The SuperTel IP Phone
Hardware Interface Description

1 Introduction
This document describes the hardware interface for SuperTel’s new IP phone. A layered perspective on the software for the IP phone is given below in Figure 1. In particular, the control program for this phone resides in flash memory and uses services provided by a real time operating system, or RTOS, that includes support for the internet protocol and a number of device drivers, DD1, DD2 and so on. This collection of services comprises the hardware interface and is specified in greater detail in the following section. Note that the device drivers control the various input and output devices that make up the phone hardware: a small display screen, a keypad, an off-hook switch, a speaker, a microphone and an Ethernet port.

![Figure 1: Software organization for the SuperTel IP phone handset.](image)

2 The Hardware Interface
The hardware interface consists of a set of possible requests that the onboard control program can issue and a set of special messages that can be received. In addition there is a general RCP mechanism based on the ultra simple object access protocol (USOAP). All messages in USOAP have the general form (where square brackets “[” and “]” enclose an optional parameter).

```
<MessageName>
   [<DestDevice>IP address</DestDevice>]
   <FirstParameterName>first parameter</FirstParameterName>
   ...
   <LastParameterName>last parameter</LastParameterName>
</MessageName>
```

If there are no parameters, a message can have the following abbreviated form.

```
<MessageName/>
```

All requests and special received messages defined below follow this protocol.

2.1 Requests
These messages are sent by the onboard control program to indirectly control the various local devices on the IP phone itself.
2.1.1 **HandsetOn**

\(<\text{HandsetOn}/>\)

The speaker and microphone on the handset needs to be explicitly turned on. Without the handset, you can’t hear tones being played or a connected audio path.

2.1.2 **HandsetOff**

\(<\text{HandsetOff}/>\)

Turns off handset speaker and microphone.

2.1.3 **LampOn**

\(<\text{LampOn}/>\)

Turn on lamp on top of phone.

2.1.4 **LampOff**

\(<\text{LampOff}/>\)

Turn off lamp on top of phone.

2.1.5 **StartRinging**

\(<\text{StartRinging}/>\)

Start phone ringing. Phone will ring on and off continuously. There is no need to tell the phone to ring every second as the phone does it for you.

2.1.6 **StopRinging**

\(<\text{StopRinging}/>\)

Stop the phone from ringing.

2.1.7 **PlayTone**

\(<\text{PlayTone}>\)

\(<\text{Tone}>\text{tone}</\text{Tone}>\)

\(<\text{/PlayTone}>\)

Play the given tone. Possible tone values are as follows.
Also, all tones are terminated by issuing a PlayTone with the tone parameter STOP. There are some
subtleties to playing tones on the phones. They do not like to be told to keep playing different tones
without first stopping the tone currently being played. If you have not explicitly stopped the current tone
by sending STOP, the current tone continues to play. Of course, the stop-playing-current-tone value
STOP cannot in turn be STOPped. If you want silence, you should explicitly turn off the tone being
played.

2.1.8 DisplayString

Display string lineStr on line lineNum starting at position linePos. The command places the display
cursor on the given line at the given position and displays the given string. The line number, lineNum,
may be a value between 0 and 2, and the line position, linePos, may be a value between 0 and 25. A
string that goes past the end of the current line will wrap to the next line.

To clear a line, simply send spaces as the string. Unfortunately there is no way to position the cursor
without also writing a character.

2.1.9 AppendString

Append string lineStr at the current cursor position.

2.1.10 StartAudioSend

Start sending audio from the current phone to the phone at the IP address phoneIP. To make a voice call,
both phones must start sending to each other and receiving from each other. The phone handset must be
turned on for this to be useful.
2.1.11 **StartAudioReceive**

```xml
<StartAudioReceive>
  <DestDevice>phoneIP</DestDevice>
</StartAudioReceive>
```

Allow receiving audio on the current phone from the phone at the IP address phoneIP. To make a voice call, both phones must start sending to each other and receiving from each other. The phone handset must be turned on for this to be useful.

2.1.12 **StopAudioSend**

```xml
<StopAudioSend>
  <DestDevice>phoneIP</DestDevice>
</StopAudioSend>
```

Stop sending audio from the current phone to the phone at the IP address phoneIP.

2.1.13 **StopAudioReceive**

```xml
<StopAudioReceive>
  <DestDevice>phoneIP</DestDevice>
</StopAudioReceive>
```

Stop sending receiving audio on the current phone from the phone at the IP address phoneIP.

2.2 **Special Received Messages**

The following messages are sent to the onboard control program.

2.2.1 **OnHook**

```xml
<OnHook/>
```

Phone was placed on-hook.

2.2.2 **OffHook**

```xml
<OffHook/>
```

Phone was placed off-hook.

2.2.3 **DigitPressed**

```xml
<DigitPressed>
  <Value>digit</Value>
</DigitPressed>
```
The button digit was pressed. Possible values of the digit parameter are as follows.

- DIGIT0, DIGIT1, DIGIT2, DIGIT3, DIGIT4, DIGIT5, DIGIT6, DIGIT7, DIGIT8, DIGIT9
- FUNC1, FUNC2

2.2.4 **DigitReleased**

```xml
<DigitReleased>
    <Value>digit</Value>
</DigitReleased>
```

The button digit was released. Possible values of the digit parameter are as above.