
GCAN-IO-8000

CANopen总线耦合器

使用文档

文档版本：V4.00 （2021/03/23）

GCAN8000 简介

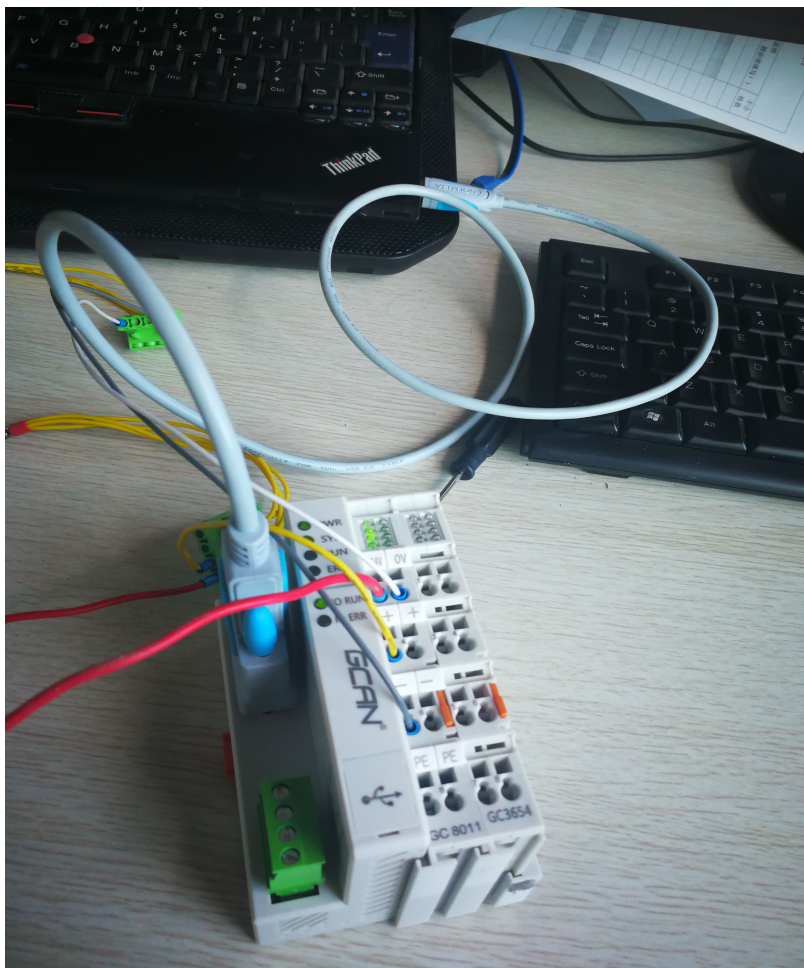
我们的 GCAN8000 产品是一个标准的 CANOPEN 从站，无需编程，可直接使用，可通过 232 端口修改波特率，设备节点号，IP 地址等参数并查询。

1. 修改 ID 及波特率

下图是 RS232 线的引脚定义

引脚序号	名 称	作 用	备 注
1	DCD (Data Carrier Detect)	数据载波检测	
2	RxD (Received Data)	串口数据输入	
3	TxD (Transmitted Data)	串口数据输出	
4	DTR (Data Terminal Ready)	数据终端就绪	
5	GND (Signal Ground)	地线	
6	DSR (Data Send Ready)	数据发送就绪	
7	RTS (Request to Send)	发送数据请求	
8	CTS (Clear to Send)	清除发送	
9	RI (Ring Indicator)	铃声指示	

通过 232 端口，可以修改相应的参数。具体操作如下，将 GCAN8000 与电脑用 232 线连接起来，如图所示



然后打开设备管理器，查看 COM 口，然后打开串口调试助手，如下图设置：选择电脑读到的 COM 口，波特率 19200，停止位 1，数据位 8，奇偶校验无，发送英文字符 `help` 或者 `?`，勾选发送新行，点击发送，就会出现如图的提示，根据提示输入相应的命令就可以修改对应的参数了。

例如：`getid`:可以获得本机 canopen 节点 ID;

`setid`:设置本机 canopen 节点 ID,方法：`setid=1`,即 ID 号为 1，范围是 1~127.

`getbaud`:获取本机 can 波特率。

`setbaud`: 设置本机波特率，方法：`setbaud=250000`,即设置波特率为 250K，支持 canopen 的 7 种标准波特率。



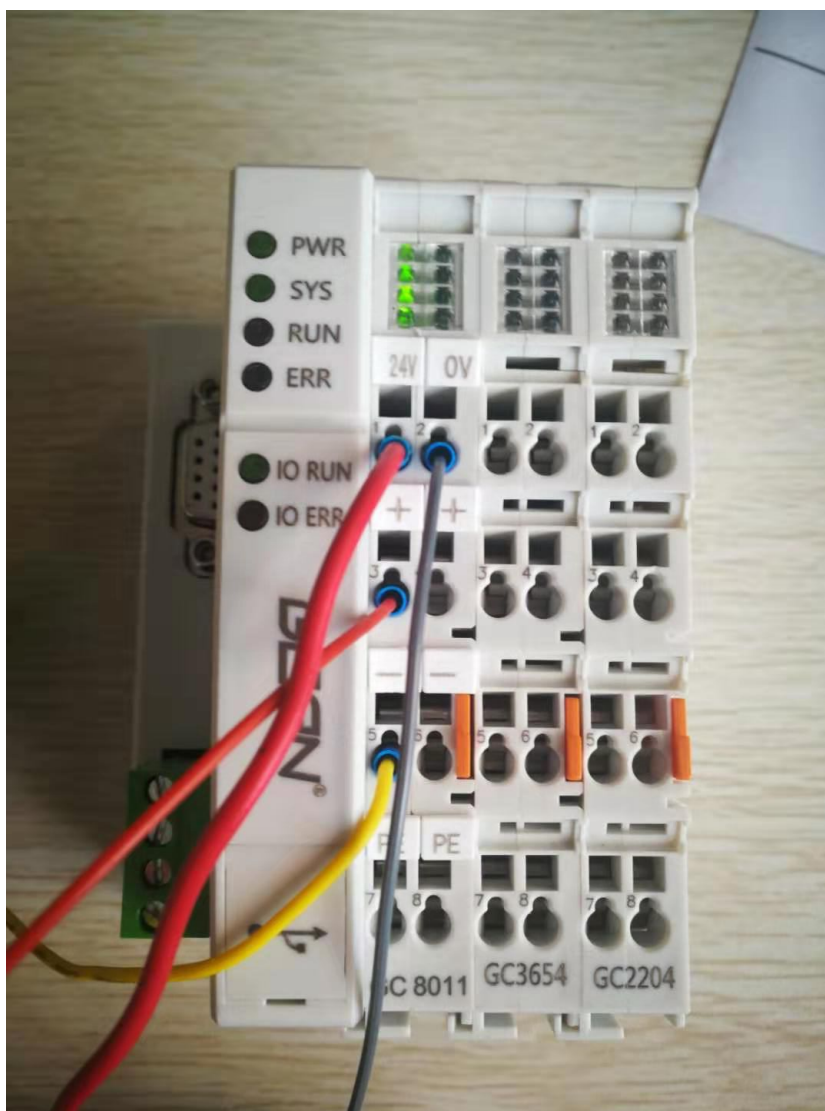
(注：不能勾选 16 进制发送和 16 进制显示)

2. 接线

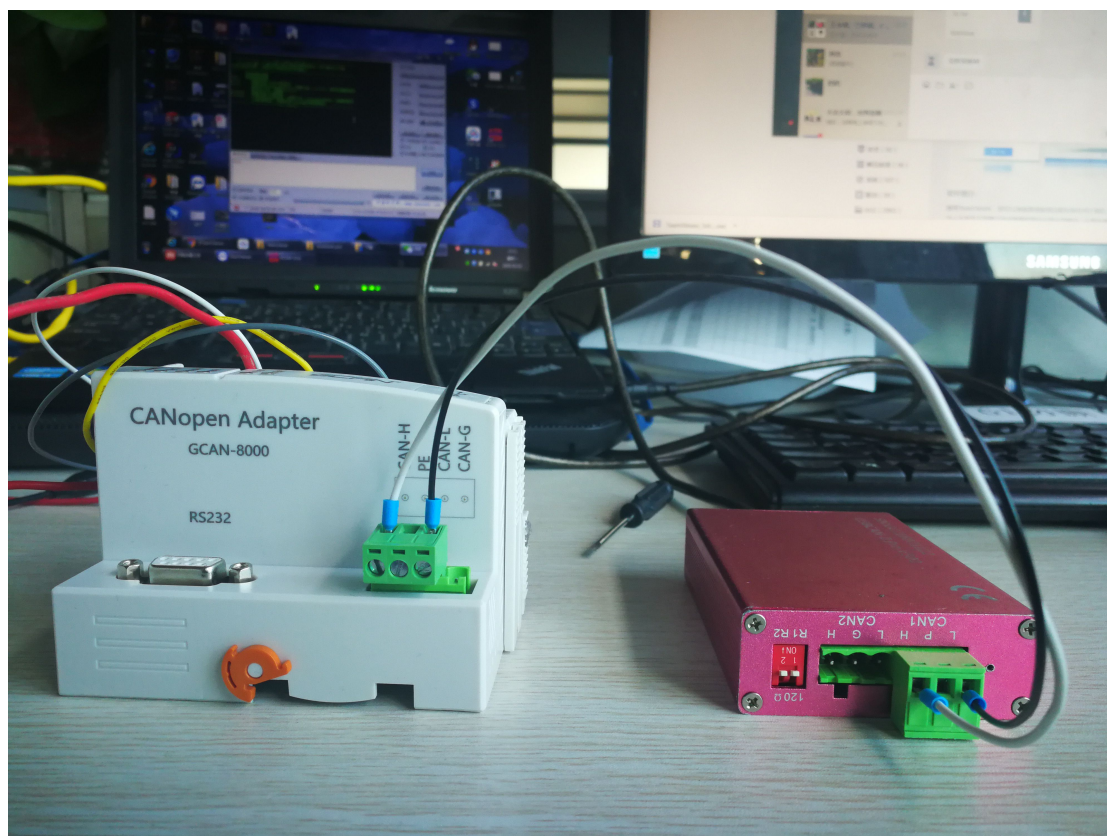
如图所示，

通电：1 和 2 分别连 24V 和 0V，3 或 4 其中一个接入 24V 即可，5 或 6 其中一个接入 0V 即可。PE 不接。这样，GCAN8000 就正常供电了。正常状态应该是 PWR 灯常亮，SYS 灯闪烁，RUN 灯闪烁，IO RUN 灯闪烁。GC8011 电源模块的左侧 4 个灯亮起。

(注：若 IO ERR 灯亮，则表示后面的 IO 片没接)



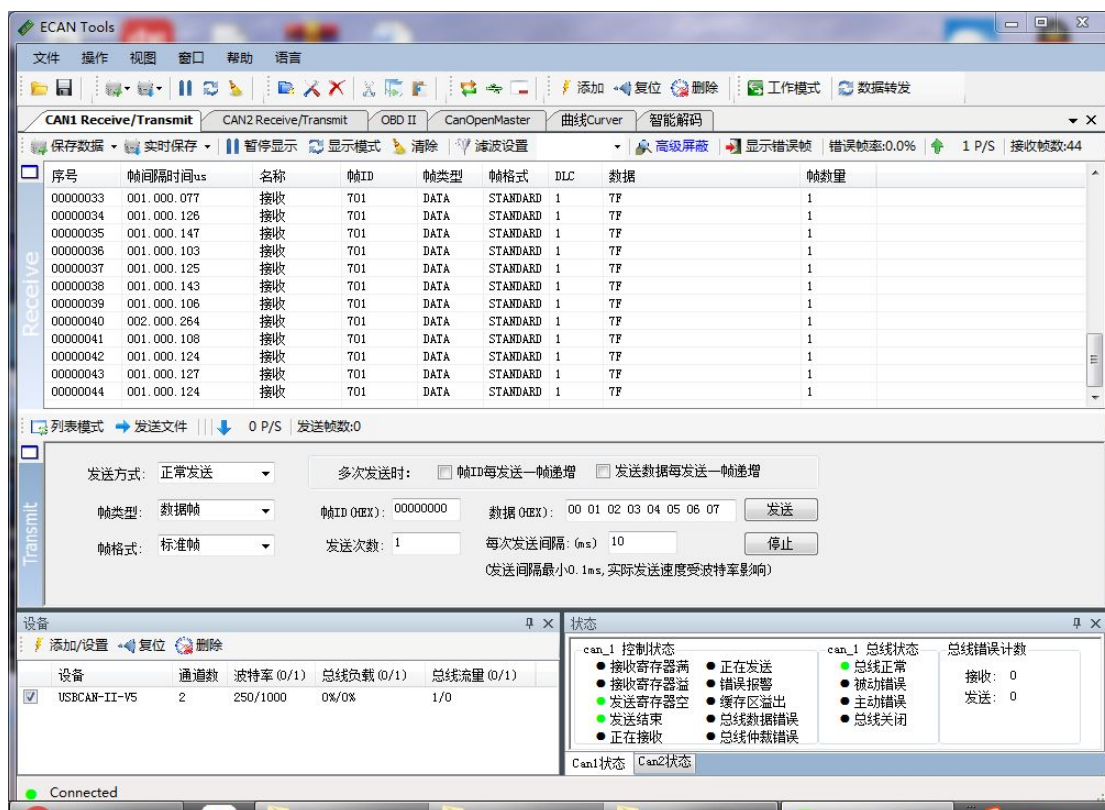
可以用我们公司的can分析仪和ECANTOOLS软件与GCAN8000进行收发数据，连接如图所示。CAN分析仪使用的是CAN1通道，将120 Ω 拨码开关拨到ON，CANL接CAN1的L，CANH接CAN1的H。



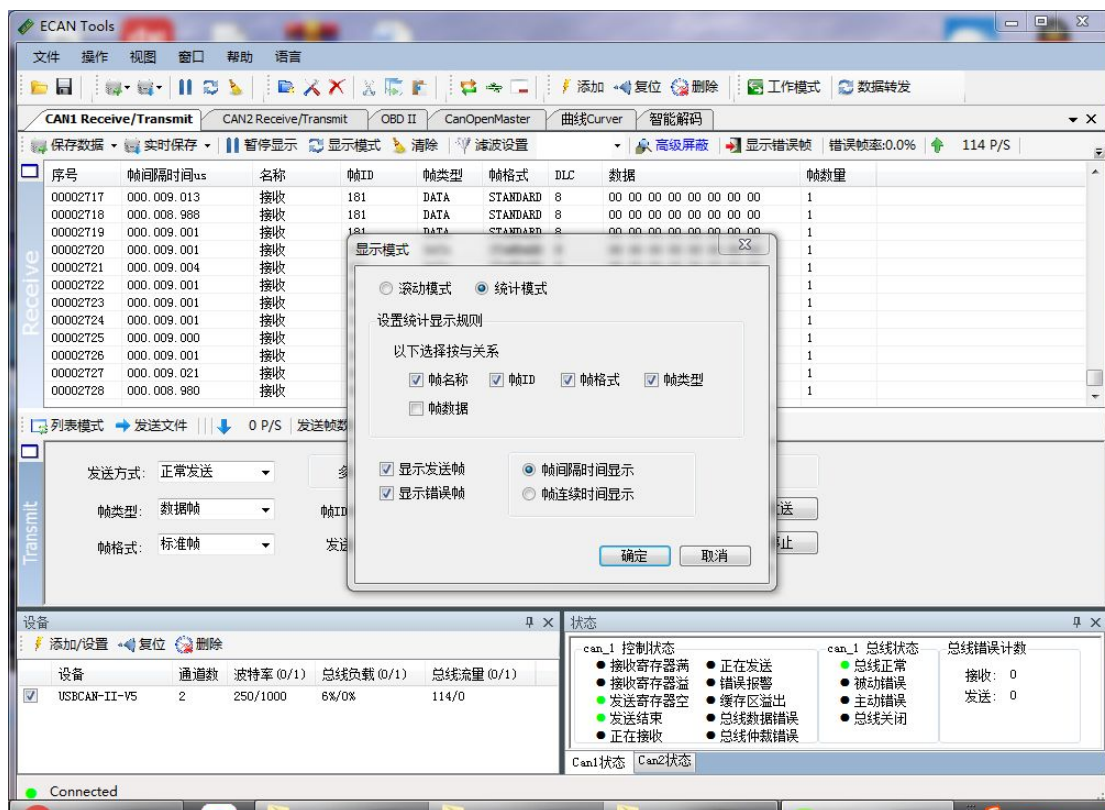
连接好之后打开 ECANTOOLS 软件，点击打开设备，用的 CAN1 通道就选择 CAN1，用的 CAN2 通道就选择 CAN2.波特率默认 250K，节点号默认是 1，设定好之后点击确定。



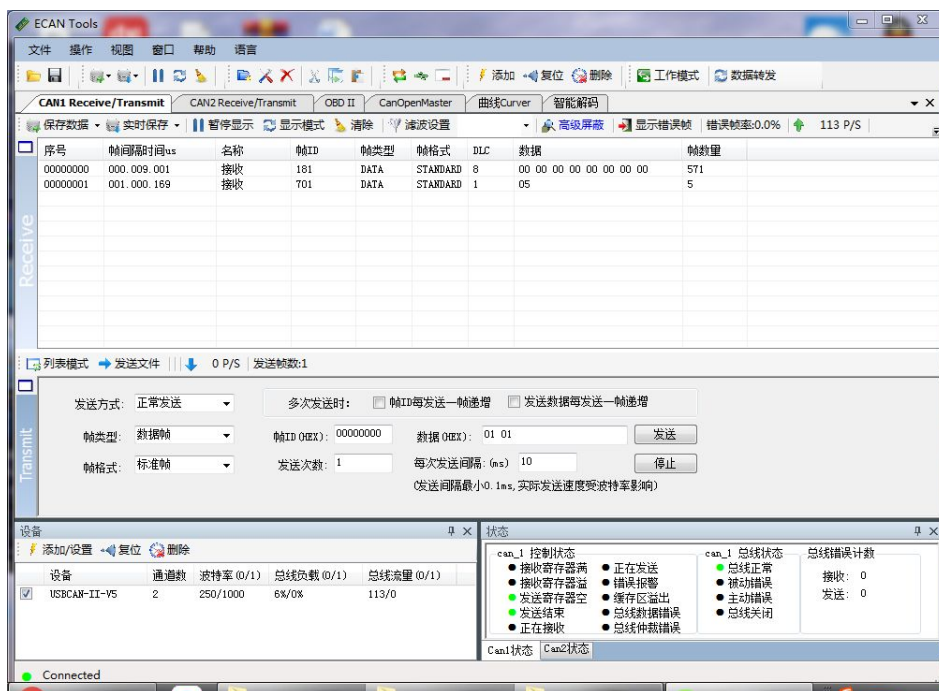
会出现如下图所示的心跳指令，帧 ID 为 701，数据为 7F。



将显示模式改成统计模式



如图所示，帧 ID 为 0，发送 01 01 数据启动之后，会收到两种数据，一种是帧 ID 为 181 的 8 字节数据，这是因为 GCAN8000 接了一个 3654(后面会详细讲解)；和一个帧 ID 为 701 的数据，显示为 05，只有一个字节，一秒钟 1 帧。这个是心跳数据，无论 GCAN8000 后面是否接 IO 模块，只要发送启动指令后，就会接收到这个数据。



3. 与 IO 片连接

搭载 GC-1008 模块

数字量输入的状态由一个字节来表示，通道 8 在高位，通道 1 在低位。

例如，GCAN-IO-8000 模块节点号设为 1。通道 8 和通道 4 状态为 1，其他状态均为 0，则 CAN 总线一端显示的 DI 状态数据为 88。则发出的帧 ID 为 0x181，数据长度（DLC）为 8，帧数据为 0x88, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00。请注意，仅插入一个 GCAN-1008 模块时，帧数据中只有第一个字节是有效的。下表列举了两种常见的 DI 状态及其对应的状态数据。

DI 状态								
通道数	8	7	6	5	4	3	2	1
状态	1	0	0	0	1	0	0	0
CAN 总线显示的数据	88							

DI 状态								
通道数	8	7	6	5	4	3	2	1
状态	0	1	0	1	1	0	1	0
CAN 总线显示的数据	5A							

仅插入一个 GCAN-1008 模块时，GCAN-IO-8000 模块发出的 TPDO 数据帧 ID 为 0x180+节点号（Node ID），数据长度为 8，帧数据第一个字节即为该模块数字量输入状态。

搭载 GC-2008 模块

数字量输出的状态由一个字节来表示，通道 8 在高位，通道 1 在低位。

例如，GCAN-IO-8000 模块节点号设为 1。需设置通道 8 和通道 4 状态为 1，设置其他状态均为 0，则需要发送的 CAN 总线 DO 状态数据为 88。需要发送给 GCAN-IO-8000 的帧 ID 为 0x201，数据长度（DLC）为 8，帧数据为 0x88, 0x00,

0x00, 0x00, 0x00, 0x00, 0x00, 0x00。请注意，帧数据中只有第一个字节是有效的。下表列举了两种常见的 DO 状态及其对应的状态数据。

DO 状态								
通道数	8	7	6	5	4	3	2	1
状态	1	0	0	0	1	0	0	0
CAN 总线显示的数据	88							

DO 状态								
通道数	8	7	6	5	4	3	2	1
状态	0	1	0	1	1	0	1	0
CAN 总线显示的数据	5A							

GCAN-IO-8000 模块在接收 RPDO 数据时，需保证帧 ID 为 0x200+节点号 (Node ID)，数据长度为 8，帧数据第一个字节即为需要设置的数字量输出状态。

搭载 GC-3804 模块

每个通道的温度状态由两个字节来表示，四个通道共八个字节。

其中，代表温度状态的两个字节，第一个字节为低位，需将该字节的数据转换为十进制后乘以 0.1；第二个字节为高位，需将该字节的数据转换为十进制之后乘以 25.6。最后将两个数值加和，即为最终的温度值，单位为摄氏度。

例如，GCAN-IO-8000 模块节点号设为 1。四个通道的温度分别为 25.6 度，25.5 度，20 度，30 度。则发出的帧 ID 为 0x181，数据长度 (DLC) 为 8，帧数据为 0x00, 0x01, 0xFF, 0x00, 0xC8, 0x00, 0x2C, 0x01。下表列举了两种可能的 CAN 数据及其对应的温度值。

GC-3804 温度与 CAN 数据对应关系		
CAN 总线显示的数据	低字节 C8	高字节 00

系数	200 (0xC8) x0.1	0 (0x00) x25.6
温度值	20℃	

GC-3804 温度与 CAN 数据对应关系		
CAN 总线显示的数据	低字节 2C	高字节 01
系数	44 (0x2C) x0.1	1 (0x01) x25.6
温度值	30℃	

GCAN-IO-8000 模块在发出 TPDO 数据时, 帧 ID 为 0x180+节点号(Node ID), 数据长度为 8。如未接入 PT100, 则对应通道的 CAN 数据会显示为 FF 7F。

4.5 同时搭载多组模块

若 GCAN-IO-8000 同时搭载多组 GC-1008 模块, 那么我们以它们距离 GCAN-IO-8000 的远近, 从近到远进行编号, 离得最近的为 1 号。GCAN-IO-8000 耦合器发出的 TPDO 数据将按照下表进行发送。例如, 当 GCAN-IO-8000 节点号为 3 且同时搭载 9 个 GC-1008 模块时, 您将收到帧 ID 为 0x183 和 0x283 两组数据。其中, 帧 ID 为 0x183 的八个数据字节依次对应 1-8 号 GC-1008 模块; 帧 ID 为 0x283 的一个数据字节对应 9 号 GC-1008 模块。

帧 ID	帧数据							
0x180+Node ID	1 号	2 号	3 号	4 号	5 号	6 号	7 号	8 号
0x280+Node ID	9 号	10 号	11 号	12 号	13 号	14 号	15 号	16 号
0x380+Node ID	17 号	18 号	19 号	20 号	21 号	22 号	23 号	24 号
0x480+Node ID	25 号	26 号	27 号	28 号	29 号	30 号	31 号	32 号

表 4.1 多组 GC-1008 模块与 CAN 数据对应关系

若 GCAN-IO-8000 同时搭载多组 GC-2008 模块, 那么我们以它们距离 GCAN-IO-8000 的远近, 从近到远进行编号, 离得最近的为 1 号。GCAN-IO-8000 耦合器接收的 RPDO 数据需按照下表进行发送。例如, 当 GCAN-IO-8000 节点

号为 3 且同时搭载 9 个 GC-2008 模块时，您需要发送帧 ID 为 0x203 和 0x303 两组数据以控制全部的 9 个模块。其中，帧 ID 为 0x203 的八个数据字节依次对应 1-8 号 GC-2008 模块；帧 ID 为 0x303 的一个数据字节对应 9 号 GC-2008 模块。

帧 ID	帧数据							
0x200+Node ID	1 号	2 号	3 号	4 号	5 号	6 号	7 号	8 号
0x300+Node ID	9 号	10 号	11 号	12 号	13 号	14 号	15 号	16 号
0x400+Node ID	17 号	18 号	19 号	20 号	21 号	22 号	23 号	24 号
0x500+Node ID	25 号	26 号	27 号	28 号	29 号	30 号	31 号	32 号

表 4.2 多组 GC-2008 模块与 CAN 数据对应关系

若 GCAN-IO-8000 同时搭载多组 GC-3804 模块，那么我们以它们距离 GCAN-IO-8000 的远近，从近到远进行编号，离得最近的为 1 号。GCAN-IO-8000 耦合器发出的 TPDO 数据将按照下表进行发送。例如，当 GCAN-IO-8000 节点号为 3 且同时搭载 4 个 GC-3804 模块时，您将收到帧 ID 为 0x183 等四组数据。其中，帧 ID 为 0x183 的八个数据字节依次对应 1 号 GC-3804 模块的 4 个通道；帧 ID 为 0x283 的八个数据字节依次对应 2 号 GC-3804 模块的 4 个通道。

帧 ID	帧数据
0x180+Node ID	1 号 GC-3804 的 4 个通道
0x280+Node ID	2 号 GC-3804 的 4 个通道
0x380+Node ID	3 号 GC-3804 的 4 个通道
0x480+Node ID	4 号 GC-3804 的 4 个通道

表 4.3 多组 GC-3804 模块与 CAN 数据对应关系

若 GCAN-IO-8000 同时搭载 GC-1008 模块与 GC-3804 模块，那么我们以它们距离 GCAN-IO-8000 的远近，从近到远进行编号，离得最近的为 1 号。

GCAN-IO-8000 耦合器发出的 TPDO 数据将按照下表进行发送。例如，当 GCAN-IO-8000 模块节点号为 3 且同时搭载 3 个 GC-1008 模块和 1 个 GC-3804 模块时，您将收到帧 ID 为 0x183 和 0x283 两组数据。数据对应关系如下表所示。

帧 ID	帧数据							
0x180+Node ID	1 号 GC-100 8	2 号 GC-100 8	3 号 GC-380 4 第一	3 号 GC-380 4 第二	3 号 GC-380 4 第三	3 号 GC-380 4 第四	3 号 GC-380 4 第五	3 号 GC-380 4 第六

			个字节	个字节	个字节	个字节	个字节	个字节
0x280+Node ID	3 号 GC-380 4 第七 个字节	3 号 GC-380 4 第八 个字节	4 号 GC-100 8	00	00	00	00	100

表 4.4 多组 GC-1008 模块与 GC-3804 模块混用时与 CAN 数据的对应关系

搭载 GC-2302 模块

一个 2302 有两个通道，即 ch1 和 ch2. 2302 的 device 从靠近 8000 的开始是 1，第二个是 2，以此类推。

1:电机模式初始化:

canopen index=0x3000,subindex=0,data=pulse(2 byte)+device index(1 byte)+channel(1 byte)

example:device=1,ch=1,pulse=5000

send pdo: 0x601 23 00 30 00 01 01 88 13

example:device=1,ch=2,pulse=5000

send pdo: 0x601 23 00 30 00 01 02 88 13

example:device=2,ch=1,pulse=5000

send pdo: 0x601 23 00 30 00 02 01 88 13

example:device=2,ch=2,pulse=5000

send pdo: 0x601 23 00 30 00 02 02 88 13

2:电机模式使能:

canopen index=0x3001,subindex=0,data=00+ EN(1 byte)+device index(1 byte)+channel(1 byte)

example:device=1,ch=1,EN=1

send pdo:0x601 23 01 30 00 01 01 01 00

example:device=1,ch=1,EN=0

send pdo:0x601 23 01 30 00 01 01 00 00

example:device=1,ch=2,EN=1

send pdo:0x601 23 01 30 00 01 02 01 00

先发送初始化命令，再发送使能命令，2302 才会工作。

3:速度控制：

发送 PDO 以控制电机速度、速度值（2 字节），对于 0.1x，EX：如果您设置速度 Vale 为 1000，则实际电机速度为 $1000 \times 0.1 = 100$

例如：GCAN8000 有 1 个 GC2302 模块，Canopenid=1，ch1 速度=1000，ch2 速度=2000

发送 PDO 数据：0x201 E8 03 D0 07 00 00 00 00

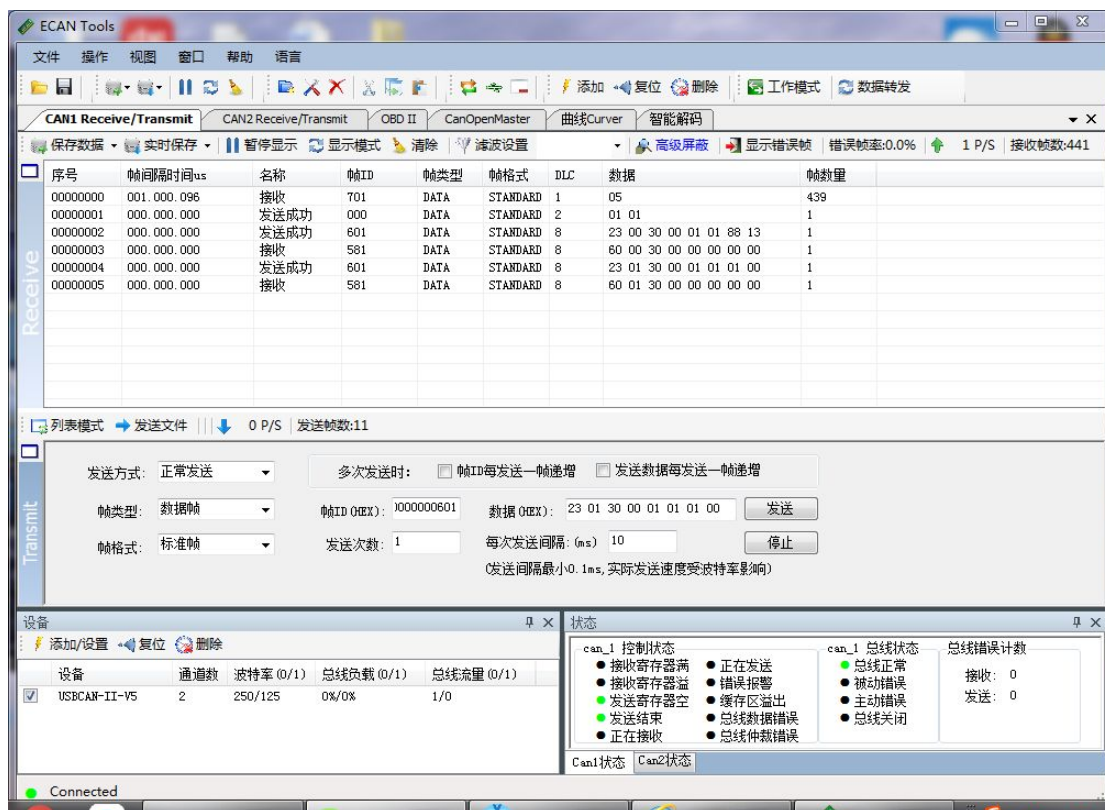
如果你停止 CH1 电机，发送 PDO 值为零

比如:0x201 00 00 D0 07 00 00 00 00

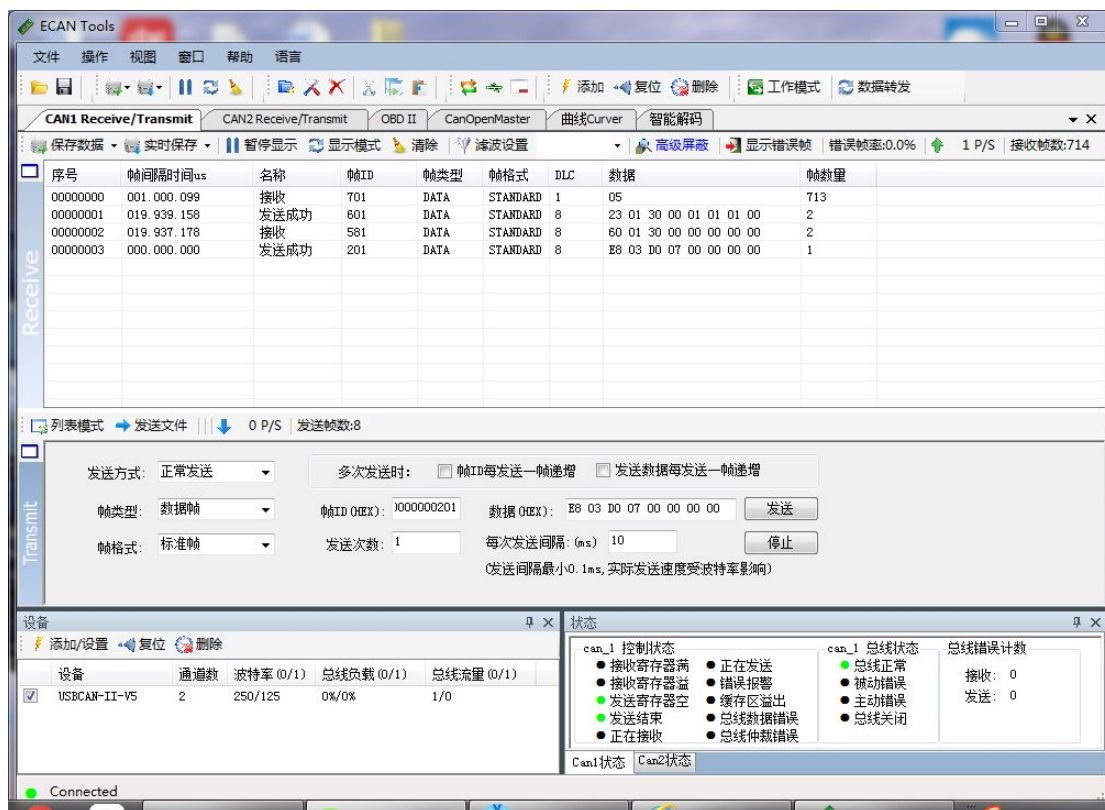
帧 ID	帧数据							
0x601	23	00	30	00	device	ch	Pulse (低字节)	Pulse (高字节)
0x601	23	01	30	00	device	ch	EN	00
0x201	Ch1 速度 (低字节)	Ch1 速度 (高字节)	Ch2 速度 (低字节)	Ch2 速度 (高字节)	00	00	00	00

下面举例说明：

如图所示：依次发送 01 01 启动指令、电机初始化指令、电机使能指令。



然后发送电机速度控制指令，如图所示，发送的是 E8 03 D0 07 00 00 00 00，表示 ch1 电机速度是 1000，ch2 电机速度是 2000，实际电机速度 ch1 为 100，ch2 为 200。



搭载 GC-3654 模块

一个 3654 有 4 个通道，一个通道占两个字节，高字节在后，低字节在前。其他 36x4 系列产品数据格式同 3654。若搭载多个 3654，则对应关系如下表。

3604: -5V~+5V 对应-32768~+32767

3624: -10V~+10V 对应-32768~+32767

3654: 4~20mA 对应 0~65535

3644: 0~20mA 对应 0~65535

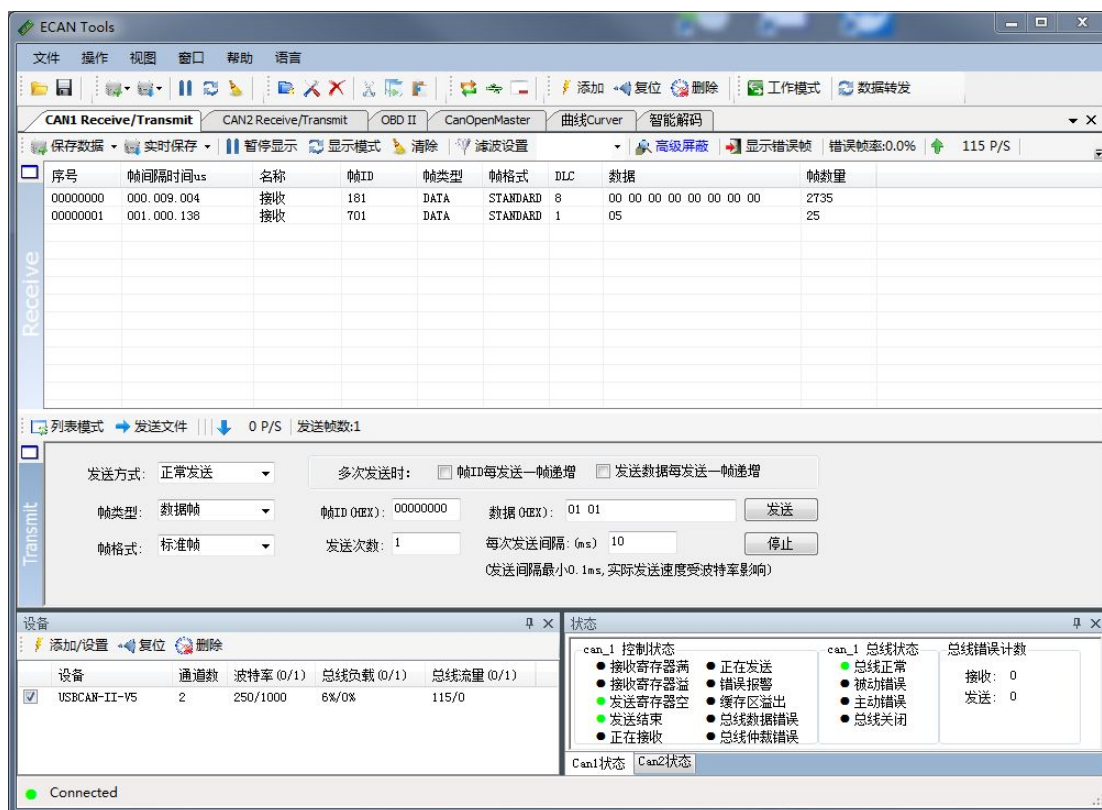
3664: 0~5V 对应 0~65535

3674: 0~10V 对应 0~65535

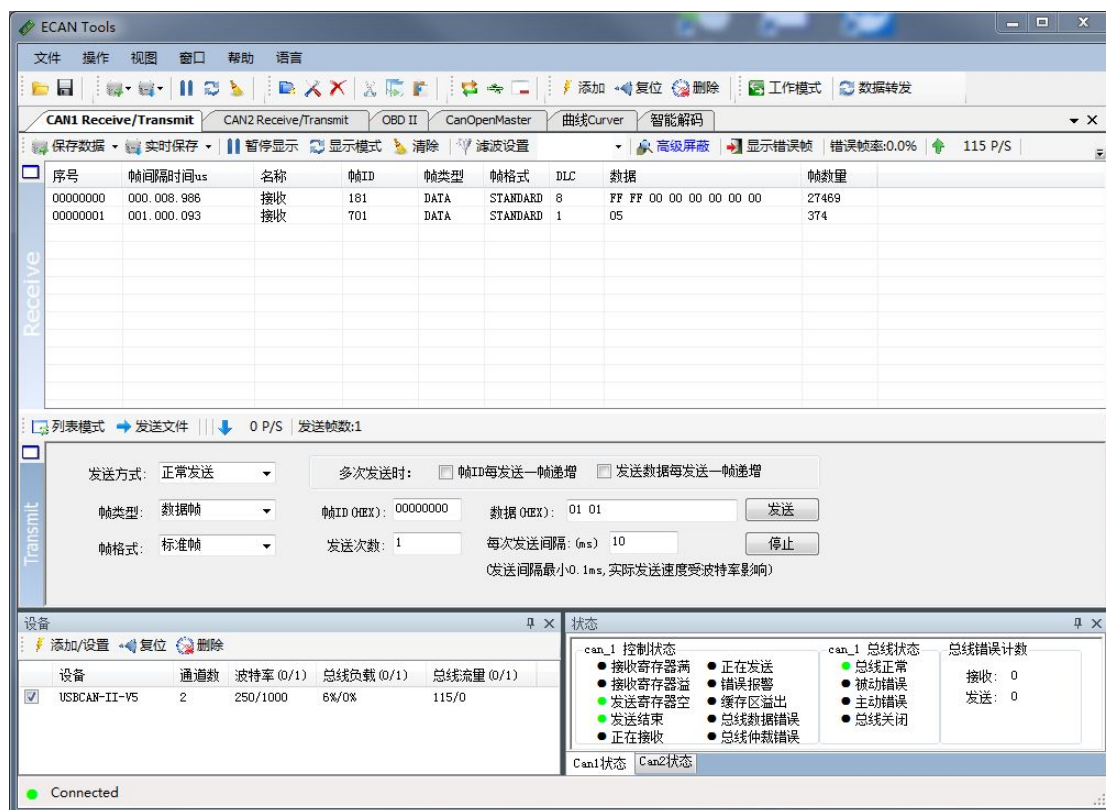
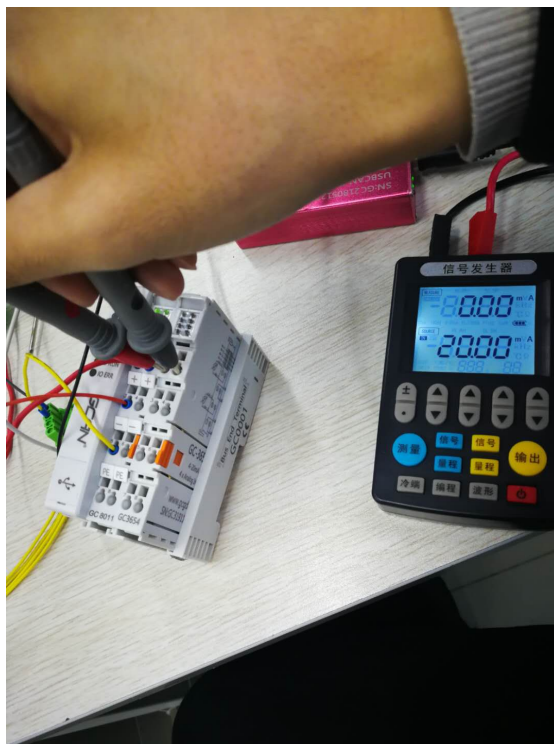
帧 ID	帧数据
0x180+Node ID	1 号 GC-3604 的 4 个通道
0x280+Node ID	2 号 GC-3604 的 4 个通道
0x380+Node ID	3 号 GC-3604 的 4 个通道
0x480+Node ID	4 号 GC-3604 的 4 个通道

下面举例说明

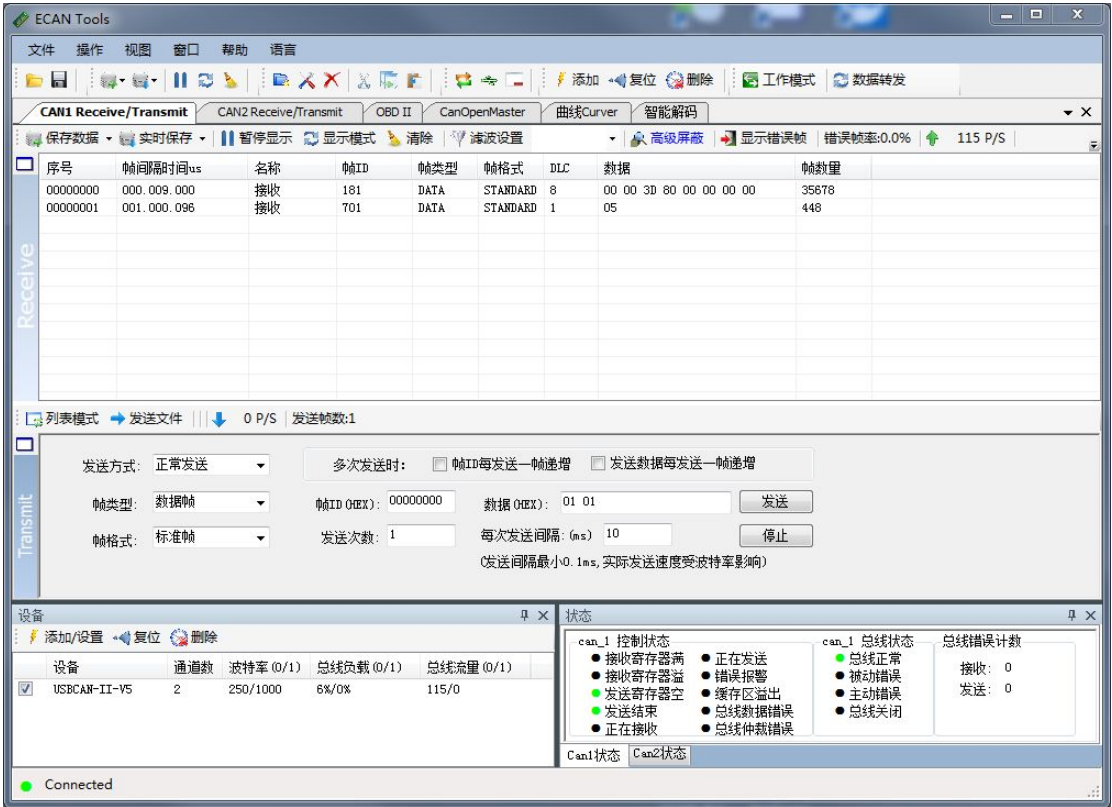
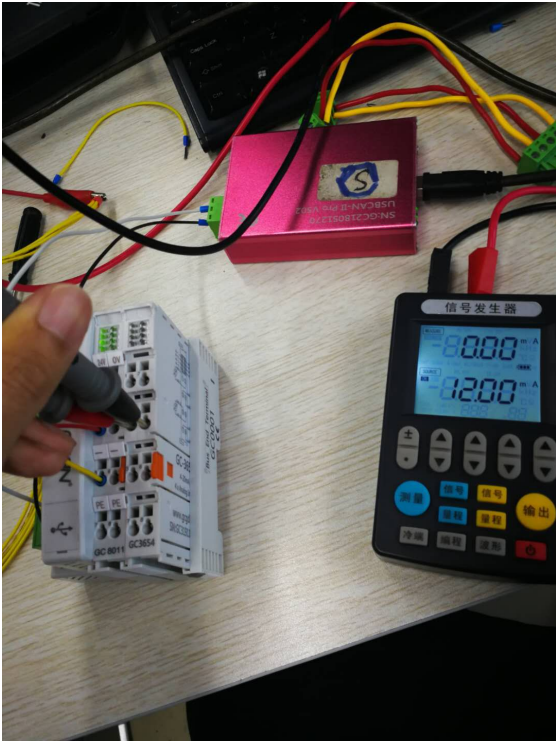
如下图所示，8000 搭载了一个 3654，发送 01 01 启动指令后，会接收到 181 数据。



如下图所示，当给第一通道 20mA 输入时，181 数据发生了变化，前两个字节接收数据变为 FF FF，高字节在前，低字节在后。（3654 的 4-20mA 对应 0-65535）



如下图所示，给第二通道 12mA 输入，3,4 字节显示 3D 80，（正常应该是 FF 7F，但是会有些误差，不影响使用）



其他 36 系列产品使用方法同 3654，只是模拟量与数字量的对应关系不同。

搭载 GC-4602 模块

一个 4602 有两个通道，一个通道两个字节，高字节在后，低字节在前。其他 46 系列产品数据格式同 4602。若搭载多个 4602，则对应关系如下表。

4602: -5V~+5V 对应-32768~+32767

4622: -10V~+10V 对应-32768~+32767

4642: 0~20mA 对应 0~65535

4652: 4~20mA 对应 0~65535

4662: 0~5V 对应 0~65535

4672: 0~10V 对应 0~65535

帧 ID	帧数据
0x200+Node ID	1 号 GC-4602 的 4 个通道
0x300+Node ID	2 号 GC-4602 的 4 个通道
0x400+Node ID	3 号 GC-4602 的 4 个通道
0x500+Node ID	4 号 GC-4602 的 4 个通道

(注: 4642,4652,4662,4672 需在上电前接 24V 与 0V, 5 孔接 24V,6 孔接 0V)

搭载 GC-1502 模块

一个 1502 有俩个通道，一个通道占 8 个字节，其中计数占 4 个字节，频率占 4 个字节，计数在前，频率在后，低字节在前，高字节在后。

帧 ID	帧数据							
0x180+Node ID	GC-1502 的第一个通道							
	计数				频率			
	45 (低位)	8F	8C	03 (高位)	B8 (低位)	0B	00	00 (高位)
0x280+Node ID	GC-1502 的第二个通道							
	计数				频率			
	00 (低位)	20	53	00 (高位)	10 (低位)	27	00	00 (高位)

如上表所示，即通道一的计数为 59543365，频率为 3kHz. 通道二的计数为 5447680，频率为 10kHz.