



高等学校应用型新工科创新人才培养计划指定教材
高等学校汽车工程类专业“十三五”课改规划教材



Android Studio

车载终端 应用开发技术

青岛英谷教育科技股份有限公司 编著
德州学院

立体化教辅

- ▶ 教学PPT
- ▶ 教学大纲
- ▶ 考试大纲
- ▶ 开源代码
- ▶ 配套设备
- ▶ 在线题库
- ▶ 视频讲解



西安电子科技大学出版社
<http://www.xduph.com>

高等学校应用型新工科创新人才培养计划指定教材

高等学校汽车工程类专业“十三五”课改规划教材

车载终端应用开发技术

青岛英谷教育科技有限公司

编著

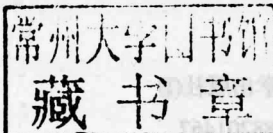
德州学院

主 编 柳永亮

副主编 侯崇升 吴 娜

编 委 王 燕 赵长利 孟俊英

王文成 吴延霞



西安电子科技大学出版社

内 容 简 介

车载终端平台上的应用开发是目前车联网领域的重要组成部分,本书首次尝试基于 Android Studio 开发环境介绍终端应用层的开发技术。

本书主要介绍了 Android 开发的基础知识和车载终端主流应用开发技术的相关知识。全书共分为 10 章,包括车载系统与应用开发概述、活动、用户界面、意图、服务、数据存储、通信开发、行车记录仪开发、车载地图开发和 OBD 开发等内容。

本书内容精练、案例翔实、适用面广,可作为车联网、物联网、计算机科学与技术、计算机软件等专业的教材使用,也可为有志于从事车联网工作的读者提供一定的参考。

图书在版编目(CIP)数据

车载终端应用开发技术/青岛英谷教育科技有限公司,德州学院编著. —西安:西安电子科技大学出版社,2018.2

ISBN 978-7-5606-4826-2

I. ① 车… II. ① 青… ② 德… III. ① 汽车—移动终端—计算机网络—网络安全
IV. ① U463.67 ② TN929.53

中国版本图书馆 CIP 数据核字(2018)第 011147 号

策 划 毛红兵

责任编辑 许青青

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2018 年 2 月第 1 版 2018 年 2 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 22

字 数 520 千字

印 数 1~3000 册

定 价 57.00 元

ISBN 978-7-5606-4826-2/U

XDUP 5128001-1

如有印装问题可调换

高等学校汽车工程类专业
“十三五”课改规划教材编委会

主 编 柳永亮

副主编 侯崇升 吴娜

编委 王燕 赵长利 孟俊焕

王文成 吴延霞

◆◆◆ 前 言 ◆◆◆

从全球发展态势来看，车联网是一个新兴事物，其技术特点是跨学科、跨领域。随着交通和交通工具智能网联化的高速发展，车联网市场注定将成为极具价值的新兴市场。当前，车联网市场正面临迈向智能网联汽车时代的最佳“超车机会”——汽车正在成为继笔记本电脑、手机、平板等移动设备之后最主要的互联网终端设备之一，也必将是未来十年创新创业爆发最多的产业之一，更是物联网众多应用场景中重要的应用范畴。同时，V2X将是未来5G通信技术的主要应用场景。

车联网的应用场景较多，本书选取车载终端作为移动应用开发的目标平台，基于Android5.0系统，采用Android Studio作为开发环境。书中首先从Android开发的基础知识入门，中间合理穿插项目案例讲解，理论与实践结合，让读者可以快速掌握Android基础开发要点，达到入门级水平。在此基础上，针对当前车载端的主流应用，本书详细介绍了如何进行专业开发，如行车记录仪、定位与导航、语音命令识别、OBD数据等应用的开发。本书以由易到难、循序渐进的原则安排项目开发内容，使读者可以充分理解和掌握具有车联网特色的终端应用开发技术。

总结而言，本书对国内车联网专业高等教育教材有建设性贡献，第一次基于Android Studio开发环境，对移动终端平台，尤其是车载终端的应用层开发技术作了总结。

本书由青岛英谷教育科技股份有限公司和德州学院编著，参与编写工作的有宋乃华、卢超、王一军、邵舟、姜竞超、卢玉强、刘伟伟、袁文明、孙锡亮、金成学、邓宇、张玉星、侯方超、王燕等。本书在编写期间得到了各合作院校专家及一线教师的大力支持和协助。在本书出版之际，要特别感谢合作院校师生给予我们的建议和鼓励，感谢开发团队每一位成员所付出的艰辛劳动与努力。

由于编者水平有限，书中难免有不当之处。读者若在阅读过程中发现问题，可以通过邮箱(yinggu@121ugrow.com)联系我们，以期不断完善。

本书编委会
2017年12月

目 录

第 1 章 车载系统与应用开发概述.....1	
1.1 车载系统概述.....2	
1.1.1 车载系统的特点.....2	
1.1.2 主流车载系统解析.....3	
1.2 YunOS Auto.....5	
1.3 认识 Android Studio.....8	
1.3.1 Windows 系统下环境配置.....9	
1.3.2 下载和安装 Android Studio.....12	
1.3.3 认识开发界面.....13	
1.3.4 创建第一个 APP 项目.....14	
1.3.5 运行项目.....17	
本章小结.....20	
本章练习.....20	
第 2 章 活动(Activity).....21	
2.1 Activity.....22	
2.1.1 Activity 生命周期.....22	
2.1.2 Activity 示例.....24	
2.1.3 设置生命周期.....28	
2.2 Android 中的资源使用.....30	
2.2.1 字符串资源.....31	
2.2.2 图片资源.....34	
本章小结.....35	
本章练习.....35	
第 3 章 用户界面.....37	
3.1 用户界面元素分类.....38	
3.1.1 视图组件(View).....38	
3.1.2 视图容器(ViewGroup).....39	
3.1.3 布局方式(Layout).....40	
3.2 事件处理机制.....40	
3.3 布局方式(Layout).....44	
3.3.1 线性布局(LinearLayout).....45	
3.3.2 相对布局(RelativeLayout).....48	

3.3.3 表格布局(TableLayout).....49	
3.3.4 绝对布局(AbsoluteLayout).....52	
3.3.5 框架布局(FrameLayout).....53	
3.3.6 网格布局(GridLayout).....54	
3.4 提示信息 and 对话框.....56	
3.4.1 提示信息(Toast).....56	
3.4.2 对话框.....57	
3.5 常用 Widget 组件.....60	
3.5.1 Widget 组件通用属性.....60	
3.5.2 文本框(TextView).....61	
3.5.3 按钮(Button).....62	
3.5.4 编辑框(EditText).....62	
3.5.5 复选框(CheckBox).....62	
3.5.6 单选按钮组(RadioGroup).....63	
3.5.7 下拉列表(Spinner).....64	
3.5.8 图片视图(ImageView).....71	
3.5.9 滚动视图(ScrollView).....72	
3.5.10 网格视图(GridView).....73	
3.5.11 列表视图(ListView).....75	
3.6 菜单.....76	
3.6.1 选项菜单(OptionsMenu).....76	
3.6.2 上下文菜单(ContextMenu).....78	
3.7 ActionBar.....82	
3.7.1 ActionBar 的显示与隐藏.....82	
3.7.2 修改图标和标题.....83	
3.7.3 添加 ActionBar.....84	
3.7.4 添加导航按钮.....86	
3.7.5 添加 ActionView(活动视图).....86	
本章小结.....88	
本章练习.....88	
第 4 章 意图(Intent).....89	
4.1 Intent 概述.....90	

4.1.1 Intent 组成属性	90	第 7 章 通信开发	163
4.1.2 Intent 启动	93	7.1 通信方式简介	164
4.2 Intent 消息传递	97	7.2 Socket 通信	164
4.3 Intent Filter	102	7.2.1 Socket 和 ServerSocket	164
4.3.1 <intent-filter>元素	102	7.2.2 Socket 应用	167
4.3.2 IntentFilter 类	105	7.3 HTTP 网络编程	173
4.4 广播接收 Intent	105	7.4 Wi-Fi	178
4.4.1 广播和接收 Intent 机制	106	7.4.1 Wi-Fi 开发概述	179
4.4.2 广播 Intent 示例	108	7.4.2 扫描周围的 Wi-Fi	181
4.5 设置 Activity 许可	110	7.4.3 连接到指定 Wi-Fi 网络	184
本章小结	113	7.5 Bluetooth(蓝牙)	191
本章练习	113	7.5.1 传统蓝牙概述	191
第 5 章 服务(Service)	115	7.5.2 传统蓝牙通信	194
5.1 Service 简介	116	7.5.3 BLE 技术概述	218
5.2 实现 Service	116	7.5.4 通过 BLE 技术与设备通信	220
5.2.1 创建 Service 类	117	本章小结	235
5.2.2 Service 的使用	118	本章练习	236
5.2.3 Service 示例	121	第 8 章 行车记录仪开发	237
5.3 Android 系统服务	126	8.1 播放音频和视频	238
本章小结	131	8.1.1 播放音频	238
本章练习	131	8.1.2 播放视频	241
第 6 章 数据存储	133	8.2 随车拍	245
6.1 数据存储简介	134	8.3 简易行车记录仪	250
6.2 Preference 存储数据	134	8.4 车载摄像头	261
6.2.1 访问 Preference 的 API	134	本章小结	262
6.2.2 Preference 应用	135	本章练习	262
6.3 File 存储数据	141	第 9 章 车载地图开发	263
6.3.1 文件操作	141	9.1 高德地图配置	264
6.3.2 File 应用	142	9.1.1 配置 Key 及开发权限	264
6.4 SQLite 存储数据	145	9.1.2 工程配置(添加开发包)	265
6.4.1 SQLite 简介	145	9.2 显示地图	267
6.4.2 SQLite 数据库操作	146	9.3 显示定位结果	270
6.4.3 SQLiteOpenHelper	151	9.4 显示定位蓝点	279
6.5 数据共享 ContentProvider	156	9.5 高德导航开发	283
6.5.1 ContentProvider	156	本章小结	291
6.5.2 ContentResolver	157	本章练习	292
6.5.3 ContentProvider 应用	158	第 10 章 OBD 开发	293
本章小结	161	10.1 车联网场景	294
本章练习	161	10.1.1 车载终端互联产品	294

10.1.2 T-BOX 与 OBD	295	10.6 故障码	330
10.2 OBD 简介	297	10.6.1 故障码简介	330
10.3 蓝牙数据传输	299	10.6.2 故障码获取	331
10.4 虚拟仪表开发	305	本章小结	341
10.5 OBD 综合开发	314	本章练习	341
10.5.1 DI 框架应用	315	参考文献	342
10.5.2 综合界面布局	316		
10.5.3 代码功能解析	320		

第 3 章 车载系统应用开发概述

本章目标

- 了解车载系统的定义及组成
- 了解车载系统的组成及分类
- 了解 YunOS Auto 车载系统的特色
- 了解 Android Auto 的特色
- 掌握 Android Auto 的应用原理
- 掌握 Android Auto 的开发流程
- 使用 Android Studio 完成第一个项目

第1章 车载系统与应用开发概述



本章目标

- 了解车载系统的意义及现状。
- 了解车载系统的特点及其应用。
- 了解 YunOS Auto 车载系统的特色。
- 了解 Android Studio 的特点。
- 掌握 Android Studio 的环境搭建。
- 掌握 Android Studio 的开发界面。
- 使用 Android Studio 完成第一个项目。



1.1 车载系统概述

随着移动通信技术、计算机技术以及互联网技术的发展,信息实时交换成为可能,将这些技术与汽车有机地结合已成为车载系统(也可称作车载嵌入式系统)的发展方向和研究热点。车载嵌入式系统作为汽车电子技术的核心,集中体现了车联网的发展方向。

从移动互联网的角度看,未来的汽车就是一台“移动的电脑”,是一个移动的信息载体。汽车的互联网化在为人们带来便利的同时,也带来了互联网方面的不确定性因素,如车辆遥控安全、用户信息隐私以及电子产品的可靠性问题。

车载智能终端为用户提供信息通信、地图导航、生活服务和安防等功能,就像互联网中的电脑、移动端的手机,车载终端是车主与车联网得以交互的平台,更是车联网中最重要的移动节点。

由于当前各大汽车公司对汽车智能网联化趋势的认同,新车型的车载辅助功能在增多,服务配置也在加强,所以靠单一的 ECU(Electronic Control Unit, 电子控制单元)不足以协调和支持整个车辆内部和外部数据之间的高速通信,因此引入车载终端的操作系统已经势在必行。

1.1.1 车载系统的特点

自 20 世纪 70 年代起,嵌入式系统开始逐步进入汽车电子领域,从发动机控制逐步深入到底盘控制、车身舒适性与安全控制,又扩展至信息通信与车载多媒体,现已成为推动车辆技术进步的基本技术手段。80 年代中期,现场总线技术被引入到车载嵌入式系统中,出现了以 CAN 总线为代表的车载网络系统。基于总线通信以微控制器为基本节点的分布式控制系统将车载嵌入式控制应用推向更高的阶段。

21 世纪以来,车载嵌入式系统的焦点逐步聚焦在高性能微控制器、实时多任务操作系统,以及高可靠性和高实时性现场总线技术上。车载嵌入式系统的特殊性主要体现为汽车作为被控对象的复杂性,现代汽车是耦合了物理学、机械学、电工学、动力学、流体力学、热力学甚至电化学问题的综合体,对它们的理解所需要的知识结构从学科跨度来讲,大大突破了电子与控制专业的局限。具体而言,车载嵌入式系统要满足以下几点要求:

- ✧ 实时性强,以满足车辆高速移动时的安全性和发动机以及整车控制器精确控制的需要。
- ✧ 适应恶劣的工作环境,必须能够满足剧烈变化的苛刻环境要求,温度、光照、潮湿的变化甚至振动等都会对控制系统的性能造成不利影响。
- ✧ 灵活性大,可以满足各种法规、系统兼容及不同客户的需求,适应多变的工作环境、不同风格的驾驶员操作。
- ✧ 可靠性和安全性要高,控制系统需要有诊断、容错、失效安全保护等功能。
- ✧ 涉及范围广,不仅融合了机械、电磁、流体、软件和硬件设计,而且系统建



模也需要结合控制算法和系统标定数据。

◆ 可实现系列化、规模化、低成本的生产。

1.1.2 主流车载系统解析

汽车产业历经十余年高速成长后迎来行业拐点,传统车企也在谋求破局,如何通过实现数据融合场景互通,让汽车运行在互联网上,通过大数据与操作系统形成智能化的操控模式,推动中国汽车工业的互联网化改造以便参与更高层次的市场竞争,将构成互联网汽车未来竞争力的关键。汽车行业正在成为互联网下一个进军的重要目标,互联网将使其发生革命性的变化。实际上,汽车行业的文化相对保守,各个企业之间有技术壁垒,在这一点上与互联网的“开放”思想有所冲突,也是未来车联网发展需要破解的重要难题。

众所周知,操作系统是智能设备的基础和灵魂。基于何种操作系统设计开发车载终端、手机移动端以及云服务平台系统,使其能够高效地支撑整个车联网系统,成为车联网产业链中应用开发的基石,这是一个重要、紧迫且值得深入研究的课题。

微软的 Windows 操作系统在 PC 行业独大,尽管有诸多类 UNIX、类 Linux 系统层出不穷,但依然不能撼动 Windows 操作系统在 PC 领域的领导地位。直到智能手机及其他手持式移动设备的出现,才打破了终端操作系统 Windows 一家独大的局面,出现了 Linux、QNX、Android、YunOS 等系统。另外,各种系统自身均有不同的生态环境,这让操作系统之争变得更加激烈。

那么,在车联网产业生态中,具体到车载移动端方面,其操作系统的开发应用的现状如何呢?

车载智能终端在操作系统方面的选择变得日益重要。车载终端与手机同样属于智能移动终端,除了完成本身的信息传播类功能之外,其娱乐、资讯等功能也越来越受到重视,集成更多的传感器来实现更多先进的功能也成了普遍现象。不同系统拥有不同的生态环境,这意味着对操作系统的选择,其实也是对其生态环境的选择。目前车载终端市场上存在多种操作系统平台,主要有 QNX(Quick UNIX)、Linux(Genivi)和 Android。

除了汽车企业前装市场份额较大的 QNX 车载系统外,开源操作系统平台 Android 和 Linux 也有希望成为车载终端的主流操作系统。Android 专为触摸操作进行了优化,体验良好,可个性化定制,应用丰富且应用数量快速增长,已经形成了成熟的网络生态系统。而 Genivi 联盟主推的 Linux 操作系统在车载平台上应用较广,具有实时、稳定的优点。为此,我们针对汽车行业中主流车企所搭载的车载终端系统进行了市场调研,并绘制成表格,不完全统计结果如表 1-1 所示,表中对涉及的车载系统的使用情况做了专门的优劣势分析和对比。

从汽车行业市场的实际应用情况可以发现,QNX 车载系统属于商业级嵌入式操作系统,所以它在性能以及服务上有较好的保证,但是价格昂贵,且不公开核心代码,可定制性较差;而 Linux 和 Android 系统属于开源操作系统,核心代码是公开的,尽管其在技术支持和服务上有所欠缺,且对开发人员有较高的要求,但具有用户可定制性好、可持续开发性强及费用低廉等优势。正因为开源系统具有的特点及优势,从目前来看,它有可能成



为车载端操作系统应用的主要推动力。

表 1-1 主流车联网品牌及其采用系统统计表

车载系统	所属公司	车联网品牌	优点	缺点
QNX	通用	ON STAR	稳定、可靠、安全，实时性好，兼容 Android 系统，市场占有率高	非开源，价格昂贵，开发成本高，仅适用于高端车型，需结合应用层开发
	丰田	Entune		
	吉利	G-Netlink		
	福特	SYNC3		
	FCA	Uconnect		
	大众	MIB II		
	沃尔沃	Sensus(当前)		
Android	奔驰、宝马、奥迪	COMMAND、ConnectedDrive、MMI	开源，便于应用层开发，扩展性好，开发成本低，阿里云具有导航本土化和互联网生态优势	根据车规级标准要求，需要对原 Android 系统的底层架构进行相应的改造
	雷诺、DS	R-Link DSConnect		
	上汽+斑马	YunOS Auto		
	沃尔沃	Sensus(未来)		
Android Auto	谷歌	Android Auto	兼容多种车载系统	依赖手机映射到车机端
Linux	凯迪拉克	CUE	稳定、可靠，开源，开发成本低	需结合应用层开发
	特斯拉	Autopilot		
车机映射	百度	Carlife	兼容多种车载系统	依赖手机映射到车机端
	苹果	CarPlay		

下面对表 1-1 中部分车载系统或车联网品牌进行详述。

目前，QNX 已成为汽车领域最大的操作系统。QNX 遵从 POSIX 规范，类似于 UNIX 实时操作系统，目标市场主要是面向嵌入式系统。QNX 是建立在微内核上的，这个架构的特点是既可以支持小型的缺乏运行资源的嵌入式系统，也适合大型分布式实时系统。该系统的大多数系统服务是基于多进(线)程的形式来表示的，这些进(线)程被封装在自己的地址空间里面，与用户空间隔离。微内核本身提供操作系统的基本管理功能，扩展模块提供设备、网络、文件和图像用户接口，这些模块都是可以裁减的。这样的特点让 QNX 可以适合非常广泛的嵌入式应用场景，支持更多的嵌入式处理器，提供可靠性很强的操作系统环境和接近实时的运行环境。QNX 是基于微内核的嵌入式操作系统中最成功的一个。微内核“跑”在自己的空间，而中间件、应用和驱动都在内核空间外运行，如果某一部分程序出错，对内核不会造成任何影响，这就是为什么在很多重要领域(比如外科手术、核电站控制室及军工产品)都使用 QNX 系统的原因。

QNX CAR 系统符合 ISO26262 标准的 ASIL-D 级要求，这是标准中定义的最高安全等级。针对功能性安全标准的认证提供独立的验证，即当产品被用于乘用车的电气、电子和基于软件的系统中时，能提供非常高的可靠性并且降低风险。



苹果的 CarPlay 并不是直接运行 iOS 应用的车载系统，它必须通过手机 iPhone 和车机系统建立连接，打开手机上的 CarPlay 应用，实现手机和车机的双向控制。新版本的手机系统可以利用 Wi-Fi 或蓝牙与 CarPlay 系统连接。

Android Auto 与 CarPlay 类似，也在手机上运行并投影至车载屏幕上。如果在手机上安装了一个适配车载系统的应用程序，那么应用程序将会显示在车载系统上。CarPlay 符合苹果保守、非开源的作风；Android Auto 则把 API 和 SDK 都提供给了开发者。

Android 不是实时系统，在汽车运行中，从安全的角度考虑，对实时通信的要求非常高，而且安卓的系统长期运行后会产生卡机，所以有观点认为安卓系统不适合作为车载底层操作系统。不过，谷歌不久前发布了基于 Android N 定制的内有汽车专用硬件控制的车载系统，系统可以访问 CAN 总线，从而实现对诸如座椅加热、空调制冷和车窗控制等局部系统的控制操作。

1.2 YunOS Auto

互联网自全球化以来，已经进入人类生产生活的各个领域，并且成为不可或缺的组成部分，极大地提高了行业的生产效率，而汽车行业也是当下众多互联网企业争夺的领域。

目前在国内，汽车厂家将车载端系统成功融入传统车机中的就数上汽与阿里合作开发的荣威互联网汽车 RX5 了。阿里为上汽荣威 RX5 开发的 YunOS Auto 系统，完全以车为主体，不再依赖手机(手机与车机进行界面映射是过渡阶段)，并与原车各个系统的 ECU 控制单元深度融合，实现了车内外各类数据的实时传输，完成了车联网初级阶段的跨领域技术融合，是一款真正意义上的独立的国产车载系统。将“车”作为移动端融入到互联网中，成为互联网的一个重要入口，从而带来了不同于以往车机系统的良好用户体验。

YunOS Auto 系统开发基于四点：第一是安全性；第二是开放、完整的阿里生态服务；第三是一批强有力的合作伙伴；第四则是像 Android 一样可以定制。该系统整合了阿里原有的生态资源，用于车载移动服务，如高德高精度导航、人机语音交互、云平台服务、车辆远程控制、身份认证、用户个性化定制等。当然，YunOS Auto 系统也是基于 Android 开源系统的，针对车载端应用场景经过特殊裁减而形成的专用系统，二者的系统架构如图 1-1 所示，所以，经由 Android Studio 开发的 APP 完全适应于 YunOS Auto 这样的车载系统。

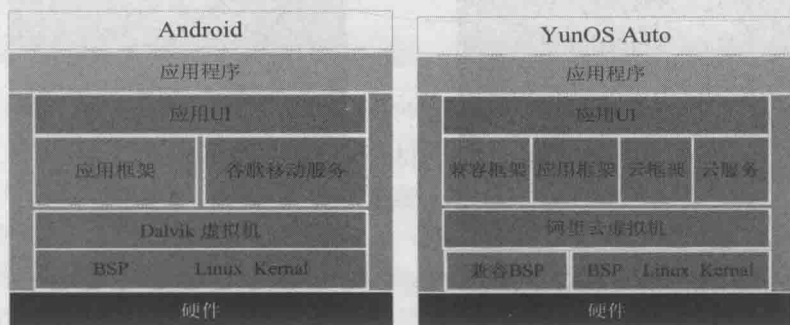


图 1-1 Android 和 YunOS Auto 系统架构



YunOS Auto 系统更大的突破之一在于它的服务、任务处理、设置导航等都在云端进行操作,每个车主将拥有独立的 ID 身份账号,脱离手机的局限,而不是简单地将手机功能投影到屏幕上。YunOS Auto 智能车载系统采用卡片式的图形交互设计,用户在桌面上就可以一览各个应用的详细信息并进行操作,用户可以通过和汽车对话的方式控制各种功能。车与系统融为一体,扩展了用户群的边界。阿里 YunOS Auto 的突破之处还在于自动驾驶学习,系统可不断学习和记忆用户的驾驶路线信息,熟悉用户的驾驶习惯,与中控、仪表、HUD 深度整合,让汽车地图有更多场景服务能力,比如在旅途中可以提前告知各种生活餐饮信息与加油站等信息,出现道路关闭或拥堵时可以提前通知用户并建议新的路线,还可以对用户生活中的各类需求喜好做出提示,推送车主喜欢的音乐等。该系统集成在车载端的中控上,系统界面效果如图 1-2 所示,对应手机 APP 界面如图 1-3 所示。



图 1-2 车载中控端系统界面



图 1-3 手机端应用界面



阿里互联网汽车的本质是“让汽车运行在互联网上”，而不是仅仅让汽车连上互联网，这是未来车联网的进化方向。通过 YunOS Auto 系统的大数据和云计算能力，真正提升了汽车的能力，通过车载系统平台将大量驾驶数据整合为可供车主定制使用的数据库，由此让汽车成为 YunOS Auto 多终端布局中重要的一环，进一步放大了阿里的生态圈。另一方面，汽车在数据场景互联中真正变成了一个入口，在汽车产业链上也有了潜在的利润增长点。这背后是阿里 YunOS Auto 通过对车辆接收的来自道路多元化指示信号数据的收集分析，形成应对策略并实现半自动驾驶，这是人工智能化的一步。从驾驶层面来看，系统已经做到了对驾驶的智能化控制与管理。

阿里 YunOS Auto 与国内互联网巨头相比，在于其在操作系统底层已经占据先发优势，目前已经成长为国内市场移动端的第三大操作系统，也是当下国内车载端使用最成功的互联系统。加之阿里与上汽合作构成了量产优势，这些都是在车联网层面赶超国外巨头的机会。

说到底，车联网并不是简单地做一个硬件产品或者一个连接器，所有的生态系统必须要有操作系统作为基层，这是掌控主导权的基础。在破除谷歌与苹果的局限的同时，YunOS Auto 使得汽车具备了互联网特征的又一大终端实体，并将人-车-服务的链条打通。这种以新兴的互联网企业与传统汽车厂商合作模式的成功尝试，已经引领了一波车载智能互联终端配置，也代表未来几年在车载终端上进行特色应用开发具有极大的市场潜力。YunOS Auto 系统支持的车载端的硬件设备如图 1-4 所示。



图 1-4 YunOS Auto 支持的硬件设备

基于以上对几个主流车载系统的综合分析，本书选择以 Android 系统作为车载应用开发的基础，并在书中逐步介绍在 Android 系统上如何利用 Android Studio 工具进行车载应用程序的开发。

本书以车载终端的特色场景作为车联网专业的应用开发方向，顺应移动端应用开发的技术潮流，选择 Android Studio 作为开发环境，这样能使学生在以后的学习和工作中、在使用 Android 项目开发工具时做到无缝衔接。



1.3 认识 Android Studio

Android Studio 是谷歌在 IntelliJ IDEA 的基础上作的二次开发,目的是为 Android 开发专门定制的 IDE(集成开发工具)。

IntelliJ IDEA 由捷克一家专做 IDE 的 JetBrains 公司出品,它的社区版是开源的。IntelliJ IDEA 被认为是当前 Java 开发效率最快的 IDE,整合了开发过程中很多实用的功能。开发者通过使用快捷键,几乎可以不用鼠标就能完成想要做的事情,最大限度地加快了开发效率。

作为谷歌官方的 IDE,Android Studio 提供了开发和构建 Android 应用的所有工具,包括智能代码编辑器、布局编辑器、代码分析和调试工具、应用构建系统、模拟器以及性能分析工具等,这些工具在 Windows、Mac OS X 和 Linux 平台上均可运行。

Android Studio 是 Android 平台上构建高质量应用的高效开发工具,所开发的应用适用场景包括手机、平板、Android Auto、Android Wear 和 Android TV。Android Studio 于 2013 年 5 月 16 日在谷歌 I/O 大会上正式对外发布,目前已更新到 3.0 版本。

需要注意的是,Android 的母公司谷歌已于 2016 年 11 月 2 日在 Google Android Developer 官方微博发文,正式宣告停止对 Eclipse ADT 的支持。这意味着,以后新加入的开发者在进行基于 Android 系统移动终端应用的开发时,采用 Android Studio 作为首选开发环境将是大势所趋。对比之前成熟的 Eclipse ADT,下面介绍 Android Studio 的特点。

(1) 智能代码编辑器。

Android Studio 最突出的特点就是智能代码编辑器,它能非常高效地完成代码补全、重构和代码分析。它还支持多种实用的视图模式,如演示模式、免打扰模式。它的快捷键、代码的显示方式、颜色、主题等都是可配的。

(2) 代码模板和 GitHub 集成。

Android Studio 的新建项目向导使新建项目变得非常简单。新建项目的途径有三种:新建一个全新的 Activity 模板;从 GitHub 上直接导入项目;直接通过导入代码模板来快速开始项目的新建。

(3) 专为 Android 设备所开发。

Android Studio 支持构建适用于 Android 手机、平板电脑、Android Wear、Android TV、Android Auto 的应用。全新的项目视图和模块支持让应用和资源管理变得更加轻松。

(4) 快速且功能丰富的模拟器。

Android Studio 提供了丰富的模拟器,可以更加便捷地供开发者选择,使用模拟器可以非常方便地测试和调试应用程序,从而提高开发的效率。

(5) 基于 Gradle 的灵活构建系统。

Android Studio 通过集成 Gradle 构建系统,支持自动构建、依赖管理和自定义构建配置等功能,通过在同一个项目中定义不同的配置来构建多个有特殊需求的应用。

(6) 强大的即时运行功能。

顾名思义,即时运行(Instant Run)就是开发者一边写代码,一边可以在模拟器或真机



上立即看到修改后的运行效果，不用重新开始编译运行，大大提高了开发效率。

需要特别注意的是，若开发者选择使用 Android Studio 作为移动端 APP 的开发环境，那么相应的开发平台的配置条件必须满足表 1-2 的基本要求，这样才能实现良好的开发体验，顺利及时地完成开发项目。

表 1-2 Android Studio 开发系统配置要求

环境要求	Windows 系统(本书所采用的系统)
操作系统版本	Windows®7/8/10(32 位或 64 位)
内存大小	RAM 不低于 4 GB，强烈推荐 8GB RAM
硬盘空间	可用磁盘空间不低于 8 GB，推荐固态硬盘
JDK 版本	JDK≥8
屏幕分辨率	不低于 1280×800
系统网络	开通网络连接，用于下载 SDK

1.3.1 Windows 系统下环境配置

本节将针对教学中常用的 Windows 系统平台下如何下载和搭建 Android Studio 开发环境并作相应的参数配置进行详细说明。

开始使用 Android Studio 之前，需要先进行基于 Windows 系统下的 JDK 配置，本书使用的是 JDK1.8 版本。首先，下载安装 JDK，其步骤如下：

(1) 打开 JDK 的下载地址 <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>，如图 1-5 所示。

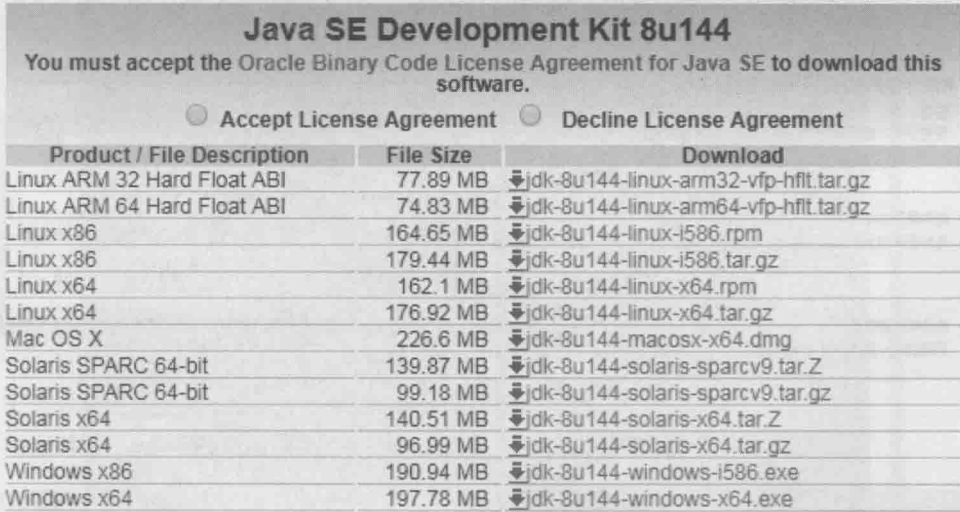


图 1-5 JDK1.8 下载界面