

嵌入式 Linux 下移动机器人超声波测距系统*

单承刚

(枣庄学院 计算机科学系, 山东 枣庄 277160)

摘要: 设计了基于 S3C2410+Linux 平台的移动机器人超声波避障系统。介绍了 SRF05 新型高精度传感器的工作原理、测距系统多路超声波传感器的软硬件实现及机器人平台的整个软件工作流程。Linux 系统超声波驱动模块中采用的轮循工作方式实现了机器人在未知环境下的避障功能, 机器人平台运行效果良好。

关键词: 机器人; ARM9; 超声波; 导航; 避障

中图分类号: TP274 **文献标识码:** A

Ultrasonic measuring system of mobile robot in embedded Linux system

SHAN Cheng Gang

(Computer and Science Department, Zaozhuang College, Zaozhuang 277160, China)

Abstract: This paper designed the system of unknown environment obstacle avoidance and navigation for wheeled mobile robot based on S3C2410 and embedded Linux system. This paper focuses on the high precision ultrasonic sensor principle, the realization of measuring system software and hardware and the software flow of mobile robot. Multi-ultrasonic measuring system is realized by circulation in embedded Linux system driver modules. This paper implements robotic monitoring and obstacle avoidance in uncertain circumstance with driver procedures and robotic platform system is to motion perfectly in uncertain circumstance.

Key words: robot; ARM9; ultrasonic; navigation; obstacle avoidance

在机器人的导航系统中, 超声波传感器以其信息处理简单、速度快和价格低等优点, 被广泛用作移动机器人的测距传感器, 以实现避障、定位和导航等功能。实验室机器人以 S3C2410+Linux 系统为架构, 采用了高精度的 SRF05 超声波传感器测距, 实现对机器人的导航避障。该传感器的回波反馈与测距方式与通用的超声波传感器有所区别, 在 S3C2410 系统中实现也稍有难度, 由于传感器精度可达到 1cm, 因此使用该传感器弥补了通用超声波传感器近距离测距存在盲点的问题, 节省了外围红外测距模块等硬件资源。

1 SRF05 传感器工作原理

S3C2410 微控制器利用 10 μ s 的高频脉冲触发 SRF05 传感器, 触发下传感器会发出 8 个 40 kHz 的超频脉冲, 当 SRF05 传感器的 ECHO 引脚电平为高时, 开启定时器

计数器计时; 当 ECHO 变为低电平时, 表明机器人遇到障碍物, 编程实现定时器停止计时。高脉冲宽度与测距距离有正比关系, SRF05 的测距有效范围为 1 cm~4 m, 无障碍物或遇超大障碍物时, ECHO 脚会变为低电平, 此过程需要时间为 30 ms, 设置定时器的周期比超声传感器探测最大距离所需的渡越时间稍长。因此, 在测距时定时器周期设置应至少大于 30 ms 才能有效测距。

2 系统方案设计

2.1 系统组成结构

轮式移动机器人核心控制系统由触摸屏显示、摄像头图像采集、超声波避障、直流电机闭环控制、人感应传感器、水(火)检测和气体检测模块、串口无线 Modem 模块部分组成。其中超声波避障模块直接影响机器人行进位移的精度以及在有效范围内的图像采集控制。

* 基金项目: 山东省教育厅科技计划项目(J07WJ29)

整个系统基于 ARM9 架构的 S3C2410 微控制器与 Linux 系统平台, 使用 Linux 系统驱动程序控制各个硬件模块^[1]。

2.2 超声波测距系统

为了实现导航、避障, 移动机器人的前、后、左、右各安装 1 路 SRF05 超声波传感器, 使机器人能够感知 4 个方位的障碍信息。另外, 为了满足系统扩展, 可添加安装多路 SRF05 超声波传感器, 进一步增强机器人的导航避障性能。安装超声波传感器之间角度应该大于它的波束角 30° , 这样传感器同时工作时才不会有干扰。采用分组的方式, 左右两套和前后两套分别为一组, 组与组之间采用轮循的方式工作。这样既可以得到很高的采集频率, 同时也满足了系统的实时性要求。

采用相关 GPIO 口作为超声波传感器的 INIT 触发引脚, 并且配置定时器周期, 设置定时中断及外部中断。测距流程: 开始打开超声波, 激发传感器 INIT 引脚 $10\mu\text{s}$, 打开定时器并开始计时, 在定时器周期内收到传感器 ECHO 低电平触发外部中断, 在外部中断处理程序中读取定时器计数值, 计算障碍物距离, 并且传送给中央处理器。不同组传感器的开启在定时中断中完成, 并且开启另一定时器完成计数功能^[2]。系统测距流程图如图 1 所示。

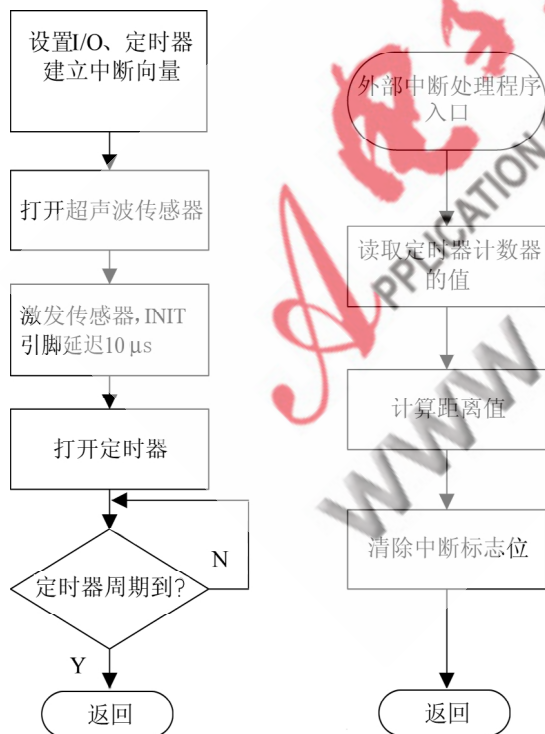


图 1 测距流程图

2.3 采集回波

S3C2410 GPIO 触发中断只能设置一种触发方式, 如果设为上升沿触发打开定时器计时, 则无法捕获下降沿, 也就无法关闭定时器, 因此采用软件方式实现。设

置 2 个时间延迟函数 `usdelay()` 和 `msdelay()`, 在给出 $10\mu\text{s}$ 高频脉冲后等超声波触发再使能定时器计时, 这些在该组传感器对应的定时器中断中完成。并设置外部中断 EINT1 接收超声波传感器的 ECHO 回波引脚的变化。其触发方式为下降沿触发, 等下降沿来临关闭定时器读出计数值即为脉宽的时间长度时, 然后根据距离公式 $= (\text{ECHO 高电平的时间}) \times (\text{声纳速度}) / 2$ 得出距离。空气中 V 与温度关系: $V = 331.5 \sqrt{1 + T/273} \text{ m/s}$, T 为摄氏温度^[3]。一般情况下超声波速近似声速, 在室内温度影响下取约 343.2 m/s 。

3 软件实现

3.1 延迟函数实现

设计软件延迟函数 `usdelay()`, 代码实现如下:

```
static int delayLoopCount1=10;
void usdelay(int time)
{ int i;
  for(;time>0;time--)
    for(i=0;i<delayLoopCount1;i++); }
```

函数延迟时间为 $10\mu\text{s}$, 打开传感器激发高频脉冲后延迟 $10\mu\text{s}$ 然后关脉冲, 即可利用该函数。

设计 `msdelay()` 函数用于衡量整个测距周期的时间值。`msdelay(1000)` 延迟大约为 1s , 这样系统可以有足够的回波检测时间, 同时也不会对其他多路超声波传感器造成信号干扰。`msdelay()` 代码实现如下:

```
static int delayLoopCount2=1310;
void msdelay(int time)
{ int i;
  for(;time>0;time--)
    for(i=0;i<delayLoopCount2;i++); }
```

可以利用 PC 口相关引脚作为超声波传感器使能端口, 程序实现如下:

```
rGPCCON=0x555; // 端口 C 为输出口用于启动超声
// 波, 可激发多路超声波
rGPCDAT=0x0; // 数据位清 0
rGPCDAT=0xf; // 设置为高频脉冲, 发出 8T 40kHz 超
// 频波
usdelay(1); // 延迟 10 μs
rGPCDAT=0x0; // 清数据位
```

3.2 回波检测实现

利用定时器计数器功能和外部中断可以实现回波检测计算障碍物距离, 外部中断接超声波的 ECHO 并设置为下降沿触发。定时器的初始化如下:

```
rTCFG0=0x9595; // 预分频值为 95
rTCFG1=0x00000; // 分频值 1/2
rTCNTB0=10 000;
```

根据 $T = [TCNTB0 \times (TCFG0 + 1) \times (1/TCFG1)] / 50\text{MHz}$ 得出定时器的周期为 60ms, 在这个时间段内足以计算回波时间。

外部中断 1 初始化如下:

```
rGPFCON=0xaa;           //GPF1 设置为 EINT1
rINTMOD=0x0;             // 为普通中断
rGPFUP=0xf;              // 禁止 GPF0 上拉电阻
pISR_EINT1=(unsigned)Eint1Handler; // 设置外部中断
                                // 中断向量
EnableIrq(BIT_EINT1);    // 使能中断
rEXTINT0 = 0x492;        // EINT1 下降沿触发
```

3.3 障碍物距离计算

```
void __irq Eint1Handler(void) // eint1 中断服务子程序
{
    sonar_time_front = ((10000 - rTCNT00) >> 1); // 定时
                                                // 器的计数值除以 2 即为障碍物距离
    ClearPending(BIT_EINT1); // 清除中断标志位
    i++;
    Uart_Printf("\nEINT1 occurred times=%d\n", i); // 串口打
                                                // 印调试显示中断发生次数
    Uart_Printf("\nsonar_time_front=%d", sonar_time_front);
// 串口打印显示计算器计数值
    t1 = sonar_time_front * 0.204; // 计算距离值
    front_range = t1; // 赋值全局变量
}
```

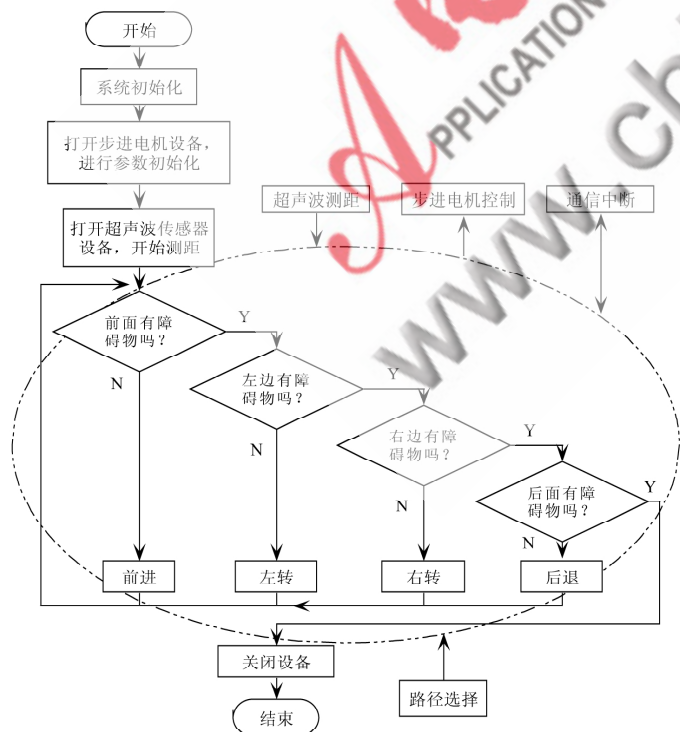


图2 机器人行进控制框图

3.4 轮循方式测距

由于超声波之间有干扰, 采用轮询^[4]的方式依次打开超声波, 定时中断中启动下一路超声波传感器工作, 下一路对应的定时中断再启动其他路, 每次只测 1 个, 这样可以有效地避免干扰, 为了提高实时性可以控制定时器周期, 如设置为 35 ms, 这样 4 路超声波轮循 1 次大约为 140 ms, 足以满足实时要求。

4 机器人行进过程

利用多路超声波传感器的反馈障碍物信息控制机器人直流电机驱动系统的软件流程图如图 2 所示。

嵌入式 Linux 系统中 SRF05 模块为只读的字符设备。在应用层下打开该设备后, 驱动程序完成测距, 测得的数据传到应用层, 应用层根据算法判断障碍物的位置给直流电机, 控制系统实现避障导航^[5]。

机器人的每次行为都是根据超声波测距模块和当前机器人运行的即时速度给出运行时间, 在遇到障碍物时 ($0 < t \leq 30 \mu s$ 时达到超声波的最大探测距离), 开启摄像头图像采集模块, 采集的障碍物图像利用无线 Modem 传回上位机, 这样用户就可以知道前方是何障碍物, 该系统实现了移动机器人的自主避障和采集障碍物信息的功能, 可以用来探测未知环境^[6]。

采用新型 SRF05 超声波传感器, 在 S3C2410+Linux 平台的移动机器人上实现了超声波的测距, 并利用超声波传感器控制驱动系统和图像采集系统。各模块驱动程序已成功加载到嵌入式 Linux 操作系统中, 在此平台下机器人运行效果良好。

参考文献

- [1] RUBINI A. Linux 设备驱动程序. LISOLEG 译. 北京: 中国电力出版社, 2000: 66-76.
- [2] BILLUR B. Fast processing techniques for accurate ultrasonic range measurements[J]. Measurement Science and Technology, 2000, 11(1): 45-50.
- [3] 王峰, 葛立峰. 一种高精度超声测距系统的设计[J]. 计算机技术与发展, 2008, 18(1): 229-236.
- [4] 陈春林, 陈宗海, 卓睿. 基于多超声波传感器的自主移动机器人探测系统[J]. 测控技术, 2004, 23(6): 11-13.
- [5] TIKANMKI A, MKEL T, PIETIKINEN A, et al. Multi-robot system for exploration in an outdoor environment[J]. Robotics and Applications and Telematics, 2007, 9(1): 563-567.
- [6] DAI G, CHEN T. Design on measurement and control system of cleaning robot based on sensor array detection[J]. Control and Automation, 2007. ICCA 2007. IEEE International Conference on Volume, Issue, 2007-05-30 ~ 2007-06-01: 1319-1322.

(收稿日期: 2008-12-25)