UIROBOT®
United Intelligence Robot Technology

使用手册

UID828

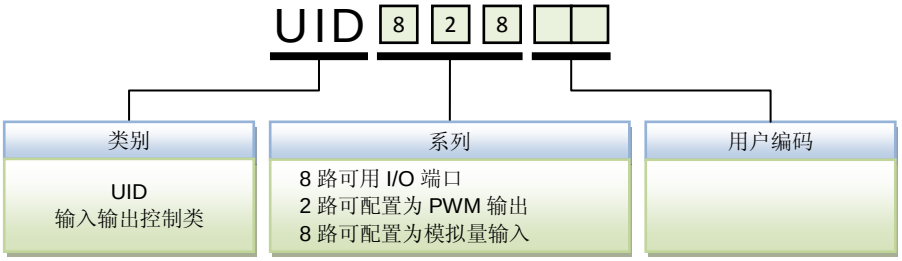
CAN2.0B 总线通讯

多功能模拟数字 I/O 及 PWM 控制器

[UID828 产品订购说明]

在订购 UID828 产品时请按以名下格式提供产品号，以便我们准确及时地为您提供产品：

UID828 产品牌号



UID828
多功能模拟数字 I/O 及 PWM 控制器

多功能 IO 端口特性

- 单机 8 路多功能 TTL 数字量输入 / 输出，可指令配置，实时调整
- 单机 8 路模拟量（12 位）输入，可指令配置，实时调整
- 单机 2 路独立 PWM 输出，可指令配置，实时调整，基频 0.01 ~ 5000Hz，占空比分辨率 0.5%
- 最大 800 路 TTL 数字量输入/输出（100 个 UID828 组网）
- 最大 200 路独立 PWM 输出（100 个 UID828 组网）
- 最大 800 路模拟量（12 位）输入（100 个 UID828 组网）
- 可与 UIM242 步进电机控制器混合组网
- 配合相应的继电器模块，可对开关阀，比例阀和直流电机的执行器件进行控制
- 配合相应的电平转换模块，可接受多种行程开关，限位开关和传感器输入

嵌入式微处理机

- 内置高性能嵌入式微处理器系统
- 指令结构简单直观
- 免费提供基于 MS Windows 的 VC 源代码和指令封装动态链接库

CAN2.0B 通讯特性

- 主动 CAN 2.0, 全网络仅用一对双绞线（两根导线）
- 1 百万通讯比特率，10 公里通讯距离
- 可连接节点高达 100 个
- 采用差分总线，具有很强的抗噪特性

电气特性

- 宽电压输入 6～40VDC

简介

UID828 是 CAN 总线系列数字端口输入输出控制器, 采用优爱宝的 SimpleCAN 通讯协议。它具备 8 个方向和功能可指令配置的输入输出端口。这 8 个端口可配置为数字输入输出或者模拟量输入。其中两个数字 I/O 端口可配置为 PWM（脉宽调制）输出。PWM 波形的基频和占空比均可经由指令实时调整。UID828 可与 UIM242 系列步进电机控制驱动器混合组成控制局域网络，亦可单独组成数据采集和输入输出控制网络。

对于熟悉 CAN 通讯协议并具备 CAN 主控机的用户，可直接采用优爱宝提供的 SimpleCAN 协议实现对整个电机-传感器-电磁阀网络的控制。

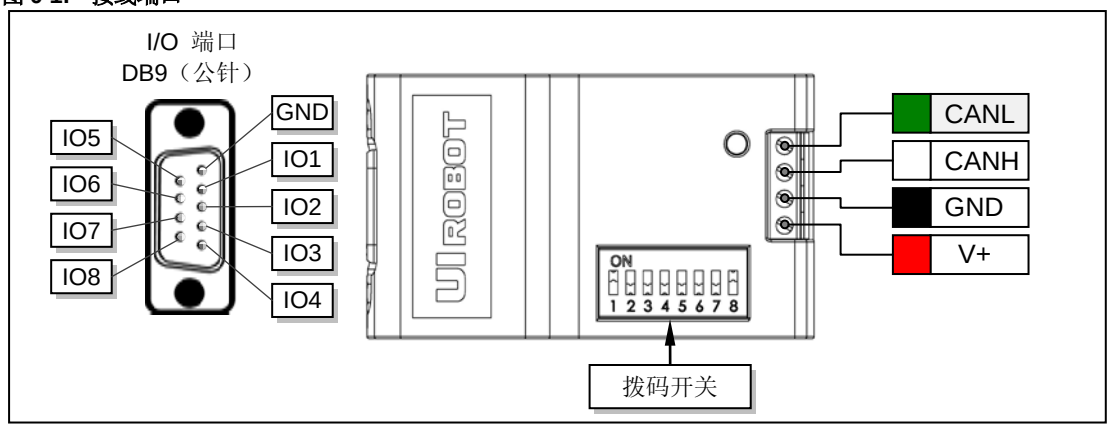
对于不想了解 CAN 协议或不具备 CAN 主控机的用户，可采用优爱宝的 UIM2501 CAN/RS232 转换器，通过基于 RS232 简单直观的指令来控制整个电机-传感器-电磁阀网络，

免去了用户直接使用相对较复杂的 CAN 协议。一台用户上位机只需一个 UIM2501，就可同时控制最多 100 台 UID828 数据采集器与 UIM242 驱动器。

UID828 的控制指令结构简单，高容错。数据采集器外壳为全铝合金铸件，坚固耐用，散热性能好。

接线端口

图 0-1: 接线端口



控制端口

端口	符号	说明
1	V+	工作电压正极。电压：6 - 40V 直流
2	GND	工作电压地线，即 0V（工作电压正负极不可接错）
3	CANH	CAN 总线的高位线
4	CANL	CAN 总线的低位线

DB9 I/O 端口

标准 DB9 接插口（公针），其中 DB9 的第 1 针为地，第 2 针到第 9 针分别对应 8 路输入/输出的第 1 路到第 8 路，即 DB9 的第 2 针为第 1 路输入/输出，第 3 针为第 2 路输入/输出，依次类推。

拨码开关

UID828 带有一个 8 位的多功能拨码开关。拨码 1 到拨码 7 在上电时，提供了 UID828 的地址标识码。拨码 8 置于 ON 时（图示位置），UID828 内置的 CAN 总线终端电阻（Terminating Resistor）被使能。出厂时，UID828 地址标识码设置为 7，同时**终端电阻未使能**。

警告：除非 UID828 位于总线最末端，否则切勿使能终端电阻！

用户通过调整拨码开关定义 UID828 定义地址标识码。地址设定采用二进制：

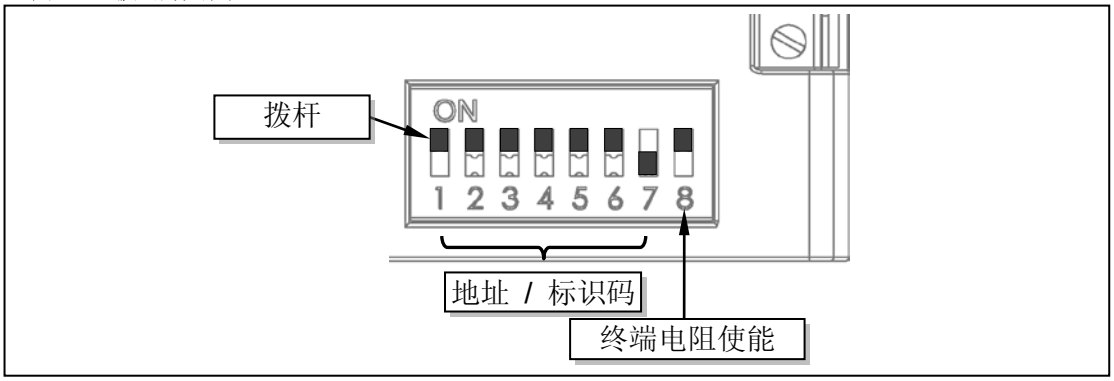
$$\text{地址/标识码} = D1*2^0 + D2*2^1 + D3*2^2 + D4*2^3 + D5*2^4 + D6*2^5 + D7*2^6$$

当拨码开关 Dx (x = 1, 2, …7) 拨到“ON”端时，Dx = 1；反之，Dx = 0。例如当需要设定 地址 = 63 时：

$$63 = 1*2^0 + 1*2^1 + 1*2^2 + 1*2^3 + 1*2^4 + 1*2^5 + 0*2^6$$

即 D1, D2, D3, D4, D5, D6 拨到字母“ON”端，D7 拨到数字端，如图 0-2 所示。

图 0-2：拨码开关设定



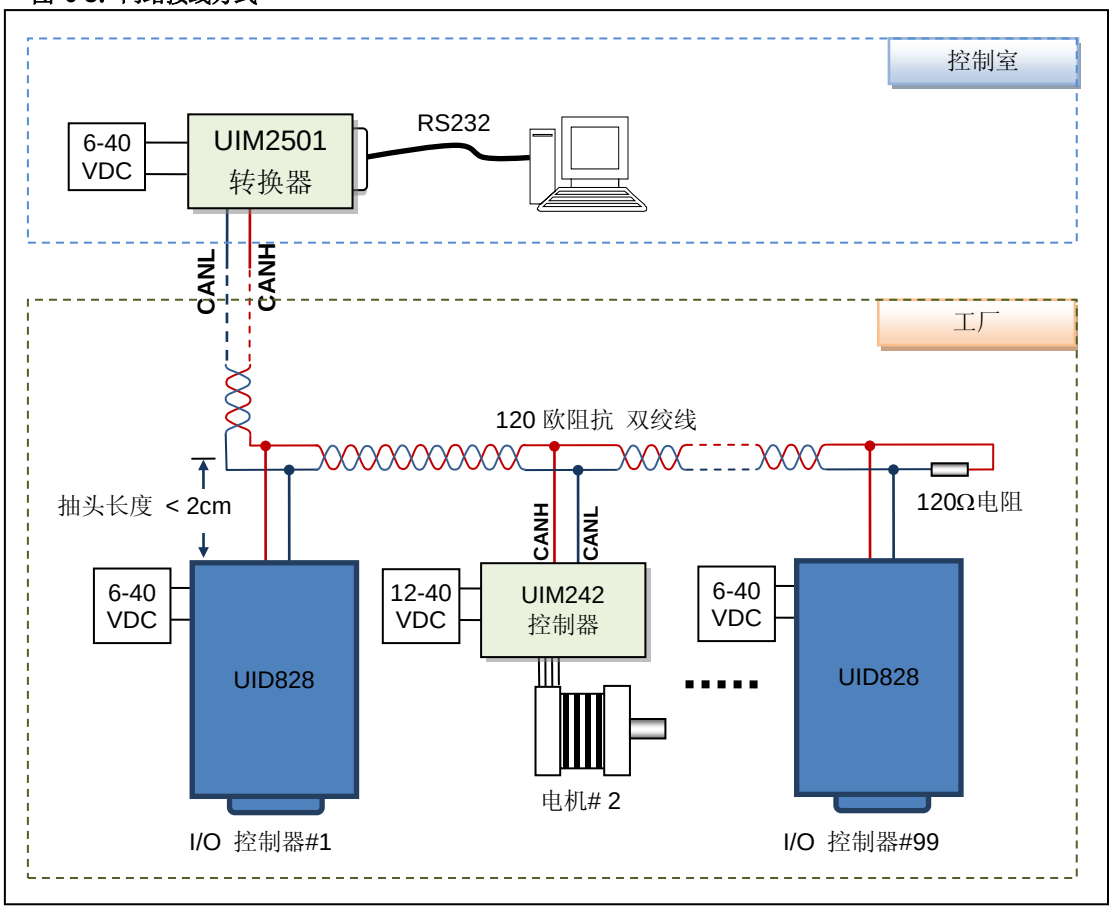
注意：UID828 站点地址必须在 5 – 125 之间（包括 5 和 125）。

典型接线

CAN 总线提供了一个非常可靠和简洁的网络组建方案。如果采用一个 RS232/CAN 驱动转换器挂带多台数据采集器与电机驱动器的控制方式，可采用如下接线方式（图 0-3）。

注意: 在多节点（即多数据采集器/驱动器）的情况下，应该用一根双绞线将所有节点连接起来，如图 0-3 所示。一定要避免使用星形连接方式。同时，每个节点抽头线的长度不应超过 2 cm，且越短越好。在总线距离较长时，双绞线的两端应各连上一个 120 欧姆的终端电阻(Terminating Resistor)。特别是在总线距离超过 100 米时应考虑采用 CAN 总线专用的 120 欧姆阻抗屏蔽双绞线。

图 0-3：网络接线方式



指令总表

指令	说明	消息头	标识码	页码
AVAη;	查询端口模拟量输入值	CC	C9	28
BTR;	查询当前 CAN 通讯比特率	AA	BC	37
BTRη;	设置 CAN 通讯比特率	AA	BC	36
DVA;	查询端口输入电平逻辑值	AA	C5	39
DVAη;	设置端口输出电平逻辑值	AA	C4	38
IOC η;	设置端口功能以及输入/输出方	AA	C3	40
IOC;	查询端口功能以及输入/输出方	AA	C3	41
MCF;	查询主配置寄存器	AA	B0	43
MCFη;	设置主配置寄存器	AA	B0	42
PBR;	查询 PWM 波形输出时基倍率	AA	C8	45
PBRη;	设置 PWM 波形输出时基倍率	AA	C8	44
PMB;	查询 PWM 波形输出时基	AA	C7	47
PMBη;	设置 PWM 波形输出时基	AA	C7	46
PMD;	查询 PWM 输出占空比	AA	C6	49
PMDη;	设置 PWM 输出占空比	AA	C6	48

性能指标

绝对最大值（注 1）

供电电压..... 6V 至 40V
输入输出端口相对于 GND 的电压.....-0.3V 至+5.3V
偏置电压下的环境温度..... -40°C 至+85°C
储存温度..... -50°C 至+150°C
注 **1:** 如器件工作条件超过上述"绝对最大值"，可能会对器件造成永久性损坏。上述值仅为运行条件极大值，建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性会受到影响。

工作电气性能（环境温度 25℃时）

供电电压范围	6V - 40VDC
供电输入电流	100mA 最大
I/O 口输入低电平	最大 1.5V（1.5V 以下即认为低电平）
I/O 口输入高电平	最少 3.5V（3.5V 以上即认为高电平）
I/O 口输入电流	10uA 最大（即端口消耗电流；I/O 端口配置为输入时，内阻为 100K
I/O 口输出低电平	小于 0.6V
I/O 口输出高电平	大于 4.5V
I/O 口输出电流	5mA 最大（即端口驱动能力）
I/O 口短路电流	10mA
PWM 基频可调范	0.01 Hz – 5000 Hz（即 PWM 波形周期 200 微秒 – 100 秒）
PWM 占空比分辨	0.5%

通讯方式（环境温度 25℃时）

通讯协议	主动 CAN 2.0
物理连接	二线制，CANH、CANL，双绞线
CAN 总线驱动	• 支持 1 百万比特率的运行速率，满足 ISO-11898 标准物理层要求 • 短路保护，高压瞬态保护，自动热关断保护 • 可连接节点 100 个 • 采用差分总线，具有很强的抗噪特性

使用环境及参数

冷却方式	自然冷却
使用场合	避免粉尘、油雾及腐蚀性气体
使用温度	-40 ℃ ～ 85 ℃
使用湿度	<80%RH，无凝露，无结霜
使用震动	3G Max
保存温度	-50 ℃ ～ 150 ℃

尺寸及重量

外形尺寸	66.4mm x 38mm x 18mm
重　　量	0.2kg

目　录

1.0产品介绍

UID828 模拟/数字 I/O 及 PWM 输入输出控制器主要功能如下：

- 可与 UIM242XX 控制驱动器、UIM2501 转换器组成 CAN 网络，完成数字量输入 / 输出，模拟量输入 以及 PWM 波形输出。
- 2 路可指令配置 PWM 波形的输出，基频 0.01 ~ 5000 Hz 指令可调，占空比分辨率 0.5%。
- 8 路可指令配置数字输入 / 输出。
- 8 路可指令配置模拟量输入。

配合 UIM2501 的使用，能让不了解 CAN 总线协议的用户在享受 CAN 总线协议带来的高速（1Mbps）、长距离（10 公里）和高抗干扰等一系列优点的同时，还可以充分利用简单易用的通讯界面，进而能够专注于应用，提高设计效率、缩短开发时间。

UID828 支持 1Mbps 的 CAN 通讯比特率。精简后的指令在 CAN 总线上的传输时间小于 0.1 毫秒，一般为 0.05 毫秒。通讯比特率可由用户指令修改，以适应不同距离的传输需要。

UID828 数据采集器内置嵌入式微处理系统。上位机（PC 机或控制设备）通过串行口连接到转换器与数据采集器后，向其送 ASCII 指令即可控制数据采集器的输入/输出。

指令结构简单，高容错。如果输入了错误指令，数据采集器将返回错误信息给上位机。错误指令不会被执行，避免发生事故。

该转换控制器可以使用 6V～40V 宽电压范围直流供电。

优爱宝公司同时免费提供运行在 Microsoft Windows 上的控制步进电机的演示软件及例程（VB/VC）

注：优爱宝所提供的演示软件和例程仅作为用户学习和参照使用。因为环境和使用环境的不同，该例程可能需要调整，请用户谨慎。

2.0 CAN2.0 通讯设置

UID828 能够实现以下 CAN 通讯比特率（表 2-1），同时能够通过指令在这些 CAN 通讯比特率之间动态调整。调整波特率的目的主要是为了取得较长的通讯距离和稳定的工作状态。用户可以用 BTR 编号来通知 UID828 切换通讯比特率。指令为 BTRη，其中 η 就是上述通讯比特率编号。每个 UID828 出厂时 BTR 都设置为 1，即 800Kbps/50 米通讯距离。一般说来，每一条指令或者消息需要占用 64 – 128 个比特。

请注意，本文中所提到的反馈信息都是指 UID828 通过 UIM2501 发给上位机的反馈信息。

表 2-1 CAN 总线通讯比特率

BTR 编号	比特率 (bps)	总线长度 (米)
0	1000K	25
1	800K	50
2	500K	100
3	250K	250
4	125K	500
5	50K	1000
6*	20K	2500
7*	10K	5000

*注：该通讯速率为定制，用户在购买时需提前通知，否则无法设置。

2.1 指令列表

本章所涉及指令列表如下，各指令详细解释位于本文档末尾，具体页码请参见表格：

指令	说明	详解页码
BTRη;	设置全局网络的 CAN 通讯比特率η	36
BTR;	查询当前网络的 CAN 通讯比特率	37

3.0 指令和反馈信息结构

UID828 输入输出控制器接收上位机通过转换器 UIM2501 发来的操作信息（指令）并执行该信息所要求的操作；同时回复 ACK 信息（收到指令复述确认）；并且按客户要求返回目前各项操作参数。在没有收到客户机新的指令前，控制器完成指令后将保持现有工作状态。

对于使用自备 CAN 主机，而不用 RS232-CAN 协议转换器 UIM2501 的用户，请参阅优爱宝 SimpleCAN 协议编程手册来学习如何与 UID828 通讯。以下的内容都是针对使用 UIM2501 的用户而言的。

3.1 指令结构

指令是上位机向 UID828 发送的，指示其完成一定功能的信息。UID828 所接受的指令都遵循以下规则：

INS η ； 或者 **INSx η ；** 或者 **INS ;**

指令符 **INS** 由三个不间断的字母组成，不分大小写。若带有 x（如 **INS**x），则表示该指令附带的数据输入为十六进制形式。数据 **η** 由一串数字组成。有些指令没有数值，例如查询指令等。每句指令必须以分号，即 **";**"结尾。没有分号结尾的指令，将导致不可预期的后果。

反馈报文是运动控制器向上位机发送的信息。UIM 运动控制器产生的信息长度不固定，最大 13 字节。

UID828 通过 UIM2501 转换器发出的反馈报文使用如下结构：

[报文头] [控制器标识码] [报文标识码] [报文数据] [结束符]

报文头有三种 AA、CC 和 EE。

控制器站点 表示当前控制器在一个网络中的识别标号(又称站点)。 标识范围：5 – 125 。

报文标识码 标明了该条信息的属性。

报文数据 采用 7 位数据结构排列，高位在先，低位在后。反馈报文中的 7 位数据字节通过移位操作转化为 16 位数据。16 位数据占用 3 个反馈数据字。

结束符 标明一条信息的结束。UID828 采用 FF 或 FE 作为结束符。若结束符为 FF 表示本条报文没有后续报文，若结束符为 FE 则表示本条报文还有后续报文。

注意，有两类反馈报文是没有信息标识码的：基本 ACK 和电机状态反馈（针对 FBK 指令的反馈）。另外有些反馈报文是没有报文数据的，比如一些实时状态变化通知。

4.0 实时状态变化通知RTCN

UID828 支持实时状态变化通知。与微处理器的中断相仿，实时状态变化通知是指在发生某个事件时，UID828 主动向上位机发送一个简短的信息。信息长度一般在 4 个字节左右。从事件发生到发送通知，时间小于 1 毫秒。传送的时间取决于通讯波特率。当 UIM2501 转换器以 57600 波特率的 RS232 传输时，传输时间小于 1 毫秒。所以从事件发生到上位机收到通知，时间大约 2 毫秒（CAN 总线传输时间小于 0.05 毫秒可以忽略）。

对于 UID828，事件主要指当某端口配置为数字输入端口后，在端口上侦测到电平出现上升边沿或下降边沿。请注意，不是电平变化，而是边沿变化。

4.1 状态变化通知消息格式

当侦测到输入电平发生时，UID828 通过 UIM2501 自动给用户发送以下格式的实时反馈报文：

CC [控制器标识码] [反馈标识码] FF

控制器标识码为控制器的站点（由用户通过拨码设置）。

如果用户直接采用 SimpleCAN 与 UID828 进行通讯，则需要参阅 UIRobot 的 SimpleCAN2.0A 或者 SimpleCAN2.0B 协议编程手册。

表 4-1: 实时状态变化通知事件

编号	事件名称	信息标识码	相关备注
1	IO 端口 n 下降沿	0xE0 + n*2	端口 n 侦测到电平发生：高 >>>低 变化时
2	IO 端口 n 上升沿	0xE1 + n*2	端口 n 侦测到电平发生：低 >>>高 变化时

注：上表中，n=1, 2,…8。

例如，当用户收到反馈信息 **CC 07 E8 FF** 时，则表示站点为 7 的 UID828 侦测到输入到端口 4 的电平发生了由高到低的变化（因为 0xE8 = 0xE0 + 4*2）。

4.2 使能 / 禁止状态变化通知

所有实时状态变化通知可被指令使能或者禁止。

使能和禁止是通过写主配置寄存器（MCFG 指令）的**P1IE**（MCFG<0>）到**P8IE**（MCFG<7>）位来实现的。注意，只有将端口配置为数字输入后，才能最终使用 RTCN 变化通知。 端口配置为输出后不支持 RTCN。

5.0 主配置寄存器

5.1 主配置寄存器

UID828 设有一个主配置寄存器，以实现以下功能：1）使能或禁止系统的 RTCN 变化通知；2）设置上电时，输出端口（配置为数字）的电平。一旦配置成功，主配置寄存器的参数会立刻生效。出于安全性考量，MCFG 低 8 位（即 RTCN 变化通知配置）不会保存在 EEPROM 中，每次开机必需重新设置。与之相反，MCFG 高 8 位（上电时输出端口的电平）会保存在 EEPROM 中。

主配置寄存器由 16 位比特组成，结构如下：

MCFG 寄存器位定义

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
定义	IL8	IL7	IL6	IL5	IL4	IL3	IL2	IL1	P8IE	P7IE	P6IE	P5IE	P4IE	P3IE	P2IE	P1IE

位 15- 8 上电时，端口 8 至端口 1 的电平输出
0 = 相应端口（配置为数字输出）上电时输出低电平 （即 0V）
1 = 相应端口（配置为数字输出）上电时输出高电平 （即 5V）

位 7 - 0 端口 8 至端口 1 变化通知使能开关
0 = 禁止相应端口（配置为数字输入）的 RTCN 变化通知
1 = 使能相应端口（配置为数字输入）的 RTCN 变化通知

5.2 指令列表

本章所涉及指令列表如下，各指令详细解释位于本文档末尾，具体页码请参见表格：

指令	说明	详解页码
MCFη;	设定主配置寄存器数值	42
MCF;	查询当前主配置寄存器数值	43

6.0 操作指令

用户可以将 UID828 作为单独器件进行操作，也可将数个 UID828 接入 CAN 网络进行数字输入/输出控制。在上位机（主机）一侧，用户可以经由 RS232/CAN 转换器 UIM2501，通过基于 RS232 的指令来控制整个 CAN 网络中的每个 UID828 以及 UIM242 步进电机控制驱动器（如果有 UIM242 的话）。

依照 CAN 网络协议的规定，每个 UID828 必须有唯一的地址/标识码。标识码是上位机用来识别用户指令发送对象的依据，同时也是用户判断 UID828 数据采集器反馈来自何方的依据。如果一个步进电机控制网络下有两个或多个相同的标识码将导致错误。指定该唯一标识码的方法是通过拨码设置。详情见图 0-2。

这一章主要描述数字量输入/输出、模拟量输入、PWM 输出设置方面的操作。

6.1 UID828 端口功能

UID828 的多功能端口可由用户通过指令配置。各端口可用配置如下：

端口	*TTL 输入*	*TTL 输入 内部上拉*	*TTL 输出*	*PWM 输出*	*模拟量输入*
IO1	✓	✓	✓	-	✓
IO2	✓	✓	✓	-	✓
IO3	✓	✓	✓	-	✓
IO4	✓	✓	✓	-	✓
IO5	✓	✓	✓	-	✓
IO6	✓	✓	✓	-	✓
IO7	✓	✓	✓	✓	✓
IO8	✓	✓	✓	✓	✓

6.2 端口功能配置寄存器（IOC寄存器）

UID828 数据采集器设有一个端口功能配置寄存器，用以设置端口的输入/输出功能。

一旦配置完成，配置寄存器的参数会立刻生效并且自动存入 UID828 数据采集器上 EEPROM，掉电不会丢失。该存入过程不影响控制的实时性。

端口功能配置寄存器由 16 位组成，结构如下：

IOC 寄存器位定义

位	*15*	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
定义	IOC8		IOC7		IOC6		IOC5		IOC4		IOC3		IOC2		IOC1	

上表中 IOCn （n=1，2…8）按以下方式决定端口的功能（即数字输入，数字输出，模拟量输入或者 PWM 输出）。

IOCn = 00 （二进制）时，端口 n 为 PWM 输出；

IOCn = 01 （二进制）时，端口 n 为 数字 TTL 电平输出；

IOCn = 10 （二进制）时，端口 n 为 模拟量输入；

IOCn = 11 （二进制）时，端口 n 为 数字 TTL 电平输入；

6.3 端口数字量输出控制寄存器（DVA寄存器，Digital Value）

UID828 的所有 I/O 端口，配置为数字输出时，可以输出高/低电平。高电平为+5V，低电平为 0V。

DVA 寄存器位定义

位	7	6	5	4	3	2	1	0
定义	D8	D7	D6	D5	D4	D3	D2	D1

位 7 – 0 端口 8 至端口 1 的电平输出
0 = 相应端口输出低电平 （即 0V）
1 = 相应端口输出高电平 （即 5V）

需要特别说明的是，IO8、IO7、IO6 三个端口在没有外部输入信号时（端口悬空）其状态是随机的。如果此时将该三端口配置为数字输入时，将会得到不可预料的结果。

警告，当开启了变化通知 RTCN 反馈后，不停跳动的端口电平会导致过量的变化通知出现在 CAN 总线上从而导致总线阻塞。

IO1、IO2、IO3、IO4、IO5 五个端口内置有上拉电阻，端口悬空时读为 1，对地短接时读为 0。

如果某端口的 IOC 配置不是数字输入功能的话，则对应于该端口的 DVA 值永远为 0。

6.4 模拟量输入控制寄存器（AVA寄存器，Analog Value）

UID828 的 8 个端口可全部或者部分配置为接受模拟量信号输入（0 – 5 V）并且将端口的模拟量转化为 12 位分辨率的数字量供用户查询。

端口的模拟量输入功能由 IOC 寄存器控制。用户可以使用 AVA 指令查询端口模拟量信号的 AD 转换值。 AVA 寄存器的为定义如下。当某一位设置为 1 时，该位对应端口的模拟量输入将会被 AD 转换并发送到上位机。

AVA 寄存器位定义

位	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
定义	A8	A7	A6	A5	A4	A3	A2	A1

位 7 – 0 表示是否查询端口 8 至 端口 1 的模拟量输入
0 = 不发送相应端口 AD 转换值
1 = 发送相应端口 AD 转换值

6.5 PWM输出时基设置指令

脉冲宽度调制（Pulse Width Modulation，缩写为 PWM），简称脉宽调制。它是利用微处理器的数字输出来对模拟电路进行控制的一种非常有效的技术，广泛应用于测量、通信、功率控制与变换等许多领域。其基本实现方式是输出一列固定周期但是高低电平比例可调的方波。该列方波的周期称为时基，该列方波中高电平持续时间和时基的比例称为占空比。UID828 能提供两路 PWM 波形输出。其 IO7、IO8 两端口可配置为 PWM 输出。PWM 输出的时基用 PMB 指令设置。实际的输出时基还同 PWM 时基倍率相关。时基倍率由 PBR 指令设置，将在下一节介绍。

PMB 寄存器位定义

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
定义	PWMBase2								PWMBase1							

上表中 PWMBase1 和 PWMBase2 决定端口 7 和端口 8 的 PWM 波形的时基。

端口 7 的 时基 = PWMBase1 * 200 微秒 * PBR （其中 1 ≤ PWMBase1 ≤ 200）

端口 8 的 时基 = PWMBase2 * 200 微秒 * PBR （其中 1 ≤ PWMBase2 ≤ 200）

用户在设置 PMB 之前，应确保已经使用 ”IOCη;” 指令将端口 7 或/和端口 8 设置为 PWM 输出。

PMB 指令执行后，PMB 寄存器的数值将会保存入片上 EEPROM，断电不会丢失。PMB 指令执行后，PWM 波形输出会终止。用户需再次设置 PWM 占空比以启动 PWM 波形输出。

应用案例 6-4

需要 IO8 的时基为 3000 微秒，IO7 时基为 5000 微秒，

可选取 PBR = 1，则有：

PWMBase2 = 3000 微秒 / （200 微秒 * 1） = 15 (十进制) = 0x0F（十六进制）
PWMBase1 = 5000 微秒 / （200 微秒 *1） = 25 (十进制) = 0x19（十六进制）

所以 PMBx 0F19;

应用案例 6-5

需要 IO8 的时基为 3 秒（3000000 微秒），IO7 的时基为 4 秒（4000000 微秒），

可选取 PBR = 100，则有：

PWMBase2 = 3000000 微秒 / （200 微秒 * 100） = 150 (十进制) = 0x96（十六进制）
PWMBase1 = 4000000 微秒 / （200 微秒 * 100） = 200 (十进制) = 0xC8（十六进制）

所以 PMBx 96C8;

6.6 PWM输出时基倍率设置（PBR）

为了能够实现PWM输出时基200微秒到100秒的宽幅调整，UID828引入了时基倍率（PWM Base Rate） 的概念。

PWM 时基 = PWM Base * 200 微秒 * PBR

其中, 1 ≤ PWMBase ≤ 200, 1 ≤ PBR ≤ 2500

6.7 PWM输出占空比设置指令（PMD）

UID828 的 PWM 输出的占空比用 PMD 指令设置。

PMD 寄存器位定义

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
定义	PWMDuty2								PWMDuty1							

上表中 PWMDuty1 和 PWMDuty2 决定端口 7 和端口 8 的 PWM 波形的占空比。分辨率是 0.5%。

端口 7 的 输出占空比 = PWMDuty1 * 0.5% （0 ≤ PWMDuty1 ≤ 200）

端口 8 的 输出占空比 = PWMDuty2 * 0.5% （0 ≤ PWMDuty2 ≤ 200）

出于安全考量，PMD 寄存器的数值**不会保存**入片上 EEPROM。

PMD 指令执行后，系统启动 PWM 波形输出。对于一个已知系统，PMB 只需设置一次，PMD 可以不断更新。

应用案例 6-6

需要 IO8 占空比为 30%，IO7 占空比为 50%，

PWMDuty2 = 30% / 0.5% = 60(十进制) = 0x3C (十六进制)
PWMDuty1 = 50% / 0.5% = 100（十进制）= 0x64（十六进制）

因此，PMDx 3C64;

6.8 指令列表

本章所涉及指令列表如下，各指令详细解释位于本文档末尾，具体页码请参见表格：

指令	说明	详解页码
AVAη;	查询端口模拟量输入值	28
DVAη;	设定端口输出电平控制寄存器 DVA 的数值	38
DVA;	查询当前端口输入的数字电平	39
IOCη;	设定端口功能配置寄存器 IOC 的数值	40
IOC;	查询当前端口功能配置寄存器数值	41
PBRη;	设定 PWM 波形的时基倍率	44
PBR;	查询 PWM 波形的时基倍率	45
PMBη;	设定 PWM 波形的时基	46
PMB;	查询 PWM 波形的基频周期	47
PMDη;	设定 PWM 波形的占空比	48
PMD;	查询 PWM 波形的占空比	49

6.9 数字量 及 PWM 配置示例

描述：

配置 UID828 输出五路高电平，一路输入，同时输出两路周期为 5ms, 占空比为 50% 的 PWM 波形，同时当输入状态变化时自动反馈。

要求：

IO1、IO2、IO3、IO4、IO5 输出高电平，IO6 输入，IO7、IO8 输出 PWM 波形。

实现：

1. 将 IO6 接地或接 5V。不可悬空!（注，用户可以在完成本示例后，尝试让 IO6 端口悬空，以观察不可预知电平对 UID828 造成的影响）
2. **发送指令：** **MCFx 0020;** （配置 IO6 的变化通知。只启动 P6IE，其他所有端口的变化通知都禁止。用户可以根据实际情况，自行调整)。
3. 发送指令：**IOCx 0D55;** （配置 IOC1、IOC2、IOC3、IOC4、IOC5 为 01（二进制）数字输出，IOC6 为 11（二进制）数字输入，IOC7、IOC8 为 00（二进制）PWM 输出)。
4. 发送指令：**DVAx 001F;** （D1 – D5 为 1（高电平））。
5. 查询 IO6 当前输入状态，**发送指令：** **DVA;**
6. 设置 PWM 时基倍率为 1，**发送指令：** **PBR 1;**
7. 配置 CH7、CH8 周期为 5ms;

   PWMBase1 = 5ms / (0.2ms*PBR) = 25(十进制) = 0x16（十六进制）

   PWMBase2 = 5ms / (0.2ms*PBR) = 25(十进制) = 0x16（十六进制）

   发送指令： **PMBx 1616;**
8. 配置 CH7、CH8 占空比为 50%；

   PWMDuty1 = 50% / 0.5% =100(十进制) = 0x64（十六进制）

   PWMDuty1 = 50% / 0.5% =100(十进制) = 0x64（十六进制）

   发送指令： **PMDX 6464;**
9. 观察：

   用万用表测量 IO1、IO2、IO3、IO4、IO5 相对于 GND 的电压，可得到：输出+5V 高电平；

   变换加载于 IO6 上的电平，可以看到有实时变化通知发送到上位机；

   指令 DVA; 查询可知输入 IO6 端口上的输入电平；

   通过示波器，应该能观测到 IO7、IO8 端口上的输出的周期为 5ms，占空比为 50%的 PWM 波形。

7.0 指令说明

本章将详细介绍之前各章所涉及的指令。

请注意，在本使用手册中，如果没有特别说明，所有报文都是基于 RS232 字符串报文的结构，形式和解析方法。对于基于 UI simpleCAN 的 CAN 报文的结构，形式和解析方法请参阅 UI simpleCAN 编程手册。

7.1 指令报文结构

指令是上位机向运动控制器发送的，指示完成一定功能的信息。UID820 接受的指令都遵循以下规则：

1. 单条指令总长度（包括结尾分号）不能超过 20 个字符。
2. 所有指令字符均以 7 位的标准 ASCII 码（1 – 127）表示，不可以加长 ASCII 码表示。
3. 指令结构如下：

INS η ;

或者 **INSx η ;**

或者 **INS ;**

其中，

INS **指令符** 由三个不间断的字母组成， 不分大小写。

若带有 x（INSx），则表示该指令附带的数据为 16 进制形式。

请注意，使用 16 进制数据时，必须确保数据为偶数位。如 00，01，0A 等。奇数位数据将导致错误，例如 001，10A 等为非法。

η **数值** 由一串不间断的数字组成。有些指令没有数值，例如查询指令 SPD; STP; 等。

; **结束符** 每句指令必须以分号，即 ";"结尾。

注意：没有分号结尾的指令将导致不可预期的后果。

7.2 反馈报文结构

反馈报文是运动控制器向上位机发送的信息。UID820 产生的信息长度不固定，最大 13 字节。

UID820 通过 UIM2501 转换器发出的反馈报文使用如下结构：

[报文头] [控制器站点] [报文标识码] [报文数据] [结束符]

报文头

表示一条反馈报文的开始。有如下三种：

– AA 表示指令确认反馈（ACK），是对收到的指令的一种重复。
– CC 表示状态反馈，是对现状的描述。
– EE 表示收到的信息有错误，不能被执行。

控制器站点

表示当前控制器在一个网络中的识别标号(又称站点)。对于 UID820，该标识范围：5 – 125 。

报文标识码

标明了该条信息的属性。例如：CC 05 E0 FF, 中的 E0 表示该信息是端口 1 出现下降沿。详细的内容在后面章节针对具体指令展开。

报文数据

采用 7 位数据结构排列，高位在先，低位在后。

图 7-1 演示了反馈报文中的 7 位数据字节通过移位操作转化为 16 位数据。

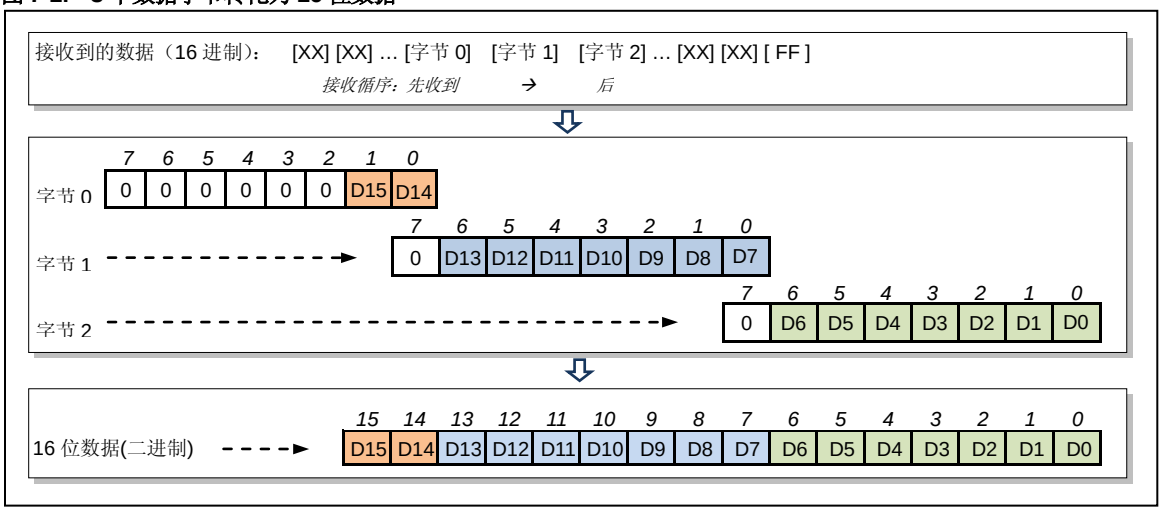
16 位数据占用 3 个反馈数据字节。

结束符

标明一条信息的结束。UIM 运动控制器采用 FF 或 FE 作为结束符。若结束符为 FF 表示本条报文没有后续报文，若结束符为 FE 则表示本条报文还有后续报文。

注意，有两类反馈报文是没有信息分类码的：基本 ACK 和电机状态反馈（针对 FBK 指令的反馈）。另外有些反馈报文是没有报文数据的，比如一些实时状态变化通知。

图 7-1: 3 个数据字节转化为 16 位数据



编程示例 7-1

3 个数据字节转化为 16 位数据的移位操作（C 编程）

```
unsigned int    data;                   //data     16 位数据
data = Byte[1]    & 0x03;               //Byte[1]  第 1 个收到的数据字节
data = ( data << 7 ) + Byte[2];         //Byte[2]  第 2 个收到的数据字节
data = ( data << 7 ) + Byte[3];         //Byte[3]  第 3 个收到的数据字节
```

1. AVAη / AVAxη　查询模拟量输入

语　　法：　AVAη; 或者 AVAxη;

指令描绘：　当有外部信号接入 UID828，

需要知道出现在端口上的模拟量测量值时可以使用该指令查询。

η = 0、1…255 （整数） 或者 η = 0x0000…0x00FF （十六进制）

ACK 信息：　CC　[UID828 标识码]　C9　[AVA1]　[AVA2] … [AVA8]　[ACB]　FF

ACK 解析：　C9　　>>　为 AVAη;指令的信息标识码；

[AVA1] ~ [ACB]　>>　返回数据 1 ~ 9。

[AVA1] ~ [AVA8] 转换成 12 位数据后表示端口模拟输入量（见图 7-1）。

[ACB]是一个组合字节，表示本条消息中包含的数据字节数量以及所涉及的端口号的最高位（MSB）。其数据结构定义在下文详述。

ACB 数据结构

ACB 表示本条消息中包含的数据字节数量以及所涉及的端口号的最高位（MSB）。其数据结构定义如下：

位	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
定义	0	Size			a3	a2	a1	a0

其中第 6，5，4 位组成的 3 位二进制数表示本条消息包含的数据长度（Size）

Size = 001，表示有 2 个数据字节（即 AVA1, AVA2）

Size = 011，表示有 4 个数据字节（即 AVA1, AVA2, AVA3, AVA4）

Size = 101，表示有 6 个数据字节（即 AVA1, AVA2, AVA3, AVA4, AVA5, AVA6）

Size = 111，表示有 8 个数据字节（即 AVA1, AVA2, AVA3, AVA4, AVA5, AVA6, AVA7, AVA8）

第 0 位 到 第 3 位 分别表示了所涉及端口的端口号的最高位。

AVA 数据解析

由于每两个数据字节表达了一个端口的 AD 转换值，所以，一条 AVA 反馈消息最多可表示 4 个端口的 AD 转换值。

因此，当用户使用 AVA 指令查询 4 个以上端口的 AD 转换值时，UID828 会发回 2 条 AVA 反馈消息。

在解读 AVA 反馈消息时，用户需根据 ACB 的第 0-3 位来确定某个端口的端口号和相应 AD 转换值。具体解析方法见下文应用案例。

应用案例 7-3

要求： 取得 IO1 至 IO8 端口模拟量输入的 AD 转换值。

首先将所有端口配置为模拟量输入，可如下配置 IOC 寄存器：

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
填写	10		10		10		10		10		10		10		10	
注解	IOC8 = 10		IOC7 = 10		IOC6 = 10		IOC5 = 10		IOC4 = 10		IOC3 = 10		IOC2 = 10		IOC1 = 10	

因此： IOC = 0xAAAA（十六进制）

由此，可使用指令： **IOCx AAAA;** 来实现要求的配置。

其次如下设置 DVA 寄存器：

位	7	6	5	4	3	2	1	0
定义	1	1	1	1	1	1	1	1

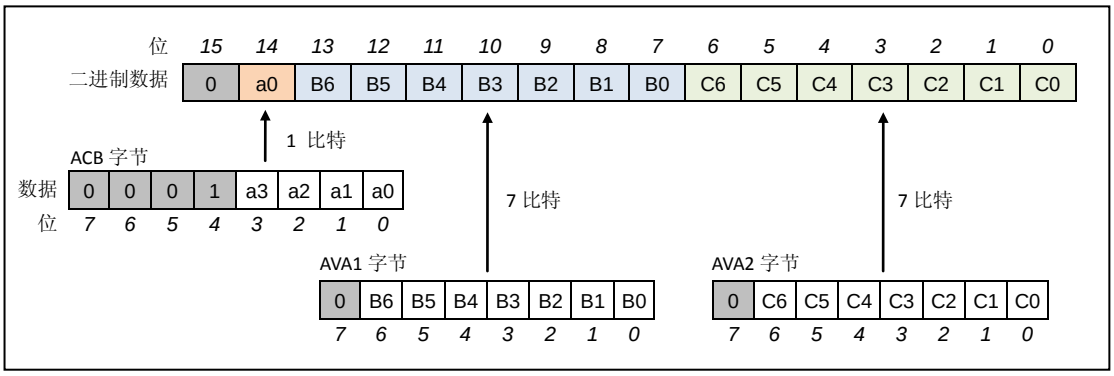
因此： DVA = 0xFF（十六进制）

由此，可使用指令： **DVAx FF;** 来实现要求的端口 AD 转换值查询。

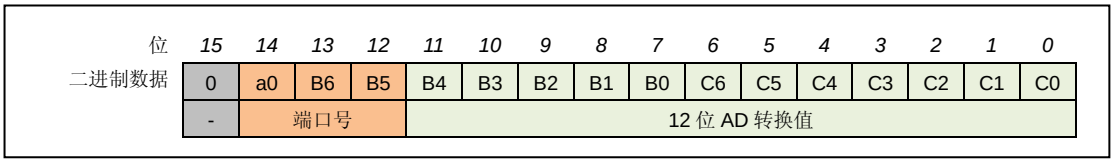
收到 UID828 反馈后，按如下步骤解反馈消息：

如果 Size = 001，表示有 2 个数据字节（即 AVA1, AVA2）。

1）将 ACB 的第 0 位 （即 a0），AVA1 以及 AVA2 按下图方式拼接组成端口数据。



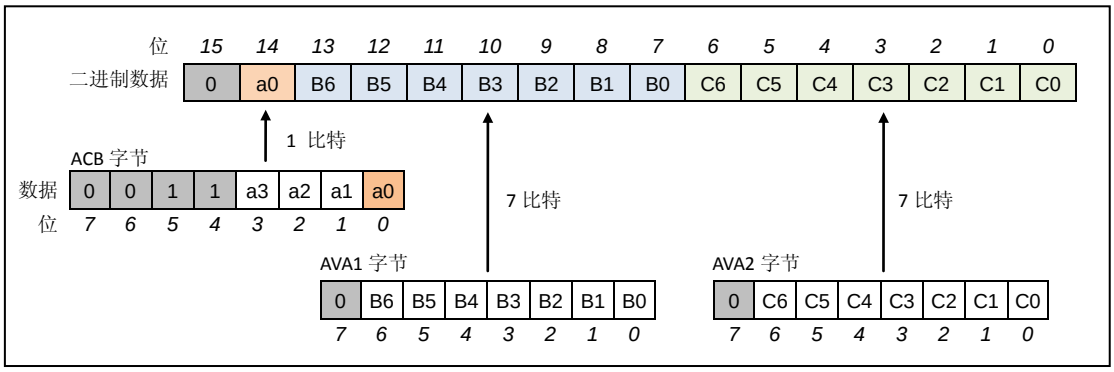
2）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



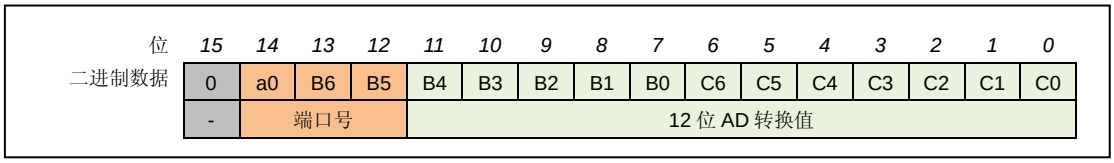
其中，如果端口号= 000，表示端口 1，如果端口号= 001，表示端口 2，如果端口号=111，表示端口 8

如果 Size = 011，表示有 4 个数据字节（即 AVA1, AVA2, AVA3, AVA4）。

1）将 ACB 的第 0 位 （即 a0），AVA1 以及 AVA2 按下图方式拼接组成端口数据。



2）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值

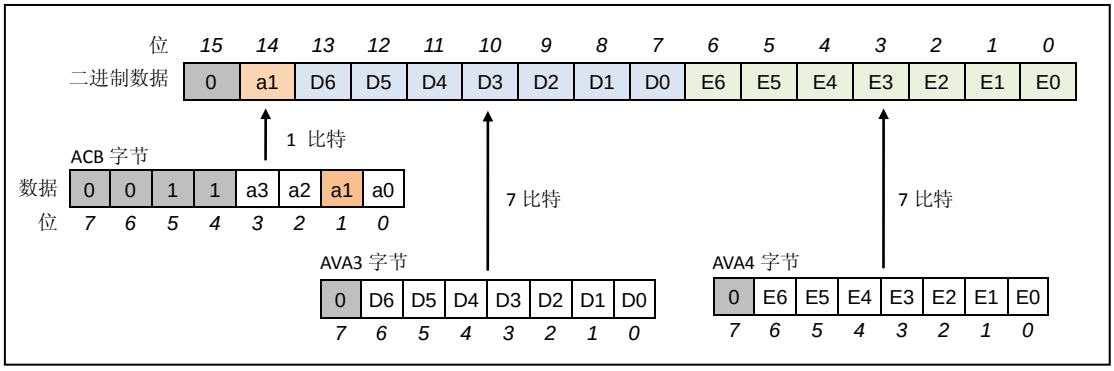


其中，

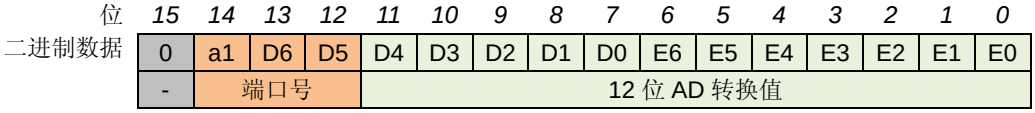
如果端口号 = 000 ，表示 端口 1

如果端口号 = 001 ，表示 端口 2

3）将 ACB 的第 1 位 （即 a1），AVA3 以及 AVA4 按下图方式拼接组成端口数据。



4）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



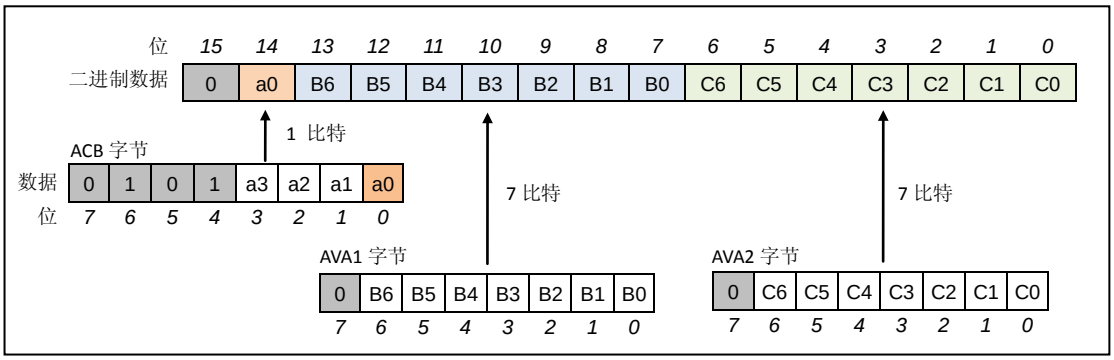
其中，

如果端口号 = 000 ，表示 端口 1

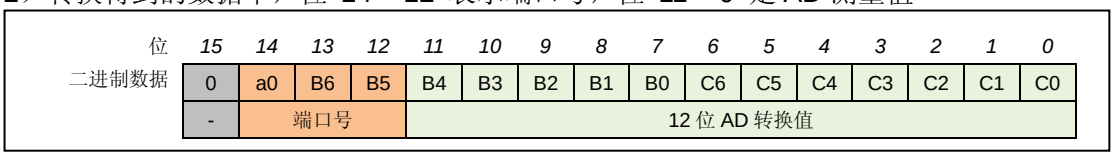
如果端口号 = 001 ，表示 端口 2

如果 Size = 101，表示有 6 个数据字节（即 AVA1, AVA2, AVA3, AVA4, AVA5, AVA6）。

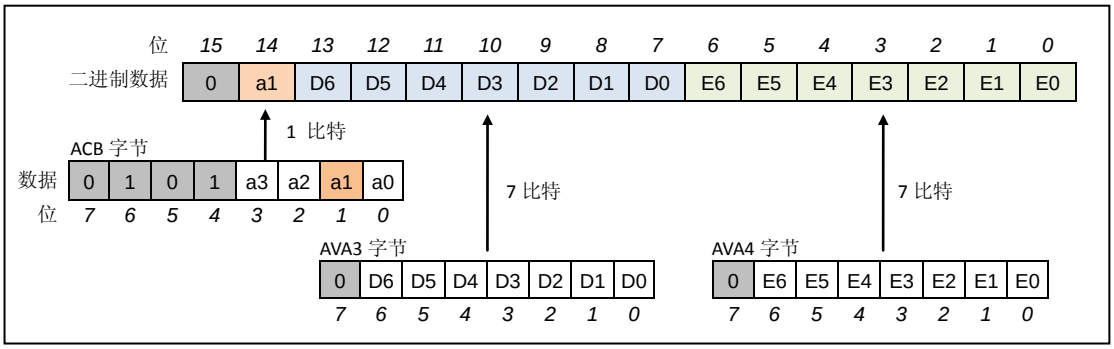
1）将 ACB 的第 0 位 （即 a0），AVA1 以及 AVA2 按下图方式拼接组成端口数据。



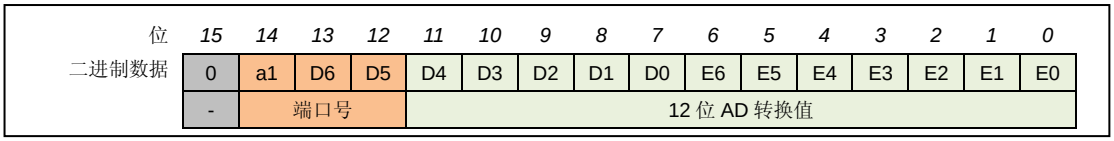
2）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



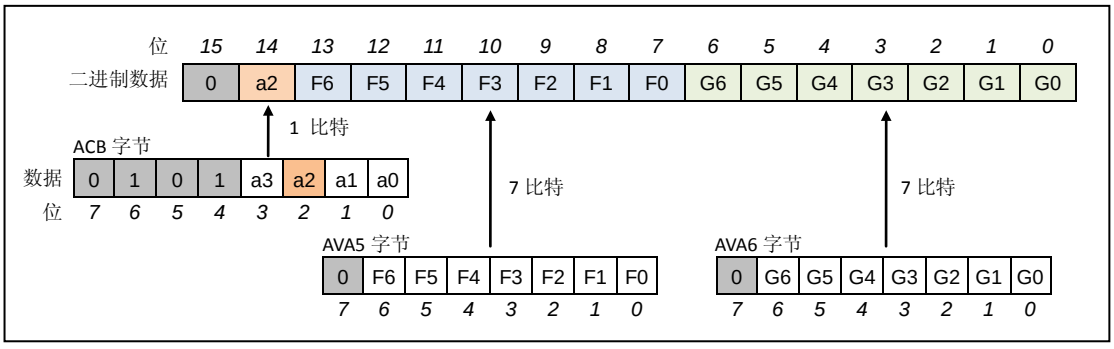
3）将 ACB 的第 1 位 （即 a1），AVA3 以及 AVA4 按下图方式拼接组成端口数据。



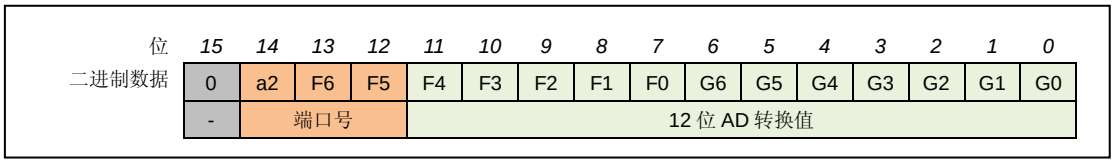
4）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



5）将 ACB 的第 2 位 （即 a2），AVA5 以及 AVA6 按下图方式拼接组成端口数据。

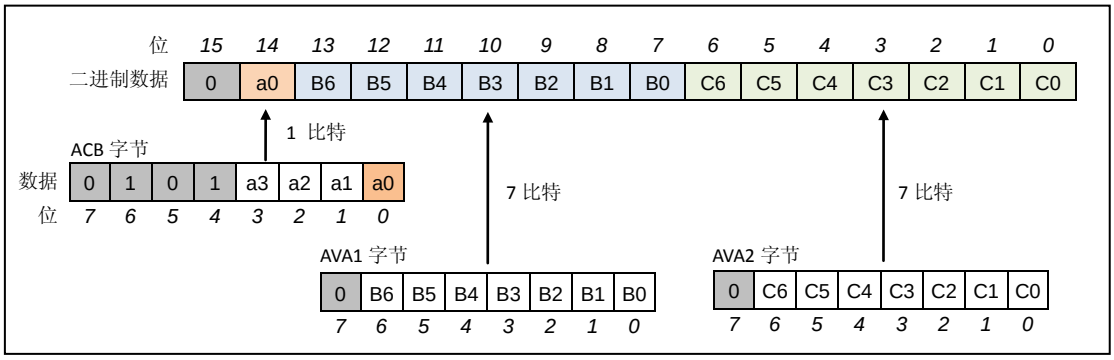


6）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值

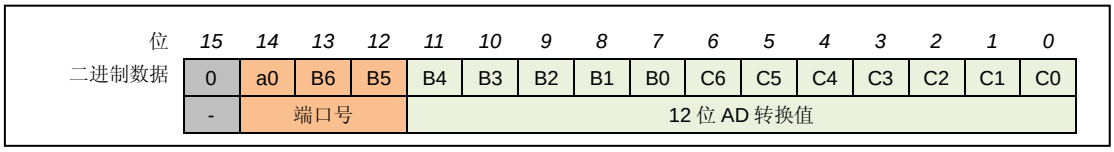


如果 Size = 111，表示有 8 个数据字节（即 AVA1, AVA2, AVA3, AVA4, AVA5, AVA6, AVA7, AVA8）。

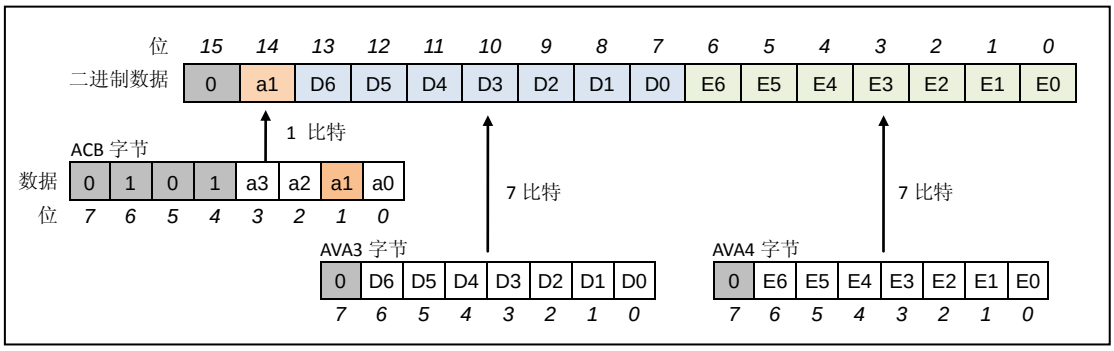
1）将 ACB 的第 0 位 （即 a0），AVA1 以及 AVA2 按下图方式拼接组成端口数据。



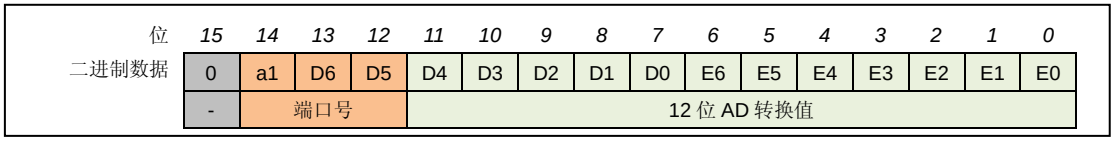
2）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



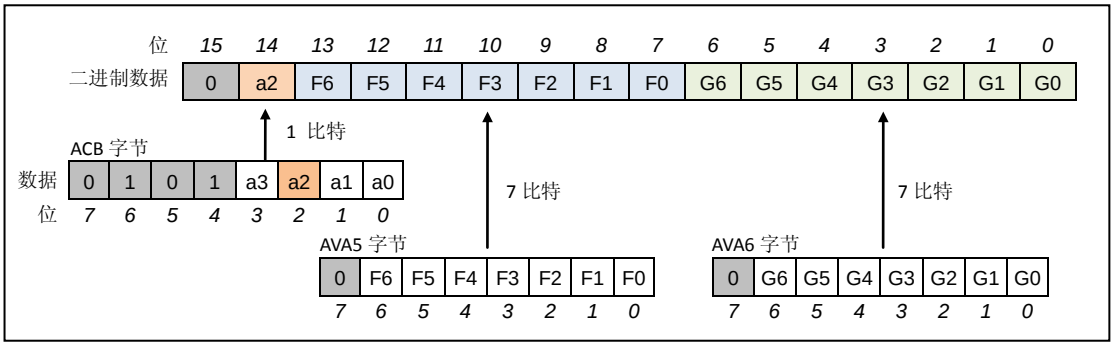
3）将 ACB 的第 1 位 （即 a1），AVA3 以及 AVA4 按下图方式拼接组成端口数据。



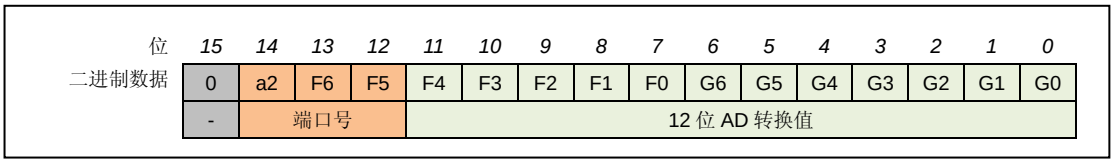
4）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



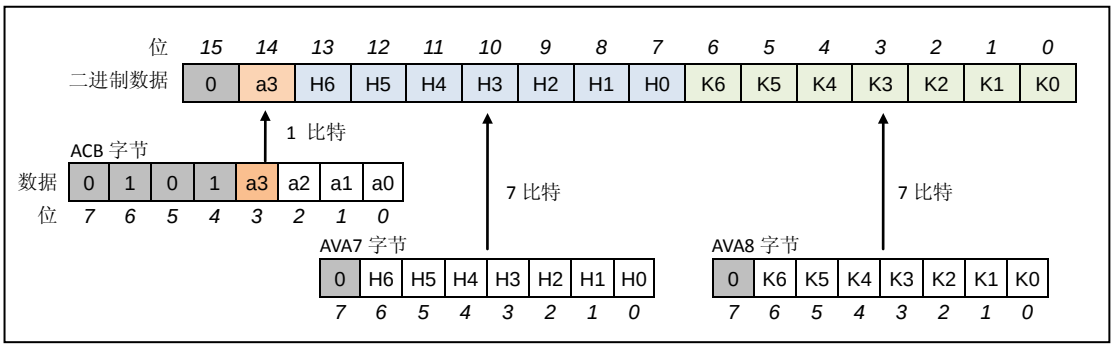
5）将 ACB 的第 2 位 （即 a2），AVA5 以及 AVA6 按下图方式拼接组成端口数据。



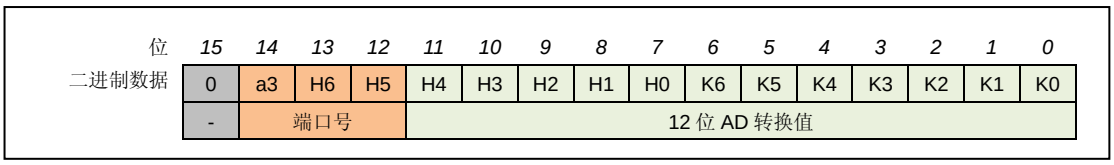
6）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



7）将 ACB 的第 3 位 （即 a3），AVA7 以及 AVA8 按下图方式拼接组成端口数据。



8）转换得到的数据中，位 14 – 12 表示端口号，位 11 – 0 是 AD 测量值



编程示例 6-1

以上 AVA 反馈消息的解析过程可以**使用移位操作很轻松实现**（C 编程）

```
//AVA 反馈消息解析
//ReturnMsg = { [CC]，[ID]，[C9]，[A1]，[A2]，[A3]，[A4]，[A5]，[A6]，[A7]，[A8]，[ACB]，[FF] }
// msg_length 为本条消息的全部字节数，包括CC，ID，C9，FF 在内
void FBK_AVA_MSG(unsigned int msg_length)
{
    unsigned int i = 0;              //用于循环的临时变量
    unsigned int n = 0;              //本消息中包含的端口数量 n
    unsigned int a = 0;              //ACB字节中 第0位 到 第3位
    unsigned int temp = 0;           //临时变量
    unsigned int ADC = 0;            //包含端口号在内的端口AD转换值

    //读取本消息中包含的端口数量 n
    n = ReturnMsg [ msg_length – 1 ]   >> 4;

    //解析并打印每个端口的端口号以及AD转换值
    for (i=0; i<n; i++)
    {
        temp = ReturnMsg[i*2+3];
        a = ( ReturnMsg [ msg_length – 1 ] >> i ) & 0x1;
        ADC = ( a << 14 )   |   ( temp << 7 )   |   ReturnMsg [ i*2 + 4 ];
        printf ("IO%d Reads: %d \n", (ADC>>12)+1, (ADC & 0xFFF));
    }
}
```

2. BTRη　　设置CAN通讯比特率

语　　法：　BTRη;

指令描绘：　设置全局网络的 CAN 通讯比特率η。

η = 0, 1, 2, …, 7。

ACK 报文：　AA　[BTR#]　BC　FF

ACK 解析：　[BTR#]　>>　当前的 CAN 比特率代码；

BC　　>>　CAN 比特率代码的报文标识码。

3. BTR 查询CAN通讯比特率

语 法： BTR;

指令描绘： 查询当前网络的 CAN 通讯比特率。

ACK 报文： AA [BTR#] BC FF

ACK 解析： 参见 BTRη;指令的 ACK 解析。

4. DVAη / DVAxη　端口输出电平控制寄存器设置

语　　法：　DVAη; 或者 DVAxη;

指令描绘：　设定端口输出电平控制寄存器 DVA 的数值η。

η = 0、1…255 （整数） 或者 η = 0x0000…0x00FF （十六进制）

ACK 报文：　AA　[UID828 标识码]　C4　[DVA0]　[DVA1]　FF

ACK 解析：　C4　>>　为 DVAη;指令的报文标识码；

[DVA0] [DVA1]　>>　返回数据 0,1

[DVA0] [DVA1] 转换成 12 位数据后表示 DVA 寄存器数值(见图 7-1)。

注意事项：　如采用整数输入，用户需首先将上述主配置寄存器的各位填写 0 或者 1，然后再将得到的 16 位二进制数转成十进制数，然后将得到的十进制数填入语法结构中η的位置。

需要注意的是，如果某个端口没有配置为数字输出，则该指令对该端口是不起作用的。

例如，端口 7 已经被 IOC 指令配置为 PWM 输出（即 IOC7 = 00 二进制）。此时当用户使用 DVA 指令将 D7 位设置为 0 或 1（即输出高/低电平）时，UID820 不会对端口 7 做任何操作。

示　　例：　当需要 IO1、IO2、IO3、IO4 输出低电平，

IO5、IO6、IO7、IO8 输出高电平时，用户应如下配置 DVA 寄存器：

位	*15*	*14*	*13*	*12*	*11*	*10*	*9*	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*	*0*
定义	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

因此用户应首先保证 IOC = 0101 0101 0101 0101（二进制）= 0x5555。

然后设置 DVA=0000 0000 1111 0000 (二进制) = 0x00F0 (16 进制)。

5. DVA 查询端口输入电平

语 法： DVA;

指令描绘： 当有外部信号接入 UID820，需要知道信号的电平时可以使用该指令查询。

ACK 报文： AA [UID828 标识码] C5 [DVA0] [DVA1] FF

ACK 解析： C5 >> 为 DVA;指令的报文标识码；

[DVA0] [DVA1] >> 返回数据 0,1

[DVA0] [DVA1] 转换成 12 位数据后表示 DVA 寄存器数值(见图 7-1)。

6. IOCη / IOCxη　端口功能配置

语　　法： IOCη; 或者 IOCxη;

指令描绘： 设定端口功能配置寄存器 IOC 的数值η。

η = 0、1…65535 （整数） 或者 η = 0x0000…0xFFFF （十六进制）

ACK 报文： AA [UID828 标识码] C3 [IOC0] [IOC1] [IOC2] FF

ACK 解析： C3 >> 为 IOC 的报文标识码；

[IOC0] ~ [IOC2] >> 返回数据 0 ~ 2

[IOC0] ~ [IOC2] 转换成 16 位数据后表示端口配置寄存器数值(见图 7-1).

注意事项： 如采用整数输入，用户需首先将上述主配置寄存器的各位填写 0 或者 1，然后再将得到的 16 位二进制数转成十进制数，然后将得到的十进制数填入语法结构中η的位置。

使用IOC指令设置端口功能成功后，该设置后一直存在于片上EEPROM，掉电不会丢失。

7. IOC 查询功能配置寄存器

语　　法： IOC;

指令描绘： 查询当前端口功能配置寄存器数值。

ACK 报文： AA [UID828 标识码] C3 [IOC0] [IOC1] [IOC2] FF

ACK 解析： 参见 IOCη;指令 ACK 解析。

8. MCFη / MCFxη 主配置寄存器设置

语　　法： MCFη; 或者 MCFxη;

指令描绘： 设定主配置寄存器数值η。

η = 0、1…255 （整数） 或者 η = 0x0000…0x00FF （十六进制）

ACK 报文： AA [UID828 标识码] B0 [CFG0] [CFG1] [CFG2] FF

ACK 解析： B0 >> 为 MCFG 指令的报文标识码;

[CFG0] ~ [CFG2] >> 返回数据 0 ~ 2

[CFG0] ~ [CFG2] 转换成 16 位数据后表示主配置寄存器数值(见图 7-1)。

注意事项： 如采用整数输入，用户需首先将上述主配置寄存器的各位填写 0 或者 1，然后再将得到的 16 位二进制数转成十进制数，然后将得到的十进制数填入语法结构中η的位置。

9. MCF 查询主配置寄存器

语　　法： MCF;

指令描绘： 查询当前主配置寄存器数值。

ACK 报文： AA [UID828 标识码] B0 [CFG0] [CFG1] [CFG2] FF

ACK 解析： 参见 MCFη;指令 ACK 解析。

10. PBRη　PWM输出时基倍率设置

语　　法： PBRη;

指令描绘： 设定 PWM 波形的时基倍率。

η = 0、1…2500

ACK 报文： AA [UID828 标识码] C8 [PBR0] [PBR1] [PBR2] FF

ACK 解析： C8 >> 为 PBR 的报文标识码；

[PBR0] ~ [PBR2] >> 返回数据 0 ~ 2

[PBR0] ~ [PBR2] 转换成 16 位数据后表示 PBR 寄存器数值(见图 7-1)。

11. PBR　　查询PWM输出时基倍率

语　　法：　PBR;

指令描绘：　查询 PWM 波形的时基倍率。

ACK 报文：　AA　[UID828 标识码]　C8　[PBR0]　[PBR1]　[PBR2]　FF

ACK 解析：　参见 PBRη;指令 ACK 解析。

12. PMBη / PMBxη　　PWM输出时基设置

语　　法：　PMBη; 或者 PMBxη;

指令描绘：　设定 PWM 波形的时基。

η = 0、1…65535 （整数） 或者 η = 0x0000…0xFFFF （十六进制）

ACK 报文：　AA [UID828 标识码] C7 [PB0] [PB1] [PB2] FF

ACK 解析：　C7 >> 为 PWBη;指令的报文标识码；

[PB0] ~ [PB2] >> 返回数据 0 ~ 2

[PB0] ~ [PB2] 转换成 16 位数据后表示 PMB 寄存器数值(见图 7-1)

注意事项：　如采用整数输入，用户需首先将上述主配置寄存器的各位填写 0 或者 1，然后再将得到的 16 位二进制数转成十进制数，然后将得到的十进制数填入语法结构中η的位置。

示　　例：　例 1，当需要 IO8 的基频周期为 3000 微秒，IO7 基频周期为 5000 微秒时，可选取 PBR = 1，则有：

PWMBase2 = 3000 微秒 / （200 微秒 * 1） = 15 (十进制) = 0x0F（十六进制）

PWMBase1 = 5000 微秒 / （200 微秒 *1） = 25 (十进制) = 0x19（十六进制）

所以 PMB 0x0F19;

例 2，当需要 IO8 的基频周期为 3 秒（3000000 微秒），IO7 基频周期为 4 秒（4000000 微秒）时，可选取 PBR = 100，则有：

PWMBase2 = 3000000 微秒 /（200 微秒 * 100）= 150 (十进制) = 0x96（十六进制）

PWMBase1 = 4000000 微秒 /（200 微秒 * 100）= 200 (十进制) = 0xC8（十六进制）

所以 PMB 0x96C8;

13. PMB 查询PWM输出时基

语　　法： PMB;

指令描绘： 查询 PWM 波形的基频周期。

ACK 报文： AA [UID828 标识码] C7 [PB0] [PB1] [PB2] FF

ACK 解析： 参见 PMBη; / PMBxη; 指令的 ACK 解析

14. PMDη / PMDxη PWM输出占空比

语　　法： PMDη; 或者 PMDxη;

指令描绘： 设定 PWM 波形的占空比。

η = 0、1…65535 （整数） 或者 η = 0x0000…0xFFFF （十六进制）

ACK 报文： AA [UID828 标识码] C6 [PD0] [PD1] [PD2] FF

ACK 解析： C6 >> 为 PWDη;指令的报文标识码；

[PD0] ~ [PD2] >> 返回数据 0 ~ 2；

[PD0] ~ [PD2] 转换成 16 位数据后表示 PMD 寄存器数值(见图 7-1)

注意事项： 如采用整数输入，用户需首先将上述主配置寄存器的各位填写 0 或者 1，然后再将得到的 16 位二进制数转成十进制数，然后将得到的十进制数填入语法结构中η的位置。

示　　例： 当需要 IO8 占空比为 30%，IO7 占空比为 50%时

PWMDuty2 = 30% / 0.5% = 60(十进制) = 0x3C (十六进制)

PWMDuty1 = 50% / 0.5% = 100（十进制）= 0x64（十六进制）

因此，PMDx 3C64;

15. PMD　　查询PWM输出占空比

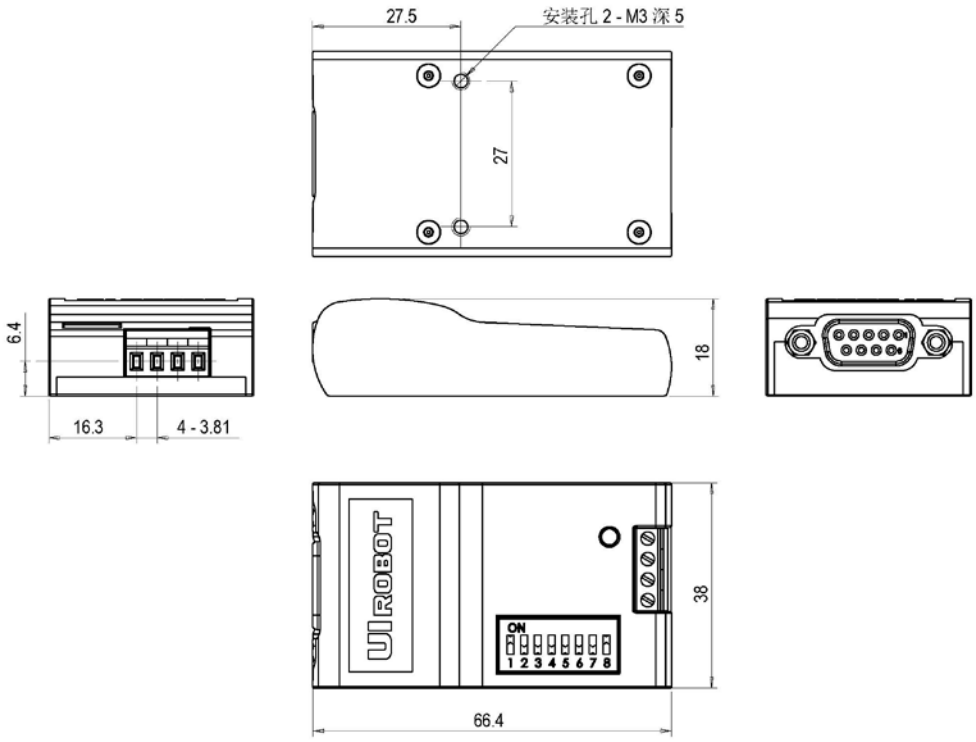
语　　法：　PMD;

指令描绘：　查询 PWM 波形的占空比。

ACK 报文：　AA　[UID828 标识码]　C6　[PD0]　[PD1]　[PD2]　FF

ACK 解析：　参见 PMDη;指令 ACK 解析。

附录A　外形尺寸图



单位：mm