**数字舵机液晶控制器需求说明书**

—深圳安哲汇智科技有限公司

1 功能概述



液晶控制器采用串口通讯方式，分别与上位机控制软件和数字舵机进行通讯，实现数字舵机的参数进行读取、下载、编辑、管理和显示等功能。

2 框图



3性能要求

输入电压： DC5V – 8.4V

输出电压： DC 5V – 8.4V

输出电流： DC 2A

通讯速率： 9600/19200

舵机接口： 单线半双工通讯方式（三线连接 GND/VDD/RXTX）

上位机接口： USB接口

显示屏： 1602字符型液晶

按键： 4个

存储空间： 100K（I2C—EEPROM）

4 功能描述

4.1 存储空间管理功能

存储空间： AT24C128 （16384\*8） I2C接口

参数组空间： 64\*2\*8 （可存储128参数组）

液晶空间： 1参数组 空间编码0 【液晶存储参数空间】

出厂空间： 8参数组 空间编码1-6 【出厂存储参数空间】

客户空间： 100参数组 空间编码9-108 【客户存储参数空间】

用户空间： 20参数组 空间编码109 -128 【用户存储参数空间】

存储空间权限：

出厂权限（1）：可读写全部空间，

客户权限（2）：可读写客户空间和用户空间，只读出厂空间

用户权限（3）：可读写用户空间，只读客户空间和出厂空间

4.1 上位机交互功能

4.1.1 波特率设置功能

设置与上位机通讯波特率 19200（默认）/9600

4.1.2 上位机查询

上位机发送查询指令，液晶控制器直接查询舵机参数，并反馈上位机

4.1.3 上位机指令

上位机发送指令，液晶控制器直接将指令发送舵机

4.1.4 上位机管理

上位机发送权限，空间编码，液晶控制器读写存储空间数据，并反馈结果

**存储地址编码说明**

编码0 对应出厂空间编码 0

权限 1 编码1-6 对应出厂空间编码 1-6

权限 2 编码 1-99 对应客户空间编码 9-108

权限 3 编码 1-19 对应用户空间编码 109 -128

4.2 数字舵机交互功能

4.2.1 液晶控制器读取舵机参数

4.2.2 液晶控制器下载校验舵机参数

4.2.3 液晶控制器读写存储空间

4.2.4 液晶控制器控制舵机动作

4.2.5 液晶控制器自动检测舵机 （插入舵机以后，检测信号线电平变化）

4.3 按键功能

从左到右 A——B——C——D

A键: 确认键

B键: 上翻键

C键: 下翻键

D键: 退出键

4.4 菜单显示功能

4.4.1 开机显示

显示LOGO(POWEHD) 延时2秒，然后自动显示Please Connect SERVO

如果检测到有舵机连接，则显示Connect OK? Y

舵机拔出或者N键有效话，显示Please Connect SERVO

Y确认键有效话，则读取舵机参数，并显示 SERVO READING

读取完成后，显示舵机程序版本号 SERVO XXXXX

等待A键输入 进入下级菜单 默认显示ServoSetting 菜单

4.4.2 ServoSetting 菜单

第一行显示舵机型号 （读取舵机版本号 5位整数值，具体附录说明）

第二行显示子菜单

01 MAXPOW servo\_max\_pwm\_set [500-1500]

02 BOOST servo\_work\_base\_voltage [0-100]

03 DBAND servo\_zero\_zone\_set [2-10]

04 FORCE servo\_position\_pid\_parm\_p\_set [1-10]

05 STERETC servo\_speed\_pid\_parm\_p\_set [1-10]

06 BRAKE servo\_speed\_run\_sample\_k\_set [1-100]

07 SFTSTA servo\_init\_flag\_set [0/1]

注：以上参数值按照0-100%比例显示设置，如1-10 对应 10%=1 100%=10 参数

08 Setting Data SAVE (用户区间)

01 Write Servo

02 Write Memory 文件名显示：UDATA\_ XXX00 — UDATA\_ XXX19

09 Setting Data READ (用户区间)

01 Read Servo

02 Read Memory 文件名显示：UDATA \_XXX00 — UDATA\_ XXX19

10 Reset Data READ (客户区间)

01 Reset Factory 文件名显示：FDATA\_ X00XX — FDATA\_X99XX

注：文件名编号规则

舵机反馈版本: servo\_program\_version = XXXXX

万数位：程序版本 0-6

千百位：工厂版本 00-99

个十位：用户版本 00-19

例如：10110 即程序版本1 工厂版本01 用户版本10

4.4.3 SendPosition 菜单

01 Broadband servo

自动发送舵机到中间位置指令

B键按下后，自动切换最高位置，中间位置，最低位置循环

02 Narrowband servo(无法识别)

自动发送舵机到中间位置指令

C键按下后，自动切换最高位置，中间位置，最低位置循环

4.4.4 Information 菜单

01 显示编程卡软件版本 Version Ver\_XXXXX

02 显示舵机程序版本 Servo Ver\_XXXXX

4.4.5 LCD\_setting液晶设置

01 Servo\_baudrate\_set

01 19200(默认)

02 9600

03 38400

04 115200

02 Host\_baudrate\_set

01 19200(默认)

02 9600

03 38400

04 115200

03 LCD\_upgrade

4.4.6 液晶掉电时，自动将液晶临时内存空间参数保存外部内存空间。

液晶上电时，自动从外部内存空间参数读取到液晶临时内存空间

5 软件架构



5.1消息处理模块

5.1.1人机交互指令处理

功能： 生产外部内存管理指令

生成舵机控制指令（反馈/下发）并发送到通讯协议处理模块

返回液晶显示数据并发送到人机交互处理模块

5.1.2上位机指令处理

功能： 生产外部内存管理指令

生成上位机指令并发送到通讯协议处理模块

生成舵机控制指令（反馈/下发）并发送到通讯协议处理模块

5.1.3外部内存指令处理

功能： 外部内存管理指令，解析内存逻辑地址并发送到外部内存管理模块

5.1.4任务消息调度模块

功能：管理消息任务的调度

5.2 液晶显示模块

根据输入的界面编号和显示数值，驱动液晶显示屏

5.3 按键处理模块

采取轮询方式，获取键值编号

5.4 人机交互处理模块

处理显示界面和按键信号处理，并接收消息处理模块指令

5.4.1 生成人交互指令

根据当前的显示界面编号和键值，生成人机交互指令，发送到消息处理模块

5.4.2 生成界面显示指令

根据显示界面编号和显示数据，生成液晶显示指令

5.5 内存管理处理模块

根据内存的参数逻辑首地址和偏移地址，生成外部内存的物理地址，进行对数据的读写，并可控制读写的周期间隔

5.6 I2C通讯处理模块

5.6.1 读外部内存空间数据

5.6.2 写外部内存空间数据

5.7 通讯协议处理模块

5.7.1 解析上位机数据包，生成人机交互指令

5.7.2 根据消息处理模块的上位机指令，进行数据封装

5.7.3 解析舵机数据包，生成舵机反馈结果指令

5.7.2 根据消息处理模块的舵机控制指令，进行数据封装

5.8 USB通讯处理模块

5.8.1 发送USB数据包

5.8.2 接收USB数据包

5.9 UART通讯处理模块

5.9.1 发送UART数据包

5.9.2 接收UART数据包

**5.10 新增加指令**

**上位机发送液晶控制器权限和编码指令**

指令码 SERVO\_COMMAND\_SET\_ADDRESS 31 发送权限和编码命令字

参数ID1 MENU\_SET\_SERVO\_MEMORY\_PRIORITY 160 权限参数ID [1-3]

参数ID2 MENU\_SET\_SERVO\_MEMORY\_CODE 161 内存首地址编号

校验和 31+1+160 = 192 = C0

**上位机发送指令：FA 10 15 01 A0 XX XX A1 XX XX C0 FE**

**液晶内存数据下载校验完成后，将结果反馈给上位机指令**

**当地址输入9-108范围内，将参数下载完成后，液晶会反馈如下指令**

指令码 MEMORY\_COMMAND\_DOWMLOAD\_RETURN 32 发送权限和编码命令字

参数ID1 MEMORY\_WORK\_PARM 162 权限参数ID [1-3]

参数VAL1 0 = 校验失败 1 = 校验成功

参数ID2 MEMORY\_DEBUG\_PARM 163 内存首地址编号

参数VAL2 内存首地址编号

校验和 31+2+162 = 195 = C3

**液晶发送上位机指令：FA 10 15 02 A2 XX XX A3 XX XX C3 FE**

**液晶参数下载到舵机，并校验完成后，将结果反馈给上位机指令**

**当地址输入1-6范围内，将参数下载完成后，液晶会反馈如下指令**

指令码 SERVO\_COMMAND\_DOWMLOAD\_RETURN 33 发送权限和编码命令字

参数ID1 SERVO\_WORK\_PARM 164 权限参数ID [1-3]

参数VAL1 0 = 校验失败 1 = 校验成功

参数ID2 SERVO\_DEBUG\_PARM 165 内存首地址编号

参数VAL2 内存首地址编号

校验和 33+2+164 = 199 = C7

**液晶发送上位机指令：FA 10 15 02 A4 XX XX A5 XX XX C7 FE**

**注：当权限为0 地址为0 的时候，将上位机的下载参数下发到舵机**

2.1.1 增加指令

二类参数查询增加指令

servo\_speed\_base\_voltage\_set

FA 10 09 01 7F 00 00 00 00 00 89 FE

FA 10 09 02 7F 00 00 7F 00 00 90 FE 舵机反馈

servo\_integral\_time\_set

FA 10 09 01 80 00 00 00 00 00 8A FE

FA 10 09 02 80 00 00 80 00 00 8B FE 舵机反馈

二类参数下载指令

servo\_speed\_base\_voltage\_set

FA 10 07 01 43 00 01 00 00 00 4B FE

servo\_integral\_time\_set

FA 10 07 01 44 00 01 00 00 00 4C FE

**6 文件结构**

**6.1Main.c根据输出的指令进行消息处理**

System\_init();

Menu\_command\_process();

Memory\_command\_process();

Host\_command\_process();

Servo\_command\_process();

**6.2Key\_check.c检测按键，去抖后生成按键值**

按键值检测

# define KEY\_BUTTON\_OK 1

# define KEY\_BUTTON\_UP 2

# define KEY\_BUTTON\_DOWN 3

# define KEY\_BUTTON\_ESC 4

输出

Unsigned char key\_value\_now

Void key\_scan();

**6.3 Lcd\_disp.c 根据界面显示编号和显示数据，在LCD上显示字符**

界面显示编号

#define MENU\_DISP \_LOGO 00

#define MENU\_DISP \_CONNECT 11

#define MENU\_DISP \_READING 12

#define MENU\_DISP \_WRITING 13

#define MENU\_DISP \_OK 14

#define MENU\_DISP \_ERROR 15

#define MENU\_DISP\_FILE\_NAME 250 显示文件名

#define MENU\_DISP\_DATA\_VALUE 251 显示数值

#define MENU\_DISP \_CLASS\_SETTING 100

#define MENU\_DISP \_CLASS\_MAXPOW 110

#define MENU\_DISP \_CLASS\_BOOST 120

#define MENU\_DISP \_CLASS\_DBAND 130

#define MENU\_DISP \_CLASS\_FORCE 140

#define MENU\_DISP \_CLASS\_STERETC 150

#define MENU\_DISP \_CLASS\_BRAKE 160

#define MENU\_DISP \_CLASS\_SFTSTA 170

#define MENU\_DISP \_CLASS\_DATAWRITE 200

#define MENU\_DISP \_CLASS\_WRITE\_SERVO 201

#define MENU\_DISP \_CLASS\_WRITE\_MEMORY 202

#define MENU\_DISP \_CLASS\_DATAREAD 210

#define MENU\_DISP \_CLASS\_READ\_SERVO 211

#define MENU\_DISP \_CLASS\_READ\_MEMORY 212

#define MENU\_DISP \_CLASS\_DATARESET 220

#define MENU\_DISP \_CLASS\_RESET\_SERVO 221

#define MENU\_DISP \_CLASS\_RESET\_MEMORY 222

#define MENU\_DISP \_CLASS\_POSITION 230

#define MENU\_DISP \_CLASS\_PULSE\_MID 231

#define MENU\_DISP \_CLASS\_PULSE\_UPPER 232

#define MENU\_DISP \_CLASS\_PULSE\_LOWER 233

#define MENU\_DISP \_CALSS\_INFORMATION 240

#define MENU\_DISP \_CALSS\_LCD\_VERSION 241

#define MENU\_DISP \_CALSS\_LDC\_UPGRADE 242

#define MENU\_DISP \_CALSS\_SERVO\_VERSION 243

根据当前显示菜单，实时更新菜单项

Unsigned char menu\_disp\_one\_level\_value

Unsigned char menu\_disp\_two\_level\_value

Unsigned char menu\_disp\_three\_level\_value

输入：

Unsigned int menu\_disp\_num

Signed int menu\_disp\_value

根据菜单编码，进行界面显示，分两行显示

Void LCD\_disp\_one\_row (unsigned int menu\_disp\_one\_row\_num) 第一行显示字符

Void LCD\_disp\_two\_row (unsigned int menu\_disp\_two\_row\_num) 第二行显示字符

Void LCD\_disp\_data\_value (signed int menu\_disp\_value) 第二行5位数据

Void LCD\_disp\_file\_name(signed int menu\_disp\_value ) 第二行文件编号

Void LCD \_menu\_disp()

举例

开机界面

第一行 MENU\_DISP \_LOGO

第二行 MENU\_DISP \_CONNECT

MAXPWO 参数

第一行 MENU\_DISP \_CLASS\_110\_MAXPOW

第二行 MENU\_DISP\_DATA\_VALUE

显示 显示下载文件

第一行 MENU\_DISP \_CLASS\_202\_WRITE\_MEMORY

第二行 MENU\_DISP\_FILE\_NAME

**6.4 Menu\_interact.c 根据菜单和键值，生成指令码和显示数据**

人机交互指令码

#define MENU\_ COM \_SERVO\_TEST\_PULSE\_UPPER 10 最大脉宽

#define MENU\_ COM \_ SERVO\_TEST\_PULSE\_MID 11 中间脉宽

#define MENU\_ COM \_ SERVO\_TEST\_PULSE\_LOWER 12 最小脉宽

#define MENU\_ COM \_ SERVO\_READ\_WORK\_PARM 20 读取舵机工作参数

#define MENU\_ COM \_ SERVO\_READ\_DEBUG\_PARM 21 读取舵机配置参数

#define MENU\_ COM \_ SERVO\_READ\_ALL\_PARM 22 读取舵机所有参数

#define MENU\_ COM \_ SERVO\_ CHECKSUM \_ALL\_PARM 23 读取校验舵机参数

#define MENU\_ COM \_ SERVO\_WRITE\_WORK\_PARM 30 下载舵机工作参数

#define MENU\_ COM \_ SERVO\_WRITE \_DEBUG\_PARM 31 下载舵机配置参数

#define MENU\_ COM \_ SERVO\_ PROGRAM \_ALL\_PARM 32 下载 读取 校验

#define MENU\_ COM \_ MEMORY\_READ\_WORK\_PARM 40 读取内存工作参数

#define MENU\_ COM \_ MEMORY \_READ\_DEBUG\_PARM 41 读取内存配置参数

#define MENU\_ COM \_ MEMORY \_READ\_ALL\_PARM 42 读取内存所有参数

#define MENU\_ COM \_ MEMORY \_ CHECKSUM \_ALL\_PARM 43 读取校验内存参数

#define MENU\_ COM \_ MEMORY \_WRITE\_WORK\_PARM 50 下载内存工作参数

#define MENU\_ COM \_ MEMORY \_WRITE \_DEBUG\_PARM 51 下载内存配置参数

#define MENU\_ COM \_ MEMORY \_ PROGRAM \_ALL\_PARM 52 下载 读取 校验

#define MENU\_ COM \_ SERVO\_RESET\_FACTORY 60 复位舵机客户参数

#define MENU\_ COM \_ SERVO\_RESET\_ORGINAL 61 复位舵机出厂参数

输入：

Unsigned char menu\_disp\_one\_level\_value

Unsigned char menu\_disp\_two\_level\_value

Unsigned char menu\_disp\_three\_level\_value

Unsigned char key\_value\_now

输出：

Unsigned int menu\_command\_value

Unsigned int menu\_disp\_num

Signed int menu\_disp\_value

**6.5 Host\_protocol.c 根据上位机通讯协议，获取上位机的指令**

上位机指令码

#define HOST\_ COM \_SERVO\_TEST\_PULSE\_UPPER 110 最大脉宽

#define HOST \_ COM \_ SERVO\_TEST\_PULSE\_MID 111 中间脉宽

#define HOST \_ COM \_ SERVO\_TEST\_PULSE\_LOWER 112 最小脉宽

#define HOST \_ COM \_ SERVO\_READ\_WORK\_PARM 120 读取舵机工作参数

#define HOST \_ COM \_ SERVO\_READ\_DEBUG\_PARM 121 读取舵机配置参数

#define HOST \_ COM \_ SERVO\_READ\_ALL\_PARM 122 读取舵机所有参数

#define HOST \_ COM \_ SERVO\_ CHECKSUM \_ALL\_PARM 123 读取校验舵机参数

#define HOST \_ COM \_ SERVO\_WRITE\_WORK\_PARM 130 下载舵机工作参数

#define HOST \_ COM \_ SERVO\_WRITE \_DEBUG\_PARM 131 下载舵机配置参数

#define HOST \_ COM \_ SERVO\_ PROGRAM \_ALL\_PARM 132 下载 读取 校验

#define HOST \_ COM \_ MEMORY\_READ\_WORK\_PARM 140 读取内存工作参数

#define HOST \_ COM \_ MEMORY \_READ\_DEBUG\_PARM 141 读取内存配置参数

#define HOST \_ COM \_ MEMORY \_READ\_ALL\_PARM 142 读取内存所有参数

#define HOST \_ COM \_ MEMORY \_ CHECKSUM \_ALL\_PARM 143 读取校验内存参数

#define HOST \_ COM \_ MEMORY \_WRITE\_WORK\_PARM 150 下载内存工作参数

#define HOST \_ COM \_ MEMORY \_WRITE \_DEBUG\_PARM 151 下载内存配置参数

#define HOST \_ COM \_ MEMORY \_ PROGRAM \_ALL\_PARM 152 下载 读取 校验

#define HOST \_ COM \_ SERVO\_RESET\_FACTORY 160 复位舵机客户参数

#define HOST \_ COM \_ SERVO\_RESET\_ORGINAL 161 复位舵机出厂参数

#define HOST \_ COM \_ FEEDBACK\_ALL\_PARM 170 反馈液晶内存参数

输入

上位机通讯输入数据包，获取上位机指令码

Unsigned int host\_command\_value

输出

更新上位机数据

Unsigned int host\_feedback\_data

Void host \_command\_feedback\_all\_parm();

Void host \_command\_feedback\_all\_data();

void host\_command\_check();

**6.6 Servo\_protocol.c 根据指令码和舵机通信协议，进行舵机控制**

舵机指令码

#define SERVO \_ COM \_TEST\_PULSE\_UPPER 210 最大脉宽

#define SERVO \_ COM \_ TEST\_PULSE\_MID 211 中间脉宽

#define SERVO \_ COM \_ TEST\_PULSE\_LOWER 212 最小脉宽

#define SERVO \_ COM \_ READ\_WORK\_PARM 220 读取舵机工作参数

#define SERVO \_ COM \_ READ\_DEBUG\_PARM 221 读取舵机配置参数

#define SERVO \_ COM \_ READ\_ALL\_PARM 222 读取舵机所有参数

#define SERVO \_ COM \_ CHECKSUM \_ALL\_PARM 223 读取校验舵机参数

#define SERVO \_ COM \_ WRITE\_WORK\_PARM 230 下载舵机工作参数

#define SERVO \_ COM \_ WRITE \_DEBUG\_PARM 231 下载舵机配置参数

#define SERVO \_ COM \_ PROGRAM \_ALL\_PARM 232 下载 读取 校验

#define SERVO \_ COM \_ SERVO\_RESET\_ORGINAL 240 复位舵机出厂参数

#define SERVO \_ COM \_ FEEDBACK\_ALL\_PARM 270 反馈液晶内存参数

输入：指令码

输出：舵机通讯数据包

Unsigned int servo\_command\_value

Void servo\_command\_test\_plus\_upper();

Void servo \_command\_test\_plus\_mid();

Void servo \_command\_test\_plus\_lower();

Void servo \_command\_read\_work\_parm();

Void servo \_command\_read\_debug\_parm();

Void servo \_command\_read\_all\_parm();

Void servo \_command\_checksum\_all\_parm();

Void servo \_command\_write\_work\_parm();

Void servo \_command\_write\_degug\_parm();

Void servo \_command\_program\_all\_parm();//将操作结果反馈给上位机

Void servo\_command\_servo\_reset\_orginal();

Void servo \_command\_feedback\_all\_parm();

void servo\_command\_check();

**6.7 Memory\_**[**man.c**](http://www.baidu.com/link?url=9ZDtkj0bapitMUUhEZ5I12lGyhFcCuEg-FuFUFh02kmmqaYm0pe8LdH3O8_4rP240Z_Md1MNgqWS5ZFgSPTOAFDKsbwjzgWzVwTU8vwHdFdPwTfHpntiuILFav_R-lVJ) **根据输入的逻辑地址和指令码，进行对内存的读写**

#define MEMORY\_COM\_READ\_WORK\_PARM 340 读取内存工作参数

#define MEMORY \_ COM\_READ\_DEBUG\_PARM 341 读取内存配置参数

#define MEMORY \_ COM\_READ\_ALL\_PARM 342 读取内存所有参数

#define MEMORY \_ COM\_CHECKSUM \_ALL\_PARM 343 读取校验内存参数

#define MEMORY \_ COM\_WRITE\_WORK\_PARM 350 下载内存工作参数

#define MEMORY \_ COM\_WRITE \_DEBUG\_PARM 351 下载内存配置参数

#define MEMORY \_ COM\_PROGRAM \_ALL\_PARM 352 下载 读取 校验

输入：

Unsigned int memory\_logic\_address

Unsigned int memory\_pram\_address

Unsigned int memory\_parm\_value

Void memory \_command\_read\_work\_parm();

Void memory \_command\_read\_debug\_parm();

Void memory \_command\_read\_all\_parm();

Void memory \_command\_checksum\_all\_parm();

Void memory \_command\_write\_work\_parm();

Void memory \_command\_write\_degug\_parm();

Void memory \_command\_program\_all\_parm();//将操作结果反馈给上位机

Unsigned int memory\_read\_data(memory\_logic\_address, memory\_ parm\_address);

voidmemory\_write\_data(memory\_logic\_address,

memory\_ parm\_address, memory\_parm\_value);

void memory \_command\_check();

**6.8 Control\_command.c 根据指令码，读写内存数据,并进行指令码转换操作**

输入：

上位机指令码 界面指令码 舵机指令码 内存指令码

输出：

舵机指令或者反馈指令

Unsigned int servo\_command\_value

Unsigned int host\_command\_value

Unsigned int memory\_command\_value

Unsigned int control\_command\_return

函数

void menu\_command\_check();

void host\_command\_check();

void servo\_command\_check();

void memory\_command\_check();

**6.9 Uart\_com.c 舵机串口底层驱动**

Void servo\_com\_ rev\_data();

Void servo\_com\_write\_data()

**6.10 Host\_com.c 上位机串口底层驱动**

Void host\_com\_ rev\_data();

Void host\_com\_write\_data()

**6.11 I2c\_com.c 外部内存底层驱动**

Void memory\_com\_ rev\_data();

Void memory\_com\_write\_data()

**6.12 define.c 全局变量声明**

**7 数据结构**

**7.1 通讯数据结构体**

**struct\_com\_data**

**{**

Unsigned char com\_device\_id

Unsigned char com \_device\_cmd

Unsigned char com \_device\_state

Unsigned char com \_parm\_id\_1

Unsigned char com \_parm\_valueH\_1

Unsigned char com \_parm\_valueL\_1

Unsigned char com \_parm\_id\_2

Unsigned char com \_parm\_valueH\_2

Unsigned char com \_parm\_valueL\_2

Unsigned char com \_parm\_check\_sum

}communication\_data

**7.2 WORK\_PARM数据结构体**

struct\_work\_parm

{

Unsigned int servo\_epprom\_parm\_flag

Unsigned int control\_pulse\_upper

Unsigned int control\_pulse\_mid

Unsigned int control\_pulse\_lower

Unsigned int control\_set\_upper

Unsigned int control\_set\_mid

Unsigned int control\_set\_lower

Unsigned int servo\_zero\_zone\_set

Unsigned int servo\_zero\_zone\_time

Unsigned int servo\_free\_lock\_flag

Unsigned int servo\_dir\_inverse\_flag

Unsigned int servo\_pwm\_inverse\_flag

Unsigned int insig\_value\_diff\_set

Unsigned int servo\_program\_version

Unsigned int servo\_epprom\_parm\_reset\_flag

Unsigned int servo\_unique\_address\_id

Unsigned int servo\_common\_command\_address

Unsigned int servo\_reset\_flag\_set

Unsigned int servo\_reset\_position

Unsigned int servo\_reset\_time\_set

Unsigned int servo\_reset\_step\_set

Unsigned int control\_interval\_time\_set

Unsigned int control\_interval\_step\_set

Unsigned int servo\_halt\_zone\_set

Unsigned int servo\_halt\_time\_set

Unsigned int servo\_protect\_time\_set

Unsigned int servo\_protect\_pwm\_cmp

Unsigned int servo\_protect\_pwm\_set

Unsigned int servo\_max\_pwm\_set

Unsigned int servo\_motor\_work\_pwm\_cycle

Unsigned int motor\_pwm\_cycle

Unsigned int servo\_init\_flag\_set

Unsigned int servo\_work\_base\_voltage

}work\_parm\_data

**7.3 DEBUG\_PARM数据结构体**

struct\_debug\_parm

{

Unsigned int servo\_work\_mode\_lock\_flag\_set

Unsigned int servo\_work\_mode\_now

Unsigned int servo\_pram\_config\_set

Unsigned int servo\_command\_fb\_flag

Unsigned int servo\_position\_pid\_parm\_p\_set

Unsigned int position\_pid\_speed\_parm\_radio

Unsigned int servo\_speed\_run\_sample\_k\_set

Unsigned int servo\_position\_sample\_ov\_time\_set

Unsigned int servo\_speed\_sample\_ov\_time\_set

Unsigned int servo\_speed\_pid\_parm\_p\_set

Unsigned int servo\_speed\_pid\_parm\_p\_radio

}debug\_parm\_data

8 电路板机械尺寸



以上图片和接口 ，仅为参考，可根据具体定

电路板 长8CM 宽4CM

**9 项目验收及付款**

项目开发费用 4500

**9.1硬件开发 时间1周**

开发硬件测试板 5块，

USB通讯

UART通讯

LCD显示

KEY操作

电源测试

仿真器连接编译

完成后付款1500

**9.2 软件开发 时间1周**

根据需求说明书中，软件架构及其文件结构，接口等编写和编译

完成后付款1000

**9.3 功能测试 时间1周**

根据需求说明书中，功能进行测试和调试，实现所有功能

完成后付款1500

**9.4 可靠性测试 时间1周**

交付内测和客户进行可靠性测试

完成后交付500