基于 Qt 的便携式心电监护仪 GUI 应用软件设计①

喻 晓[1], 夏 澎[2]

[1](佛山职业技术学院 电子信息系, 佛山 528137)
[2](广州市保伦电子有限公司, 广州 511400)

摘 要: 便携式心电监护设备是现代化医院、社区、家庭不可缺少的医疗设备. 本文采用图形界面 GUI 开发软件 Qtopia Core 设计了操作简单、人机友好的心电监护系统界面, 该人机交互界面可实现心电数据的采集、处理、显示、分析、存储以及传输等功能. 最后通过实验, 验证了该系统的有效性.

关键词: 心电监护; QRS 波检测; Qtopia Core

Design of GUI for ECG monitoring System Based on Qtopia Core

YU Xiao[1], XIA Peng[2]

[1](Department of Electronic and Information, Foshan Polytechnic, Foshan 528137, China)
[2](Guangzhou ITC Electronic Co. Ltd., Guangzhou 511400, China)

Abstract: The ECG monitor is an indispensable medical instrument for modern hospitals, communities and families. In this paper, GUI development software Qtopia Core is used to design a simple and human-friendly interface for ECG monitoring system. This interface is realized the functions of data acquisition, processing, display, analysis, storage and transmission of ECG data etc. Finally, the experiments verify the effectiveness of the system.

Key words: ECG monitoring; QRS wave detection; Qtopia Core

心电监护设备能够对病人进行长时间的、连续的自动实时监护, 并且能够提供多种心电参数的测量和分析, 有利于医生及时了解患者的病情变化, 是现代化医院、社区、家庭不可缺少的医疗设备[1]. 随着科技水平的不断提高, 人们对该设备的操控性也提出了更高的要求. 为方便用户使用, 设计具有良好图形用户接口(Graphical User Interface, GUI)的仪器设备将是未来的发展方向.

现代操作系统一般都提供图形化的操作界面, 这种 GUI 系统一般由视窗、图标、菜单、对话框及其他一些可视特征组成, 它允许终端用户方面的利用鼠标和键盘操作电脑. 目前, 便携式心电监护仪所使用的操作系统大多是都是基于嵌入式 Linux 操作系统, 根据嵌入式 Linux 系统的硬件条件, 需要选择一个高性能、轻量级的 GUI 系统. 目前常见的面向嵌入式 Linux 的 GUI 系统主要有 Qtopia Core、Micro Windows、Tiny X 以及国内的 MiniGUI 等. 其中, Qtopia Core 是挪威 Trolltech 公司发布的一款基于嵌入式 Linux 的面向单一应用的嵌入式产品的 Qt 开发平台, 它的前身是 Qt/Embedded(常简称为 Qt/E)[2]. Qt 是一种 C++跨平台开发工具, 具有一次编程, 多平台运行的优点.

因此, 本文采用 GUI 开发软件 Qtopia Core 设计了操作简单、人机友好的心电监护系统界面, 该人机交互界面可实现心电数据的采集、处理、显示、分析、存储以及传输等功能.

1 心电监护系统应用软件设计

根据对便携式心电监护系统的需求分析, 可以将该系统分为五个功能模块, 其结构图如图 1 所示. 分别是主控模块、参数设置定模块、实时任务处理模块、数据存储模块和传输通信模块. 各功能模块在各自独立完成系统所分配任务的同时又可以通过系统的框架协议进行联系.

① 收稿时间:2016-09-06;收到修改稿时间:2016-10-19 [doi:10.15888/j.cnki.csa.005766]

© 中国科学院软件研究所 http://www.c-s-a.org.cn

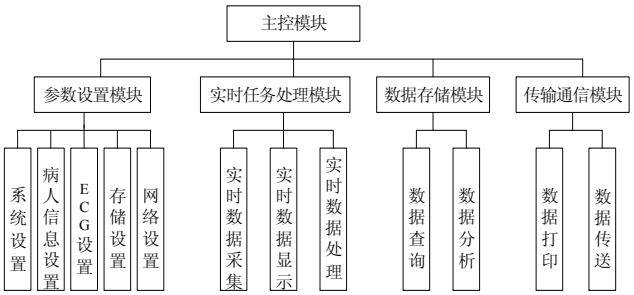

图 1　功能模块结构图

2　心电监护系统功能模块设计

2.1 主控模块

主控模块的设计目的是为了使各模块在系统的框架下协调工作和相互通信, 以实现人机交互.

当监护系统上电初始化后, 显示监护系统主界面. 在主控模块上设计了各类功能按钮, 通过点击各个功能按钮, 主控模块对按钮的动作进行响应, 并进行调度. 通过 QT 的信号与槽机制, 主控模块和各个功能模块实现通信. 通过 QT 的多线程编程, 实现多个实时任务同时进行. 本文设计的心电监护系统主界面如图 2 所示.

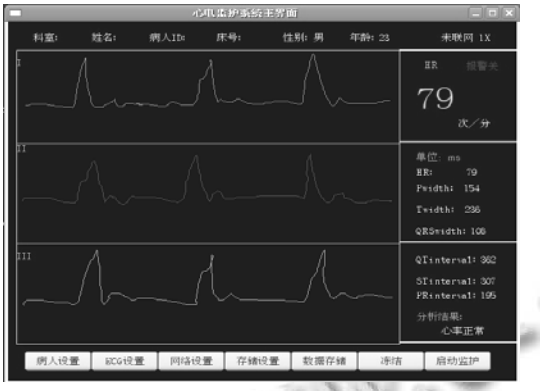

图 2　Web 信息抽取流程

2.2 参数设置模块

参数设置模块用于设置监护仪的一些必要参数和病人的相关信息. 本系统将参数设置部分分为四个类别: 病人信息设置、ECG 设置、网络设置、存储设置.

为录入及共享住院病人信息, 本系统设计了病人信息设置这一项. 病人信息设置包括入院科室、病人病历号、床号、姓名、性别、年龄、主治医生、身高体重、血型以及入院时间、出生日期, 病人类别等信息. 病人信息设置窗口如图 3 所示.

图 3　病人信息设置窗口

ECG 设置包括导联设置和报警设置. 可以选择 3 导联或者 5 导联, 在设置中选择相应的导联. 同时还可以进行报警级别设置、心率报警上下限的设置、波形滚动速度的设置. ECG 设置窗口如图 4 所示.

图 4　ECG 设置窗口

为了实现心电监护设备的网络化, 实现远程心电监护, 就必须要求监护终端和中央监护中心能够进行通信, 因此在通信之前, 必须进行网络设置. 网络的设置主要包括终端 IP 的设置、监护中心的 IP 设置、网关、子网掩码的设置.

存储设置主要是设置存储位置, 可以将心电数据存在系统的NAND FLASH 里, 也可以将其存储在SD卡里. 网络设置和存储设置的界面分别如图 5、图 6 所示.

图 5　网络设置窗口

图 6　存储设置窗口

2.3 实时任务模块

实时任务模块是本系统设计中的一个重点和难点.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

当心电监护系统完成初始化设置进入工作状态后, 实时任务模块将实现对心电信号的采集、波形的显示以及心电数据的分析, 如进行 QRS 波群的检测, 提取心电特征信息, 将分析结果在显示终端上进行呈现. 因此, 实时任务模块主要包括实时数据的采集、实时数据的显示和实时数据的分析三个部分. 本系统中分别通过三个线程来完成数据的采集、显示和分析.

首先通过实时采集线程采集心电信号, 从 A/D 转换器得到三通道的数据并存入数组中. 再经过平滑滤波, 滤除 50HZ 的工频干扰, 显示光滑的心电波形. 当数组中存储了 5s 的数据时, 通过心电检测线程检测 QRS 波群, 提取出心电特征信息, 并在系统界面上显示分析结果. 本系统主要完成心率的提取计算以及心率失常分析. 实时任务模块的流程图如图 7 所示.

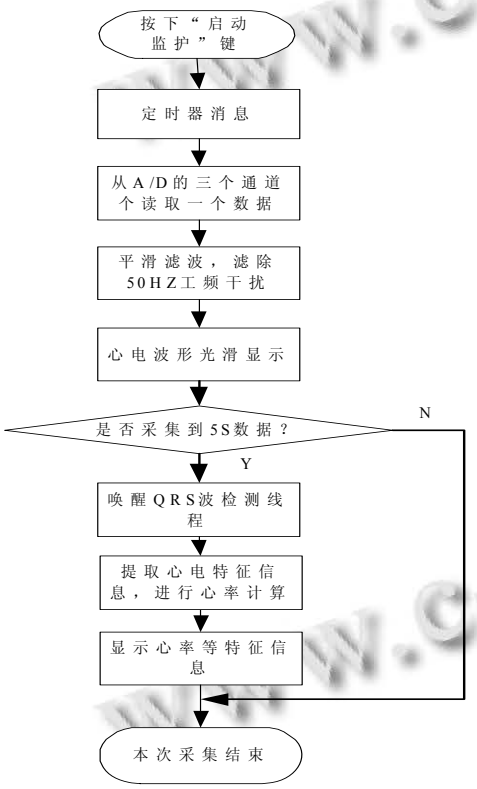

图 7 实时任务模块流程图

2.3.1 实时任务模块流程图

Qt 的应用程序提供了一种非常方便的定时机制, 本系统心电信号采集就是利用该定时机制实现定时采样. 定时器遵循信号与槽的机制, 当定时器信号到期时, 触发槽函数运行. 本系统在主界面 main_widget 类中定义了一个定时器 display_ caiji_timer 和连接函数, 为了使采样频率达到 200HZ, 定义定时器的时间间隔为 5ms, 即每隔 5ms 触发一次槽函数 display_caiji()代码如下:

```
QTimer *display_caiji_timer=new QTimer(this, "display_caiji_timer");
connect(display_caiji_timer,SIGNAL(timeout()),this,SLOT(display_caiji()));
display_caiji_timer->start(5);
```

2.3.2 心电信号的显示

心电信号的显示包括心电数据的数值显示和心电的波形显示.

1) 心电数据的数值显示

通过心电分析线程检测出 QRS 波群、P 波、T 波等, 提取出心电信号的特征信息, 即为心电数据, 其包括心率、P 波宽度、T 波宽度、QRS 波宽度、QT 间期、ST 间期、PR 间期、ST 偏移量. 系统主要完成心率的提取计算.

本系统设计了一个 Dataxd 类来实现心电数据的更新显示. 为了达到实时更新显示心电特征数据, 在显示界面 Data_widget 类中, 定义了一个定时器 display_ data_timer 和连接函数, 且定义定时器的时间间隔是 5s, 即每隔 5s 触发一次槽函数, 槽函数 display_data()代码如下:

```
QTimer *display_data_timer=new QTimer(this, "display_data_timer");
connect(display_data_timer,SIGNAL(timeout()),this,SLOT(display_data()));
display_data_timer->start(5000);
```

每当槽函数 display_data()被触发时, 首先调用 qrs(int data), 将采集到的数据按照 qrs(int data)中的心率计算公式, 计算得到心率 data1, 然后调用 Qt 里的库函数 display(int),将心率结果显示出来. 不断循环定时就能将数据正常显示.

2) 心电的波形显示

心电波形的显示主要通过绘图的方式将心电数据以波形的方式在系统界面上显示出来. 心电波形的绘图方案有两种, 即移屏式与扫描式[3], 本系统中采用扫描式.

本系统设计了一个 wave 类来实现在系统界面上绘制波形的功能, 该类是继承 Qt 中的窗口组件类 QWidget. 在绘制波形时, 也是采用定时器定时机制.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

在主界面 main_widget 类中定义了一个定时器 display_curve_timer 和连接函数, 且定义定时器的时间间隔为 10ms, 即每隔 10ms 触发一次槽函数 display_wave()代码如下:

```
QTimer *display_wave_timer=new QTimer(this, "display_wave_timer");
connect(display_wave_timer,SIGNAL(timeout()),this,SLOT(display_wave()));
display_wave_timer->start(10);
```

槽函数 display_wave()是 win_wave 类的成员函数, 在 display_wave()中调用了 wave 对象的成员函数 draw (int data)来画波形, 而 draw(int data)先调用 hmap(int data), 采集到数据通过 hmap(int data)中的公式转换后赋值给 data1, 然后调用 drawbar()绘制扫描线, 调用 change_t()将扫描线的横坐标加 1, 用语句 p.drawLine (x+t-1, y+hmap(data0)-1, x+t, y+Hmap(data1)-1); 和 change_t()将当前点与前一点直线相连, 不断循环这一过程, 就形成了显示屏上的动态波形.

2.3.3 心电数据的分析

心电信号基本上都包括一个 P 波、一个 QRS 波群和一个 T 波, 有时在 T 波之后还出现一个小的 U 波[4], 如图 8 所示.

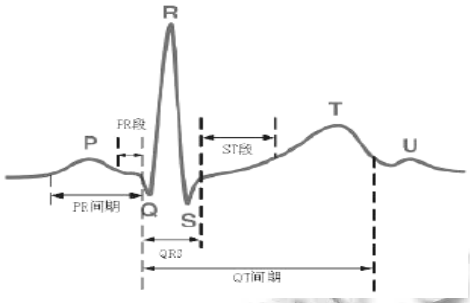

图 8　典型的心电波形图

心电监护系统主要是通过心电电极采集心电信号, 经过放大, 滤波, 送入心电处理器系统内, 然后实时显示出心电波形. QRS 波检测是提取心电信号的特征信息, 供医护人员进行心电分析的首要问题. 分析计算的特征参数包括: 心率(HR)、P 波宽度(PWidth)、T 波宽度(TWidth)、QRS 波宽度(QRSWidth)、QT 间期(QTInterval)、ST 间期(STInterval)、PR 间期(PRInterval)、ST 偏移量. 鉴于篇幅, 本文主要介绍 QRS 波群的检测和 HR 的提取计算, 其余参数如 T 波、P 波宽度分析计算可参考相关文献[5].

利用 R 波检测算法可以检测到 R 波幅值的极大值点, 利用这些极大值点找出所测量的心电波形的周期, 选取一个周期进行判断, 找出一些特征点, 如 QRS 的长度, P 波、T 波的宽度、R_R 间期等, 根据所得到的特征值判断病人的健康状况. 其中, 瞬时心率 HR 的计算公式如式(1).

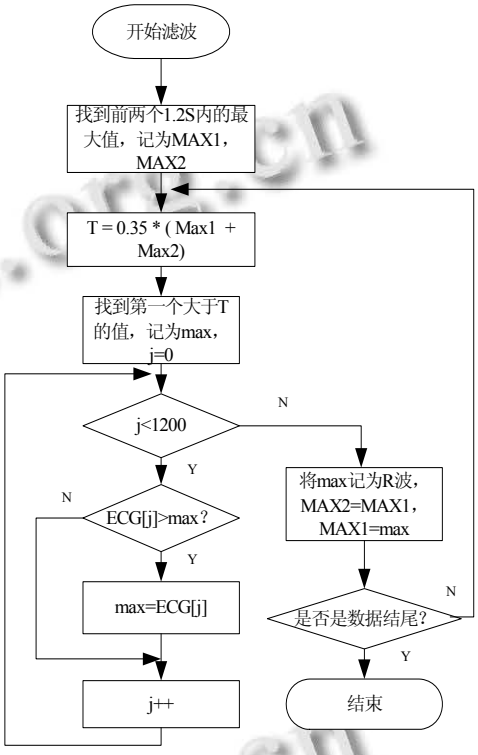

图 9　R 波检测流程图

$$HR=\frac{60\times Sample\_rate}{R\_R} \tag{1}$$

根据上面的公式就可以算出瞬时心率 HR, 式中 Sample_rate 表示心电信号采样频率, R_R 间期为两个相邻 R 波之间的间隔时间. 通过 QRS 波的检测, 就可以得到 R_R 间期, 取相邻四个 R_R 间期的平均值作为 R_R 间期. 计算得到 HR 后, 就可以判断心率是正常、过快还是过慢, 以提供给医生分析参考.

2.3.4 数据存储和传输打印模块

将数据采集并处理完毕后, 需将其保存以供分析及打印. 本文设计了数据存储模块, 以实现自动保存心电数据的结果, 从而实现心电数据的查询回放以及数据分析.

程序中定义 stru_termidatat 结构体, 用于保存数据. 当数据经过采集处理并完成显示后, 触发数据保存函

© 中国科学院软件研究所 http://www.c-s-a.org.cn

数，将最终检测结果的数值保存起来. 以上数据是存储在 NANDFLASH 中，本系统心电数据还可以存储在 SD 卡中. SD 卡具有体积小、容量大、价格低廉、可携带的特点，因此，使用 SD 卡，可以存储大量的心电数据，更有利于医生分析病人心电的变化.

按下“存储”按钮，系统会按照存储设置进行存储. 当设置为 SD 卡存储时，插入 SD 卡，系统会自动识别并挂载 SD 卡，在 SD 卡里创建并打开数据文件，并将心电数据存储在文件里面. 心电数据存储的流程图如图 10 所示.

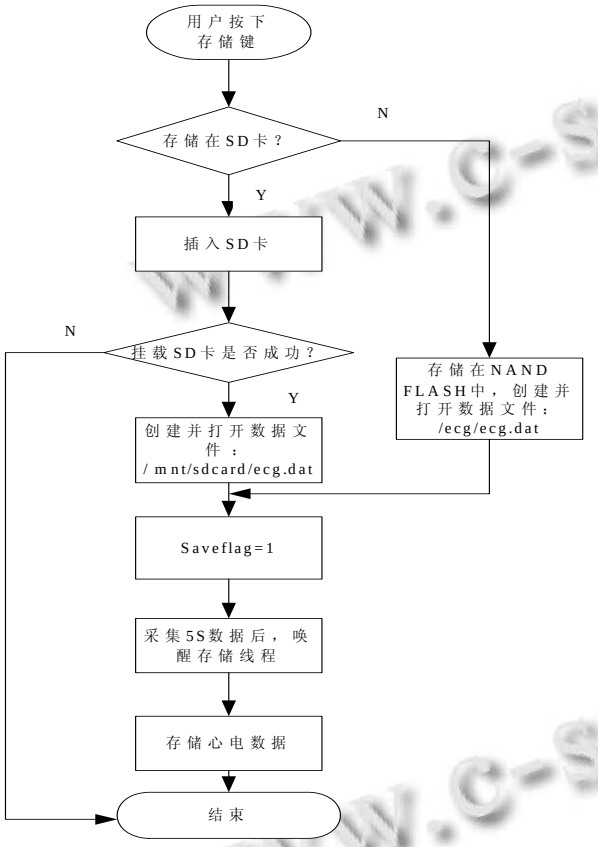

图 10 心电数据存储的流程图

在实现数据的存储之后，用户可根据情况对其进行打印输出或者通过串口、网线等形式进行数据的传输. 本系统设计了传输通信模块，以完成打印功能.

设计中，将数据的输出接在异步串行接口 UART 上，其实质就是实现主控制平台与打印机或者外部系统之间的串口通信. 系统在串口传输程序中，采用了中断方式，来实现双向数据传输，达到实时控制的目的. 串口程序数据发送过程为：将保存的检测数据结果调出，并以数组的方式放置在 SDRAM 中，然后再将调用 Uart_Setdata()函数发送字符，用以完成数据的传输工作. 串口打印数据流程图如图 11 所示.

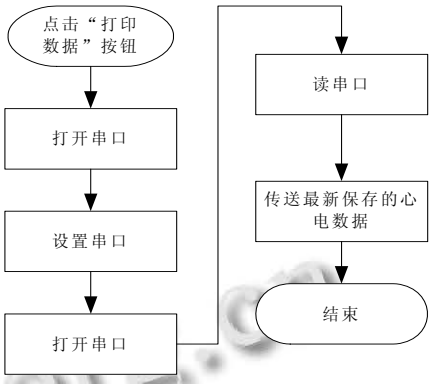

图 11 串口打印数据流程图

3 实验测试与结果分析

为了检验本系统的准确性，在实验阶段，采用模拟心电信号发生器产生各种心电信号，用本系统进行采集. 图 12 为本心电监护系统运行的硬件平台结构图.

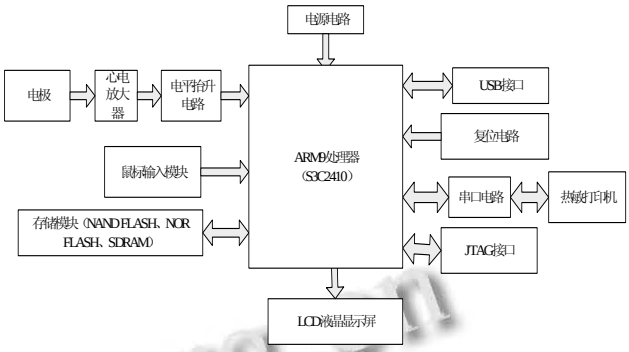

图 12 心电监护系统硬件平台结构图

开机后，进入心电监护系统主界面，点击监护界面上的“启动监护”键，并打开模拟心电发生器，设置心率为 79 次/分钟，幅度为 1mv，系统开始采集心电信号并在监护界面上显示心电波形和心电信号的特征参数，此时显示心率为 79 次/分，并显示心率正常. 如图 13 所示.

图 13 心率正常显示图

© 中国科学院软件研究所 http://www.c-s-a.org.cn

改变模拟心电发生器上的心电心率的频率，设置为 40 次/分钟，查看心电监护系统的监护界面，此时，心率栏显示 40 次/分，并在分析结果栏显示心率过慢. 如图 14 所示.

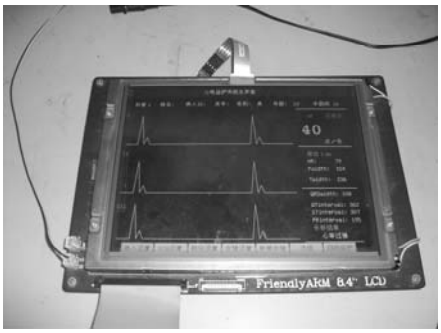

图 14　心率过慢显示图

接着再次改变模拟心电发生器上的心电心率的频率，设置为 144 次/分钟，观察监护界面波形的变化情况，此时心率显示 142 次/分，并显示心率过快. 如图 15 所示.

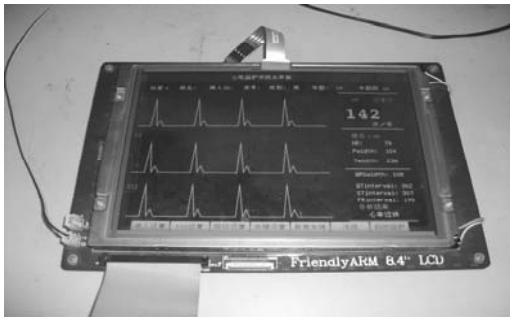

图 15　心率过快显示图

按照上面的实验进行 10 次，得到 10 组心电率数据，如表 1 所示. 将监护系统上显示的心率与模拟心电发生器的心率进行比较，发现系统检测得到的心率与产生的心率基本一致.

表 1　心率检测结果

序号 / 心率	1	2	3	4	5	6	7	8	9	10
检测值	92	59	112	64	85	153	46	52	49	78
实际值	92	58	112	63	84	151	46	52	48	76

同时，还对本系统的其他功能进行了测试. 测试结果表明本系统能够正常设置各种参数，进行正常存储数据、报警、冻结、解冻当前心电界面，并且能够长时间稳定、准确的进行监护. 通过以上的实验，证明本系统基本实现了设计的要求，能够进行正常、可靠的心电波形显示.

4　结语

本文中心电监护系统的实验测试，是通过采集模拟心电发生器产生的心电信号来进行. 本系统采用的 QRS 检测算法能够准确检测到 QRS 波群并进行心率的提取，基本实现了该系统的功能. 从实验的效果看，该系统工作性能良好，运行稳定. 实验的结果证明了系统设计的准确性和可靠性，有一定的实用价值，而且适应性较好，易于推广应用.

参考文献

1 黄澍涛.医用监护仪的临床应用与发展特点.中国医疗设备,2011,(2):59–61,156.
2 刘伟民,韩斌,李征.基于 Linux 的数据采集及在 QT 界面的显示.微计算机信息,2008,24(8-1):97–99
3 刘莉.多参数监护仪的研制[硕士学位论文].桂林:桂林电子科技大学,2007.
4 姚欢,王建钢.ECG 信号 QRS 波群检测算法的进展.现代生物医学进展,2012,12(20):3988–3991
5 杨守祥.基于小波变换的心电信号预处理及特征参数检测方法研究[硕士学位论文].兰州:兰州理工大学,2014.

© 中国科学院软件研究所 http://www.c-s-a.org.cn