

# Aeneas R2023T Evaluation Platform 使用手册

产品型号：R2023T

手册版本：V1.00

# 目 录

目 录 .....	2
1 Aeneas R2023T Evaluation Platform概述 .....	3
1.1 平台简介 .....	3
1.2 平台清单 .....	3
1.3 平台技术指标 .....	4
1.3.1 GY7051A USB-I2C Adapter .....	4
1.3.2 R2023T Demo Board .....	4
1.4 Demo Board线路 .....	5
2 接口描述 .....	5
2.1 PIN 脚描述 .....	5
2.2 I/O的直流电气特性 .....	6
3 USB-I2C适配器硬件驱动安装 .....	6
4 I2C Tool 软件安装 .....	9
5 平台硬件操作说明 .....	10
5.1 平台器件连接 .....	10
5.2 Demo Board供电 .....	10
6 平台软件操作说明 .....	11
6.1 软件运行说明 .....	11
6.2 软件功能介绍 .....	11
6.2.1 软件参数描述 .....	11
6.2.2 软件按钮操作 .....	13
7 接口函数与用户二次开发 .....	13
8 USB-I2CAdapter工作原理 .....	14
8.1 I2C 操作过程简述 .....	14
8.2 接收数据（读操作） .....	14
8.3 发送数据（写操作） .....	15
9 R2023T 寄存器设置 .....	15
9.1 时间计数器(0~2h) .....	16
9.2 周计数器(3h) .....	19
9.3 日期计数器(4-6h) .....	19
9.4 振荡调整寄存器(7h) .....	20
9.5 Alarm_W 寄存器(8-Ah) .....	20
9.6 Alarm_W 寄存器(B-Ch) .....	21
9.7 控制寄存器 1(Eh) .....	22
9.8 控制寄存器 2(Fh) .....	23
10 附录 .....	23

# 1 Aeneas R2023T Evaluation Platform概述

## 1.1 平台简介

该平台用于RICOH R2023T 实时时钟芯片的测试，R2023T包括一个外围晶体振荡器、时钟日期计数器和2个中断预警电路，该平台可以测试RTC所有功能及性能指标，比如时钟精度。电池和晶振型号选择等硬件性能测试。控制该设备的软件很容易安装并能设定RTC所有功能，两个分离设备由10Pin connector连接后插入PC USB接口，通过软件即可控制该演示板。

## 1.2 平台清单

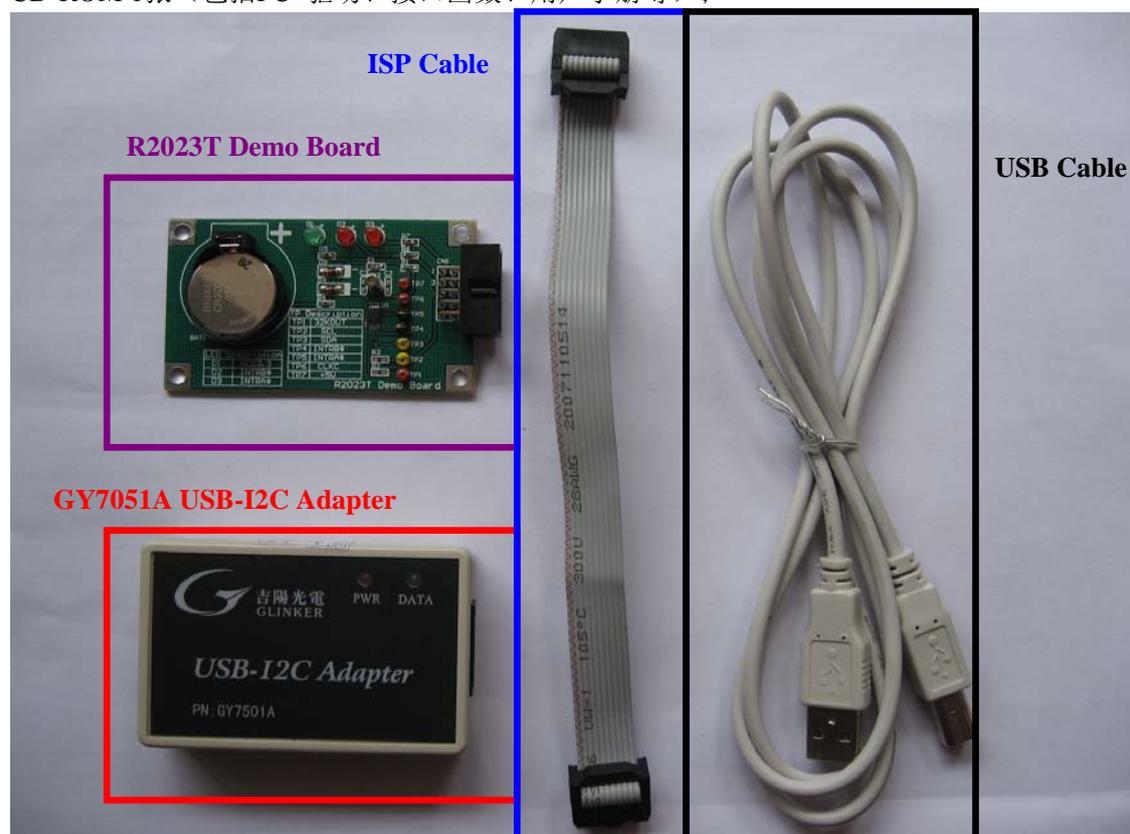
USB 转I2C 适配器1台；

R2023T Demo Board 1PCS

USB 连接线1根；

ISP 扁平接口线缆1根

CD-ROM 1张（包括PC 驱动、接口函数、用户手册等）；



## 1.3 平台技术指标

### 1.3.1 GY7051A USB-I2C Adapter

- USB 2.0转I2C接口适配器，USB总线供电，无需外部电源；
- I2C主机接口，Master 方式，兼容SMbus协议；
- 1通道独立的I2C 接口；
- 硬件I2C 控制器，标准的I2C，非IO 模拟产生；
- 提供电源输出：+3.3V, +5V；
- 接口信号:SCL,SDA,GND,+5V,+3.3V 以及4 路IO 口；
- 输出信号3.3V TTL，输入5VTTL 可承受。
- I2C 总线速率可由软件设置，支持1k—800kHz。
- 支持一体化傻瓜式读写模式 (Easy I2C)，以及分步I2C 时序控制读写模式 (Timing I2C)。
- Easy I2C 模式读写长度最大512 字节，Timing I2C 模式则可读写任意长度。
- 支持通过调用VCI\_GYI2C 的DLL 库函数，进行二次开发；
- 提供Visual c++开发例程；
- 提供I2C 工具软件I2CTools；
- 尺寸：70\*45\*18mm；
- 工作温度：-40°C - +85°C
- 最大特点：用户完全不需要了解复杂的I2C 协议，时序。

### 1.3.2 R2023T Demo Board

- 主电源和备用电池供电，保证寄存器数据持续有效；
- 耗流1.7uA(测试条件:  $V_{DD}=3V$ , SCL=SDA=CLKC=High, 32KOUT=ON, OUTPUT=Colse, CGout=CDout=0pF)，寿命可以长达14年；
- 2 Channel 中断信号INTRA#、INTRB#；
- 6种中断频率设置；
- 3个状态指示LED及12个测试点，方便User随时了解RTC工作状况；
- 3个Jumper设置方便User测试Demo Board各电源部分耗流；
- I2C 总线速率可由软件设置，最大支持400kHz；
- 到2099年并能自动识别闰年；
- RTC register是16字节，Easy I2C 模式读写长度最大512 字节，可以满足要求；
- 尺寸：80\*40\*9mm；
- 工作温度：-40°C - +85°C

## 1.4 Demo Board线路

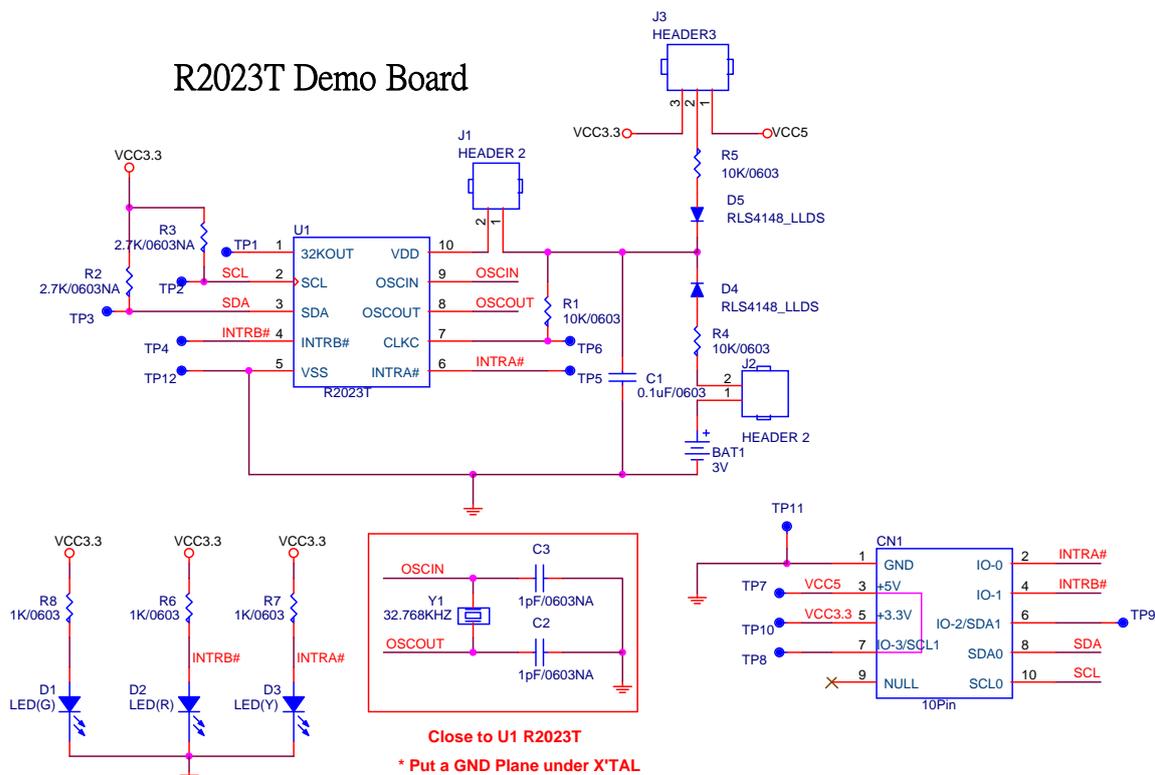


图1 R2023T Demo Board线路

## 2 接口描述

### 2.1 PIN 脚描述

适配器直接的输出接口为10pin 的针式接口，通过10 芯的直连排线延长以后，则得到孔式接口，如下图：

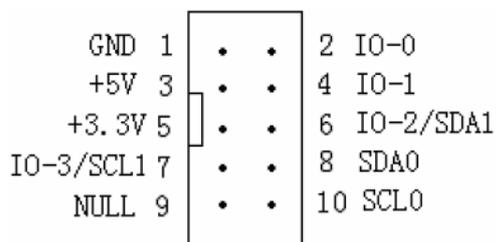


图2 适配器对外接口（针式）

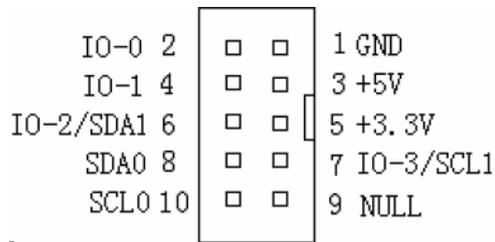


图3 适配器对外排线接口

Pin No	Symbol	描述
Pin1	GND	电源地与信号地
Pin2	I/O-0	I/O 口。默认值为高电平。输出高 3.3V，输入 5V 可承受
Pin3	+5V	电源输出，驱动电流 max 200mA
Pin4	IO-1	I/O 口。默认值为高电平。输出高 3.3V，输入 5V 可承受

Pin5	+3.3V	电源输出, Normal 3.0-3.6V, 驱动电流 max 60mA 注1
Pin6	IO-2/SDA1	I/O 口, 或 SDA1 注2
Pin7	IO-3/SCL1	I/O 口, 或 SCL1 注2
Pin8	SDA0	内部 2.7K 电阻已上拉
Pin9	Null	空脚
Pin10	SCL0	内部 2.7K 电阻已上拉

注1: 如果从机的电路上已有供电, 则请不要使用适配器提供的电源。因适配器提供电源信号的驱动电流比较小, 请用户使用使谨慎评估, 以防PC 的USB 接口损坏。

注2: GY7501A的该信号只能被用于IO 口, 不能用作I<sup>2</sup>C Bus。

## 2.2 I/O的直流电气特性

VDD=2.7-3.3V, -40°C- +85°C(除非特别说明)

参数	条件	最小值	典型值	最大值	单位
输出高电压(V <sub>OH</sub> )	I <sub>OH</sub> =-10uA	VDD-0.1			V
	I <sub>OH</sub> =-3mA	VDD-0.7			V
	I <sub>OH</sub> =-10mA		VDD-0.8		V
输出低电压(V <sub>OL</sub> )	I <sub>OL</sub> =10uA			0.1	V
	I <sub>OL</sub> =8.5mA			0.6	V
	I <sub>OL</sub> =25mA		1.0		V
输入高电压(V <sub>OH</sub> )		2.0			V
输入低电压(V <sub>OL</sub> )				0.8	V

注: USB-I2C 适配器中, I/O输出采用的是推挽输出。

安装 RTC 控制软件和 USB 驱动

## 3 USB-I2C适配器硬件驱动安装

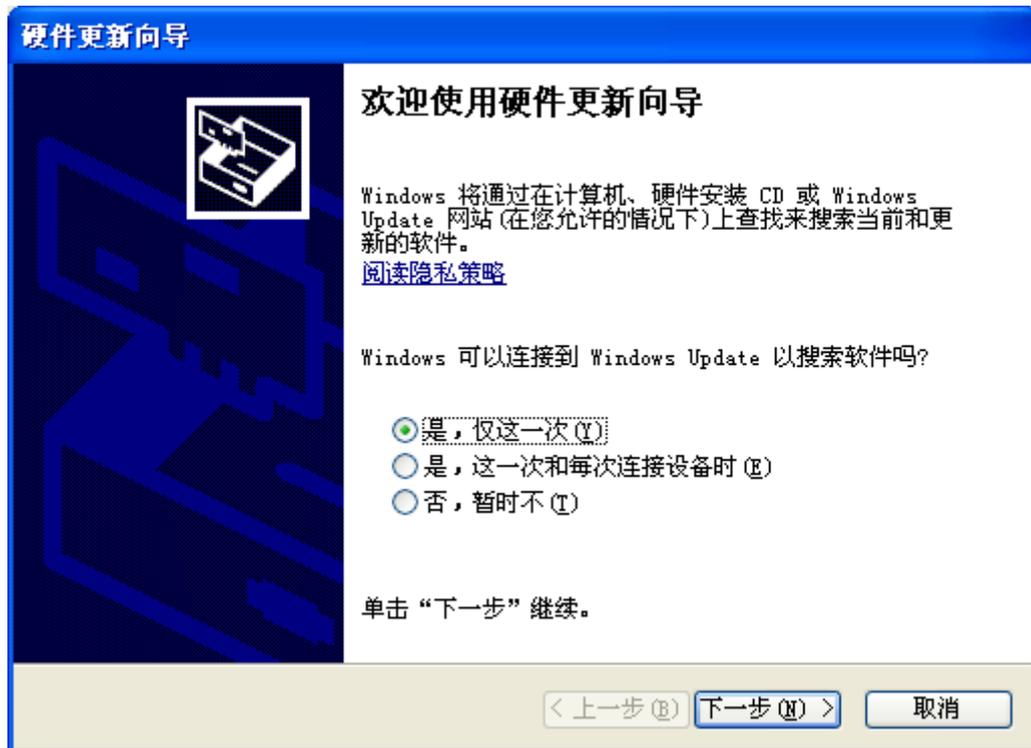
1、连接好测试平台:



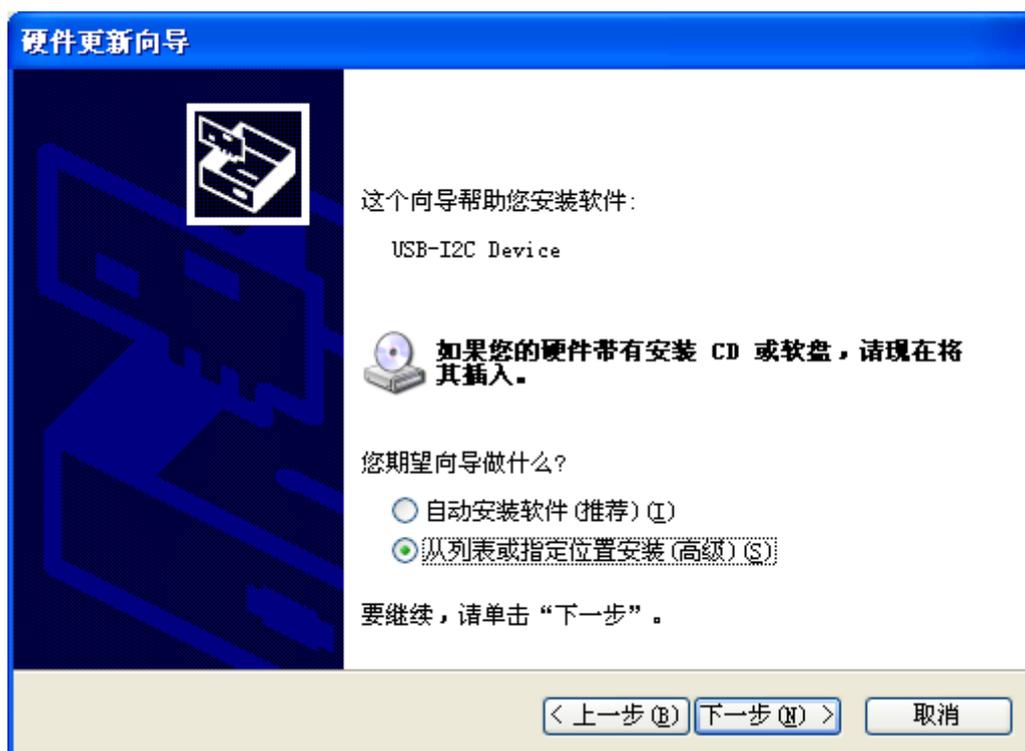
2、插入吉阳光电提供的驱动光盘，

3、将USB-I2C适配器接入PC 或笔记本电脑的USB 接口，此时USB-I2C Adapter的PWR & DATA LED都会被点亮，根据提示安装我们提供的驱动程序。

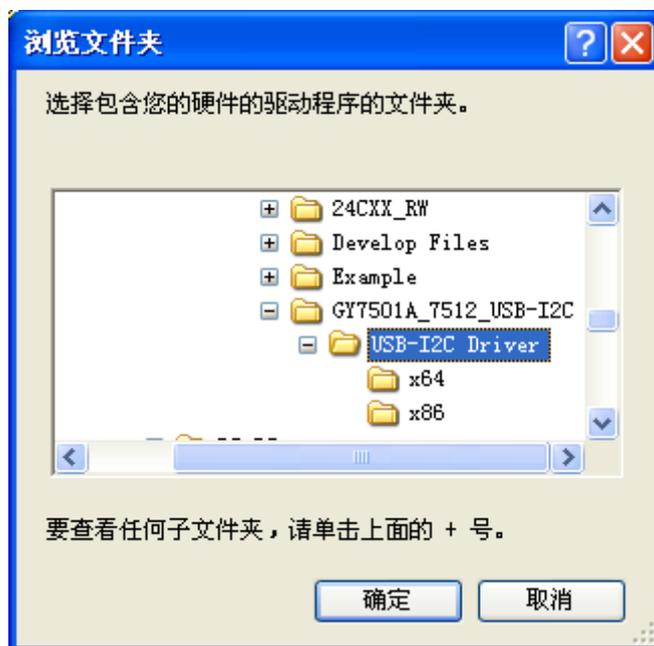
首次插入适配器，系统会自动检测到新硬件，并弹出如下对话框，选择‘是，仅这一次’点击下一步：



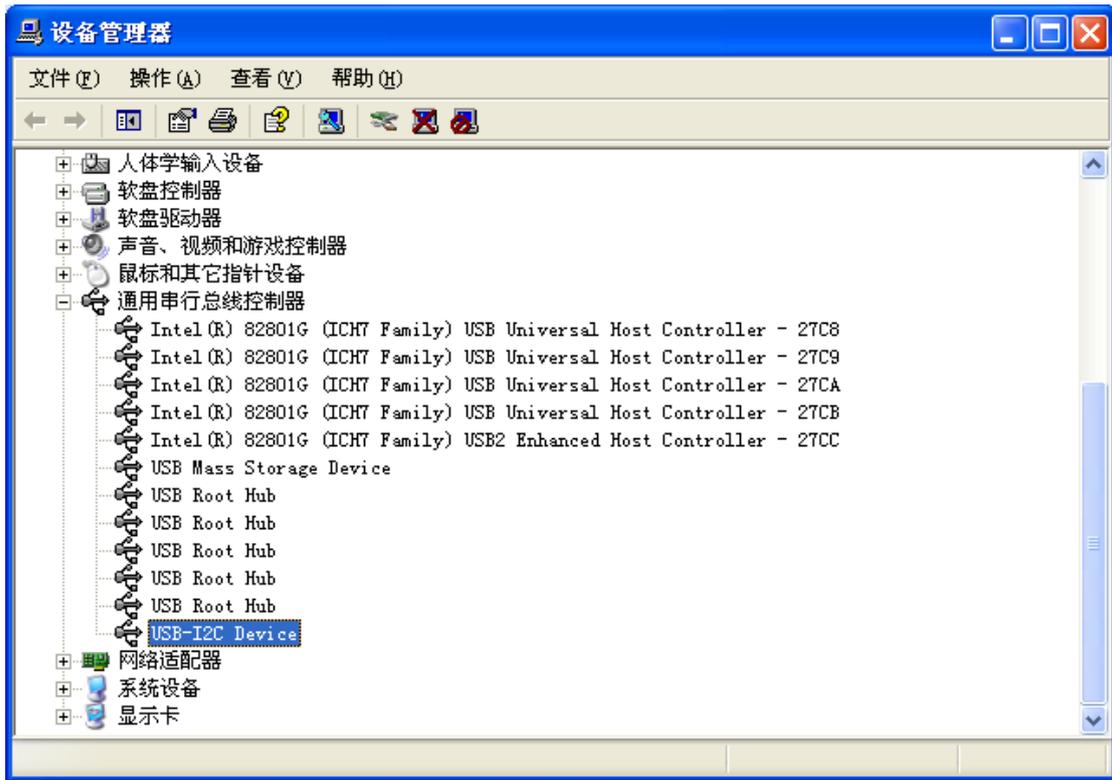
选择从列表中安装，以缩短系统搜寻驱动时间。点击下一步：



如下图，选择驱动程序所在的目录进行安装(X86是32位系统USB驱动，X64是64位系统USB驱动，如果不确定装哪个驱动，可以只选择下图中USB Driver即可，安装程序会自动识别并安装正确对应的驱动)，过程中会提示该驱动程序是否确认安装，点确认认可。



出现搜寻驱动界面，稍等系统会开始安装驱动：  
传输画面结束后，点击完成即可。至此适配器驱动正确安装完毕。安装完成后，设备管理器中会指示有“USB-I2C Device”如下图蓝色部分

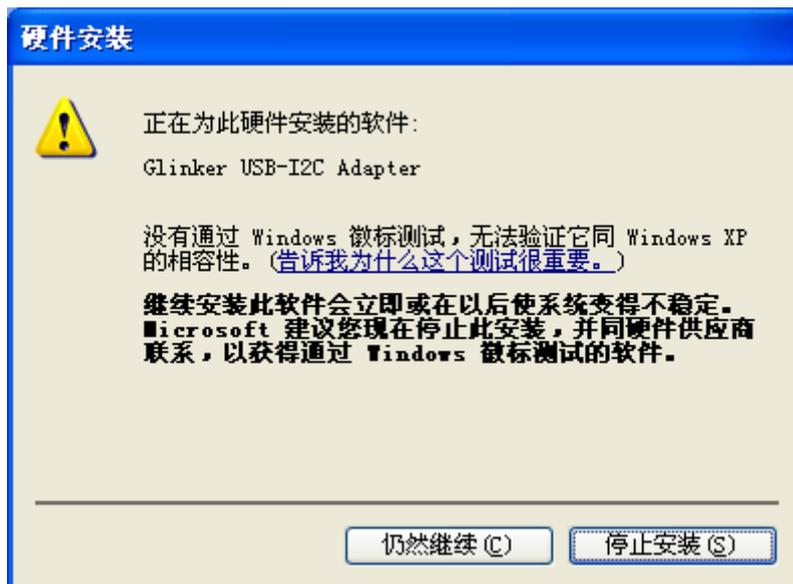


## 4 I2C Tool 软件安装

1、 双击光盘中的Tool安装程序图标



按照提示设定安装目录并点击NEXT，安装过程中会出现以下对话框，点击‘仍然继续’即可。



最后点击 Finish。自此完成安装，桌面上会显示快捷图标

## 5 平台硬件操作说明

### 5.1 平台器件连接

此平台主要由两部分组成，一部分是武汉吉阳光电的USB to I2C 适配器，另一部分是我们Aenes的RTC Demo Board，

1、两部分通过10Pin排线连接。

2、USB Cable 一端插入 PC USB 接口，另一端插入吉阳光电的 USB to I2C 适配器一旦接上，Demo Board 上 LED1 和 Adapter 上的 PWR & DATA LED 都将被点亮，表明 USB 接口的+5V Power 已经输入到设备。如下图示：



3、使用时注意开启及关闭顺序，开始使用时，先将连接 OK 的测试平台通过插入 USB 接口，然后在运行 I2C Tool 软件，退出使用时，先关闭 I2C Tool 软件，然后载移除测试平台。如果不按此操作，将会在使用时弹出错误对话框并导致 device 打开失败。

### 5.2 Demo Board供电

R2023T Demo Board 包含一颗 3V 的 R2032 纽扣电池，可以作为后备电源给 RTC 供电，当主电源 VCC3.3 Power Down 后，如果 User 想完全使 Demo Board 掉电，可以把 Jumper2 断开以达到目的，在 RTC 停止工作之前，位于 VDD Pin 上的 Bypass 电容必须完全放电，如果 VDD 接到+5V，放电时间可能长达 1Second，注意当电源完全移除时，Demo Board 不能受软件控制。



- 1) Product: 供 User 在下拉菜单中选用对应型号产品, 我们选择 GY7501A;
- 2) ComPort: 当 User 插入设备时在运行软件, 会自动识别 COM Port, 如上图 COM1;
- 3) BaudRate: 波特率, 软件自动根据产品选择速率, 这里是 115200;

## 2、Setting 属性框

- 1) Work Mode: I2C 的工作模式, 在我们的产品软件中, 提供了2 种模式, 分别是Easy I2C 模式和TimingI2C 模式。每次上电后默认的工作模式为**Easy I2C** (一体化I2C 模式), 建议一般用户采用该方式。

### **Easy I2C** (一体化I2C 读写模式):

这是我们提供的一种傻瓜式的I2C 操作模式。

可直接通过命令或调用一次函数, 进行读写数据, 无须考虑去产生I2C 的时序。

工作过程: 转换器/模块得到该命令以后, 进行解析, 然后启动内部的I2C 读写控制时序, 将上位机要求的操作完成以后, 再将结果返回给上位机。

优点: 该方式简单方便, 快速, 推荐使用。用户不需要了解I2C 时序协议。

I2C 时钟频率从1k-800khz 可设置。我们的R2023T的最大频率是400kHZ, 所以请不要超过该值, 否则I2C读写Fail。

局限性: 受内部缓冲区的限制, 一次命令最多读出来的数据为512个, 一次最多写入的数据为520 个 (包含命令字)。

### **Timing I2C** (分步控制I2C的时序模式):

I2C 时序由上位机软件或命令来控制, 分如下4 种命令。

I 产生I2C 启动时序状态。

II 写入8 个bit, 即一个字节, 之后获取并返回ACK 状态

III 读出8 个bit, 即一个字节, 之后给出ACK 或NACK

IV 产生I2C 停止时序状态

优点: 上位机用户自行控制I2C 的时序, 时序完全透明开放。可读写的长度不受限制, 由用户控制。

局限性: 用户需要熟悉I2C 时序才能使用该方式。来回通信握手判断, 对速度有影响。I2C 时钟频率可设置的范围: 1k-235kHZ

- 2) I2C Channel: I2C 的通道索引号选择。GY7501 只有索引号为0 的I2C 接口。默认选择 I2C 通道0

- 3) 参数I2C Clock: 时钟频率, 也就是I2C 通信的速率。默认I2C 速率100khz

## 3、I2C Read/Write属性框

- 1) Slave Address: I2C 从设备 (被测设备) 的物理地址。该地址一般由7 位地址+R/W 位组成。请将这7 位实际地址左移一位填入。例如: 我们的R2023T地址是0x32, 二进制为0110010, 则将其左移一位得01100100, 十六进制表示为0x64, 所以请将64填入该编辑框中。

- 2) ROM/REG Address: 请填入希望读写的ROM 或者寄存器地址。如果连续多字节读写, 则填入的是起始地址。(16 进制表示)。该参数可以不选择。例如依次填入00、10、20、、、A0、B0、、、F0, Read Request设置为1, 那么可以依次读取每个寄存器的值。

- 3) 2-Bytes Addr: 如果设备ROM或REG的地址是2个字节, 则请将此打勾。这种具有2个字节地址特性的一般是存储容量较大的器件。

- 7) Write Data Num: 需要写入的数据的个数, 这个值为编辑框内有效的数据个数。请将这些数据填入Data buffer 编辑框中。如果是连续写2个字节, 则该参数填2, 同时Data buffer 中最少要填入2个字节的数据。然后, 执行写操作。

8) Read Request: 填入需要连续读的个数, 最多填入256。如果读一个字节, 则填入1。

## 6.2.2 软件按钮操作

1) Open: 将设备打开, 并配置选择的参数信息。如下图所示, 正确连接设备后, 运行软件, 选择GY7501A, 点击Open, 则Open按钮变灰色无效, History Operated Logs框中显示打开设备成功的信息并显示工作模式。

2) Close: 将设备关闭, 点击后变灰色无效, 此时将不能操作设备。

3) Read: 将根据“Slave Address”, “ROM/REG Address”(如果使能), “Required Read Num”这三个信息进行I2C 总线的读操作: 成功后则依次将数据返回到Data buffer 编辑框中。

4) Write: 将根据“Slave Address”, “ROM/REG Address”, “Write Data Num”以及缓冲区中的数据内容, 进行I2C 写操作。完成后会有状态返回。

注: 软件中的完成提示, 并不意味着这些数据已经真正的被写入到从设备中了。仅表示这些内容已经按照I2C 的协议时序发出。一般如果您的从设备是稳定的支持I2C 协议, 则都可以被正确写入。您可以通过读操作将数据读出来判断。

5) Data buffer 编辑框是一个Hex 和ASCII 码能同时编辑显示的输入控件。内容为将要写入或者读取到的数据。

6) Load File: 只能导入bin 格式的二进制码文件, 内容将依次存入缓冲区。

7) Save As: 将依次存入缓冲区的数据存入到文件中, 文件类型bin 格式。

8) History Operated Logs: 记录了用户的操作过程和状态, 以方便用户查看。用户可以将其复制到文件中(选中并点右键copy 即可)。

注: 该编辑框有容量限制, 用户应及时清空, 否则可能会没有新内容进来。

## 7 接口函数与用户二次开发

一般的用户直接使用我们的I2CTools 软件即可, 对接口函数和二次开发方面不需要关注。

如果用户觉得我们提供的I2CTools 软件不是很方便使用, 则可以自己编写用户软件, 利用我们提供的函数库进行二次开发, 光盘中提供了例程可供参考。

开发文件共4 个: VCI\_GYI2C.DLL, VCI\_GYI2C.LIB, VCI\_GYI2C.H, SiUSBxp.DLL。请将这4 个文件拷贝到您的Visual C++的工程目录。如果用户使用其他开发环境如VB, C++Builder 等也可以, 请参考相关资料。关于VCI\_GYI2C 库函数的调用以及内部函数的详细说明, 请参考吉阳光电公司提供的专门文档。

VCI\_GYI2C 的支持型号:GY7501/GY7512/GY7505/GY7506/GY760X。

## 8 USB-I2CAdapter工作原理

### 8.1 I2C 操作过程简述

I2C 操作有两种可能的数据传输类型：从主发送器到所寻址的从接收器（写）和从被寻址的从发送器到主接收器（读）。这两种数据传输都由主器件启动，主器件还在SCL 上提供串行时钟。SMBus接口可以工作在主方式或从方式，总线上可以有多个主器件。如果两个或多个主器件同时启动数据传输，仲裁机制将保证有一个主器件会赢得总线。注意：没有必要在一个系统中指定某个器件作为主器件；任何一个发送起始条件（START）和从器件地址的器件就成为该次数据传输的主器件。

一次典型的SMBus 数据传输包括一个起始条件（START）、一个地址字节（位7-1：7 位从地址；位0：R/W 方向位）、一个或多个字节的数据和一个停止条件（STOP）。每个接收的字节（由一个主器件或从器件）都必须用SCL 高电平期间的SDA 低电平（见图16.3）来确认（ACK）。如果接收器件不确认，则发送器件将读到一个“非确认”（NACK），这用SCL 高电平期间的SDA 高电平表示。

方向位（R/W）占据地址字节的最低位。方向位被设置为逻辑1 表示这是一个“读”（READ）操作，方向位为逻辑0 表示这是一个“写”（WRITE）操作。

所有的数据传输都由主器件启动，可以寻址一个或多个目标从器件。主器件产生一个起始条件，然后发送地址和方向位。如果本次数据传输是一个从主器件到从器件的写操作，则主器件每发送一个数据字节后等待来自从器件的确认。如果是一个读操作，则由从器件发送数据并等待主器件的确认。在数据传输结束时，主器件产生一个停止条件，结束数据交换并释放总线。

### 8.2 接收数据（读操作）

USB-I2C Adapter 此时工作在主接收方式。Adapter 在SDA 上接收串行数据，在SCL 上输出串行时钟。Aapter I2C 接口首先产生一个起始条件，然后发送含有目标从器件地址和数据方向位的第一个字节。在这种情况下数据方向位（R/W）应为逻辑‘1’，表示这是一个“读”操作。接着从SDA接收来自从器件的串行数据并在SCL 上输出串行时钟。从器件发送一个或多个字节的串行数据。每收到一个字节后，写ACK 位，以定义要发出的确认值。Adapter 应在接收到最后一个字节后向ACK位写‘0’，以发送NACK。最后主机发出一个停止条件后结束主接收器方式。

下图给出了典型的主接收器时序，只给出了接收两个字节的传输时序，尽管可以接收任意多个字节。

USB-I2C Adapter I2C Read Timing



 从机发送，主机接收

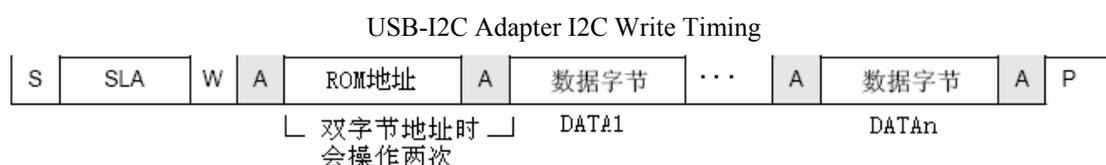
 主机发送，从机接收

S = START (开始)  
P = STOP (停止)  
A = ACK (确认)  
N = NACK (非确认)  
R = READ (读)  
SLA = 从地址

### 8.3 发送数据（写操作）

USB-I2C Adapter 此时工作在主发送方式。Adapter 在SDA 上发送串行数据，在SCL 上输出串行时钟。Adapter 的I2C 接口首先产生一个起始条件，然后发送含有目标从器件地址和数据方向位的第一个字节。在主发送器方式数据方向位（R/W）应为逻辑‘0’，表示这是一个“写”操作。主发送器接着发送一个或多个字节的串行数据。在每发送一个字节后，从器件发出确认位。当STO 位被置‘1’并产生一个停止条件后，串行传输结束。

下图给出了典型的主发送器时序，发送任意多个字节。



 从机发送，主机接收

 主机发送，从机接收

S = START (开始)  
P = STOP (停止)  
A = ACK (确认)  
W = WRITE (写)  
SLA = 从地址

## 9 R2023T 寄存器设置

Address Mapping, 16 字节寄存器(0-Fh), 地址用 4 个 bit 表示, 数据用 8 个 bit 表示

	Address				Register Name	Data							
	A3	A2	A1	A0		D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	Second Counter	- *2)	S40	S20	S10	S8	S4	S2	S1
1	0	0	0	1	Minute Counter	-	M40	M20	M10	M8	M4	M2	M1
2	0	0	1	0	Hour Counter	-	-	H20	H10	H8	H4	H2	H1
3	0	0	1	1	Day-of-week Counter	-	-	P/ $\bar{A}$	-	-	W4	W2	W1
4	0	1	0	0	Day-of-month Counter	-	-	D20	D10	D8	D4	D2	D1
5	0	1	0	1	Month Counter and Century Bit	19 /20	-	-	MO10	MO8	MO4	MO2	MO1
6	0	1	1	0	Year Counter	Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1
7	0	1	1	1	Oscillation Adjustment Register *3)	DEV *4)	F6	F5	F4	F3	F2	F1	F0
8	1	0	0	0	Alarm_W (Minute Register)	-	WM40	WM20	WM10	WM8	WM4	WM2	WM1
9	1	0	0	1	Alarm_W (Hour Register)	-	-	WH20	WH10	WH8	WH4	WH2	WH1
A	1	0	1	0	Alarm_W (Day-of-week Register)	-	WW6	WW5	WW4	WW3	WW2	WW1	WW0
B	1	0	1	1	Alarm_D (Minute Register)	-	DM40	DM20	DM10	DM8	DM4	DM2	DM1
C	1	1	0	0	Alarm_D (Hour Register)	-	-	DH20	DH10	DH8	DH4	DH2	DH1
D	1	1	0	1				DP/ $\bar{A}$					
E	1	1	1	0	Control Register 1 *3)	WALE	DALE	12 /24	CLEN2	TEST	CT2	CT1	CT0
F	1	1	1	1	Control Register 2 *3)	VDSL	VDET	XST	PON *5)	CLEN1	CTFG	WAFG	DAFG

## 9.1 时间计数器(0~2h)

Second Counter (Address 0h)

D7	D6	D5	D4	D3	D2	D1	D0
-	S40	S20	S10	S8	S4	S2	S1
0	S40	S20	S10	S8	S4	S2	S1
0	Indefinite						

(For Writing)  
(For Reading)  
Default Settings \*)

设置 D7—D0: 00000000—01011001(二进制)=00—59(BCD 码), 以上数据对应 0 秒到 59 秒, 59 秒后秒计数器从 0 开始继续循环, 分钟则进一位。例如, 写 58 秒到 RTC 寄存器:

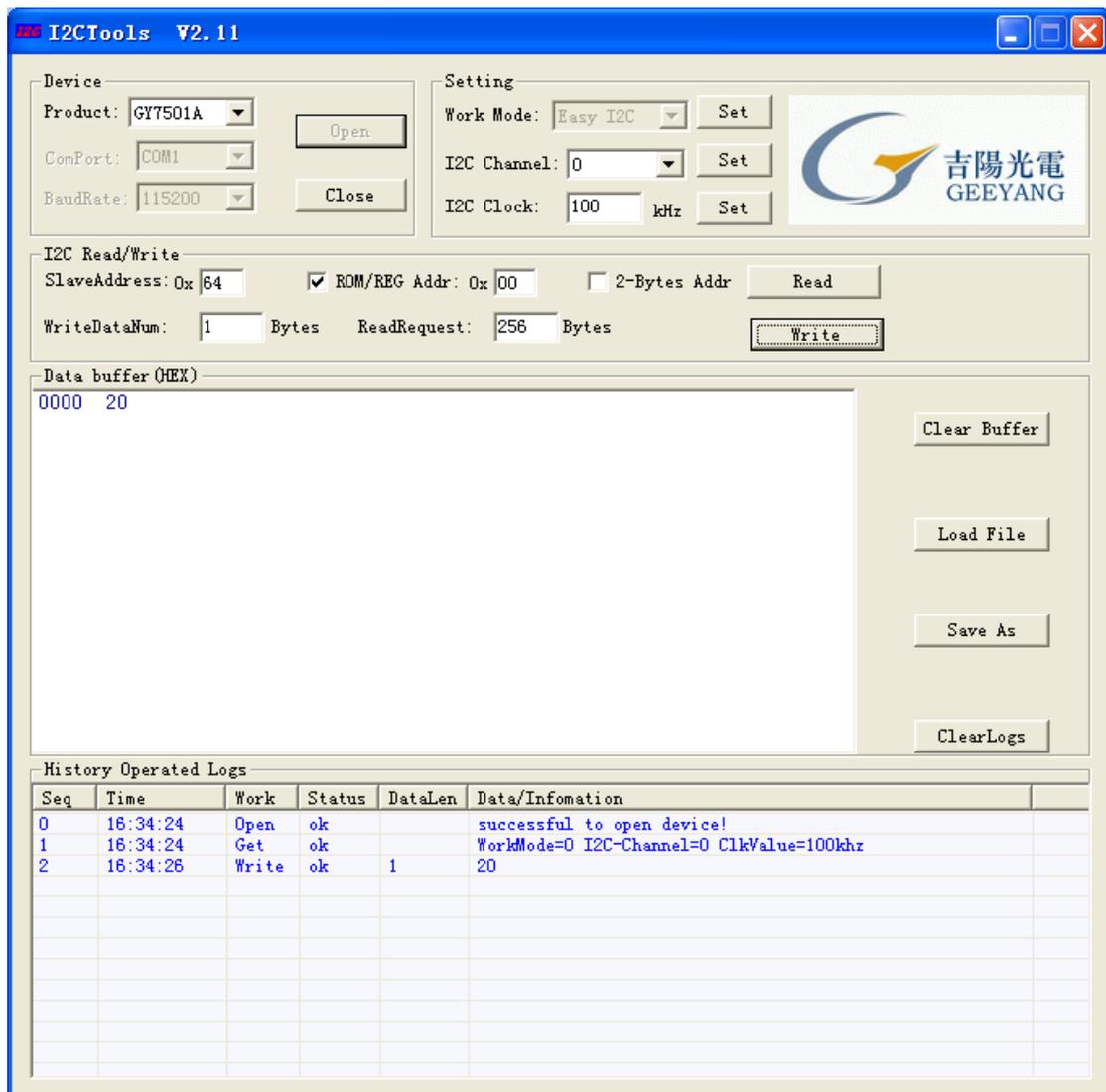
Slave Address: 0x64,前面软件参数描述中已经说明;

ROM/REG Addr: 00 代表时间计数器 0h;

Write Data Num: 1, 表示写进一位;

Read Request: 1, 表示读取一位;

在 Data Buffer 中输入 20, 然后点击 Write, 下面 History Operation Logs 将显示写 20 成功。

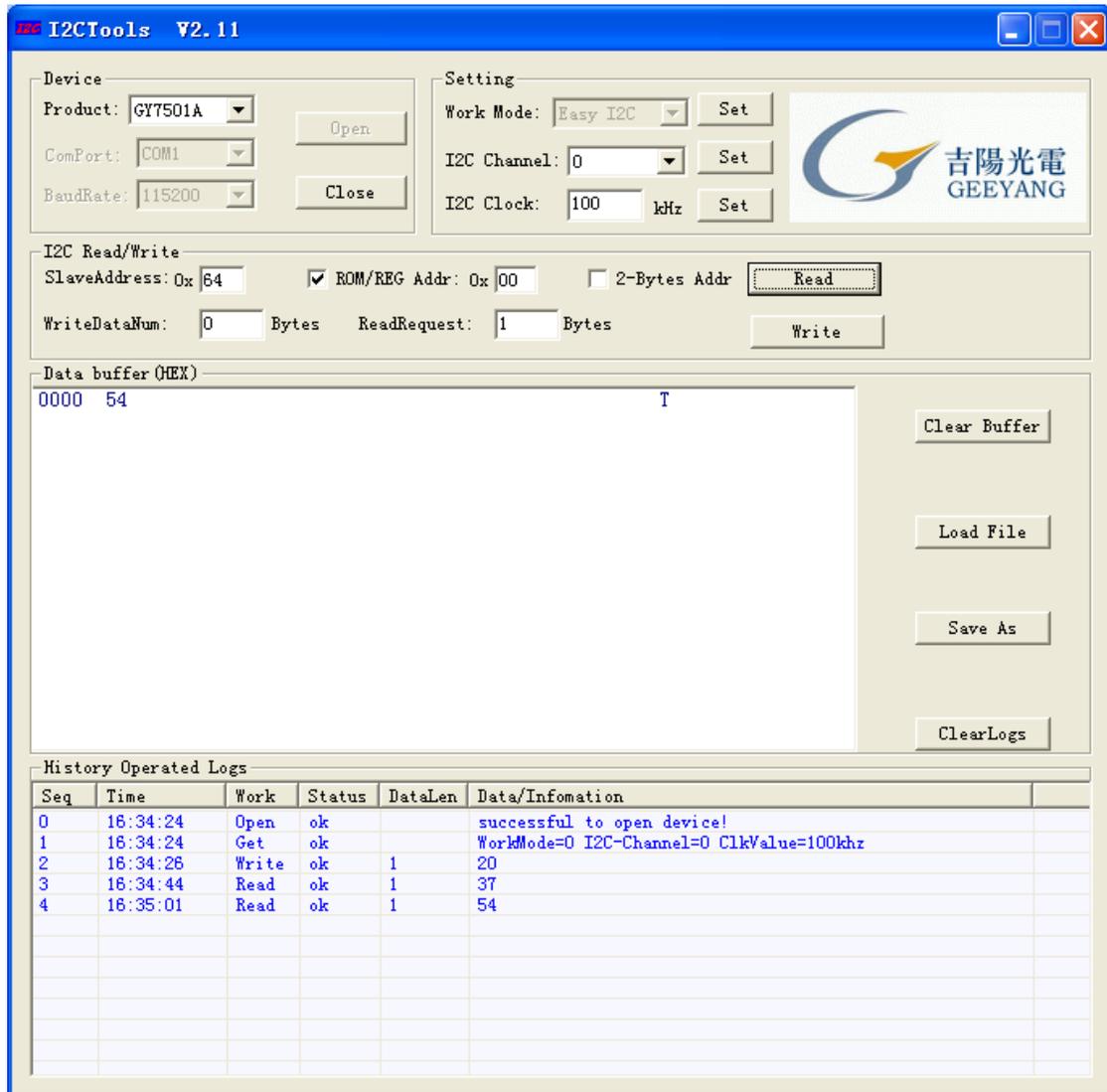


读操作:

Write Data Num: 0, 表示不执行写操作

Read Request: 1, 表示读取一位;

然后点击 Read, 下面 History Operation Logs 将显示秒数。因为操作时间关系, 系统时间和 RTC 时间间隔一致, 说明读操作成功。



### Minute Counter (Address 1h)

D7	D6	D5	D4	D3	D2	D1	D0
-	M40	M20	M10	M8	M4	M2	M1
0	M40	M20	M10	M8	M4	M2	M1
0	Indefi nite						

与秒计数器的操作一样。

### Hour Counter (Address 2h)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	P/ $\bar{A}$ or H20	H10	H8	H4	H2	H1
0	0	P/ $\bar{A}$ or H20	H10	H8	H4	H2	H1
0	0	Indefi nite	Indefi nite	Indefi nite	Indefi nite	Indefi nite	Indefi nite

寄存器设置如下：

24H 模式	12H 模式	24H 模式	12H 模式
D7—D0	D7—D0	D7—D0	D7—D0
00	12(AM12)	12	32(PM12)
01	01(AM1)	13	21(PM1)
02	02(AM2)	14	22(PM2)
03	03(AM3)	15	23(PM3)
04	04(AM4)	16	24(PM4)
05	05(AM5)	17	25(PM5)
06	06(AM6)	18	26(PM6)
07	07(AM7)	19	27(PM7)
08	08(AM8)	20	28(PM8)
09	09(AM9)	21	29(PM9)
10	10(AM10)	22	30(PM10)
11	11(AM11)	23	31(PM11)

## 9.2 周计数器(3h)

Day-of-week Counter (3h)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	-	-	-	W4	W2	W1
0	0	0	0	0	W4	W2	W1
0	0	0	0	0	Indefinite	Indefinite	Indefinite

1) (D2,D1,D0)=W4, W2, W1) = (0, 0, 0) → (0, 0, 1) → ... → (1, 1, 0) → (0, 0, 0) = 周日 → 周一 → ... → 周六 → 周日 7 天一个轮回。

(1,1,1)只有在不用该计数器功能时才可以被写入，否则禁止写入

## 9.3 日期计数器(4-6h)

Day-of-month Counter (Address 4h)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	D20	D10	D8	D4	D2	D1
0	0	D20	D10	D8	D4	D2	D1
0	0	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite

D7—D0:在 1、3、5、7、8、10、12 月份时可以设置为 01~31

在 4、6、9、11 月份时可以设置为 01~30

在闰年 2 月份可以设置为 01~29，平年 2 月份可以设置为 01~28

### Month Counter + Century Bit (Address 5h)

D7	D6	D5	D4	D3	D2	D1	D0
19 /20	-	-	MO10	MO8	MO4	MO2	MO1
19 /20	0	0	MO10	MO8	MO4	MO2	MO1
Indefinite	0	0	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite

D6~D0: 可接受 01~12

19/20:D7=1 时 20xx 年, D7=0 时 19xx 年

### Year Counter (Address 6h)

D7	D6	D5	D4	D3	D2	D1	D0
Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1
Y80	Y40	Y20	Y10	Y8	Y4	Y2	Y1
Indefinite							

D7~D0: 可接受 00~99

## 9.4 振荡调整寄存器(7h)

D7	D6	D5	D4	D3	D2	D1	D0
DEV	F6	F5	F4	F3	F2	F1	F0
DEV	F6	F5	F4	F3	F2	F1	F0
0	0	0	0	0	0	0	0

(For Writing)  
(For Reading)  
Default Settings \*)

DEV:调整比率设置, DEV=0, 在 00、20、40S 时调整; DEV=1, 在 00S 时调整

F6-F0:1、当写入寄存器的时间与震荡调整电路工作时间(00、20、40S)一致时, 它将不工作

2、F6=0 时, 时间增量((F5, F4, F3, F2, F1, F0) - 1) × 2.

F6=1 时, 时间减量((F5, F4, F3, F2, F1, F0) + 1) × 2.

(F6, F5, F4, F3, F2, F1)=(0,0,0,0,0,\*)时, 当前时间不变(\*表示任意 0 或 1)

例如:

(DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 0, 0, 0, 1, 1, 1), 在 00, 20, 40S 时,  $32768 + (7-1) \times 2 = 32780$  (a current time count loss).

(DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 0, 0, 0, 0, 0, 1), 即使在 00, 20, 40S 时, 32768 不变.

(DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 1, 1, 1, 1, 0), 在 00S 时,  $32768 - (1+1) \times 2 = 32764$  (a current time count gain).

## 9.5 Alarm\_W 寄存器(8-Ah)

### Alarm\_W Minute Register (Address 8h)

D7	D6	D5	D4	D3	D2	D1	D0
-	WM40	WM20	WM10	WM8	WM4	WM2	WM1
0	WM40	WM20	WM10	WM8	WM4	WM2	WM1
0	Indefinite						

D7—D0: 0000 0000—0101 1001=00—59 与地址为 1h 的分钟计数器设置方法一样。

### Alarm\_W Hour Register (Address 9h)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	WH20 WP/ $\bar{A}$	WH10	WH8	WH4	WH2	WH1
0	0	WH20 WP/ $\bar{A}$	WH10	WH8	WH4	WH2	WH1
0	0	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite	Indefinite

与地址为 2h 的小时计数器设置方法一致。

### Alarm\_W Day-of-week Register (Address Ah)

D7	D6	D5	D4	D3	D2	D1	D0
-	WW6	WW5	WW4	WW3	WW2	WW1	WW0
0	WW6	WW5	WW4	WW3	WW2	WW1	WW0
0	Indefinite						

D6—D0: 周六 周五 周四 周三 周二 周一 周日

对应 Bit 为 1 时 Alarm Enable; 为 0 时 Alarm Disenable, 例如:

Example of Alarm Time Setting

Alarm Preset alarm time	Day-of-week							12-hour mode				24-hour mode				
	Sun.	Mon.	Tue.	Wed.	Th.	Fri.	Sat.	1	1	1	1	1	1	1	1	
	WW	WW	WW	WW	WW	WW	WW	h	r	m	in	h	r	m	in	
00:00 a.m. on all days	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0	0
01:30 a.m. on all days	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0	0
11:59 a.m. on all days	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9	0
00:00 p.m. on Mon. to Fri.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0	0
01:30 p.m. on Sun.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0	0
11:59 p.m. on Mon., Wed., and Fri.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9	0

每天凌晨 12 点报警  
每天凌晨 1:30 报警  
每天上午 11:59 报警  
周 1 到周 5 下午 0  
仅周天下午 1:30 报警  
周 1、3、5 晚上 11:59 报警

## 9.6 Alarm\_W 寄存器(B-Ch)

### Alarm\_D Minute Register (Address Bh)

D7	D6	D5	D4	D3	D2	D1	D0
-	DM40	DM20	DM10	DM8	DM4	DM2	DM1
0	DM40	DM20	DM10	DM8	DM4	DM2	DM1
0	Indefinite						

D7—D0: 0000 0000—0101 1001=00—59 与地址为 1h 的分钟计数器设置方法一样。

### Alarm\_D Hour Register (Address Ch)

D7	D6	D5	D4	D3	D2	D1	D0
-	-	DH20 DP/ $\bar{A}$	DH10	DH8	DH4	DH2	DH1
0	0	DH20 DP/ $\bar{A}$	DH10	DH8	DH4	DH2	DH1
0	0	Indefi nite	Indefi nite	Indefi nite	Indefi nite	Indefi nite	Indefi nite

与地址为 2h 的小时计数器设置方法一致。

## 9.7 控制寄存器 1(Eh)

D7	D6	D5	D4	D3	D2	D1	D0	
WALE	DALE	$\overline{12/24}$	$\overline{CLEN2}$	TEST	CT2	CT1	CT0	(For Writing)
WALE	DALE	$\overline{12/24}$	$\overline{CLEN2}$	TEST	CT2	CT1	CT0	(For Reading)
0	0	0	0	0	0	0	0	Default Settings *)

D7、D6: Alarm\_W , Alarm\_D 位, 1—预警中断开启, 0—预警中断关闭(默认)

D5:12/24 时间制选择位, 1—24 小时模式, 0—12 小时模式(默认), 如下表格:

24H 模式	12H 模式	24H 模式	12H 模式
00	12(AM12)	12	32(PM12)
01	01(AM1)	13	21(PM1)
02	02(AM2)	14	22(PM2)
03	03(AM3)	15	23(PM3)
04	04(AM4)	16	24(PM4)
05	05(AM5)	17	25(PM5)
06	06(AM6)	18	26(PM6)
07	07(AM7)	19	27(PM7)
08	08(AM8)	20	28(PM8)
09	09(AM9)	21	29(PM9)
10	10(AM10)	22	30(PM10)
11	11(AM11)	23	31(PM11)

D4:32kHz 时钟输出位 2, 1—输出 Disable, 0—输出 Enable (默认)

D3:测试位, 1—测试模式。0—正常工作模式(默认)

D2、D1、D0:周期中断方式选择位, 如下表格:

CT2	CT1	CT0	描述	
			波形模式	中断时序
0	0	0	/	OFF(默认 High)
0	0	1	/	输出 Low
0	1	0	脉冲	2Hz
0	1	1	脉冲	1Hz
1	0	0	电平	1 次/秒
1	0	1	电平	1 次/分

1	1	0	电平	1次/时
1	1	1	电平	1次/月

## 9.8 控制寄存器 2(Fh)

D7	D6	D5	D4	D3	D2	D1	D0	
VDSL	VDET	XST	PON	QLENJ	CTFG	WAFG	DAFG	(For Writing)
VDSL	VDET	XST	PON	QLENJ	CTFG	WAFG	DAFG	(For Reading)
0	0	Indefinite	1	0	0	0	0	Default Settings *)

D7: VDD 输入电压 VTH 设置位, 1—1.3V, 0—1.6V(默认)

D6:输入电压电平指示位, 1—VDET<VTH, 0— VDET>VTH (默认)

D5:振荡挂起侦测位, 0—侦测 Enable, 1—Disable

D4:Power-On-Reset 标识位, 0—正常状况, 1—检测 VDD Power-On-Reset(默认)

D3:32kHz 时钟输出位 1, 1—输出 Disable, 0—输出 Enable (默认)

D2:周期中断指示位, 1—周期中断输出 Low, 0—周期中断输出 High (默认)

D1、D0:Alarm\_W 和 Alarm\_D 指示位, 1—当前时间和预设报警时间一致, 0—当前时间和预设报警时间不一致(默认)

## 10 附录

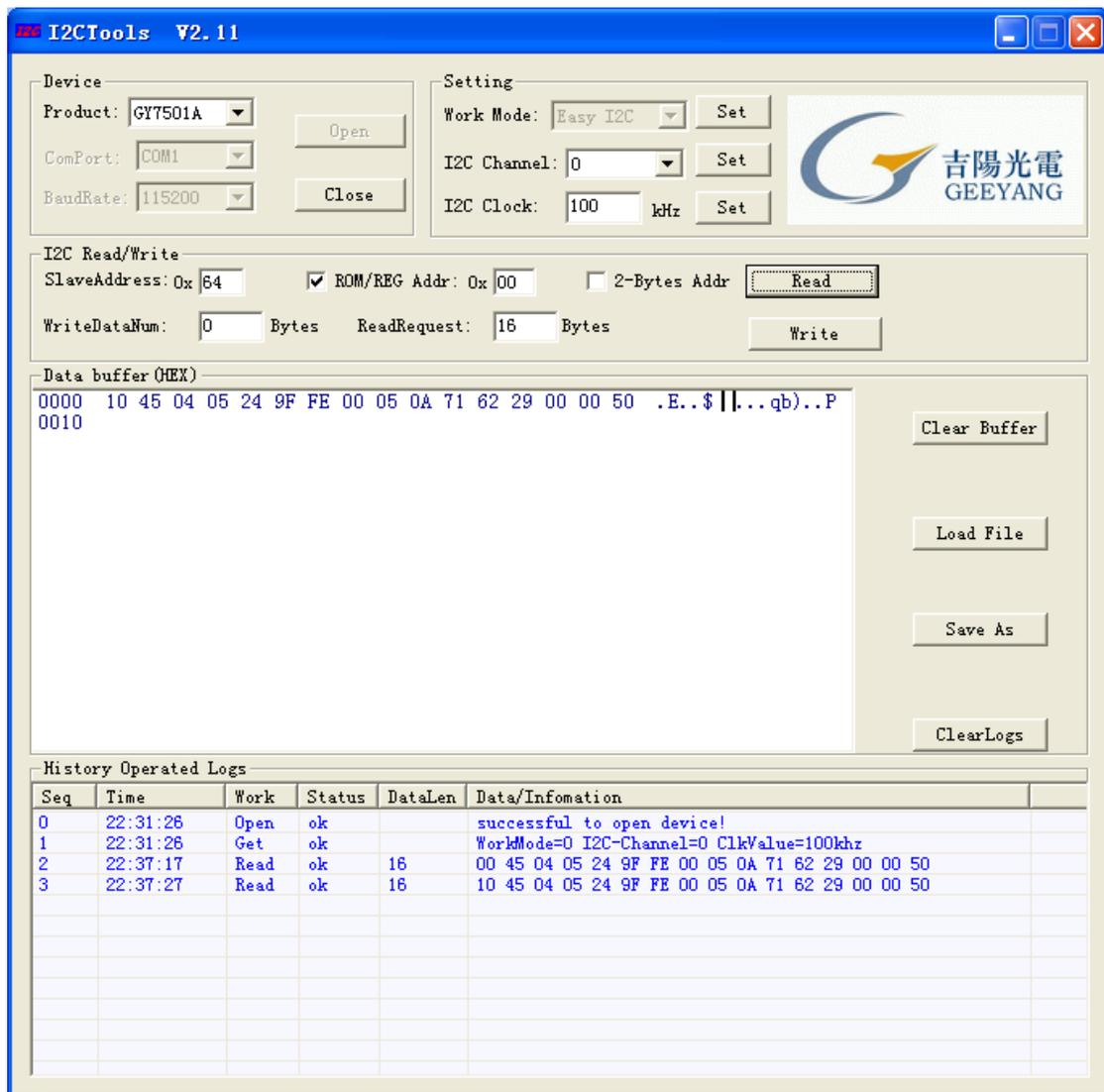
读操作:

RICOH R2023T 初始化设置解析:

第一次使用设备时, 插入 USB 口, 会看到 Demo Board 上所有灯亮起,除电源 LED1 外, 另外两个分别是 LED3(INTRA#)、LED2(INTRB#), 表明此时没有中断信号。如下图:



运行 I2C Tool 软件, 打开设备,执行 Read 操作, 画面如下:



00 45 04 05 24 9F FE 00 05 0A 71 62 29 00 00 50 对应分别是 0—Fh 的寄存器值

0h 00: 60 秒或 1 秒

1h 45: 45 分

2h 04: 上午 4 点

3h 05: 周 5

4h 24: 24 日

5h 9F: 不定

6h FE: 不定

7h 00: RTC 始终保持 32768kHz 不变

8h 05: 5 秒

9h 0A: 上午 10 点

Ah 71: 周 4、5、6、7 可以报警

Bh 62: 2 秒

Ch 29: 晚上 9 点

Dh 00: 不定

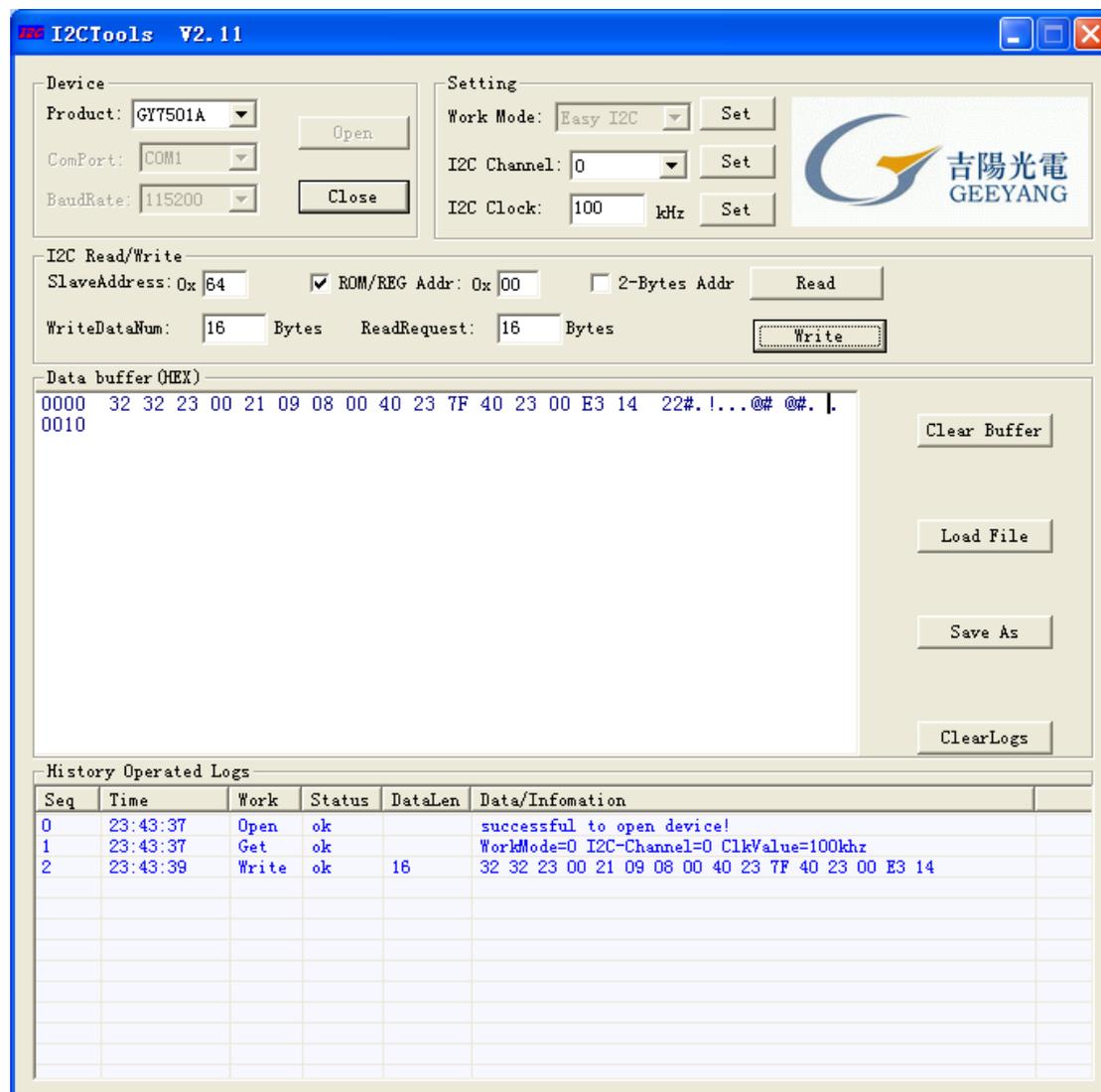
Eh 00: 中断信号关闭, 12 小时模式, 32KOUT 使能, 处于正常 Mode

Fh 50: 1.6V 的  $V_{TH}$ , 同时电压检测功能 Disable。

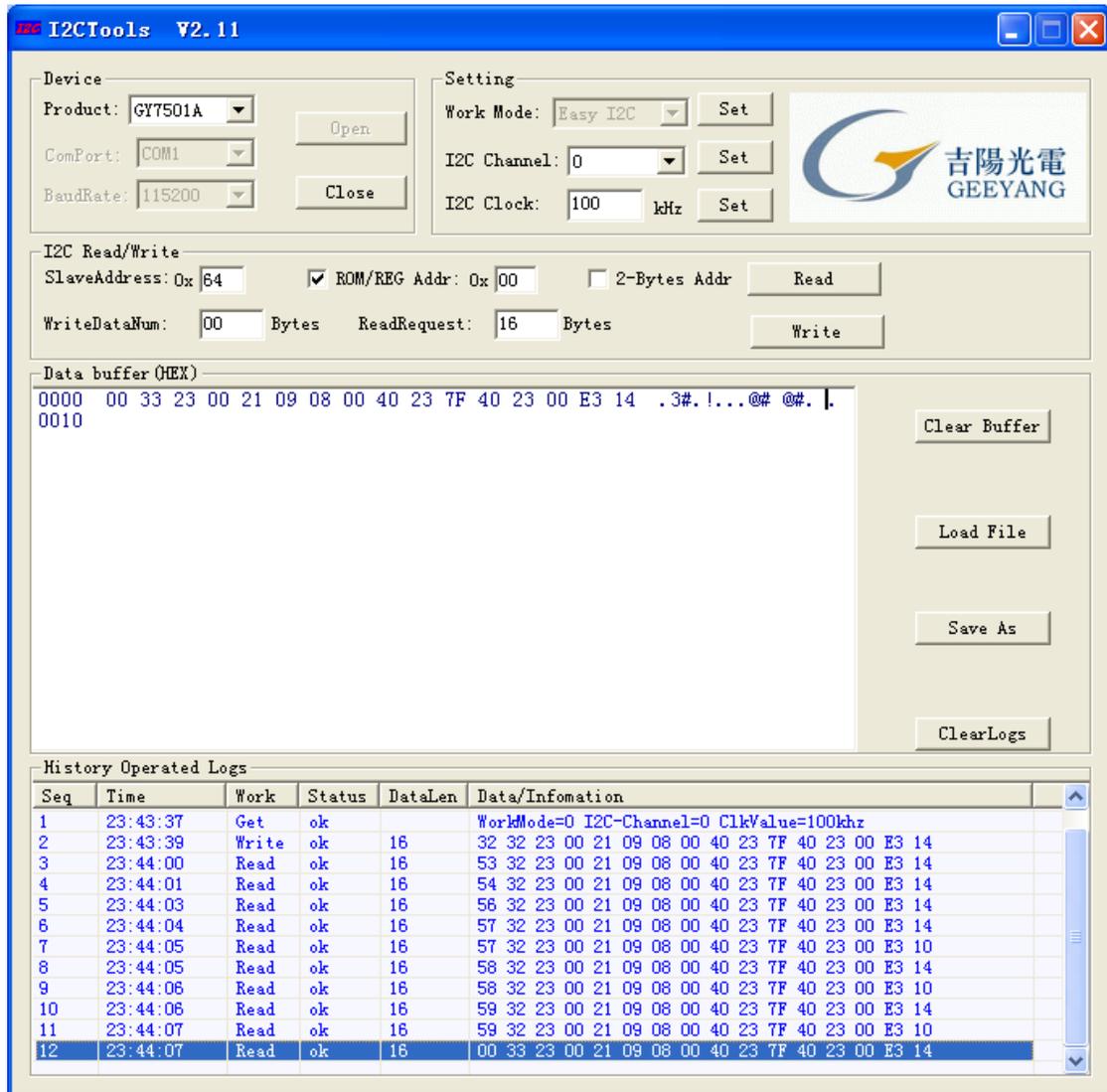
写操作:

写入 2008-9-21 周天 23 点 30 分 30 秒, 并在 23 点 40 分报警, RTC 始终保持 32768kHz 不变, 同时中断开启, 并以 1S 的频率发出, 显示在 LED3 上是每一秒闪一次。

写入 16 位数据, 点击 Write, History Operation Logs 会显示出写成功。



读取刚才的数据:



- 1、从上图发现，数据写入成功，在 Demo Board 上看见 LED3 每秒闪一次，在 History Operation Logs 栏中我们看到 Fh 位寄存器读取时 10 和 14 不断交替，表明中断信号与 LED3 的关系，10 的时候无中断，灯亮，14 的时候中断发出，灯灭。正好间隔一秒。
- 2、我们定在 23 点 40 分报警，当我们 RTC 时间运行到 23 点 40 分时可以看到 LED2 灯熄灭，表明中断已经发出。这时候 LED3 也停止闪烁。中断有效。