盲人避障设备

详细设计文档—避障数据滤波

叶志活

2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 时间 | 内容 | 编者 | 备注 |
| V1.0 | 2018/12/1 | 初稿 | 叶志活 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 功能设计 1](#_Toc531457783)

[1.1 功能描述 1](#_Toc531457784)

[1.2 活动流程 1](#_Toc531457785)

[2 详细设计 2](#_Toc531457786)

[2.1 初始化功能 2](#_Toc531457787)

[2.1.1 滤波器初始化 2](#_Toc531457788)

[2.1.2 输入捕获定时器初始化 2](#_Toc531457789)

[2.1.3 普通定时器初始化 2](#_Toc531457790)

[2.1.4 引脚初始化 3](#_Toc531457791)

[2.2 原始数据捕获 3](#_Toc531457792)

[2.3 数据处理 5](#_Toc531457793)

[2.3.1 滤波处理 5](#_Toc531457794)

[2.3.2 多波结合 6](#_Toc531457795)

[2.4 障碍物判断 6](#_Toc531457796)

[2.4.1 语音提示 6](#_Toc531457797)

[3 数据结构 7](#_Toc531457798)

[3.1.1 宏定义 7](#_Toc531457799)

[3.1.2 结构体 7](#_Toc531457800)

[3.1.3 全局变量 7](#_Toc531457801)

[4 附录 8](#_Toc531457802)

[4.1 源文件 8](#_Toc531457803)

# 功能设计

## 功能描述

障碍物提示功能用于向盲人提示前方障碍物信息。提示方式包括语音提示、震动提示、频率提示三种。

系统首先通过位于眼镜合拐杖上的超声波模块获取障碍物距离数据，依据所得距离数据再结合当前所采用的提示方式进行相应处理和提示。

语音提示：系统判断障碍物距离大小，若障碍物距离小于给定阈值，则根据对应的超声波位置触发提示，提示包含障碍物位置信息。

震动/频率提示：震动/频率提示工作方式大致相同，系统将障碍物距离进行分级，距离越近，级别越高，对应频率越高。该提示方式包含障碍物距离信息。

## 活动流程

障碍物检测功能分为初始化、数据捕获、数据处理、障碍物判断四个流程。

初始化：系统首先初始化与障碍物有关的硬件资源以及软件资源，包括定时器、引脚、滤波器初始状态。

数据捕获：数据捕获用来获取原始的障碍物距离数据，系统以定时器输入捕获的方式获取超声波传感器的电平信号，再将电平信号转换为距离信号。

数据处理：系统获得的原始数据存在较大毛刺和波动，因此需要对原始数据进行滤波处理，以此提高数据准确率。数据处理方式包括：卡尔曼滤波、多波结合。

障碍物判断：系统根据处理后的数据判断盲人前方障碍物距离，当障碍物距离小于阈值时，触发相应提示。



图 1‑1 障碍物检测流程



图 1‑2 系统功能结构

# 详细设计

## 初始化功能

系统开始监测障碍物前，需要对监测障碍物功能所需要使用的所有软硬件资源进行初始化，包括滤波器的初始状态、用于捕捉信号的输入捕获定时器、触发测距请求的普通定时器、输入输出引脚。

### 滤波器初始化

系统在对数据进行处理前，需要设置滤波器初始状态。当前系统采用两种滤波方式：卡尔曼滤波、平均求和（未完善）。

卡尔曼滤波初始化：设置滤波器的方差、误差以及其他参数（需要理解卡尔曼滤波原理）

平均滤波初始化：设置需要处理的数据数量size（每个超声波传感器所采集的数据对应一组数据），设置每组数据所记录的历史数据容量degree（即该组数据对应的传感器前degree次采集的历史数据），通过对前degree次历史数据求平均来作为当前时间的最终数据。

### 输入捕获定时器初始化

系统通过定时器输入捕获的方式来捕获超声波传感器电平信号，在进行信号捕获前，需要定时器进行初始化。

初始化时需要设置定时器优先级，输入捕获定时器优先级应尽可能的高，以避免被其他中断打断或者无法及时响应，从而导致信号捕捉错误。

将定时器设置为输入捕获模式，配置相应引脚及其中断。一个定时器最多对应4个捕获通道，因此眼镜上需要一个输入捕获定时器，拐杖上需要两个输入捕获定时器。

### 普通定时器初始化

系统检测障碍物时，需要先向超声波传感器发送测距请求信号以触发测距。超声波传感器每次测距之间时间间隔应不少于60ms，以避免余波干扰。

系统通过普通定时器来控制每次测距的时间间隔，系统需要为普通定时器设置足够长的中断时间，当定时器进入中断时，系统进行一次测距。

### 引脚初始化

初始化超声波传感器对应的输入输出引脚，用于输出或捕获电平信号。

## 原始数据捕获

系统完成初始化后开始数据捕获，其获取流程如下：

**眼镜**

1. 系统进入定时中断
2. 向传感器发送高电平信号，开始测距
3. 向拐杖发送测距请求，请求获取拐杖数据
4. 获取传感器传回的电平信号
5. 将电平信号转化为距离数据
6. 接收拐杖传回的距离数据
7. 记录眼镜和拐杖的原始距离数据。

**拐杖**

1. 系统进入定时中断
2. 收到测距请求，开始获取数据
3. 向传感器发送高电平信号，开始测距
4. 获取传感器传回的电平信号
5. 将电平信号转化为距离数据
6. 将数据传回眼镜
7. 等待下一次测距请求。

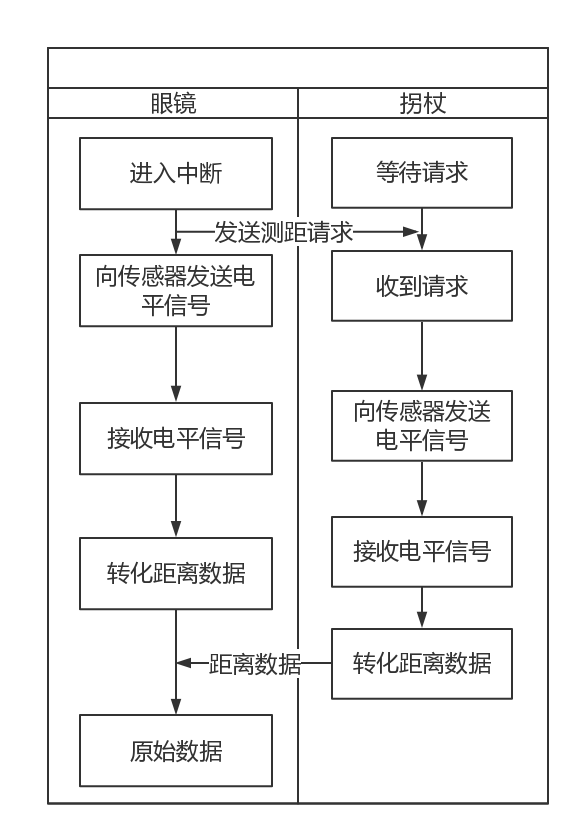


图 2‑1 数据获取流程

**定时器中断**

系统进入定时中断后，有两种处理流程：发送测距信号和测距请求；对接收到的原始数据进行下一步处理。

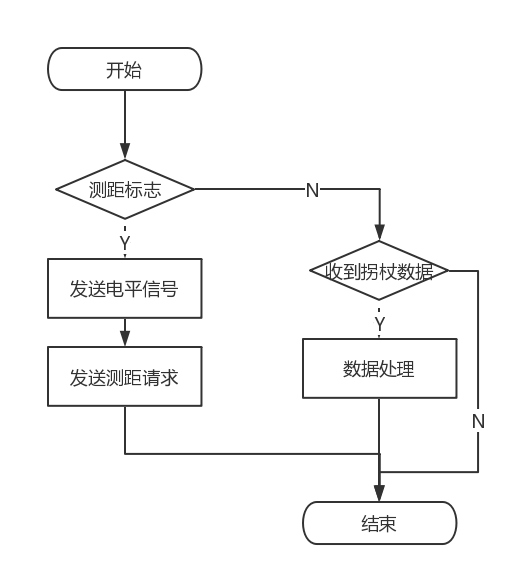


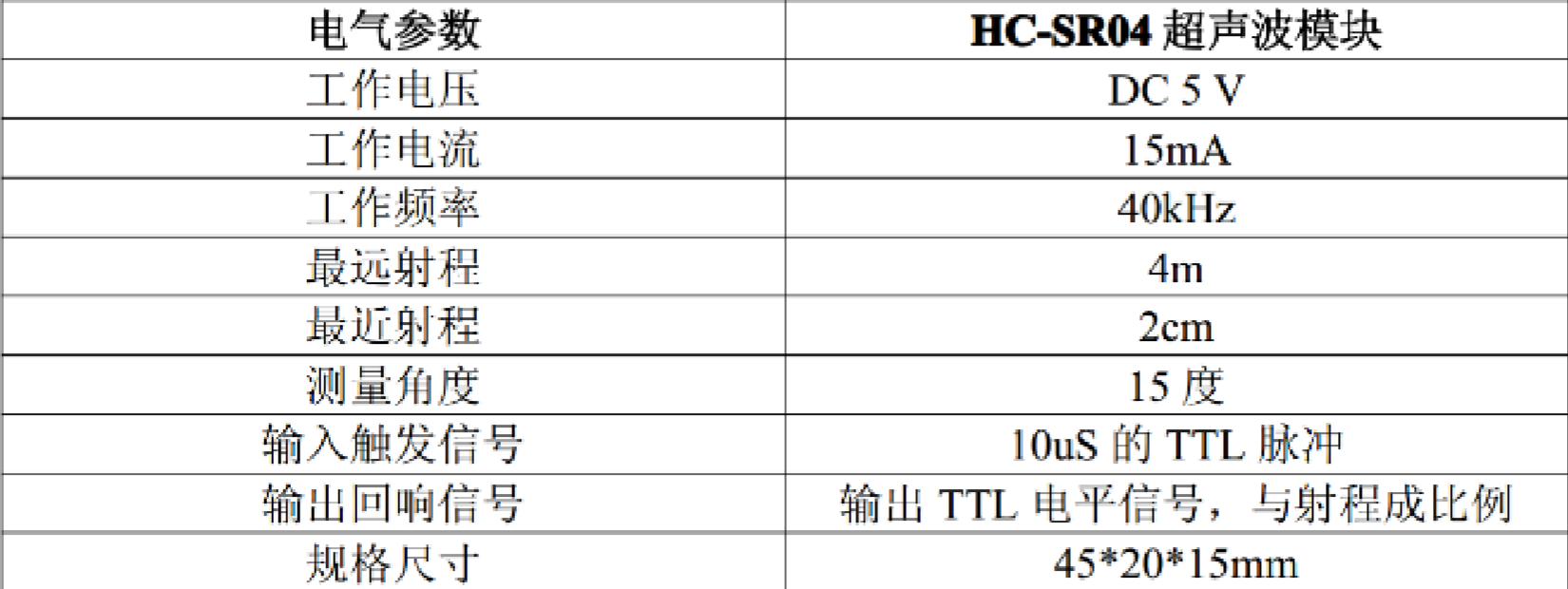
图 2‑2 定时器处理流程

**超声波传感器工作原理**

输入信号：<10ms高电平

输出信号：连续的高电平信号，距离数据=声速\*信号时间/2。

HC-SR04 超声波测距模块可提供2cm400cm 的非接触式距离感测功能，测距精度可达高到 3mm；模块包括超声波发射器、接收器与控制电路。



详细见模块手册。

**接收电平信号**

系统通过输入捕获定时器捕获传感器高电平信号时间。当输入捕获定时器进入中断时，首先判断是否是定时周期溢出产生更新中断，若是则将输入捕获时间将上定时器周期时间；否则，判断哪个通道触发中断，在针对相应中断进行相应处理。

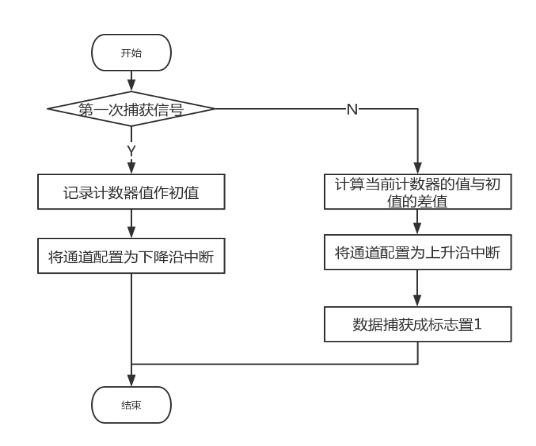


图 2‑3 输入捕获流程

## 数据处理

系统接收到原始数据后，数据进行滤波处理和多波结合，提高数据准确率。

### 滤波处理

1. 卡尔曼滤波处理

将传感器获取的原始距离数据进行滤波处理，剔除其中的毛刺，使得数据趋于平滑。（理解卡尔曼滤波原理）

每个模块采集到原始数据后，对数据进行优化处理，去除噪音。其处理方式如下：

其中X(k|k-1)是利用上一状态预测的结果，X(k-1|k-1)是上一状态最优的结果，A为矩阵参数，其中，为k时刻障碍物与用户间的最优距离，v为用户行走速度，，dt为超声波两次测距间的时间间隔；P(k|k-1)是X(k|k-1)对应的协方差，是X(k-1|k-1)对应的协方差，A’表示A的转置矩阵，Q为预测方差。 R为测量方差，Z(k)为k时刻从超声波模块获得的原始距离数据，X(k|k)为k时刻的最优估算，即最终优化得出的数据，P(k|k)为k时刻对应的协方差,I为单位矩阵，H=|1 0|；

获得优化数据后，对综合所有模块数据分析障碍物位置并给用户做出相应提示。

1. 处理方案二

记录传感器的前n次数据，判断n次数据是否均小于阈值，若是则判断存在障碍物。

### 多波结合

对滤波数据进行进一步处理，将多个传感器数据进行融合，从而获得融合后的数据，融合数据中，各传感器数据所占比重不同，所占比重取决于数据稳定性。（未完成）

方案一：

将同一水平位置的两个超声波数据做综合判断，当两个传感器数据均小于阈值时才判定存在障碍物。

方案二：（未实现）

将同一水平位置的两个超声波数据做融合处理，融合处理公式如下：

其中s0为最终融合数据，s1和s2分别为左右两个传感器的距离数据，a，b为权重系数。

由于检测角度和传感器自身原因，不同传感器之间获取的数据可靠度存在差异，可靠性越大则认为传感器数据越准确。系统应优先考虑可靠性大的数据，在以上公式中，可靠性越大，权重系数越大。

权重系数可通过传感器的数据稳定度计算，即一段时间内，数据越趋于平滑，则可靠性越高，系数越大。稳定度判断可以通过数据变化速率、方差、标准差等方式实现（未实现）

## 障碍物判断

获取眼镜和拐杖上所有超声波模块的优化数据后，综合所有数据判断障碍物信息，包括障碍物大致位置和距离。障碍物位置通过超声波模块的检测范围来判定，超声波模块在竖直位置上呈线性分布，同一水平面包含两个超声波模块，两个模块的检测范围部分重合，不同水平面间的模块的检测范围不重合。

### 语音提示

根据超声波检测范围可以判断障碍物位于用户的正前方、左前方和右前方（未实现），同时可以判断障碍物是否会碰撞到用户的头部、手腕以及脚下。其具体检测过程如下：

通过判断物体与用户的距离是否小于阈值来判断用户前方的物体是否会成为阻碍用户前进障碍物，用户可以自由修改阈值以适应不同环境的需求。当某个超声波模块检测到障碍物时，通过该模块的检测范围来判断障碍物所处位置。

当多个超声波同时检测到障碍物时，通过分析超声波检测范围以及检测到的距离判断是否属于同一个障碍物，若属于同一个障碍物则对检测到的所有数据取均值作为障碍物实际距离，若判断为不属于同一个障碍物则向用户提示距离最近的障碍物信息。

1. 水平位置判断（未实现）

障碍物的水平位置通过位于同一个水平面的超声波模块确定，当只有某模块检测障碍物时，根据模块所处位置判断障碍物位于用户的左前方或有前方；当两个模块同时检测到障碍物时则判定障碍物位于用户正前方。

1. 竖直位置判断

障碍物的竖直位置通过位于不同水平面的模块判断。当眼镜上的模块检测到障碍物时，提示用户小心头部。当拐杖上的模块检测到障碍物时，若处于拐杖上端则提示用户小心手腕；若处于拐杖下端，则提示用户小心脚下。

频率/震动提示

频率/震动提示模式下，系统取所有超声波传感器数据中的最小值作为盲人前方障碍物距离。系统将障碍物距离分为三个等级：0-2。0：障碍物距离>2米；1：障碍物距离<2米，>1米；2：障碍物距离<1米。

# 数据结构

### 宏定义

|  |  |  |
| --- | --- | --- |
| ONLY\_GLASS | 眼镜单独测试标志，当需要单独测试眼睛时使用 | |
| AVER\_NUM\_GLASS | 2 | 眼睛上超声波数量 |
| AVER\_NUM\_WALK | 5 | 拐杖上超声波数量 |
| AVER\_NUM\_ALL | 7 | 超声波总数量 |
| LATE\_NUM | 3 | 保留最近？次数据 |
| OBSTACLE\_HEAD | 3 | 头部障碍物标志 |
| OBSTACLE\_AHEAD | 2 | 前方障碍物标志 |
| OBSTACLE\_FOOT | 1 | 脚下障碍物标志 |
| OBSTACLE\_NO | 0 | 无障碍物 |

### 结构体

|  |
| --- |
| typedef struct  { uint8\_t Capture\_FinishFlag; // 捕获结束标志位  uint8\_t Capture\_StartFlag; // 捕获开始标志位  uint16\_t Capture\_CcrValue; // 两次捕获寄存器的值之差  uint8\_t Capture\_CCx; //通道标记  }TIM\_ICUserValueTypeDef; |
| typedef struct{  KalmanType filterValue[2]; //k-1时刻的滤波值，即是k-1时刻的值,第一个值表示距离，第二个值表示速度  double kg[2]; // Kalamn增益  double Q[2]; //预测过程噪声偏差的方差, Q[0]距离估计误差 Q[1]速度估计误差  double R; //测量噪声偏差，(系统搭建好以后，通过测量统计实验获得)  double P[2][2]; //估计误差协方差  double dt; //时间分量  } KalmanInfo; // 一维滤波器信息结构体 |

### 全局变量

|  |  |  |  |
| --- | --- | --- | --- |
| 变量 | 类型 | 初值 | 含义 |
| UltrasonicWave\_Distance\_Walk[AVER\_NUM\_WALK] | Int | {500,500,500,500,500}; | //拐杖采集数据 |
| MODE\_FLAG | Int | 1 | 1：语音  0：频率  2：震动 |
| MEASURE\_FLAG | Int8\_t | 1 | 1眼镜采集数据，  0等待拐杖采集数据 |
| GET\_WALK\_FLAG | Int8\_t | 0 | 接收拐杖数据标志 |
| UltrasonicWave\_Distance[AVER\_NUM\_GLASS]; | Int |  | 存储距离数据 |
| MAX\_DISTACE | Int16\_t | 150 | 障碍物距离阈值 |
| lateobstacle[AVER\_NUM\_WALK+AVER\_NUM\_GLASS] | Int8\_t | ｛0｝ | 记录最近几次测距障碍物状态，连续监测障碍物时+1，  未监测到障碍物时清零 |
| Kalman[AVER\_NUM\_ALL] | KalmanInfo |  | 记录卡尔曼滤波数据 |

# 附录

## 源文件

|  |  |
| --- | --- |
| UltraConfig.c | 初始化相关函数（定时器、引脚） |
| UlsonicWave.c | 数据获取、障碍物判断 |
| Kalman.c | 卡尔曼滤波 |
| Filter.c | 其他滤波方式 |
| Stm32f10x\_it.c | 定时器中断函数 |