Driver documentation for yealink usb-p1k phones

0. Status

The p1k is a relatively cheap usb 1.1 phone with:

- keyboard	full support,	yealink.ko / input event API
- LCD	full support,	yealink.ko / sysfs API
- LED	full support,	yealink.ko / sysfs API
- dialtone	full support,	yealink.ko / sysfs API
- ringtone	full support,	yealink.ko / sysfs API
- audio playback	full support,	snd_usb_audio.ko / alsa API
- audio record	full support,	snd_usb_audio.ko / alsa API

For vendor documentation see http://www.yealink.com

1. Compilation (stand alone version)

Currently only kernel 2.6.x.y versions are supported.
In order to build the yealink.ko module do

```
make
```

If you encounter problems please check if in the MAKE_OPTS variable in
the Makefile is pointing to the location where your kernel sources
are located, default /usr/src/linux.

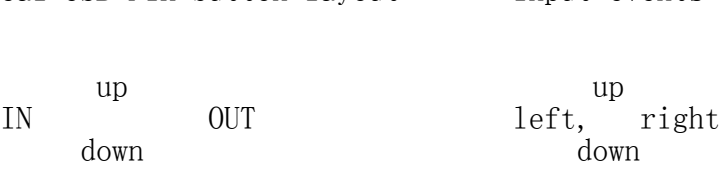
1.1 Troubleshooting

Q: Module yealink compiled and installed without any problem but phone
   is not initialized and does not react to any actions.
A: If you see something like:
   hiddev0: USB HID v1.00 Device [Yealink Network Technology Ltd. VOIP USB Phone
   in dmesg, it means that the hid driver has grabbed the device first. Try to
   load module yealink before any other usb hid driver. Please see the
   instructions provided by your distribution on module configuration.

Q: Phone is working now (displays version and accepts keypad input) but I can't
   find the sysfs files.
A: The sysfs files are located on the particular usb endpoint. On most
   distributions you can do: "find /sys/ -name get_icons" for a hint.

2. keyboard features

The current mapping in the kernel is provided by the map_p1k_to_key
function:



pickup	C	hangup	enter, backspace, escape
1	2	3	1, 2, 3
4	5	6	4, 5, 6,
7	8	9	7, 8, 9,
*	0	#	*, 0, #,

The "up" and "down" keys, are symbolised by arrows on the button.
The "pickup" and "hangup" keys are symbolised by a green and red phone
on the button.

3. LCD features

The LCD is divided and organised as a 3 line display:

```
    |[]   [][]   [][]   [][]   in   |[][]
    |[] M [][] D [][] : [][]   out  |[][]
                              store

    NEW REP          SU MO TU WE TH FR SA

    [] [] [] [] [] [] [] [] [] [] [] []
    [] [] [] [] [] [] [] [] [] [] [] []
```

```
Line 1  Format (see below)      : 18.e8.M8.88...188
        Icon names              :   M  D  :  IN OUT STORE
Line 2  Format                  : .........
        Icon name               : NEW REP SU MO TU WE TH FR SA
Line 3  Format                  : 888888888888
```

Format description:
From a userspace perspective the world is separated into "digits" and "icons".
A digit can have a character set, an icon can only be ON or OFF.

Format specifier
'8' : Generic 7 segment digit with individual addressable segments

Reduced capability 7 segm digit, when segments are hard wired together.
'1' : 2 segments digit only able to produce a 1.
'e' : Most significant day of the month digit,
able to produce at least 1 2 3.
'M' : Most significant minute digit,
able to produce at least 0 1 2 3 4 5.

Icons or pictograms:
'.' : For example like AM, PM, SU, a 'dot' .. or other single segment
elements.

4. Driver usage

For userland the following interfaces are available using the sysfs interface:
/sys/.../
line1 Read/Write, lcd line1

line2	Read/Write, lcd line2
line3	Read/Write, lcd line3
get_icons	Read, returns a set of available icons.
hide_icon	Write, hide the element by writing the icon name.
show_icon	Write, display the element by writing the icon name.
map_seg7	Read/Write, the 7 segments char set, common for all yealink phones. (see map_to_7segment.h)
ringtone	Write, upload binary representation of a ringtone, see yealink.c. status EXPERIMENTAL due to potential races between async. and sync usb calls.

4.1 lineX

Reading /sys/../lineX will return the format string with its current value:

```
Example:
cat ./line3
888888888888
Linux Rocks!
```

Writing to /sys/../lineX will set the corresponding LCD line.
- Excess characters are ignored.
- If less characters are written than allowed, the remaining digits are unchanged.
- The tab '\t'and '\n' char does not overwrite the original content.
- Writing a space to an icon will always hide its content.

```
Example:
date +"%m.%e.%k:%M" | sed 's/^0/ /' > ./line1
```

Will update the LCD with the current date & time.

4.2 get_icons

Reading will return all available icon names and its current settings:

```
cat ./get_icons
on M
on D
on :
   IN
   OUT
   STORE
   NEW
   REP
   SU
   MO
   TU
   WE
   TH
   FR
```

SA
LED
DIALTONE
RINGTONE

4.3 show/hide icons

Writing to these files will update the state of the icon.
Only one icon at a time can be updated.

If an icon is also on a ./lineX the corresponding value is
updated with the first letter of the icon.

Example - light up the store icon:
echo -n "STORE" > ./show_icon

cat ./line1
18.e8.M8.88...188
S

Example - sound the ringtone for 10 seconds:
echo -n RINGTONE > /sys/..../show_icon
sleep 10
echo -n RINGTONE > /sys/..../hide_icon

5. Sound features

Sound is supported by the ALSA driver: snd_usb_audio

One 16-bit channel with sample and playback rates of 8000 Hz is the practical
limit of the device.

Example - recording test:
arecord -v -d 10 -r 8000 -f S16_LE -t wav foobar.wav

Example - playback test:
aplay foobar.wav

6. Credits & Acknowledgments

- Olivier Vandorpe, for starting the usbb2k-api project doing much of
the reverse engineering.
- Martin Diehl, for pointing out how to handle USB memory allocation.
- Dmitry Torokhov, for the numerous code reviews and suggestions.