

HarmonyOS Hi3516DV300设备开发实践

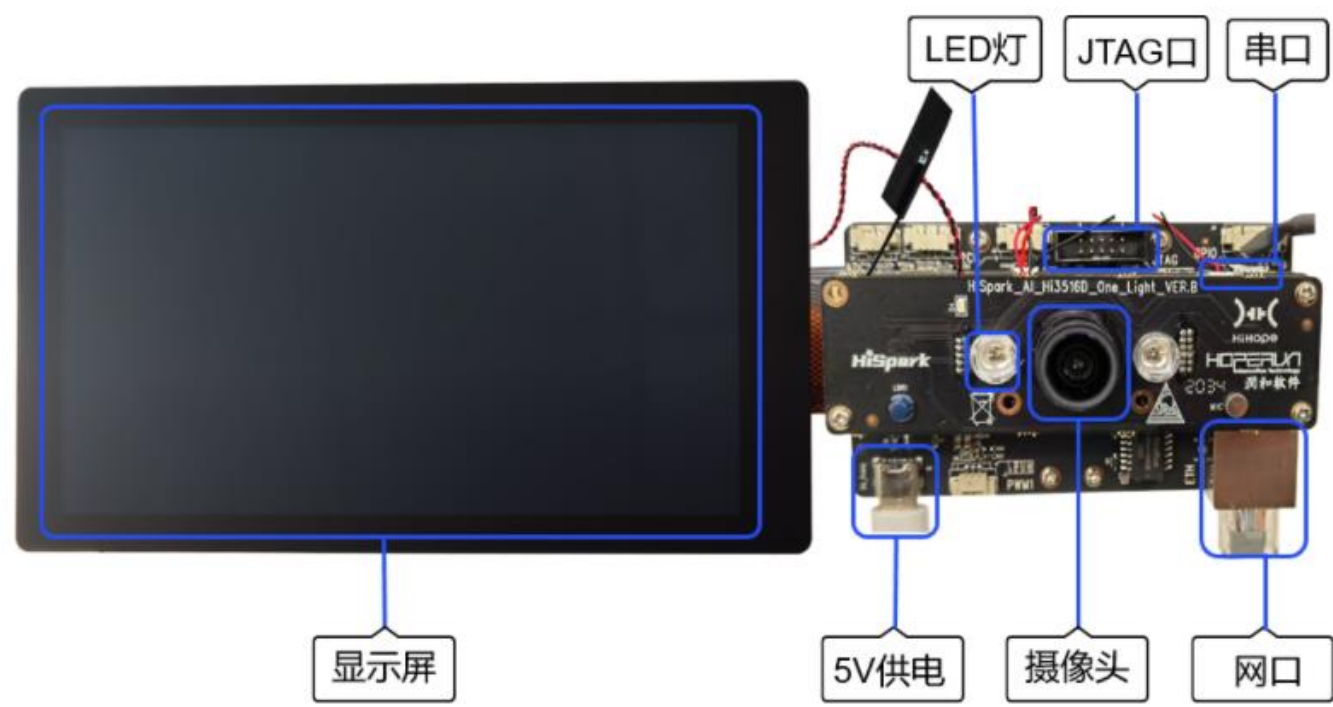


目录

1. **IoT Camera开发板（ Hi3516DV300 ）介绍**
2. 开发环境准备
3. 轻量ACE UI开发框架简介
4. HarmonyOS开发实战

Hi3516开发版本介绍

- Hi3516DV300作为新一代行业专用Smart HD IP摄像机SOC，集成新一代ISP、H.265视频压缩编码器，同时集成高性能NNIE引擎，使得Hi3516DV300在低码率、高画质、智能处理和分析、低功耗等方面引领行业水平。



规格类型	规格清单
处理器及内部存储	<ul style="list-style-type: none">Hi3516DV300芯片DDR3 1GBeMMC4.5, 8GB容量
外部器件	<ul style="list-style-type: none">以太网口音频视频<ul style="list-style-type: none">1路语音输入1路单声道(AC_L)输出, 接3W功放(LM4871)MicroHDMI (1路HDMI 1.4)摄像头<ul style="list-style-type: none">传感器IMX335镜头M12, 焦距4mm, 光圈1.8显示屏<ul style="list-style-type: none">LCD连接器 (2.35寸)LCD连接器 (5.5寸)外部器件及接口<ul style="list-style-type: none">SD卡接口JTAG/I2S 接口ADC接口舵机接口Grove连接器USB2.0(Type C)功能按键3个, 2个用户自定义按键, 1个升级按键LED指示灯, 绿灯, 红灯

目录

1. IoT Camera开发板（ Hi3516DV300 ）介绍
- 2. 开发环境准备**
3. 轻量ACE UI开发框架简介
4. HarmonyOS开发实战

DevEco Studio IDE工具下载

- HarmonyOS开发者网站下载地址：

- <https://developer.harmonyos.com/cn/develop/deveco-studio#download>

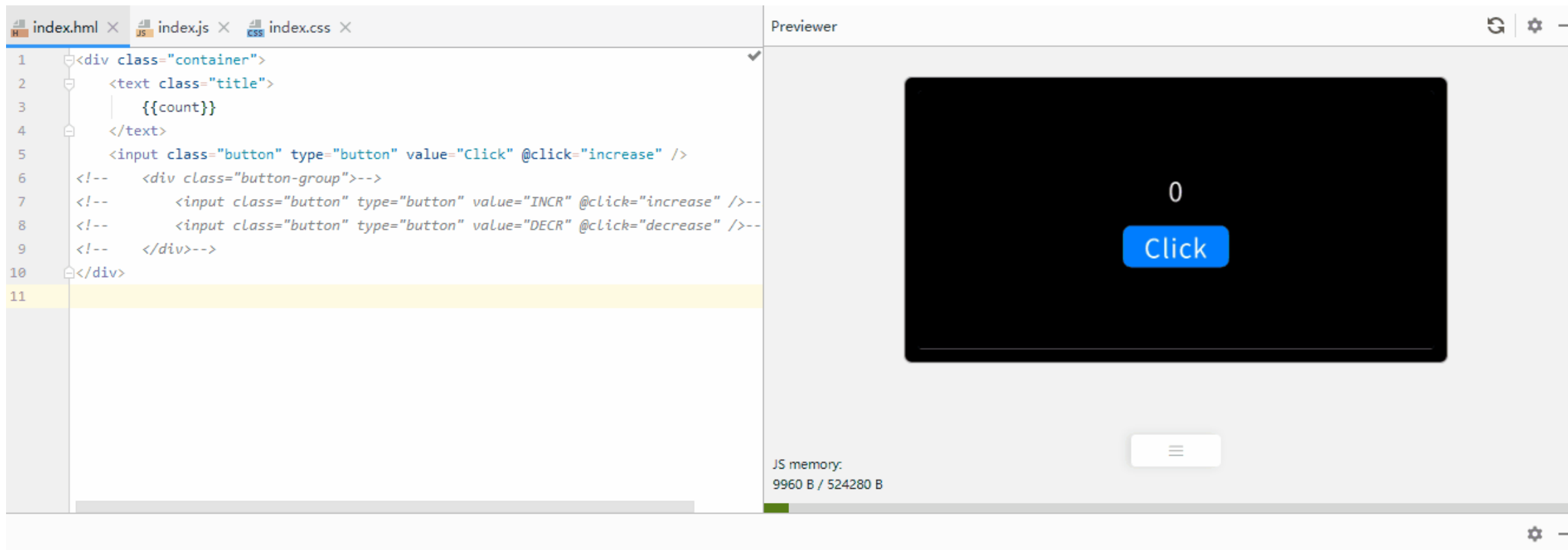
- 安装：一路默认安装即可



应用启动后的首界面

IDE工具调试开挂器：Preview预览功能

- 代码修改保存后即可在右边看到效果



IDE工具单步断点调试

The image shows a development environment with two main panels: a code editor on the left and a simulator on the right.

Code Editor (index.js):

```
1 export default {
2   data: {
3     tasks: [
4       {
5         title: '早睡早起: 7点半起床',
6         done: true
7       }, {
8         title: '健康生活: 慢跑5公里',
9         done: false
10      }, {
11        title: '健康生活: 每日10000步',
12        done: false,
13      }, {
14        title: '能力提升: 完成2道LeetCode题目',
15        done: false
16      }, {
17        title: '早睡早起: 23点半睡觉',
18        done: false
19      }
20    ]
21  },
22  toggle(idx) {
23    this.tasks[idx].done = !this.tasks[idx].done;
24  },
25  remove(idx) {
26    this.tasks.splice(idx, 1);
27  }
28 }
```

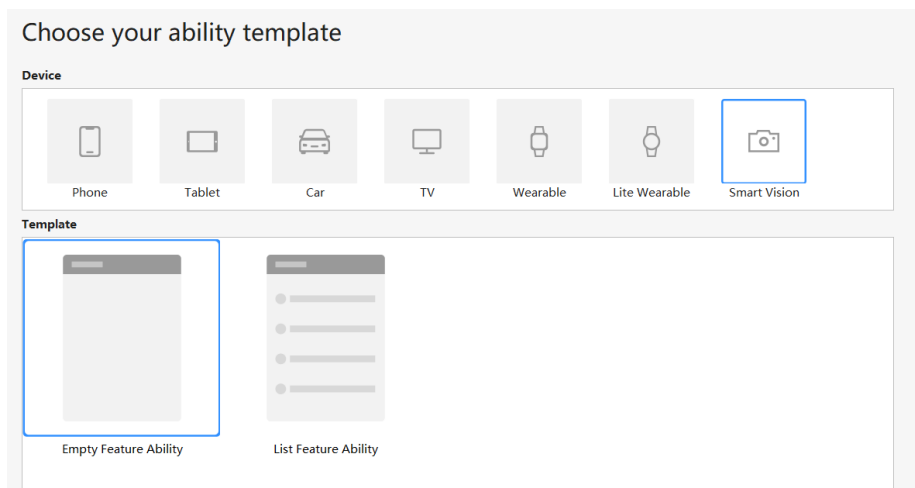
Debugger: A breakpoint is set at line 21, which is highlighted in orange. The execution flow is visible on the left margin.

Simulator: The right panel shows a simulated mobile application interface. It has a title "今日" (Today) and a list of tasks:

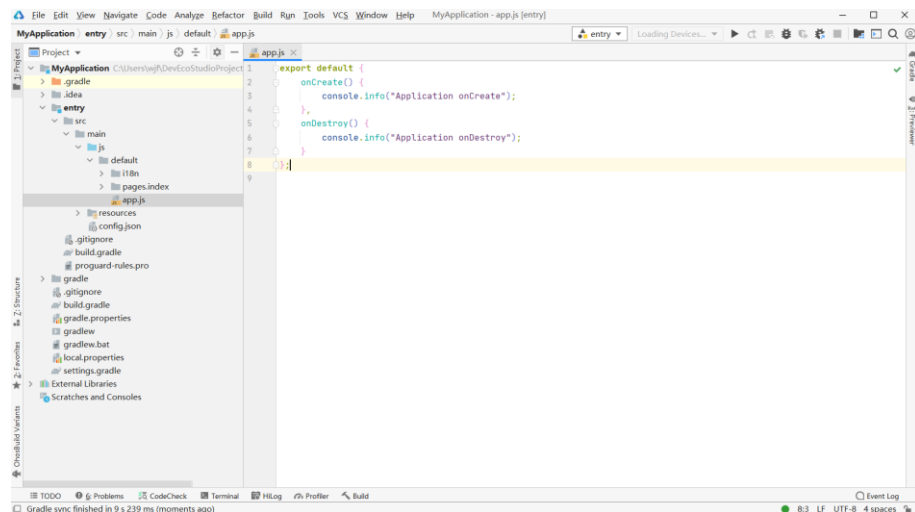
- [已完成]早睡早起: 7点半起床 [删除]
- [未完成]健康生活: 慢跑5公里 [删除]
- [未完成]健康生活: 每日10000步 [删除]
- [未完成]能力提升: 完成2道LeetCode题目 [删除]

At the bottom of the simulator, there is a "JS memory:" indicator showing "21336 B / 524280 B".

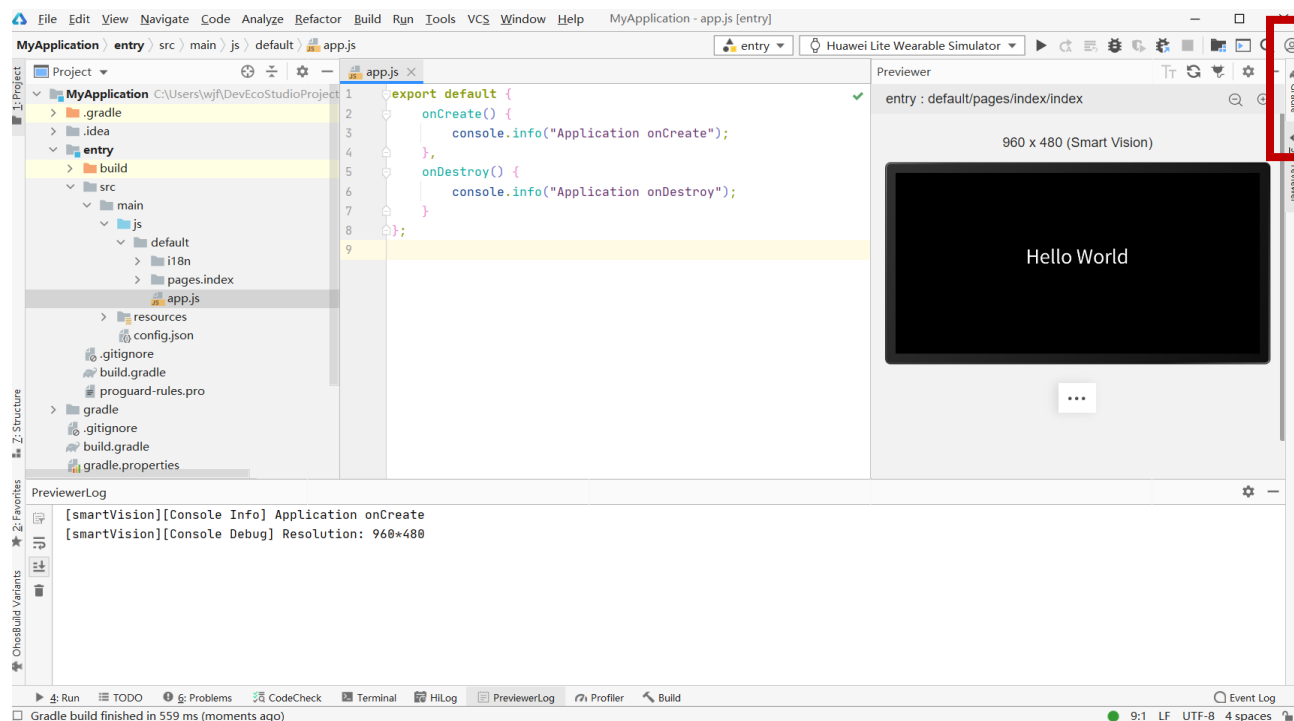
第一个JS应用：Hello World



1、创建工程，选择SmartVision设备类别和对应应用模板



2、进入该页面后，工具右下脚会弹出安装配套工具的提示，点击“Install”



3、预览器看效果

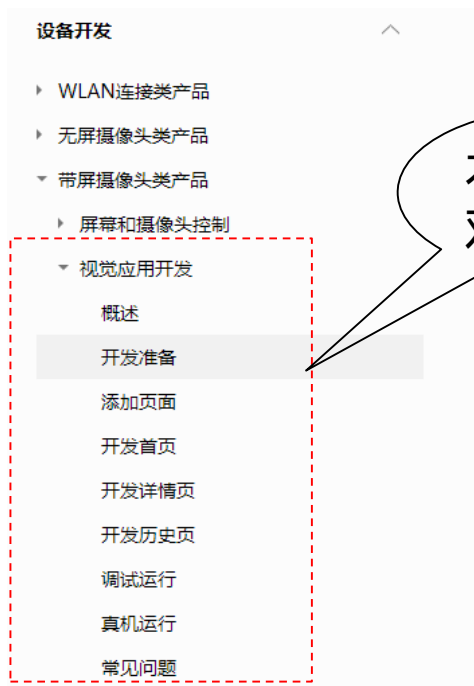
目录

1. IoT Camera开发板（ Hi3516DV300 ）介绍
2. 开发环境准备
- 3. 轻量ACE UI开发框架简介**
4. HarmonyOS 开发实战

HarmonyOS开发者网站

设备开发文档入口：

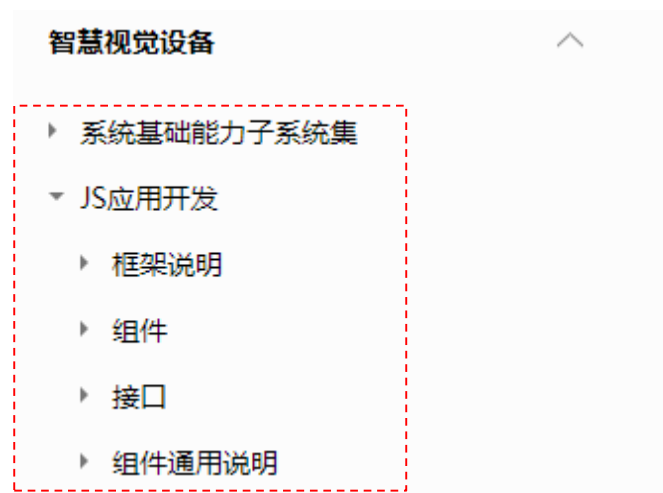
https://device.harmonyos.com/cn/docs/develop/demo/oem_wifi_sdk_des-0000001050059068



本次仅针对该品类

API参考文档入口：

· <https://device.harmonyos.com/cn/docs/develop/apiref/abilitykit-0000001054598111>



外部教程（51CTO）：<https://edu.51cto.com/course/25054.html>

面向轻量智能硬件设备的ACE应用程序框架



轻量化框架，可延伸到百K级设备



- 核心框架下沉
- 轻量UI & 组件直通

HarmonyOS应用 - 先从Hello World开始

index.html

```
<div class="container">
  <text class="title">
    Hello {{title}}
  </text>
</div>
```

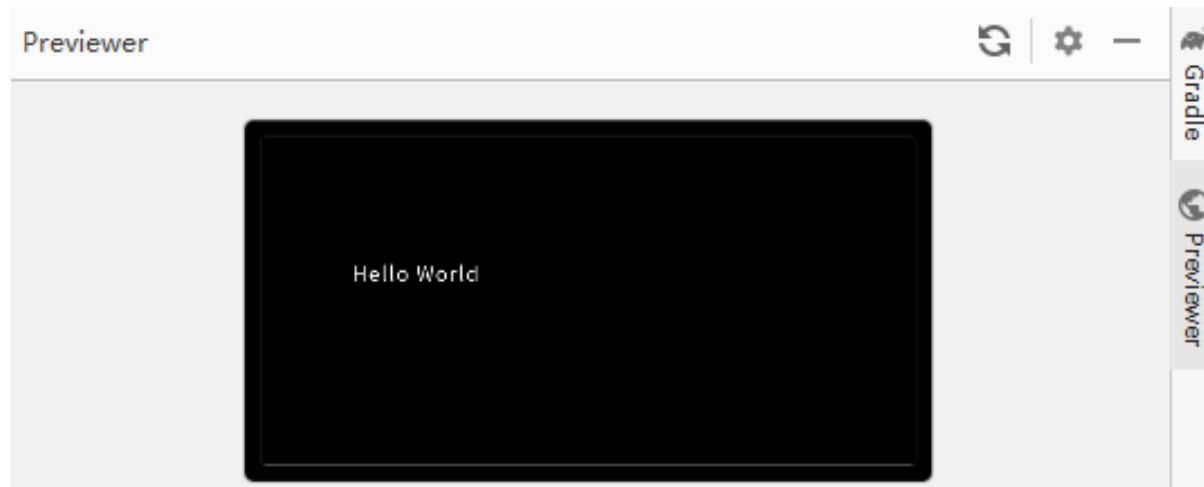
index.css

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
  left: 0px;
  top: 0px;
  width: 454px;
  height: 454px;
}

.title {
  font-size: 30px;
  text-align: center;
  width: 200px;
  height: 100px;
}
```

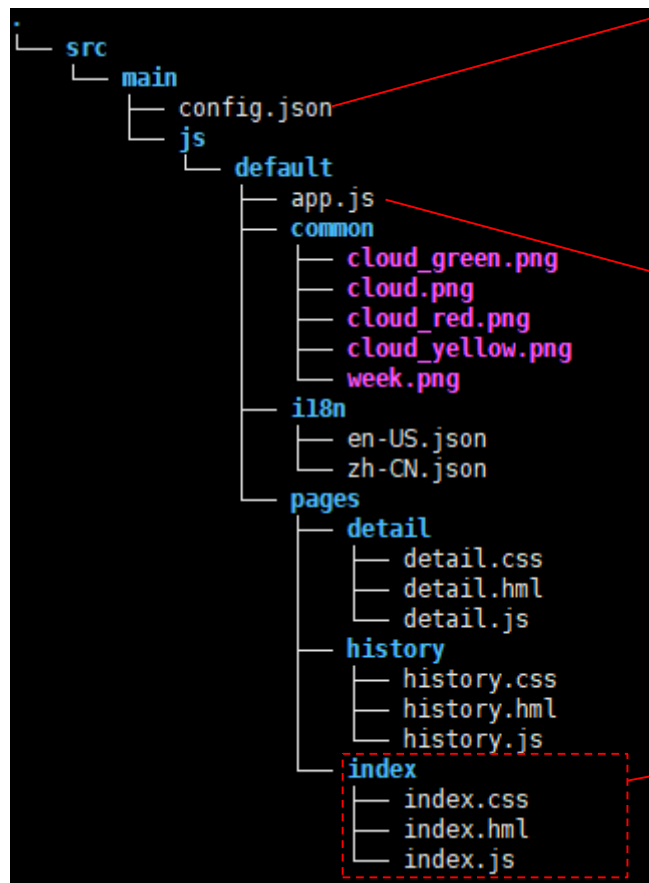
index.js

```
export default {
  data: {
    title: 'World'
  }
}
```



HarmonyOS应用 - 工程目录

- JS应用配置文件



JS应用入口

首页

```
1. <!--config.json-->
2. {
3.   ...
4.   "pages": [
5.     "pages/index/index", //首页, 即entry入口
6.     "pages/detail/detail"
7.   ]
8.   ...
9. }
```

```
1. // app.js
2. export default {
3.   onCreate() {
4.     console.info('Application onCreate');
5.   },
6.   onDestroy() {
7.     console.info('Application onDestroy');
8.   },
9. }
```

目录结构中文件分类如下:

- .html结尾的HML模板文件, 这个文件用来描述当前页面的文件布局结构。
- .css结尾的CSS样式文件, 这个文件用于描述页面样式。
- .js结尾的JS文件, 这个文件用于处理页面和用户的交互。

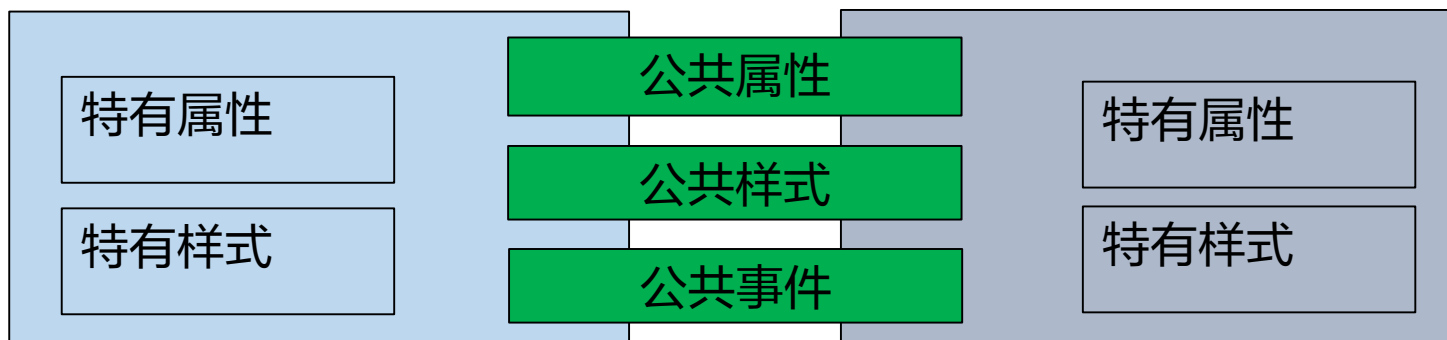
HarmonyOS应用 - ACE应用语法

- 1、JS语法规则全量支持ECMAScript5.1
- 2、声明式UI写法（数据绑定）

```
1. <!-- index.html -->
2. <div class="container">
3.   <image-animator class="image-player" ref="animator" images="{{images}}" duration="1s" onclick="handleClick"></image-animator>
4. </div>
```

```
1. // index.js
2. export default {
3.   data: {
4.     images: [
5.       { src: '/common/frame1.png' },
6.       { src: '/common/frame2.png' },
7.       { src: '/common/frame3.png' },
8.     ],
9.   },
10.  handleClick() {
11.    const animator = this.$refs.animator; //获取ref属性为animator 的DOM元素
12.    const state = animator.getState();
13.    if (state === 'paused') {
14.      animator.resume();
15.    } else if (state === 'stopped') {
16.      animator.start();
17.    } else {
18.      animator.pause();
19.    }
20.  },
21. };;
```

HarmonyOS应用 - UI组件：基础组件 + 容器组件



基础组件

- Image
- Image-animator
- Progress
- Text
- Marquee
- Chart
- Input
- Slider
- Switch
- Picker-view

容器组件

- Div
- List
- List-item
- Stack
- Swiper

HarmonyOS应用 - UI组件：通用事件

名称	参数	描述
click	-	点击动作触发该事件。
longpress	-	长按动作触发该事件。
swipe	SwipeEvent	组件上快速滑动后触发。

表1 SwipeEvent 基础事件对象属性列表（继承BaseEvent）

属性	类型	说明
direction	string	滑动方向，可能值有： 1. left: 向左滑动； 2. right: 向右滑动； 3. up: 向上滑动； 4. down: 向下滑动。

Example:

```
<!-- xxx.html -->
<div>
  <!-- 正常格式 -->
  <div onclick="clickfunc"></div>
  <!-- 缩写 -->
  <div @click="clickfunc"></div>
</div>
```

```
// xxx.js
export default {
  clickfunc: function() {
    this.obj = 'Hello World';
  },
}
```

也可以

```
clickfunc() {
  this.obj = 'hello world';
}
```



HarmonyOS应用 - UI组件：通用属性

- 常规属性：

名称	类型	默认值	必填	描述
id	string	-	否	组件的唯一标识。
style	string	-	否	组件的样式声明。
class	string	-	否	组件的样式类，用于引用样式表。
ref	string	-	否	用来指定指向子元素的引用信息，该引用将注册到父组件的\$refs 属性对象上。

- 渲染属性：

名称	类型	默认值	描述	
for	Array	-	根据设置的数据列表，展开当前元素。	
if	boolean	-	根据设置的boolean值，添加或移除当前元素。	条件渲染
show	boolean	-	根据设置的boolean值，显示或隐藏当前元素。	

HarmonyOS应用 - UI组件：通用样式

名称	类型	默认值	必填	描述	可以在style属性或css文件中设置组件外观样式
width	<length>	-	否	设置组件自身的宽度。 未设置时组件宽度默认为0。	
height	<length>	-	否	设置组件自身的高度。 未设置时组件高度默认为0。	
padding	<length>	0	否	使用简写属性设置所有的内边距属性。	
margin	<length>	0	否	使用简写属性设置所有的外边距属性，该属性可以有1到4个值。 •只有一个值时，这个值会被指定给全部四个边。 •两个值时，第一个值被匹配给上和下，第二个值被匹配给左和右。 •三个值时，第一个值被匹配给上，第二个值被匹配给左和右，第三个值被匹配给下。 •四个值时，会依次按上、右、下、左的顺序匹配（即顺时针顺序）。	
margin-[left top right bottom]	<length>	0	否	设置左、上、右、下外边距属性。	
border-width	<length>	0	否	使用简写属性设置元素的所有边框宽度。	
border-color	<color>	black	否	使用简写属性设置元素的所有边框颜色。	
border-radius	<length>	-	否	border-radius属性是设置元素的外边框圆角半径。	
background-color	<color>	-	否	设置背景颜色。	
display	string	flex	否	确定一个元素所产生的框的类型，可选值为： •flex：弹性布局。 •none：不渲染此元素。	
[left top]	<length>	-	否	left top需要配合position样式使用，来确定元素的偏移位置。 •left属性规定元素的左边缘。该属性定义了定位元素左外边距边界与其包含块左边界之间的偏移。 •top属性规定元素的顶部边缘。该属性定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。	

Padding：元素与内容的间隔

Margin：元素与元素的间隔

tips：不设置为0，会不显示

HarmonyOS应用 - UI基础组件： progress

属性				
名称	类型	默认值	必填	描述
type	string	horizontal	否	设置进度条的类型，该属性不支持动态修改，可选值为： <ul style="list-style-type: none">horizontal：线性进度条；arc：弧形进度条。 type=arc

样式				
type=horizontal				
名称	类型	默认值	必填	描述
color	<color>	#6b9ac7	否	设置进度条的颜色。
stroke-width	<length>	32px	否	设置进度条的宽度。

名称	类型	默认值	必填	描述
color	<color>	-	否	弧形进度条的颜色。
background-color	<color>	-	否	弧形进度条的背景色。
stroke-width	<length>	-	否	弧形进度条的宽度。 <div>说明 进度条宽度越大，进度条越靠近圆心。即进度条始终在半径区域内。</div>
start-angle	<deg>	240	否	弧形进度条起始角度，以时钟0点为基线。范围为0到360（顺时针）。
total-angle	<deg>	240	否	弧形进度条总长度，范围为-360到360，负数标识起点到终点为逆时针。
center-x	<length>	-	否	弧形进度条中心位置，（坐标原点为组件左上角顶点）。该样式需要和center-y和radius一起。
center-y	<length>	-	否	弧形进度条中心位置，（坐标原点为组件左上角顶点）。该样式需要和center-x和radius一起。
radius	<length>	-	否	弧形进度条半径，该样式需要和center-x和center-y一起。

进度条，用于显示内容加载或操作处理进度



HarmonyOS应用 - 基础组件：input

属性

名称	类型	默认值	必填	描述
type	string	button	否	input组件类型，可选值为button、checkbox、radio。 button、checkbox、radio不支持动态修改。可选值定义如下： <ul style="list-style-type: none">button：定义可点击的按钮；checkbox：定义多选框；radio：定义单选按钮，允许在多个拥有相同name值的选项选中其中一个；
checked	boolean	false	否	当前组件是否选中。
name	string	-	否	input组件的名称。
value	string	-	否	input组件的value值，当类型为radio时必须填且相同name值的选项该值唯一。

功能：
输入框，包括单选框，多选框，按钮

事件

- 当input类型为checkbox、radio时，支持如下事件：

名称	参数	描述
change	{ checked:true false}	checkbox多选框或radio单选框的checked状态发生变化时触发该事件。

样式

名称	类型	默认值	必填	描述
color	<color>	#ffffff	否	单行输入框或者按钮的文本颜色。
font-size	<length>	30px	否	单行输入框或者按钮的文本尺寸。 目前仅支持30px和38px 两个字体大小。



HarmonyOS应用 - UI基础组件：slider

属性

名称	类型	默认值	必填	描述
min	number	0	否	滑动选择器的最小值。
max	number	100	否	滑动选择器的最大值。
value	number	0	否	滑动选择器的初始值。

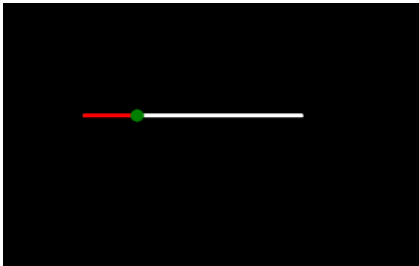
滑动条组件，用来快速调节设置值，如音量、亮度

事件

名称	参数	描述
change	{ progress: progressValue }	选择值发生变化时触发该事件。

样式

名称	类型	默认值	必填	描述
color	<color>	#000000	否	滑动条的背景颜色。
selected-color	<color>	#ffffff	否	滑动条的已选择颜色。
block-color	<color>	#6b9ac7	否	滑动条的滑块颜色。



HarmonyOS应用 - UI基础组件： switch

属性

名称	类型	默认值	必填	描述
checked	boolean	false	否	是否选中。

事件

名称	参数	描述
change	{ checked: checkedValue }	选中状态改变时触发该事件。

开关选择器，通过开关，开启或关闭某个功能

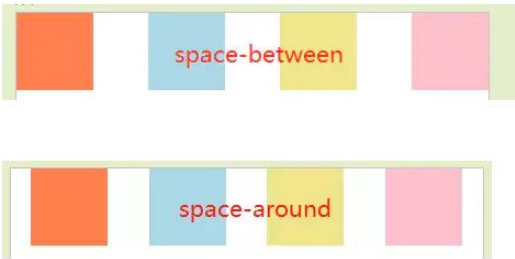


HarmonyOS应用 - UI容器组件：div

样式

名称	类型	默认值	必填	描述
flex-direction	string	row	否	flex容器主轴方向。可选项有： <ul style="list-style-type: none">column：垂直方向从上到下row：水平方向从左到右
flex-wrap	string	nowrap	否	flex容器是单行还是多行显示，该值暂不支持动态修改。可选项有： <ul style="list-style-type: none">nowrap：不换行，单行显示。wrap：换行，多行显示。
justify-content	string	flex-start	否	flex容器当前行的主轴对齐格式。可选项有： <ul style="list-style-type: none">flex-start：项目位于容器的开头。flex-end：项目位于容器的结尾。center：项目位于容器的中心。space-between：项目位于各行之间留有空白的容器内。space-around：项目位于各行之前、之间、之后都留有空白的容器内。
align-items	string	flex-start	否	flex容器当前行的交叉轴对齐格式，可选值为： <ul style="list-style-type: none">flex-start：元素向交叉轴起点对齐。flex-end：元素向交叉轴终点对齐。center：元素在交叉轴居中。

基础容器，用作页面结构的根节点或将内容进行分组



子组件：all

HarmonyOS应用 - 容UI器组件: stack

- 堆叠容器，子组件按照顺序依次入栈，后一个子组件覆盖前一个子组件
- 在布局上，非常有用的布局控件
- 子组件: all

HarmonyOS应用 - UI容器组件：swiper

属性

名称	类型	默认值	必填	描述
index	number	0	否	当前在容器中显示的子组件的索引值。
loop	boolean	true	否	是否开启循环轮播。
duration	number	-	否	子组件切换的动画时长。
vertical	boolean	false	否	是否为纵向滑动，纵向滑动时采用纵向的指示器。不支持动态修改。

事件

名称	参数	描述
change	{ index: currentIndex }	当前显示的组件索引变化时触发该事件。

滑动容器，提供切换子组件显示的能力



tips: 不能直接嵌套list标签

子组件：exclude lis

```
<div class="container">
  <swiper index="0" loop="true" duration="10" class="container">
    <div class="wrapper">
      <list class="todo-wrapper">
        <list-item for="{{todolist}}" class="todo-item">
          <text class="todo-title">{{$item.title}}
        </text>
          <text class="todo-title">{{$item.date}}
        </text>
        </list-item>
      </list>
    </div>
    <picker-view type="time" class="picker" selected="11:11"></picker-view>
  </swiper>
</div>
```


HarmonyOS应用 - 系统API

- 应用生命周期
- 应用上下文
- 页面生命周期
- 日志打印
- 页面路由

HarmonyOS应用 - 应用生命周期

属性	类型	参数	返回值	描述	触发时机
onCreate	Function	无	无	应用创建	当应用创建时调用。
onDestroy	Function	无	无	应用退出	当应用退出时触发。

在app.js中调用

```
// app.js
export default {
  onCreate() {
    console.info("Application onCreate");
  },
  onDestroy() {
    console.info("Application onDestroy");
  }
};
```

HarmonyOS应用 - 应用上下文

app.getInfo()

获取当前应用配置文件中声明的信息。

- 参数
无
- 返回值

参数名	类型	说明
appName	string	表示应用的名称。
versionName	string	表示应用的版本名称。
versionCode	number	表示应用的版本号。

- 示例

```
1. var info = app.getInfo();  
2. console.log(JSON.stringify(info));
```

app.terminate()

退出当前Ability

- 参数
无
- 示例

```
1. app.terminate();
```

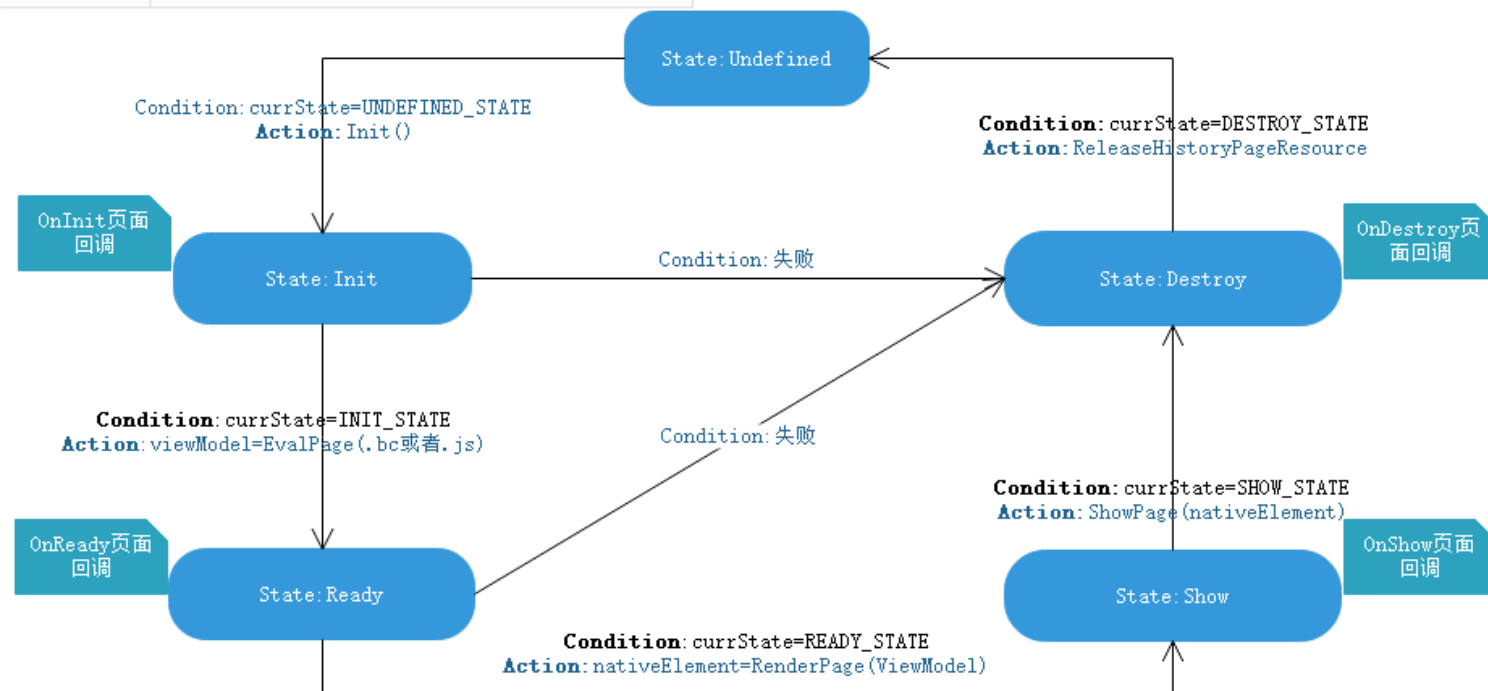
文件开头需要导入：

```
import app from '@system.app';
```

HarmonyOS应用 - 页面生命周期

页面生命周期

属性	类型	参数	返回值	描述	触发时机
onInit	Function	无	无	页面初始化	页面数据初始化完成时触发，只触发一次。
onReady	Function	无	无	页面创建完成	页面创建完成时触发，只触发一次。
onShow	Function	无	无	页面显示	页面显示时触发。
onDestroy	Function	无	无	页面销毁	页面销毁时触发。



HarmonyOS应用 - 日志打印

日志分类

打印一段文本信息，`console.debug|log|info|warn|error(message)`

- 参数

参数名	类型	必填	说明
message	string	是	表示要打印的文本信息。

- 示例

```
1. var versionCode = 1.1;  
2. console.info('Hello World. The current version code is ' + versionCode);
```

i 说明

`console.log()`打印的是debug级别日志信息。

HarmonyOS应用 - 页面路由

router.replace(OBJECT)

用应用内的某个页面替换当前页面，并销毁被替换的页面。

- 参数

参数名	类型	必填	说明
uri	string	是	目标页面的uri，可以是一下的两种格式： <ul style="list-style-type: none">▪ 页面绝对路径，由配置文件中pages列表提供，例如：<ul style="list-style-type: none">◦ pages/index/index◦ pages/detail/detail▪ 特殊值，如果uri的值是"/"，则跳转到首页。
params	Object	否	跳转时要同时传递到目标页面的数据，跳转到目标页面后，参数可以在页面中直接使用，如this.data1(data1为跳转时params参数中的key值)。

- 示例

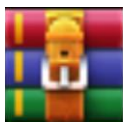
```
1. // 在当前页面中
2. router.replace({
3.   uri: 'pages/detail/detail',
4.   params: {
5.     data1: 'message',
6.   },
7. });
8. // 在detail页面中
9. console.info('showData1:' + this.data1);
```

需要导入：

import app from '@system.router';

ShowCase，帮助开发者入门

初学者，可参考我们的showcase代码



showcase.rar

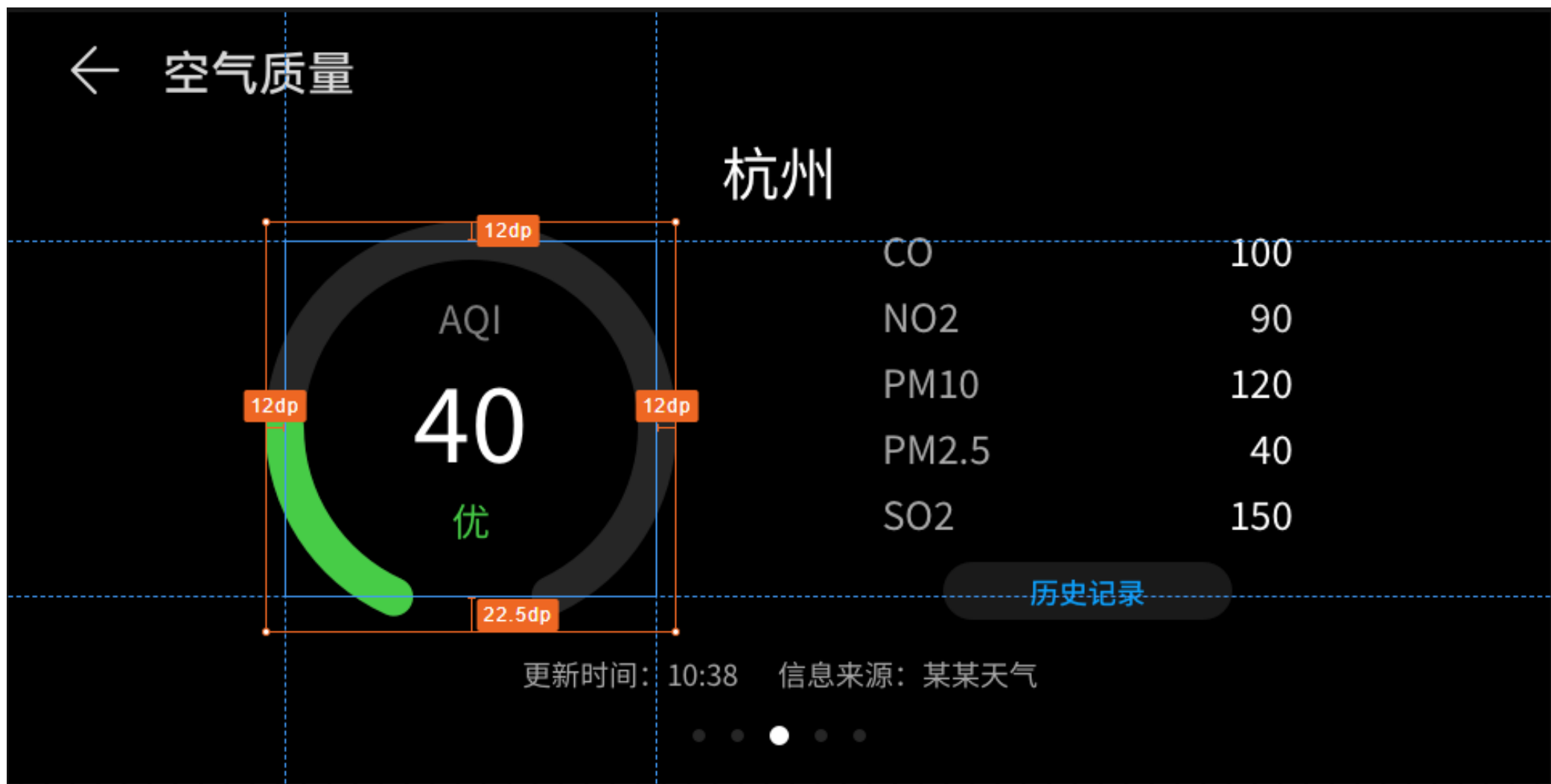
目录

1. IoT Camera开发板（ Hi3516DV300 ）介绍
2. 开发环境准备
3. 轻量ACE UI开发框架简介
- 4. HarmonyOS开发实战**

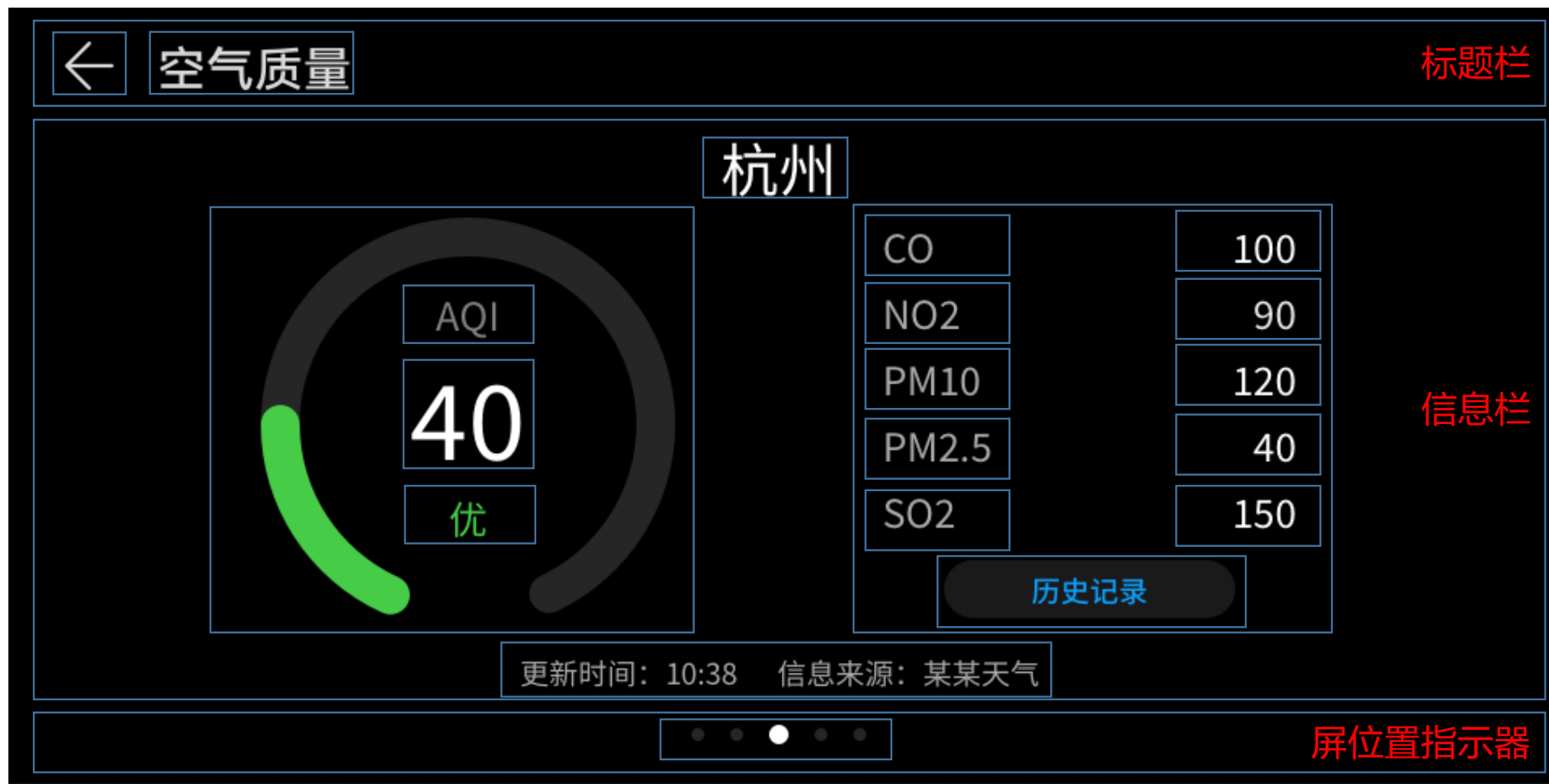
主要功能需求

- 总共有两屏，每屏通过左右滑动进入
- 每屏显示一个城市的空气质量数据：主要包括城市名称、AQI指数、污染物指数、更新时间和信息来源等数据，且数据可动态更新
- 应用可主动退出

UX设计师设计界面



开始UI框架设计



写好框架



标题栏开发和调试

index.css

```
/* 标题栏样式 */
.title {
  width: 960px;
  height: 72px;
  font-size: 50px;
  text-align: center;
}

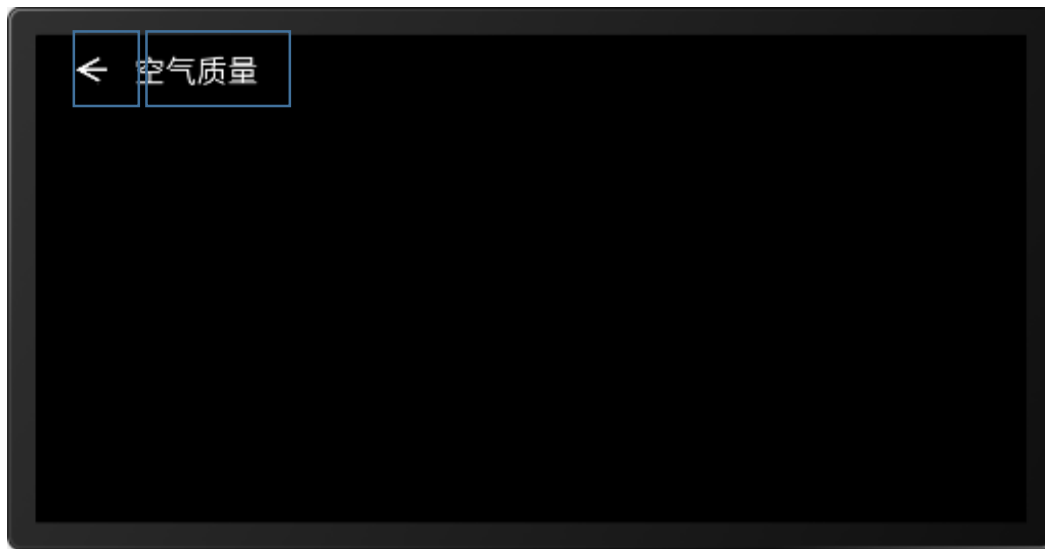
.titleImage {
  width: 30px;
  height: 26px;
  margin-top: 21px;
  margin-left: 39px;
}

.titleText {
  margin-top: 15.8px;
}

.titleText {
  margin-left: 27px;
}
```

index.html

```
<div id="title" class="title">
  <image class="titleImage" src="/common/images/back.png" onclick=""></image>
  <text class="titleText">空气质量</text>
</div>
```



预览器效果

用到的UI组件:

Div

Image

Text

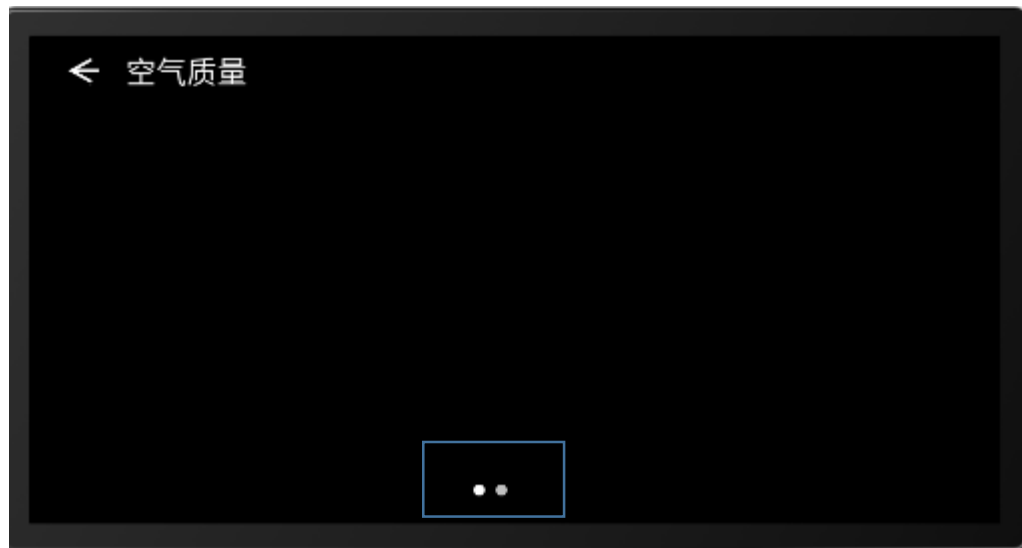
屏位置显示器开发和调试

index.html

```
<div id="indicator" class="indicator">
  <div style="background-color: white;width:12px;height:12px;border-radius:6px;"></div>
  <div style="background-color: silver;width:12px;height:12px;border-radius:6px;margin-left: 10px;"></div>
</div>
```

index.css

```
/* 屏位置指示器样式 */
.indicator {
  width: 960px;
  height: 30px;
  margin-left: 440px;
}
```



用到的UI组件:

Div

预览器效果

内容区AQI开发和调试

index.html

```
<swiper id="content" class="content">
  <stack id="swiper_page1" class="swiper_page1">
    <text id="city" class="city">杭州</text>
    <progress id="circleBar" class="circleBar" type="arc" percent="20"></progress>
    <text class="aqi">AQI</text>
    <text class="aqiValue">40</text>
    <text class="aqiLevel">优</text>
    <div id="detailInfo">
```

index.css

```
.swiper_page1 {
  width: 960px;
  height: 370px;
}

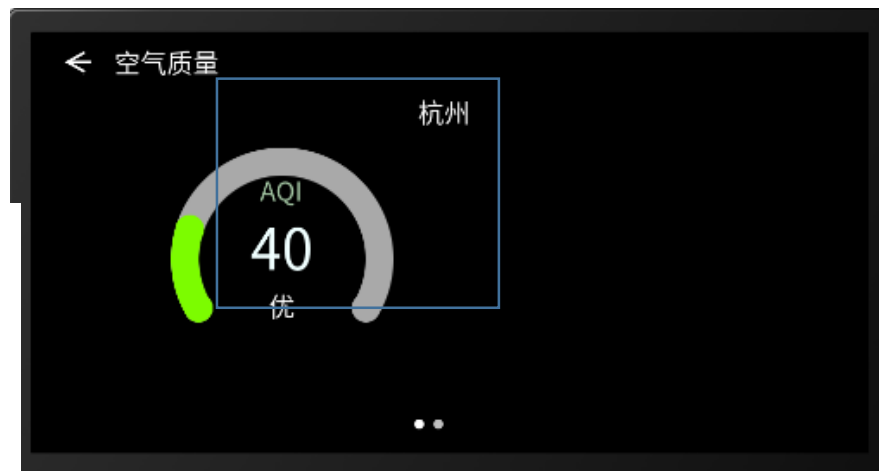
.city {
  width: 72px;
  height: 52px;
  left: 444px;
}
```

```
.circleBar {
  width: 256px;
  height: 256px;
  background-color: darkgrey;
  color: lawngreen;
  left: 160px;
  top: 58px;
}
```

```
.aqi {
  text-align: center;
  color: #a2c4a2;
  width: 156px;
  height: 45px;
  top: 90px;
  left: 210px;
}
```

```
.aqiValue {
  text-align: center;
  font-size: 65px;
  color: #f0ffff;
  width: 156px;
  height: 92px;
  top: 134px;
  left: 210px;
}
```

```
.aqiLevel {
  text-align: center;
  width: 156px;
  height: 45px;
  left: 210px;
  top: 222px;
}
```



预览器效果

用到的UI组件:

Swiper/Stack/Text/Progress/Div

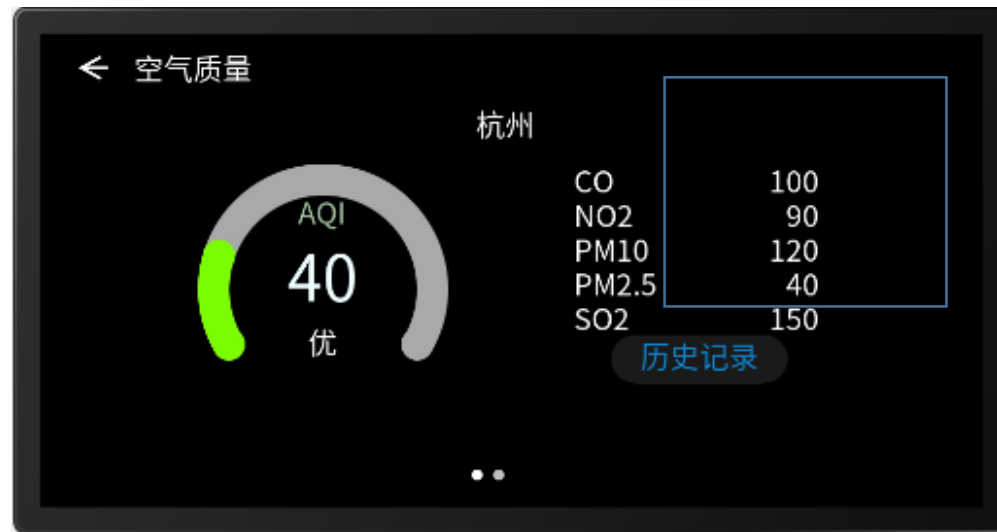
UI界面开发：内容区详细信息开发和调试

index.html

```
<div id="detailInfo" class="detailInfo">
  <text class="key">CO</text>
  <text class="value">100</text>
  <text class="key">NO2</text>
  <text class="value">90</text>
  <text class="key">PM10</text>
  <text class="value">120</text>
  <text class="key">PM2.5</text>
  <text class="value">40</text>
  <text class="key">SO2</text>
  <text class="value">150</text>
  <div class="history">
    <input class="historyBtn">历史记录</input>
  </div>
</div>
```

index.css

```
.detailInfo {
  width: 256px;
  height: 256px;
  left: 544px;
  top: 58px;
  flex-wrap: wrap;
  /* flex-direction: column; */
}
.key {
  width: 125px;
  height: 35px;
}
.value {
  width: 125px;
  height: 35px;
  text-align: right;
}
.history {
  width: 256px;
  height: 256px;
  justify-content: center;
}
.historyBtn {
  width: 180px;
  height: 50px;
  background-color: #1a1a1a;
  color: #1085CE;
}
```



预览器效果

用到的UI组件：

Div/Text/Input

UI界面开发：内容区其他信息开发和调试

index.html

```
<div id="otherInfo" class="otherInfo">
  <text class="updateTime">更新时间: 10:38</text>
  <text class="infoSource">信息来源: tianqi.com</text>
</div>
```



预览器效果

index.css

```
.otherInfo {
  width: 960px;
  height: 38px;
  top: 310px;
}

.updateTime {
  width: 480px;
  height: 28px;
  font-size: 25px;
  color: #A9A9A9;
  text-align: right;
}

.infoSource {
  width: 450px;
  height: 28px;
  font-size: 25px;
  color: #A9A9A9;
  text-align: left;
  margin-left: 24px;
}
```

用到的UI组件:

Div/Text

业务主要逻辑实现

- 返回键实现应用退出功能
- 两屏页面实现滑动切换
- 空气质量数据动态更新

返回键功能实现

index.js

```
import app from 'system.app';  
  
exitApp() {  
    console.log("exit app.");  
    app.terminate();  
}
```

index.html

```
<image class="" src="xxx"  
onclick="exitApp"></image>
```

两屏页面滑动切换功能实现

index.js

```
data:{
    firstIndicator : '#ffffff ',
    secIndicator : '#262626 ',
},
onSwiperChange(e) {
    if(e.index==0) {
        this. firstIndicator == '#ffffff';
        this. secIndicator == '#262626';
    } else {
        this. firstIndicator == '# 262626';
        this. secIndicator == '#ffffff';
    }
}
```

index.html

```
<swiper ... onchange="onSwiperChange">
.....
<div class="indicator">
<div style="background-
color={{firstIndicator}}";...></div>
<div style="background-
color={{secIndicator}}";...></div>
</div>
```

空气质量数据动态更新功能实现

index.js

```
data:{
  currIndicatorIndex: 0,
  airData: [
    {city:'杭州',aqi:40,aqi_quality:'
    优',co:100,no2:90,pm10:120,pm2_5:40,so2:150},
    {city:'广州',aqi:40,aqi_quality:'
    优',co:100,no2:90,pm10:120,pm2_5:40,so2:150}],
    .....
  },
  onSwiperChange(e) {
    this. currIndicatorIndex = e.index;
    .....
  }
```

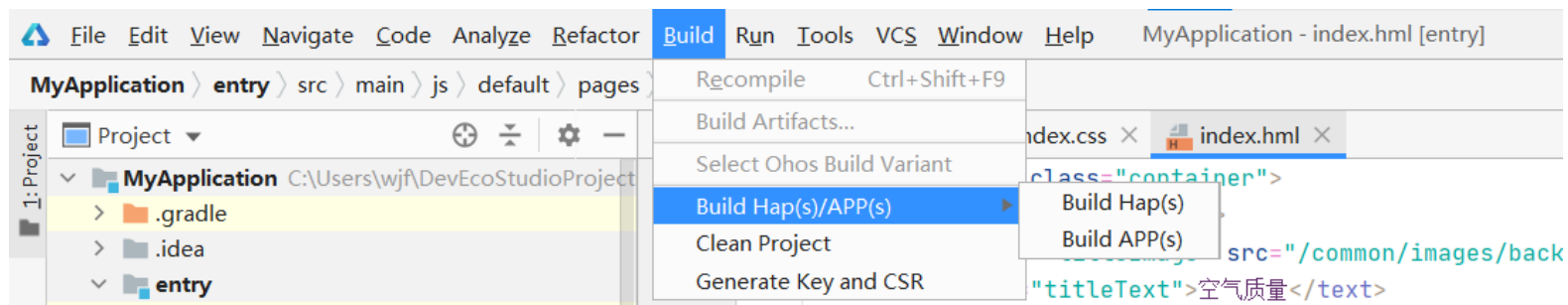
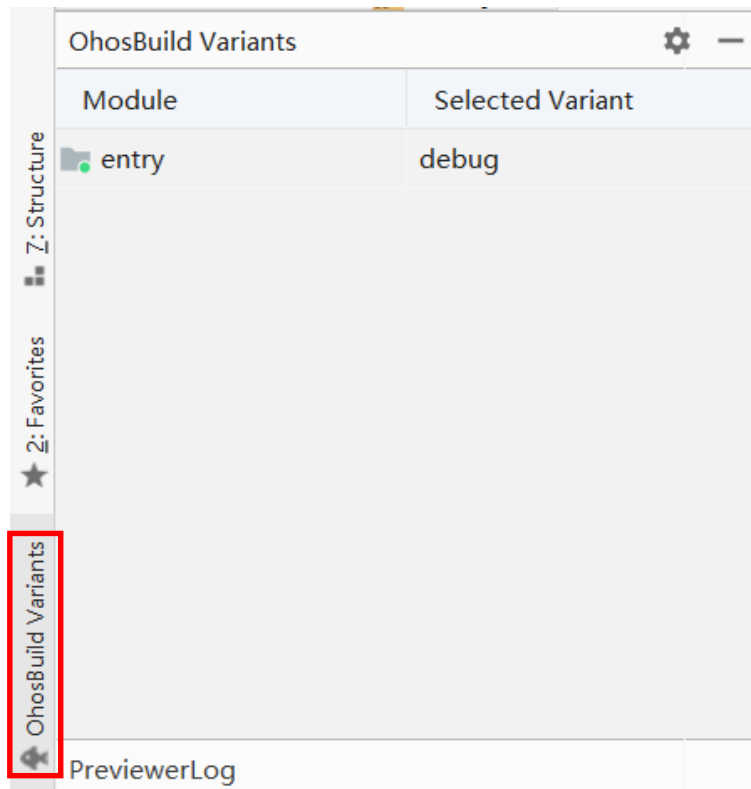
index.html

```
<stack id="swiper_page1" class="swiper_page1">
  <text id="city" class="city">{{airData[0].city}}</text>
  <progress id="circleBar" class="circleBar" type="arc" percent="20"></progress>
  <text class="aqi">AQI</text>
  <text class="aqiValue">{{airData[0].aqi}}</text>
  <text class="aqiLevel">{{airData[0].aqi_quality}}</text>
  <div id="detailInfo" class="detailInfo">
    <text class="key">CO</text>
    <text class="value">{{airData[0].co}}</text>
    <text class="key">N02</text>
    <text class="value">{{airData[0].no2}}</text>
    <text class="key">PM10</text>
    <text class="value">{{airData[0].pm10}}</text>
    <text class="key">PM2.5</text>
    <text class="value">{{airData[0].pm2_5}}</text>
    <text class="key">S02</text>
    <text class="value">{{airData[0].so2}}</text>
```

应用打包（不带签名）

- 1、打开左下角的**OhosBuild Variants**，检查并设置模块的编译构建类型为debug，默认类型为debug；
- 2、在主菜单栏，点击**Build > Build APP(s)/Hap(s) > Build Hap(s)**，生成不带签名的调试Debug HAP。

hap包生成位置：工程根目录\entry\build\outputs\hap\debug\entry-debug-lite-unsigned.hap



开发板上安装HAP包

- 目前两种方式可以把HAP包发送到开发板：
 - NFS（工具：nfs1169.exe）
 - SD卡

以NFS方式，在开发板上安装HAP包

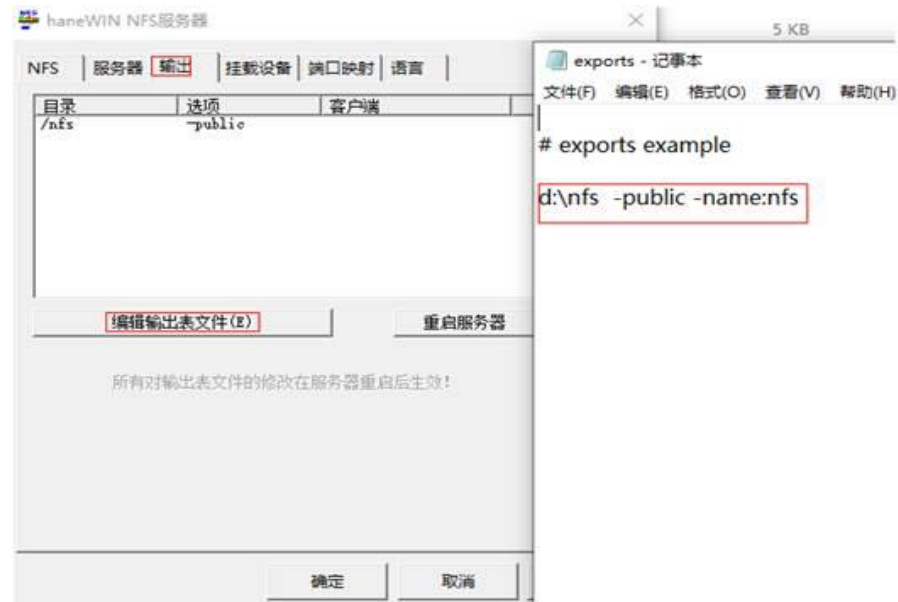
- 1、以管理员身份运行nfs服务器
- 2、服务器目录设置
- 3、关闭软件，进入nfs安装目录，以管理权限执行 RestartServer.bat
- 4、进入Hitool工具中串口窗口，并输入

```
mkdir user
```

```
mount 192.168.1.2:/nfs /user nfs
```

（注意，可能需要在PC上进入控制面板，关闭防火墙）

- 5、将编译生成的不带签名的HAP和安装工具（dev_tools文件夹），拷贝到PC侧c:\nfs
- 6、单板侧关闭校验机制，串口窗口中输入： ./user/dev_tools/bin/bm set -s disable
- 7、安装Hap包，串口窗口中输入： ./user/dev_tools/bin/bm install -p /user/airquality.hap



```
OHOS # ./user/dev_tools/bin/bm set -s disable
OHOS # 01-02 20:41:56.008 15 81 I 01800/Samgr: Initialize Client Registry!
success
01-02 20:41:56.009 5 53 D 01800/Samgr: Judge Auth<bundlems, BmsInnerFeature> ret:0
01-02 20:41:56.009 5 53 D 01800/Samgr: Find Feature<bundlems, BmsInnerFeature> id<58, 0> ret:0
01-02 20:41:56.009 15 81 I 01800/Samgr: Create remote sa proxy[0x204b6fd0]<bundlems, BmsInnerFeature>!
01-02 20:41:56.019 5 51 I 00000/(null): current sign mode is 0
```

体验HarmonyOS应用



安装效果



运行效果

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home, and
organization for a fully connected,
intelligent world.

**Copyright©2021 Huawei Technologies Co., Ltd.
All Rights Reserved.**

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

