Wearable Computing

Gabriel Reyes
CS-HCI PhD Student
School of Interactive Computing

CS 4470 / CS 6456
W. Keith Edwards

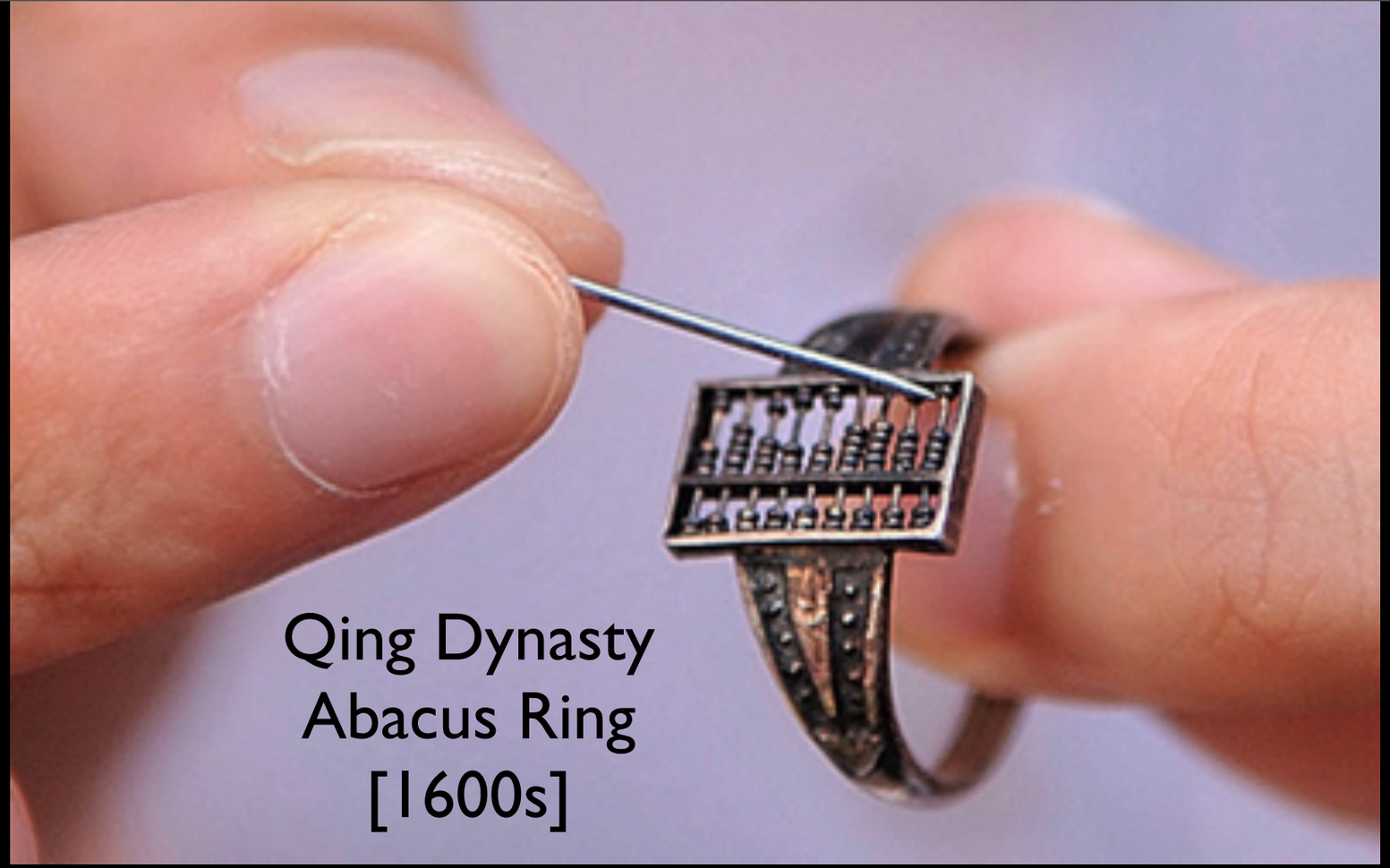
Quick Experiment

How many forms of computing are you "wearing" today?

(Don't count what's in your purse or backpack)

What does J.A.R.V.I.S. stand for?

"The Nuremberg Egg" [1510]



Qing Dynasty
Abacus Ring
[1600s]

Breguet Watch
[1810]

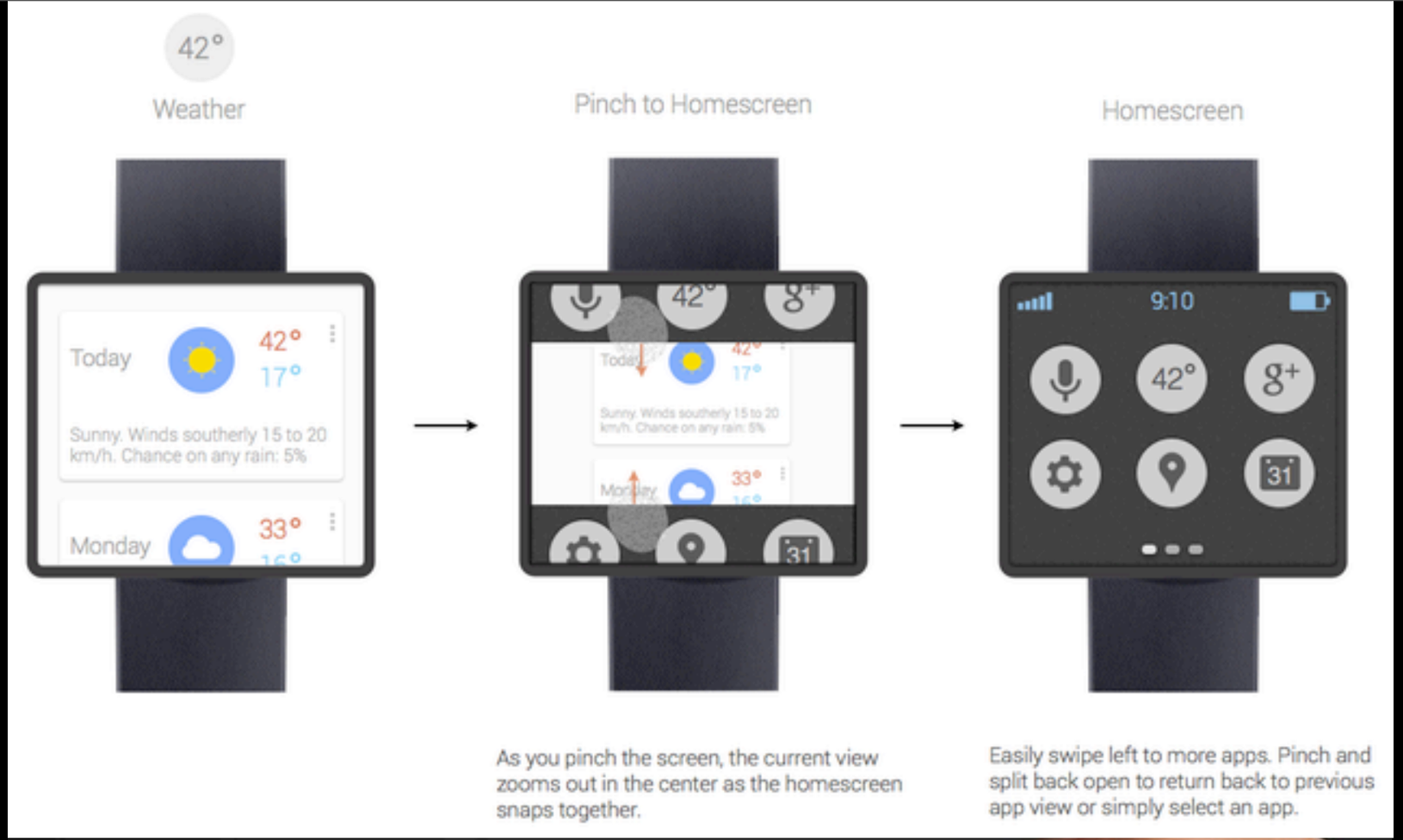
Timing Device
for Gambling
[1960s]

4,718

FUEL

9:12 am
NEW EMAIL
Sarah M.
"Remember to pick up some milk!"

JAWBONE

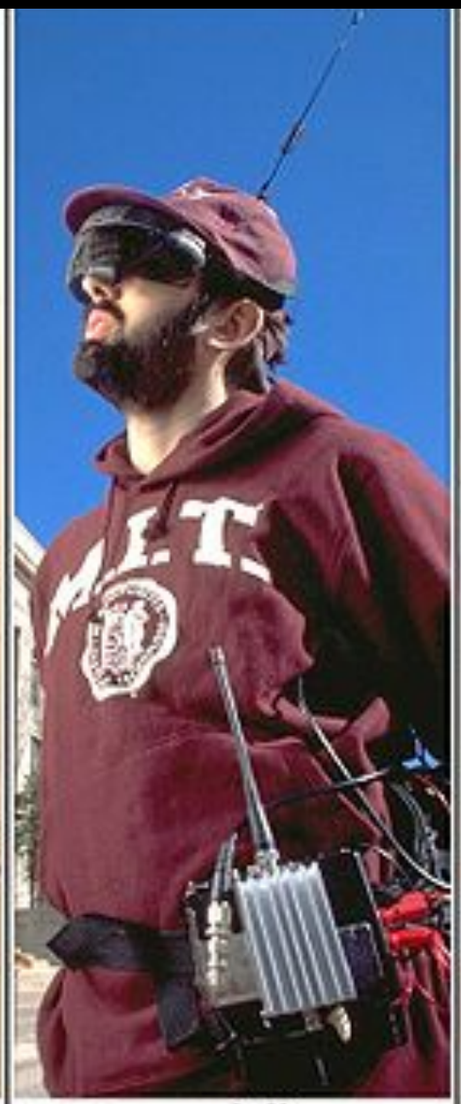

42°
Weather
Today
42°
17°
Sunny. Winds southerly 15 to 20 km/h. Chance on any rain: 5%
Monday
33°
Pinch to Homescreen
Homescreen
9:10
As you pinch the screen, the current view zooms out in the center as the homescreen snaps together.
Easily swipe left to more apps. Pinch and split back open to return back to previous app view or simply select an app.

8:20
Monday, September 21
slide to unlock

(a)
1980

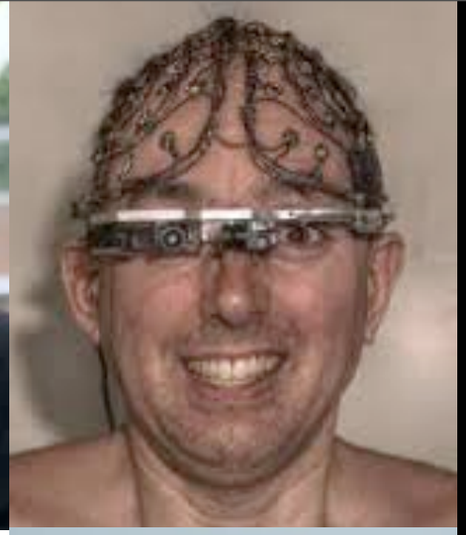
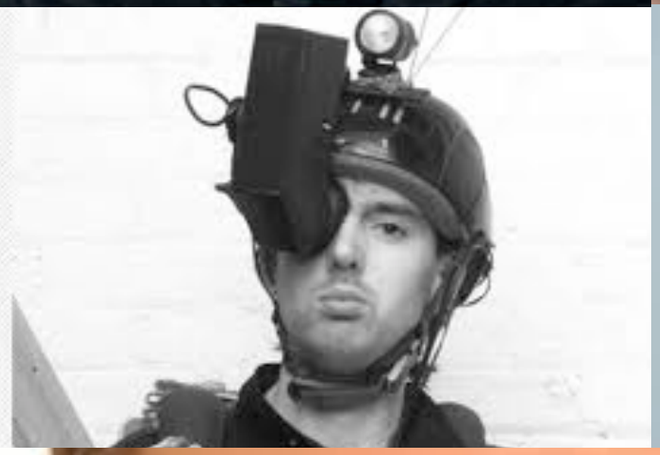
(b)
Mid 1980s

(c)
Early 1990s



(d)
Mid 1990s

(e)
Late 1990s

Steve Mann
MIT Media Lab
University of Toronto

Thad Starner
MIT Media Lab
Georgia Tech, Google [x]

"Wearable computing pursues an interface ideal of a continuously worn, intelligent assistant that augments memory, intellect, creativity, communication, and physical senses and abilities"

The Challenges of Wearable Computing: Part I -- by Thad Starner

Why use wearable computers?

Mediate interactions with environment and interfaces around you

Assisting human-to-human communication

Provide context-sensitive reminders

Augment reality

Enable new forms of communication

Ideal Attributes

Persist and provide constant and fast access to information services

Sense and model context

Adapt interaction modalities based on the user's context

Augment and mediate interactions with the user's environment

Challenges of Wearable Computing

Power requirements and heat dissipation

Body area networks and connectivity

Privacy and social issues

Interfaces

"Always design your system around the battery"

Why now?

"A perfect storm of innovation within low power wireless connectivity, sensor technology, big data, cloud services, voice user interfaces and mobile computing power is coming together"

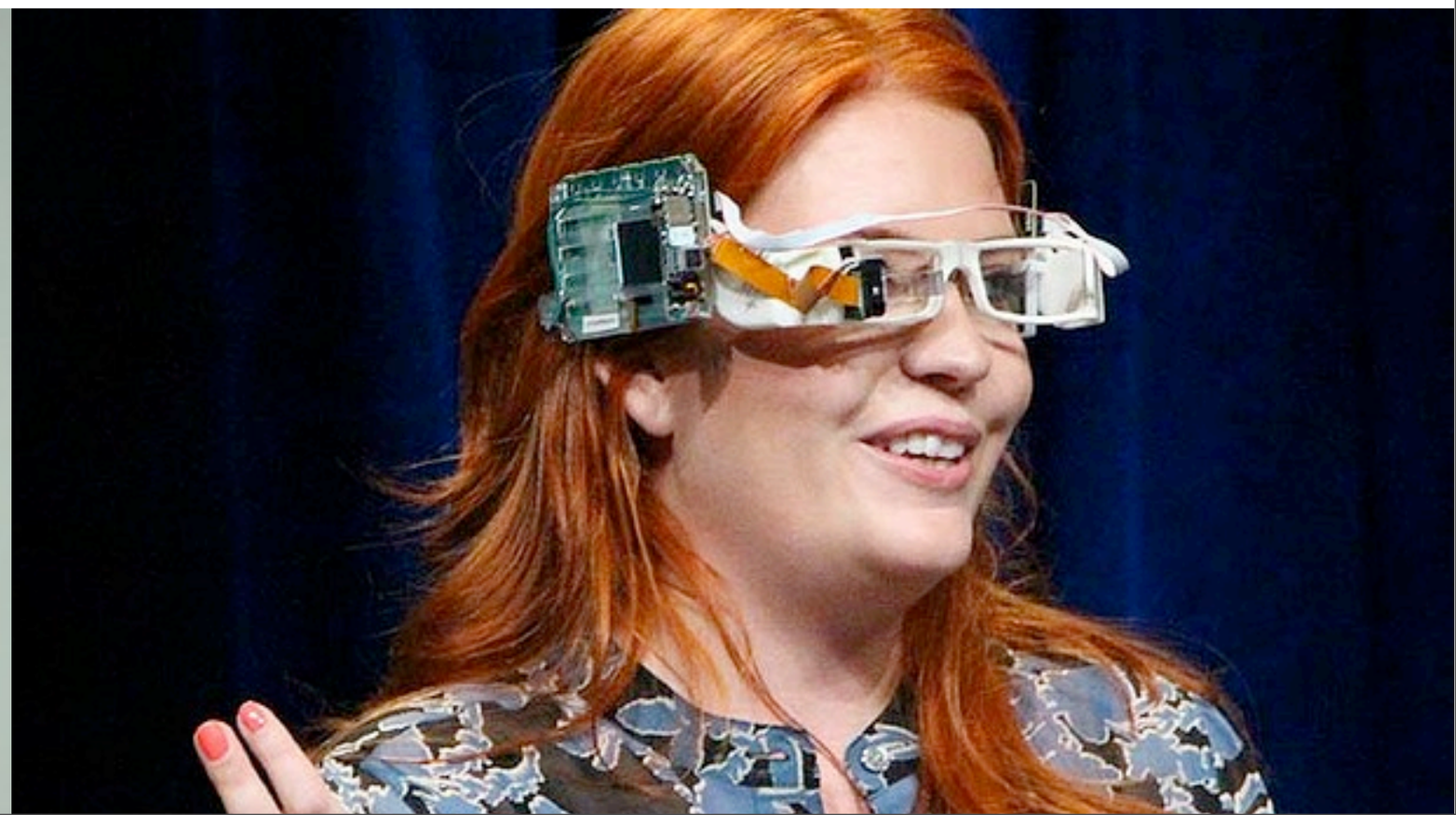
http://bgr.com/2013/10/25/wearable-computer-shipment-projection/

Growth Market

Sales of smart glasses, smart watches and wearable fitness trackers reached 8.3 million units worldwide in 2012

Total shipments of wearable technology devices are expected to reach 64.0 million units in 2017

Prototyping

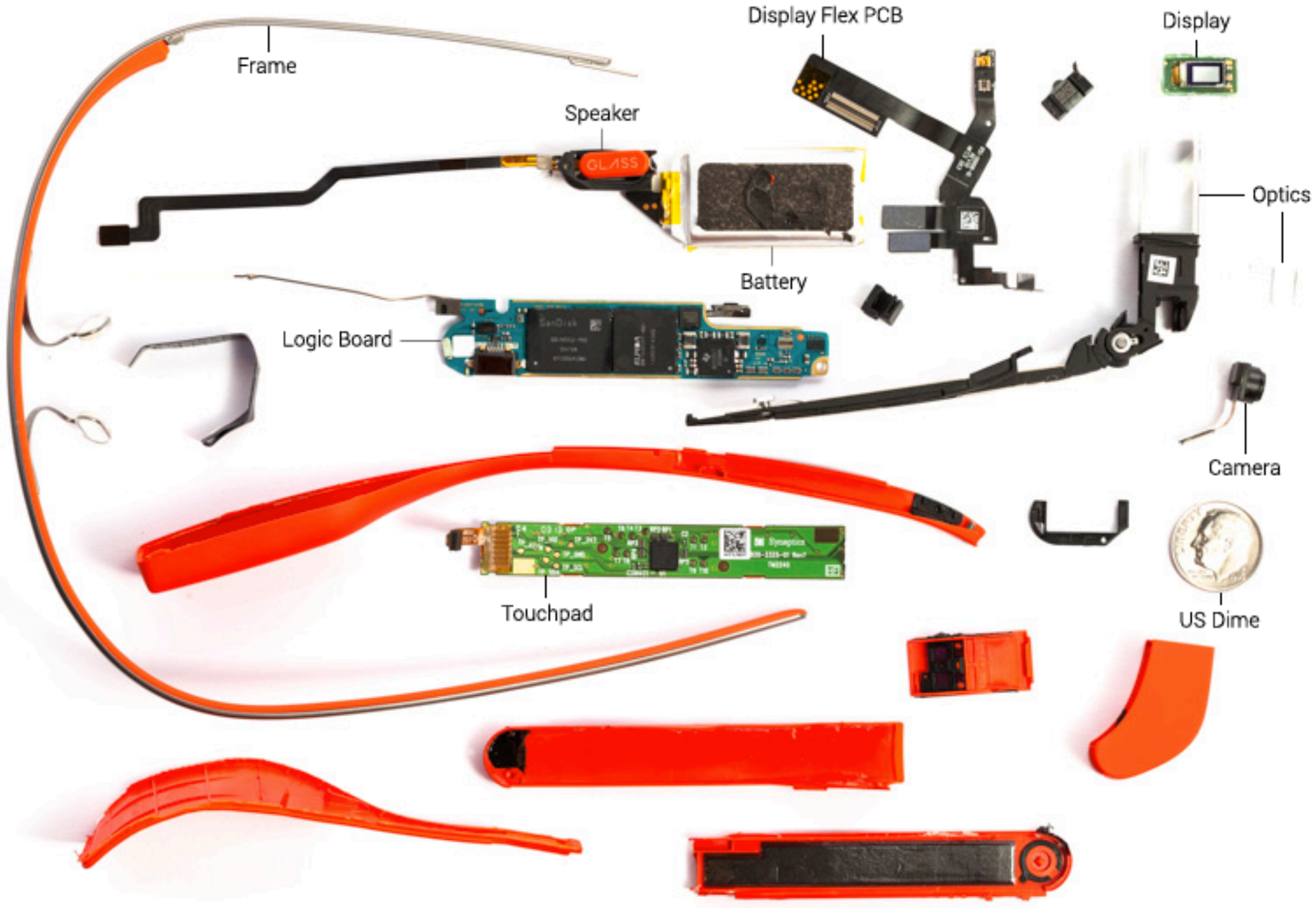


http://www.youtube.com/watch?v=v1uyQZNg2vE

Google Glass

Release Date:	Developers: Feb 2013 Consumers: 2014?
Introductory Price:	$1500 Explorer $300-$500 Consumer University Programs
Operating System:	Android 4.0.4
Manufacturer:	Foxconn

What's Inside Google Glass?



Hardware

Power:	Lithium Polymer (2.1Wh, 570mAh)
CPU:	OMAP 4430 SoC 1.2GHz Dual ARM7
Storage:	16GB Flash Memory (12GB Usable)
Memory:	1GB RAM

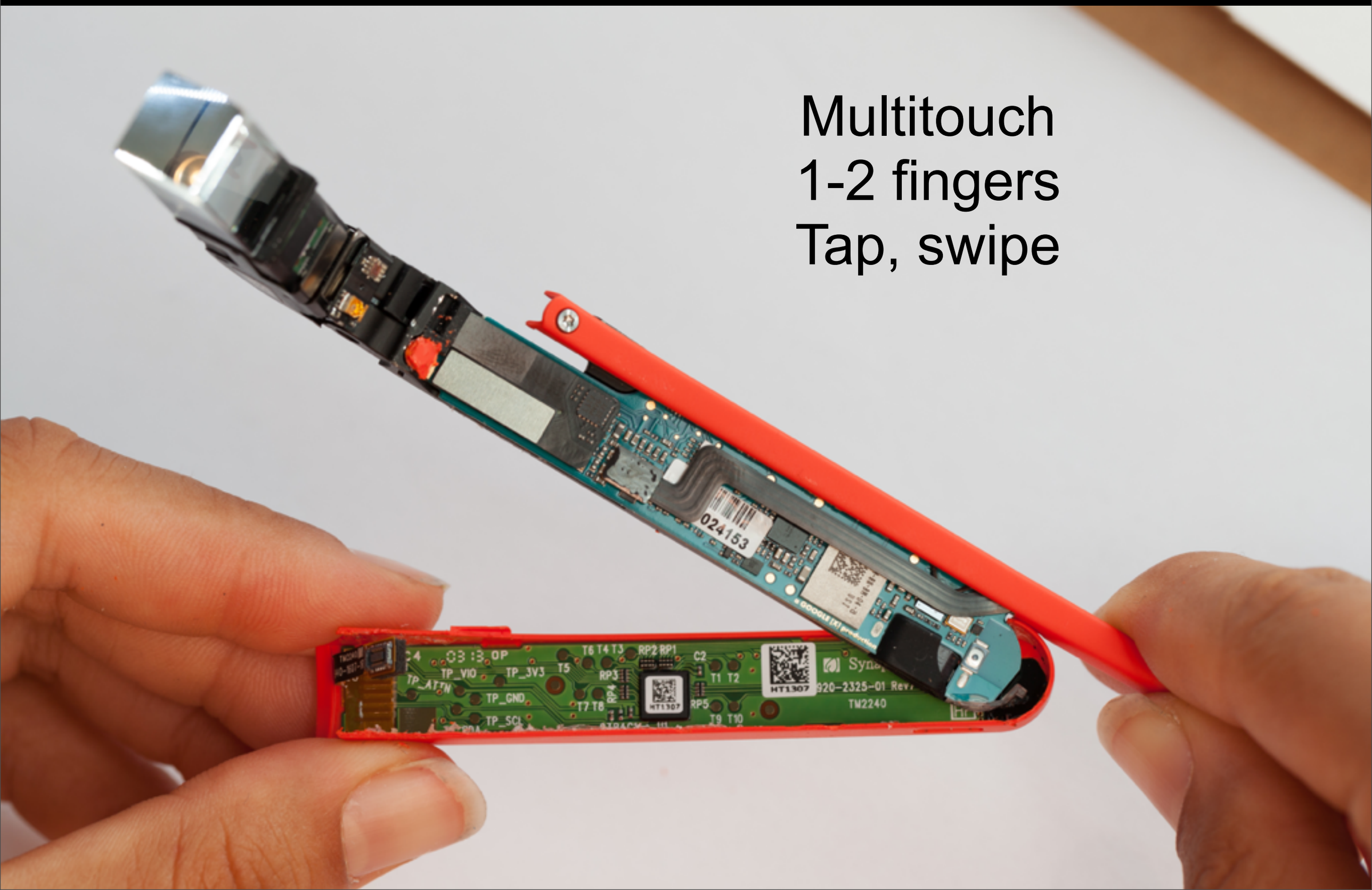
Main CPU Board

TI OMAP4430 CPU
16GB of SanDisk flash storage
Elpida mobile DRAM memory

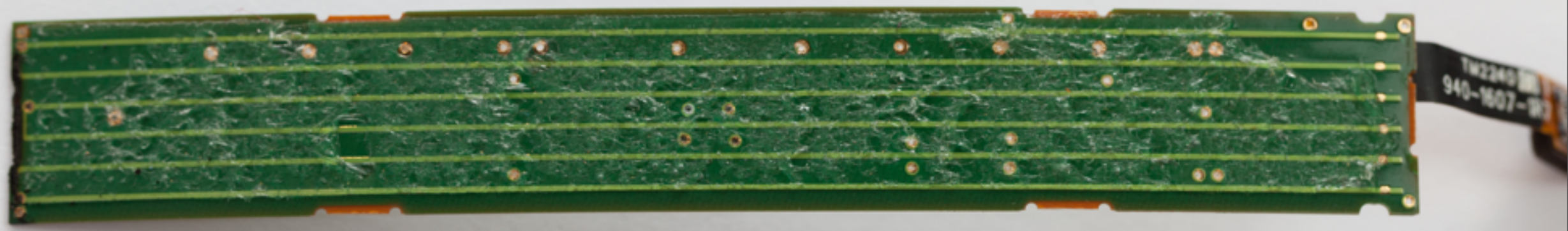
SiRFstarIV GSD4e GPS engine
Bluetooth/WiFi module
Flex PCB and antenna

Synaptics Touchpad

Multitouch
1-2 fingers
Tap, swipe



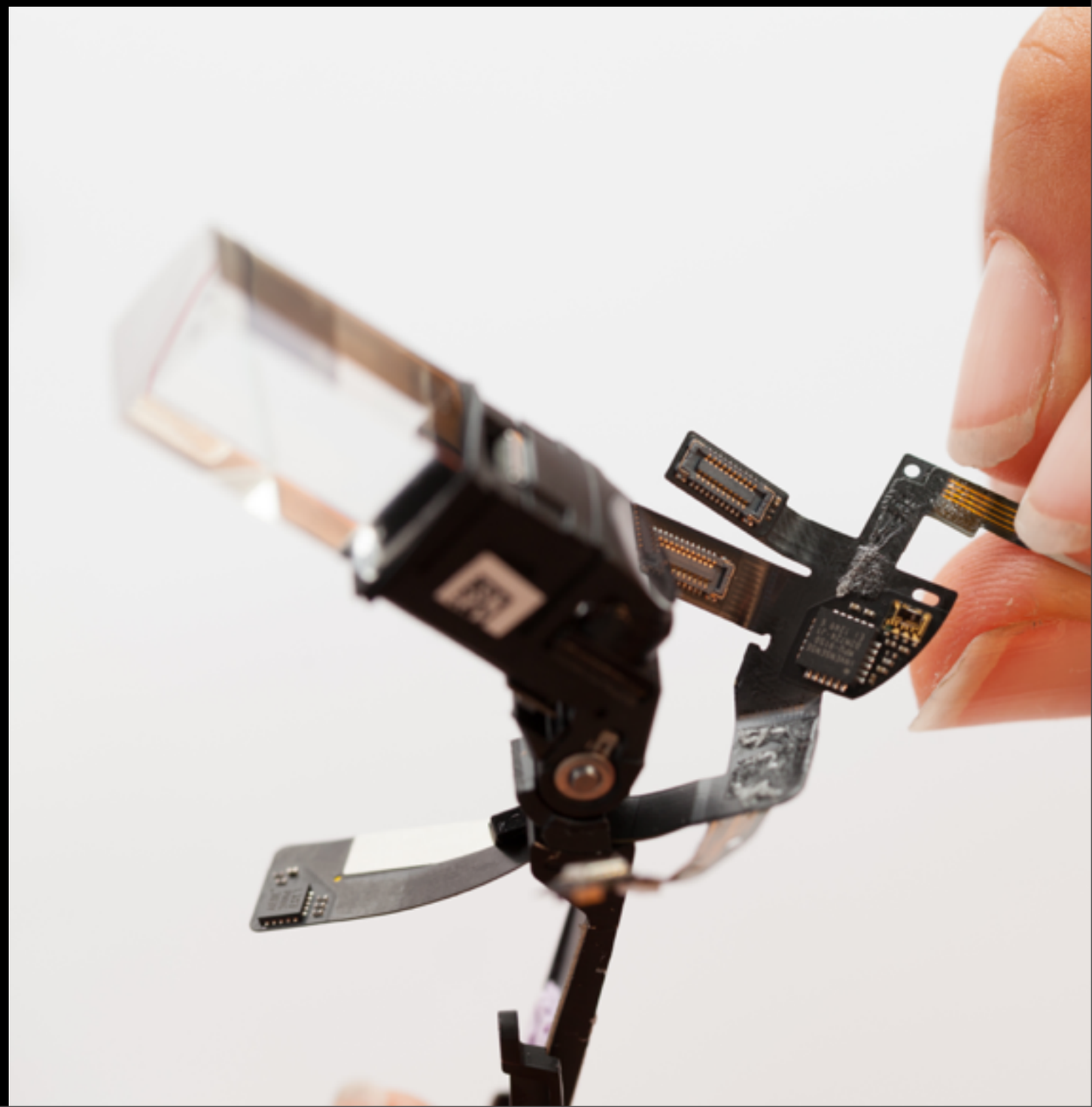
The touchpad is a full custom module made by Synaptics



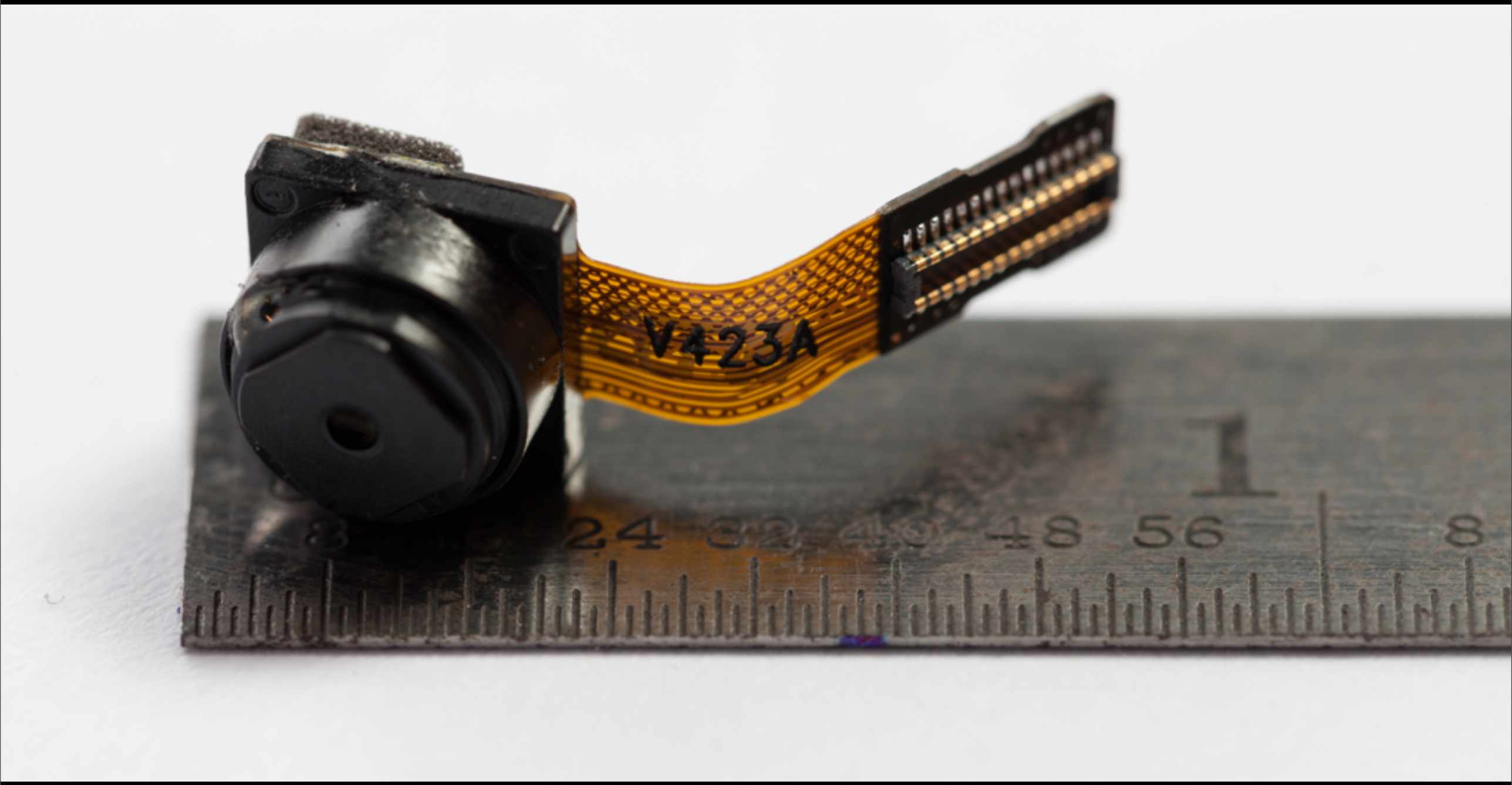
Capacitive sensing driven by a Synaptics T1320A touchpad controller

On-Board Sensors

Microphone
Accelerometer
Gyroscope
Magnetometer
Ambient light sensor
Proximity sensor
GPS



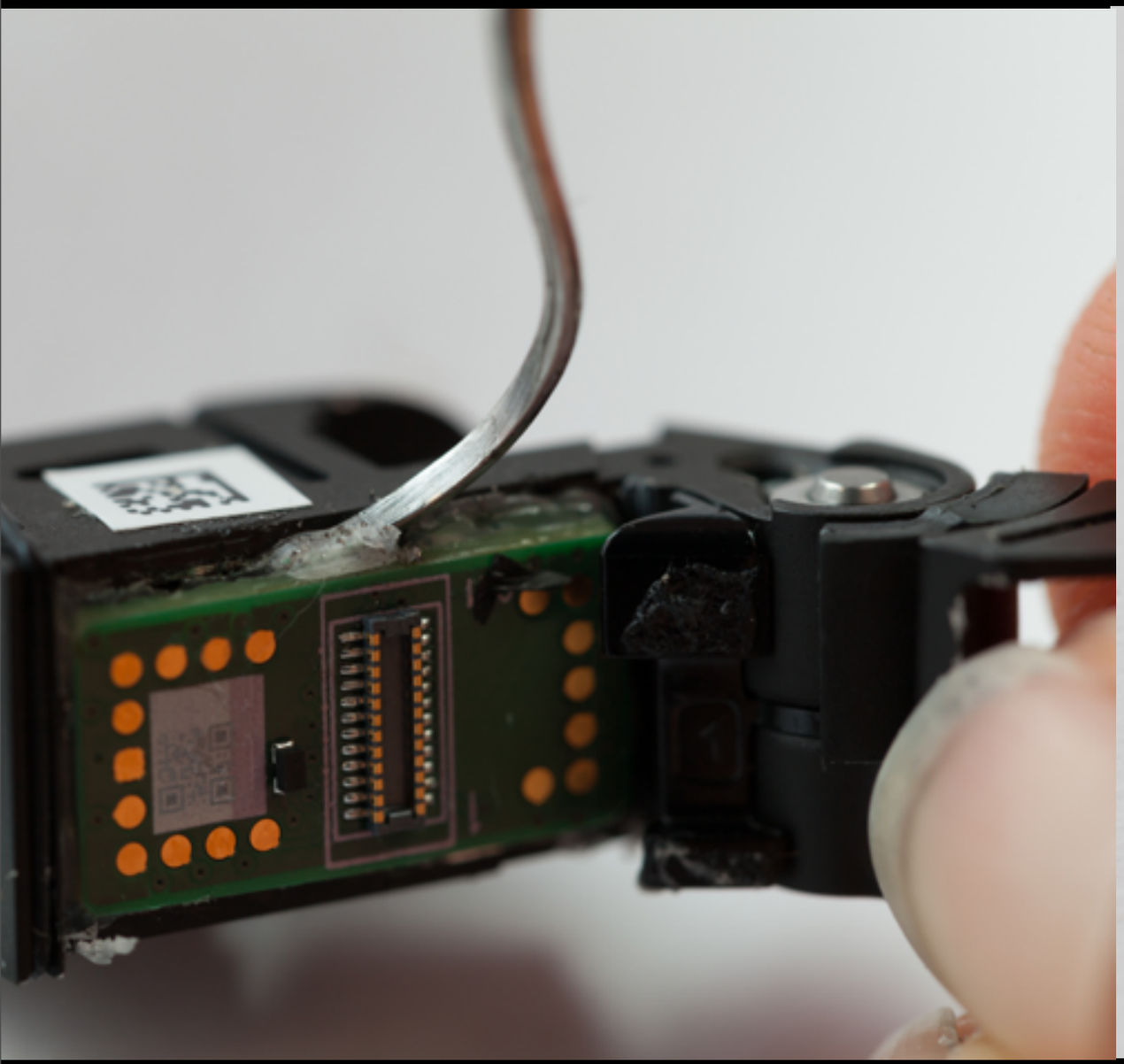
Camera

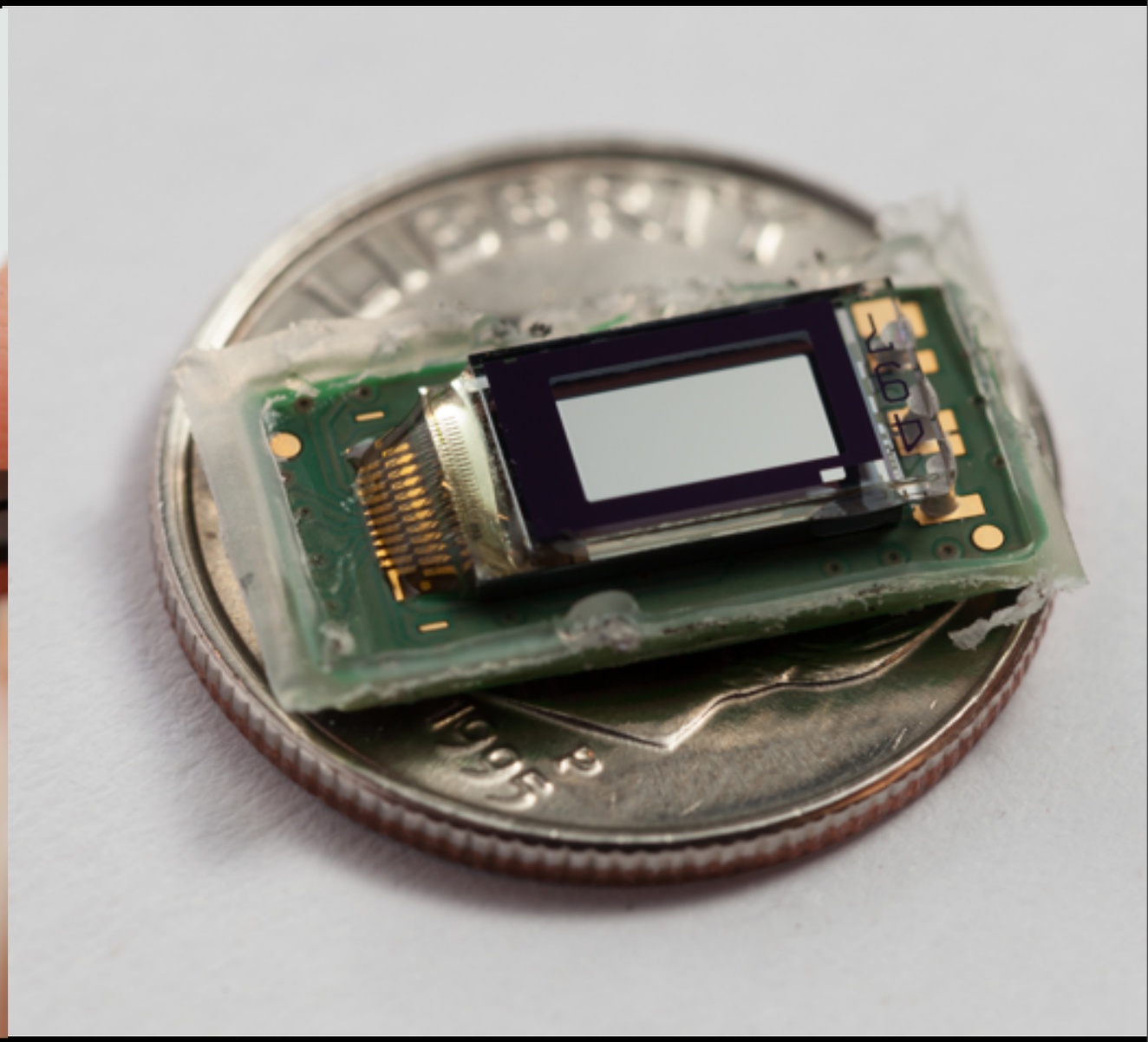


5 megapixel camera
(2528 x 1856 pixels)

720p video recording
(1280 x 720 pixels)

Display



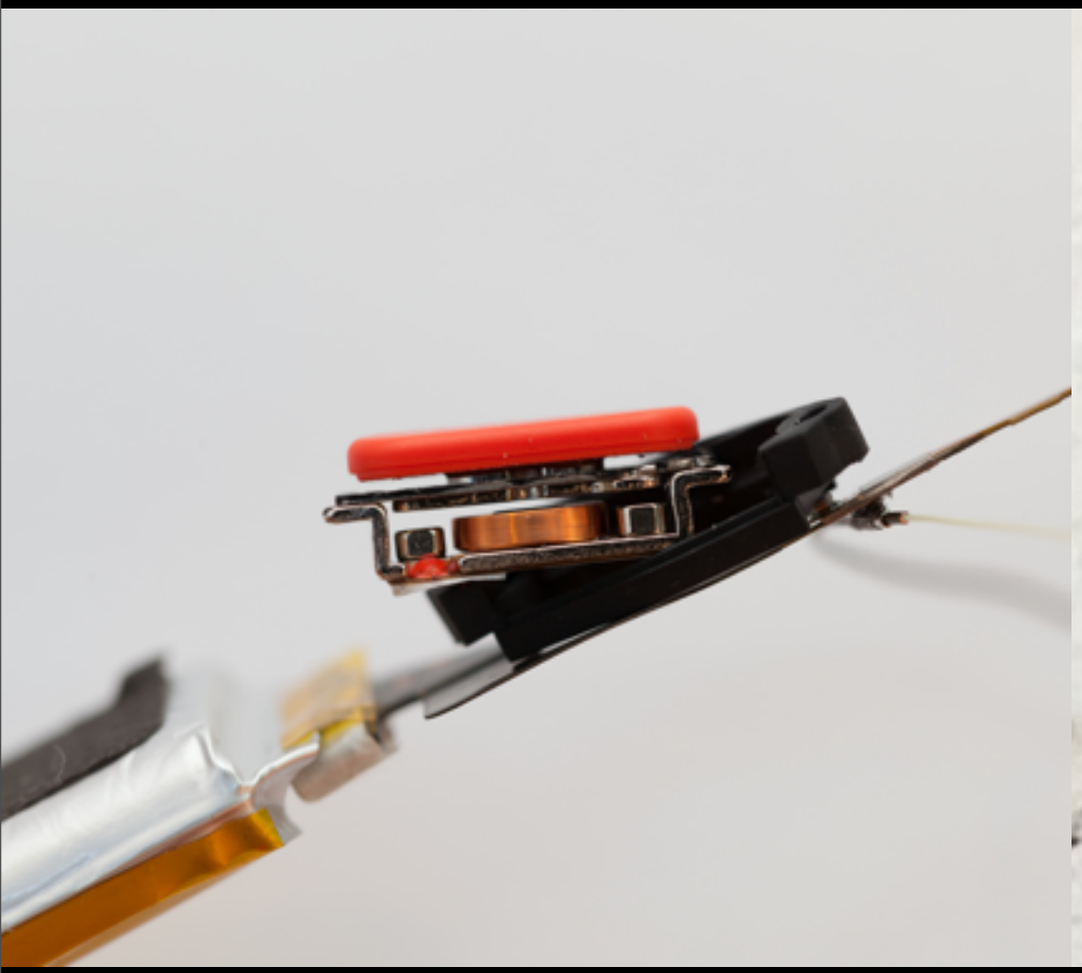


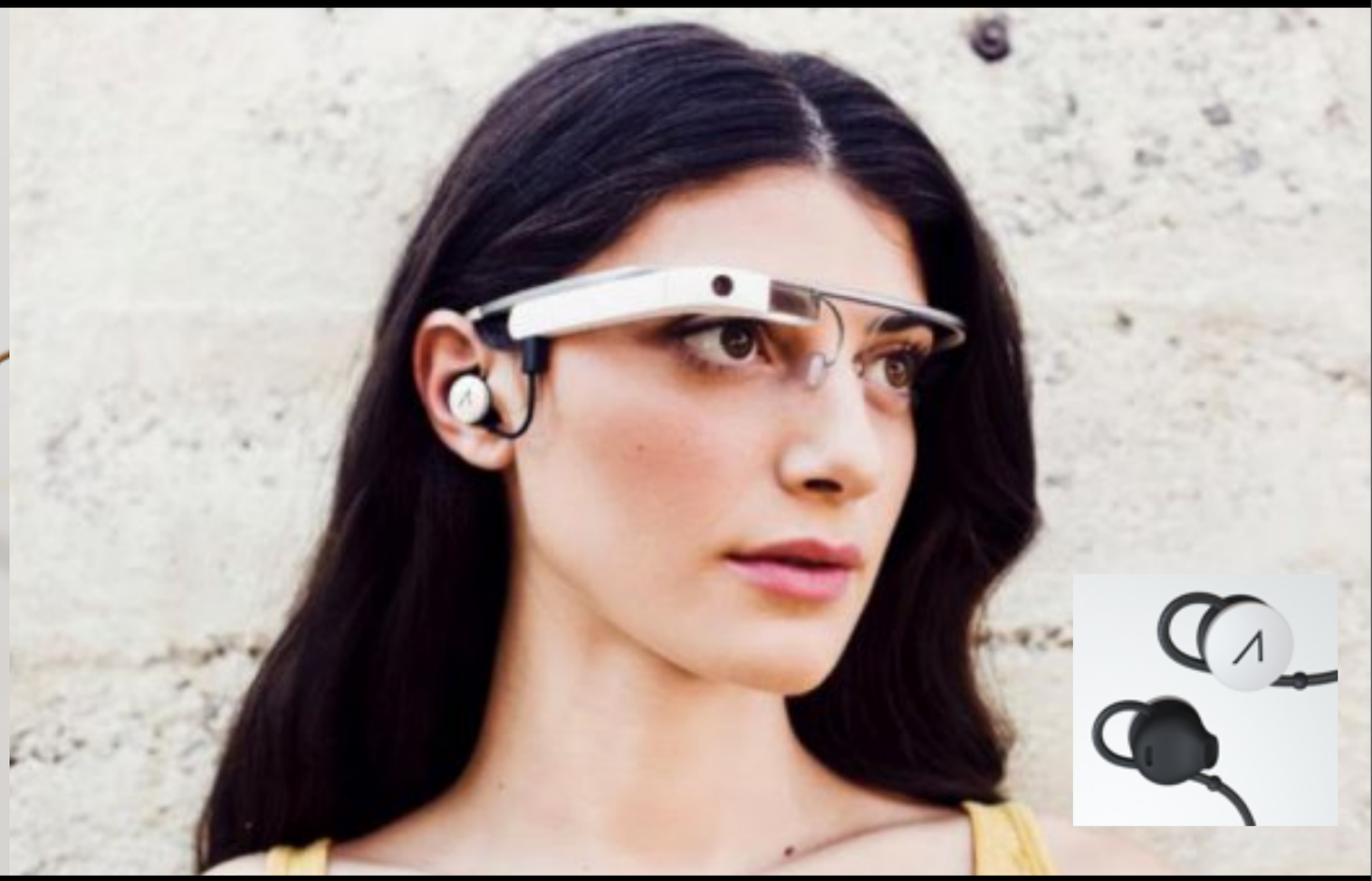
640 x 360 pixels

25 in at 8 ft / 64 cm at 2.4 m screen

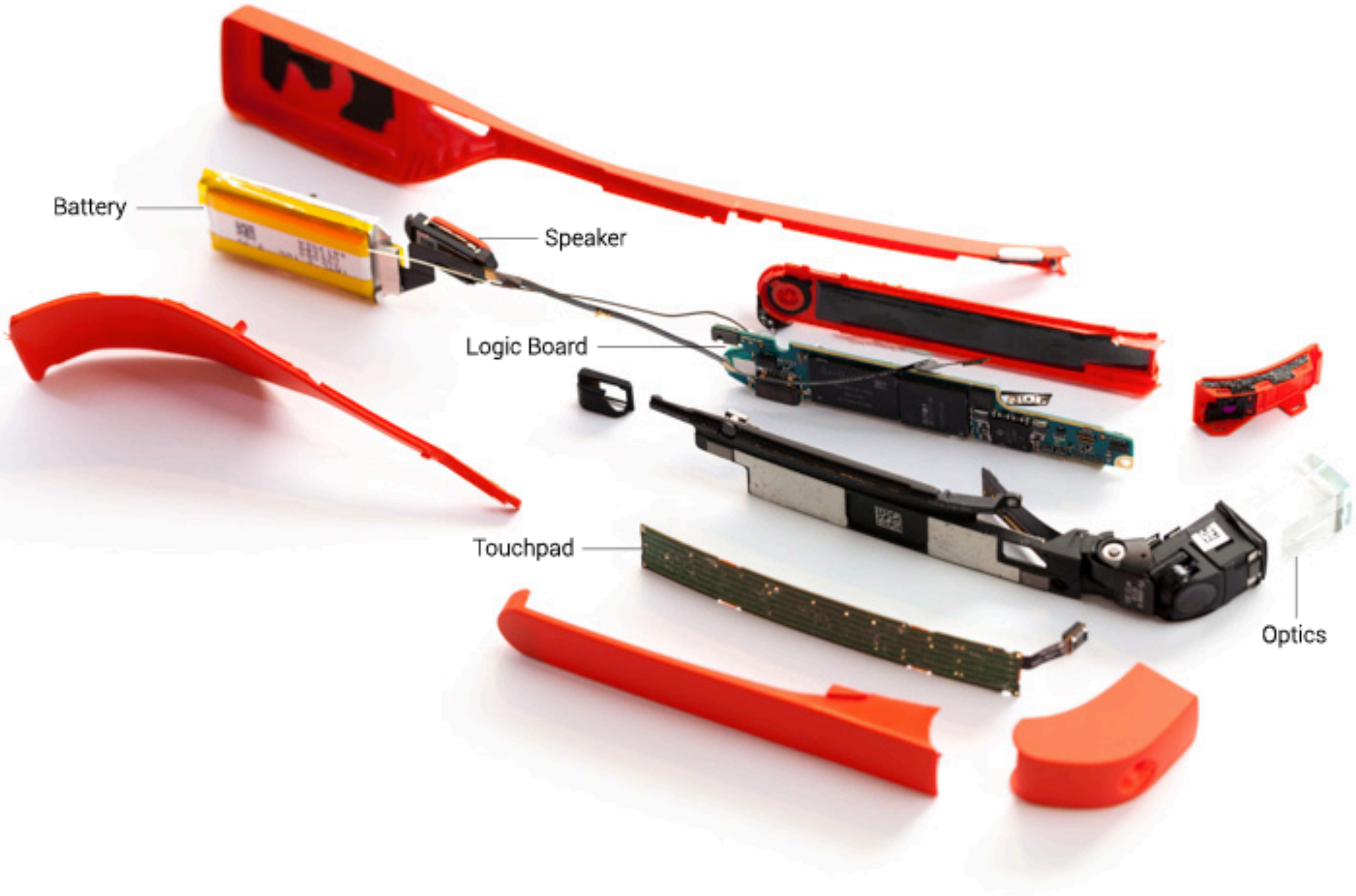
Size of Galaxy S3 screen at arm’s length

Audio Output





Audio output through bone conduction
Moving forward, earbuds available


Battery
Speaker
Logic Board
Touchpad
Optics

http://www.catwig.com/google-glass-teardown/

Interactions on Glass

Modalities

Touch

Buttons

Speech

Head motion

(blink)

Social Cues

Touching the side of device

Pressing a button

Speaking

Head motion

Transparent screen

“Looking up”

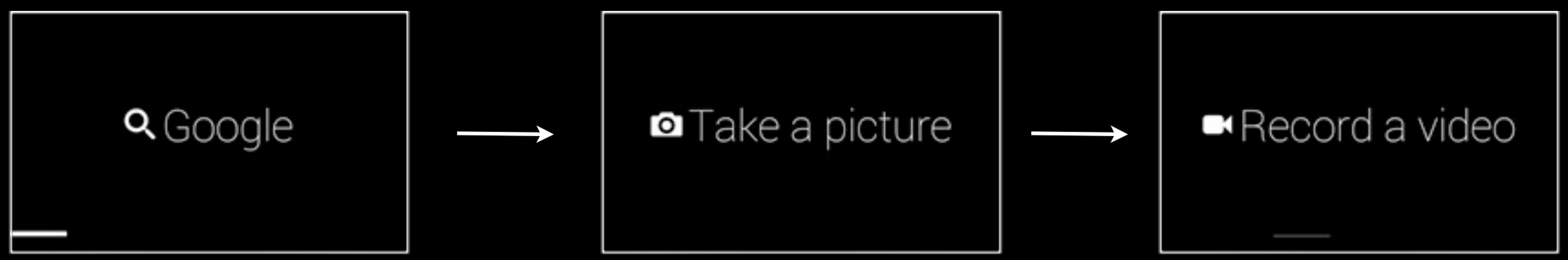
Demo

Timeline

Demo

Action Items

ok glass, google...
take a picture
record a video
get directions to...
send a message to...
make a call to



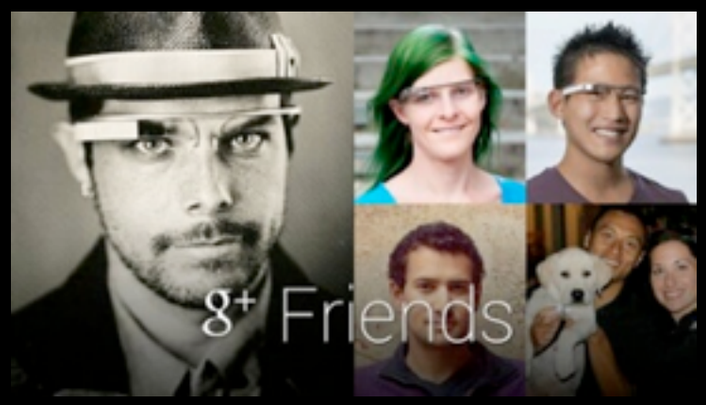
Glass User Interface

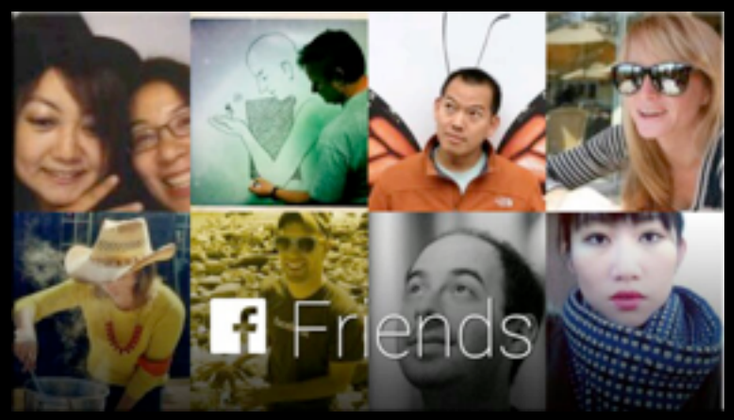
Demo

Timeline Cards

Menu Items

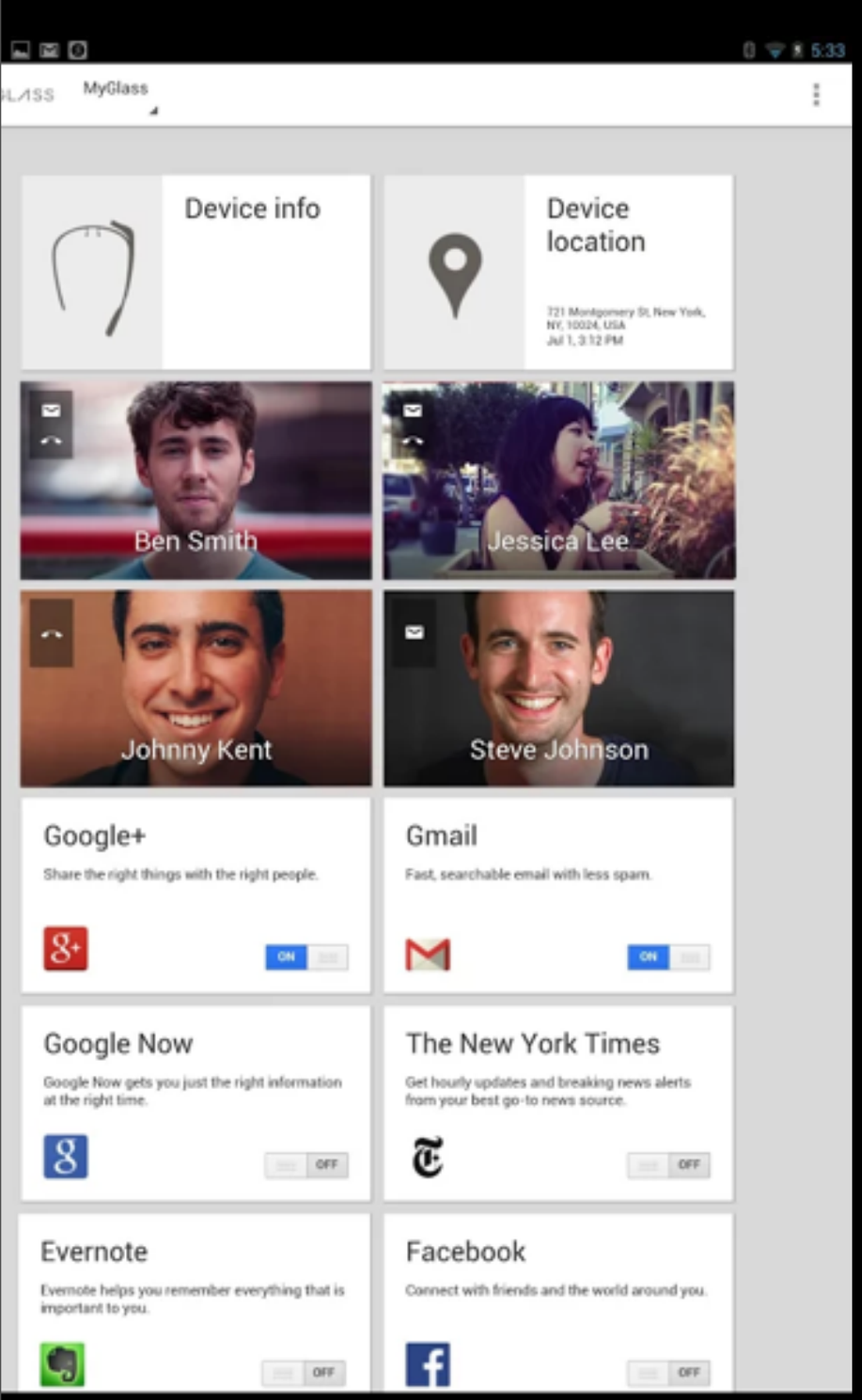
Contacts

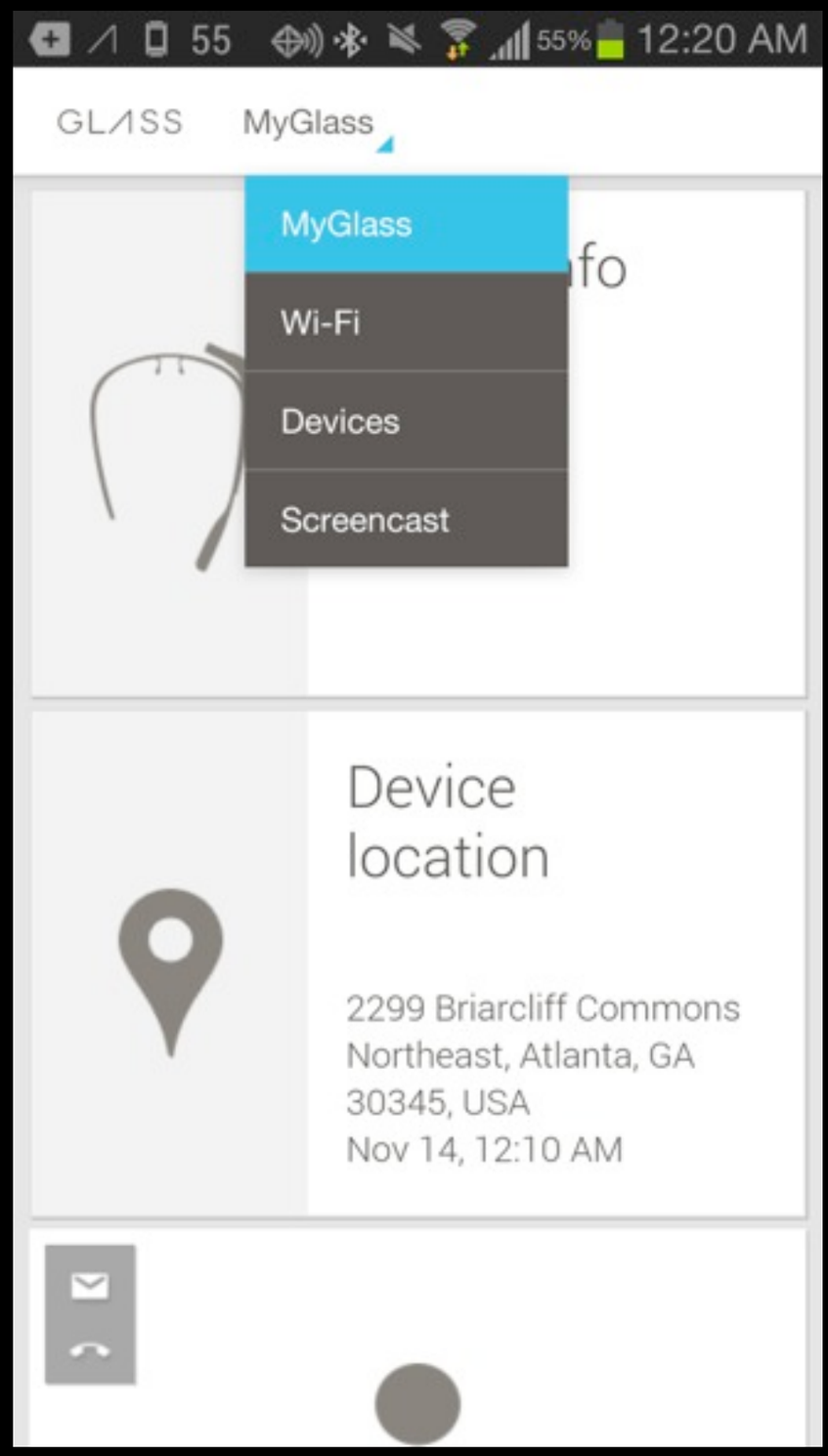


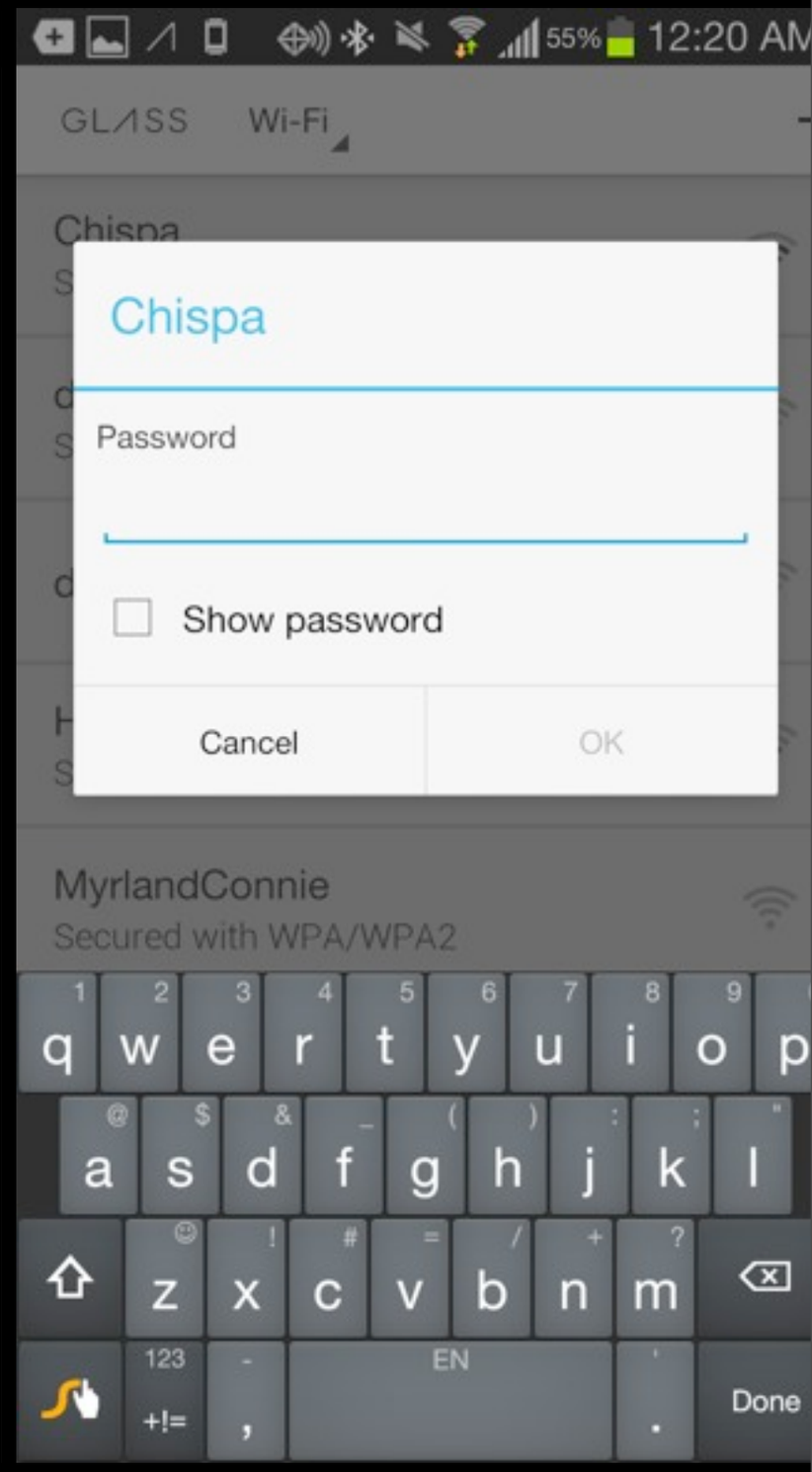


MyGlass App for Android

Demo







How to develop for Glass

Glass Software

Glass is built on the Android 4.0.4 platform

Today - develop using the Mirror API

Future - develop using Glass Development Kit

Glass Mirror API

Allows you to build applications called Glassware, currently these are web-based services

Glassware services interact with Google Glass and provide the functionality over a cloud-based API

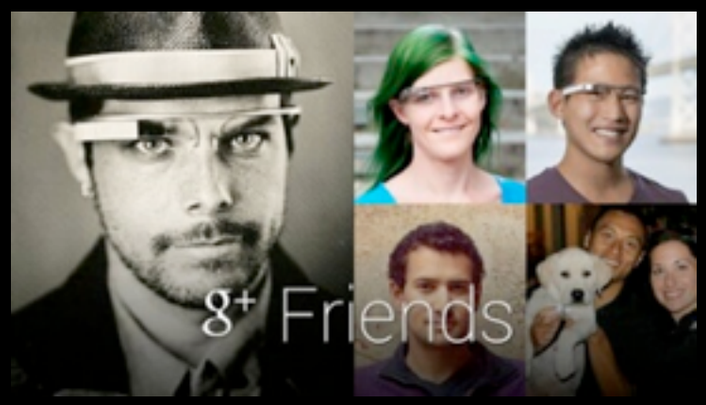
Does not require running code on Glass

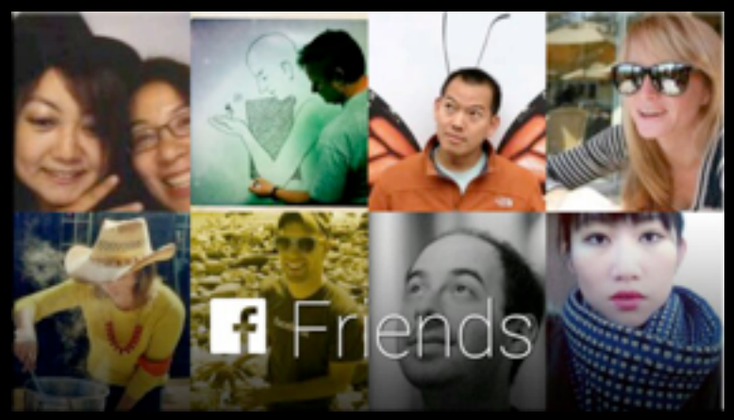
Glass Mirror API

Timeline Cards

Menu Items

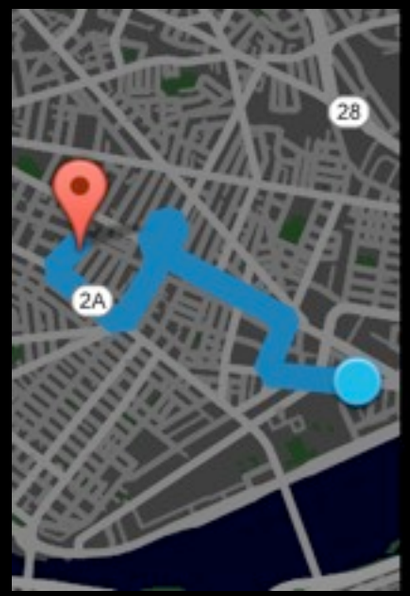
Contacts





Glass Mirror API

Subscriptions

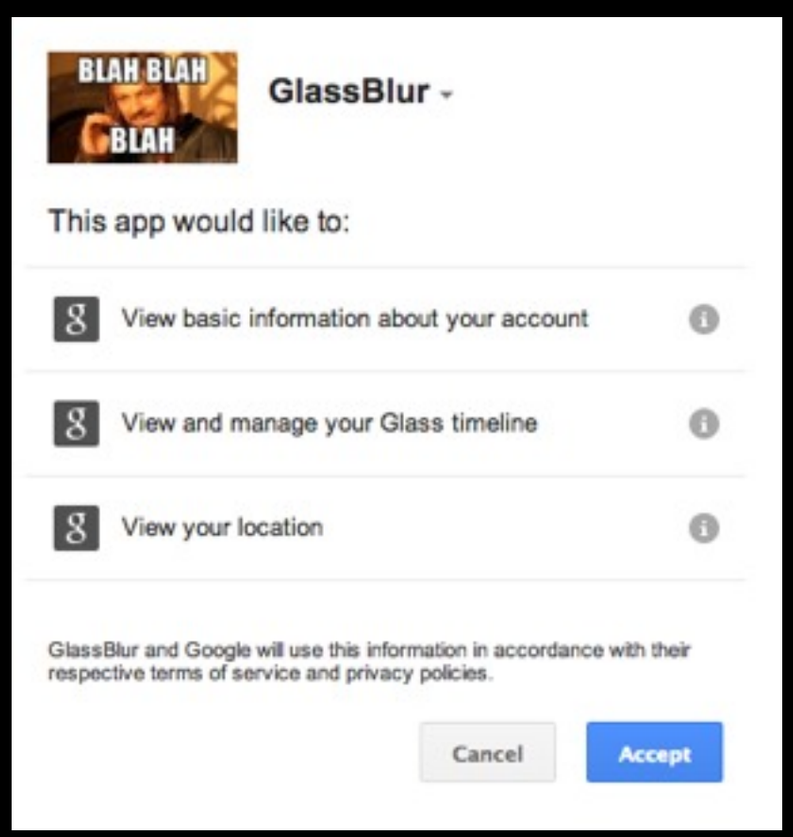


12 minutes to home
Medium traffic on Broadway

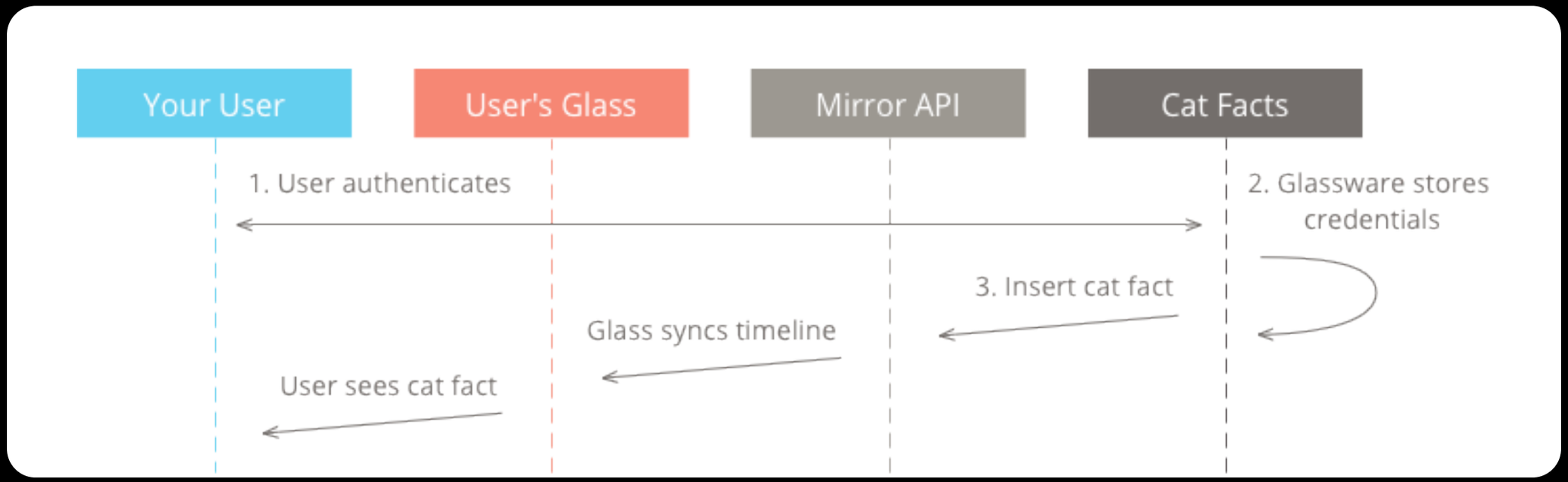
33 minutes ago

Location

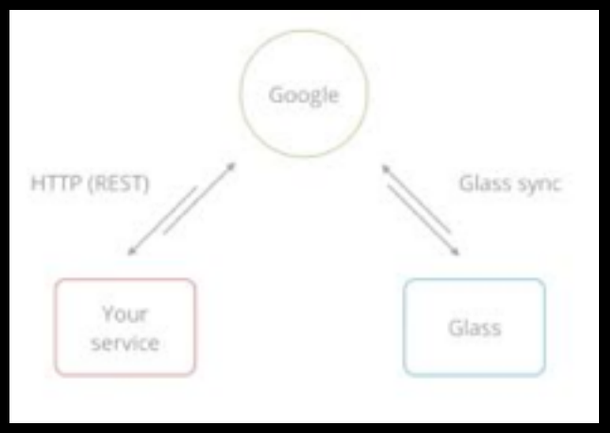
Authorization



Glass Mirror API



Application would be implemented using the timeline insert method



Google Mirror API Playground

Developer Preview

The Playground lets you experiment with how content is displayed on Glass. For more information on how to setup and use the Playground, see Playground Usage.

To authorize the Playground to send and receive data from your account, enter your Google APIs project client ID in the following text field and click **Authorize**.

Update Item

```
Hello Explorers,

Welcome to Glass!

+Project Glass
```

https://developers.google.com/glass/playground

Glass Development Kit (GDK)

Not released yet

Expected to be similar to Android SDK

Goal is to build Glassware in form of APKs

Currently develop Glassware using existing Android tools and SDK

Android on Glass

Use standard Android SDK tools to write APK files and sideload them onto Glass

Android SDK provides wide range of APIs:

Access the low-level hardware
Render OpenGL graphics
Use stock Android UI widgets
Android NDK to integrate native code

Android on Glass Examples

Demo

Head movement - uses inertial measurement sensors

Waveform - receives audio input from the microphone and displays waveform

Compass - visual and auditory output of current orientation

Level - uses sensors to determine horizon and provide visual feedback

Stopwatch - internal timer to provide stopwatch interface

Demo

Typing on Google Glass

Non-standard input modality

Modified Settings.apk and Launcher2.apk
Sideload Evernote.apk

Connect a Bluetooth wireless keyboard

Access Evernote web-based notes
- Add notes
- View stored notes

Demo

Google Play Music

Modified GooglePlayMusic.apk

Sideload onto Glass

Speech to activate and search music

Play through speakers

Best design practices for Glass

Developer Guidelines

Design for Glass

Don’t get in the way

Keep it timely

Avoid the unexpected

Design for Glass

Immediately accessible computing

Visual data overlay

No touch screen to interact with

System is aware of the user

Input via speech, taps, head motion, (blink), typing

Limited battery power

Don’t get in the way

Be there when the user wants it and out of the way when they don’t

Avoid frequent or unexpected notifications

Provide appropriate controls for users to interact

If a timeline from your service is missed, it should not degrade the user experience

Keep it timely

Most effective as a platform in-the-moment and up-to-date

Always deliver fresh and relevant content to users

Developers have access to a real-time notification system that can inform your Glassware about certain events

Respond to user actions (or notifications) in a timely and expected manner

Avoid the unexpected

Surprising the user with unexpected functionality is not advised on any platform, especially on Glass

Device worn on the user's face, and thus is much closer to a user's daily experience and senses

Always be honest about the intention of your Glassware

Get explicit permission before you do anything on the user's behalf

What's next?

Next steps

Optics/photonics

Miniaturization of components

Novel transducers and sensing

Ultra low power designs

Novel interactions

Artificial intelligence

Questions?

greyes@gatech.edu | www.gareyes.com | @greyesgt

Research opportunity

Looking for students with a strong research interest in wearable and ubiquitous computing

Multiple projects related to interactions using on-body hardware on the lower body, hands, above the neck

Skills with Android, Arduino, rapid prototyping, sensors, signal processing, machine learning

Research for course credit in Spring ’14

greyes@gatech.edu | www.gareyes.com | @greyesgt