

---

## 如何在 Linux®下使用 SAMA5D2 I2C

---

### 简介

---

本应用笔记介绍在 Linux 下使用 SAMA5D2 I2C 的入门信息。

在大多数情况下，I2C 器件由内核驱动程序控制。不过，它也可以通过 I2C dev 接口从用户空间访问适配器上的所有器件。I2C dev 是一个字符设备节点文件，可通过 read()、write()和 ioctl()进行访问。每条 I2C 总线的接口均可通过其自己的 I2C dev 器件节点导出至用户空间。

I2C dev 应用程序演示代码在[应用](#)一节提供。

### 参考文档

---

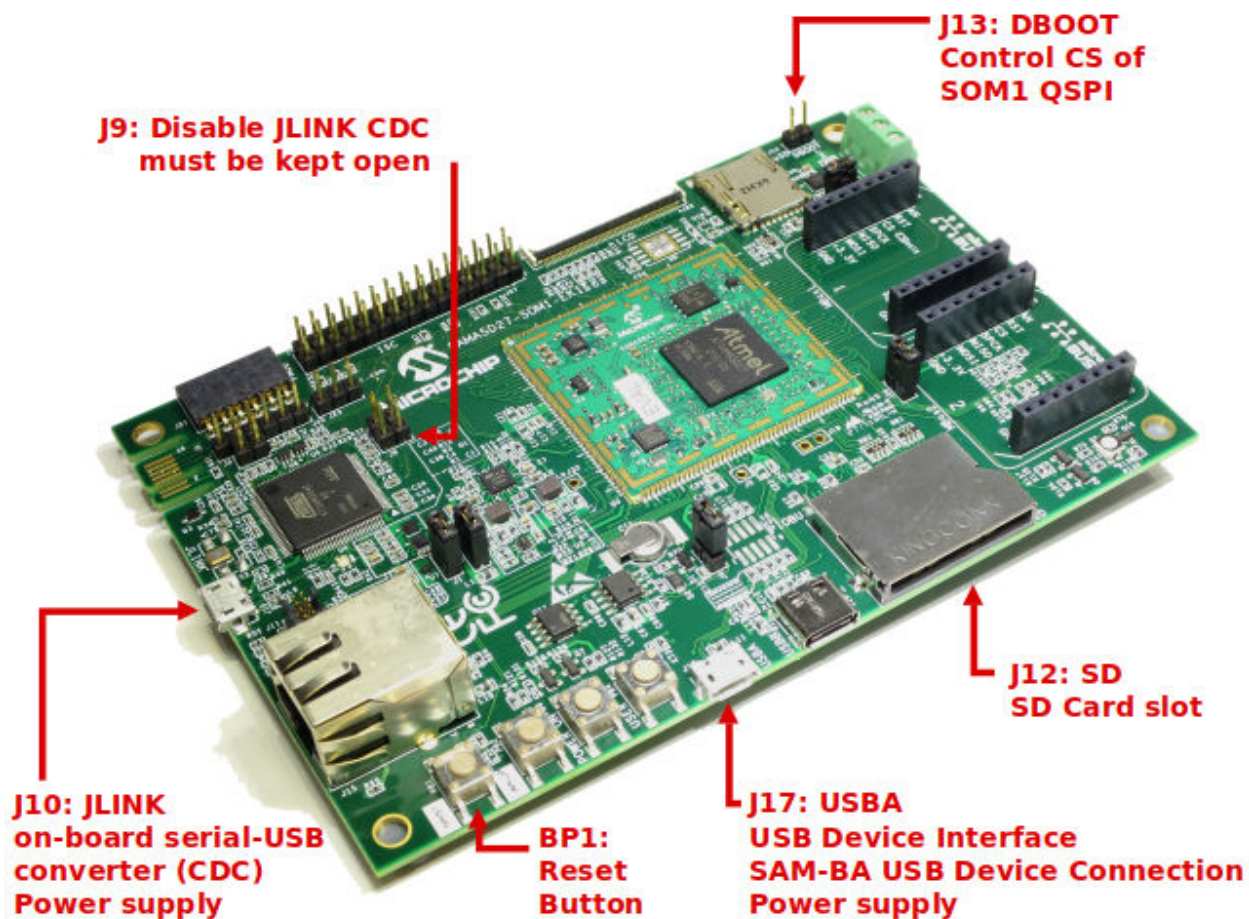
标题	编号	下载
SAMA5D2 Series Datasheet	DS60001476	<a href="https://www.microchip.com/design-centers/32-bit-mpus">https://www.microchip.com/design-centers/32-bit-mpus</a>
SAMA5D27 SOM1 Kit1 用户指南	DS50002667C_CN	<a href="http://www.microchip.com.cn/newcommunity//Uploads/2019617/50002667c_cn.pdf">http://www.microchip.com.cn/newcommunity//Uploads/2019617/50002667c_cn.pdf</a>

### 前提条件

---

- 硬件
  - PC
  - SAMA5D27 SOM1 评估工具包（部件编号：ATSAMA5D27-SOM1-EK1）
  - SD 卡
  - mikroBUS™ Thermo 5 click boards™
- 软件

本演示在 Buildroot 编译的 AT91 Linux 平台上运行。第一步是建立 AT91 Buildroot 开发环境。请访问以下网站：  
<http://www.at91.com/linux4sam/bin/view/Linux4SAM/BuildRoot>



---

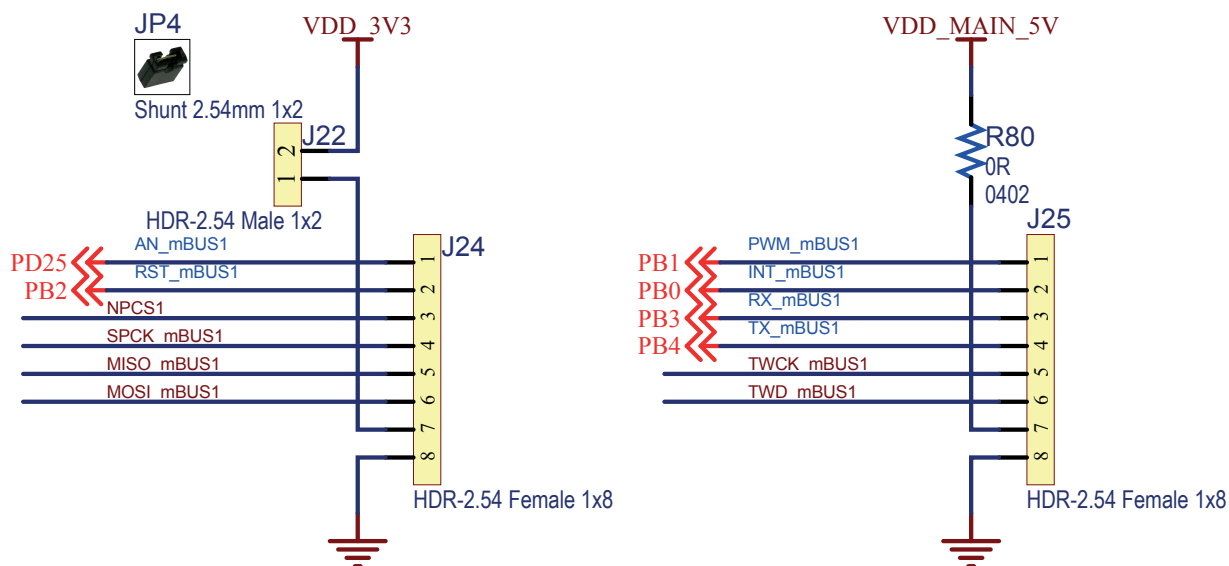
## 目录

---

简介.....	1
参考文档.....	1
前提条件.....	1
1. 硬件设计.....	4
1.1. 接口.....	4
1.2. 器件连接.....	4
2. 软件设计.....	6
2.1. 器件树.....	6
2.2. 内核.....	7
2.3. Rootfs.....	8
2.4. 应用程序.....	9
3. 动手实验.....	11
4. 工具和实用程序.....	12
5. Microchip 外设 I/O Python® (MPIO) .....	14
5.1. buildroot 中的 MPIO.....	14
5.2. 示例.....	16
6. 版本历史.....	17
6.1. 版本 A——2019 年 9 月.....	17
Microchip 网站.....	18
产品变更通知服务.....	18
客户支持.....	18
Microchip 器件代码保护功能.....	18
法律声明.....	18
商标.....	19
质量管理体系.....	19
全球销售及服务网点.....	20

## 1. 硬件设计

### 1.1 接口



mikroBUS1 连接器用于简化测试和监视。

要基于 Linux 控制 mikroBUS1 I2C，FLEXCOM1 (I2C 模式) 需与 SAMA5D27-SOM1-EK 上 mikroBUS1 接口的 I2C 总线相连接，具体如下所述：

#### FLEXCOM1 I2C

- FLEXCOM1\_IO1 → PA23 → TWCK\_mBUS1
- FLEXCOM1\_IO0 → PA24 → TWD\_mBUS1

有关 SAMA5D2 的引脚复用的更多详细信息，请参见 SAMA5D2 数据手册中的“引脚说明（所有封装）”表。

#### FLEXCOM I/O 线说明

名称	说明			类型
	USART/UART	SPI	TWI	
FLEXCOM_IO0	TXD	MOSI	TWD	I/O
FLEXCOM_IO1	RXD	MISO	TWCK	I/O
FLEXCOM_IO2	SCK	SPCK	-	I/O
FLEXCOM_IO3	CTS	NPCS0/NSS	-	I/O
FLEXCOM_IO4	RTS	NPCS1	-	O

### 1.2 器件连接

本演示使用 Thermo 5 click 板。该板必须连接到 SAMA5D27-SOM1-EK mikroBUS1。

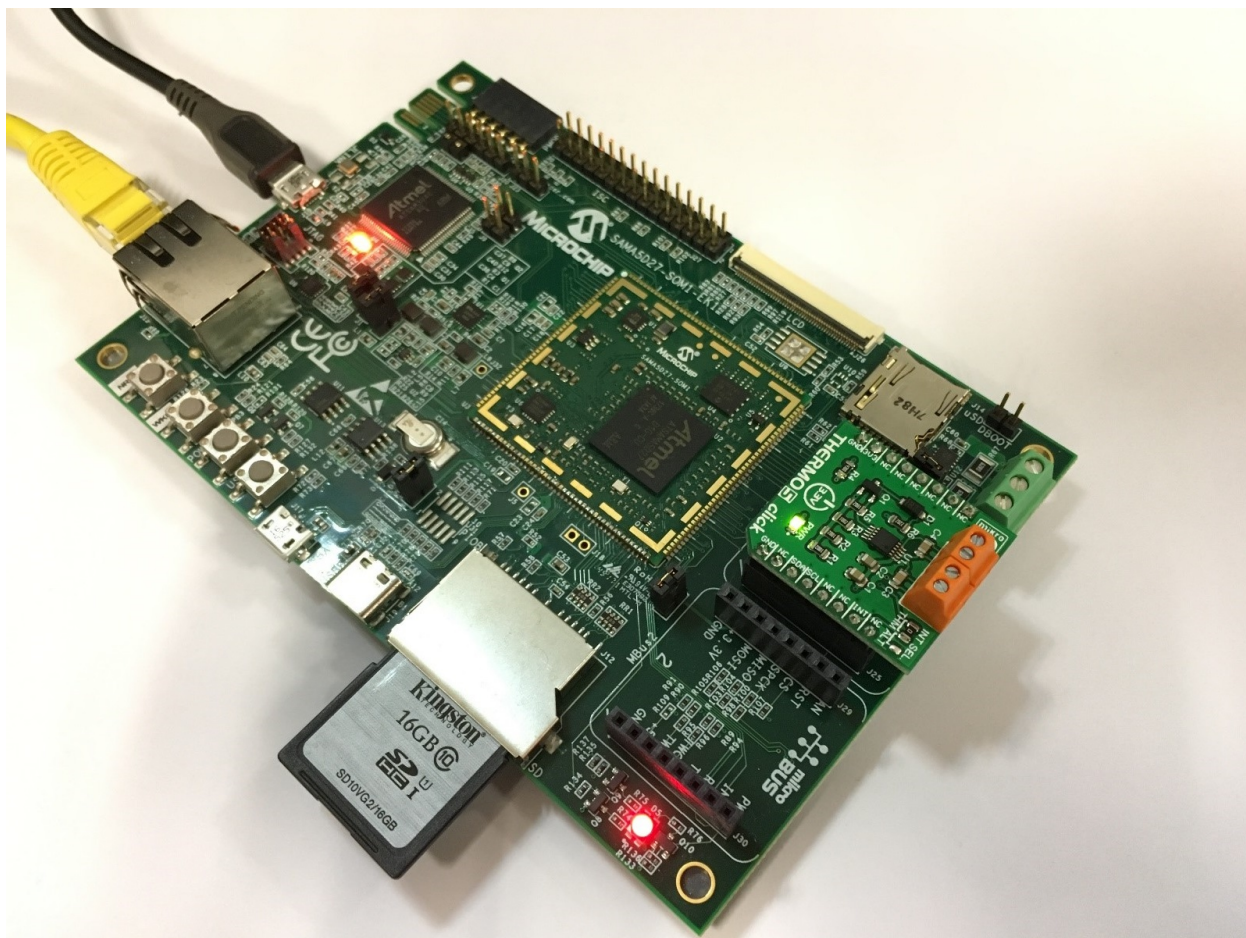
Thermo 5 click 板可测量的默认温度范围为 0°C 至 127°C，扩展温度范围为 -64°C 至 191°C，精度为  $\pm 1^{\circ}\text{C}$ 。该板配有 **EMC1414** 温度传感器。

请参见 <https://www.mikroe.com/thermo-5-click>。

图 1-1. Thermo 5 click 板



图 1-2. SAMA5D27-SOM1-EK1 评估板上的 Thermo 5





## 2. 软件设计

Microchip Linux 平台是使用 Buildroot 通过以下配置编译的：

`atmel_sama5d27_som1_ek_mmc_dev_defconfig`

I2C 总线驱动程序在此默认配置下工作。

有两种方法可以访问 I2C 总线驱动程序：

- 在内核空间中：
  - 通过 `i2c_add_driver()` 接口注册您自己的 I2C 驱动程序，然后通过 `struct i2c_client` 句柄访问 I2C 总线驱动程序。
- 在用户空间中：
  - 使能 `I2C_CHARDEV` 内核功能，然后通过 `/dev/i2c-X` 器件节点访问 I2C 总线驱动程序。

`I2C_CHARDEV` 是一个理想的选择，因为使能了默认配置 `I2C_CHARDEV` 功能之后，可在用户空间中运行完整代码（易于开发）。

### 2.1 器件树

- 操作：无需更改
- 位置：`buildroot-at91/output/build/linux-linux4sam_6.0/arch/arm/boot/dts`
- 来源：
  - `sama5d2.dtsi`
  - `at91-sama5d27_som1_ek.dts`

`sama5d2.dtsi` 中 `FLEXCOM1` 的器件树：

```
flx1: flexcom@f8038000 {
    compatible = "atmel,sama5d2-flexcom"; // 指定哪个驱动程序将用于此
                                           // FLEXCOM 器件
    reg = <0xf8038000 0x200>; // FLEXCOM1 基址为 0xf8038000，大小为 0x200

    clocks = <&flx1_clk>; // FLEXCOM1 时钟源的定义
    #address-cells = <1>;
    #size-cells = <1>;
    ranges = <0x0 0xf8038000 0x800>;
    status = "disabled"; // 默认为禁止状态，并将替换为 "okay"
};

flx1_clk: flx1_clk {
    #clock-cells = <0>;
    reg = <20>; // FLEXCOM1 的 PID 为 20，此偏移量定义将用于使能
               // PMC 中的 FLEXCOM1 时钟
    atmel,clk-output-range = <0 83000000>; // FLEXCOM1 输入时钟，最大频率为 83 MHz
};
```

at91-sama5d27\_som1\_ek.dts 中 I2C 器件功能的器件树:

```
aliases {
    serial0 = &uart1; /* DBGU */
    serial1 = &uart4; /* mikro BUS 1 */
    serial2 = &uart2; /* mikro BUS 2 */
    i2c1 = &i2c1;
    i2c2 = &i2c2; // FLEXCOM1 I2C 的别名为 i2c2, 因此 FLEXCOM1 I2C 将
                // 注册为 I2C 适配器 2
};

flx1: flexcom@f8038000 {
    atmel,flexcom-mode = <ATMEL_FLEXCOM_MODE_TWI>; // 为此 FLEXCOM 端口指定 I2C 模式
    status = "okay"; // 使能此器件

    i2c2: i2c@600 {
        compatible = "atmel,sama5d2-i2c"; // 指定哪个驱动程序将用于此 I2C
                                           // 器件
        reg = <0x600 0x200>; // FLEXCOM1 中 I2C 的寄存器偏移地址为 0x600,
                             // 大小为 0x200
        interrupts = <20 IRQ_TYPE_LEVEL_HIGH 7>; // FLEXCOM4 的 PID 为 23, 高级
                                                  // 触发, 优先级为 7
                                                  // 用于在 AIC 中配置 FLEXCOM4 中断

        dmas = <0>, <0>; // DMA 功能未使能
        dma-names = "tx", "rx";
        #address-cells = <1>;
        #size-cells = <0>;
        clocks = <&flx1_clk>; // FLEXCOM1 时钟源的定义
        pinctrl-names = "default";
        pinctrl-0 = <&pinctrl_mikrobus_i2c>; // I2C 的引脚定义
        atmel,fifo-size = <16>; // fifo 的指定大小为 16
        status = "okay"; // 使能此 I2C 器件
    };
};

pinctrl_mikrobus_i2c: mikrobus1_i2c {
    pinmux = <PIN_PA24_FLEXCOM1_IO0>, // PA24 的多路开关将切换到 FLEXCOM1_IO0
            <PIN_PA23_FLEXCOM1_IO1>; // PA23 的多路开关将切换到 FLEXCOM1_IO1
    bias-disable; // 上拉/下拉功能已禁止
};
```

## 2.2 内核

- 操作: 无需更改
- 位置: buildroot-at91/output/build/linux-linux4sam\_6.0/
- Defconfig: sama5\_defconfig
- 驱动程序文件:
  - drivers/i2c/i2c-core-base.c
  - drivers/i2c/i2c-dev.c
  - drivers/i2c/busses/i2c-at91.c

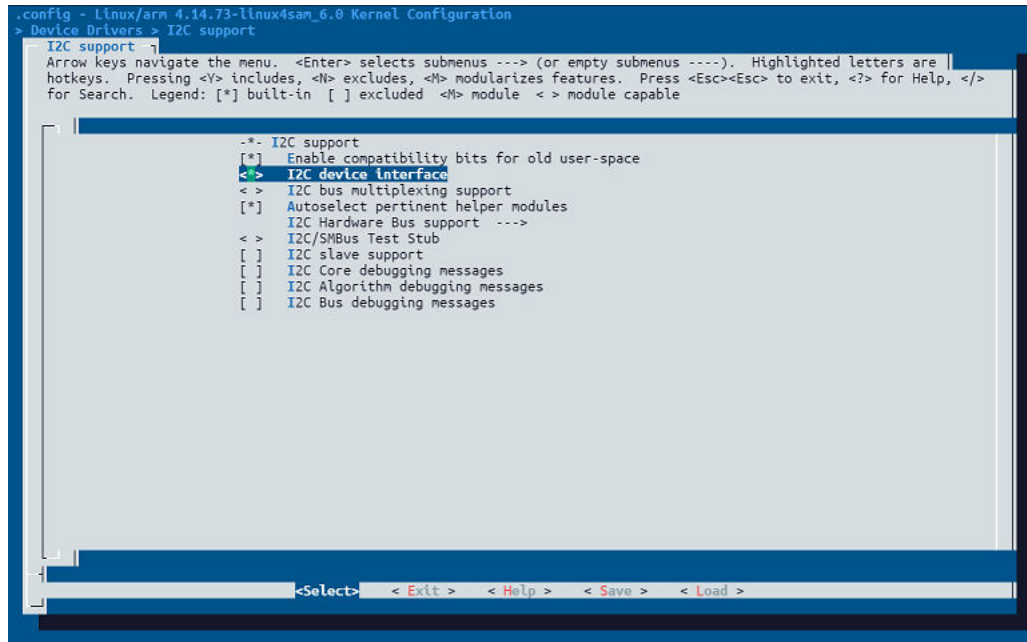
检查 ADC 功能的内核配置:

```
user@at91:~/buildroot-at91$ make linux-menuconfig
```

**Device Drivers > I2C support > I2C device interface** (器件驱动程序 > I2C 支持 > I2C 器件接口)

I2C\_CHARDEV 器件的驱动程序。

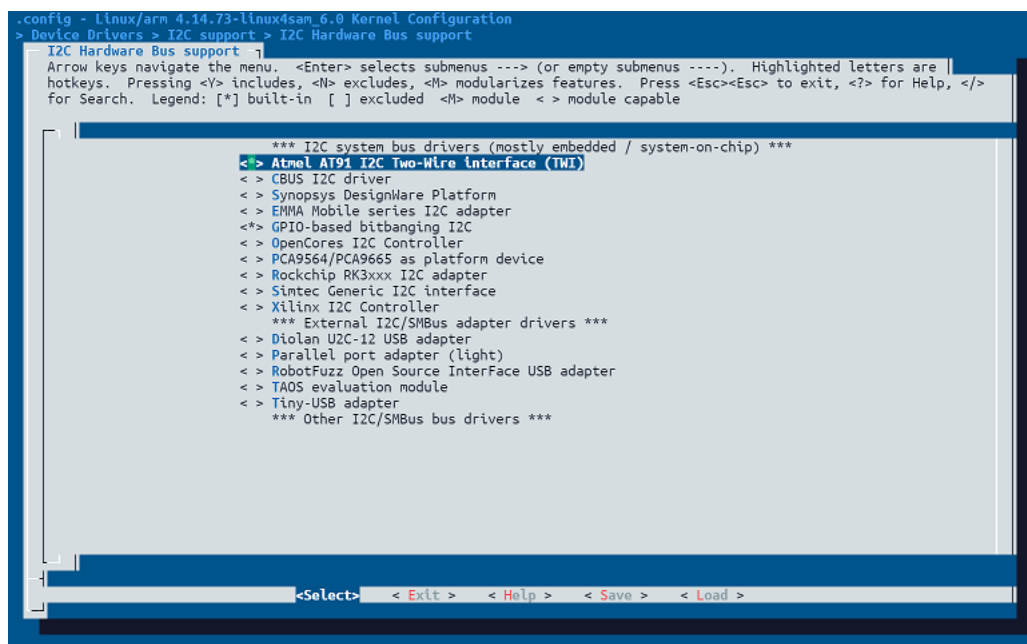
在默认配置中, 应选中此项。



Device Drivers > I2C support > Atmel AT91 I2C Two-Wire interface (TWI) (器件驱动程序 > I2C 支持 > Atmel AT91 I2C 双线接口 (TWI))

AT91 I2C 控制器的驱动程序。

在默认配置中，应选中此项。



## 2.3 Rootfs

- 操作：无需更改
- 位置：buildroot-at91/output/images/rootfs.tar

如[器件树](#)一节所述，FLEXCOM1 I2C 注册为 I2C 适配器 2。



以下器件节点用于访问 FLEXCOM1 I2C 总线驱动程序：

- /dev/i2c-2

## 2.4 应用程序

本节提供通过/dev/i2c-2 节点从 Thermo 5 click 板读取温度数据的 C 语言演示 i2c\_dev.c。

### 编译方式

```
user@at91:~$ buildroot-at91/output/host/bin/arm-buildroot-linux-uclibcgnueabi-gcc
i2c_dev.c -o i2c_test
```

### 源代码

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>

#define DEV_I2C "/dev/i2c-2"

#define SLAVE_ADDR 0x4C /* EMC1414 I2C 从器件地址 */
// #define COMBINED_TRANSACTION

int main(int argc, char *argv[])
{
    int fd;
    int ret;
    unsigned char buf[2];

    // 打开器件节点
    fd = open(DEV_I2C, O_RDWR);
    if (fd < 0 {
        printf("ERROR open %s ret=%d\n", DEV_I2C, fd);
        return -1;
    }

    if (ioctl(fd, I2C_SLAVE, SLAVE_ADDR) < 0) {
        printf("ERROR ioctl() set slave address\n");
        return -1;
    }

#ifdef COMBINED_TRANSACTION
    struct i2c_rdwr_ioctl_data data;
    struct i2c_msg messages[2];

    // 设置转换速率
    buf[0] = 0x04; // 转换速率寄存器地址
    buf[1] = 0x04; // 将转换速率设置为 1 秒
    messages[0].addr = SLAVE_ADDR; // 器件地址
    messages[0].flags = 0; // 写入
    messages[0].len = 2;
    messages[0].buf = buf; // 数据地址

    data.msgs = &messages[0];
    data.nmsgs = 1;
    if (ioctl(fd, I2C_RDWR, &data) < 0 {
        printf("ERROR ioctl() conversion rate\n");
        return -1;
    }

    // 读取温度
    buf[0] = 0x00; // 内部二极管高字节寄存器地址
    buf[1] = 0; // 清除接收缓冲区
    messages[0].addr = SLAVE_ADDR; // 器件地址
    messages[0].flags = 0; // write
```

```

    messages[0].len    = 1;
    messages[0].buf    = &buf[0]; // 数据地址

    messages[1].addr   = SLAVE_ADDR; // 器件地址
    messages[1].flags  = I2C_M_RD; // 读取
    messages[1].len    = 1;
    messages[1].buf    = &buf[1];

    data.msgs = messages;
    data.nmsgs = 2;
    while (1) {
        if (ioctl(fd, I2C_RDWR, &data) < 0) {
            printf("ERROR ioctl() read data\n");
            return -1;
        }

        printf("Temperature is %d\n", buf[1]);
        sleep(1);
    }
#else
    // 设置转换速率
    buf[0] = 0x04; // 转换速率寄存器地址
    buf[1] = 0x04; // 将转换速率设置为 1 秒
    ret = write(fd, buf, 2);
    if (ret != 2) {
        printf("ERROR write() conversion rate\n");
        return -1;
    }

    // 读取温度
    // 设置内部地址寄存器指针
    buf[0] = 0x00; // 内部二极管高字节寄存器地址
    ret = write(fd, &buf[0], 1);
    if (ret != 1) {
        printf("ERROR write() register address\n");
        return -1;
    }

    while (1) {
        // 读取温度
        // 读取数据
        buf[1] = 0; // 清除接收缓冲区
        ret = read(fd, &buf[1], 1);
        if (ret != 1) {
            printf("ERROR read() data\n");
            return -1;
        }

        printf("Temperature is %d\n", buf[1]);
        sleep(1);
    }
#endif

    // 关闭器件节点
    close(fd);

    return 0;
}

```

### 3. 动手实验

将 i2c\_test 应用程序复制到目标并执行。随后将显示温度数据。

```
# chmod +x i2c_test
# ./i2c_test
Temperature is 26
Temperature is 26
Temperature is 26
Temperature is 26
Temperature is 26
Temperature is 26
```

## 4. 工具和实用程序

I2c-tools 是一款用于执行 I2C 总线测试的工具，包含在 Buildroot 中。

在默认的 Buildroot 配置中，将选择此工具。

```
user@at91:~/buildroot-at91$ make menuconfig
```

**Target packages > Hardware handling > i2c-tools** (目标软件包 > 硬件处理 > i2c-tools)

i2c-tools 中有多个命令：

```
# i2cdetect
Error: No i2c-bus specified!
Usage: i2cdetect [-y] [-a] [-q|-r] I2CBUS [FIRST LAST]
       i2cdetect -F I2CBUS
       i2cdetect -l
       I2CBUS is an integer or an I2C bus name
       If provided, FIRST and LAST limit the probing range.
# i2cdetect -y 3
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: UU 51 52 53 54 55 56 57 -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

# i2cdump
Error: No i2c-bus specified!
Usage: i2cdump [-f] [-y] [-r first-last] I2CBUS ADDRESS [MODE [BANK [BANKREG]]]
       I2CBUS is an integer or an I2C bus name
       ADDRESS is an integer (0x03 - 0x77)
       MODE is one of:
         b (byte, default)
         w (word)
         W (word on even register addresses)
         s (SMBus block)
         i (I2C block)
         c (consecutive byte)
       Append p for SMBus PEC
# i2cdump -f -y 3 0x50
No size specified (using byte-data access)
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f      0123456789abcdef
00: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
10: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
20: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
30: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
40: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
50: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
60: 20 4d 43 48 49 50 20 52 46 4f 12 02 42 42 31 ff  MCHIP RFO??BB1.
70: 00 00 44 32 37 2d 53 4f 4d 31 2d 45 4b 31 44 f7  ..D27-SOM1-EK1D?
80: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
90: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
a0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
b0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
c0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
d0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
e0: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff  .....
f0: ff ff ff ff ff ff ff ff ff ff ff 54 10 ec 33 c5 7d .....T??3??

# i2cget
Usage: i2cget [-f] [-y] I2CBUS CHIP-ADDRESS [DATA-ADDRESS [MODE]]
       I2CBUS is an integer or an I2C bus name
       ADDRESS is an integer (0x03 - 0x77)
       MODE is one of:
         b (read byte data, default)
         w (read word data)
         c (write byte/read byte)
       Append p for SMBus PEC
# i2cget -f -y 3 0x50 0x60
0x20
```

```
# i2cget -f -y 3 0x50 0x61
0x4d

# i2cset
Usage: i2cset [-f] [-y] [-m MASK] [-r] I2CBUS CHIP-ADDRESS DATA-ADDRESS [VALUE] ... [MODE]
I2CBUS is an integer or an I2C bus name
ADDRESS is an integer (0x03 - 0x77)
MODE is one of:
  c (byte, no value)
  b (byte data, default)
  w (word data)
  i (I2C block data)
  s (SMBus block data)
Append p for SMBus PEC
WARNING: DO NOT modify any data in Eeprom of SAMA5D27_SOM1

# i2ctransfer
Usage: i2ctransfer [-f] [-y] [-v] [-V] I2CBUS DESC [DATA] [DESC [DATA]]...
I2CBUS is an integer or an I2C bus name
DESC describes the transfer in the form: {r|w}LENGTH[@address]
  1) read/write-flag 2) LENGTH (range 0-65535) 3) I2C address (use last one if omitted)
DATA are LENGTH bytes for a write message. They can be shortened by a suffix:
  = (keep value constant until LENGTH)
  + (increase value by 1 until LENGTH)
  - (decrease value by 1 until LENGTH)
  p (use pseudo random generator until LENGTH with value as seed)
```

示例（总线 0，从 0x50 处的 EEPROM 中的偏移 0x64 处读取 8 字节）：# i2ctransfer 0 w1@0x50 0x64 r8

示例（同一 EEPROM，在偏移 0x42 处写入 0xff 0xfe ... 0xf0）：# i2ctransfer 0 w17@0x50 0x42 0xff-

## 5. Microchip 外设 I/O Python® (MPIO)

Microchip 外设 I/O (MPIO) Python 软件包可轻松访问运行 Linux 的 Microchip MPU 处理器和评估板上的各种硬件外设。这套 API 干净、一致、灵活、记录完备且经过全面测试，即使是最复杂的硬件外设，也可通过它来轻松浏览和使用。

更多信息，请参见 [github.com/linux4sam/mpio](https://github.com/linux4sam/mpio)。文件夹 `mpio/examples` 中提供了相关代码示例来说明如何使用 MPIO 接口模块。

### 5.1 buildroot 中的 MPIO

要在 buildroot 配置中充分利用 MPIO 的优势，请执行以下步骤：

#### 1. 使能 Python

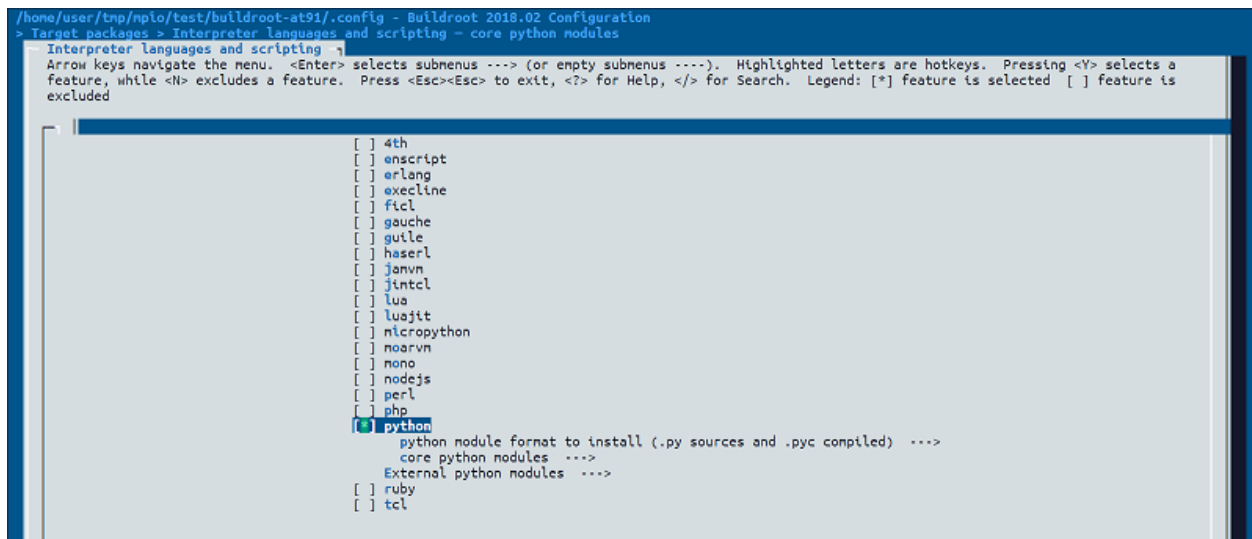
```
user@at91:~/buildroot-at91$ make menuconfig
```

选择 “python” 以使能 python 支持：

- Target packages > Interpreter languages and scripting > [\*] **python** (目标软件包 > 解释程序语言和脚本 > [\*] python)

然后进入 “python module format to install” (要安装的 python 模块格式) 并选择 “.py sources and .pyc compiled” (.py 源和已编译.pyc)。

- Target packages > Interpreter languages and scripting > python > python module format to install > **.py sources and .pyc compiled** (目标软件包 > 解释程序语言和脚本 > python > 要安装的 python 模块格式 > .py 源和已编译.pyc)



必须选择一些其他的 python 模块。进入 “core python modules” (内核 python 模块)，然后选择 “curses module” (curses 模块)、“readline” 和 “hashlib module” (hashlib 模块)。

- Target packages > Interpreter languages and scripting > core python modules > [\*] **curses module** (目标软件包 > 解释程序语言和脚本 > 内核 python 模块 > [\*] curses 模块)
- Target packages > Interpreter languages and scripting > core python modules > [\*] **readline** (目标软件包 > 解释程序语言和脚本 > 内核 python 模块 > [\*] readline)



- Target packages > Interpreter languages and scripting > core python modules > [\*] hashlib module

```

/home/user/tmp/mpio/test/buildroot-at91/.config - Buildroot 2018.02 Configuration
> Target packages > Interpreter languages and scripting > core python modules
core python modules
Arrow keys navigate the menu. <Enter> selects submenus --- (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N> excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected [ ] feature is excluded

*** The following modules are unusual or require extra libraries ***
[ ] bzip2 module (NEW)
[ ] bsddb module (NEW)
[ ] codescs module (NEW)
[*] curses module
[ ] ossaudiodev module (NEW)
[*] readline
[ ] ssl (NEW)
[ ] unicodedata module
[ ] sqlite module
[ ] xml module (NEW)
[ ] zlib module (NEW)
[*] hashlib module

```

(目标软件包 > 解释程序语言和脚本 > 内核 python 模块 > [\*] hashlib 模块)

进入 “External python modules” (外部 python 模块)，然后选择 “python-setuptools”。

- Target packages > Interpreter languages and scripting > External python modules > [\*] python-setuptools (目标软件包 > 解释程序语言和脚本 > 外部 python 模块 > [\*] python-setuptools)

```

/home/user/tmp/mpio/test/buildroot-at91/.config - Buildroot 2018.02 Configuration
> Target packages > Interpreter languages and scripting > External python modules
External python modules
Arrow keys navigate the menu. <Enter> selects submenus --- (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N> excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected [ ] feature is excluded

^(-)- [ ] python-pyudev (NEW)
[ ] python-pyusb (NEW)
[ ] python-pyxb (NEW)
[ ] python-pyyaml (NEW)
[ ] python-pyzmq (NEW)
[ ] python-raven (NEW)
[ ] python-remi (NEW)
[ ] python-requests (NEW)
[ ] python-requests-oauthlib (NEW)
[ ] python-requests-toolbelt (NEW)
[ ] python-rpi-gpio (NEW)
[ ] python-rtplib-fb (NEW)
[ ] python-scandir (NEW)
[ ] python-schedule (NEW)
[ ] python-sdnottyfy (NEW)
[ ] python-secretstorage (NEW)
[ ] python-see (NEW)
[ ] python-serial (NEW)
[ ] python-service-identity (NEW)
[ ] python-setproctitle (NEW)
[*] python-setuptools
[ ] python-sh (NEW)
[ ] python-shutilwhich (NEW)

```

## 2.使能 MPIO 模块

进入 “External options” (外部选项) 并选择 “python-mpio”。

- External options > [\*] python-mpio (外部选项 > [\*] python-mpio)

```

/home/user/tmp/mpio/test/buildroot-at91/.config - Buildroot 2018.02 Configuration
> External options
External options
Arrow keys navigate the menu. <Enter> selects submenus --- (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N> excludes a feature. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] feature is selected [ ] feature is excluded

*** MCHP (in /home/user/tmp/mpio/test/buildroot-external-microchip) ***
[ ] g1-decoder
*** gst1-at91 needs a toolchain w/ glibc ***
*** libplanes depends on libdrm, cairo, lua, and cJSON ***
[ ] ptc_examples
*** python-iocontrol depends on pyqt5 and mpio ***
[*] python-mpio
[ ] dt-overlay-at91
Init configuration (none) ---

```

## 3.完成 buildroot 配置并进行编译

进入 “Filesystem images” (文件系统映像) 并将 rootfs 的确切大小设置为 120 MB。

- Filesystem images > (120M) exact size (文件系统映像 > (120M) 确切大小)

```

/home/user/tmp/mpio/test/buildroot-at91/.config - Buildroot 2018.02 Configuration
Filesystem Images
Filesystem Images
[ ] axfs root filesystem
[ ] cloop root filesystem for the target device
[ ] cpio the root filesystem (for use as an initial RAM filesystem)
[ ] cramfs root filesystem
[*] ext2/3/4 root filesystem
    ext2/3/4 variant (ext4) --->
    () filesystem label
    (120M) exact size
    (0) exact number of inodes (leave at 0 for auto calculation)
    (5) reserved blocks percentage
    (-O ^64bit) additional mke2fs options
    Compression method (no compression) --->

```

保存后，以下新设置将添加到 buildroot 的配置文件中：

.....

```

BR2_PACKAGE_PYTHON=y
BR2_PACKAGE_PYTHON_PY_PYC=y
BR2_PACKAGE_PYTHON_CURSES=y
BR2_PACKAGE_PYTHON_READLINE=y
BR2_PACKAGE_PYTHON_HASHLIB=y
BR2_PACKAGE_PYTHON_SETUPTOOLS=y
BR2_PACKAGE_PYTHON_MPIO=y
BR2_TARGET_ROOTFS_EXT2_SIZE="120M"

```

.....

然后重新配置并编译 buildroot：

```
user@at91:~/buildroot-at91$ make atmel_sama5d27_som1_ek_mmc_dev_defconfig
```

```
user@at91:~/buildroot-at91$ make
```

## 5.2 示例

编译成功后，使用 buildroot-at91/output/images/sdcard.img 烧写 SD 卡。

在目标板上执行 python 代码，例如：

```

# ./adc2.py DEVICE
# ./gpio1.py PIN
# ./pwm_led.py DEVICE CHANNEL

```

.....

注：python 示例代码可从 <https://github.com/linux4sam/mpio/examples> 获得

---

## 6. 版本历史

### 6.1 版本 A——2019 年 9 月

第一版。

---

## Microchip 网站

---

Microchip 网站 ([www.microchip.com/](http://www.microchip.com/)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容:

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

---

## 产品变更通知服务

---

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请访问 [www.microchip.com/pcn](http://www.microchip.com/pcn), 然后按照注册说明进行操作。

---

## 客户支持

---

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 [www.microchip.com/support](http://www.microchip.com/support) 获得网上技术支持。

---

## Microchip 器件代码保护功能

---

请注意以下有关 Microchip 器件代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信: 在正常使用的情况下, Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿意与关心代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下, 能访问您的软件或其他受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

---

## 法律声明

---

提供本文档的中文版本仅为为了便于理解。请勿忽视文档中包含的英文部分, 因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中提供的信息仅仅是为了方便您使用 Microchip 产品或使用这些产品来进行设计。本出版物中所述的器件应用信息及其他类似内容仅为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。

MICROCHIP“按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，MICROCHIP 概不承担任何责任，即使 MICROCHIP 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，MICROCHIP 在任何情况下所承担的全部责任均不超出您为获得这些信息向 MICROCHIP 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

## 商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 及 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQL、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 均为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2020, Microchip Technology Incorporated 版权所有。

ISBN: 978-1-5224-7317-6

AMBA、Arm、Arm7、Arm7TDMI、Arm9、Arm11、Artisan、big.LITTLE、Cordio、CoreLink、CoreSight、Cortex、DesignStart、DynamIQ、Jazelle、Keil、Mali、Mbed、Mbed Enabled、NEON、POP、RealView、SecurCore、Socrates、Thumb、TrustZone、ULINK、ULINK2、ULINK-ME、ULINK-PLUS、ULINKpro、µVision 和 Versatile 均为 Arm Limited（或其子公司）在美国和/或其他国家/地区的商标或注册商标。

## 质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 [www.microchip.com/quality](http://www.microchip.com/quality)。

## 全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
<b>公司总部</b> 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: <a href="http://www.microchip.com/support">www.microchip.com/support</a> 网址: <a href="http://www.microchip.com">www.microchip.com</a> <b>亚特兰大</b> 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 <b>奥斯汀, 德克萨斯州</b> 电话: 512-257-3370 <b>波士顿</b> 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 <b>芝加哥</b> 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 <b>达拉斯</b> 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 <b>底特律</b> 诺维, 密歇根州 电话: 248-848-4000 <b>休斯顿, 德克萨斯州</b> 电话: 281-894-5983 <b>印第安纳波利斯</b> 诺布尔斯维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 <b>洛杉矶</b> 米慎维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 <b>罗利, 北卡罗来纳州</b> 电话: 919-844-7510 <b>纽约, 纽约州</b> 电话: 631-435-6000 <b>圣何塞, 加利福尼亚州</b> 电话: 408-735-9110 电话: 408-436-4270 <b>加拿大 - 多伦多</b> 电话: 905-695-1980 传真: 905-695-2078	<b>澳大利亚 - 悉尼</b> 电话: 61-2-9868-6733 <b>中国 - 北京</b> 电话: 86-10-8569-7000 <b>中国 - 成都</b> 电话: 86-28-8665-5511 <b>中国 - 重庆</b> 电话: 86-23-8980-9588 <b>中国 - 东莞</b> 电话: 86-769-8702-9880 <b>中国 - 广州</b> 电话: 86-20-8755-8029 <b>中国 - 杭州</b> 电话: 86-571-8792-8115 <b>中国 - 香港特别行政区</b> 电话: 852-2943-5100 <b>中国 - 南京</b> 电话: 86-25-8473-2460 <b>中国 - 青岛</b> 电话: 86-532-8502-7355 <b>中国 - 上海</b> 电话: 86-21-3326-8000 <b>中国 - 沈阳</b> 电话: 86-24-2334-2829 <b>中国 - 深圳</b> 电话: 86-755-8864-2200 <b>中国 - 苏州</b> 电话: 86-186-6233-1526 <b>中国 - 武汉</b> 电话: 86-27-5980-5300 <b>中国 - 西安</b> 电话: 86-29-8833-7252 <b>中国 - 厦门</b> 电话: 86-592-2388138 <b>中国 - 珠海</b> 电话: 86-756-3210040	<b>印度 - 班加罗尔</b> 电话: 91-80-3090-4444 <b>印度 - 新德里</b> 电话: 91-11-4160-8631 <b>印度 - 浦那</b> 电话: 91-20-4121-0141 <b>日本 - 大阪</b> 电话: 81-6-6152-7160 <b>日本 - 东京</b> 电话: 81-3-6880-3770 <b>韩国 - 大邱</b> 电话: 82-53-744-4301 <b>韩国 - 首尔</b> 电话: 82-2-554-7200 <b>马来西亚 - 吉隆坡</b> 电话: 60-3-7651-7906 <b>马来西亚 - 槟榔屿</b> 电话: 60-4-227-8870 <b>菲律宾 - 马尼拉</b> 电话: 63-2-634-9065 <b>新加坡</b> 电话: 65-6334-8870 <b>台湾地区 - 新竹</b> 电话: 886-3-577-8366 <b>台湾地区 - 高雄</b> 电话: 886-7-213-7830 <b>台湾地区 - 台北</b> 电话: 886-2-2508-8600 <b>泰国 - 曼谷</b> 电话: 66-2-694-1351 <b>越南 - 胡志明市</b> 电话: 84-28-5448-2100	<b>奥地利 - 韦尔斯</b> 电话: 43-7242-2244-39 传真: 43-7242-2244-393 <b>丹麦 - 哥本哈根</b> 电话: 45-4485-5910 传真: 45-4485-2829 <b>芬兰 - 埃斯波</b> 电话: 358-9-4520-820 <b>法国 - 巴黎</b> 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 <b>德国 - 加兴</b> 电话: 49-8931-9700 <b>德国 - 哈恩</b> 电话: 49-2129-3766400 <b>德国 - 海布隆</b> 电话: 49-7131-72400 <b>德国 - 卡尔斯鲁厄</b> 电话: 49-721-625370 <b>德国 - 慕尼黑</b> 电话: 49-89-627-144-0 传真: 49-89-627-144-44 <b>德国 - 罗森海姆</b> 电话: 49-8031-354-560 <b>以色列 - 若那那市</b> 电话: 972-9-744-7705 <b>意大利 - 米兰</b> 电话: 39-0331-742611 传真: 39-0331-466781 <b>意大利 - 帕多瓦</b> 电话: 39-049-7625286 <b>荷兰 - 德卢内市</b> 电话: 31-416-690399 传真: 31-416-690340 <b>挪威 - 特隆赫姆</b> 电话: 47-72884388 <b>波兰 - 华沙</b> 电话: 48-22-3325737 <b>罗马尼亚 - 布加勒斯特</b> 电话: 40-21-407-87-50 <b>西班牙 - 马德里</b> 电话: 34-91-708-08-90 传真: 34-91-708-08-91 <b>瑞典 - 哥德堡</b> 电话: 46-31-704-60-40 <b>瑞典 - 斯德哥尔摩</b> 电话: 46-8-5090-4654 <b>英国 - 沃金厄姆</b> 电话: 44-118-921-5800 传真: 44-118-921-5820