
Google
Developers

ADK 2012

Android Open Accessory Development Kit

Eric Schlaepfer
Hardware Engineer

Dmitry Grinberg
Software Engineer

Travis Geiselbrecht
Software Engineer

Accessories for Android

- Connects using
  - USB (Android Open Accessory API)
  - Bluetooth (L2CAP, RFCOMM, or A2DP)
- Types
  - Commercial (Audio dock, exercise equipment, etc)
  - Hobbyist (Robot controller, etc)

What is it?

- Alarm clock
- Audio dock (Bluetooth and USB)
- Android accessory hacker platform

ADK 2012

Features

Light Sensor

Ambient light, proximity, colorimeter

Capacitive Slider

Volume, brightness, color, and more

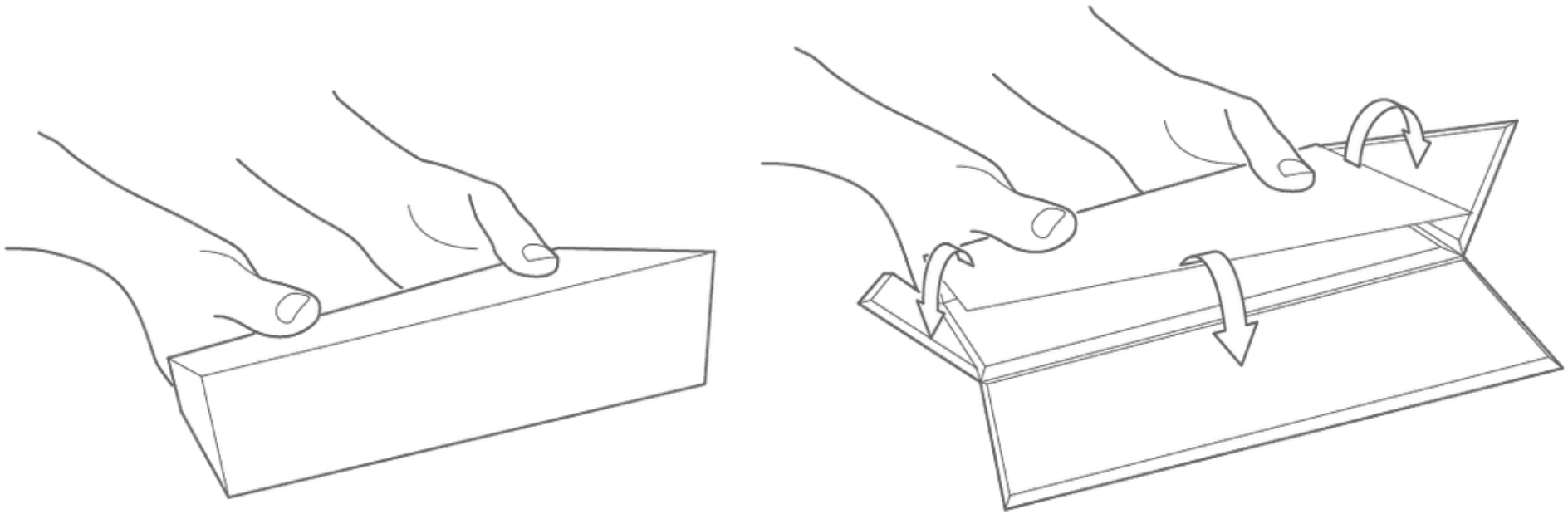
Capacitive Buttons

Two per digit (up and down)
One per mode icon

RGB LEDs

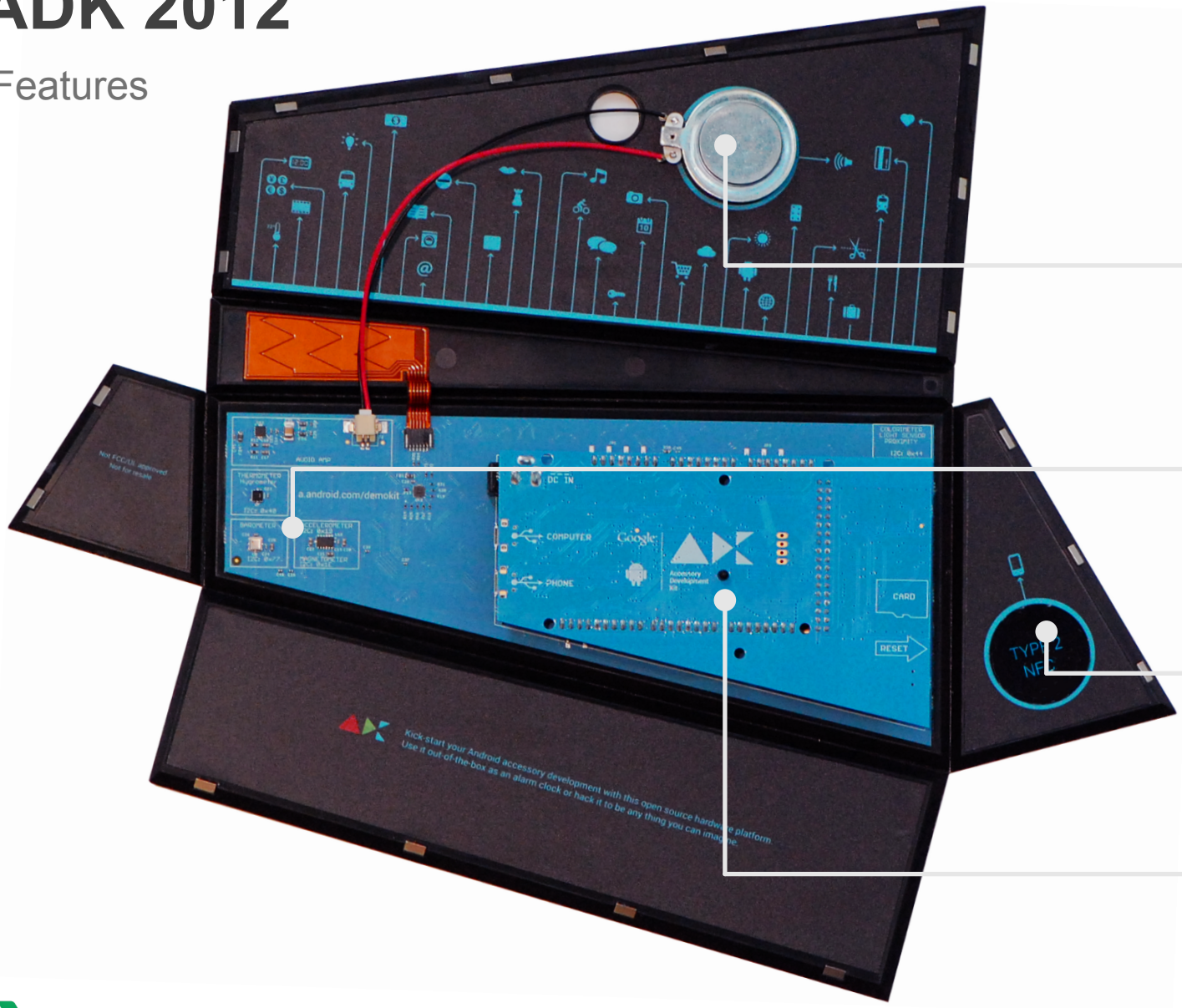
Six 7-segment digits
Eight mode icons

Let's Take it Apart



ADK 2012

Features



Speaker

Alarm sounds
Play audio from BT or USB

Sensors

Temperature, Humidity,
Barometric Pressure,
Acceleration, Magnetism

NFC Tag

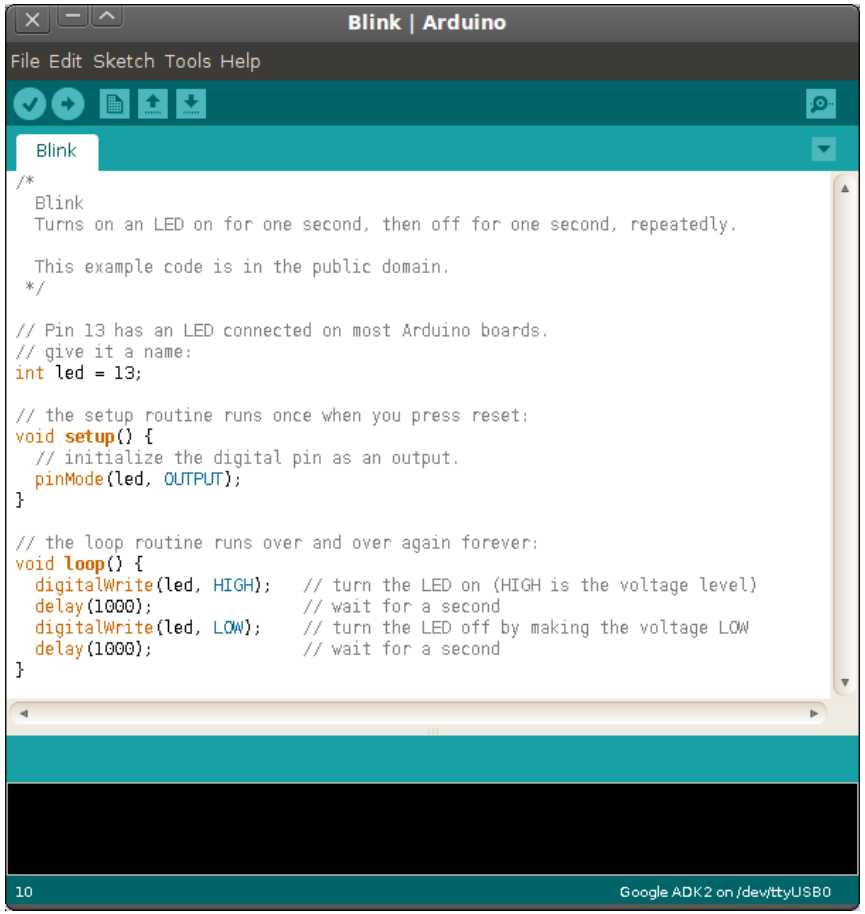
Hover phone to get app

Microcontroller

32-bit ARM Cortex M3
Bluetooth and SD card

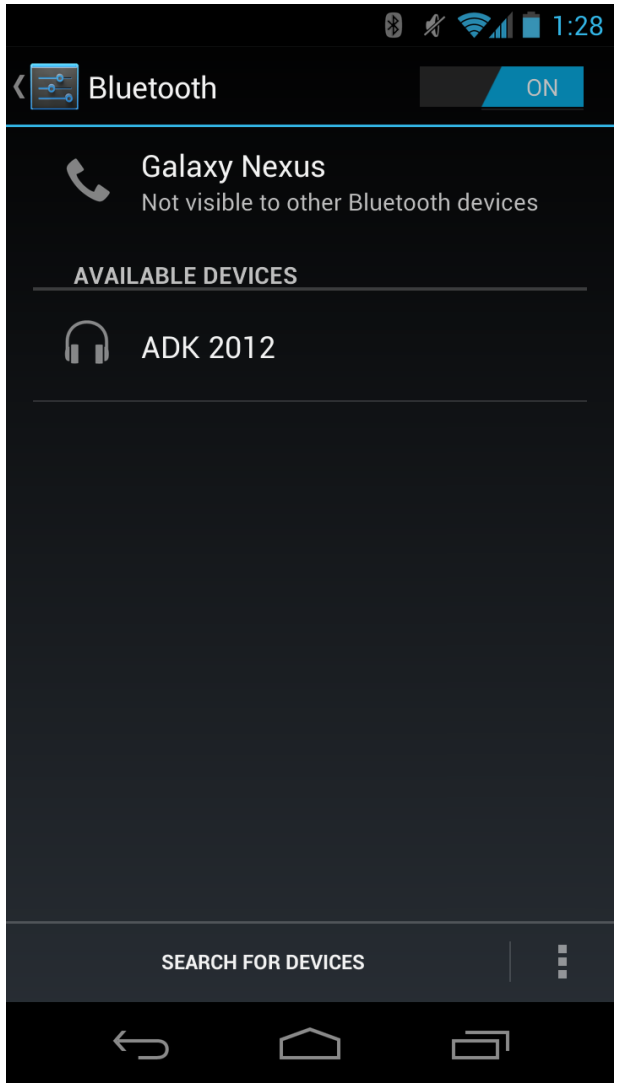
For Arduino

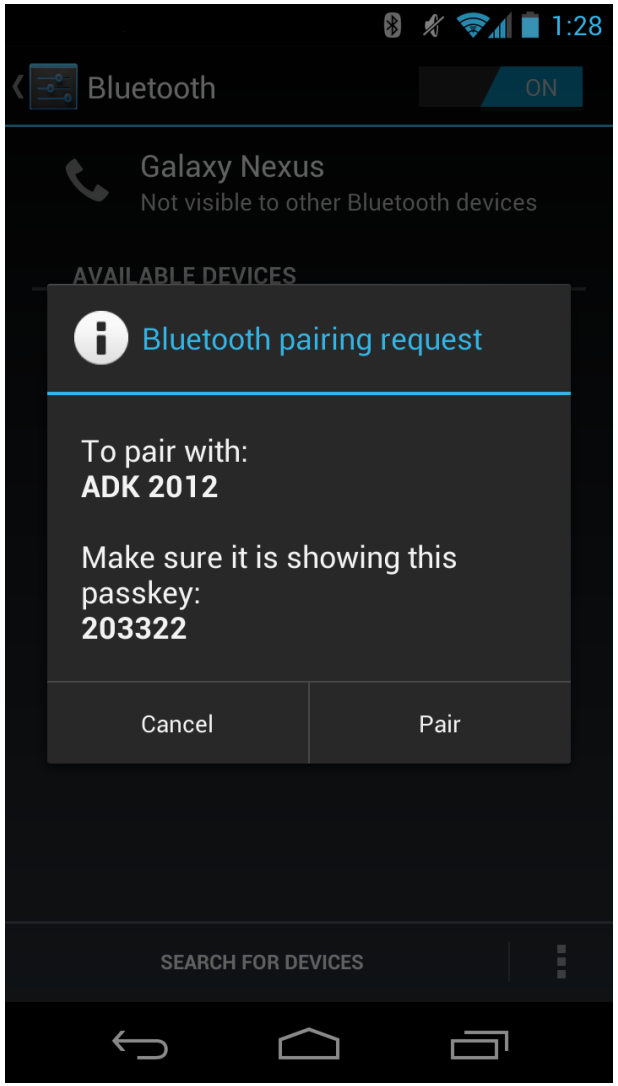
- Use the Arduino IDE!
- ADK board is 3.3V
  - Use caution with legacy shields
  - Uses ref pin

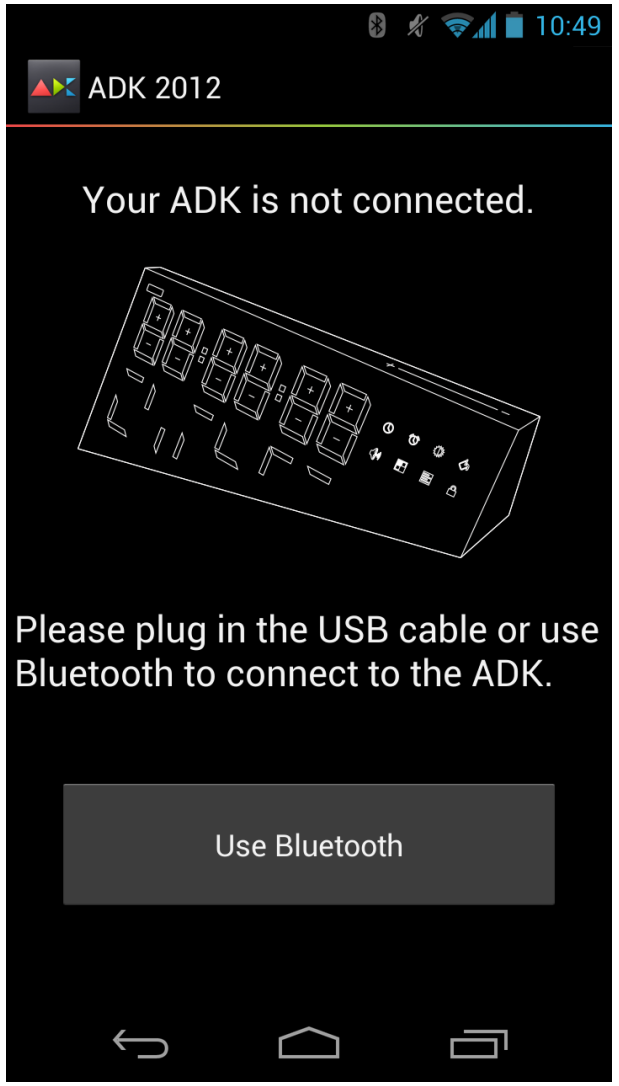


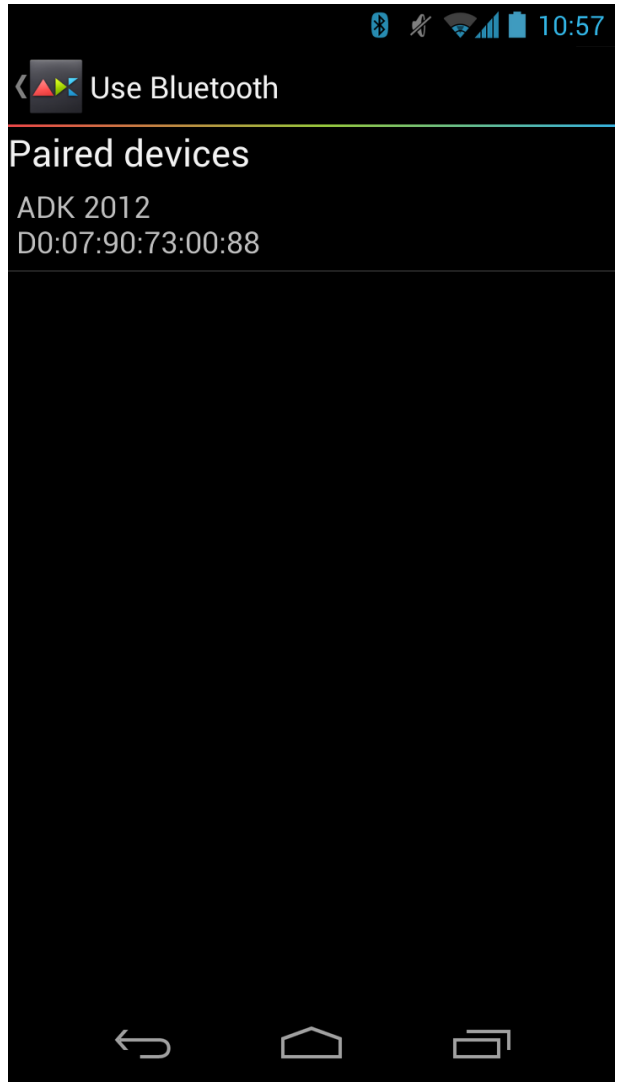
ADK 2012 Phone App

Connecting Using Bluetooth



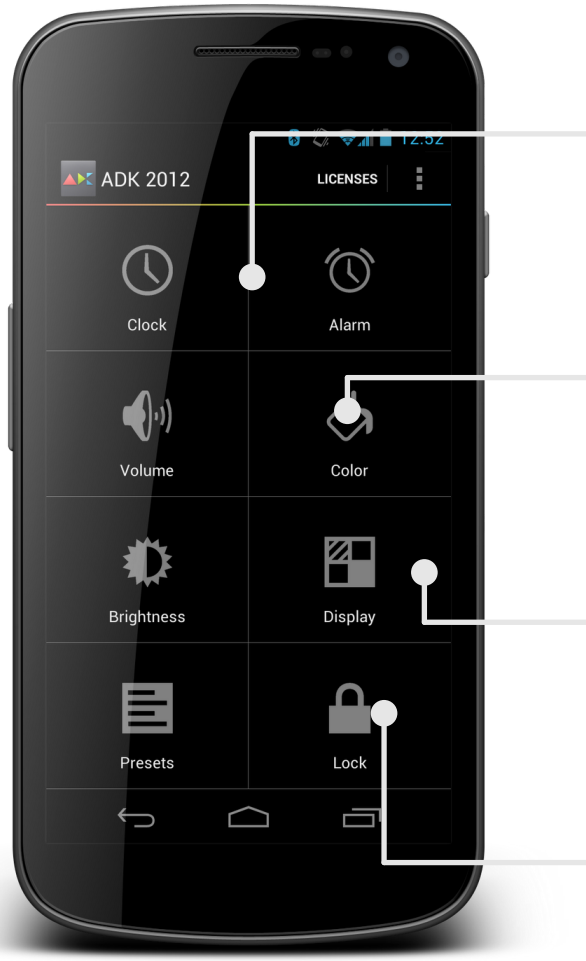






Connecting Using USB

Main Screen



Set Clock/Alarm

Sets time and alarm time

Set Color

Manually or from colorimeter

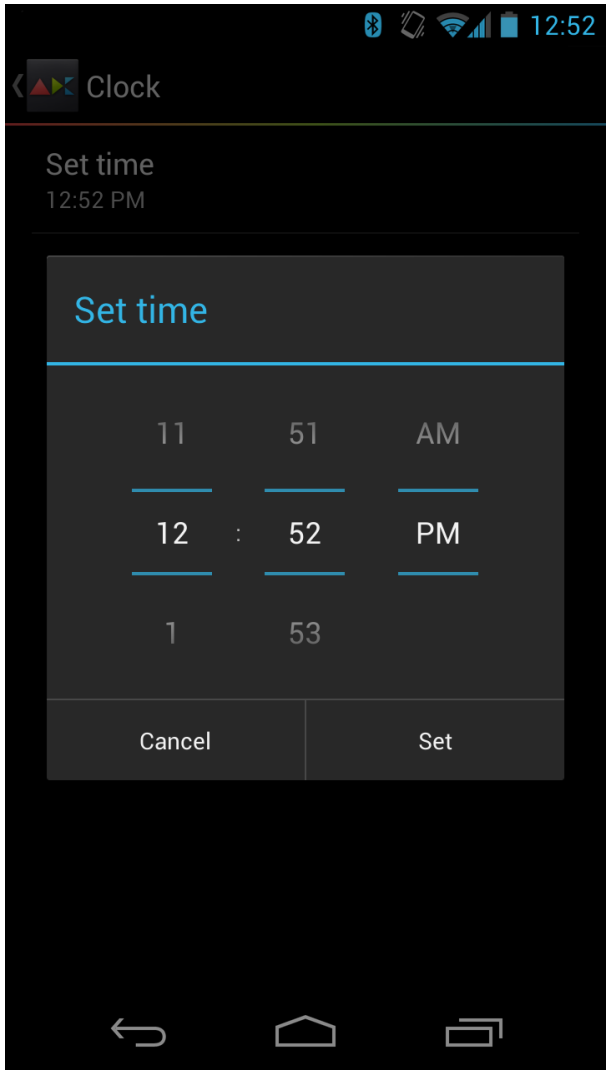
Choose Display

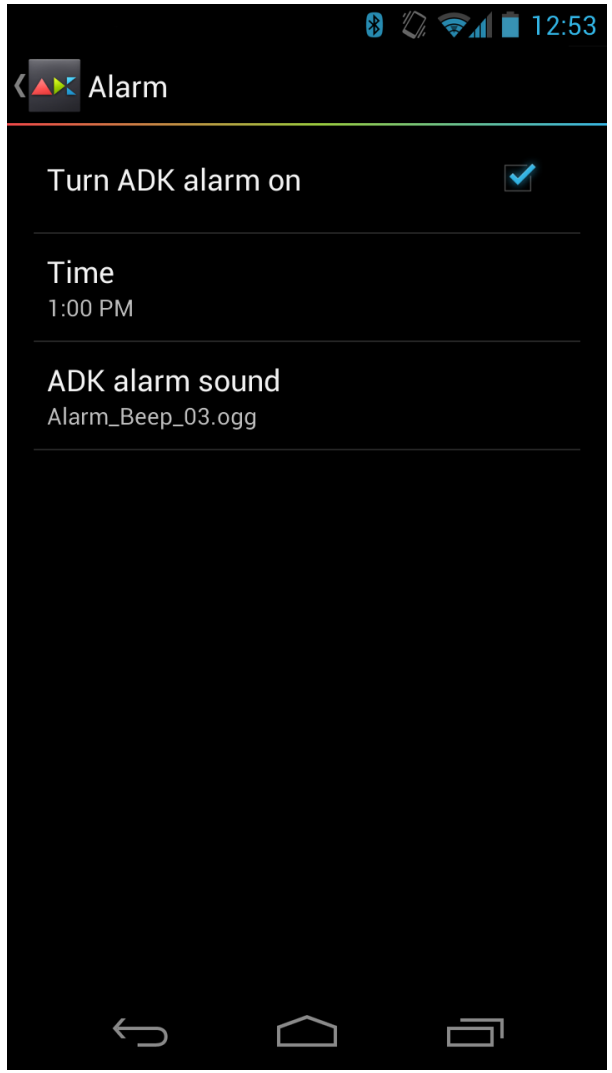
Display temperature, humidity, other parameters

Lock

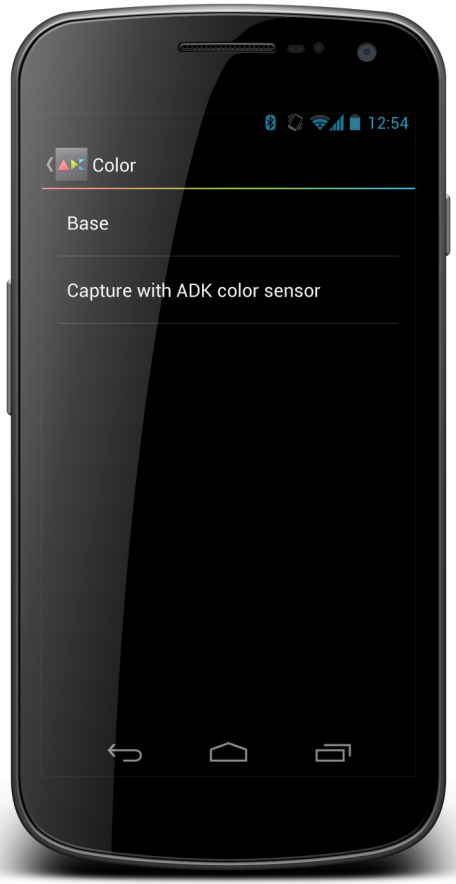
Locks capacitive buttons on clock

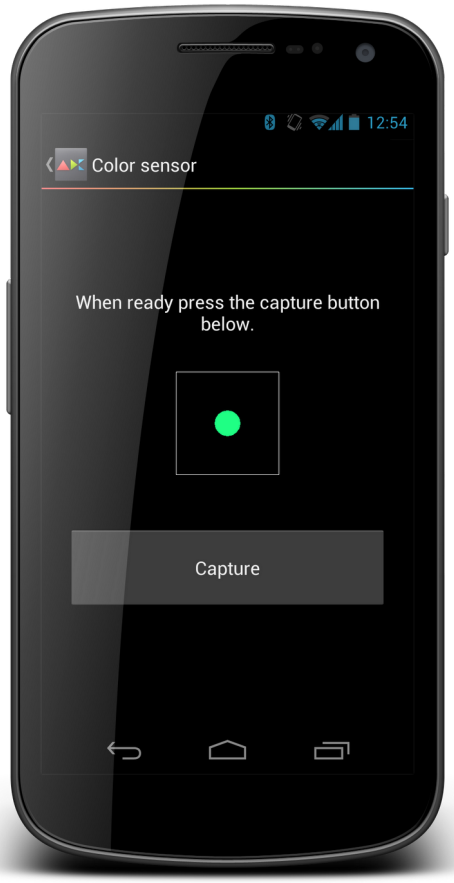
Setting the Time and Alarm





Setting the LED color





Bluetooth, Sound, and Storage

Feature Details

Bluetooth Hardware

- Real Bluetooth chip: CC2564
- HCI over UART at 1.75Mbps
- Onboard antenna

Bluetooth Stack

- Simple, bare metal, open source
  - BSD license
- Supports pairing and multiple concurrent connections
- Feature highlights
  - SDP server - dynamically add/remove services
  - RFCOMM - 1Mbps full duplex, simultaneous connections

C

```c
char btSetLocalName(const char* name);
char btGetRemoteName(uint8_t* mac, uint8_t PSRM, uint8_t PSM,
                     uint16_t co, char* nameBuf);
void btScan(void);
char btDiscoverable(char on);
char btConnectable(char on);
char btSetDeviceClass(uint32_t cls);
```

L2CAP

- Device accepts multiple concurrent L2CAP connections
  - Multiple hosts
- Very simple API
- Asynchronous sending and receiving
  - No need to keep buffers around

C

```c
void* (*serviceInstanceAllocate)(uint16_t conn, uint16_t chan,
                                 uint16_t remChan);
void (*serviceInstanceFree)(void* service);
void (*serviceRx)(void* service, const uint8_t* data, uint16_t size);
void l2capServiceTx(uint16_t conn, uint16_t remChan,
                    const uint8_t* data, uint32_t size);
void l2capServiceCloseConn(uint16_t conn, uint16_t chan);
char l2capServiceRegister(uint16_t PSM, const L2capService* svcData);
char l2capServiceUnregister(uint16_t PSM);
```

SDP

- Supports SDP (including fragmentation)
- Dynamically add or remove advertised services
- Runs in the background (no intervention required)
- No limit in number of services

c

```c
void btSdpServiceDescriptorAdd(const uint8_t* descriptor,
                               uint16_t descrLen);
void btSdpServiceDescriptorDel(const uint8_t* descriptor);
```

RFCOMM

- Device accepts multiple concurrent RFCOMM connections
  - Multiple hosts
- Very simple API
- Asynchronous sending and receiving
  - No need to keep buffers around
- Compatible with Android, Linux, etc

C

```c
static void adkPortRx(void* port,
                      uint8_t dlci,
                      uint8_t* data,
                      uint16_t sz)
{
  uint8_t *r;
  r = adkProcessCommand(data);
  if(r) {
    btRfcommPortTx(port, dlci,
                   r, r[0]);
    free(r);
  }
}
```

A2DP

- Compliant with A2DP 1.0 spec
  - Supports mono, all three stereo variants
- Efficient SBC decoder (ASM)
- Tested with Android, Linux, etc

Bluetooth (Phone side)

- See BTConnection.java

Java

```
mAdapter = BluetoothAdapter.getDefaultAdapter();
BluetoothDevice device = mAdapter.getRemoteDevice(address);
mSocket = device.createInsecureRfcommSocketToServiceRecord(ADK_UUID);
mSocket.connect();

mInStream = mSocket.getInputStream();
mOutStream = mSocket.getOutputStream();
```

Sound

- DMA from RAM to DAC output
- Super-simple API
- Sources
  - SD Card
  - Vorbis Tremor Ogg decoder
  - Bluetooth A2DP
  - USB

C

```c
void audioOn(int source,
             uint32_t samplerate);
void audioAddBuffer(int source,
                    uint16_t* samples,
                    uint32_t numSamples);
int audioTryAddBuffer(int source,
                      uint16_t* samples,
                      uint32_t numSamp);

void playOgg(const char* path);
void playOggBackground(char* path,
                       char *complete,
                       char *abort);
void setVolume(uint8_t vol);
uint8_t getVolume(void);
```

Storage

- Built-in MicroSD slot
- SD/SDHC driver
- FAT12/FAT16/FAT32 filesystem driver
  - Read and write
  - Mkfs, rename, etc

C

```
char fatfsOpen(FatFile**, char* path,
               uint8_t mode);
char fatfsRead(FatFile*, void* buf,
               uint32_t num,
               uint32_t* done);
char fatfsWrite(FatFile*, void* buf,
                uint32_t num,
                uint32_t* done);
char fatfsSeek(FatFile*, uint8_t whence,
               int32_t pos);
char fatfsOpenDir(FatDir**, char* path);
char fatfsReadDir(FatDir*, FatFileInfo*);
char fatfsUnlink(char* path);
char fatfsMkdir(char* path);
```

USB Host, USB Audio

Feature Details

USB Host

- Built in USB OTG port, running in host mode
- Full source to the host stack, implemented in the ADK library
- Implements Android Open Accessory Protocol

```c
void usbStart();
void usbSetAccessoryStringVendor(const char *str);
void usbSetAccessoryStringName(const char *str);
void usbSetAccessoryStringLongname(const char *str);
void usbSetAccessoryStringVersion(const char *str);
void usbSetAccessoryStringUrl(const char *str);
void usbSetAccessoryStringSerial(const char *str);

int  accessoryConnected();
int  accessorySend(const void *buf, unsigned int len);
int  accessoryReceive(void *buf, unsigned int len);
```

USB Host (Clock side)

C

```
ADK L;
void setup() {
  L.adkInit();
  L.usbSetAccessoryString...
  L.usbStart();
}

void loop() {
  if (L.accessoryConnected()) {
    int recvLen = L.accessoryReceive(msg, sizeof(msg));
    if (recvLen > 0) {
   ... // process message
       }

    L.accessorySend(outmsg, outmsgLen);
  }
  L.adkEventProcess();
}
```

USB Host (App side)

- Uses the USB Accessory API
- Supported in Android 2.3.4+
- See UsbConnection.java

Java

```
import android.hardware.usb.UsbAccessory;
import android.hardware.usb.UsbManager;

mUSBManager = (UsbManager)
                getSystemService(
                Context.USB_SERVICE );
UsbAccessory acc = (UsbAccessory)
                      intent.getParcelableExtra(
                      UsbManager.EXTRA_ACCESSORY );
if (!mUSBManager.hasPermission(acc)) return;
ParcelFileDescriptor mFD =
              mUSBManager.openAccessory(acc);
if (mFD != null) {
  FileDescripter fd = mFD.getFileDescriptor();
  mIS = new FileInputStream(fd);
  mOS = new FileOutputStream(fd);
}
```

USB Audio

- Works like a USB audio device
- Streams audio from phone
- Requires Jellybean (no additional app required)

Device

Host

LEDs, Buttons, and Sensors

Feature Details

LEDs

- 64 RGB LEDs
  - Six 7-segment digits
  - Two colons
  - Eight icon indicators
  - Twelve extra "party mode" lights
- Simple API controls the display
  - Interrupt driven, framebuffer-like

C

```c
void ledWrite(uint8_t led_id, uint8_t r, uint8_t g, uint8_t b);
void ledDrawIcon(uint8_t icon, uint8_t r, uint8_t g, uint8_t b);
void ledDrawLetter(char letter, uint8_t val, uint8_t r, uint8_t g,
uint8_t b);
```

Capacitive Touch

- 12 up/down buttons for digits
- 8 mode icon buttons
- Slider control on the top

- Simple interface to Atmel AT42QT2120
- btnProcess() in clock sketch handles button events, key repeat, debouncing

C

```c
static uint16_t btnProcess();

uint16_t capSenseButtons(void);
uint16_t capSenseIcons(void);
uint8_t capSenseSlider(void);
```

Sensors

- Simple API
- Handles calibration calculations

Thermometer
Hygrometer

Temperature (C)
Relative humidity (%)

Barometer

Barometric pressure(kPa)
Can work as an altimeter

Accelerometer
Magnetometer

Orientation (where is up)
Motion
Compass

Colorimeter

Ambient light levels
Ambient IR level
IR proximity
Color of incident light

Sensors

C

```
uint16_t prox, clear, ...;
long pressure, temp, ...;
alsRead(&prox, &clear, &r, &g, &b, &infrared, &temp);
baroRead(oversample, &pressure, &temp);
hygroRead(&temp, &humidity);
accelRead(&x, &y, &z);
magRead(&x, &y, &z);
```

The Clock Sample

Excerpts

Code Excerpts

C++

```
L.adkEventProcess(); //let the adk framework do its thing
...
L.rtcGet(&year, &month, &day, &h, &m, &s);
L.ledDrawLetter(0, h / 10 + '0', r, g, b);
L.ledDrawLetter(1, h % 10 + '0', r, g, b);
...
L.playOggBackground(settings.almTune, &alarmEnded,
&alarmStop);
...
processUSBAccessory();
```

Code Excerpts

C++

```
static void btAdkPortRx(void* port, uint8_t dlci,
                        uint8_t* data, uint16_t sz) {
...
sendSz = adkProcessCommand(cmd, cmdBuf + 4, cmdSz, 1, reply + 4,
MAX_PACKET_SZ - 4);
...
L.btRfcommPortTx(port, dlci, reply, sendSz);
```

Thank You!

Visit a.android.com/demokit for downloads and documentation

+Eric Schlaepfer

+Dmitry Grinberg

+Travis Geiselbrecht

Google
Developers