



SIM8500系列 外设应用文档

智能模组

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话: 86-21-31575100

技术支持邮箱: support@simcom.com

官网: www.simcom.com

名称:	SIM8500系列 外设应用文档
版本:	1.00
日期:	2022.03.15
状态:	已发布

版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<https://cn.simcom.com/>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2021，保留一切权利。

关于文档

版本历史

版本	日期	作者	备注
1.00	2022.03.15	葛正阳	第一版

适用范围

本文档适用于 SIMCom SIM8500E 系列，SIM8500CE 系列。

目录

关于文档.....	3
版本历史.....	3
适用范围.....	3
目录.....	4
1 介绍.....	5
1.1 本文目的.....	5
1.2 参考文档.....	5
1.3 术语和缩写.....	5
2 端口功能描述.....	6
3 GPIO 使用方法.....	7
4 I2C 使用方法.....	9
5 UART 使用方法.....	11
6 SPI 使用方法.....	12
7 Pin mux 使用方法.....	14

1 介绍

1.1 本文目的

基于 SIM8500 平台，本文讲述了 I2C、SPI、UART、GPIO 外设端口及驱动适配指南。参考此应用文档，开发者可以很快理解并快速开发相关业务。

1.2 参考文档

[1] MU8500x_硬件设计手册

1.3 术语和缩写

表 1: 术语和缩写

英文缩写	解释与描述
SPI	Serial Peripheral Interface 串行外设接口总线
I2C	Inter-Integrated Circuit 内部集成总线
UART	Universal Asynchronous Receiver/Transmitter 通用串行总线
GPIO	General-purpose input/output 通用输入输出端口
CPOL	Clock Polarity 时钟极性
CPHA	Clock Phase 时钟相位

2 端口功能描述

SIM8500 系列智能模组采用了紫光展锐 SL8541E SOC，该芯片有 3 组 SPI、4 组 UART、4 组 I2C。SPI 仅支持主设备模式，指纹识别建议 SPI0。SPI 最大速率 48M，I2C 最大速率 3.4M，UART 的最大频率 3.25Mbps。SL8541E 虽然 UART 有 4 组端口，但只有 2 个 UART Core 在 AP 侧，在 kernel 中只能用 2 组端口，默认设计只使能了 DBG_UART 和 UART0。其中 DBG_UART 专用于系统调试打印 log，因此实际可单独使用的 UART 只有 1 个。

模块引脚名	模块引脚号	复用功能	SOC 点位
UART0_TXD	PIN_34	UART0_TXD	U0TXD
UART0_RXD	PIN_35	UART0_RXD	U0RXD
UART0_CTS	PIN_36	UART0_CTS	U0CTS
UART0_RTS	PIN_37	UART0_RTS	U0RTS
DBG_RXD	PIN_93	UART1_RXD	U1RXD
DBG_TXD	PIN_94	UART1_TXD	U1TXD
UART2_RXD	PIN_153	UART2_RXD	U2RXD
UART2_TXD	PIN_154	UART2_TXD	U2TXD
GPIO_10	PIN_104	UART3_TXD	RFCTL19
GPIO_11	PIN_103	UART3_RXD	RFCTL20
TP_I2C_SCL	PIN_47	I2C3_SCL	SCL3
TP_I2C_SDA	PIN_48	I2C3_SDA	SDA3
CAM_I2C_SCL	PIN_83	I2C0_SCL	SCL0
CAM_I2C_SDA	PIN_84	I2C0_SDA	SDA0
SENSOR_I2C_SCL	PIN_91	I2C2_SCL	SCL2
SENSOR_I2C_SDA	PIN_92	I2C2_SDA	SDA2
I2C4_SCL	PIN_168	I2C4_SCL	SIMCLK2
I2C4_SDA	PIN_167	I2C4_SDA	SIMDATA2
I2C1_SCL	PIN_166	I2C1_SCL	SCL1
I2C1_SDA	PIN_205	I2C1_SDA	SDA1
GPIO_134	PIN_102	SPI1_CSN	SD2_D3
GPIO_135	PIN_101	SPI1_DO	SD2_D0
GPIO_136	PIN_100	SPI1_DI	SD2_CMD
GPIO_139	PIN_99	SPI1_CLK	SD2_CLK
GPIO_52	PIN_107	SPI2_CSN	SPI2_CSN
GPIO_53	PIN_108	SPI2_DO	SPI2_DO
GPIO_54	PIN_109	SPI2_DI	SPI2_DI
GPIO_55	PIN_110	SPI2_CLK	SPI2_CLK
GPIO_90	PIN_117	SPI0_CSN	SPI0_CSN
GPIO_91	PIN_119	SPI0_DO	SPI0_DO
GPIO_92	PIN_118	SPI0_DI	SPI0_DI
GPIO_93	PIN_116	SPI0_CLK	SPI0_CLK

3 GPIO 使用方法

GPIO 使用方法参考 Linux Kernel Document,

<https://www.kernel.org/doc/html/v4.19/driver-api/gpio/consumer.html>。

部分代码示例:

1、在 Kernel 中使用 GPIO

首先在设备树中引用 SOC 的 GPIO

```
eth_switch:ethswitch {
    compatible = "simcom,ethswitch";
    signal-gpio = <&ap_gpio 91 0>;
};
```

然后在代码中请求 GPIO

```
sw->gpiod = devm_gpiod_get_optional(dev, "signal", GPIOD_IN);
if(IS_ERR(sw->gpiod)) {
    dev_err(dev, "failed to get gpios\n");
    return PTR_ERR(sw->gpiod);
}
```

2、在 uboot 中使用 GPIO

输出首先需要 request gpio, 然后设置输出电平高低 (1/0), 53 为 GPIO_53。

```
sprd_gpio_request(NULL, 53);
sprd_gpio_direction_output(NULL, 53, 1);
```

输入输出首先需要 request gpio, 然后设置输入模式, 再读取。

```
sprd_gpio_request(NULL, CONFIG_LCM_GPIO_ID);
sprd_gpio_direction_input(NULL, CONFIG_LCM_GPIO_ID);
value = sprd_gpio_get(NULL, CONFIG_LCM_GPIO_ID);
```

调试技巧:

如何查看 GPIO 状态?

```
cat /sys/kernel/debug/gpio
```

如何查看中断状态?

```
cat /proc/interrupts
```

如何快速控制 GPIO?

```
ls /sys/class/gpio #查看 gpiochip
```

gpiochip160 为 SL8541E SOC 的 GPIO 控制器, 因此所有申请到的 GPIO 号都会有+160 的偏移值

以设置 GPIO_20 为例:

首先挂载 gpio20,可能会出现错误并返回错误码, 这种情况需要查看 GPIO 占用情况, 后面会讲到。

```
echo 180 > /sys/class/gpio
```

```
echo (out/in) > /sys/class/gpio180/direction #设置 gpio 输入或输出
```

```
(echo/cat) /sys/class/gpio180/value #读取或写入 gpio 值
```

如何排查 GPIO 被占用情况?

全开 kernel log

```
echo "8" > /proc/sys/kernel/printk
```

打开 gpiolib 动态调试

```
echo 'file gpiolib.c +p' > /sys/kernel/debug/dynamic_debug/control
```

通过 GPIO 号申请 GPIO

```
echo 93 > /sys/class/gpio/export
```

查看 kernel 日志

```
dmesg | grep gpiod_request
```

或者

```
cat /proc/kmsg | grep gpiod_request
```

输出结果如下:

```
[50143.858326] c3 gpio-235 (Volume Up Key): gpiod_request: status -16
```

那么我们只需要到内核代码中搜索 Volume Up Key 这个关键字, 就能找到冲突。

4 I2C 使用方法

在 uboot 中使用 i2c 参考示例： bsp/bootloader/u-boot15/drivers/power/battery/fan54015.c

首先需要初始化 i2c 控制器，读写 i2c 系统已经提供了相关接口：i2c_red_reg()和 i2c_reg_write()。

```
uint16_t sprd_fan54015_i2c_init(void)
{
    i2c_set_bus_num(CONFIG_SPRDCHG_I2C_BUS);
    i2c_init(I2C_SPEED, SLAVE_ADDR);
    return 0;
}

static int fan54015_write_reg(int reg, u8 val)
{
    i2c_reg_write(SLAVE_ADDR, reg, val);
    return 0;
}

int fan54015_read_reg(int reg, u8 *value)
{
    int ret;
    ret = i2c_reg_read(SLAVE_ADDR, reg);
    if (ret < 0) {
        printf("%s reg(0x%x), ret(%d)\n", __func__, reg, ret);
        return ret;
    }

    ret &= 0xff;
    *value = ret;
    printf("#####fan54015readreg reg = %d value = %d/%x\n", reg, ret, ret);
    return 0;
}
```

Kernel 中已经配置好 i2c 控制器驱动与设备树，所有 i2c 引用节点如下：

bsp/kernel/kernel4.14/arch/arm64/boot/dts/sprd/sharkle.dtsi

```
19     aliases {
20         serial0 = &uart0;
21         serial1 = &uart1;
22         i2c0 = &i2c0;
23         i2c1 = &i2c1;
24         i2c2 = &i2c2;
25         i2c3 = &i2c3;
26         i2c4 = &i2c4;
27         spi0 = &spi0;
28         spi1 = &spi1;
29         spi2 = &spi2;
30     };
31
```

设备树引用配置参考：

bsp/kernel/kernel4.14/Documentation/devicetree/bindings/i2c/i2c-sprd.txt

clock-frequency 属性在主节点中未配置，需要在引用节点中配置，最大支持 3.4M (fast-mode)，默认情况请配置成 400K 或 100K。

```
&i2c1 {
+   status = "okay";
+   clock-frequency = <400000>;

+   sensor_main2: sensor-main2@6C {
+       compatible = "sprd,sensor-main2";
+       reg = <0x6C>;
+       clock-names = "clk_src", "sensor_eb",
+       +       "clk_96m", "clk_76m8",
+       +       "clk_48m", "clk_26m";
+       clocks = <&mm_clk CLK_SENSOR1>, <&mm_gate CLK_SENSOR1_EB>,
+       +       <&p11 CLK_TWPLL_96M>, <&p11 CLK_TWPLL_76M8>,
+       +       <&p11 CLK_TWPLL_48M>, <&ext_26m>;
+       vddio-supply = <&vddcamio>;
+       vddcama-supply = <&vddcama>;
+       vddcamd-supply = <&vddcamd>;
+       vddcammot-supply = <&vddcammot>;
+       +       dvdd-gpios = <&ap_gpio 32 0>;
+       sprd,phyid = <0>;
+       csi = <&csi0>;
+       reset-gpios = <&ap_gpio 41 0>;
+       power-down-gpios = <&ap_gpio 40 0>;
+   };
+ };
```

调试技巧：

如何测试 I2C 接口？

SL8541E 的 i2c 都已经给出，直接引用就能够正常使用，在使用时需要注意 I2C 速率需要配置成芯片时钟所能够支持的速率，如 100K 或 400K。如果需要测试工具，可联系 SIMCOM 工程师，可以提供 I2C 测试工具，可以读取 I2C 所有设备地址，验证 I2C 波形。

调用 i2c_transfer 返回值为 -6？

返回值为 -6 表示为 NACK 错误，即对方设备无应答响应，这种情况一般为外设的问题，常见的有以下几种情况：

- 1、I2C 地址错误，解决方法是测量 I2C 波形，确认是否 I2C 设备地址错误；
- 2、I2C slave 设备不处于正常工作状态，比如未给电，错误的上电时序等；
- 3、时序不符合 I2C slave 设备所要求也会产生 NACK 信号。

5 UART 使用方法

Kernel 中已经配置好 uart 控制器驱动与设备树，所有 uart 引用节点如下：

```
19  aliases {
20      → serial0 = &uart0;
21      → serial1 = &uart1;
22      → i2c0 = &i2c0;
23      → i2c1 = &i2c1;
24      → i2c2 = &i2c2;
25      → i2c3 = &i2c3;
26      → i2c4 = &i2c4;
27      → spi0 = &spi0;
28      → spi1 = &spi1;
29      → spi2 = &spi2;
30  };
31
```

uart0 为 debug uart，uart1 为用户可用的 uart。uart 默认使能，kernel 无需配置。波特率、流控由应用配置，kernel 无需关注。

调试技巧：

如何测试 UART 接口？

SL8541E 的 UART 都已经给出，直接启用即可。在使用时需要注意 UART 速率不能超过芯片时钟所能够支持的最大速率 3.25M。最简单的测试方法是短接 UART 的 TX 和 RX，开两个 adb 调试窗口，一个 cat /dev 目录下的串口节点，一个 echo 测试数据到串口节点，进行回环测试。如果需要测试工具，可联系 SIMCOM 工程师，可以提供 UART 测试工具，可以测试收发二进制数据流，也提供回环测试工具。

6 SPI 使用方法

SPI 最大速率为 48M，kernel 中已经配置好 spi 控制器驱动与设备树，所有 spi 引用节点如下：

bsp/kernel/kernel4.14/arch/arm64/boot/dts/sprd/sharkle.dtsi

```

18
19 → aliases {
20 →     i2c0 = &i2c0;
21 →     i2c1 = &i2c1;
22 →     i2c2 = &i2c2;
23 →     i2c3 = &i2c3;
24 →     i2c4 = &i2c4;
25 →     i2c5 = &i2c5;
26 →     i2c6 = &i2c6;
27 →     i2c7 = &i2c7;
28 →     spi0 = &spi0;
29 →     spi1 = &spi1;
30 →     spi2 = &spi2;
31 →     spi3 = &spi3;
32 →     spi4 = &adi_bus;
33 → };
34

```

Linux 提供了一个功能有限的 SPI 用户接口，如果不需要用到 IRQ 或者其他内核驱动接口，可以考虑使用接口 spidev 编写用户层程序控制 SPI 设备。该设备节点的路径为：/dev/spidevX.0

spidev 对应的驱动代码：kernel/drivers/spi/spidev.c

内核 defconfig 需要使能 SPI_SPIDEV：

bsp/kernel/kernel4.14/arch/arm/configs/sprd_sharkle_defconfig

CONFIG_SPI_SPIDEV = y

设备树配置参考如下：

```

spidev@0 {
+ compatible = "spidev";
+ spi-max-frequency = <5000000>;
+ reg = <0>;
};

```

详细使用说明请参考文档 [spidev](#) 。

部分代码示例：

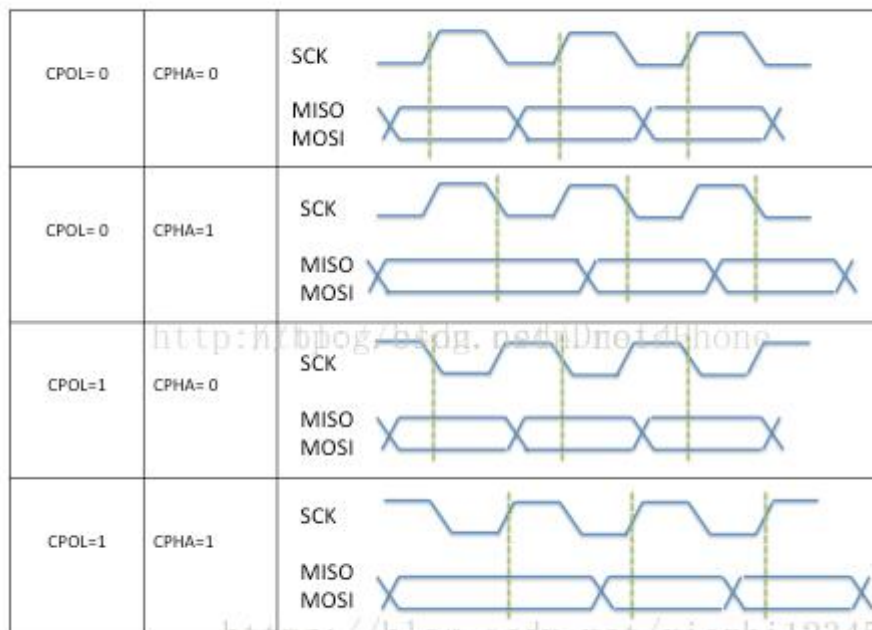
```
flash@1 {
    #address-cells = <1>;
    #size-cells = <1>;
    compatible = "sst25wf040b", "jedec,spi-nor";
    spi-cpol; // 增加该属性相当于 CPOL=1
    spi-cpha; // 增加该属性相当于 CPHA=1
    reg = <1>;
    spi-max-frequency = <1000000>;
};
```

spi 4 种模式配置：

- 1、CPOL=0，CPHA=0 模式为 0
- 2、CPOL=0，CPHA=1 模式为 1
- 3、CPOL=1，CPHA=0 模式为 2
- 4、CPOL=1，CPHA=1 模式为 3

CPOL：表示时钟信号的初始电平的状态，0 为低电平，1 为高电平。

CPHA：表示在哪个时钟沿采样，0 为第一个时钟沿采样，1 为第二个时钟沿采样。



调试技巧：

如何测试 SPI 接口？

SL8541E 的 SPI 都已经给出，直接启用即可。在使用时需要注意 SPI 速率不能超过芯片时钟所能够支持的最大速率 48M。如果需要测试工具，可联系 SIMCOM 工程师，可以提供 SPI 测试工具进行回环测试，验证波形。

7 Pin mux 使用方法

展锐 SL8541E 平台没有提供 kernel 侧的 pinctrl 子系统，所有的端口复用都是由 uboot 进行设置。并且 SIM8500E 系列和 SIM8500CE 系列所使用的配置文件路径是不同的。

SIM8500CE 系列 pin mux 配置文件路径：

bsp/bootloader/u-boot15/board/spreadtrum/sl8541e_1h10_32b/pinmap-sl8541e.c

SIM8500E 系列 pin mux 配置文件路径：

bsp/bootloader/u-boot15/board/spreadtrum/sl8541e_1h10wifi5g_32b/pinmap-sl8541e.c

部分配置示例：

```
34 {REG_PIN_CTRL2,0x60400300}, //uart1->Digital IO; SPI0->Normal Mode; SPI2->Normal Mode; sim2->AP SIM0; sim1->PUBCP SIM1; sim0->PUBCP SIM0;
35 //0110 0000 0100 0000 0000 0000 0000 0000
36 {REG_PIN_CTRL3,0x00000000}, //wdrst->AP watch dog; iis1->AP IIS0;
37 {REG_PIN_CTRL4,0x00000000}, //复用功能
38 {REG_PIN_CTRL5,0x00000000}, //New Function, all set to default value 0
39
40 {REG_PIN_IIS1DI, ..... BITS_PIN_AF(2)}, //引脚标号
41 {REG_MISC_PIN_IIS1DI, ..... BITS_PIN_DS(1)|BIT_PIN_NULL|BIT_PIN_WPD|BIT_PIN_SLP_AP|BIT_PIN_SLP_WPD|BIT_PIN_SLP_IE}, //FTID_INT?
42 {REG_PIN_IIS1DO, ..... BITS_PIN_AF(3)},
43 {REG_MISC_PIN_IIS1DO, ..... BITS_PIN_DS(1)|BIT_PIN_NULL|BIT_PIN_NUL|BIT_PIN_SLP_AP|BIT_PIN_SLP_NUL|BIT_PIN_SLP_OE}, //FTID_RSTN
44 {REG_PIN_IIS1CLK, ..... BITS_PIN_AF(3)},
45 {REG_MISC_PIN_IIS1CLK, ..... BITS_PIN_DS(1)|BIT_PIN_NULL|BIT_PIN_WPD|BIT_PIN_SLP_AP|BIT_PIN_SLP_WPD|BIT_PIN_SLP_Z}, //NC
46 {REG_PIN_IIS1LRCK, ..... BITS_PIN_AF(3)},
```

调试技巧：

如何配置 UART 复用端口？

SL8541E 的 UART 虽然有 4 个端口，但是实际上供 AP 侧使用的控制器只有 2 个，其中一个被用作 debug 串口，剩下能分配给用户使用的只有 1 个，因此不管怎么配置复用也不会平白无故多出可用的 UART 端口，并且现有的硬件设计已经引出了唯一可用的 UART 端口，因此完全没有必要再将 UART 端口调换位置，这是徒劳。

端口复用功能一览表：

Ball Name	Power	Pull up	Pull down	Function1	Function2	Function3	Function4
RFSDA0	VIO1V8	1.8V,4.7K/20K	50K	RFSDA0		DBG_BUS20(G0)	GPIO1
RFSCK0	VIO1V8	1.8V,4.7K/20K	50K	RFSCK0		DBG_BUS21(G0)	GPIO2
RFSEN0	VIO1V8	1.8V,4.7K/20K	50K	RFSEN0		DBG_BUS22(G0)	GPIO3
LVDSRF0_ADCON	VIO1V8	1.8V,4.7K/20K	50K	RF_LVDS0_ADC_ON	SE_GPIO9	DBG_BUS29(G0)	GPIO4
LVDSRF0_DACON	VIO1V8	1.8V,4.7K/20K	50K	RF_LVDS0_DAC_ON	SE_GPIO10	DBG_BUS30(G0)	GPIO5
RFCTL16	VIO1V8	1.8V,4.7K/20K	50K	RFCTL16	PWMC(G1)	DBG_BUS15(G0)	GPIO7
RFCTL17	VIO1V8	1.8V,4.7K/20K	50K	RFCTL17	PPS(G0)	DBG_BUS16(G0)	GPIO8
GPIO9	VIO1V8	1.8V,4.7K/20K	50K		BUA_SIM2_DET	DBG_BUS17(G0)	GPIO9
RFCTL19	VIO1V8	1.8V,4.7K/20K	50K	RFCTL19	U3TXD	DBG_BUS18(G0)	GPIO10
RFCTL20	VIO1V8	1.8V,4.7K/20K	50K	RFCTL20	U3RXD	DBG_BUS19(G0)	GPIO11
RFCTL0	VIO1V8	1.8V,4.7K/20K	50K	RFCTL0		DBG_BUS0(G0)	GPIO19
RFCTL1	VIO1V8	1.8V,4.7K/20K	50K	RFCTL1		DBG_BUS1(G0)	GPIO20

RFCTL2	VIO1V8	1.8V,4.7K/20K	50K	RFCTL2		DBG_BUS2(G0)	GPIO21
RFCTL3	VIO1V8	1.8V,4.7K/20K	50K	RFCTL3		DBG_BUS3(G0)	GPIO22
RFCTL4	VIO1V8	1.8V,4.7K/20K	50K	RFCTL4		DBG_BUS4(G0)	GPIO23
RFCTL5	VIO1V8	1.8V,4.7K/20K	50K	RFCTL5		DBG_BUS5(G0)	GPIO24
RFCTL6	VIO1V8	1.8V,4.7K/20K	50K	RFCTL6		DBG_BUS6(G0)	GPIO25
RFCTL7	VIO1V8	1.8V,4.7K/20K	50K	RFCTL7		DBG_BUS7(G0)	GPIO26
RFCTL8	VIO1V8	1.8V,4.7K/20K	50K	RFCTL8		DBG_BUS8(G0)	GPIO27
RFCTL9	VIO1V8	1.8V,4.7K/20K	50K	RFCTL9		DBG_BUS9(G0)	GPIO28
RFCTL10	VIO1V8	1.8V,4.7K/20K	50K	RFCTL10		DBG_BUS10(G0)	GPIO29
RFCTL11	VIO1V8	1.8V,4.7K/20K	50K	RFCTL11		DBG_BUS11(G0)	GPIO30
GPIO31	VIO1V8	1.8V,4.7K/20K	50K		BUA_SIM1_DET	DBG_BUS12(G0)	GPIO31
GPIO32	VIO1V8	1.8V,4.7K/20K	50K		PWMA	DBG_BUS13(G0)	GPIO32
GPIO33	VIO1V8	1.8V,4.7K/20K	50K		PWMB(G1)	DBG_BUS14(G0)	GPIO33
RFFE0_SCK	VIO1V8	1.8V,4.7K/20K	50K	RFFE0_SCK		DBG_BUS24(G0)	GPIO36
RFFE0_SDA	VIO1V8	1.8V,4.7K/20K	50K	RFFE0_SDA		DBG_BUS25(G0)	GPIO37
CMPD2	VCAM	1.8V,4.7K/20K	50K	CMPD2		DBG_BUS18(G1)	GPIO40
CMRST2	VCAM	1.8V,4.7K/20K	50K	CMRST2		DBG_BUS19(G1)	GPIO41
CMMCLK0	VCAM	1.8V,4.7K/20K	50K	CMMCLK0	CLK_AUX2	DBG_BUS20(G1)	GPIO42
CMMCLK1	VCAM	1.8V,4.7K/20K	50K	CMMCLK1		DBG_BUS21(G1)	GPIO43
CMRST0	VCAM	1.8V,4.7K/20K	50K	CMRST0		DBG_BUS22(G1)	GPIO44
CMRST1	VCAM	1.8V,4.7K/20K	50K	CMRST1		DBG_BUS23(G1)	GPIO45
CMPD0	VCAM	1.8V,4.7K/20K	50K	CMPD0		DBG_BUS24(G1)	GPIO46
CMPD1	VCAM	1.8V,4.7K/20K	50K	CMPD1		DBG_BUS25(G1)	GPIO47
SCL0	VCAM	1.8V,1.8K/20K	50K	SCL0		DBG_BUS26(G1)	GPIO48
SDA0	VCAM	1.8V,1.8K/20K	50K	SDA0		DBG_BUS27(G1)	GPIO49
LCM_RSTN	VIO1V8	1.8V,4.7K/20K	50K	LCM_RSTN			GPIO50
LCM_FMARK	VIO1V8	1.8V,4.7K/20K	50K	DSL_TE			GPIO51
SPI2_CSN	VIO1V8	1.8V,4.7K/20K	50K	SPI2_CSN		CM4_GPIO5	GPIO52
SPI2_DO	VIO1V8	1.8V,4.7K/20K	50K	SPI2_DO		CM4_GPIO0	GPIO53
SPI2_DI	VIO1V8	1.8V,4.7K/20K	50K	SPI2_DI		CM4_GPIO1	GPIO54
SPI2_CLK	VIO1V8	1.8V,4.7K/20K	50K	SPI2_CLK		CM4_GPIO2	GPIO55
U0TXD	VIO1V8	1.8V,4.7K/20K	50K	U0TXD	EXT_XTL_EN2	DBG_BUS10(G1)	GPIO60
U0RXD	VIO1V8	1.8V,4.7K/20K	50K	U0RXD	EXT_XTL_EN3	DBG_BUS11(G1)	GPIO61
U0CTS	VIO1V8	1.8V,4.7K/20K	50K	U0CTS	PWMC(G0)	DBG_BUS12(G1)	GPIO62
U0RTS	VIO1V8	1.8V,4.7K/20K	50K	U0RTS	SE_GPIO6	DBG_BUS13(G1)	GPIO63
GNSS_LNA_EN	VIO1V8	1.8V,4.7K/20K	50K	GNSS_LNA_EN			GPIO69
U1TXD	VIO1V8	1.8V,4.7K/20K	50K	U1TXD			GPIO70
U1RXD	VIO1V8	1.8V,4.7K/20K	50K	U1RXD	PPS(G1)		GPIO71
U2TXD	VIO1V8	1.8V,4.7K/20K	50K	U2TXD	SE_GPIO4	DBG_BUS14(G1)	GPIO72
U2RXD	VIO1V8	1.8V,4.7K/20K	50K	U2RXD	SE_GPIO5	DBG_BUS15(G1)	GPIO73
SCL1	VCAM	1.8V,1.8K/20K	50K	SCL1	EXTINT14		GPIO74
SDA1	VCAM	1.8V,1.8K/20K	50K	SDA1	EXTINT15		GPIO75
NF_DATA_1	VIO_NAND	1.8V,4.7K/20K	50K	NF_DATA_1	NF_DATA_1_T		GPIO76
T_DIG	VIO1V8	1.8V,4.7K/20K	50K	T_DIG			GPIO77

EXTINT9	VIO1V8	1.8V,4.7K/20K	50K	EXTINT9		BUA_TF_DET	GPIO78
EXTINT10	VIO1V8	1.8V,4.7K/20K	50K	EXTINT10		BAT_DET	GPIO79
TCK_ARM	VIO1V8	1.8V,4.7K/20K	50K	MTCK_ARM			GPIO82
TMS_ARM	VIO1V8	1.8V,4.7K/20K	50K	MTMS_ARM			GPIO83
TDO_LTE	VIO1V8	1.8V,4.7K/20K	50K	DTDO_LTE	DTDO_TWG	DBG_BUS26(G0)	GPIO85
TDI_LTE	VIO1V8	1.8V,4.7K/20K	50K	DTDI_LTE	DTDI_TWG	DBG_BUS27(G0)	GPIO86
TCK_LTE	VIO1V8	1.8V,4.7K/20K	50K	DTCK_LTE	DTCK_TWG	DBG_BUS28(G0)	GPIO87
TMS_LTE	VIO1V8	1.8V,4.7K/20K	50K	DTMS_LTE	DTMS_TWG	DBG_BUS23(G0)	GPIO88
RTCK_LTE	VIO1V8	1.8V,4.7K/20K	50K	DRTCK_LTE	DRTCK_TWG	DBG_BUS31(G0)	GPIO89
SPI0_CSN	VIO1V8	1.8V,4.7K/20K	50K	SPI0_CSN		EXTINT5	GPIO90
SPI0_DO	VIO1V8	1.8V,4.7K/20K	50K	SPI0_DO		EXTINT6	GPIO91
SPI0_DI	VIO1V8	1.8V,4.7K/20K	50K	SPI0_DI		EXTINT7	GPIO92
SPI0_CLK	VIO1V8	1.8V,4.7K/20K	50K	SPI0_CLK		EXTINT8	GPIO93
EMMC_D0	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D0	NF_WPN		GPIO98
EMMC_CMD	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_CMD	NF_RBN		GPIO99
EMMC_D6	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D6	NF_CLE		GPIO100
EMMC_D7	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D7	NF_ALE		GPIO101
EMMC_CLK	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_CLK	NF_RE_T		GPIO102
EMMC_D5	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D5	NF_DATA_4		GPIO103
EMMC_D4	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D4	NF_DATA_5		GPIO104
EMMC_DS	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_DS	NF_DATA_3		GPIO105
EMMC_D3	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D3	NF_DATA_7		GPIO106
EMMC_RST	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_RST	NF_CEN1		GPIO107
EMMC_D1	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D1	NF_DQS		GPIO108
EMMC_D2	VEMMCIO	1.8V,4.7K/20K	50K	EMMC_D2	NF_DATA_6		GPIO109
KEYOUT0	VIO1V8	1.8V,4.7K/20K	50K	KEYOUT0	EXTINT11		GPIO121
KEYOUT1	VIO1V8	1.8V,4.7K/20K	50K	KEYOUT1	EXTINT12	CM4_GPIO6	GPIO122
KEYOUT2	VIO1V8	1.8V,4.7K/20K	50K	KEYOUT2	PWMB(G0)	CM4_GPIO7	GPIO123
KEYIN0	VIO1V8	1.8V,4.7K/20K	50K	KEYIN0	EXTINT2	DBG_BUS30(G1)	GPIO124
KEYIN1	VIO1V8	1.8V,4.7K/20K	50K	KEYIN1	EXTINT3	PLL_LOCK	GPIO125
KEYIN2	VIO1V8	1.8V,4.7K/20K	50K	KEYIN2	EXTINT4	DBG_BUS31(G1)	GPIO126
SCL2	VIO1V8	1.8V,1.8K/20K	50K	SCL2			GPIO127
SDA2	VIO1V8	1.8V,1.8K/20K	50K	SDA2			GPIO128
CLK_AUX0	VIO1V8	1.8V,4.7K/20K	50K	CLK_AUX0	PROBE_CLK	DBG_BUS29(G1)	GPIO129
IIS1DI	VIO1V8	1.8V,4.7K/20K	50K	IIS1DI	SE_GPIO0	EXTINT13	GPIO130
IIS1DO	VIO1V8	1.8V,4.7K/20K	50K	IIS1DO	SE_GPIO1	CM4_GPIO3	GPIO131
IIS1CLK	VIO1V8	1.8V,4.7K/20K	50K	IIS1CLK	SE_GPIO2	CM4_GPIO4	GPIO132
IIS1LRCK	VIO1V8	1.8V,4.7K/20K	50K	IIS1LRCK	SE_GPIO3	EXT_XTL_EN1	GPIO133
SD2_CLK	VSD2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD2_CLK	SPI1_CLK	DBG_BUS0(G1)	GPIO134
SD2_CMD	VSD2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD2_CMD	SPI1_DI	DBG_BUS1(G1)	GPIO135
SD2_D0	VSD2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD2_D0	SPI1_DO	DBG_BUS2(G1)	GPIO136

SD2_D1	VSD2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD2_D1	SE_GPIO7	DBG_BUS3(G1)	GPIO137
SD2_D2	VSD2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD2_D2	SE_GPIO8	DBG_BUS4(G1)	GPIO138
SD2_D3	VSD2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD2_D3	SPI1_CSN	DBG_BUS5(G1)	GPIO139
NF_CEN0	VIO_NAND	1.8V,4.7K/20K	50K	NF_CEN0	NF_CEN0_T		GPIO140
NF_DATA_0	VIO_NAND	1.8V,4.7K/20K	50K	NF_DATA_0	NF_DATA_0_T		GPIO141
NF_WEN	VIO_NAND	1.8V,4.7K/20K	50K	NF_WEN	NF_WEN_T		GPIO142
NF_DATA_2	VIO_NAND	1.8V,4.7K/20K	50K	NF_DATA_2	NF_DATA_2_T		GPIO143
EXTINT0	VIO1V8	1.8V,4.7K/22K	50K	EXTINT0	WDRST	SE_GPIO14	GPIO144
EXTINT1	VIO1V8	1.8V,4.7K/22K	50K	EXTINT1		SE_GPIO15	GPIO145
SCL3	VIO1V8	1.8V,1.8K/22K	40K	SCL3		EXT_XTL_EN0	GPIO146
SDA3	VIO1V8	1.8V,1.8K/22K	40K	SDA3			GPIO147
SD0_D3	VSD0	1.8V,4.7K/20K	50K	SD0_D3		DBG_BUS6(G1)	GPIO148
SD0_D2	VSD0	1.8V/3.0V,, 1.8K/4.7K/20K	50K	SD0_D2		DBG_BUS7(G1)	GPIO149
SD0_CMD	VSD0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD0_CMD		DBG_BUS8(G1)	GPIO150
SD0_D0	VSD0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD0_D0		DBG_BUS9(G1)	GPIO151
SD0_D1	VSD0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD0_D1		DBG_BUS28(G1)	GPIO152
SD0_CLK	VSD0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SD0_CLK0			GPIO153
SIMCLK2	VSIM2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMCLK2	SCL4	SE_GPIO11	GPIO154
SIMDAT2	VSIM2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMDA2	SDA4	SE_GPIO12	GPIO155
SIMRST2	VSIM2	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMRST2	CLK_AUX1	SE_GPIO13	GPIO156
SIMCLK0	VSIM0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMCLK0			GPIO157
SIMDAT0	VSIM0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMDA0			GPIO158
SIMRST0	VSIM0	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMRST0			GPIO159
SIMCLK1	VSIM1	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMCLK1			GPIO160
SIMDAT1	VSIM1	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMDA1			GPIO161
SIMRST1	VSIM1	1.8V/3.0V, 1.8K/4.7K/20K	50K	SIMRST1			GPIO162