八、BeagleBone Black I2C及温湿度传感器SHT31的使用

一、确认BeagleBone Black已有的I2C总线。

```
root@beaglebone:~# cd /dev/
root@beaglebone:/dev# ls i2*
i2c-0  i2c-1
root@beaglebone:/dev#
```

可知当前存在两个设备：i2c-0 i2c-1 对应的I2C总线时I2C1，I2C2（也许是错误的）。

二、通过设备树覆盖层和外设管理器增加I2C1总线（i2c-2）.

2.1、设备树覆盖层

扁平设备树（The Flattened Device Tree, FDT）对于输入和输出运行时间的配置（也就是Linux启动后）时不合适的，幸运的是，可以用设备树覆盖层（device tree overlays, DTOs）和一个外设管理器来解决。

capes是外扩子板，它能通过插头连接到BeagleBone Black上。每个外扩子板有一个设备树二进制对象.dtbo文件。理论上讲，当你插入一个外扩子板到BeagleBone Black上时，相应的.dtbo文件会被加载，用来配置P8/P9插头引脚和外扩子板接口。外设管理器被设计用来允许外扩子板在运行时被动态加载，而不需要重新编译Linux内核。外设管理器已经在Linux内核中被完全实现了，但是它会在运行时从/lib/firmware目录中加载二进制.dtbo文件。

2.1.1、加载一个设备树覆盖层

2.1.1.1、为了使用外设管理器，首先应该设置两个环境变量

```
export SLOTS=/sys/devices/bone_capemgr.9/slots
```

```
debian@beaglebone:~$ cd /sys/devices/bone_capemgr.9/slots
-bash: cd: /sys/devices/bone_capemgr.9/slots: Not a directory
debian@beaglebone:~$ cd /sys/devices/bone_capemgr.9/
debian@beaglebone:/sys/devices/bone_capemgr.9$ ls
baseboard  driver  modalias  power  slot-4  slot-5  slots  subsystem  uevent
debian@beaglebone:/sys/devices/bone_capemgr.9$ ls slots
slots
debian@beaglebone:/sys/devices/bone_capemgr.9$ more slots
 0: 54:PF---
 1: 55:PF---
 2: 56:PF---
 3: 57:PF---
 4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
 5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
debian@beaglebone:/sys/devices/bone_capemgr.9$ cd ~
debian@beaglebone:~$ export SLOTS=/sys/devices/bone_capemgr.9/slots
debian@beaglebone:~$ cat $SLOTS
 0: 54:PF---
 1: 55:PF---
 2: 56:PF---
 3: 57:PF---
 4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
 5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
debian@beaglebone:~$
```

```
export PINS=/sys/kernel/debug/pinctrl/44e10800.pinmux/pins
```

```
root@beaglebone:/sys/kernel/debug/pinctrl/44e10800.pinmux# ls
gpio-ranges     pinconf-pins  pinmux-functions  pins
pinconf-groups  pingroups     pinmux-pins
root@beaglebone:/sys/kernel/debug/pinctrl/44e10800.pinmux# pwd
/sys/kernel/debug/pinctrl/44e10800.pinmux
root@beaglebone:/sys/kernel/debug/pinctrl/44e10800.pinmux# su debian
rnel/debug/pinctrl/44e10800.pinmux/pinsctrl/44e10800.pinmux$ export PINS=/sys/ke
debian@beaglebone:/sys/kernel/debug/pinctrl/44e10800.pinmux$ cat $P
$PATH        $PIPESTATUS  $PS1         $PS4
$PINS        $PPID        $PS2         $PWD
debian@beaglebone:/sys/kernel/debug/pinctrl/44e10800.pinmux$ cat $P
$PATH        $PIPESTATUS  $PS1         $PS4
$PINS        $PPID        $PS2         $PWD
debian@beaglebone:/sys/kernel/debug/pinctrl/44e10800.pinmux$ cat $PINS
cat: /sys/kernel/debug/pinctrl/44e10800.pinmux/pins: Permissisudo cat $PINS
registered pins: 142
pin 0 (44e10800) 00000031 pinctrl-single
pin 1 (44e10804) 00000031 pinctrl-single
pin 2 (44e10808) 00000031 pinctrl-single
pin 3 (44e1080c) 00000031 pinctrl-single
pin 4 (44e10810) 00000031 pinctrl-single
```

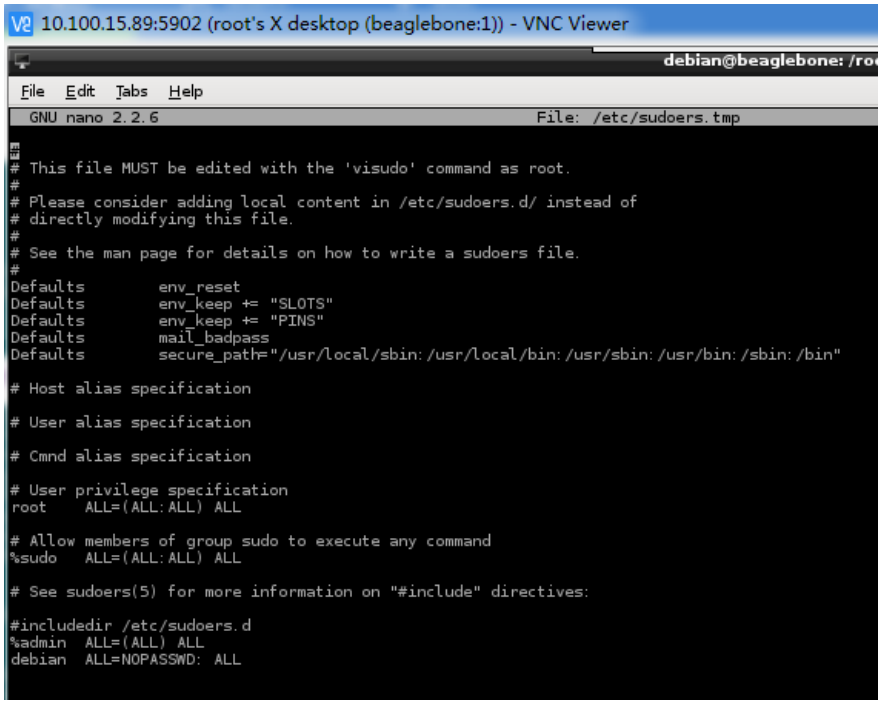
2.1.1.2、使用source命令来设置环境变量。当你启动时他们将自动设置。

```
source ~/.profile
```

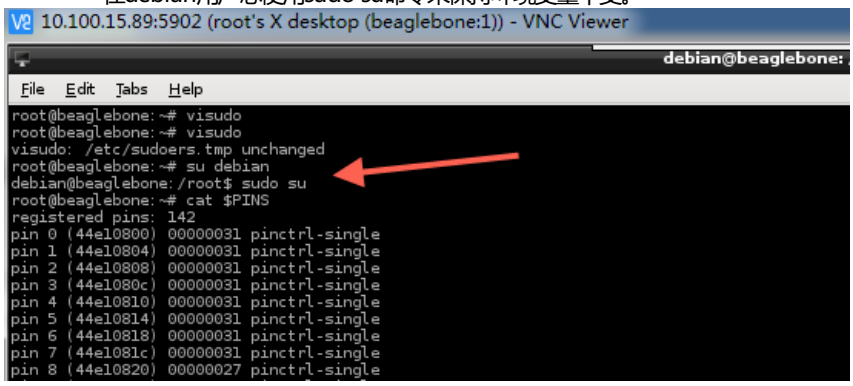
2.1.1.3、传递环境变量给根用户接口，固定SLOTS和PINS。

以root身份登录并输入visudo，在env_reset下加入

```
Defaults    env_keep += "SLOTS"
Defaults    env_keep += "PINS"
```



在debian用户态使用sudo su命令来保持环境变量不变。



2.1.1.4、加载I2C1（i2c-2）总线设备树覆盖层。

```
debian@beaglebone:/dev$ ls i2c*
i2c-0  i2c-1
debian@beaglebone:/dev$ sudo sh -c "echo BB-I2C1 > $SLOTS"
debian@beaglebone:/dev$ ls i2c*
i2c-0  i2c-1  i2c-2
debian@beaglebone:/dev$
```

三、使用Linux I2C工具（i2c-tools）.

3.1、更新或安装i2c-tools。

```
sudo apt-get install i2c-tools
```

3.2、i2cdetect

用于检测I2C总线及总线上的设备。

```
sudo i2cdetect -l
```

```
debian@beaglebone:/dev$ ls i2c*
i2c-0  i2c-1
debian@beaglebone:/dev$ sudo sh -c "echo BB-I2C1 > $SLOTS"
debian@beaglebone:/dev$ ls i2c*
i2c-0  i2c-1  i2c-2
debian@beaglebone:/dev$ i2cdetect -l
bash: i2cdetect: command not found
debian@beaglebone:/dev$ su
root@beaglebone:/dev# i2cdetect -l
i2c-0   i2c             OMAP I2C adapter                        I2C adapter
i2c-1   i2c             OMAP I2C adapter                        I2C adapter
i2c-2   i2c             OMAP I2C adapter                        I2C adapter
root@beaglebone:/dev# su debian
debian@beaglebone:/dev$ i2cdetect -l
bash: i2cdetect: command not found
debian@beaglebone:/dev$ sudo i2cdetect -l
i2c-0   i2c             OMAP I2C adapter                        I2C adapter
i2c-1   i2c             OMAP I2C adapter                        I2C adapter
i2c-2   i2c             OMAP I2C adapter                        I2C adapter
debian@beaglebone:/dev$
```

i2cdetect -y -r 0

```
root@beaglebone:~# i2cdetect -y -r 0
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- UU -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- UU -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: UU -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: UU -- -- -- -- -- -- --
root@beaglebone:~# i2cdetect -y -r 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- UU UU UU UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@beaglebone:~#
```

因为我的温湿度传感器接在I2C2（P9_20,P9_19）,所以在i2c-1上探测到地址为0x44的设备，即我的温湿度传感器。

SHT3x-DIS	I2C Address in Hex. representation	Condition
I2C address A	0x44 (default)	ADDR (pin 2) connected to logic low
I2C address B	0x45	ADDR (pin 2) connected to logic high

Table 7 I2C device addresses.

三、编写sht31温湿度传感器应用程序.

最终效果如下图：

```
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- 44 -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- UU UU UU UU -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
root@beaglebone:~# ./sht31
Starting the SHT31 test application
The data 0 is: 0x67
The data 1 is: 0xcd
The data 2 is: 0x71
The data 3 is: 0x9c
The data 4 is: 0xb5
The data 5 is: 0x32
The Temp data 0 is: 0x67
The Temp data 1 is: 0xcd
The Temp data 2 is: 0x71
The Hum data 0 is: 0x9c
The Hum data 1 is: 0xb5
The Hum data 2 is: 0x32
The temperature is 26   The humidity is 61
```

代码：

```
/** Simple I2C example to read the first address of a device in C
* Written by Derek Molloy for the book "Exploring BeagleBone: Tools and
* Techniques for Building with Embedded Linux" by John Wiley & Sons, 2014
* ISBN 9781118935125. Please see the file README.md in the repository root
* directory for copyright and GNU GPLv3 license information.          */

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

```
#include <unistd.h>

// Small macro to display value in hexadecimal with 2 places

#define SHT31_ADDR      0x44
#define BUFFER_SIZE 6

unsigned char RXData_Temp[3];
unsigned char RXData_Hum[3];
volatile int TXByteCtr;
volatile int RXByteCtr;

unsigned int temp_sht3x;
volatile float temperatureDegC_sht3x;
volatile float temperatureDegF_sht3x;
int degrees_sht3x;

unsigned int RH_sht3x;
volatile float Relative_Humidity_sht3x;
int humidity_sht3x;

char readBuffer[BUFFER_SIZE];

float temperature_converter_sht3x(void);

float humidity_converter_sht3x(void);

int data_copy(void);

int crc_detect(void);

void display(void);

int main(){
  int file;
  int n;
  char writeBuffer[2] = {0x2C, 0x06};

  printf("Starting the SHT31 test application\n");
  if((file=open("/dev/i2c-1", O_RDWR)) < 0){
    perror("failed to open the bus\n");
    return 1;
  }
  if(ioctl(file, I2C_SLAVE, SHT31_ADDR) < 0){
    perror("Failed to connect to the sensor\n");
    return 1;
  }

  while (1) {
     if(write(file, writeBuffer, 2)!=2){
       perror("Failed to reset the read address\n");
       return 1;
     }

     usleep(16000);

     if(read(file, readBuffer, BUFFER_SIZE)!=BUFFER_SIZE){
       perror("Failed to read in the buffer\n");
       return 1;
     }

     for (n = 0; n < BUFFER_SIZE; n++) {
        printf("The data %d is: 0x%02x\n", n, readBuffer[n]);
     }

     data_copy();
```

```
        if (crc_detect()) {
            printf("sensor data is error\n");
        }
        display();

        sleep(30);
    }

    close(file);

    return 0;
}

int data_copy(void)
{
    int n, m;
    for (m = 0; m < 6; m++) {
        if (m < 3)
            RXData_Temp[m] = readBuffer[m];
        else
            RXData_Hum[m-3] = readBuffer[m];
    }

    for (n = 0; n < 6; n++) {
        if (n < 3)
            printf("The Temp data %d is: 0x%02x\n", n, RXData_Temp[n]);
        else
            printf("The Hum data %d is: 0x%02x\n", n-3, RXData_Hum[n-3]);
    }

    return 0;
}

// Generator polynomial for CRC
#define POLYNOMIAL  0x131 // P(x) = x^8 + x^5 + x^4 + 1 = 100110001

static unsigned char SHT3X_CalcCrc(unsigned char data[], unsigned char nbrOfBytes)
{
    unsigned char bit;        // bit mask
    unsigned char crc = 0xFF; // calculated checksum
    unsigned char byteCtr;    // byte counter

    // calculates 8-Bit checksum with given polynomial
    for(byteCtr = 0; byteCtr < nbrOfBytes; byteCtr++) {
        crc ^= (data[byteCtr]);
        for(bit = 8; bit > 0; --bit) {
            if(crc & 0x80)
                crc = (crc << 1) ^ POLYNOMIAL;
            else
                crc = (crc << 1);
            }
    }

    return crc;
}

int crc_detect(void)
{
    unsigned char temp_crc, hum_crc;
    temp_crc = SHT3X_CalcCrc(RXData_Temp, 2);
    hum_crc = SHT3X_CalcCrc(RXData_Hum, 2);

    if (temp_crc != RXData_Temp[2])
        return 1;
    if (hum_crc != RXData_Hum[2])
        return 2;

    return 0;
```

```
}

float temperature_converter_sht3x(void)
{
   // Temperature in Celsius. See the Device Descriptor Table section in the
   // System Resets, Interrupts, and Operating Modes, System Control Module
   // chapter in the device user's guide for background information on the
   // used formula.
   temp_sht3x = RXData_Temp[0];
   temp_sht3x= temp_sht3x << 8 | RXData_Temp[1];
   temperatureDegC_sht3x = (long)temp_sht3x / 65535.0f * 175.0f - 45.0f;

   // Temperature in Fahrenheit Tf = (9/5)*Tc + 32
   temperatureDegF_sht3x = temperatureDegC_sht3x * 9.0f / 5.0f + 32.0f;
   return temperatureDegC_sht3x;
}

float humidity_converter_sht3x(void)
{
   RH_sht3x = RXData_Hum[0];
   RH_sht3x= RH_sht3x << 8 | RXData_Hum[1];
   Relative_Humidity_sht3x = (long)RH_sht3x / 65535.0f * 100.0f;
   return Relative_Humidity_sht3x;
}

void display(void)
{
   degrees_sht3x = (int)(temperature_converter_sht3x() + 0.5);

   humidity_sht3x = (int)(humidity_converter_sht3x() + 0.5);

   printf("The temperature is %d    The humidity is %d\n", degrees_sht3x, humidity_sht3x);
}
```