

Skinning Android for Embedded Applications



**Start-up Screen
Customization**

Direct Boot to Application

Full Screen Display Mode

Q & A

A green arrow pointing to the right, with a white chevron on its right side. It has a white vertical bar on its left side.

Start-up Screen Customization

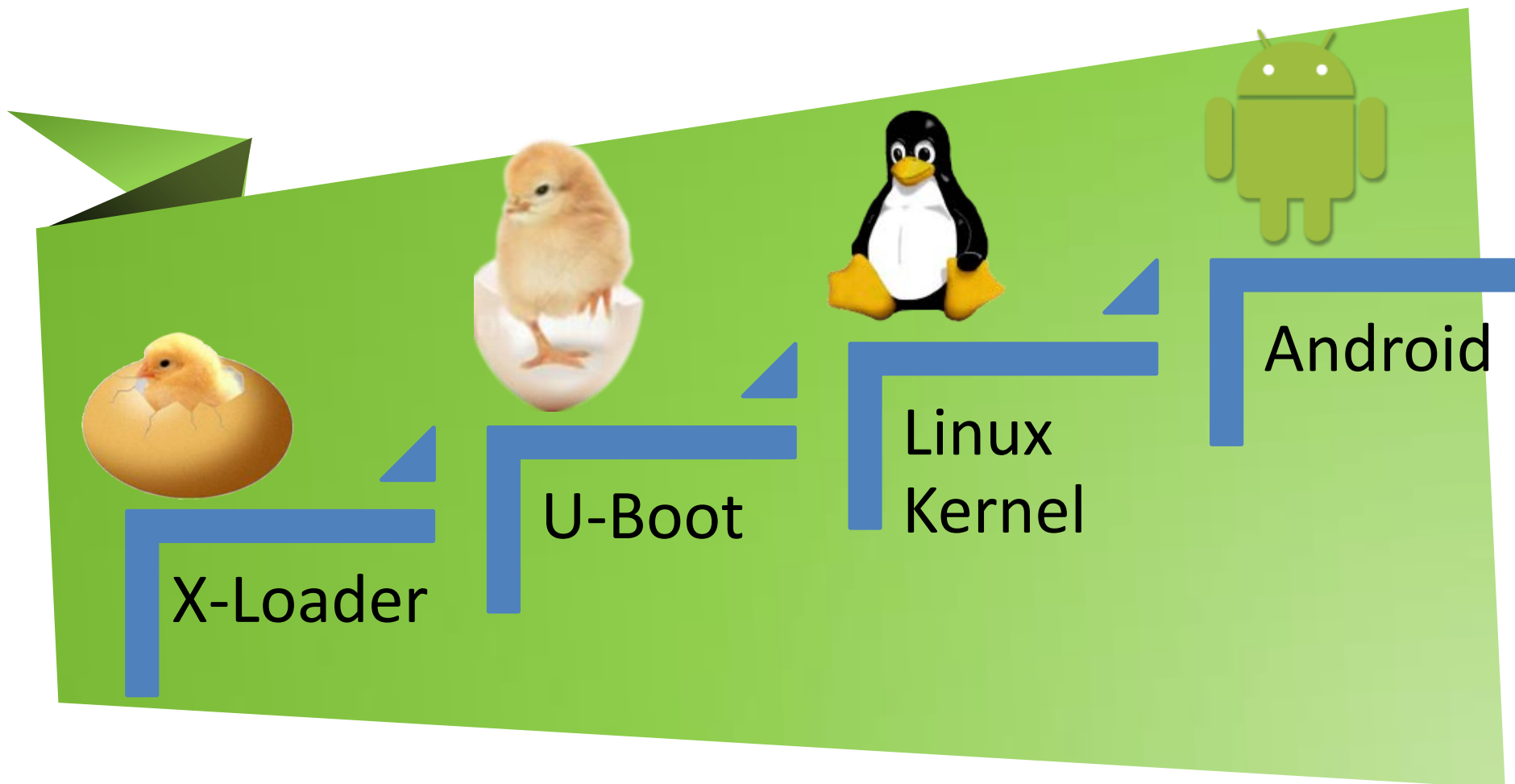
A grey arrow pointing to the right, with a white chevron on its right side. It has a white vertical bar on its left side.

Direct Boot to Application

A grey arrow pointing to the right, with a white chevron on its right side. It has a white vertical bar on its left side.

Full Screen Display Mode

System Boot Process



Start-up Screen Customization



X-Loader



U-Boot

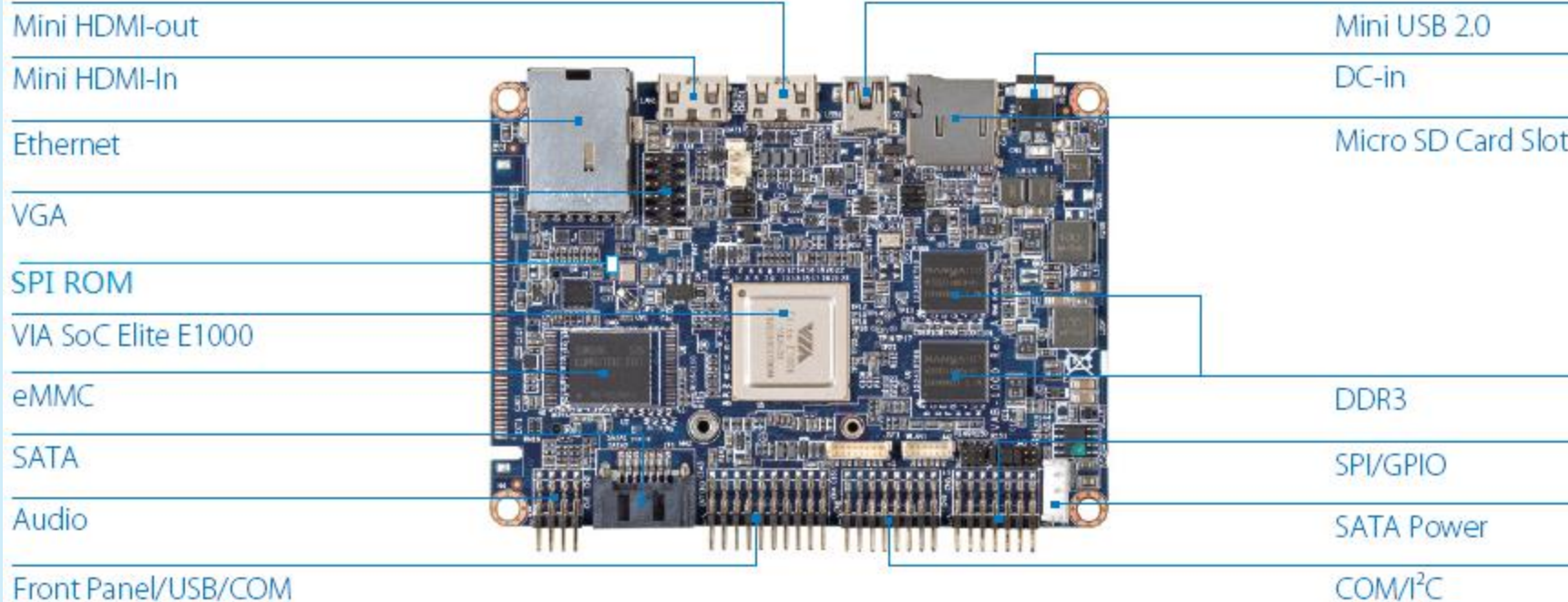


Linux
Kernel



Android

VAB-1000 Specifications





E-Loader (Elite 1000)

Machine Code

```
reset:
    @bl cpu_init_crit
    mrs r0, cpsr
    bic r0, r0, #0x1f
    orr r0, r0, #0xd3
    msr cpsr,r0

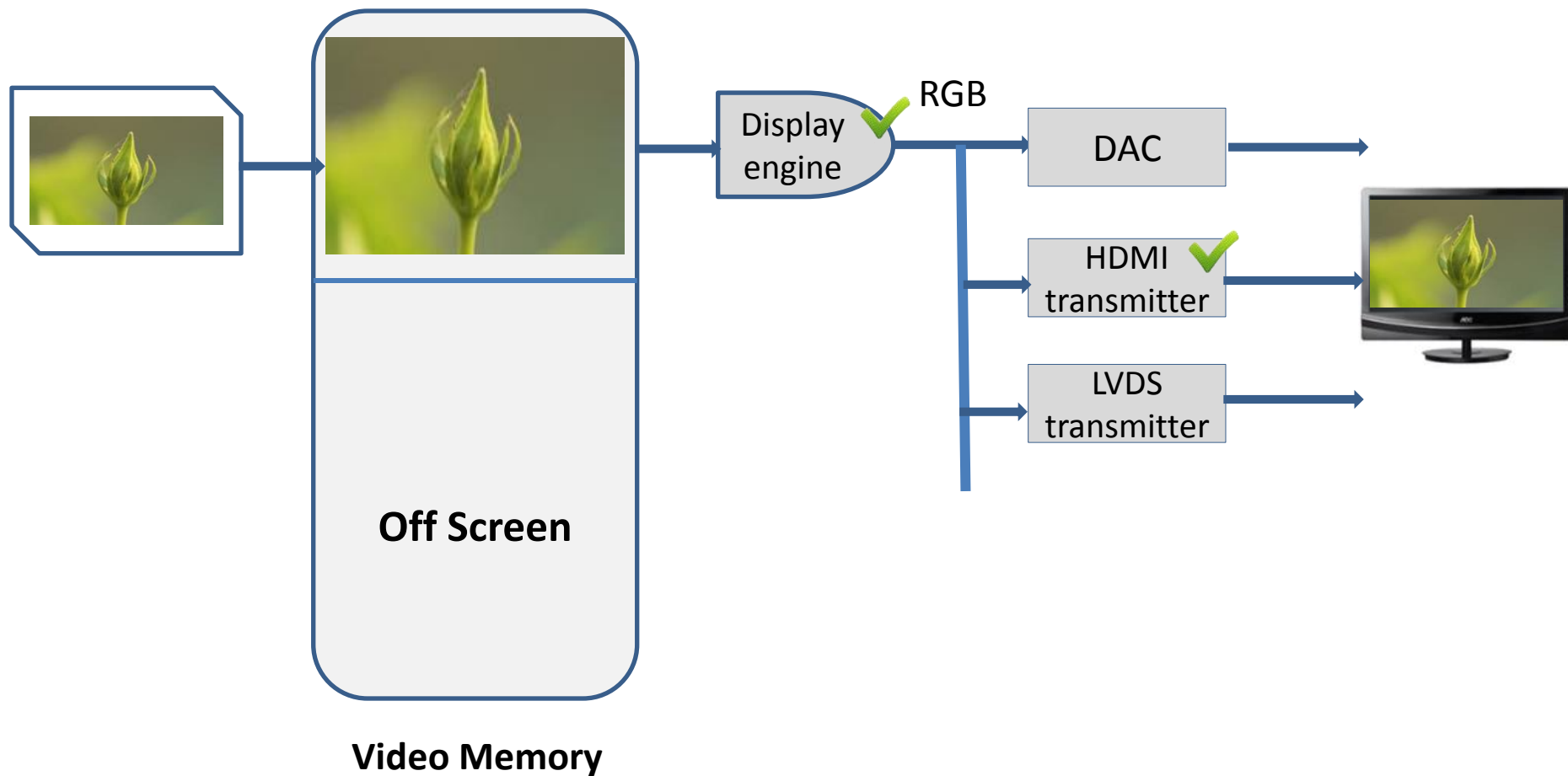
_start_armboot:
    .word    main
```

```
cpu_init_crit:
    /* set the cpu to SVC32 mode
    */
    mrs r0, cpsr
    bic r0, r0, #0x1f
    orr r0, r0, #0xd3
    msr cpsr,r0
    .....
```

C Code

```
void main(void)
{
    load_timing_table();
    init_sdram_timing();
    switch_cpu_freq();
    serial_init();
    printf("\n%s start up!\n", BINARY_VERSION());
    load_boot_image();
    boot_entry = (b12*)BOOT_MEM_ADDR;
    boot_entry();
}
```

Display System Fundamentals





How to Display a Logo in U-Boot

eMMC

- Mmc init 0 ext4load mmc 0:1 0x1000000 splash.data
- Mmc init 2 mmc write 0x1000000 0xE0 0x7200

Memory

- `addr = (u8*) adapter->fb_base + IGA1_FB_OFFSET;`
- `dev_desc = mmc_get_dev(emmc_dev);`
- `dev_desc->block_read(2, START, BOOTLOGO_SIZE, addr);`

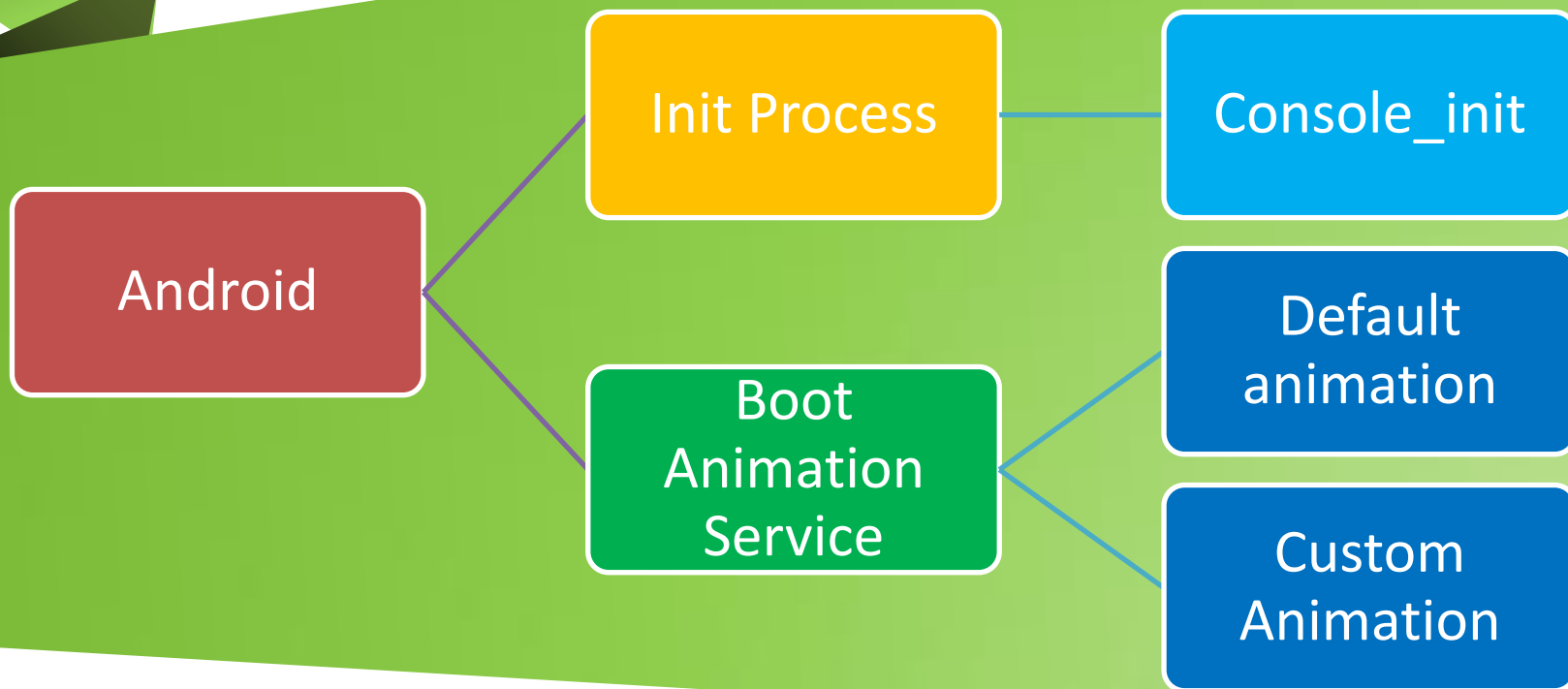
Display Engine

- Width, Height, depth, refresh rate, start address...

HDMI

- Get ready to transmit display engine's data to HDMI format


How to Make an Android Startup Logo





Putting a Logo into Console_init

```
#define INIT_IMAGE_FILE "/initlogo.rle"
static int console_init_action(int nargs, char **args)
{
    ....
    if( load_565rle_image(INIT_IMAGE_FILE) ) {
        fd = open("/dev/tty0", O_WRONLY);
        if (fd >= 0) {
            const char *msg;
            msg = "\n" "\n" "\n" "\n" "\n" "\n"
                "\n" // console is 40 cols x 30 lines
                "\n" "\n" "\n" "\n" "\n" "\n"
                "      A N D R O I D ";
            write(fd, msg, strlen(msg));
            close(fd);
        }
    }
    return 0;
}
```



```
int load_565rle_image((char *fn)
{
    fd = open(fn, O_RDONLY);
    data = mmap(0, s.st_size, PROT_READ, MAP_SHARED, fd, 0);
    fb_open(&fb)
    bits = fb.bits;
    android_memset16(fb.bits, data, n << 1);
}
```



Putting a Logo in the Boot Animation



init.rc

```
service bootanim /system/bin/bootanimation
    class main
    user graphics
    group graphics
    disabled
    oneshot
```



**frameworks\base\cmds\bootanimation\
BootAnimation.cpp**

```
bool BootAnimation::threadLoop()
{
    bool r;
    if (mAndroidAnimation) {
        r = android();
    } else {
        r = movie();
    }
}
```

Putting a Logo in the Boot Animation

Default Effect



`android/frameworks/base/core/
res/assets/images`



android-logo-mask.png



android-logo-shine.png

Putting a Logo in the Boot Animation



bootanimation.zip

- ▶ "/data/local/bootanimation.zip"
- ▶ "/system/media/bootanimation.zip"
- ▶ "/system/media/bootanimation-encrypted.zip"

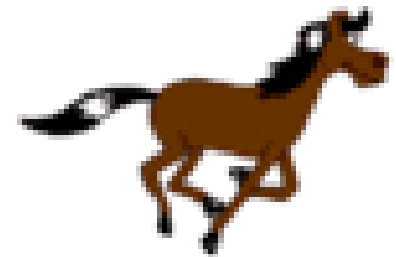


/bootanimation.zip → Desc.txt

▶ 720 480 5

▶ p 1 0 part1

▶ p 0 0 part2



Boot Screen Results



A grey button with a white chevron pointing right. A white vertical bar is on the left side of the button.

Start-up Screen Customization

A red button with a white chevron pointing right. A white vertical bar is on the left side of the button.

Direct Boot to Application

A grey button with a white chevron pointing right. A white vertical bar is on the left side of the button.

Full Screen Display Mode

4 Components of Android Applications



Android Application Component #1

Activity



An activity is a visual interaction with the user interface



An application may contain one or more activities



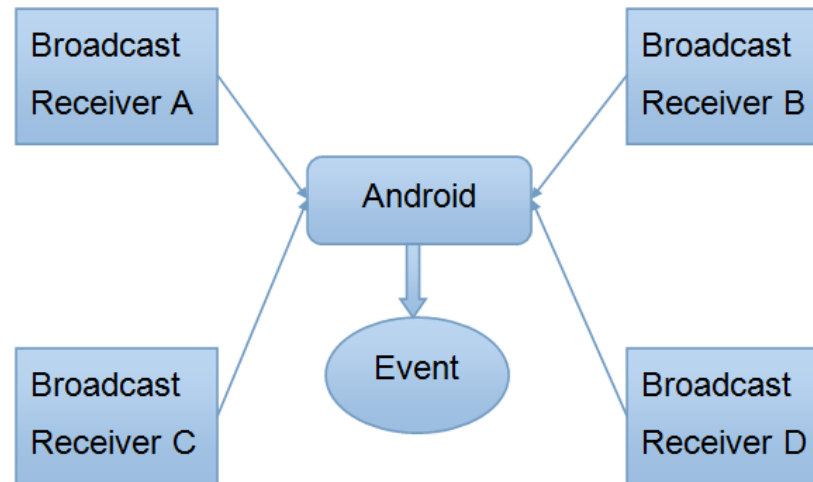
Each application has one main activity, the Interface is first displayed when the application is started



Android Application Component #2

BroadcastReceiver

A Broadcast can be set to have multiple receivers



Upon receiving a broadcast notification, the Receiver can initiate an Activity or other actions to notify the user

If an APP does not start it can still receive/broadcast

System Broadcasts

Defined in `frameworks\base\core\java\android\content\intent.java`,

String	<code>ACTION_BATTERY_CHANGED</code>	Broadcast Action: This is a <i>sticky broadcast</i> containing the charging state, level, and other information about the battery.
String	<code>ACTION_BATTERY_LOW</code>	Broadcast Action: Indicates low battery condition on the device.
String	<code>ACTION_BATTERY_OKAY</code>	Broadcast Action: Indicates the battery is now okay after being low.
String	<code>ACTION_BOOT_COMPLETED</code>	Broadcast Action: This is broadcast once, after the system has finished booting.
String	<code>ACTION_SCREEN_OFF</code>	Broadcast Action: Sent when the device goes to sleep and becomes non-interactive.
String	<code>ACTION_SCREEN_ON</code>	Broadcast Action: Sent when the device wakes up and becomes interactive.

Main Types of Broadcasts

Normal Broadcasts




Ordered Broadcasts




How to Set a Broadcast



Static registration: use the `AndroidManifest.xml <receiver>labels`

-  The other components of the program are not running, yet you still receive the Broadcast

Dynamic Registration: Calling register Receiver function

-  You can choose to receive a Broadcast dynamically based on runtime scenario

Static Registration

```
<receiver android:enabled=["true" | "false"]
  android:exported=["true" | "false"]
  android:icon="drawable resource"
  android:label="string resource"
  android:name="string"
  android:permission="string"
  android:process="string" >
  ...
</receiver>
```

Set the priority order of the Receiver in Ordered Broadcasts

```
<intent-filter android:icon="drawable resource"
  android:label="string resource"
  android:priority="integer" >
  ...
  <action android:name="string" />
</intent-filter>
```

Set up to receive Broadcasts



Dynamic Registration

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mBroadcastReceiver = new MyBroadcastReceiver();  
        IntentFilter intentFilter = new IntentFilter();  
        intentFilter.addAction(BROADCAST_ACTION);  
        registerReceiver(mBroadcastReceiver, intentFilter);  
    }  
}
```


Booting Directly to an Application

Principle

- ▶ When Android OS BOOT phase is completed it will send a broadcast named ACTION_BOOT_COMPLETED
- ▶ We can capture this broadcast in a BroadcastReceiver
- ▶ Then we can start our Activity

Step

Implement a BroadcastReceiver to capture ACTION_BOOT_COMPLETED when we want to start an Activity

```
public class BootCompletedReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals(Intent.ACTION_BOOT_COMPLETED))
        {
            Intent newIntent = new Intent(context, BootTestActivity.class);
            newIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            // Note, you must add this flag or boot will fail
            context.startActivity(newIntent);
        }
    }
}
```

Booting Directly to an Application

Step

Register the BroadcastReceiver in AndroidManifest.xml

```
<receiver android:name=".BootCompletedReceiver">
    <intent-filter>
        <action
            android:name="android.intent.action.BOOT_COMPLETED" />
        </intent-filter>
    </receiver>
```

In AndroidManifest.xml get permission to add this broadcast

```
<uses-permission
    android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
```



Booting Directly to an Application

Manual Start

To avoid malware attacks in Android 3.1 and above, applications which have never been started or have been manually forced to stop will not receive `BOOT_COMPLETED` broadcasts. So the App which you want to boot directly into must be manually started the first time.

Manual Unlock

The system screen lock feature must be removed. To do so, go to: [Settings> Security> LockScreen](#) and set to **None**.

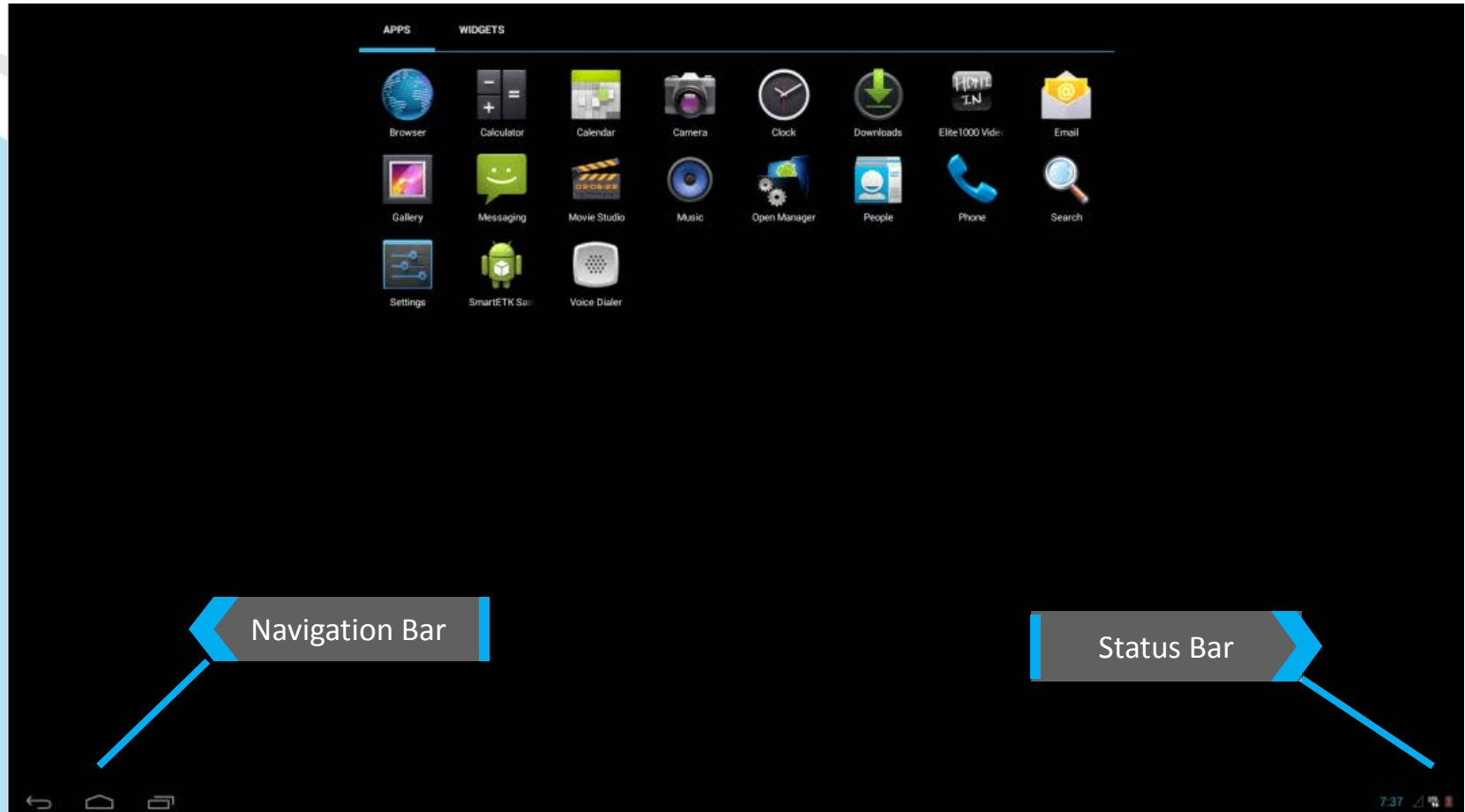
A vertical list of three features, each in a chevron-shaped box. The first two are grey, and the third is orange. A white vertical bar is on the left.

Start-up Screen Customization

Direct Boot to Application

Full Screen Display Mode

System Tray (System Bar)



System Bar



Low-Profile Mode



```
// This example uses decor view, but you can use any visible view.
View decorView = getActivity().getWindow().getDecorView();
int uiOptions = View.SYSTEM_UI_FLAG_LOW_PROFILE;
decorView.setSystemUiVisibility(uiOptions);
```



How to Hide the Status Bar



```
<application
```

```
...
```

```
android:theme="@android:style/Theme.Holo.NoActionBar.Fullscreen">
```

```
...
```

```
</application
```

```
if (Build.VERSION.SDK_INT < 16) {
```

```
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

```
}
```

Android 4.0-



```
View decorView = getWindow().getDecorView();
```

```
// Hide the status bar.
```

```
int uiOptions = View.SYSTEM_UI_FLAG_FULLSCREEN;
```

```
decorView.setSystemUiVisibility(uiOptions);
```

```
// Remember that you should never show the action bar if the  
// status bar is hidden, so hide that too if necessary.
```

```
ActionBar actionBar = getActionBar();
```

```
actionBar.hide();
```

Android 4.1+



How to Hide the Navigation Bar

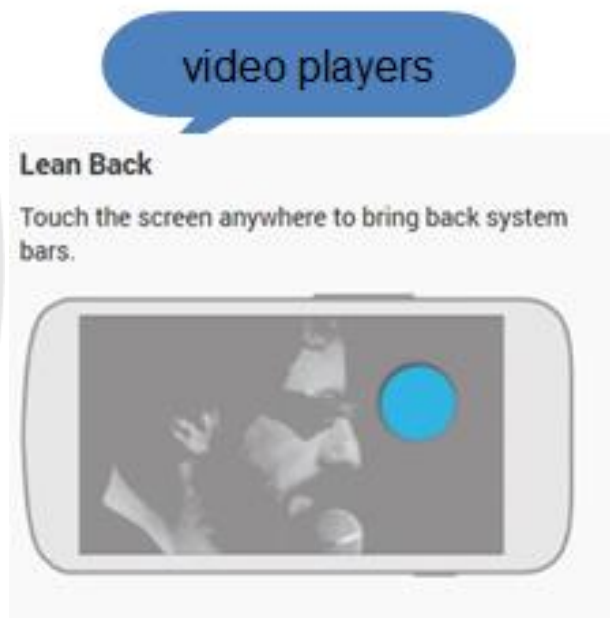


```
View decorView = getWindow().getDecorView();  
// Hide both the navigation bar and the status bar.  
// SYSTEM UI FLAG_FULLSCREEN is only available on Android 4.1 and higher, but as  
// a general rule, you should design your app to hide the status bar whenever you  
// hide the navigation bar.  
int uiOptions = View.SYSTEM_UI_FLAG_HIDE_NAVIGATION  
                | View.SYSTEM_UI_FLAG_FULLSCREEN;  
decorView.setSystemUiVisibility(uiOptions);
```

Lean Back and Immersive Modes



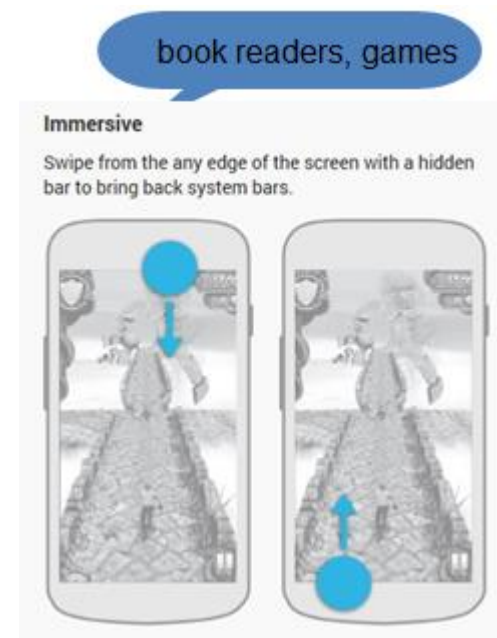
Lean Back



Android 4.3-



Immersive



Android 4.4+

Immersive Mode





How to Enter Immersive Mode

```
decorView.setSystemUiVisibility(  
    View.SYSTEM_UI_FLAG_LAYOUT_STABLE  
    | View.SYSTEM_UI_FLAG_FULLSCREEN  
    | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_FULLSCREEN  
    | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION  
    | View.SYSTEM_UI_FLAG_FULLSCREEN  
    | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY);}
```



How to Enable Immersive Sticky Effect

Android 4.3

羊和平發展基金會董事長連戰昨天率團抵達北京，將於十八日下午與中共

OSD

H.264, 1920x1080, 10M, 30FPS, CABAC, mp4 (1/1)

國臺辦主任張誌軍在南京首次正式會面時，邀請張誌軍訪臺。針對張主

OSD

00:00 01:22

2:59



How to Enable Immersive Sticky Effect

Android 4.3



OnSystemUiVisibilityChange

```
View decorView = getWindow().getDecorView();
decorView.setSystemUiVisibilityChangeListener
    (new View.OnSystemUiVisibilityChangeListener() {
        @Override
        public void onSystemUiVisibilityChange(int visibility) {
            // Note that system bars will only be "visible" if none of the
            // LOW_PROFILE, HIDE_NAVIGATION, or FULLSCREEN flags are set.
            if ((visibility & View.SYSTEM_UI_FLAG_FULLSCREEN) == 0) {
                // TODO: The system bars are visible. Make any desired
                // adjustments to your UI, such as showing the action bar or
                // other navigational controls.
            } else {
                // TODO: The system bars are NOT visible. Make and desired
                // adjustments to your UI, such as hiding the action bar or
                // other navigational controls.
            }
        }
    });
```



How to Enable Immersive Sticky Effect

Android 4.3



Start the 5 Second Timer

```
mViewTimer = new Timer();
mViewTimer.schedule(new TimerTask(){
    int t = 5;
    @Override
    public void run(){
        t--;
        if (t == 0){
            mHandler.post(new Runnable(){
                @Override
                public void run(){
                    getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_NAVIGATION
                                                                    | View.SYSTEM_UI_FLAG_FULLSCREEN);
                }
            });

            mViewTimer.cancel();
            mViewTimer.purge();
            mViewTimer = null;
        }
    }
}, 1000, 1000);
```

Summary

Boot Screen

Rapid customization from the U-boot to the Android start-up screen by providing complete process/tool kits for client customizations



Help customers develop different full-screen display mode effects for various versions of Android, by providing customers with an in-house developed sample

Full Screen Mode

Program Startup

Help customers to enable devices to boot directly into applications and to troubleshoot problems encountered after booting



© 2014 VIA Technologies, Inc All Rights Reserved.

- *VIA reserves the right to make changes in its products without notice in order to improve design or performance characteristics.*
- *This publication neither states nor implies any representations or warranties of any kind, including but not limited to any implied warranty of merchantability or fitness for a particular purpose. No license, express or implied, to any intellectual property rights is granted by this document.*
- *VIA makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication or the information contained herein, and reserves the right to make changes at any time, without notice. VIA disclaims responsibility for any consequences resulting from the use of the information included herein.*
- *VIA C7®, VIA C7®-D, VIA C7®-M, and VIA Eden™ are trademarks of VIA Technologies, Inc.*