ADH815 自动售货机货道驱动板

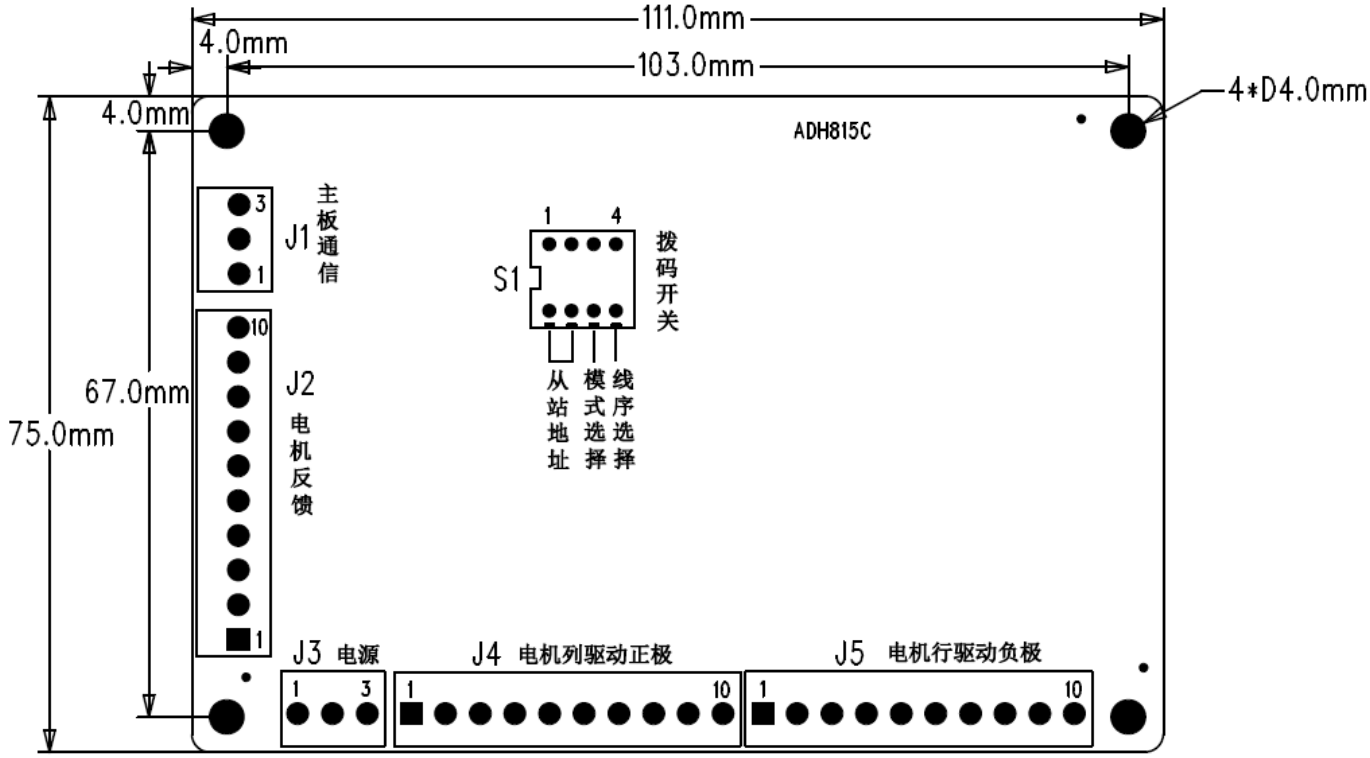
ADH815是一款应用于自动售货机，用于驱动电机、电磁锁等设备的驱动板。RS485接口与售货机主控板进行通信，具有以下的基本特点：

- 一块驱动板最大可驱动 100（10x10）个货道，带货道反馈功能；
- 采用 RS485 通信模式时，驱动板可以进行级联。

技术规格

名称	自动售货机货道驱动板
型号	ADH815
电源输入	24VDC/12VDC
电源输出	24VDC/12VDC,2A（控制电机或电磁锁）
通信接口	1 个 RS485
货道数	100 路，10x10，带反馈
物理特性	
工作温度	0～55℃
存储温度	-40～70℃
相对湿度	5～95%(无凝结)
尺寸规格	111×75（单位：mm）
安装方式	螺钉安装

尺寸结构



- 端子定义及说明

J1（RS485）接主柜	端子说明	J3（电源）	端子说明
1	NC	**1**	电源+24V
2	RS485-	**2**	电源 GND
3	RS485+	**3**	NC
J2（电机反馈）	**端子说明**		
1	输出第 10 行反馈信号	**6**	输出第 5 行反馈信号
2	输出第 9 行反馈信号	**7**	输出第 4 行反馈信号
3	输出第 8 行反馈信号	**8**	输出第 3 行反馈信号
4	输出第 7 行反馈信号	**9**	输出第 2 行反馈信号
5	输出第 6 行反馈信号	**10**	输出第 1 行反馈信号
J5（电机驱动行）	**端子说明**		
1	第 1 行输出	**6**	第 6 行输出
2	第 2 行输出	**7**	第 7 行输出
3	第 3 行输出	**8**	第 8 行输出
4	第 4 行输出	**9**	第 9 行输出
5	第 5 行输出	**10**	第 10 行输出
J4（电机驱动列）	**端子说明**		
1	第 1 列输出	**6**	第 6 列输出
2	第 2 列输出	**7**	第 7 列输出
3	第 3 列输出	**8**	第 8 列输出
4	第 4 列输出	**9**	第 9 列输出
5	第 5 列输出	**10**	第 10 列输出

S1（拨码开关）				
1/2	**OFF/OFF**	**ON/OFF**	**OFF/ON**	**ON/ON**
	驱动板地址为1	驱动板地址为2	驱动板地址为 3	驱动板地址为 4
3	**ON**	电机模式	**OFF**	电磁锁模式
4	**ON**	行、列、反馈均为从10-1	**OFF**	行、列、反馈均为从 1-10

\注 1：驱动采用矩阵扫描方式，即一个行输出信号和一个列输出信号共同驱动一个电机（或电磁锁），列输出信号接电机（或电磁锁）的正极，行输出信号接电机（或电磁锁）的负极。

- **通讯规约**

1. 通讯参数

主机与外设之间采用串口方式进行通讯。通讯参数为 9600，n，8，1。

2. 指令格式

主机至从机的指令由“从机地址”,“指令”,“数据”和”校验代码”三部分组成。

从机地址	指令	数据	校验代码

从机响应主机时采用“主机地址”+“指令”+“数据”+“校验代码”的方式

主机地址(0x00)	指令	数据	校验代码

备注：地址 1 字节 、指令1 字节 、数据 n 字节、校验代码2 字节；数据字段中16 位数据高字节在前低字节在后；校验代码字段中 低字节在前高字节在后。

驱动板详细指令格式

指令	代码	描述
ID	01H	查询身份信息
POLL	03H	查询状态
RUN	05H	启动电机
ACK	06H	确认主机已经收到当前运行状态

3. 详细指令模式

指令	代码	返回数据
POLL	03H	9 Bytes

POLL 可能回应零条或多条消息。如果没有消息，则驱动模块回发 ACK 来响应 POLL。

Z1	控制板状态	0=空闲 1=出货中 2=出货结束（注意仅表示结束，不代表成功或者失败）
Z2	当前操作的电机索引	00~79
Z3	电机操作结果	0= 无故障 1=过流 2=断线 3=未检测到停止信号
Z4	最大电流高字节	毫安
Z5	最大电流低字节	
Z6	平均电流高字节	毫安
Z7	平均电流低字节	
Z8	运行时间高字节	毫秒
Z9	运行时间低字节	

指令	代码	设置数据
RUN	05H	Z1

用于通知驱动模块启动电机。设置参数如下

Z1 = 1 字节 表示电机索引号（00~79）。

响应数据

Y1 = 1 个字节 表示是否启动成功，0 表示成功，大于 0 时表示失败，具体的值来表示失败的原因。1 表示无效的电机索引号，

2 表示当前有另一个电机正在运行，3 表示另一台电机的运转结果还未取走

指令	代码
ACK	06H

用于通知驱动模块主机已经获取到了上次运行的结果

4. 通讯数据例子

（例子中从机地址 设置成了 2 号）

ID

02 01 C1 10

(47 ms)

00 01 00 00 A4 3C 47 0A

阴影部分为 id

POLL

空闲

02 03 40 D1

(62 ms)

00 03 00 00 00 00 00 00 00 00 00 F6 EB

启动 OA 号马达

02 05 0A 53 57

(47 ms)

00 05 00 72 90

表示启动成功

02 03 40 D1

(47 ms)

00 03 02 0A 00 03 E0 00 D1 09 E7 F8 32

02=执行完毕

0A=10 号电机

00=无故障

03E0 =最大电流 992mA

00D1=平均电流 209mA

09E7=运行时间 2535ms

02 03 40 D1

(47 ms)

00 03 02 0A 00 03 44 00 47 09 DA A8 16

最大电流 0344 836mA 平均电流 0047 71mA 运行时间 09DA 2522ms

ACK 指令

```
02 06 80 D2
(31 ms)
00 06 81 B2
```

ACK 完毕后 可以使用 POLL 指令 核实一下

```
02 03 40 D1
(47 ms)
00 03 00 00 00 00 00 00 00 00 00 F6 EB
```

5. CRC 校验代码

```
const unsigned int CrcTbl[256]=
{
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
    0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
    0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
    0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
    0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
    0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
    0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
    0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
    0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
    0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
    0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
    0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
    0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
    0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
    0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
    0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
    0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
    0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};
```

```
//查表法计算 CRC-16
unsigned int crcVal(unsigned char *pcMess,unsigned int wLen)
{
	unsigned int Index;
	unsigned int nCRCData;
	nCRCData=0xffff;
	while(wLen--)
	{
		Index=nCRCData>>8;
		Index=Index^(*pcMess++&0x00ff);
		nCRCData=(nCRCData^CrcTbl[Index])&0x00ff;
		nCRCData=(nCRCData<<8)|(CrcTbl[Index]>>8);
	}
	return (nCRCData>>8|nCRCData<<8);
}
/*************************************************************
功能： CRC 校验
pcMess: 待求数据的首指针
wLen : 待求数据的长度
*************************************************************/
unsigned int crcVal1(unsigned char *pcMess,unsigned int wLen)
{
	long MSBInfo;
	int i,j ;
	unsigned int nCRCData;

	nCRCData = 0xffff;
	for(i = 0; i < wLen;i++)
	{
		pcMess[i]&=0x00ff;
		nCRCData = nCRCData ^ pcMess[i] ;
		for(j= 0 ; j < 8 ;j ++)
		{
			MSBInfo = nCRCData & 0x0001;
			nCRCData = nCRCData  >> 1;
			if(MSBInfo != 0 )
				nCRCData = nCRCData ^ 0xa001;
		}
	}
	return nCRCData;
}
```