

IPD 调节说明文档

V3.0.0

瑞声科技控股有限公司

目录

1、简介	4
2、 I2C 设计	4
2.1 I2C 的 gpio 配置	4
2.2 I2C 配置	4
3、 相关寄存器	4
3.1 运动控制寄存器	4
3.2 位置查询寄存器	5
3.3 hall 标定寄存器	5
3.4 瞳距调节寄存器	6
3.5 I2C 固件升级寄存器 1	6
3.6 I2C 固件升级寄存器 2	6
3.7 I2C 固件升级校验寄存器 1	7
3.8 I2C 固件升级校验寄存器 2	7
3.9 I2C 升级停止寄存器	7
4、 ipd 电机运动实现	8
4.1 电机运动速度	8
4.2 上报 ipd 位置	8
5、 I2C 固件升级	8
5.1 流程	8
1) 上电 10 秒内	8
2) 上电后	8
5.2 示例	9

修改历史

版本号	修改内容	修改人	日期
1.0.0	初稿	黄欣然	2023/10/09
2.0.0	第二版	黄欣然	2024/1/15
3.0.0	第三版	黄欣然	2024/1/26

1、简介

IPD SDK，是为第三方应用开发者提供的调用程序，能极大的方便开发者应用 ARGLASS 瞳距调节相关功能。

2、I2C 设计

IPD SDK 是 I2C 从机的程序。采用中断的方式接收数据。

2.1 I2C 的 gpio 配置

gpio 的电压域为 1.8 v。

2.2 I2C 配置

I2C 速度支持标速模式（100 KHz），支持 7 位的地址模式。

I2C 从机 7 位地址 0x72。

3、相关寄存器

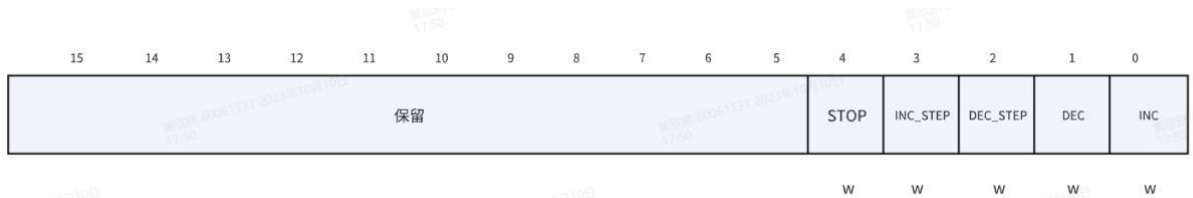
3.1 运动控制寄存器

地址偏移：0x01

复位值：0x0000

位/位域	名称	描述
15:5	保留	必须保持复位值；
4	STOP	此位置 0 时，电机停止运动；
3	INC_STEP	此位置 1 时，左镜头向左运动 1.6mm，同时右镜头向右运动 1.6mm；此位置 0 时，左镜头向左运动 0.1mm，同时右镜头向右运动 0.1mm；
2	DEC_STEP	此位置 1 时，左镜头向右运动 1.6mm，同时右镜头向左运动 1.6mm；此位置 0 时，左镜头向右运动 0.1mm，同时右镜头向左运动 0.1mm；

1	DEC	此位置 1 时，表示左镜头向左运动，右镜头向右运动；
0	INC	此位置 1 时，表示左镜头向右运动，右镜头向左运动；



说明：注意 0 和 1 位、2 和 3 位、stop 和其他位不能同时置 1。

3.2 位置查询寄存器

地址偏移：0x02

复位值：0x00

说明：I2C 主机接收的数据是 ipd 位置。

例如：55 表示当前瞳距 5.5mm，225 表示瞳距 22.5mm。

3.3 ha11 标定寄存器

地址偏移：0x07

复位值：0x00



位/位域	名称	描述
------	----	----

7:1	保留	必须保持复位值；
-----	----	----------

0	flag	此位置 1 时，表示 hall 重新标定整个行程的极限位；
---	------	-------------------------------

说明：在发现行程偏差、换磁铁、电机模组初始位置变化时，为确保行程精确，必须重新标定。

3.4 瞳距调节寄存器

地址偏移：0x08

复位值：0x00

有效数据范围：55-225

说明：I2C 主机写该寄存器的值应该是期望瞳距的值。

例如：发送 55 表示移动到瞳距 5.5mm，发送 225 表示移动到瞳距 22.5mm。

3.5 I2C 固件升级寄存器 1

地址偏移：0xAA

复位值：0x00



位/位域	名称	描述
7:1	保留	必须保持复位值；
0	flag	此位置 1 时，表示需要升级；

3.6 I2C 固件升级寄存器 2

地址偏移：0x55

复位值：0x00



位/位域	名称	描述
7:1	保留	必须保持复位值；
0	flag	此位置 1 时，表示从机可以开始接收主机发送的升级数据；

3.7 I2C 固件升级校验寄存器 1

地址偏移：0xDD

复位值：0x00

说明：用于主机发送数据的校验值

3.8 I2C 固件升级校验寄存器 2

地址偏移：0xEE

复位值：0x00

说明：主机读取该寄存器的值，读出 0xee，代表数据校验正确；读出 0x01，代表校验错误。

3.9 I2C 升级停止寄存器

地址偏移：0x66

复位值：0x00

说明：次寄存器仅用于刚上电(10s 内)，主机向该寄存器发送 0x01，则不进入升级模式；

4、ipd 电机运动实现

控制 ipd 电机，系统启动后，主机通过 I2C 控制寄存器以开始运动，运动过程中如触发极限位则停止运动。

4.1 电机运动速度

单向距离为 8.5mm, 电机速度设置为 10.625 mm/s。

4.2 上报 ipd 位置

从位置查询寄存器读取位置值 pos。

pos=55 表示瞳距 55mm；

pos=225 表示瞳距 225mm；

5、I2C 固件升级

5.1 流程

进入升级模式方法：

1) 上电 10 秒内

1. 主机向 I2C 固件升级寄存器 2 (0x55) 发送 0x01 提示从机开始接收 I2C 数据，然后主机睡眠 10ms；

2) 上电后

1. 主机向 I2C 固件升级寄存器 1 (0xAA) 发送 0x01 升级命令，然后主机睡眠 0.5s；

2. 主机向 I2C 固件升级寄存器 2 (0x55) 发送 0x01 提示从机开始接收 I2C 数据，然后主机睡眠 10ms；

进入升级模式后：

1. 主机发送接下来一次发送的字节个数（一次发 1024 个字节，字节个数分两个 uint_8 发送[比如 1024 个发送 0x04, 0x00]，不足 1024 的放在最后一次发送，如果最后一次也是 1024，发完最后一次后，再发送[0x00, 0x00]），然后主机睡眠 10ms；
2. 向 0xdd 寄存器发校验应得的值，然后主机睡眠 10ms；
3. 主机发送数据，然后主机睡眠 10ms；
4. 读 0xee 寄存器，如果读出 0xee，代表校验正确，如果读出 0x01，代表校验错误，重新从第 3 步开始发送这包数据即可；
5. 重复 1、2、3、4 步骤，分多次发送完成升级；

注意：各步骤需要主机睡眠延时，给从机响应时间。

5.2 示例

上电后升级流程

1. 发送 0xAA 0x01

```
ipd_i2c_write(ipd, ipd->i2c->addr, IPD_I2C_TYPE_BYTE, IPD_I2C_TYPE_BYTE,  
IPD_UPDATA_FLAG_REG, arg);
```

```
usleep(500000);
```

2. 发送 0x55 0x01

```
ipd_i2c_write(ipd, ipd->i2c->addr, IPD_I2C_TYPE_BYTE, IPD_I2C_TYPE_BYTE,  
IPD_UPDATA_SEND_REG, arg);
```

```
usleep(10000);
```

3. 0x0400 表示发送 1024 个字节

```
ipd_i2c_write(ipd, ipd->i2c->addr, IPD_I2C_TYPE_BYTE, IPD_I2C_TYPE_BYTE,  
0x04, 0x00);
```

```
usleep(10000);
```

4. 向 0xDD 寄存器发校验应得的值，然后主机睡眠 10ms；

```
ipd_i2c_write(ipd, ipd->i2c->addr, IPD_I2C_TYPE_BYTE, IPD_I2C_TYPE_BYTE,  
0xDD, 0x32);
```

5. 1024 字节数据存储在一个 buffer 发送

```
ipd_i2c_write_data(ipd, ipd->i2c->addr, buffer);  
usleep(10000);
```

6. 读 0xEE 寄存器

```
i2cget -f -y 2 0x72 0xEE b
```