Accessorizing Android

Rajesh Sundaram (rajeshsweb at gmail dot com)

Introduction

Studying how android can be accessorized to tailor it for different professions. Attempting to find answers to a few relevant questions: Is it possible to use existing sensors? Can we integrate off-the-shelf accessories easily while keeping the complexity and cost low? What are the challenges in software and hardware?

In particular a.android.com has details about adding accessories.

Sensors & Interfaces

Sensors

Typical inbuilt sensors include

- **camera**: at least one on back but using this continuously can be a power drain
- **magnetic compass**: the usual but some use this to detect proximity of ferromagnetic materials (eg: lost house/car key) for fun
- **accelerometer**: detects acceleration due to gravity or motion in 3D
- **gyroscope**: available on newer devices to help smooth out compass & accelerometer reading
- **proximity**: on front to help turn the screen off and disregard touch input when held close to face (say on a phone call)
- **ambient light**: on front to help dim the screen
- **touch screen**: some support single finger tracking only
- **dedicated hardware keys**: volume & power on most, menu, back, home on few, qwerty keyboard on few, ...
- **microphone**: typically on the front; some have a secondary mic on back for noise cancellation

Some devices might sport air-pressure, relative-humidity and ambient temperature sensors as well. Recently learnt of a Japanese phone with radiation detection capabilities (Fukushima effect, I suppose).

Bi-directional Interfaces

Typical interfaces include

- **serial**: exposed through 3.5mm headphone jack (see Square for how somebody made big business out of it)
- **USB**: phones might have slave mode only (newer tablets and phones might support USB OTG, USB host

- **Bluetooth**: limited profiles are included in the stack. SPP/RFCOMM can however help add generic serial bluetooth accessories
- **Wi-Fi**: typically STA mode only and not many support ad hoc (IBSS) mode out of the box. Some support hotspot (virtual AP mode)
- **GSM/GPRS/3G/...**: not easy to find other accessories that directly talk over this interface

And …

Other than these bi-directional type, all mentioned sensors are indeed mono-directional input interfaces and then we have LED display and speakers for output. Some devices extend support for external display over HDMI.

Of course, interfaces can also act as sensors in some cases.

Thoughts

For quick start off-the shelf accessories need to be considered. Of all fields, it is easy to source accessories relevant to medical field. Obviously competition is higher there. Many of these have interfaces that can be soldered onto the mainboard of a smartphone or at least used externally say over serial (3.5mm jack), USB or Bluetooth.

Hurdles/Challenges

Software is a hurdle if we attempt to add our own protocols. Reverse engineering interface without datasheets would be a nightmare. And here are some hurdles or challenges with each interface.

With USB

Unfortunately since USB host/OTG capable android smartphones are rare and costly as of 2012, we may wish to consider the other options. Even the cheapest first generation android devices had serial out and Bluetooth but USB was slave only.

Even when available, not all USB hosts might have enough power to drive external devices.

With Bluetooth

Bluetooth capable accessories available off the shelf are costlier than the ones without by about US$10. Also some of the bluetooth profiles might not be supported on cheap android devices. Even if built on top of generic RFCOMM Bluetooth profile, software might not be trivial.

With Serial too!

Serial sounds easier than other options to integrate with cheap accessories … it

> shouldn’t be too tough to interface readily available digital or even quantize analog to suit the headphone jack.
>
> However the larger hurdle here is multiplexing or arbitrating when we attempt to use the single serial interface to talk to several accessories. TD MUX or a proper protocol might help but complicates the design drastically. Well this is exactly why we have USB; a single USB interface can talk to 128 devices.

Instead of trying to solve all these in one go, it might be reasonable to blindly procure a few cheap accessories to investigate through practice.