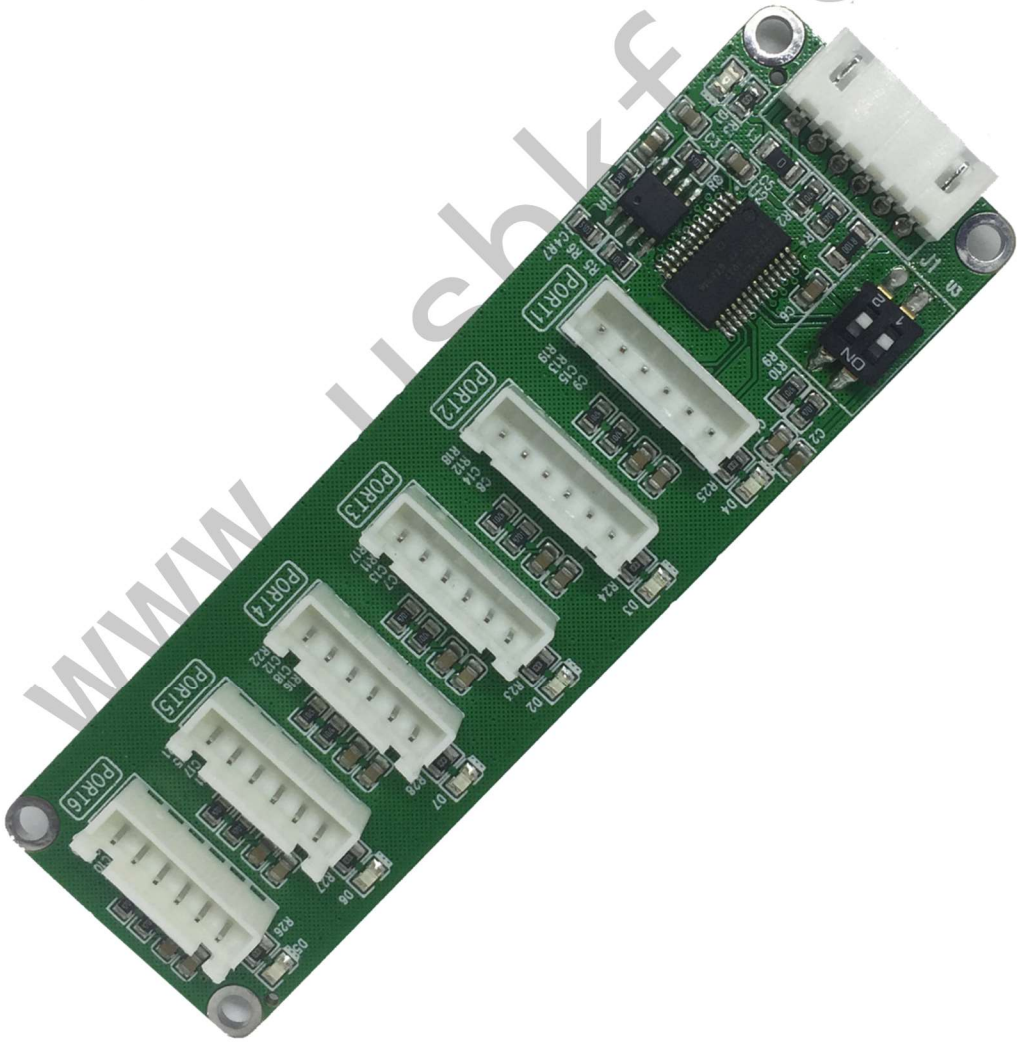
深圳市宏利自动化科技有限公司 WWW.HUNGLEETECH.COM
技术支持: USB 开发网 WWW.USBKF.COM WWW.USBKF.CN

IIC 扩展板说明书

→ IIC SPLITTER BOARD

一拖六 I2C/IIC 多路复用板（第 3 版）



2017-07-12

IICEXPANDERV3 SPEC		
V1.0	NEW ISSUE	2017-07-12
V1.1	UPDATED PAGE7	2017-07-13

勘误:

	*1. 目前在售版本为 3.1 版本，去掉了 OP 放大器电路部分。*

目 录

第一章

一拖六 I2C/IIC 多路复用板硬件结构

1.1 功能特点

IICEXPANDERV3 是一款通过 I2C 总线控制的六进制单向转换多路复用板， 具有响应速度快，抗干扰性能优异，专为 Channel Tester 一代测试头配套的数据传输的功能。

本扩展板是由深圳市宏利自动化科技有限公司设计的 IIC 扩展板系列第三代专用产品，在第一版本的基础上,增加了每个端口的电源控制,之外还增加有中断和复位管脚以便在硬件异常时上位机进行处理。本产品使用客户群广泛，设计之初是专门针对我司设计的多通道高灵敏度电容量测试板套件的一部分，经过大量客户长时间实际工业环境下使用来看，本板运行稳定可靠。

本板采用 1.6mm 纤维板材四层板设计，坚固耐用。

为保证运行可靠，基本核心部件全部采用贴片元件，可靠性高。端口直接供电，无须另置电源。

本产品提供：

（01）1 个连接 USB 桥接板的主接口

（02）6 个连接扩展的接口

（03）1 个接 USB 桥接板的复位引脚(暂未有功能)

（04）1 个接 USB 桥接板的中断引脚(暂未有功能)

（05）1 组 IIC 地址拨码开关选择，最多可提供 4 组地址在线选择。

（06）每个扩展的端口均可以单独控制电源供电与断开。

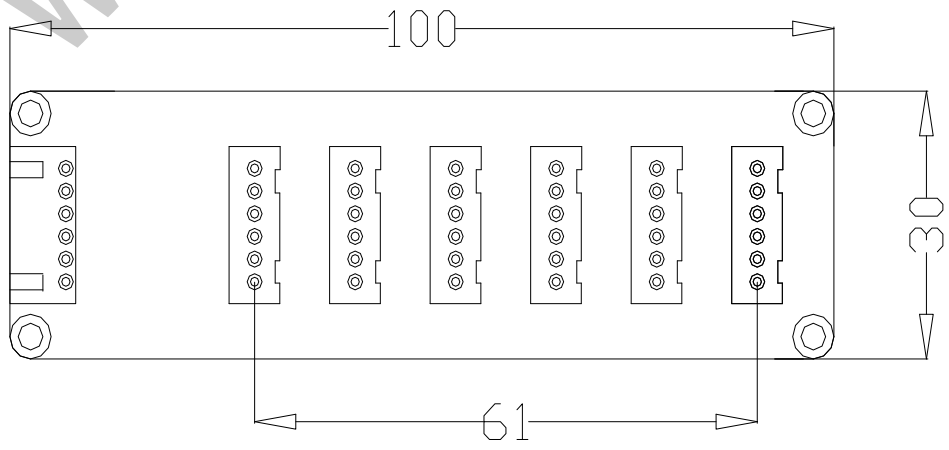
（07）每个扩展的端口均有电源指示灯。

1.2 硬件

本板硬件力求简洁明了，除去纷繁复杂的功能，硬件实现以最简洁的方式设计，用户可以快速的进行应用。

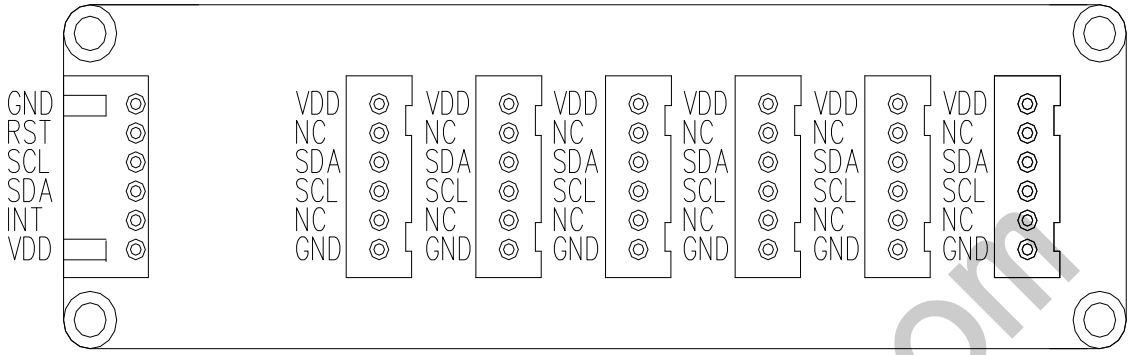
下面将会对各个部分予以简单介绍。

1.2.1 外形尺寸(mm)



1.2.2 接口定义

本板最端 HX-2.54 弯角白色连接器为 Main IIC 接口， 右边 6 个端口为 Extern IIC 接口。



	1	2	3	4	5	6
Main IIC Port	VDD	INT	SDA	SCL	/RST	GND
Extern IIC Port	GND	NC	SCL	SDA	NC	VDD

Main IIC Port (Control Address=0x40~0x43)		
序号	名称	说明
1	VDD	电源,电源电压运行时范围为 2.7V 至 5.5V.
2	INT①	中断输出管脚 (暂未有功能)
3	SDA	IIC 总线穿行数据信号,
4	SCL	IIC 总线穿行时钟信号,最高 400 kHz
5	/RST	低电平输入有效复位(暂未有功能)
6	GND	地
①当扩展 6 个端口任意有数据准备好时时，该引脚 Active LOW。		

Extern IIC Port		
序号	名称	说明
1	GND	地
2	NC	悬空

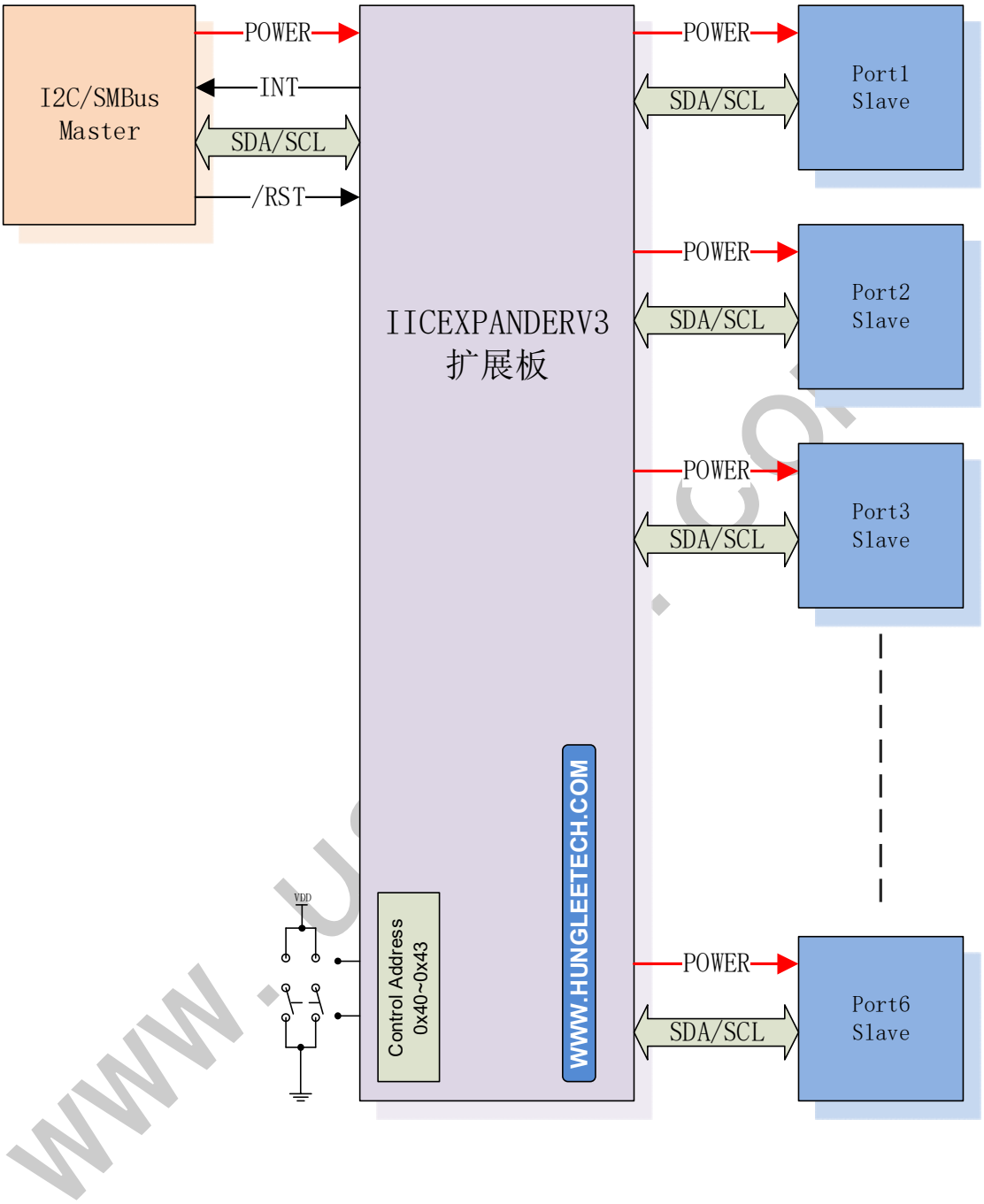
3	SCL	IIC 总线穿行时钟信号,最高 400 kHz
4	SDA	IIC 总线穿行数据信号,
5	NC	悬空
6	VDD	电源,电源电压运行时范围为 2.7V 至 5.5V.

1.2.3 电气性能

本板具备6 路单向转换的功能，具体的特性如下：

★ I2C 总线逻辑接口，兼容的SMBus 标准；
★ 低电压输入复位；
★ 4组拨码开关组合地址选择口允许I2C 总线连接多达4 个设备；
★ 在任意组合中，可以通过 I2C 总线选择通道；
★ 上电时所有开关通道被禁止；
★ 低导通电阻的多路复用器；
★ 转换器允许总线电压在3.3V 和5V 的电平之间转换；
★ 上电时无干扰脉冲信号；
★ 支持热插拔；
★ 待机电流更低；
★ 电源电压运行时范围为 2.7V 至5.5V ；
★ 5 伏宽压输入；
★ 0 Hz 到400 kHz 的时钟频率；
★ ESD 保护：JESD22-A114 为2000V HBM，JESD22-A115 为200V 和JESD22-C101为1000V CDM ；

框图如下：


I2C/SMBus
Master
POWER
INT
SDA/SCL
/RST
IICEXPANDERV3
扩展板
WWW.HUNGLEETECH.COM
VDD
Control Address
0x40~0x43
POWER
SDA/SCL
Port1
Slave
POWER
SDA/SCL
Port2
Slave
POWER
SDA/SCL
Port3
Slave
POWER
SDA/SCL
Port6
Slave

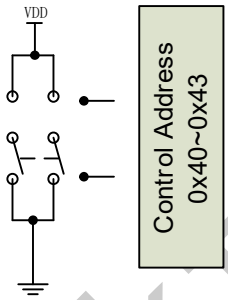
1.2.4 极限参数

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD}	supply voltage		–0.5	+7.0	V
V_I	input voltage		–0.5	+7.0	V
I_I	input current		-	±20	mA
I_O	output current		-	±25	mA
I_{DD}	supply current		-	±100	mA
I_{SS}	ground supply current		-	±100	mA
P_{tot}	total power dissipation		-	400	mW
T_{stg}	storage temperature		–60	+150	°C
T_{amb}	ambient temperature	operating	–40	+85	°C

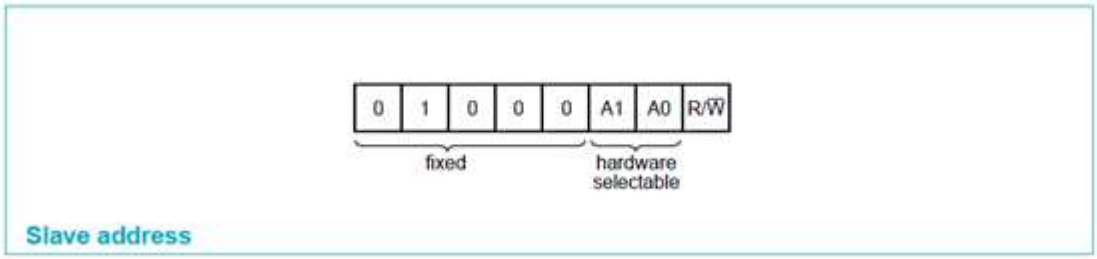
第二章
驱动方法

2.1 通讯协议

本扩展板 IIC 地址默认是 0x40~0x43，可以通过拨码开关地址调整 IIC 的默认控制寄存器地址。



器件地址如下：



控制寄存器

寄存器值	值	说明	读写性
CMD_GETINFO	0x01	获取扩展板信息	R

信息结构体如下：

```
#pragma  pack (push,1)
         struct ExpandBoardV3InfoStu
         {
              unsigned short version;//扩展板固件版本号
              unsigned short  portnumber;//可以扩展的端口数目
              unsigned char  sensortype;//测试头类型
              unsigned char  hwversion;//硬件版本号
              struct{
                   unsigned short year;
                   unsigned char month;
```

```
                unsigned char day;
            }vertime;//固件日期
        }ExpandBoardV3Info;
#pragma pack(pop)
```

CMD_EEPROMINFO	0x10	获取/设置扩展 NVRAM 中值的信息	R/W

信息结构体如下：

```
#pragma  pack (push,1)
        union ExpandBoardV3EEPROMInfoStu
        {
            struct{
                unsigned char bSerial[16];//序列号
                unsigned char bKey[16];//注册码
                unsigned char bDistributer[8];//发行商代码
                struct{
                    unsigned short year;
                    unsigned char month;
                    unsigned char day;
                }mfdtime;//生产日期
                struct{
                    unsigned short year;
                    unsigned char month;
                    unsigned char day;
                }exptime;//质保过期日期
            };
            struct{
                unsigned char unused[64];
            };
        }ExpandBoardV3EEPROMInfo;
#pragma pack(pop)
```

写入 EEPROM 时 Regs.para 需要设置 0xaa55 才可以写入。

CMD_GETRAWDATA	0x02	获取/设置测试头的 Rawdata	R/W

读写格式如下：

```
#pragma  pack (push,1)
        typedef struct I2C_Regs {    // Example I2C interface structure
            unsigned char bCMD = SWAPWORD(CMD_GETRAWDATA);
            unsigned short para = 0;
            unsigned short data[32];
        } MyI2C_Regs;
#pragma pack(pop)
```

读写成功 MyI2C_Regs. para = 1，否则为 0；

若读写成功，读到的 rawdata 返回在 unsigned short data[32];

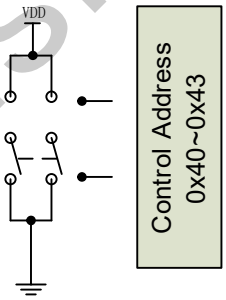
CMD_SETPOWER	0x03	获取/设置端口的电源状态	R/W

读写格式如下：

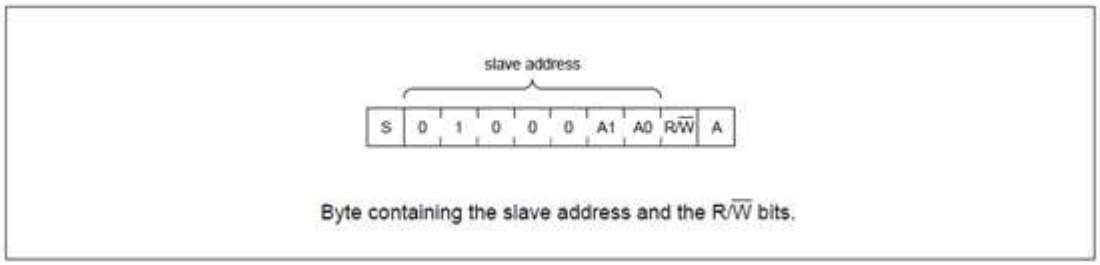
```
#pragma  pack (push,1)
        typedef struct I2C_Regs {    // Example I2C interface structure
            unsigned char bCMD = SWAPWORD(CMD_SETPOWER);
            unsigned short para = MAKEWORD((BYTE)fpower, bport);
             // fpower电源状态，1为开启电源，0为关闭电源
             // bport 端口号1~6；若端口号设置0，则写入fpower为一次性设置所有端口的电源状态
        } MyI2C_Regs;
#pragma pack(pop)
```

读写成功 MyI2C_Regs. para = 1，否则为 0；

本扩展板 PORT 的中断脚和复位脚控制 IIC 地址默认是(0x40)，可以通过拨码开关的地址调整 IIC 的默认控制寄存器地址。



器件地址如下：



2.2 示例代码

设定流程

上电->设定需要打开的端口 PORT 电源->设定需要打开的端口 PORT->主程序对端口的 IIC 器件进行读写(EEPROM,RTC,etc…)->设定需要关闭的端口->设定需要关闭的端口的电源(需要的话)-> ……

写入子程序头文件定义

```
#pragma  pack (push,1)
union ExpandBoardV3EEPROMInfoStu
{
    struct{
        unsigned char bSerial[16];
        unsigned char bKey[16];
        unsigned char bDistributer[8];
        struct{
            unsigned short year;
            unsigned char month;
            unsigned char day;
        }mfdtime;
        struct{
            unsigned short year;
            unsigned char month;
            unsigned char day;
        }exptime;
    };
    struct{
        unsigned char unused[64];
    };
}ExpandBoardV3EEPROMInfo;
#pragma pack(pop)


#pragma  pack (push,1)
typedef struct I2C_Regs {    // I2C interface structure
    unsigned char bDummy;
```

```
            unsigned char bCMD;
            unsigned short para;
            unsigned short data[32];
        } MyI2C_Regs;
        #pragma pack(pop)
    public:
        int getinfo(void);
        int geteeprominfo(void);
        int seteeprominfo(void);
        int setpower(unsigned char bport, bool fpower);
        int getrawdata(unsigned char bport, unsigned short* rawdata);
    }expandboard_v3;
```

写入子程序文件定义

```
Int expandv3::getinfo(void)//读取扩展板ROM信息
{
    int ret = 0;
    MyI2C_Regs Regs;
    Regs.bDummy = 0;
    Regs.bCMD = USER_IIC_CMD_GETINFO;
    Regs.para = 0;
    ret = I2cWrite(hDev,4, EXPANDBOARDV3_ADDRESS, (unsigned char*)&Regs);
    Sleep(50);
    if (ret) return ret;
    ret = I2cRead(hDev, sizeof(ExpandBoardV3Info), EXPANDBOARDV3_ADDRESS, (unsigned
char*)&ExpandBoardV3Info);
    unsigned char* pRegs = (unsigned char*)(void*)&ExpandBoardV3Info;
    if (ret) return ret;
    return SUCCESS;
}
Int expandv3::geteeprominfo(void) //读取扩展板NVRAM信息
{
    int ret = 0;
    MyI2C_Regs Regs;
    Regs.bDummy = 0;
    unsigned char* pRegs = (unsigned char*)(void*)&Regs;
    Regs.bCMD = CMD_EEPROMINFO;
    Regs.para = 0;

    ret = I2cWrite(hDev, 4, EXPANDBOARDV3_ADDRESS, (unsigned char*)&Regs);
    if (ret) return ret;
    Sleep(50);
```

```
    ret = I2cRead(hDev, sizeof(Regs), EXPANDBOARDV3_ADDRESS, (unsigned char*)&Regs);
    if (ret) return ret;
    unsigned char* buff = (unsigned char*)&Regs;
    buff++; buff++; buff++;
    memcpy((unsigned char*)&ExpandBoardV3EEPROMInfo, (unsigned
char*)(void*)&buff[0], sizeof(ExpandBoardV3EEPROMInfo));
    return SUCCESS;
}
Int expandv3::seteeprominfo(void)
//写入扩展板NVRAM信息，需要写入的信息预先存放ExpandBoardV3EEPROMInfo
{
    int ret = 0;
    MyI2C_Regs Regs;
    Regs.bDummy = 0;
    Regs.bCMD = CMD_EEPROMINFO;
    Regs.para = 0xaa55;

    memcpy((unsigned char*)(void*)&Regs.data, (unsigned
char*)&ExpandBoardV3EEPROMInfo, sizeof(ExpandBoardV3EEPROMInfo));
    ret = I2cWrite(hDev, sizeof(Regs), EXPANDBOARDV3_ADDRESS, (unsigned
char*)&Regs);
    Sleep(50);
    if (ret) return ret;
    return SUCCESS;
}
Int expandv3::setpower(unsigned char bport, bool fpower)
{
    int ret = 0;

    MyI2C_Regs Regs;
    Regs.bDummy = 0;
    Regs.bCMD = CMD_SETPOWER;
    Regs.para = MAKEWORD((BYTE)fpower, bport);
    ret = I2cWrite(hDev, 4, EXPANDBOARDV3_ADDRESS, (unsigned char*)&Regs);
    if (ret) return ret;
}
Int expandv3::getrawdata(unsigned char bport, unsigned short* rawdata)
{
    int ret = 0;
    MyI2C_Regs Regs;
    Regs.bDummy = 0;
    Regs.bCMD = CMD_GETRAWDATA;
    Regs.para = SWAPWORD(bport);
```

```
    ret = I2cWrite(hDev, 4, EXPANDBOARDV3_ADDRESS,  (unsigned char*)&Regs);
    if (ret) return ret;
    Sleep(23);

    ret = I2cRead(hDev, sizeof(Regs), EXPANDBOARDV3_ADDRESS, (unsigned char*)&Regs);
    if (ret) return ret;
    unsigned char* buff = (unsigned char*)&Regs;
    unsigned char* buff1 = (unsigned char*)&Regs;
    buff++; buff++; buff++;
    memcpy(rawdata, (unsigned char*)(void*)&buff[0], sizeof(Regs.data));
    for (int i = 0; i<32; i++) {
        rawdata[i] = SWAPWORD(rawdata[i]);
    }
    return (buff1[2]==1)?(SUCCESS):(FAIL);
}
```

第三章

测试套件

高精度多路悬浮电容量测量系统是一款测量多路微小电容量的测试系统，本测试系统具有精度高，测量稳定，测量通道灵活扩展的功能，广泛用于电容式触摸屏 Sensor 测试，制药过程中溶液电介质变化监控，油料位置高低控制等应用范围，且经过广大客户多年来批量验证，测量可靠、稳定。

整套系统包含，上位机软件，USB 桥接板，扩展板，测试头四部分构成。

具体信息请到 www.hungleetech.com 查看详细信息， 或者直接到淘宝搜索店铺号 20752076 进行购买。

第四章
说明

5.1 硬件

1．不可经受剧烈撞击。
2．适宜在室内环境工作
3．不可经受高温、高湿、盐雾、霉菌等极端环境。

5.2 软件

所提供的软件仅仅为提供测试之用，如果用户将此板相关代码用在您的产品当中，您就接受了可能存在缺陷的危险存在，对此危险和缺陷本板的开发方不承担任何责任和义务。

5.3 开发文档

本文档属于附赠内容，不属于必须内容，对此可能出现的任何遗漏、缺陷等等问题不予承担任何责任。