone. The feature list is tentative and subject to availability of programming and circuit design resources.

2 Hardware

The current 2.1 hardware consists of separate DSP, memory, Ethernet interface, keyboard encoder, AD/DA and glue logic. This affords flexibility, but also increases cost and size. The following future versions are suggested.

The SIP hardware versions are listed below:

1.0	original "blue phone"	1
2.0	original Sprint phone	12
2.1	different voltage regulator	(*)
2.2	different PCB for modified components	100?
3.0	new chip set, custom case	millions

(*): Version 2.1 was modified in-place on existing 2.0 devices.

2.1 E*phone Hardware Version 3.0

The next hardware revision should utilize a single integrated chip for the network, keyboard encoder and DSP components, possibly also the AD/DA, LCD driver and parts of memory. Integrating the AD/DA may make it more difficult to provide high-quality audio input/output. Mitel, Netergy and Lucent may offer VoIP chips now or in the near future. (The Mitel chip is apparently sampling.) Another possibility is the Crystal Semiconductor EP 7212 or CL-PS 7500E chip set. (Apparently, there will also be a successor to the EP 7212, the EP 7213.)

Company	chip	RAM	DSP	A/D	Ethernet	keyb.	LCD	PCMCIA	USB
Mitel	MT92101	11 kB	yes	MT92303	2 MII	8x8	no	no	yes
Netergy	T2	256 kB			2 MII	GPIO	no	no?	no
Crystal	EP 7212				no		yes	CL-PS6700	no
Crystal	CL-PS 7500E								
TI	C5472	144 kB	yes		2 MII	GPIO	no	no	no

Crystal EP 7212: Low power; DRAM controller; use CL-PS6700 for PCMCIA; 74 MHz ARM clock.

Crystal CL-PS 7500E: Set-top box with CRT interface.

Crystal EP9312: Built-in Ethernet.

Mitel: 25 MHz ARM and 60 MIPS DSP; includes codecs and echo suppression; 8x8 keypad; no cache? $21

Netergy: Only GPIO for keyboard, so only small keypad possible; SRAM interface; has a flash cache allowing direct use of flash memory; 150 DSP MIPS.

TI/Telogy: 50 MHz ARM, 100 MHz DSP; I^2C interface

All devices use the ARM CPU core.
The following hardware features should be integrated:

- USB interface for video camera and other peripherals such as keyboards;
- PCMCIA or similar interface for IEEE 802.11 wireless LAN interface for home or business wireless networks;
- i-button or other identification interface, such as fingerprint recognition, to personalize phone;
- large, graphical LCD for web interface (Avantgo?).

This hardware could then be integrated in wireless residential or business handsets (similar to the Siemens Gigaset), wireless phone sets (non-portable), Ethernet "business" phone sets (possibly with special versions for vertical markets such as hotel, dormitory or ATMs) and as boards in external devices such as kiosks and ATMs.

As an alternative, the PCB could be integrated into a walkman-style enclosure, worn on a belt or in a pocket, with a microphone/ear-bud combination (e.g., Jabra). One could even combine an Internet (MP3) radio into the same device with minimal cost. Depending on how much flash memory the device contains, the device can then automatically "recharge" itself when within radio range and serve as a standard MP3 player when not.

With about 30x30 mm PCB real-estate, it is possible to integrate a GPS receiver for location-based services. (The module from u-blox costs about $151. However, GPS only works outside.

For the wireless versions, a first version would simply use the Apple AirPort or Lucent access point, connected via Ethernet to a DSL or cable modem. Later, it would be useful to cooperate with a DSL or cable modem vendor to integrate SIP proxy server functionality, 802.11 capability as well as a NAT ALG to deal with single-address residential networks into DSL and cable modems.

Design of this hardware should, in my opinion, proceed as quickly as possible. In particular, the chip set needs to be selected based on anticipated feature sets, availability and costs. Also, case design alternatives need to investigated.

3 E*phone Software

3.1 Preliminary Schedule

2.1 July 25, 2000
2.2 September 1, 2000
2.3 November 1, 2000

For each release, a user manual needs to be written and/or updated.

In the tables below, the date indicates when the feature was completed and tested. Features not yet finished are marked with the person responsible:

ka Knarig Arabshian
lw Li Wang
gn Gautam Nair
pn Pallavi Nayak

3.2 Network Module

IPv4 support	2.1	07/13/00
ARP, with cache	2.1	07/13/00
ICMP router advertisements	2.1	07/13/00
ICMP echo reply (for ping)	2.1	07/13/00
ICMP echo request (outgoing ping)	2.1	07/13/00
ICMP port unreachable	2.1	07/13/00
IGMP (multicast) support	2.1	07/13/00
DHCP for IP address	2.1	07/13/00
DHCP for net mask	2.1	07/13/00
DHCP for tftp server	2.1	07/13/00
DHCP for DNS host name	2.1	07/13/00
DHCP for DNS resolver	2.1	07/13/00
DHCP for default gateway	2.1	07/13/00
DHCP for default SIP proxy	2.1	07/13/00
DHCP for time server	2.1	07/13/00
DHCP for user name	2.1	gn
Time protocol support for clock	2.1	07/13/00
`syslog` logging	2.1	07/13/00
`tftp` upgrade	2.1	07/13/00
Handle fragmented packets	2.1	07/13/00
UDP	2.1	07/13/00
DNS A resource records	2.1	07/13/00
DNS CNAME resource records	2.1	07/13/00
DNS SRV resource records	2.1	07/13/00
IPv4 router alert option	2.2	
TOS byte setting (for diffserv)	2.2	

Network functionality can be tested with a DHCP, DNS and tftp server and via `ping`.

3.3 Audio/RTP Module

The audio module handles incoming audio packets, packetizes and encodes audio, generates RTP and parses RTP. Other audio codecs, such as G.726, G.729 or G.723.1, can be added depending on licensing restrictions and system application.

Call progress tones (busy, fast busy, ring back)	2.1	07/13/00
Audio loop back mode for hardware testing	2.1	07/13/00
G.711 (μ-law) codec	2.1	07/13/00
RTP (timestamp, seq. no, PT)	2.1	07/13/00
Robust against timestamp reset	2.1	pn/lw
Audio playout delay tested with trace	2.1	pn/lw
RTCP SR/RR sending	2.1	pn/lw
Speaker phone mode	2.2	
Audio mixing	2.2	
DTMF generation in RTP (RFC 2833)	2.2	
GSM audio codec	2.2	
DVI audio codec	2.2	
MPEG 2 L3 audio decoder	2.2	
Outgoing DTMF tone generation	2.2	
RTCP receiver support for diagnostics	2.2	
Mid-call codec changes	2.2	
Bandwidth-adaptive audio codec	2.3	
Loss compensation (FEC, RFC 2733)	2.3	
YESSIR resource reservation	2.3	
Play back RTP DTMF tones	2.3	
Acoustic echo cancellation for speaker-phone mode	2.x	

Speaker-phone mode is automatically enabled when one of the speed-dial or line buttons is pressed if the handset is in the cradle.

RTP functionality can be tested with the `rtpsend` and `rtpplay` tools, using recorded packet traces.

3.4 SIP/RTSP/SAP Module

The SIP module provides call state. In version 2.1, only a single logical line is supported, while later versions support multiple (up to four) lines.

The Session Announcement Protocol (SAP) [1] allows to advertise "radio" multicast sessions. Radio sessions are automatically muted when a call arrives.

RTSP support enables voice mail retrieval and audio-on-demand, assuming suitable codecs.

INVITE, ACK, BYE support	2.1	07/13/00
REGISTER support, with refresh	2.1	gn
SIP From, To, Call-Id	2.1	07/13/00
SIP CSeq	2.1	07/13/00
SDP, without dynamic payload types	2.1	07/13/00
Generation and processing of Via headers	2.1	07/13/00
Generation of SIP To tags	2.1	07/13/00
Change media types in mid-session	2.2	
REGISTER several identities	2.2	
Accept 30x redirect	2.2	
CANCEL generation	2.2	
CANCEL handling	2.2	
SDP, with dynamic payload types	2.2	
SAP for multicast sessions	2.2	
Call forwarding on no answer	2.2	
Call forwarding busy	2.2	
Call forwarding do-not-disturb with Retry-After	2.3	
Intercom mode (auto-answer)	2.3	
Basic authentication for incoming calls (password)	2.2	
Digest authentication for incoming calls	2.2	
Digest authentication for REGISTER	2.2	
Basic authentication for outgoing calls	2.3	
Digest authentication for outgoing calls	2.3	
Distinctive ringing based on caller id	2.3	
Distinctive ringing based on Priority	2.3	
Forwarding when noone present (sensor)	2.3	
Multiple logical "lines" ("call presences")	2.3	
Call transfer	2.3	
Conferencing between lines	2.3	
Allow camp-on and queueing	2.3	
Log-in via i-Button or keypad	2.4	
Interface to passive IR sensor	2.4	

For multiple lines, all participants are shown in the four LCD display lines:

```
 L1 John Doe
>L2 Jane Doe
 L3 Alice
 L4 Bob
```

The REGISTER request is first sent to the SIP multicast address. If there is no response or if the domain in the user's SIP URL differs from the local domain, the request is also sent to the SIP server derived from the DNS SRV record of the user's SIP domain. For example, if a phone belonging to the user `bob@example.com` is moved to `cs.columbia.edu`, it would register at the server belonging to `example.com` (in addition to the local server in the `cs.columbia.edu` domain).

SIP functionality can be tested with `siptc`.

3.5 User Interface Module

Configuration information is used only if DHCP is not available or disabled.

Display time of day	2.1	ka
Display call duration	2.1	ka
Display name/SIP URL of calling party	2.1	ka
Display name/SIP URL of called party	2.1	ka
Display subject of call	2.1	ka
Display priority of call	2.1	ka
Display call progress (1xx messages)	2.1	ka
Configuration: SIP name	2.1	ka
Configuration: net mask	2.1	ka
Configuration: own IP address	2.1	ka
Configuration: gateway address	2.1	ka
Configuration: DNS resolver address	2.1	ka
Configuration: domain	2.1	ka
Configuration: host name	2.1	ka
Configuration: logging server	2.1	ka
Configuration: time server	2.1	ka
Configuration: SIP outbound proxy	2.1	ka
Configuration: voice mail address (b/na/dnd)	2.2	
Configuration: no-answer delay	2.2	
Distinctive ringing by priority	2.2	
Address book (downloadable)	2.2	
Speed-dial buttons (downloadable)	2.2	
Configuration: busy behavior (queue, reject)	2.3	
Display packet statistics	2.3	
Call log (time, subject, caller)	2.3	
Radio interface	2.3	
RTSP-based distinctive ringing	2.4	
RTSP voicemail interface (play, ff, rw, etc.)	2.4	

Each phone can have several identities, e.g., both a user name and a telephone number. By default, identities are prepended to the local domain name.

The configuration menu has the following structure:

```
1 Network
  1 Enable (Disable) auto-configuration
  2 IP address
  3 Network mask
  4 Gateway
  5 DNS server
  6 Domain name
  7 Log server
  8 Time server
```

```
  9 Outbound SIP server
2 Phone
  1 Identities
    Id 1 - name
    Id 1 - address
    Id 2 - name
    Id 2 - address
    Id 3 - name
    Id 3 - address
    Id 4 - name
    Id 4 - address
  2 Voicemail
    Voicemail busy
    VM no answer
    VM do-not-disturb
    Wait for answer
3 Software
  1 Update server
```

Items 1/2-10 are only enabled if auto-configuration is *dis*abled. Otherwise, item '2' is replaced by the ability to list the current configuration.

Software is updated after changing or confirming the tftp server under the "Software" option. While this is sometimes configured for a local server, it should be possible to override the DHCP choice to allow upgrading from a remote server.

Under some circumstances, it is desirable to be able to configure the user name centrally, e.g., when installing a large number of phones at the same time. This also provides some modest measure of protection against people spoofing their identity. An administrator simply creates a list of hard-coded e*phone identifiers, such as their MAC address or serial number, printed on the case when phones are assigned to their owners. The user name is configured based on the client identifier (DHCP option 61). There currently does not appear to be a generic DHCP option for this, so this is probably best encoded in a vendor-specific option (43). The e*phone should identify itself as such via the vendor class identifier option (60) in the DHCP request.

There should be a button that goes up one layer in the configuration menu.

The speed dial buttons are configured either by adding a person called or calling to a speed dial button or by obtaining the information in the REGISTER response. Modified speed-dial buttons are uploaded via the REGISTER request. Speed dial buttons are configured by a simple text file containing Contact information, such as

```
1 John Doe <sip:j.doe@example.com>
2 Alice <sip:alice@wonderland.com>
```

If the e*phone is idle:

```
ephone 2.1
Th, Jun 9  6:33
7042
```

The following display appears when the handset is picked up or the speaker phone button is pressed:

```
Dial and hit Go
>_________________
 Jane
 Alice
 Carol
 Bob
```

The cursor can scroll through the call list, sorted either alphabetically or in order of last call.

If the phone is busy, the name of the caller is displayed and added to the call log.

```
6:34   1.12
John Doe
 j.doe@columbia.edu
To Alice Smith
```

The To information is displayed only if it differs from the user registered for the phone. Whenever name-addr information is displayed, name and SIP URL are split across two lines if they do not fit into a single display line. (The average email address is about 20 charactes long, while the average first and last names add up to about 14 characters.)

Distinctive ringing can be implemented by having the inbound proxy include a special ringing header, configured, for example, based on priority or caller identity. The ringing header includes an RTSP URL, which points to the audio source for ring tone. This avoids having to store ring music on the end system. Alternatively, the configuration can be done at the end system, but that's likely going to be more cumbersome and does not move from phone to phone.

3.6 WAP Functionality

It may be possible to integrate WAP access into a small-screen device (version 2.1). To limit ephone complexity, it is advisable to have a server translate WAP WDML into SIP INFO messages. Similarly, key clicks in "browser" mode are directly forwarded to the WAP gateway server.

One problem that remains to be addressed is the ability to "dial" a URL directly from the device. SIP third-party mechanisms may make this possible.

References

[1] M. Handley, C. Perkins, and E. Whelan, "Session announcement protocol," Internet Draft, Internet Engineering Task Force, Mar. 2000. Work in progress.