终端控制板 V1 系列使用手册

1. 功能简介

1.1 产品系列概述

终端控制板 V1.XX 系列专为机器人爱好者，DIY 设计开发的产品，其核心控制使用 FPGA，并行处理，响应速度快。尺寸小巧 130×50×20mm，可以直接安装在狭小的空间内比如机器人手臂。USB 或 Uart 串行接口方便使用和系统的搭建。

1.2 产品概述

终端控制板 V1.10 核心采用 FPGA，响应速度快（受限于通信的波特率），使用方便，USB 接口可以直接连接到工控板，利用串口调试助手就可以进行调试，也可以利用 Uart 接口和 ARM 等嵌入式处理器一起使用或使用无线模块远程遥控。协议简单仅 3 个字节（0xaa＋地址＋数据）。

1.3 产品特性

1.3.1 硬件特性

通信：

- ◆1 路 USB 转 232 串口
- ◆1 路 232 串口（LVTTL 电平，不可和 USB 同时使用）
- ◆波特率可调（9600，19200，115200，256000）

数字量：

- ◆22 路数字量输入
- ◆16 路数字量输出

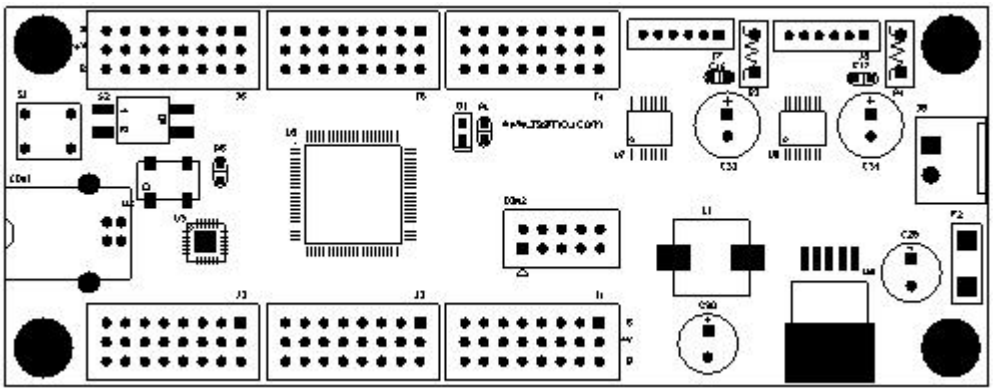
伺服电机：

- ◆8 路伺服电机
- ◆死区可调（0～840uS）
- ◆最小可调角度 1 度（0～180 度对应于 0～2mS）

直流电机：

- ◆2 路直流电机
- ◆2 路 24 位编码器
- ◆支持直流电机最大电压 20V，最大 2A 电流
- ◆过流，过热保护并报错（保护时间周期为 1.2mS）
- ◆休眠功能
- ◆2 种制动模式

1.3.2 硬件结构布局



1.3.3 接口标号及应用

标号	功能	备注
J1	数字量输入	2.54mm 间距
J2	数字量输入	2.54mm 间距
J3	数字量输入和 RXD，TXD	2.54mm 间距
J4	数字量输出	
J5	数字量输出	
J6	伺服电机输出	
J7	1＃直流电机&码盘	2.0mm 间距
J8	2＃直流电机&码盘	2.0mm 间距
J9	电源输入	+5V～+20V
CON1	USB 接口	

1.3.4 数字量输入

数字量输入电平标准为 LVTTL，兼容 TTL 电平，主要用于 IO 型的传感器采集比如红外传感器，触摸传感器，碰撞及热释电等。

标号	引脚	IO 号
J1	1→8	I-1→I-8
J2	1→8	I-9→I-16
J3	1→6	I-17→I-22
J3	7，8	TXD，RXD

1.3.5 数字量输出

数字量输出为 LVTTL 电平，主要用于开关量比如表情等等的 LED 的控制，一些传感器的使能等。

标号	引脚	IO 号	地址
J4	1→8	O-1→O-8	0x0b
J5	1→8	O-9→O-16	0x0c

以 J4 为例，J5 类似

J4-8	J4-7	J4-6	J4-5	J4-4	J4-3	J4-2	J4-1
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
数字输出 8	数字输出 7	数字输出 6	数字输出 5	数字输出 4	数字输出 3	数字输出 2	数字输出 1

数字量输出的使用很简单，直接向相应的地址发控制字就可以了，IO 号的高低和字节的高低顺序是一样的。

例如：要 O-0 为低电平则通过串口调试助手发送：

0xaa 0x0b 0xfe

这时除了 O-0 是”0”外 O-1～O-7 都是”1”。

同理，如果要 O-15～O-8 的状态是”11110000”则发送

0xaa 0x0c 0xf0

1.3.6 伺服电机输出

伺服电机输出的功能自然不必说了，只是市场上电机的品牌型号众多，它们的死区和对脉宽的要求也不尽相同，这里只支持周期是 20mS 的伺服电机。对于一般的伺服电机而言控制信号脉冲为 2ms，可以转动的角度为 180 度，因此每个角度大约对应 11.2uS，又因为每个字节的范围是 0～255，所以输出脉冲宽度为：0～2.8mS。以最常见的伺服电机为例，死区是 0.5mS，所以 0.5mS～2.5mS 对应 0～180 度，

0.5mS/11.2uS≈45＝0x2d

当输入控制字为 0x2d 时，输出为 0.5mS 宽度周期 20mS 的脉冲，这时电机在 0 度位置，想让电机转到 1 度位置则控制字为 0x2e(45+1),90 度则为 0x87(45+90),以此类推。

标号	引脚	伺服电机号	地址
J6	1	1＃	0x01
J6	2	2＃	0x02
J6	3	3＃	0x03
J6	4	4＃	0x04
J6	5	5＃	0x05
J6	6	6＃	0x06
J6	7	7＃	0x07
J6	8	8＃	0x08
伺服电机使能			0x0d

伺服电机使能主要是对于一些可以自锁的电机用的，在电机到位后可以使控制信号输出为”0”这样就不容易抖动，这个功能同样也可以通过相应地址发送 0 来实现，不过使能还可以实现多路电机的同步转动。伺服电机使能寄存器 Bit7 对应 8＃电机，Bit0 对应 1＃电机。”1”为使能，”0”为禁止。

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8# 伺服电机使能	7# 伺服电机使能	6# 伺服电机使能	5# 伺服电机使能	4# 伺服电机使能	3# 伺服电机使能	2# 伺服电机使能	1# 伺服电机使能

1.3.7 直流电机

直流电机包括 2 个部分：直流电机 PWM 调速和编码器计数。由于受到资源限制以及本控制板的设计目标，这里没有做速度或位置的闭环控制算法，相关的算法需要在上位机或其他的控制器来实现。终端控制板 V2 系列配有 ARM7 控制器，可以实现速度和位置的闭环控制。

标号	引脚	备注
J7	1	1＃直流电机+
	2	1＃直流电机-
	3	1＃编码器 A
	4	1＃编码器 B
	5	1＃编码器+5V
	6	1＃编码器 GND

J8	1	2＃直流电机+
	2	2＃直流电机-
	3	2＃编码器 A
	4	2＃编码器 B
	5	2＃编码器+5V
	6	2＃编码器 GND

直流电机控制相关的寄存器

名称	地址
1＃直流电机速度寄存器	0x09
2＃直流电机速度寄存器	0x0a
1＃&2＃直流电机控制寄存器	0x0e

直流电机控制寄存器

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
2# 编码器清零	2# 电机模式	2# 电机方向	2# 电机休眠	1# 编码器清零	1# 电机模式	1# 电机方向	1# 电机休眠

说明：编码器清零 “1”为清零

“0”为正常计数

电机模式 “1”为慢制动

“0”为快速制动

电机方向 “1”为正向

“0”为反向

电机休眠 “1”为正常

“0”为休眠

不管写哪个寄存器，都会返回一个信息帧，其中电机状态字的含义如下

返回电机状态

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留	保留	2＃电机错误	1＃电机错误	保留	保留	2＃电机转向	1＃电机转向

直流电机速度计算方法是这样的：对于直流电机而言不同的电压对应不同的电机转速，电压越高转速越快。

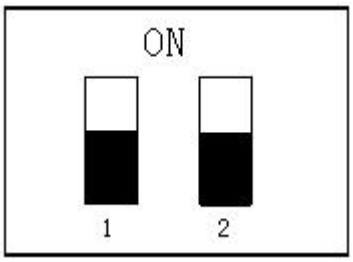
$$P = I^2R = U^2 / R = F * V$$

因此在扭矩不变的情况下（一般也就是底盘匀速）转速和电压平方成正比。在这里直流电机的电压是靠 PWM 来控制的，直流电机速度寄存器有效范围是 0～200，这意味着当为 0 是有效电压为 0，当为 100 时有效电压为电源输入电压的 1/2，当为 200 时有效电压为电源输入，这时电机全速运转。

配合编码器信息可以很容易实现位置的闭环控制，最简单的办法就是分段控制，在启动和接近位置期望的时候分段加速和减速，以笔者以前的经验只要分段合理效果非常平滑。

1.3.8 通信设置

终端控制板集成了 USB 转 Urat 芯片，方便系统构建，在 PC 端需要安装通用的驱动程序（CP2102 USB），终端控制板的拨码开关可以选择通信的波特率以适应不同系统的应用需要。



如上图所示，ON 位置为”1”,波特率选择如下表所示

拨码 1	拨码 2	波特率
OFF	OFF	9600
OFF	ON	19200
ON	OFF	115200
ON	ON	256000

为了方便不同的应用，构建不同的系统，终端控制板预留了 Uart 接口，J3 的 7 和 8 引脚分别是 TXD，RXD（注意是 LVTTL 电平，兼容 3.3V CMOS）。同时终端板返回的信息帧格式也是固定的，它们的格式是：

帧头	1＃编码器	2＃编码器	错误&方向	I-0~I-7	I-8~I-15	I-16~I-21
0xaa	3Byte	3Byte	1Byte	1Byte	1Byte	1Byte

编码器计数器为 3 个字节，高字节在前低字节在后，数字量输入字节位的高低顺序和字节位高低一致。

串口通信设置：8 个数据位，1 个停止位，无校验。



1.3.9 电源

电源输入范围＋5V～＋20V，电源输入会直接影响到直流电机的速度，输入就是电机全速转动的电压，因此，电源输入最大值不要超过电机的额定值。终端控制板会产生＋5V 供传感器等使用，输出电流最大 3A，一般红外传感器电流大约 30mA，所以可以满足一般的应用。伺服电机也可由＋5V 供电，但是会产生较大的纹波，因此建议如果伺服电机数量比较多或者负载过重的话单独外部供电（单独供电需要把 R8 电阻去除）。

2.2 软件

2.2.1 寄存器表

名称	地址	备注
伺服电机 1＃位置	0x01	0～255
伺服电机 2＃位置	0x02	
伺服电机 3＃位置	0x03	
伺服电机 4＃位置	0x04	
伺服电机 5＃位置	0x05	
伺服电机 6＃位置	0x06	
伺服电机 7＃位置	0x07	
伺服电机 8＃位置	0x08	
1#直流电机速度	0x09	0～200 有效
2#直流电机速度	0x0a	0～200 有效
数字量输出 1-8	0x0b	
数字量输出 9-16	0x0c	
伺服电机使能	0x0d	‘1’使能，’0’禁止
直流电机控制	0x0e	参见控制寄存器

2.2.2 软件设计参考

终端控制板的设计思路是快速构建机器人系统，减少系统层次，从而提高系统可靠性和降低开发难度，做为一个机器人爱好者也许不熟悉 C 语言，对单片机，ARM 这类嵌入式处理器的开发环境也知之甚少，但是只要能操作 PC 的串口就可以利用我的终端控制板进行开发，笔者就是用 VB 做的上位机程序开发,利用串口控件加上一些逻辑判断,很容易就让您的机器人动起来。

终端控制板每接收到一个正确的指令帧就会返回一帧数据，（一共 11 个字节，前面已经详细说明了内容）。我们用串口调试助手最小发送间隔是 1mS，但终端板处理速度远大于它，你可以连续不断的发送指令，指令也会立即被执行，但是返回信息帧受到波特率限制，因此我是这样来处理的：如果指令过快，在返回信息帧的时间里多次就收指令帧，那么它只返回一个信息帧。举例说明这个问题

假设波特率是 9600，那么一个返回信息帧 11 字节

$$9600/(11*10)\approx 87$$

这说明 1S 可以返回 87 帧，也就是每帧 11.5mS，如果 2 个指令帧间隔大于 11.5mS，那么你可以得到 2 帧返回数据，如果小于 11.5mS，那么只能有 1 帧返回数据。你可能会说实时性不够，那么很简单，提高波特率。假如 256000bps 的话，每个返回帧时间为不到 0.5mS，这对于大多数的应用已经足够了。



以上都是理论分析，读者可以自行调试确定合适的波特率等时间参数，在上图可以看到，如果指令帧间隔大于返回帧时间的话，串口调试助手（发送：接收＝3：11）

你也可以连续发送指令，指令执行的顺序就是你发送的先后顺序，不过它们执行的很快，等于 3 个字节的发送时间，但是每个指令帧是原子的，不允许插入其他信息。这种情况很容易出现在 USB 和 Uart 接口同时使用的情况下，由于没有同步或者互斥机制，会导致指令出错，因此不建议同时使用 USB 和 Uart 接口。

