

Работа с Android и Linux

на smart-решениях SIMCom

Wireless Solutions

Для многих типов устройств разработчикам хотелось бы иметь универсальный компонент с функциями микропроцессора, включающего память, интерфейсы, возможность работы с дисплеями и беспроводную передачу данных.

Примером могут служить SOC-чипы для смартфонов, в которых почти весь функционал содержится в одном компоненте. Предоставить доступ к подобным решениям и для производителей индустриальных решений призваны smart-модули от SIMCom Wireless Solutions. Под smart-модулем подразумевается решение, работающее под управлением открытого ПО с достаточной степенью защиты (Android/Linux), большим количеством интерфейсов, мощным процессором, встроенными LPDDR и eMMC памятью и беспроводными интерфейсами для связи. Такое сочетание позволяет уменьшить стоимость компонентной базы, габаритные размеры, а также снизить сложность разработки, помогая сконцентрироваться на программной части, а не на аппаратной стороне.

Дмитрий Новинский
novinsky.d@mt-system.ru

Обзор

Внешне smart-решения напоминают знакомые многим модули сотовой связи, однако разница кроется именно в начинке. В отличие от обычных модулей, которые в большинстве случаев являются периферийными устройствами и служат лишь для передачи данных или голоса, smart-решения считаются основным компонентом системы.

Мощная начинка позволяет модулю выполнять широкий круг задач. Во-первых, они работают под управлением Android, а в ряде случаев — Linux. Процесс разработки на этих системах довольно стандартизирован и знаком тем, кто уже сталкивался с ними ранее. Во-вторых, это беспроводные интерфейсы: сотовая связь стандарта LTE, Wi-Fi, Bluetooth. В-третьих, это физические интерфейсы для дисплея, камер, а также UART, SPI, I²C, GPIO и USB. Столь богатый выбор периферии позволяет подключать дисплей (или даже несколько), кнопочные клавиатуры, камеры, SD-карты, USB-устройства, принтеры, датчики, динамики, NFC- и Ethernet-интерфейсы. В ряде случаев в модуль встраивается и контроллер заряда для мобильных применений, а также присутствует навигационный приемник для определения положения.

В зависимости от аппаратной начинки модули подразделяются на уровни: чем он выше, тем больше внутренних ресурсов, возможностей и интерфейсов доступно пользователям. Однако вместе с тем растет и стоимость таких решений. Актуальная линейка приведена в таблице. Рассмотрим каждое из устройств подробнее.

Модуль SIM8905E является решением начального уровня, при этом обладая до-

статочной начинкой для большинства задач. Это самый популярный и доступный smart-модуль в линейке. Он способен работать под управлением Android 5-8, также в ближайшее время ожидается новая версия с поддержкой актуальных ревизий ОС. На текущий момент, это единственное решение, поддерживающее также работу под управлением Linux (Yocto), доступна среда разработки для версий ядра 3.18 и 4.9. Наличие сразу двух поддерживаемых ОС позволяет применять этот модуль даже тем, у кого нет опыта в Android разработке, но есть опыт работы с Embedded Linux.

Отдельно следует отметить версию без сотовой части (SIM8005) — она позволяет использовать все возможности решения для применений, где достаточно остального функционала. К тому же, за счет общей среды разработки обе версии модуля, с LTE и без, могут быть использованы в общем дизайне и с одинаковой программной частью.

Если говорить о типовых применениях, где модуль успешно используется, это POS-терминалы для оплаты, СКУД, системы для вендинга (выбор товара на дисплее и/или клавиатуре, взаимодействие с автоматом, передача данных о транзакциях), медицинское оборудование, а также промышленные смартфоны и планшеты.

SIM8950E относится к smart-модулям среднего уровня. Первое, что бросается в глаза, это прирост в доступных ресурсах, что позволяет задействовать модуль в ресурсоемких приложениях — к примеру, обрабатывать видео Full HD при 60 кадрах/с. Также в нем доступна более новая версия Bluetooth Low Energy, скоростной USB (3.0), ну и конечно, поддержка больших разрешений дисплея и камер. Отличительная

особенность модуля — поддержка двух дисплеев одновременно, что может быть интересно для ряда применений.

Что касается программной части, модуль работает под управлением Android 7–9, ожидается новая ревизия с поддержкой большего количества камер и более актуальной версией ОС.

Сферы применения у данных модулей отчасти дублируются с SIM8905E, но расширяются за счет более мощной аппаратной начинки. Типовые применения:

- Кассы: один дисплей может быть обращен к кассиру, другой — к покупателю со списком позиций
- Рекламные и информационные стойки: информация может выводиться на дисплей с каждой стороны, обновление транслируемой информации — по воздуху.
- СКУД с распознаванием лиц.
- Терминалы оплаты на основе биометрических данных. Это могут быть как smart-кассы, так и вендинговые аппараты.
- Мультимедиа-системы для транспорта.
- Бортовые компьютеры.
- Системы на основе ИИ: помощь при вождении, контроль состояния водителя.

Самым продвинутым решением является SIM8970E. Это максимально производительный модуль в линейке, поддерживает наиболее актуальные стандарты связи и технологии. Работает он под управлением самых последних версий Android и предназначен для систем, где требуются существенные вычислительные мощности, к примеру обработка потока с нескольких камер одновременно (до четырех), или необходимы высокие скорости передачи данных (за счет поддержки шестой категории LTE).

Для этих модулей также доступны версии без сотовой части.

Следует отметить, что использование таких решений подразумевает наличие опыта Android- или Linux-разработки. Поэтому далее рассмотрим лишь общие моменты, касающиеся работы с SDK, компиляции образов, их заливки и отладки. В качестве примера будет описан процесс работы с самым массовым и доступным решением — SIM8905E.

Android

Первый этап при работе с SDK Android — получить его. Есть два способа: использовать репозиторий или скачать со специального FTP-сервера, предоставляемого производителем. В обоих случаях рекомендуется обратиться к специалистам технической поддержки дистрибьютора («МТ-Системс») для получения актуальных ссылок и данных для входа. Для работы с SDK рекомендуется иметь систему с Ubuntu x64 14.04.5 LTS (или более новую), Intel (R) Core (TM) i7-4790 CPU 3,6 ГГц, 16 Гбайт RAM, 500 Гбайт HDD. Сборка образа довольно ресурсоемкая задача, и в случае нехватки ресурсов процесс может попросту прерваться и завершиться ошибкой.

При скачивании с FTP все довольно просто, также отмечу, что SDK сразу поставляется с нужными патчами и готов для сборки.

В случае использования репозитория требуется следующая последовательность действий:

```
# apt-get install curl //устанавливаем curl
# curl http://commondatastorage.googleapis.com/
git-repo-downloads/repo //получаем repo
# apt-get install git //устанавливаем git
# git config --global user.email "xxx@example.com"
//задаем e-mail
# git config --global user.name "yyy" //и имя
# repo init .... //задаем репозиторий для син-
хронизации, конкретные ссылки запрашивайте
у тех поддержки
# repo sync //запускаем синхронизацию
...
# Syncing work tree: 100%<...> done //загрузка
завершена
```




После этого требуется скачать патч от SIMCom и распаковать его в папку со скачанным SDK.

Полученные исходники носят стандартный характер и будут знакомы Android-разработчикам. Собственно, это и есть первый этап разработ-

ки — настроить сборку под требуемую задачу. Как я говорил ранее, здесь мы не будем углубляться в процесс настройки, отмечу лишь, что SIMCom Wireless Solutions со своей стороны предоставляет ряд вспомогательных документов, описывающих процесс настройки: интеграция сторонних сенсоров для дисплея, тач-панелей, камер и настройка интерфейсов (GPIO, I²C, UART, SPI и т. д.). В дополнение к этому есть список рекомендованных компонентов, которые ранее были интегрированы в систему (дисплеи, тач-панели, датчики, акселерометры, USB-Ethernet микросхемы и т. д.).

Следующий этап — компиляция. Перед началом сборки требуется убедиться в наличии и при необходимости установить нужные библиотеки, JDK, обновить имеющиеся пакеты и поменять командную оболочку на bash. Полный список библиотек и последовательность действий приведены в документе «Compilation and Burning Guide».

Т а б л и ц а . Smart-модули SIMCom Wireless Solutions

Модуль	SIM8905E	SIM8950E	SIM8970
Внешний вид			
Уровень	Начальный	Средний	Продвинутый
Габариты, мм	40,5×40,5 LCC/LGA	44,1×45,6 LCC/LGA	
ОС	Android / Linux	Android	
Процессор	Четырехъядерный ARM Cortex-A7 1,1 ГГц 32-бит	Восьмиядерный ARM Cortex-A53 1,8 ГГц 64-бит	Восьмиядерный ARM V-8 2 ГГц 64-бит
GPU	Andreno 304	Andreno 506	Andreno 610
Память	LPDDR 1/2* Гбайт eMMC 8/16* Гбайт	LPDDR 2/3*/4* Гбайт eMMC 16/32*/64* Гбайт	LPDDR 4 Гбайт eMMC 64 Гбайт
LTE	Cat.4		Cat.6
3G/GSM	+		+
Скорость передачи в сотовой сети, Мбит/с	до 150/50		до 350/50
Версия модуля без сотовой части	SIM8005	SIM8050	SIM8070
Контроллер заряда Li-ion батарей	Встроенный	Внешняя м/сх с поддержкой QC3.0	Встроенный с поддержкой QC3.0
Дисплей	4-lane MIPI_DSI 1280×800 60fps	Два 4-lane MIPI_DSI 1920×1200 60fps	4-lane MIPI_DSI 2520×1080 60 fps. Второй дисплей через USB Type-C
Камеры	Основная: 2-lane MIPI_CSI, 8MP. Второстепенная: 1-lane MIPI_CSI, 5MP	Основная: 4-lane MIPI_CSI, 21MP. Второстепенная: 2-lane MIPI_CSI, 8MP	До 4 камер MIPI_CSI с разрешением до 25 MP
Обработка видео	Кодирование (720p 30fps): H.264, H.263, VP8, MPEG4. Декодирование (1080p 30 fps): H.265, H.264, H.263, VP8, MPEG4	Кодирование (1080p 60fps): H.264, H.265, and VP8. Декодирование (1080p 60fps): H.264, H.265, VP8, VP9	Кодирование (4K 30 fps): HEVC/H.264/VP8/MPEG4. Декодирование (4K 30 fps): 8-бит H.264, VP8, VP9. 10-бит HEVC
Аудио интерфейс	Аналоговый: два входа, три выхода	Цифровой: I ² S. Аналоговый: три входа, четыре выхода	Цифровой: четыре I ² S. Аналоговый: три входа и выхода
GNSS	GPS / ГЛОНАСС / Beidou		
Bluetooth	V2.1+EDR, V3.0+HS, V4.0 LE	V4.2 LE	V5.x LE
Сим-карты	Две сим-карты (Dual SIM Dual Standby)		
USB	2.0	3.0/2.0; Type-C; OTG	3.1/2.0; Type-C; OTG
Wi-Fi	2.4+5 ГГц 802.11 a/b/g/n	2.4+5 ГГц 802.11 a/b/g/n/ac	
Обновление	USB, FOTA, TF Card		
Напряжение питания, В	3,4–4,4		
Интерфейсы	GPIO, ADC, LDO выход, UART, SPI, I2C, SDIO		
Температурный диапазон	Нормальный: -35...+75 °C; расширенный: -40...+85 °C		

Примечание. *Опционально.

Последовательность для сборки следующая:

```
//требуется перейти в папку с патчем (по умолчанию - «simcom») и запустить скрипт
# sh sim8905_evb-copy.sh
//возвращаемся в папку с исходниками (т. е. на уровень выше) и выполняем
# source build/envsetup.sh
# lunch msm8909-userdebug //выбираем сборку: «-user» собирает сборку без доп. возможностей для отладки (для пользователя), «-userdebug» - с отладкой и полным доступом (режим разработчика). Также можно запустить просто «lunch» и указать номер желаемой сборки из списка.
# make -j4 //команда на старт сборки, 4 — количество ядер CPU
```

Если сборка прошла без ошибок, в директории /android/out/target/product/msm8909/ будут сгенерированы файлы. Маркером того, что все прошло успешно, будет наличие следующих:

```
boot.img emmc_appsboot.mbn persist.img recovery.img system.img userdata.img
```

По сути, на этом процесс компиляции закончен. Также есть возможность сборки каждой из частей по отдельности, для этого, в make указывается требуемая часть образа.

Следует отметить, что для исходников, предоставляемых производителем (которые доступны на FTP-сервере), есть готовые образы на их основе (уже скомпилированные), их можно использовать для общей проверки

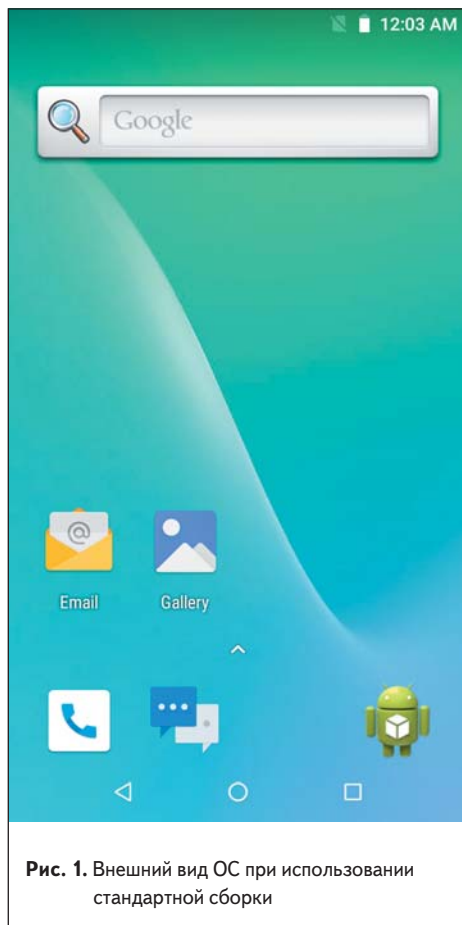


Рис. 1. Внешний вид ОС при использовании стандартной сборки

работоспособности или ознакомления с отладочным средством и работой Android непосредственно на модуле.

Для заливки полученного образа потребуется поместить содержимое /android/out/target/product/msm8909/ и /android/Non-HLOS/out/msm8909/ в общую директорию.

Из-за особенностей ПО процесс обновления рекомендуется проводить под управлением ОС Windows, для корректной работы утилит.

Для обновления требуется подключить модуль по USB, используется порт загрузки (9008). Чтобы он появился, можно воспользоваться одним из двух способов: замкнуть вывод BOOT с выходом LDO перед запуском или подать команду через утилиту ADB (adb reboot edl). Необходимые драйверы и утилиты доступны для скачивания с FTP-сервера SIMCom Wireless Solutions.

Когда все приготовления закончены, необходимо запустить утилиту QFIL (является частью пакета QPST) и задать:

```
Select Port — выбрать порт QDLoader 9008
Select Build Type — Flat Build
Select Programmer — указать файл «prog_emmc_firehose_8909_dds.mbn» из папки с утилитой
Load XML — выбрать последовательно файлы «rawprogram0.xml», «patch0.xml» (после они отобразятся в полях RawProgram и Patch соответственно).
Storage Type — выбрать «eMMC» (нижний правый угол утилиты)
```

После этого, нажать Download и дождаться окончания загрузки: сообщения Download Succeed, Finish Download.

После этого модуль готов к работе. Теперь потребуется подать питание и нажать кнопку включения (PWRKEY) не менее чем на 3 с. По умолчанию один из портов UART сконфигурирован как Debug-порт, поэтому в случае критических ошибок, мешающих старту, туда будет выдаваться отладочная информация.

В случае успеха запустится среда Android — внешний вид представлен на рис. 1. Далее, отладка и работа с модулем аналогична другим устройствам с этой ОС.

Linux

Как уже упоминалось, у модуля SIM8905E также есть возможность работать под управлением Yocto Linux. Для примера рассмотрим работу с ядром 3.18, но и для более нового процесс не будет сильно отличаться.

Исходники, документация и предварительно скомпилированные образы доступны на FTP-сервере SIMCom Wireless Solutions. Сборка рекомендуется на базе Ubuntu 14.04 и выше с установленными gcc/g++ 5.4 и Python 2.7. Также понадобится установить ряд библиотек, если они не присутствуют в системе. Подробнее необходимые действия описаны в документе «Linux Source Building».

Как и в случае с Android, если требуется, изменения вносятся в исходники, это тоже первый этап разработки.

Для создания образов в директории с SDK необходимо подать следующие команды:

```
# cd apps_proc/poky/
# source build/conf/set_bb_env.sh //в ответ на команду должны выдаться возможные варианты для сборки образов, там должны обязательно присутствовать «build-sim8905-image» и «clean-image»
# build-sim8905-image //если все успешно, будет выдано сообщение «NOTE:<...> All Succeeded»
//далее требуется перейти в директорию /apps_proc/poky/build/ и запустить скрипт
# source update.sh
```

После выполнения последнего скрипта, образ окажется в директории /sim8905-linux-1_0_2-image. Далее процесс заливки аналогичен тому, как это делалось для образов Android, обновлять также рекомендуется с помощью ОС Windows.

Для проверки базового функционала можно использовать утилиту ADB. Модуль включается так же, как это описывалось ранее, с той лишь разницей, что при загрузке появится логотип Linux (пингвин Tux) — после его появления модуль уже находится в рабочем режиме. Для работы потребуется подключить модуль по USB, в системе должно появиться устройство Android Debug Interface. Поддача в командной строке команды adb shell предоставит доступ к консоли Linux.

По умолчанию, при наличии сим-карты и антенны модуль автоматически соединяется с сетью (в принципе, так же, как и при использовании стандартного образа Android). Для проверки можно подать соответствующие команды в модемный порт (AT+CPIN?, AT+CREG?), но в целом это способ именно для отладки.

Для проверки работы передачи данных в модуле есть утилита, располагающаяся по адресу:

```
#!/usr/tests/dsi_netctrl_test
//выбираем последовательно пункты Create call handle, Automatic, IPv4v6, Automatic, Net UP, Call Automatic
//в случае успеха будут выданы сообщения «DSI_EVT_NET_IS_CONN» и «DSI_EVT_NET_RECONFIGURED»
# ifconfig //проверяем что появился адаптер (нужный нам - rmnet_data0)
# route add default dev rmnet_data0 //делаем устройством по умолчанию
# ping<адрес> //проверка
```

Этот способ, конечно же, предназначен именно для проверки, для реальных применений можно настроить автоподнятие соединения или управлять им уже из приложения (приложение), которое должен выполнять модуль. Также утилиту можно взять за основу, если требуется поднимать сеть с какими-то специфичными параметрами или для ознакомления, как это делать в принципе.

Для отработки функционала навигационного приемника потребуется отдельная утилита. Для ее установки потребуется запустить push-gps-tool.bat из архива linux-push-gps (расположен в папке /Documents/ на FTP-сервере).

После того как программа будет записана в систему, ее можно вызвать с помощью:

```
#!/system/bin/nmea_test_app -h //показать доступные параметры
```



```
#./system/bin/nmea_test_app -t 600 -m NMEA_
MASK_ALL -p 1 //к примеру, с этими параме-
трами будет установлен таймаут в 600 секунд,
в течении которых будут выдаваться все NMEA
сообщения с частотой выдачи в 1 Гц
```

Аналогичные утилиты есть и для Bluetooth (btapp) и Wi-Fi (wpa_cli), которые также предоставляют большое количество возможных действий (сканирование, подключение к устройствам, передача данных и т. д.). Для разработчика же они могут служить наглядным алгоритмом действий благодаря тому, что там все представлено пошагово. Подробное описание работы со всеми этими утилитами приводится в Linux Basic Features Guide.

Все, о чем шла речь выше, относилось именно к утилитам для командной строки. Для написания более сложных программ, в том числе с графическим интерфейсом, предлагается инструмент QT, знакомый Linux-разработчикам. Установка QT-Creator, а также интеграция библиотек в модуль описана в Linux Series Qt building and Installation. Для примера SIMCom предоставляет несколько программ, построенных на нем, например видеопроигрыватель.

Отладочное средство

С учетом большого количества возможных решений данного типа возникает вопрос о первых шагах в их освоении. Как и для всех остальных модулей, SIMCom Wireless Solutions предлагает отладочные средства для знакомства с модулем и отлаживания функционала. В качестве примера на рис. 2 приведен внешний вид отладки для модуля SIM8905E.

Отладка для smart-решений состоит из двух частей: материнской платы и мезонина с установленным модулем. На каждой из отладок задействованы почти все доступные интерфейсы: разъемы для гарнитуры и наушников, динамик, два сим-держателя, аналоговые кнопки, основная и вспомогательная камеры, разъем для SD-карты, дисплей с тачскрином, датчики освещенности и приближения, гироскоп, разъем для батареи, вибромотор. В дополнение к этому контакты модуля выведены на металлизированные отверстия для прямого доступа (они находятся под мезонином с модулем).

В целом, внешний вид каждого из отладочных средств похож, но, разумеется, есть и отличия. Для модулей с поддержкой USB Type-C на плате имеется соответствующий разъем. Для модуля SIM8950E существует версия отладки с двумя дисплеями, а для SIM8970E предусмотрена установка нескольких камер.

Кроме плат, в комплект входит блок питания, антенны и кабели. Все это позволяет оперативно запустить модуль без необходимости разбираться в тонкостях схемотехники и приступить к освоению программной части.

Заключение

В целом smart-решения предназначены лишь для определенного круга M2M-устройств. В тех случаях, где требуется компактное решение с возможностью отображения информации на экране, обработки изображений с камер, наличием большого количества доступных коммуникационных интерфейсов, они и находят свое применение. Такое решение «все-в-одном» позволяет значительно упростить схемотехнику устройства, поскольку данный модуль уже содержит основные блоки,



Рис. 2. Отладочное средство для модуля SIM8905E

требуя лишь подключения желаемой периферии. Разработчикам данные модули предлагают знакомую среду разработки — Linux или Android, а значит, есть и большое количество наработок и опыт. Такие решения становятся все более доступными и используются во многих приложениях, приходя на смену традиционной связке «хост + беспроводной модуль». ■