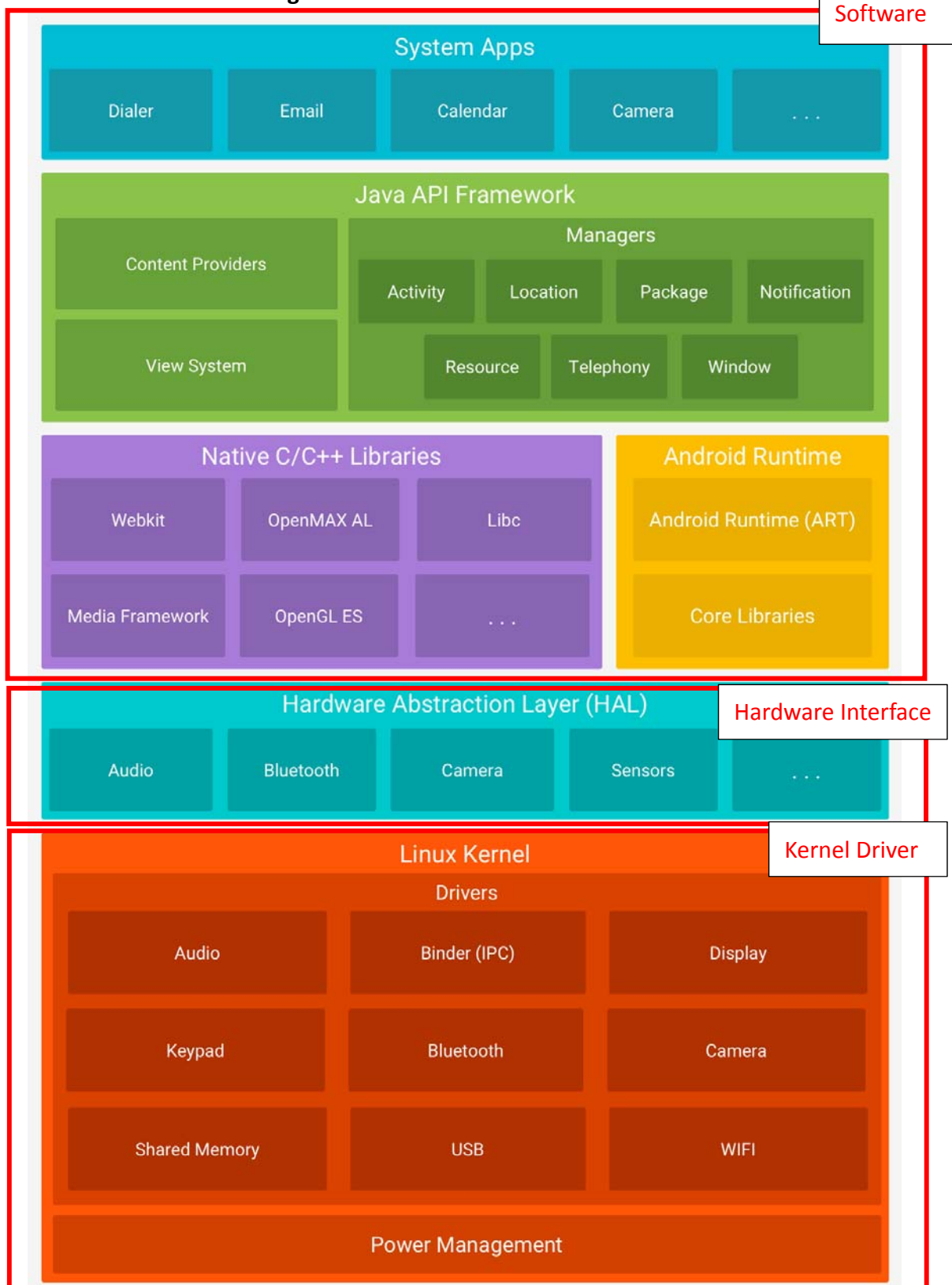


# 1. HMI Android

Android Architecture Diagram



## Development Notice

- (1) HMI is a complete product. Only the upper software can be programmed to execute on the HMI. And core and hardware underlying compilation is not provided.
- (2) The Android system itself already supports a large number of software libraries. So developers can develop the upper software and install it on the HMI as long as they don't involve the underlying hardware driver.
- (3) If the application involves controlling the underlying hardware, such as the UART, reboot, and other individual development features. The developer should call the SO file provided by DMATEK (provide the program interface to control the hardware)
- (4) If the developer wants to add hardware and sensors that are not provided. Unless the control method has been provided (such as uart, gpio, etc. already provided by the SDK), please contact us at the time of ordering.
- (5) GPIO is a fixed 8-channel output and 8-channel input. For other GPIO functional requirements, please contact us when ordering. The output voltage is 3.3V.
- (6) RS232 and RS485 functions are only potential conversion after UART output. Currently, HMI does not provide upper layer communication protocol such as Modbus.

## SDK Content

Item	SDK	Demo Function
4G	hmifunc.jar	<ol style="list-style-type: none"><li>1. Make a telephone call</li><li>2. Send a text message</li></ol>
Apk management	hmifunc.jar	<ol style="list-style-type: none"><li>1. APK installation</li><li>2. APK uninstall</li></ol>
Audio	hmifunc.jar	<ol style="list-style-type: none"><li>1. Recording</li><li>2. Play</li></ol>
Backlight	hmifunc.jar libbacklight.so	<ol style="list-style-type: none"><li>1. Backlight brightness adjustment</li></ol>
Bluetooth	hmifunc.jar	<ol style="list-style-type: none"><li>1. Initialize the Bluetooth function</li><li>2. SPP - Bluetooth Search</li><li>3. SPP - Bluetooth connection</li><li>4. BLE - Bluetooth Search</li><li>5. BLE - Bluetooth connection</li></ol>
Boot	hmifunc.jar	<ol style="list-style-type: none"><li>1. Set APK to boot</li><li>2. Remove APK boot</li></ol>

Buzzer	hmifunc.jar libbuzzer.so	<ol style="list-style-type: none"> <li>1. Turn on the buzzer</li> <li>2. Turn off the buzzer</li> </ol>
GPIO	hmifunc.jar libgprios.so	<ol style="list-style-type: none"> <li>1. Set a single output output high potential</li> <li>2. Set a single ouput output low potential</li> <li>3. Read the output potential state</li> </ol>
	gpio2.jar libgpiosa.so	<ol style="list-style-type: none"> <li>1. This feature needs to update imag</li> <li>2. Set 8-channel output output potential at the same time</li> <li>3. Read the potential state of 8-channel input at the same tim</li> </ol>
G-sensor	hmifunc.jar	<ol style="list-style-type: none"> <li>1. Get the G-sensor value</li> </ol>
System logo	hmifunc.jar	<ol style="list-style-type: none"> <li>1. Modify the system logo</li> <li>2. System logo creation</li> </ol>
TF storage	hmifunc.jar	<ol style="list-style-type: none"> <li>1. Add a new file to the TF card</li> <li>2. Write the contents of the file in the TF card</li> <li>3. Read the contents of the file in the TF card</li> </ol>
UART	hmifunc.jar libuart.so	<ol style="list-style-type: none"> <li>1. Open the serial port</li> <li>2. Receive hex message</li> <li>3. Receive string messages</li> <li>4. Send hex message</li> <li>5. Send a string message</li> </ol>
UsbCamera	hmifunc.jar libUsbCamJni.so	<ol style="list-style-type: none"> <li>1. Open Usb Camera to take the picture</li> <li>2. Close Usb Camera</li> </ol>
Video	hmifunc.jar	<ol style="list-style-type: none"> <li>1. Get a list of videos</li> <li>2. play video</li> <li>3. Pause a video</li> <li>4. Stop playing the movie</li> </ol>
WIFI	hmifunc.jar	<ol style="list-style-type: none"> <li>1. Open WIFI</li> <li>2. Search WIFI</li> <li>3. Get WIFI strength, SSID and other information</li> <li>4. Connect to WIFI</li> </ol>