PARC

Yet Another Project Team
(YAPT)

Jacques Dafflon, Samantha Rosso

The Problem

While sitting at a computer, the phone is often left idle and unused.
This is a shame considering the potential that a phone has in improving user interaction with other devices.

The Solution

PARC - Phone As a Remote Control
Different motions can trigger different commands that enable the user to interact with a computer through simple gestures.

Gestures & Actions

Swipe up

Swipe down

Swipe right

Swipe left

Long Press

Hold over

Swipe over

Play / Pause

Media track: next

Media track: previous

Volume: up

Volume: down

Volume: mute

Brightness: up

Brightness: down

Keyboard backlight: up

Keyboard backlight: down

Keyboard backlight: toggle

Lock Computer

Keynote: next slide

Keynote: previous slide

Keynote: start presenting

Sorry Tim, no photos backup (yet)

How PARC works

1. Enter your computer's IP address in the app.
   (Optionally configure gestures)

2. Start the (Python) server on your computer

3. Perform a simple gesture
   (eg. swipe up)

4. Gesture is converted to an action code

5. Action code is sent over UDP to the computer

6. Computer execute the action
   (eg. volume up for swipe up—by default)

Sensors & Actuators

Proximity

Detects whether an object is near or far from the phone.

Touch Screen

Add a variety of gestures that the user can set to specific commands.

WiFi

Allows the phone to send commands to the computer.

Current limitation

UDP

Low overhead, perfectly suited for the needs of our application.

No discovery, the user has to manually enter the computer's IP address. (Try telling that to my grand-father.)
Can be added on top of UDP, but requires some effort.

No security, anyone can send commands to your computer!
Should be added, using DTLS.

Bonus: `echo -n "<command>" | nc -4u -w1 <ip_addr> <port>`
(~= Android application in 1 line of bash)

Challenges

THANK YOU!

Bluetooth

Did not work as expected with OSX.
Unable to pair the phone with the computer properly.
Bluetooth socket unavailable on OSX using Python.

Proximity Sensor

Sampling rate too low for complex, rapid actions.
Limited range forces the hand to be really close to the screen.

Light Sensor

Better range but the sampling rate is just as low.
Varying performance in different light conditions