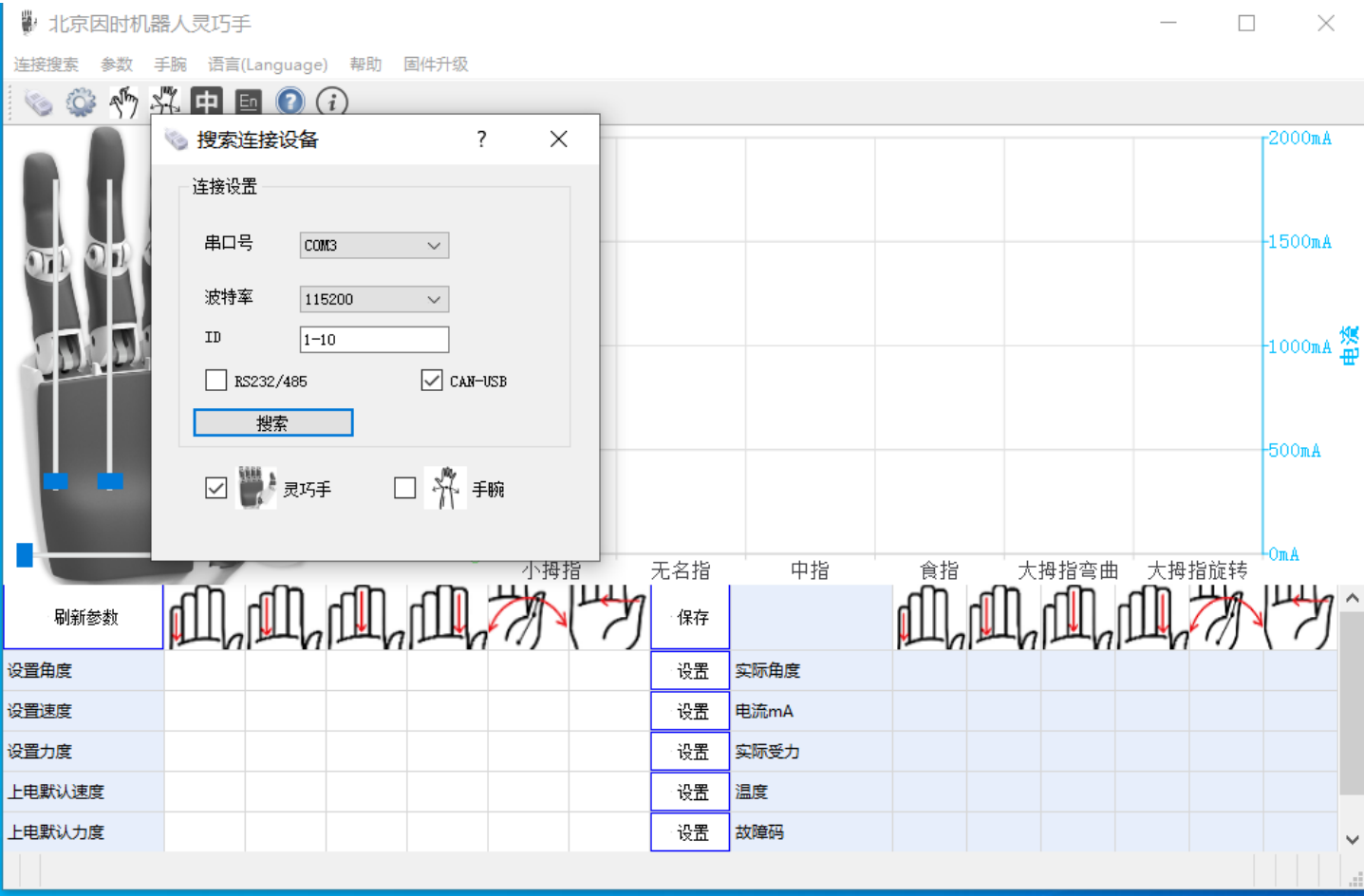


# 【孙锦程 李俊龙】 因时手的控制

因时手的控制及程序架构

类型1：CAN->USB（本质通过串口发送can帧）

Windows 示教页面





协议帧

```
1 AA AA 01 00 08 18 08 00 00 00 00 00 00 00 01 00 01 00 2B 55 55
```

AA AA是帧头；01 00 08 18是帧ID；08 00 00 00 00 00 00 00是数据内容；01是数据长度；00 01 00为固定帧格式；2B是第三位到第十八位相加的校验和；55 55是帧尾。

举例：

```
1 AA AA 01 00 75 05 58 02 00 00 00 00 00 00 02 00 01 00 D8 55 55
```

是发送05750001的帧（帧格式解释在下图），数据内容是600（0x0258），数据长度为2，校验和为D8.

功能：将ID为1的灵巧手的食指的角度设置值设置为600。

## 2. CAN 读写寄存器操作

波特率为 1000K，采用扩展标示符、数据帧格式。不使用标准标示符及远程帧。  
其中扩展标示符共 29 位，从低位至高位定义如下：

bit0~13, Hand\_ID 最大支持 16383 个设备。

bit14~25: 要操作寄存器的起始地址。

bit26~28: 读写标志位，0 表示为读灵巧手寄存器操作，1 表示为写灵巧手寄存器操作，4 表示为读灵巧手手腕寄存器操作，5 表示为写灵巧手手腕寄存器操作。

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	W/R 0:R 为读灵巧手寄存器操作 1:W 为灵巧手写寄存器操作 4:R 为读灵巧手手腕寄存器操作 5:W 为写灵巧手手腕寄存器操作	寄存器地址 Address	Hand_ID

### 2.1 读寄存器操作

读寄存器的标识符设置如下：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	0	Address	Hand_ID

数据长度为 1 Byte。

数据内容为：预读取寄存器数据的长度。

灵巧手收到以上指令并正确解析后会回复以下帧：

标识符：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	0	Address	Hand_ID

数据长度：返回寄存器数据的长度。

数据内容：寄存器数据。

例如欲读 ID 为 1 的灵巧手食指的当前实际角度，应向相 CAN 总线发送一下帧：

标识符：二进制为 0000 0001 1000 0100 0000 0000 0000 0001

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	0	ANGLE_ACT(3)的地址为 1552; 二进制为 11000010000	1

数据长度:1

数据内容：2,食指当前实际角度是一个整型数据，数据长度为 2byte。

灵巧手返回帧如下：

标识符：二进制为 0000 0001 1000 0100 0000 0000 0000 0001

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	0	ANGLE_ACT(3)的地址为 1552; 二进制为 11000010000	1

数据长度:2

数据内容如下，食指的当前实际角度 POS\_ACT(3)是整型，需要将下面数据转化为整型（低字节在前，高字节在后），高低字节交换后的 16 进制为 0x01F4，转化为

10 进制为 500，即食指的当前实际角度为 500。

byte0	byte1
0xF4	0x01

2.2 写寄存器操作

写寄存器的标识符设置标识符：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	1	Address	Hand_ID

数据长度：欲写入寄存器内数据的长度。

数据内容：欲写入寄存器内的数据。

灵巧手收到以上指令并正确解析后会回复以下帧：

标识符	bit 31-29	bit 26-28	bit14-25	bit 13-0
含义	预留	1	Address	Hand_ID

数据长度：0

例如欲将 ID 为 1 的灵巧手的食指的角度设置值设置为 600，应向相 CAN 总线发送一下帧：

标识符：二进制为 0000 0101 0111 0101 0000 0000 0000 0001

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	1	ANGLE_SET(3)的地址为 1492; 二进制为 10111010100	1

数据长度:2

数据内容如下，食指的当前实际角度 ANGLE\_SET(3)是整型，需要整型数据进行高低字节拆分，然后交换高低字节，即 600（0x0258）转为以下数据内容。

byte0	byte1
0x58	0x02

灵巧手返回帧如下：

标识符：二进制为 0000 0001 0111 0101 0000 0000 0000 0001

bit 31-29	bit 26-28	bit14-25	bit 13-0
0	1	ANGLE_SET (3)的地址为 1540; 二进制为 1001 0110	1

数据长度:0

手的位置，速度，力的限制范围，安全保护

2.4.12. FORCE\_SET (m) 各自由度的力控阈值设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度力控阈值设置值，具体说明见下表。这些参数可断电保存。用户可设置这组寄存器来实现对手指握力的控制，例如将 FORCE\_SET(3)设置为 300，当前食指实际角度 ANGLE\_ACT(3)为 1000（即完全张开状态），当用户将食指设置角度 ANGLE\_SET(3)设置为 0 后，食指将在向手掌弯曲运动，在弯曲过程中如果检测到食指实际受力（即 FORCE\_ACT(3)，具体说明参见 2.4.17）到达 300 后，食指停止运动，如果食指实际受力没有到达 300，那么食指将运动到设置角度，单位：g。

地址	名称	说明	数据类型	范围
1498-1499	FORCE_SET(0)	小拇指力控设置值	short	0-1000
1500-1501	FORCE_SET(1)	无名指力控设置值	short	0-1000
1502-1503	FORCE_SET(2)	中指力控设置值	short	0-1000
1504-1505	FORCE_SET(3)	食指力控设置值	short	0-1000
1506-1507	FORCE_SET(4)	大拇指弯曲力控设置值	short	0-1000
1508-1509	FORCE_SET(5)	大拇指旋转力控设置值	short	0-1000

寄存器数值含义是指相应手指指尖提供的握力，例如 DEFAULT\_FORCE\_SET(1)设置为 800 时，表示允许无名指手指指尖提供 800g 的握力。（如果手指接触物体的部分不是指尖，那么可提供的握力则有所不同，具体大小与力臂长度有关，力臂越短，提供的握力越大）。

2.4.13. SPEED\_SET(m) 各自由度的速度设置值

此寄存器组由六个寄存器组成，分别对应灵巧手六个自由度的速度设置值，具体说明见下表。这些参数可断电保存，无单位量纲。

地址	名称	说明	数据类型	范围
1522-1523	SPEED_SET(0)	小拇指速度设置值	short	0-1000
1524-1525	SPEED_SET(1)	无名指速度设置值	short	0-1000
1526-1527	SPEED_SET(2)	中指速度设置值	short	0-1000
1528-1529	SPEED_SET(3)	食指速度设置值	short	0-1000
1530-1531	SPEED_SET(4)	大拇指弯曲速度设置值	short	0-1000

1532-1533	SPEED_SET(5)	大拇指旋转速度设置值	short	0-1000
-----------	--------------	------------	-------	--------

速度为 1000：表示指手指在空载时从大角度运动到最小角度用时 800ms，如果负载较大时，这个实际速度会有一定程度的降低。

#### 2.4.15. ANGLE\_ACT(m) 各自由度的角度实际值

地址	名称	说明	数据类型	范围
1546-1547	ANGLE_ACT(0)	小拇指角度实际值	short	0-1000
1548-1549	ANGLE_ACT(1)	无名指角度实际值	short	0-1000
1550-1551	ANGLE_ACT(2)	中指角度实际值	short	0-1000
1552-1553	ANGLE_ACT(3)	食指角度实际值	short	0-1000
1554-1555	ANGLE_ACT(4)	大拇指弯曲角度实际值	short	0-1000
1556-1557	ANGLE_ACT(5)	大拇指旋转角度实际值	short	0-1000

#### 2.4.16. FORCE\_ACT(m) 各手指的实际受力

此寄存器组由六个寄存器组成，分别对应灵巧手六个手指的实际受力值，具体说明见下表。这些参数为只读参数，单位：g。

地址	名称	说明	数据类型	范围
1582-1583	FORCE_ACT(0)	小拇指实际受力值	short	0-1000
1584-1585	FORCE_ACT(1)	无名指实际受力值	short	0-1000
1586-1587	FORCE_ACT(2)	中指实际受力值	short	0-1000
1588-1589	FORCE_ACT(3)	食指实际受力值	short	0-1000
1590-1591	FORCE_ACT(4)	大拇指弯实际受力值	short	0-1000
1592-1593	FORCE_ACT(5)	大拇旋转实际受力值	short	0-1000

## 寄存器地址

## 2.4 寄存器说明

灵巧手对用户开放的寄存器参数如下：

地址	含义	缩写	长度	读写权限
1000 (0x03E8)	ID	HAND_ID	1byte	W/R
1002 (0x03EA)	波特率设置	REDU_RATIO	1byte	W/R
1004 (0x03EC)	清除错误	CLEAR_ERROR	1byte	W/R
1005 (0x03ED)	保存数据至 Flash	SAVE	1byte	W/R
1006 (0x03EE)	恢复出厂设置	RESET_PARA	1byte	W/R
1008 (0x03F0)	保留	reserve	1byte	W/R
1009 (0x03F1)	受力传感器校准	GESTURE_FORCE_CLB	1byte	W/R
1020 (0x03FC)	各自由度的电缸电流保护值	CURRENT_LIMIT(m)	6short(12byte)	W/R
1032 (0x0408)	各自由度的上电速度设置值	DEFAULT_SPEED_SET(m)	6short(12byte)	W/R
1044 (0x0414)	各自由度的上电力控阈值设置值	DEFAULT_FORCE_SET(m)	6short(12byte)	W/R
1472 (0x05C0)	系统电压	VLTAGE	1short	R
1474 (0x05C2)	各自由度的电缸位置设置值	POS_SET(m)	6short(12byte)	W/R
1486 (0x05CE)	各自由度的角度设置值	ANGLE_SET(m)	6short(12byte)	W/R
1498 (0x05DA)	各自由度的力控阈值设置值	FORCE_SET(m)	6short(12byte)	W/R
1522 (0x05F2)	各自由度的速度设置值	SPEED_SET(m)	6short(12byte)	W/R
1534 (0x05FE)	各自由度的电缸位置实际值	POS_ACT(m)	6short(12byte)	R
1546 (0x060A)	各自由度的角度实际值	ANGLE_ACT(m)	6short(12byte)	R



1582 (0x062E)	各手指的实际受力	FORCE_ACT(m)	6short(12byte)	R
1594 (0x063A)	各自由度的电缸的电流值	CURRENT(m)	6short(12byte)	R
1606 (0x0646)	各自由度的电缸的故障信息	ERROR(m)	6byte	R
1612 (0x064C)	各自由度的状态信息	STATUS(m)	6byte	R
1618 (0x0652)	各自由度的电缸的温度	TEMP(m)	6byte	R
2000 (0x07D0)	当前动作序列检验码 1	ACTION_SEQ_CHECKDATA1	1byte	W/R
2001 (0x07D1)	当前动作序列检验码 2	ACTION_SEQ_CHECKDATA2	1byte	W/R
2002 (0x07D2)	当前动作序列总步骤数	ACTION_SEQ_STEPNUM	1byte	W/R
2016 (0x07E0)	当前动作序列第 0 步设置	ACTION_SEQ_STEP(0)	19short(38byte)	W/R
2054 (0x0806)	当前动作序列第 1 步设置	ACTION_SEQ_STEP(1)	19short(38byte)	W/R
2092 (0x082C)	当前动作序列第 2 步设置	ACTION_SEQ_STEP(2)	19short(38byte)	W/R
2130 (0x0852)	当前动作序列第 3 步设置	ACTION_SEQ_STEP(3)	19short(38byte)	W/R
2168 (0x0878)	当前动作序列第 4 步设置	ACTION_SEQ_STEP(4)	19short(38byte)	W/R
2206 (0x089E)	当前动作序列第 5 步设置	ACTION_SEQ_STEP(5)	19short(38byte)	W/R
2244 (0x08C4)	当前动作序列第 6 步设置	ACTION_SEQ_STEP(6)	19short(38byte)	W/R
2282 (0x08EA)	当前动作序列第 7 步设置	ACTION_SEQ_STEP(7)	19short(38byte)	W/R
2320 (0x0910)	当前动作序列索引号	ACTION_SEQ_INDEX	1byte	W/R
2321 (0x0911)	保存当前动作序列	SAVE_ACTION_SEQ	1byte	W/R
2322 (0x0912)	运行当前动作序列	ACTION_SEQ_RUN	1byte	W/R
2324 (0x0914)	动作序列力控调节值	ACTION_ADJUST_FORCE_SET	2byte	W/R

## CAN

- 1 su#拷贝Linux系统类型对应的USBCAN驱动文件后进行
- 2 cp libusb.so libusb-1.0.so libECanVci.so.1 /usr/lib#进入USBCAN驱动文件夹，拷贝到gcc编译库目录。默认路径为/usr/lib
- 3 ln-sv libECanVci.so.1 libECanVci.so
- 4 make

```

5 ./test
6 3 0 1 0x1c00 0 1 1000#需要输入7位内容，具体内容解释在代码下方
7 ##如果缺少libusb-0.1-4
8 sudo apt-get update
9 sudo apt-get install libusb-0.1-4

```

输入./test 后会出现系统提示及例子，其中：第一位（16）：设备类型，单通道设备输入3，双通道输入4；

第二位（0）：设备索引号，当只接入一台USBCAN时为0；


第三位（3）：打开第几路CAN，打开CAN1输入1，打开CAN2输入2同时打开CAN1和CAN2输入3；

第四位（0x1400）：设置CAN总线波特率，0x1400表示波特率为1000K，此设备波特率为500k（0x1c00）其他波特率的值详见“EcanVCI动态库使用手册”；

第五位（0）：工作模式，0为正常模式，其他工作模式详见“EcanVCI动态库使用手册”；

第六位（1）：发送时间间隔，单位ms；

第七位（1000）：发送次数。



```

root@gckj-ThinkPad-X201:/home/gckj/Desktop/linux64# ./test
test [DevType] [DevIdx] [ChMask] [Baud] [TxType] [TxSleep] [TxFrames]
example: test 16 0 3 0x1400 0 1 1000

```

16	0	3	0x1400	0	1	1000
DevType	DevIdx	ChMask	Baud	TxType	TxSleep	TxFrames

1000 frames / channel  
tx > sleep(3ms) > tx > sleep(3ms) ....  
0-normal, 1-single, 2-self\_test, 3-single\_self\_test, 4-single\_no\_wait....  
0x1400-1M, 0x1c03-125K, ....  
bit0-CAN1, bit1-CAN2, bit2-CAN3, bit3-CAN4, 3=CAN1+CAN2, 7=CAN1+CAN2+CAN3  
Card0  
1-usbcan, ....

## 上位机操作

添加设备

ECanToolsV5. 9. 24English

选择设备类型:USBCAN-V5打开设备

name	Hardware	ID
USBCAN-II-V5	UGP. 31. 52	GC221030433

CAN1CAN2

通道: CAN1

名称: USBCAN-II-V5

协议: CAN = 1

工作模式: 正常模式

波特率: 500 K

自动识别波特率

www.gcgd.net

确定取消

ECanTools

文件 操作 视图 窗口 帮助

保存数据 实时保存 暂停显示 显示模式 清除 滤波设置 高级屏蔽 显示错误帧 错误帧率:0.0% 0 P/S 接收帧数:1

序号	帧间隔时间us	名称	帧ID	帧类型	帧格式	DLC	数据	帧数量
00000001	134. 180. 888	发送成功	05750001	DATA	EXTENDED	2	58 02	1
00000002	134. 184. 280	接收	05750001	DATA	EXTENDED	1	02	1

普通发送 发送文件 0 P/S 发送帧数:1

发送方式: 正常发送

多次发送时: ☐ 帧ID每发送一帧递增 ☐ 发送数据每发送一帧递增

帧类型: 数据帧

帧格式: 扩展帧

帧ID (HEX): 05750001 长度: 2 数据 (HEX): 58 02

发送次数: 1 每次发送间隔: (ms) 10

发送 停止

(发送间隔最小0. 1ms, 实际发送速度受波特率影响)

添加/设置 复位 删除

设备	通道数	波特率 (0/1)	总线负载 (0/1)	总线流量 (0/1)
<input checked="" type="checkbox"/> USBCAN-II-V5	2	500/250	0%/0%	0/0

Can1 状态

Can2 状态

Can1 控制状态

- 接收寄存器满
- 接收寄存器溢出
- 发送寄存器空
- 发送结束
- 正在接收

Can1 总线状态

- 正在发送
- 错误报警
- 缓存区溢出
- 总线数据错误
- 总线仲裁错误

总线错误计数

接收	发送
0	0

Connected



注:波特率为500k; 帧格式为扩展帧; 发送数据时, 数据与数据之间需要加空格。

# 参考资料

 [因时机器人仿人五指灵巧手--CAN增补协议V0.0.1cn.pdf](#)

 [因时机器人仿人五指灵巧手--RH56用户手册V1.09cn.pdf](#)

 [EcanVCI动态库使用手册5.31.pdf](#)



①GCAN Tools调试分析软件.exe

3.57MB

