



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808



威汉德科技有限公司

血压血糖技术文档

时间: 2016 年 12 月





深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

声明

此文档为深圳市威汉德科技有限专属技术文档, 只限于本司以及本司客户使用, 他人未经允许不得查看使用此文档, 如若造成不良后果将追究法律责任。

特此声明

深圳市威汉德科技有限公司

2016 年 12 月



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

目录

第一章 血压串口协议与蓝牙协议.....	4
1.1 串口通信协议.....	4
1.1.1 通讯协议数据格式.....	4
1.1.2 命令码格式.....	5
1.2 血压计蓝牙通讯协议.....	6
1.2.1 通信协议数据格式.....	7
1.2.2 上传数据.....	8
1.2.3 命令特性.....	8
1.2.4 命令码格式.....	10
第二章 血糖串口协议与蓝牙协议.....	13
2.1 通讯协议.....	13
2.1.1 发送方.....	13
2.1.2 接收方.....	14
2.1.3 数据格式.....	14
2.1.4 试纸类型事件.....	15
2.1.5 血糖仪工作状态.....	15
2.1.6 机器待机与错误事件.....	17
2.1.7 返回事件.....	18
2.1.8 终端命令.....	19
2.2 血糖蓝牙协议.....	22
2.2.1 血糖仪测试.....	22
2.2.2 血糖仪返回.....	26
2.2.3 血糖仪命令.....	29
2.2.4 UUID 定义.....	31
第三章 上位机软件参考.....	32



第一章 血压串口协议与蓝牙协议

1.1 串口通信协议

本章中，上位机是指手机、平板、电脑、监护仪等平台，我司蓝牙血压计提供一组串口对外通信，串口URAT参考如下：

波特率为：9600bit/s、8个Data位、1个Stop位、没有奇偶校验位。

1.1.1 通讯协议数据格式

血压计模块送出数据格式 (血压计模块 >>> 上位机)

序号	字节定义	内容
1	开始位	0xFFFE
2	长度码	除开始位外所有字节的长度
3	校验和	8 位校验和
4	命令码	8 位命令
5	数据	要传送的数据

枚举如下：

0xFFFE	长度	校验和	命令码	N 个数据
--------	----	-----	-----	-------

血压计模块接收数据格式 (上位机>>>血压计模块)

序号	字节定义	内容
1	开始位	0xFF
2	命令码	8 位命令

枚举如下：

0xFF	命令码
------	-----

通讯中的长度码和校验和：



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
 电话: +86-0755-27388751 传真: +86-0755-27388751-808

长度码：指一串完整的通讯数据中，除了“开始位”以外所有字节数的总合，包括长度码本身。

校验和：指“长度码”“命令码”“数据”的内容相加所得的结果只取低 8 位。

如：0FAH“长度”校验和”55H, 00H, 65H, 00H, 44H, 50H。

其中“长度”= 07H “校验和”= 07H+ 55H+ 00H + 65H+ 00H + 44H+50H = 0A2H。

1.1.2 命令码格式

上位机发出	血压计模块响应	命令内容
0x00 x 100		唤醒血压计模块
[0xFF,0xA0] x20		上位机要求血压计模块开始测量，
	[0xFFFE,0x03, 0x53,0x50] x1	血压计收到开始测量命令后，自身气压“归零”后回应 0x50 给上位机并开始充气进入测量血压。上位机收到 0x50 后进入测量状态。之后血压计在测量过程中不断发送当前的压力和测量到心跳的信息到上位机
	[0xFFFE,0x05,校验和,0x54,压力 (MSB,LSB)] x1	血压计模块发送当前压力和心跳信息，后紧跟两个字节的 数据表示压力和心跳.MSB.4=1 有心跳 MSB.4=0 无心跳 <div style="text-align: center;"> </div>
	[0xFFFE,0x08,校验和,0x55,SYS (MSB,LSB),DIA (MSB,LSB),PULS (1BYTE)]	血压计模块发送测量结果数据如下： SYS (MSB、LSB)，DIA(MSB、LSB)，PULS (Byte) 共 5 个 Byte。其中 SYS (MSB) .bit7=1 表示心率不齐。 SYS (MSB) .bit7=0 表示心率正常



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
 电话: +86-0755-27388751 传真: +86-0755-27388751-808

	[0xFFFE,0x04, 校验和,0x56, 错误码(1BYTE)] x 1	<p>血压计模块发送测量结果错误信息,后接 1 个 Byte 数据表示错误码。</p> <p>1 个 Byte 值=1 表示 7S 内打气不上 30mmHg(气袋没绑好)</p> <p>1 个 Byte 值=2 表示气袋压力超过 295mmHg 超压保护</p> <p>1 个 Byte 值=3 表示测量不到有效的脉搏</p> <p>1 个 Byte 值=4 表示干预过多(测量中移动、说话等)</p> <p>1 个 Byte 值=5 表示测量结果数值有误</p> <p>1 个 Byte 值=6 表示电池低电压</p> <p>1 个 Byte 值=7 表示 EEPROM 有误</p>
[0xFF,0xAC] x 20		上位机要求血压计模块进入校压模式,
	[0xFFFE,0x03, 0x5F,0x5C] x1	<p>血压计收到进入校压模式命令后, 回应 0x5c 给上位机。</p> <p>之后血压计模块持续发送当前的压力到上位机</p>
	[0xFFFE,0x07,校验和,0x55,SYS (MSB,LSB),DIA (MSB,LSB),PULS(1BYTE)]	<p>血压计模块发送当前压力值数据如下:</p> <p>当前压力值 (MSB、LSB), XY(MSB、LSB), Z (Byte) 共 5 个 Byte。</p>
[0xFF,0xA2] x20		上位机要求血压计模块传送记忆值
	[0xFFFE,0x0c,校验和,0x52, SYS (MSB,LSB), DIA(MSB,LSB), PULS(1BYTE), YEAR(1BYTE), MON(1BYTE), DAY(1BYTE), HOUR(1BYTE),	<p>血压计模块发送记忆结果数据如下:</p> <p>SYS (MSB、LSB), DIA(MSB、LSB), PULS (Byte) 年 (Byte),月 (Byte),日 (Byte),时 (Byte),分 (Byte) 共 10 个 Byte。其中 SYS (MSB).bit7=1 表示心率不齐。SYS (MSB).bit7=0 表示心率正常</p>
[0x00,0x00]x 20		上位机要求血压计停止测量
	[0xFFFE,0x03, 0x56,0x53] x1	<p>血压计模块确认测量中止, 放气停止测量回到省电状态。</p> <p>并发送 0x53 给上位机</p>

1.2 血压计蓝牙通讯协议

蓝牙 4.0 血压计主要是通过蓝牙 4.0 模块把血压计的数据上传到上位机中, 我司的蓝牙血压协议是基于透传来实现, 将血压计的数据



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

透传到上位机，也时也将上位机的命令下达给血压计。上位机主要有开机，关机两个命令。

UUID:

0xFFFF4 -----Notify, 血压计上传数据特性

0xFFFF1 -----Write/Read, 血压计命令特性

1.2.1 通信协议数据格式

血压计模组送出 DATA 格式 (血压计模组 >>> 上位机)

序号	字节定义	内容
1	开始位	0xFFFFE
2	长度码	除开始位外所有字节的长度
3	校验和	8 位校验和
4	命令码	8 位命令
5	数据	要传送的数据

0xFFFFE	长度	校验和	命令码	N 个数据
---------	----	-----	-----	-------

血压计模组接收 DATA 格式 (血压计模组 <<< 上位机)

序号	字节定义	内容
1	开始位	0xFF
2	命令码	8 位命令

0xFF	命令码
------	-----

通讯中的长度码和校验和:

长度码: 指一串完整的通讯数据中, 除了“开始位”以外所有字节数的总合, 包括长度码本身。

校验和: 指“长度码”“命令码”“数据”的内容相加所得的结果只取



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

低 8 位。

如: 0FAH“长度”校验和”55H, 00H, 65H, 00H, 44H, 50H

其中“长度”= 07H“校验和”= 07H+

1.2.2 上传数据

血压计在测试量, 产生的数据通过 0xFF4 的 UUID 上传数据。

上传的数据, 具体的数据格式如下:

序号	字节定义	内容
1	开始位	0xFFFE
2	长度码	除开始位外的所有字节的长度
3	校验和	8 位校验和
4	命令码	8 位命令
5	数据	要传送的数据

0xFFFE	长度	校验和	命令码	N 个数据
--------	----	-----	-----	-------

具体协议参数见上面通信协议。

1.2.3 命令特性

血压计在待机状态, 可以发送开机命令, 使血压计开始测量。

血压计在测量状态, 可以发送待机命令, 使血压计停止测量; 并进入待机模式。

序号	命令	说明
1 (图一、四)	0xA0 0xA0 0xA0 0xA0 0xA0 0xA0 0xA0 0xA0 0	机器开机, 并开始测量
2 (图五)	0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	机器停止测量, 并进入关机模式。



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

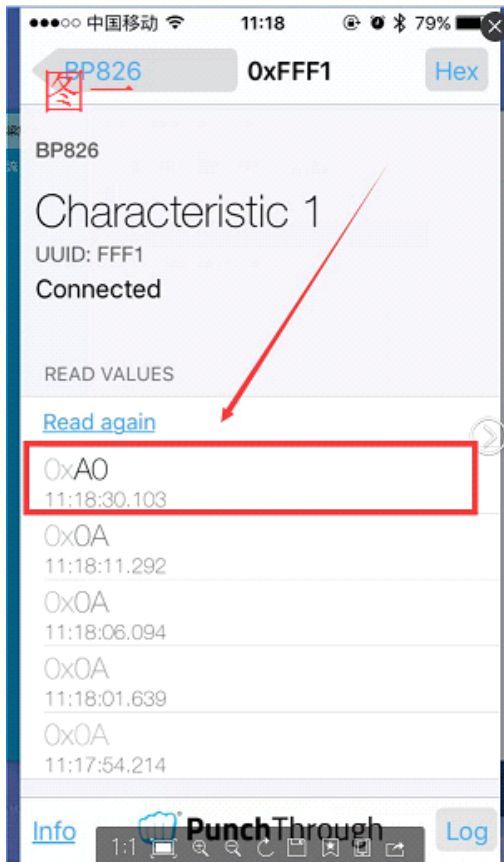


图 1 发送开机命令

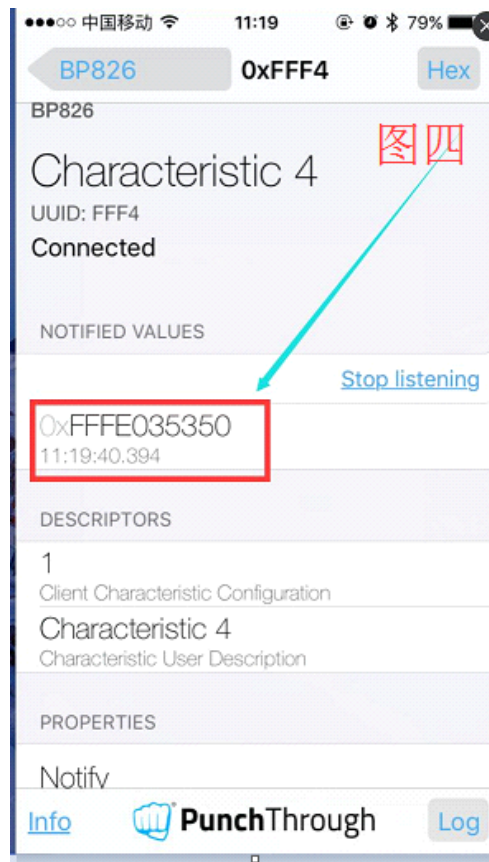


图 4 收到命令回复

上位机通过使用 lightbule 去发送 20 组开机指令（A0）到血压计蓝牙模块去唤醒蓝牙血压计，血压计唤醒后会给上位机回复 0xFFFE035350(图四)告诉上位机血压计已经被唤醒。



图 5

上位机通过使用 lightbule 去发送 20 组关机指令 (00) 到血压计蓝牙模块去关闭蓝牙血压计，血压计收到命令后会给上位机回复 0xFFFE035653 (图五) 告诉上位机血压计已经被关闭进入待机状态。

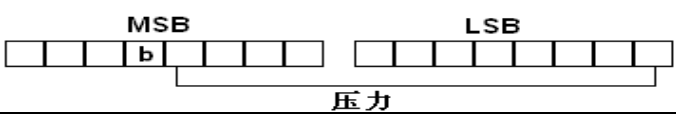
1.2.4 命令码格式

当接入串口进行调试的时候，只需要在上位机发送十次以上“F F A0”命令，即可让下位机开始工作，当测试完毕后，通过捕捉串口数据，对比下表命令即可得知血压模块是否正常工作。



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
 电话: +86-0755-27388751 传真: +86-0755-27388751-808

上位机发出	血压计模组回应	命令内容
[0xFF,0xA0]x20 (图三)	[0xFFFFE,0x03,0x53,0x50] x1	血压计收到开始测量命令后,自身气压“归零”后回应 0x50 给上位机并开始充气进入测量血压。上位机收到 0x50 后进入测量状态。之后血压计在测量过程中不断发送当前的压力和测量到心跳的信息到上位机
	[0xFFFFE,0x05,校验和,0x54,压力(MSB, LSB)] x1	血压计模组发送当前压力和心跳信息,后紧跟两个字节的 数据表示压力和心跳.MSB.4=1 有心跳 MSB.4=0 无心跳 
	[0xFFFFE,0x08,校验和,0x55,SYS(MSB, LSB),DIA(MSB,LSB),	血压计模组发送测量结果数据如下: SYS (MSB 、LSB), DIA(MSB 、LSB) , PULS (Byte) 共 5 个 Byte 。其中 SYS (MSB) .bit7=1 表示心率不齐 。 SYS (MSB) .bit7=0 表示心率正常
	[0xFFFFE,0x04,校验和,0x56,错误码(1BYTE)] x1	血压计模组发送测量结果错误信息,后接 1 个 Byte 数据表示错误码。 1 个 Byte 值=1 表示 7S 内打气不上 30mmHg(气袋没绑好) 1 个 Byte 值=2 表示气袋压力超过 295mmHg 超压保护 1 个 Byte 值=3 表示测量不到有效的脉搏 1 个 Byte 值=4 表示干预过多 (测量中移动、说话等) 1 个 Byte 值=5 表示测量结果数值有误 1 个 Byte 值=6 表示电池低电压 1 个 Byte 值=7 表示 EEPROM 有误
[0xFF,0xAC]x20	[0xFFFFE,0x03,	血压计收到进入校压模式命令后,回应 0x5c 给上位机。之后血压计模组持续发送当前的压力到上位机
	[0xFFFFE,0x07,校验和,0x55,SYS(MSB, LSB),DIA(MSB,LSB),	血压计模组发送当前压力值数据如下: 当前压力值 (MSB 、LSB), XY(MSB 、LSB) , Z (Byte) 共 5 个 Byte 。
[0x00,0x00]x20 图五	[0xFFFFE,0x03,0x56,0x53] x1	血压计模组确认测量中止,放气停止测量回到省电状态。并发送 0x53 给上位机

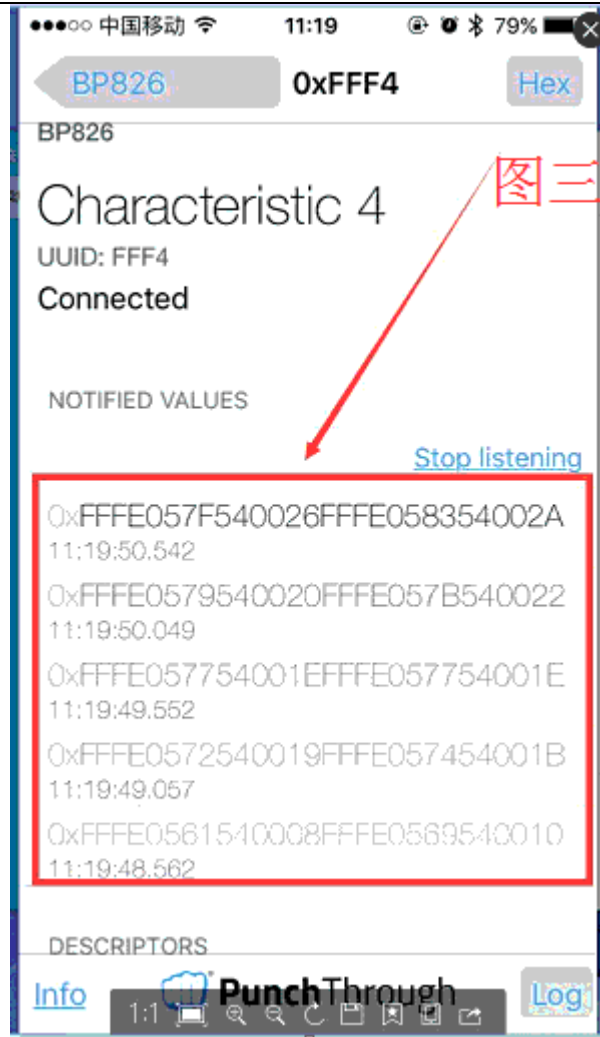


图 3

血压计收到开始测量命令后，自身气压“归零”后回应 0x50 给上位机并开始充气进入测量血压。上位机收到 0x50 后进入测量状态。之后血压计在测量过程中不断发送当前的压力和测量到心跳的信息到上位机，红色方框中的数据就是血压计发回上位机的血压、心跳的数据。



第二章 血糖串口协议与蓝牙协议

血糖模块有 6 个 I/O 接口，每个接口都有不同的定义，如下表：

表 2.1 接口定义

引脚序号	丝印	说明
1	GND	PCB 地
2	VCC	电源: $3V \leq VCC \leq 5V$
3	RXD	血糖模块串口接收端
4	TXD	自糖模块串口发送端
5	BLE_WAKEUP	机器工作状态位，高为 Sleep，低为工作态
6	WAKE_IN	外部唤醒血糖模块，下降沿唤醒。常态为高电平

2.1 通讯协议

命令采用 UART 通信方式，无主从设计，双方都可以主动发送数据、命令，并接收对方反馈信息。

2.1.1 发送方

发送方存在 4 种状态：空闲、发送、等待 ACK、重发，ACK 数据包为：0xAA 0x55 0x00。当系统初始化后，处于空闲状态，有收到发送数据请求时，系统进入发送状态。发送完数据包后，进入等待 ACK 状态，如果对应的数据命令有返回则不需要 ACK 返回，直接返回数据。如收到 ACK 或返回数据，退回到空闲模式。如果没有收到 ACK 或返回数据，超时后，进入重发。重发提出发送请求，系统进入发送状态。当重发 5 次都未成功，系统退回空闲模式。

当系统处于重发时，有新命令需要发送时。旧命令丢弃。进入新



命令的发送模式。

2.1.2 接收方

接收方存在 3 种状态：空闲，接收，发送 ACK。当系统处于空闲状态，有收到前导码后，进入接收状态，来接收数据。当接到完整的数据包时，接收方返回 ACK。当接收到非法数据包是，不做任何处理。数据包接收有超时设置，即在 200ms 内，数据出现断掉。系统退出接收模式，进入空闲模式。血糖仪做为接收时，有些命令需要会触发血糖仪发送一个命令返回给终端方。

2.1.3 数据格式

一个完整的命令由 5 部分组成：前导码，长度，命令，数据，校验和组成。其结构如下表：

表 2.2 数据格式

前导码	长度	命令	数据	校验和
0xAA,0x55	Length	CMD	Data[n]	CHECKSUM

前导码：占用 2 字节，固定为 0xAA,0x55

长度：占用 1 字节，为 $\text{sizeof}(\text{CMD})+\text{sizeof}(\text{Data})+\text{sizeof}(\text{Checksum})$ 。

CMD：占用 1 字节，为实际操作的命令识别字。

数据：占用字节不确定，跟据实际命令需求而变化。可以为空

校验和：为 $\text{CMD}+\text{Data}[0]+\text{Data}[1]+\dots+\text{Data}[n]$ 的累加和。占用两字节，高位在前面字节，低 8 位在最后一字节。

所有命令数据包，都要符合这个格式，包括 ACK 命令包,血糖仪



命令。

2.1.4 试纸类型事件

当系统为血糖试纸开机时，会对试纸进行识别。并将试纸的类型发到终端。该命令为事件型命令，无数据返回。只有 ACK 返回。

参考下表：

表 2.3 试纸类型事件

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length=4	0xC5	Strip-type(1Byte)	Checksum

Length =4, 总长度为 7 字节；

命令识别字:0xC5；

数据：占用 1 字节，表示试纸的类型。参考表 2.4。

校验和。

表 2.4 试纸类型

试纸类型	数据	注意
新试纸	0x00	
自检用试纸	0x10	2
校正电阻	0x20	1
测试电阻	0x30	1
使用过的试纸	0x40	
不能识别	0x41	

注意：

1 为量产校正时使用；

2 为自检测试使用。

2.1.5 血糖仪工作状态

(一) 滴血事件消息：

当用户在测试血糖时，对试纸进行滴血时；当系统识别到滴血事



件后，产生滴血事件，并将事件消息发送给终端。只有 ACK 返回。

表 2.5 滴血事件

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0xC6	Msg	Checksum

命令识别字:0xC6;

数据: 占用 1 字节, Msg—保留;

校验和。

(二)、测试倒计时事件:

当用户测试血糖时，滴血后；系统会进入倒计时等待状态，当倒计时到后会检测血糖的数据。并触发血糖结果事件。具体数据格式参考表。

表 2.6 计时事件

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0xC7	CNT	Checksum

命令识别字:0xC7

数据: 占用 1 字节, CNT—当时倒计时数, 0-50, 单位为 s。

校验和。

(三)、血糖结果事件:

血糖测试完成后，以及血糖查询事件。都会触发血糖结果事件。发送对应血糖数据给终端。只有 ACK 返回。具体数据格式如表:

表 2.7 结果事件

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0xC8	Data[6]	Checksum

命令识别字:0xC8

数据: 占用 6 字节, 具体数据参考表 2.8。靠近命令字的为 Data1,



靠近校验和的为 Data6。校验和。

表 2.8 试纸数据格式

数据	位域	描述	注意
Data1	7		
	0~6		
Data2	4~7		
	2~3		
	0~1	血糖测试值	
Data3	0~7	Data1.bit1是MSB Data2.bit0是LSB, 单位为dg/	
Data4	3~7		
	0~2		
Data5	6~7		
	0~5		
Data6	7		
	0~6	测试时电压	1

注意):

1 注意这些值有些模块上没有使用到, 恒为 0。

2.1.6 机器待机与错误事件

(一)、机器待机:

机器睡眠前会进入待机, 待时前发送以下数据给终端, 告知其待机。不需要 ACK。

表 2.9 机器待机

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0xC9	Data	Checksum

命令识别字:0xC9

数据: 占用 1 字节, 备用。

校验和。



(二)、错误事件:

机器发生错误时, 会发送以下数据给终端, 告知其错误。需要 ACK。同时终端可以查询错误, 该事件消息也会返回去终端。

表 2.10 错误事件

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0xCA	Data	Checksum

命令识别字:0xCA;

数据: 占用 1 字节, 错误码, 0 为无错误;

校验和。

2.1.7 返回事件

主要有三大返回事件: 查询结果返回、设置试纸码返回、机器温度返回。

(一)、查询结果返回:

在测试过程和测试完成时, 终端可以发送命令去查询测试是否完成和测试结果, 血糖仪会返回以下数据, 需要 ACK。

表 2.11 查询结果

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x62	Data	Checksum

命令识别字:0xCA;

数据: 占用 3 字节, Data[0]表示当前测试是否完成, 0 未完成, 其它完成;

当测试完成时, Data[1]~Data[2]表示测试血糖值;



校验和。

(二)、设置试纸码返回:

进行测试前, 如果试纸的 CODE 发生变化时, 请通过命令去设置试纸的 CODE。设置后, 血糖仪会返回以下数据, 需要 ACK。

表 2.12 试纸码返回

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x63	Data	Checksum

命令识别字:0xCA;

数据: 占用 2 字节, Data[0]~ Data[1]设置后的试纸 CODE, 请确认是与设置的值一样。 ;

校验和。

(三)、机器温度返回:

取温度命令的返回值。

表 2.13 温度返回

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x68	Data	Checksum

命令识别字:0xCA;

数据: 占用 2 字节, Data[0]~ Data[1]当前温度值, Data0 为高位, Data[1]为低位, 单位 0.1 °C;

校验和。

2.1.8 终端命令

终端命令主要用于发送终端命令给血糖仪, 用于查询记录, 设置时间, 以及量产测试时使用。



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

(一)、查询结果命令:

终端需要查询血糖仪的测试结果时, 发送如下命令触发血糖仪查询结果返回。如表: 2.14

表 2.14 查询结果命令

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length=2	0x52	NULL(0 字节)	Checksum

命令识别字:0x52;

数据: 数据为空;

校验和。

(二)、查询错误命令:

不知道血糖仪是否发生错误时, 请通过该命令查询错误, 触发错误返回。

表 2.15 查询错误命令

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x53	NULL(0)	Checksum

命令识别字:0x53;

数据: 数据为空;

校验和。

(三)、清除错误命令:

清除当前错误, 并触发错误返回, 返回的错误一定是 0, 如果不是 0, 则清除出错。

表 2.16 清除错误命令

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x54	NULL(0)	Checksum



命令识别字:0x54;

数据: 数据为空;

校验和。

(四) 设置试纸 CODE 命令:

当试纸 CODE 与当前试纸的 CODE 不一致时, 需要设置试纸 CODE, 设置后会有设置度纸码返回。

表 2.17 设置试纸 CODE

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x56	NULL(0)	Checksum

命令识别字:0x56;

数据: 数据 2 个字节, 为要设置的试纸 CODE, 注意的试纸 CODE 必须与该值一致;

校验和。

(五)、取当前温度:

取当前机器的温度, 每次试纸插入时会检测温度。

表 2.18 设置试纸 CODE

前导码	长度	命令字	数据	校验和
0xAA,0x55	Length	0x5D	NULL(0)	Checksum

命令识别字:0x5D;

数据: 数据 2 个字节, 为要设置的试纸 CODE, 注意的试纸 CODE 必须与该值一致;

校验和。



2.2 血糖蓝牙协议

血糖仪蓝牙协议基于标准蓝牙的标准血糖仪的 profile 修订:

1. Glucose Profile_v1.0 (蓝牙 SIG 制定)
2. Glucose Service_v1.0 (蓝牙 SIG 制定)
3. Bluetooth Core Specification V4.00

血糖命令和血糖返回是我们增加的 UUID, 血糖数据用的是 Glucose Service_V1.0 里面定义的 UUID(0x2A18), 其数据格式也如 GlucoseService_V1.0 所述;血糖数据测试是基于 Glucoservice_V1.0 的数据进行了详细的描述。

UUID 说明:

0x2A18 ----- 血糖数据, Notify;

0x2A34 ----- 血糖数据详细内容, 发前血糖协议无该部分数据, Notify;

0x2A51 ----- Glucose Feature, 当前未使用, Indicate;

0x2A52 ----- Read Access Control Point, 可读写.当前未测试;

0x2A6C ----- 血糖仪命令;

0x2A6D ----- 血糖仪返回。

2.2.1 血糖仪测试

血糖仪测试特性 (0x2A18) 主要用于返回血糖仪的测试结果, 它是基于 BLE 的蓝牙 SIG 的标准 profile。该特性的每个操作域的说明



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
 电话: +86-0755-27388751 传真: +86-0755-27388751-808

如下:

表2.19血糖仪测试特性数据描述(BGM data, 15Byte)

标志	时基	参考时间 (描述如表3-2)						时间偏移	血糖数据	采样信息
		年	月	日	时	分	秒			
1字节	2字节	2字节	1字节	1字节	1字节	1字节	1字节	2字节	2字节	1字节
0x87 (表3-1)	0x00 0x00	0Xdd 0x07	0x0A	0x19	0x13	0x32	0x24	0x00 0x00	0xF0 0X28 (Table2-3)	0x11 (Table2-4)

示例: 87 : 00 : 00 : DD : 077 : 0A : 19 : 13 : 32 : 24 : 00 : 00 : F0 : 28 :

11

时间偏移为当前, 血糖单位(mmol/L)), 餐后, 2013 年 10 月 25 日 19 时 40 分 36 秒, 血糖测试结果为 4.0mmol/L, 采样为手指毛细血管的全血。

注意: 特性和描述说明和入口参数请参考蓝牙 SIG 安排的 U U I D 号。

表 2.20 标志的位域结构

标志的位域	描述	置 0	置1
Bit0 (0x01)	是不存在时间偏移	-	存在
Bit1 (0x02)	是否存在测试的细节信息	-	存在
Bit2 (0x04)	血糖单位	Unitt_kg/L	Unit_mmol/L
Bit3 (0x08)	血糖信息为当前测试结果反馈	0(该应用常为0)	-
Bit4 (0x10)	是否有测试上下文	无上下文	有上下文
Bit5 (0x20)	保留	-	
Bit6 (0x40)	餐前/餐后	-	餐前
Bit7 (0x80)		-	餐后

注意: Bit6, Bit7同时为0时, 表示品管液(PC)。

示例: 标志(Flags)二进制值设置为00011011, 位0,1,3,4被置1, 下



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

面为设置的细节:

- 时间偏移使能;
- 测量血糖单位为kg/L (数据为2字节);
- 血类型使能 (nibble, 4位) ;
- 采样点 (nibble, 4位) ;
- 血糖值 (2 字节) ;
- 后面有测试血糖上下文。

表 2.21 参考时间数据结果 (长度为 7 字节)

参考时间字节	值
[0-1]	年(0 = 无效, 2 字节值)
[2]	月(1--12 , 其它= 无效)
[3]	日(1-31, 其它值无效)
[4]	时(0-23)
[5]	分(0-59)
[6]	秒(0-59)

表2.22 血糖测量数据

Exponent(n,4bit)	Mantissa(m,12 bit)
Bit15-bit12	Bit11-bit0
Exponent = $-(0 \times 10 - n)$; Data = Mantissa * (10^{Exponent})	

Exponent:小数点位置

Mantissa整数。



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

小数点位置为: 0×10^{-n} , 当最高四位为 $0 \times 0F$ 时, 小数点只有 1 位。

最高四位为 E 时, 小数点为 2 位。低 12 位为实际数据。

用公式表达为下表, 最高四位为 n , 低 12 位为 m 。

$$\text{血糖值} = m * 10^{-(0 \times 10^{-n})}$$

**Note 1. $\langle \text{mg/dL} \rangle = \langle \text{kg/L} \rangle * 1000,000$

**Note 2. $\langle \text{mmol/L} \rangle = \langle \text{mol/L} \rangle * 1,000$

**Note 3. $\langle \text{mg/dL} \rangle = 18 * \langle \text{mmol/L} \rangle$

血糖测试数据存在是可以选择的, 但当血糖测试数据存在时, 采样信息必须存在; 采样信息为采样血类型和采样位置组成, 两者组成一个字节。当采样血类型存在时, 采样位置也必须存在。当采样信息存在时, 请将标志的第 1 位置 1; 否则置 0。

表 2.23 采样信息

高 4 位(采样位置)		低 4 位 (血类型)	
Bit7-bit4	描述	Bit3-bit0	描述
0	保留	0	保留
1	手指	1	毛细血管全血
2	AST	2	毛细血管血浆
3	耳垂	3	静脉血全血
4	CONTROL L 的溶液	4	静脉血血浆
5-14	保留	5	动脉血全血
15	NA	6	动脉血血浆
		7	未确定位置全血
		8	未确定位置血浆
		9	组织间液(ISF)
		10	Control Solution
		11-15	保留



2.2.2 血糖仪返回

该命令仅在蓝牙检查的 Firmware 版本为 1.1 之后的版本可用。

血糖仪命令主要用于蓝牙操作，以及血糖仪操作时，需要血糖仪返回的交互信息。该特性为 Notiy。

```
#define GLU_RET_UUID 0x2A6D ///GLU_RET
```

血糖仪返回主要有以下几个数据域：操作类型、长度、数据。所有数据最长 20 字节。

表 2.24 血糖仪命令

BYTE1	0	无操作
	1	蓝牙端返回
	2	UART透传
	3~255	保留
BYTE2	N(1<n<17)	后续有n个字节
BYTE3 --- BYTEn	有效数据	命令数据实体。命令数据详细说明后续详细文件补充。

(一)、血糖仪数据返回：

糖仪的数据返回为血糖仪命令操作对应的返回。表2.25为命令返回的详细说明：

表2.25 命令返回

标号	数据	描述
----	----	----



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
 电话: +86-0755-27388751 传真: +86-0755-27388751-808

Read_EEPROM_DATA_RET	0x5B,[ADDR_H],[ADDR_L],[Data]	Read_EEPROM_DATA, WRITE_EEPROM_DATA的返回, ADDR_H,ADDR_L组成16位 EEPROM地址, Data为对应地址的数据
Read_TOTAL_RCD_NO_RET	0xBE, [No],[RES]	Read_TOTAL_RCD_NO的返回, No为血糖仪已保存的条数,RES保留
Read_RTC_RET	0xA2, [Year],[Month],[date],[hour],[minute],[second]	Read_RTC,Write_RTC返回, 返回血糖仪的当前 R T C 时间,
READ_RCD_RET	0x5F,[Data1 – Data 7],	Read_RCD返回, 返回查询的record的数据, Data1-Data7: 7字节, 详细参考表2.3 [chk_H,chk_L]:Data1 – Data7的数据累加和.

表2.26 Record Data

Data	Bits	描述
Data1	4~7	测试时间月份
	2~3	0— 餐前(AC) 1— 餐后(PC) Other –品管液(QC)
	0~1	血糖测试值(单位mg/DL)
Data2	0~8	Data1.bit1是MSB Data2.bit0是LSB
Data3	3~7	测试时间日期
	0~2	测试时间小时
Data4	6~7	Data3.bit2是MSB Data4.bit6是LSB
	0~5	测试时间分钟
Data5	4~7	试纸Code
	0~3	年份, 实际年= Data.bit0~bit3+年份Offset
Data6	7	Event
	0~6	测试时电压
Data7	6~7	保留
	0~5	测试时温度值

(二)、血糖仪事件消息



血糖仪事件是血糖仪操作时，发生的事件。主动发送到蓝牙的事件。

表2.27 血糖仪事件

标号	数据	描述
WAKEUP_EVT	0x02,0x03,0xC4,[Data],0xAA	血糖仪醒来事件: 详情参考表2.5
CARD_IN_EVT	0x02,0x03,0xC5,[Data],0xAA	血糖仪试纸插入事件 详情参考表2.6
BLOOD_EVENT	0x02,0x03,0xC6,data,0xAA	用户已滴血, data*100ms为总的测试时间。
COUNT_EVENT	0x02,0x03,0xC7,[Count],0xAA	血糖仪计数事件, 暂时无该命令 Count: 当前计数值
EXIT_EVT	0xC9,[Data],0xAA	血糖仪进入睡眠模式 0x00 - 试纸拔出 0x01 - 其它事件

表2.28 唤醒源

标号	数值	描述
WAKEUP_NOTHING	0x00	无唤醒信号
WAKEUP_S_KEY	0x10	通过 S 按键唤醒
WAKEUP_M_KEY	0x11	通过 M 按键唤醒
WAKEUP_CODE_STRIP	0x12	通过插入试纸密码卡唤醒
WAKEUP_GLUCOSE_STRIP	0x13	通过插入血糖试纸唤醒
WAKEUP_CODE_CARD	0x14	通过密码卡唤醒
WAKEUP_UART	0x15	通过串口唤醒

表2.29 试纸类型识别

标号	数值	描述
NEW_STRIP_IN	0x00	全新试纸插入
CHECK_STRIP_IN	0x10	测试用试纸插入
CAL_RESISTOR_IN	0x20	校正用电阻插入
TEST_RESISTOR_IN	0x30	测试电阻插入
USED_STRIP_IN	0x40	使用过的试纸插入
UNKNOWN_STRIP_IN	0x41	不能识别试纸插入



2.2.3 血糖仪命令

血糖仪命令主要用于蓝牙手机端，对蓝牙设备的下端进行操作，如修改时间、读EEPROM，修改试纸类型等。该特性只具有写操作有效。

```
#define GLU_CMD_UUID 0x2A6C ///GLU_CMD
```

血糖仪命令主要有以下几个数据域：操作类型、长度、数据。所有数据最长20字节。

表2.30 血糖仪命令

BYTE1	0	无操作
	1	蓝牙端操作
	2	UART透传
	3 - 2 5 5	保留
BYTE2	N(1<n<17)	后续有n个字节
BYTE3 --- BYTE _n	有效数据	命令数据实体。命令数据详细1.1,1.2详细说明。

(一)、血糖仪蓝牙端操作：

当要求有返回时，数据无返回。表示操作失败。另外进入 B T 数据传输模式后，在退出蓝牙时，请退出数据传输模式。

表2.31 蓝牙端操作

数据	返回	描述
0x01,0x01,0x01	有返回：参考第二章 UART_WAKEUP_EVT.	蓝牙使能UART，唤醒血糖仪端。 血糖仪进入Bt数据传数模式。 需等5s之后，接收到返回后，才能进行操作。无返回时，表示进入失败
0x01,0x01,0x02	无	蓝牙关闭UART，血糖仪退出Bt数据传输模式。
Others	无	保留



(二)、血糖仪 UART 命令:

血糖仪的 UART 命令, 必须在血糖仪进入 Bt 数据传数模式后, 方可进行发送查询。

所有数据, 当有返回值, 但实际无返回数据, 表示操作失败。

表 2.32UART 命令

标号	数据	是否有返回	描述
Read_EEPROM_DATA	0xAA,0x55,0x05,0xA5,[ADDR_H],[ADDR_L],CHK_H,CHK_L	有返回, 参考第二章 Read_EEPROM_DATA_RET	读Eeprom[ADDR]地址的数据, ADDR=0—2047. CHK为0xA5之后(包括0xA5)的字节累加和。
WRITE_EEPROM_DATA	0xAA,0x55,0x06,0xB5,[ADDR_H],[ADDR_L],[Data],CHK_H,CHK_L	有返回, 参考第二章 Read_EEPROM_DATA_RET	写EEPROM数据, ADDR= 0-2047 Data=要写入的数据 CHK为0xB5之后(包括0xB5)的字节累加和。
Read_TOTAL_RCD_NO	0xAA,0x55,0x03,0xAE,0x00,0xAE	有返回, 参考第二章 Read_TOTAL_RCD_NO_RET	读血糖仪已保存多少 Record
READ_RTC	0xAA,0x55,0x03,0xA1,0x00,0xA1	有返回, 参考第二章 Read_RTC_RET	读当前血糖仪RTC(实时时钟)
MODIFY_RTC	0xAA,0x55,0x09,0xB1,[year],[month],[date],[hour],[minute],[sec],CHK_H,CHK_L	有返回, 参考第二章的 Read_RTC_RET	修改血糖仪RTC, CHK为0xB1之后(包括0xA5)的字节累加和。其他为要设置的时间。年为百年取余,即2016为16
Read_RCD	0xAA,0x55,0x05,0xAF,[INDEX_H],[INDEX_L],CHK_H,CHK_L	有返回, 参考第二章 READ_RCD_RET	读取血糖记录, 0<index<recordno, 1表示最新的测试结果。 CHK为0xAF之后(包括0xAF)的字节累加和



2.2.4 UUID 定义

1. Service ID

```
#define GLUCOSE_SERV_UUID          0x1808 // Glucose
#define DEVINFO_SERV_UUID          0x180A // Device Information
```

2. 血糖服务特性 UUID

```
#define GLUCOSE_MEAS_UUID          0x2A18 // Glucose Measurement
#define GLUCOSE_FEATURE_UUID      0x2A51 // Glucose Feature
#define GLUCOSE_FEATURE_UUID      0x2A51 // Glucose Feature
#define RECORD_CTRL_PT_UUID       0x2A52 // Record Access Control Point
#define BLE_IN_CMD_GLU_UUID        0x2A6C // BLE CMD
#define BLE_OUT_RET_GLU_UUID       0x2A6D // Uart RET
```

3. 设备信息服务特性 UUID

```
#define SYSTEM_ID_UUID             0x2A23 // System ID
#define MODEL_NUMBER_UUID          0x2A24 // Model Number String
#define SERIAL_NUMBER_UUID         0x2A25 // Serial Number String
#define FIRMWARE_REV_UUID          0x2A26 // Firmware Revision String
#define HARDWARE_REV_UUID          0x2A27 // Hardware Revision String
#define SOFTWARE_REV_UUID          0x2A28 // Software Revision String
#define MANUFACTURER_NAME_UUID     0x2A29 // Manufacturer Name String
#define IEEE_11073_CERT_DATA_UUID  0x2A2A // Regulatory Certification Data
List
#define PNP_ID_UUID                0x2A50 // PnP ID
```



第三章 上位机软件参考

本章是上位机软件如何连接、如何取数据的参考。由于市面操作系统众多，本文只讲述安卓下面的大概操作，其它系统请自行学习。另外，由于水平有限，本章仅供参考。

一、扫描 BLE 设备 activity

检查该设备是否支持 BLE 设备，要开发蓝牙 BLE ,建议一定要 Android4.4 或以上开发，

[java] view plain copy

```
1. if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE_BLUETOOTH_L
   E)) {
2.     Toast.makeText(this, R.string.ble_not_supported, Toast.LENGTH_SHO
   RT).show();
3.     finish();
4. }
```

初始化获得一个 bluetoothManager，并检测设备是否支持蓝牙

[java] view plain copy

```
1. final BluetoothManager bluetoothManager =
2.     (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVIC
   E);
```




```
3. mBluetoothAdapter = bluetoothManager.getAdapter();
```

[java] view plain copy

```
1. // Checks if Bluetooth is supported on the device.
2.     if (mBluetoothAdapter == null) {
3.         Toast.makeText(this, R.string.error_bluetooth_not_supported, Toa
         st.LENGTH_SHORT).show();
4.         finish();
5.         return;
6.     }
```

扫描 BLE 设备，然后添加到 listView 里面。

[java] view plain copy

```
1. private void scanLeDevice(final boolean enable) {
2.     if (enable) {
3.         // Stops scanning after a pre-defined scan period.
4.         mHandler.postDelayed(new Runnable() {
5.             @Override
6.             public void run() {
7.                 mScanning = false;
8.                 mBluetoothAdapter.stopLeScan(mLeScanCallback);
9.                 invalidateOptionsMenu();
10.            }
11.        }, SCAN_PERIOD);
12.
13.        mScanning = true;
14.        mBluetoothAdapter.startLeScan(mLeScanCallback);
15.    } else {
16.        mScanning = false;
17.        mBluetoothAdapter.stopLeScan(mLeScanCallback);
18.    }
19.    invalidateOptionsMenu();
20. }
```

二、蓝牙控制的服务 BluetoothLeService

在这个服务里面有一个很重要的回调函数 `BluetoothGattCallback()`，蓝



牙的数据读取和状态改变都会回调这个函数。

[java] view plain copy

```
1. private final BluetoothGattCallback mGattCallback = new BluetoothGattCallbac
   k() {
2.     @Override
3.     public void onConnectionStateChange(BluetoothGatt gatt, int status,
      int newState) {
4.         String intentAction;
5.         //收到设备 notify 值 (设备上报值)
6.         if (newState == BluetoothProfile.STATE_CONNECTED) {
7.             intentAction = ACTION_GATT_CONNECTED;
8.             mConnectionState = STATE_CONNECTED;
9.             broadcastUpdate(intentAction);
10.            Log.i(TAG, "Connected to GATT server.");
11.            // Attempts to discover services after successful connection
12.
13.            Log.i(TAG, "Attempting to start service discovery:" +
14.                mBluetoothGatt.discoverServices());
15.        } else if (newState == BluetoothProfile.STATE_DISCONNECTED) {
16.            intentAction = ACTION_GATT_DISCONNECTED;
17.            mConnectionState = STATE_DISCONNECTED;
18.            Log.i(TAG, "Disconnected from GATT server.");
19.            broadcastUpdate(intentAction);
20.        }
21.    }
22.
23.    @Override
24.    public void onServicesDiscovered(BluetoothGatt gatt, int status) {
25.        if (status == BluetoothGatt.GATT_SUCCESS) {
26.            broadcastUpdate(ACTION_GATT_SERVICES_DISCOVERED);
27.        } else {
28.            Log.w(TAG, "onServicesDiscovered received: " + status);
29.            System.out.println("onServicesDiscovered received: " + statu
      s);
30.        }
31.    }
32.
33.    @Override
```



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

```
34.     public void onCharacteristicRead(BluetoothGatt gatt,
35.                                     BluetoothGattCharacteristic charact
36.                                     eristic,
37.                                     int status) {
38.         //读取到值, 在这里读数据
39.         if (status == BluetoothGatt.GATT_SUCCESS) {
40.             broadcastUpdate(ACTION_DATA_AVAILABLE, characteristic);
41.         }
42.     }
43.     @Override
44.     public void onCharacteristicChanged(BluetoothGatt gatt,
45.                                         BluetoothGattCharacteristic char
46.                                         acteristic) {
47.         broadcastUpdate(ACTION_DATA_AVAILABLE, characteristic);
48.     }
49. }
```

在官方的 demo 中还使用到广播, 可能是因为大神写的, 要严谨些。我一开始看的时候就有点麻烦, 跳转的多麻烦。

[java] view plain copy

```
1.     private void broadcastUpdate(final String action) {
2.         final Intent intent = new Intent(action);
3.         sendBroadcast(intent);
4.     }
5.
6.     private void broadcastUpdate(final String action,
7.                                 final BluetoothGattCharacteristic character
8.                                 istic) {
9.         final Intent intent = new Intent(action);
10.        // This is special handling for the Heart Rate Measurement profile.
11.        Data parsing is
12.        // carried out as per profile specifications:
13.        // http://developer.bluetooth.org/gatt/characteristics/Pages/Charact
14.        eristicViewer.aspx?u=org.bluetooth.characteristic.heart_rate_measurement.xml
15.        if (UUID_HEART_RATE_MEASUREMENT.equals(characteristic.getUuid())) {
16.            int flag = characteristic.getProperties();
17.            int format = -1;
```



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

```
16.         if ((flag & 0x01) != 0) {
17.             format = BluetoothGattCharacteristic.FORMAT_UINT16;
18.             Log.d(TAG, "Heart rate format UINT16.");
19.         } else {
20.             format = BluetoothGattCharacteristic.FORMAT_UINT8;
21.             Log.d(TAG, "Heart rate format UINT8.");
22.         }
23.         final int heartRate = characteristic.getIntValue(format, 1);
24.         Log.d(TAG, String.format("Received heart rate: %d", heartRate));
25.         intent.putExtra(EXTRA_DATA, String.valueOf(heartRate));
26.     } else {
27.         // For all other profiles, writes the data formatted in HEX.对于
           所有的文件，写入十六进制格式的文件
28.         //这里读取到数据
29.         final byte[] data = characteristic.getValue();
30.         for (int i = 0; i < data.length; i++) {
31.             System.out.println("data....." + data[i]);
32.         }
33.         if (data != null && data.length > 0) {
34.             final StringBuilder stringBuilder = new StringBuilder(data.l
           ength);
35.             for(byte byteChar : data)
36.                 //以十六进制的形式输出
37.                 stringBuilder.append(String.format("%02X ", byteChar));
38.             // intent.putExtra(EXTRA_DATA, new String(data) + "\n" + stri
           ngBuilder.toString());
39.             intent.putExtra(EXTRA_DATA, new String(data));
40.         }
41.     }
42.     sendBroadcast(intent);
43. }
```

发送了广播之后就肯定有广播接收器，这个是在控制蓝牙的 activity 中，后面再说。

还有几个重要的函数比如

readCharacteristic(BluetoothGattCharacteristic characteristic)函数，读



取蓝牙中数据。

[java] view plain copy

```
1. public void readCharacteristic(BluetoothGattCharacteristic characteristic) {
2.     if (mBluetoothAdapter == null || mBluetoothGatt == null) {
3.         Log.w(TAG, "BluetoothAdapter not initialized");
4.         return;
5.     }
6.     mBluetoothGatt.readCharacteristic(characteristic);
7. }
```

有 `readCharacteristic(BluetoothGattCharacteristic characteristic)` 函数，当然就有 `writeCharacteristic(BluetoothGattCharacteristic characteristic)`，向蓝牙中写入数据。

[java] view plain copy

```
1. public void writeCharacteristic(BluetoothGattCharacteristic characteristic)
2. {
3.     if (mBluetoothAdapter == null || mBluetoothGatt == null) {
4.         Log.w(TAG, "BluetoothAdapter not initialized");
5.         return;
6.     }
7.     mBluetoothGatt.writeCharacteristic(characteristic);
8. }
```

另外在这个 `service` 中还有其他的一些函数例如初始化 `initialize()` 函数、连接蓝牙函数 `connect(final String address)`、断开蓝牙连接函数 `disconnect()` 等。

三、蓝牙控制 `DeviceControlActivity`

扫描到蓝牙设备之后就是对蓝牙进行自己需要的控制，比如写数据，读数据，获取设备信息，设备电量等。

在 `Service` 中讲到有一个广播，广播接收器就在这个 `activity` 中，通过



不同的 action 做出相应的操作。

注册的几种事件

[java] view plain copy

```
1. private static IntentFilter makeGattUpdateIntentFilter() {
2.     final IntentFilter intentFilter = new IntentFilter();
3.     intentFilter.addAction(BluetoothLeService.ACTION_GATT_CONNECTED);
4.     intentFilter.addAction(BluetoothLeService.ACTION_GATT_DISCONNECTED);
5.     intentFilter.addAction(BluetoothLeService.ACTION_GATT_SERVICES_DISCOVERED);
6.     intentFilter.addAction(BluetoothLeService.ACTION_DATA_AVAILABLE);
7.     return intentFilter;
8. }
```

[java] view plain copy

```
1. // Handles various events fired by the Service.处理服务所激发的各种事件
2. // ACTION_GATT_CONNECTED: connected to a GATT server.连接一个 GATT 服务
3. // ACTION_GATT_DISCONNECTED: disconnected from a GATT server.从 GATT 服务
   中断开连接
4. // ACTION_GATT_SERVICES_DISCOVERED: discovered GATT services.查找 GATT 服务
5. // ACTION_DATA_AVAILABLE: received data from the device. This can be a
   result of read
6. // or notification operations.从服务中接受数据
7. private final BroadcastReceiver mGattUpdateReceiver = new BroadcastReceiver() {
8.     @Override
9.     public void onReceive(Context context, Intent intent) {
10.        final String action = intent.getAction();
11.        if (BluetoothLeService.ACTION_GATT_CONNECTED.equals(action)) {
12.            mConnected = true;
13.            updateConnectionState(R.string.connected);
14.            invalidateOptionsMenu();
15.        } else if (BluetoothLeService.ACTION_GATT_DISCONNECTED.equals(ac
```



```
tion)) {
16.         mConnected = false;
17.         updateConnectionState(R.string.disconnected);
18.         invalidateOptionsMenu();
19.         clearUI();
20.     }
21.     //发现有可支持的服务
22.     else if (BluetoothLeService.ACTION_GATT_SERVICES_DISCOVERED.equals(action)) {
23.         //写数据的服务和 characteristic
24.         mNotifyGattService = mBluetoothLeService.getSupportedGattServices(UUID.fromString("0000ffe5-0000-1000-8000-00805f9b34fb"));
25.         characteristic = mNotifyGattService.getCharacteristic(UUID.fromString("0000ffe9-0000-1000-8000-00805f9b34fb"));
26.         //读数据的服务和 characteristic
27.         readMNotifyGattService = mBluetoothLeService.getSupportedGattServices(UUID.fromString("0000ffe0-0000-1000-8000-00805f9b34fb"));
28.         readCharacteristic = readMNotifyGattService.getCharacteristic(UUID.fromString("0000ffe4-0000-1000-8000-00805f9b34fb"));
29.     }
30.     //显示数据
31.     else if (BluetoothLeService.ACTION_DATA_AVAILABLE.equals(action)) {
32.         //将数据显示在 mDataField 上
33.         String data = intent.getStringExtra(BluetoothLeService.EXTRA_DATA);
34.         System.out.println("data----" + data);
35.         displayData(data);
36.     }
37. }
38. };
```

在发现了有可支持的服务之后会回调 Service 中的 onServicesDiscovered() 函数，并发送广播，在官方的 demo 中发现了可用的 Service 之后，就查找该 BLE 设备支持的所有服务和 characteristic，在这里不需要查找所有的服务，只需要向蓝牙写数据和读取数据的 Service 和 characteristic 的 UUID 即可。通过查询低功耗蓝牙（BLE）的数据手册可以得到所需要的 UUID。



深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

➤ 蓝牙数据通道【服务 UUID: 0xFFE5】

特征值 UUID	可执行的操作	字节数	默认值	备注
FFE9 (handle: 0x0013)	Write	20	无	写入的数据将会从串口 TX 输出

说明: 蓝牙输入转发到串口输出。APP 通过 BLE API 接口向此通道写操作后, 数据将会从串口 TX 输出。详细操作规则见《串口透传协议说明(桥接模式)》章节。

➤ 串口数据通道【服务 UUID: 0xFFE0】

特征值 UUID	可执行的操作	字节数	默认值	备注
FFE4 (handle: 0x000E)	notify	20	无	从串口 RX 输入的数据将会在此通道产生通知发给移动设备

有了这两个 Service 和 characteristic 的 UUID, 就可以对蓝牙发送数据, 并发出通知 (当写数据发生改变时发出)。

[java] view plain copy

```
1. <span style="font-size:18px;">//写数据的服务和 characteristic
2.         mnotyGattService = mBluetoothLeService.getSupportedGattServices(UUID.fromString("0000ffe5-0000-1000-8000-00805f9b34fb"));
3.         characteristic = mnotyGattService.getCharacteristic(UUID.fromString("0000ffe9-0000-1000-8000-00805f9b34fb"));
4.         //读数据的服务和 characteristic
5.         readMnotyGattService = mBluetoothLeService.getSupportedGattServices(UUID.fromString("0000ffe0-0000-1000-8000-00805f9b34fb"));
6.         readCharacteristic = readMnotyGattService.getCharacteristic(UUID.fromString("0000ffe4-0000-1000-8000-00805f9b34fb"));</span>
```

得到这两个 Service 和 characteristic 就可以向蓝牙发送数据了。

[java] view plain copy

```
1. private void read() {
2.     //mBluetoothLeService.readCharacteristic(readCharacteristic);
3.     //readCharacteristic 的数据发生变化, 发出通知
4.     mBluetoothLeService.setCharacteristicNotification(readCharacteristic
```




深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808

```
, true);  
5.         //Toast.makeText(this, "读成功", Toast.LENGTH_SHORT).show();  
6.     }
```

检测 readCharacteristic 的数据发生变化，发出通知。

向蓝牙发送数据。

[java] view plain copy

```
1. read();  
2.  
3.         final int charaProp = characteristic.getProperties();  
4.  
5.         //如果该 char 可写  
6.         if ((charaProp | BluetoothGattCharacteristic.PROPERTY_READ)  
7.             > 0) {  
8.             // If there is an active notification on a characteristi  
9.             c, clear  
10.            // it first so it doesn't update the data field on the u  
11.            ser interface.  
12.            if (mNotifyCharacteristic != null) {  
13.                mBluetoothLeService.setCharacteristicNotification( m  
14.                NotifyCharacteristic, false);  
15.                mNotifyCharacteristic = null;  
16.            }  
17.            //读取数据，数据将在回调函数中  
18.            //mBluetoothLeService.readCharacteristic(characteristic)  
19.            ;  
20.            byte[] value = new byte[20];  
21.            value[0] = (byte) 0x00;  
22.            if(edittext_input_value.getText().toString().equals(""))  
23.            {  
24.                Toast.makeText(getApplicationContext(), "请输入!  
25.                ", Toast.LENGTH_SHORT).show();  
26.                return;  
27.            }else{  
28.                WriteBytes = edittext_input_value.getText().toString  
29.                ().getBytes();  
30.                characteristic.setValue(value[0],BluetoothGattCharac  
31.                teristic.FORMAT_UINT8, 0);
```



```
23.         characteristic.setValue(WriteBytes);
24.         mBluetoothLeService.writeCharacteristic(characteristic);
25.         Toast.makeText(getApplicationContext(), "写入成功!",
26.             Toast.LENGTH_SHORT).show();
27.     }
28.     if ((charaProp | BluetoothGattCharacteristic.PROPERTY_NOTIFY) > 0) {
29.         mNotifyCharacteristic = characteristic;
30.         mBluetoothLeService.setCharacteristicNotification(characteristic, true);
31.     }
32.     editText_input_value.setText("");
33. }
```

一旦数据发生改变，就会发出通知，通知发出后就会调用下面的函数并发出广播。

[java] view plain copy

```
1. @Override
2.     public void onCharacteristicChanged(BluetoothGatt gatt,
3.         BluetoothGattCharacteristic characteristic) {
4.         broadcastUpdate(ACTION_DATA_AVAILABLE, characteristic);
5.     }
```

在广播接收器中接收到广播后，把数据显示在 EditText 上。下面是测试的两张图片。



深圳市威汉德科技有限公司

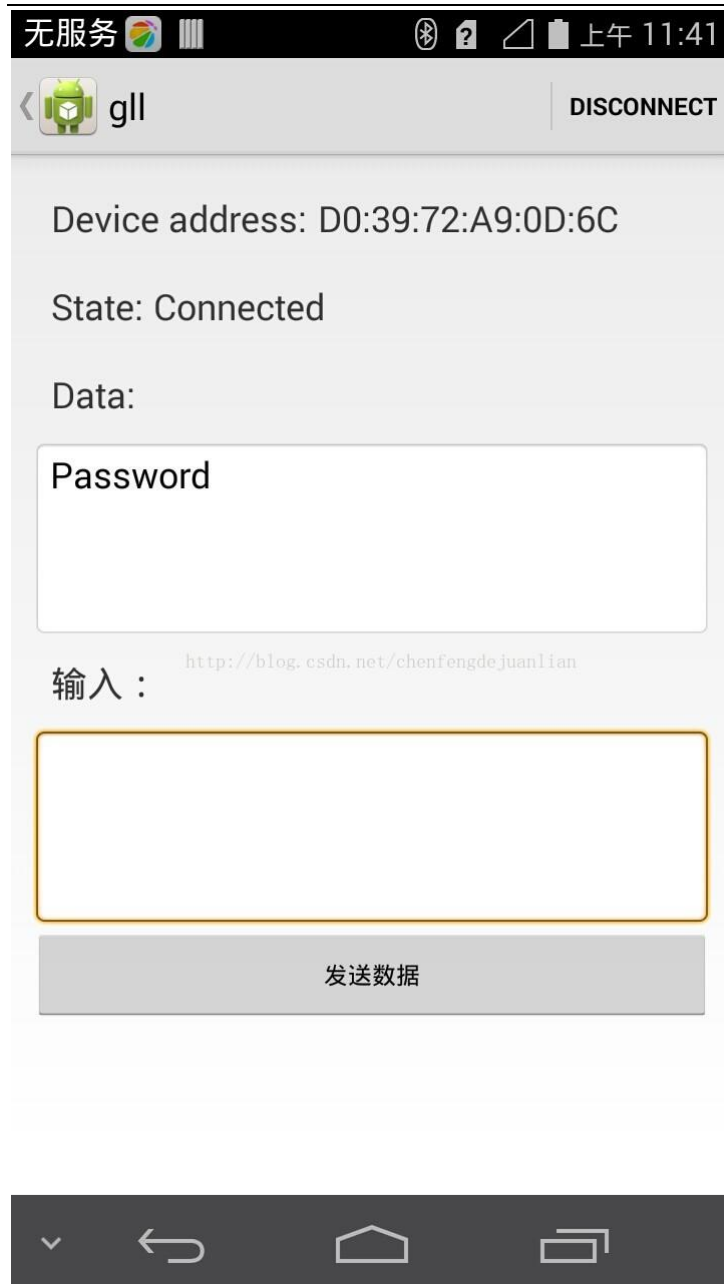
广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808





深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808





深圳市威汉德科技有限公司

广东省深圳市龙岗区同乐社区园新路 13 号永诺二楼
电话: +86-0755-27388751 传真: +86-0755-27388751-808
