

C 语言在 AVR 单片机控制的微打印应用

刘 芙¹, 陈宏明²

(1. 江苏省淮阴商业学校 计算机学院, 江苏 淮阴 223003; 2. 淮阴工学院, 江苏 淮阴 223003)

摘要: 在计算机高速测控的时代下, 运用 AVR 单片机 ATMEGA16 作为主控制芯片, 在芯片内建立打印字库, 使用 FAN8200 作为热敏打印机走纸系统步进电机的驱动芯片, 以日本精工 LTP1245 高速热敏打印机芯, 用计算机控制程序对移位实现数据加载, 在测量与控制模式下命令走纸步进电机所需的 PWM 驱动信号, 并通过计算机设计了自定义的字符库, 简化了硬件结构, 降低了系统成本; 通过对该课题的研究, 探索了单片机技术的最新发展方向以及 AVR 单片机对微型热敏打印机的控制方法, 建立简单、实用、低成本的微型热敏打印系统并取得很好的应用价值。

关键词: AVR 单片机; 热敏打印机; 计算机 C 语言; 测量与控制

Computer C Language in AVR Micro-printing Application

Liu Fu¹, Chen Hongming²

(1. School of Computer Science, Huaiyin Commercial School of Jiangsu, Huaiyin 223003, China;

2. Huaiyin Institute of Technology, Huaiyin 223003, China)

Abstract: In the era of high-speed computer measurement and control, the use of AVR microcontroller ATMEGA16 as the main control chip, in the chip to establish a print word library, the use of FAN8200 as a thermal printer paper walking system step motor drive chip, to Japan Seiko LTP1245 high-speed thermal printer core, Using the computer control program to achieve data loading of shift, command the PWM drive signal required for the paper stepper motor in measurement and control mode, and design a custom character library by computer, simplify the hardware structure and reduce the cost of the system. Through the research of this topic, the latest development direction of microcontroller technology and the control method of AVR micro-thermal printer are explored, and a simple, practical and low-cost micro thermal printing system is established.

Keywords: AVR microcontroller; thermal printer; C language; measurement and control

0 引言

微型打印机并不是局限于某个时代的产物, 自从 1946 年日本的爱普生公司研发了第一个微型打印机开始, 到现在已有 65 个年头之多, 对其适用性和可优化性的研究并没停息过。当下, 业界对体积小、重量轻、噪音低、高速度、高清晰、连接方便、功能完备的微型热敏打印机研究步伐依然如故。如何建立一个由计算机 C 语言编程的基于 AVR 单片机控制的微型热敏打印的独立系统。系统采用 LTP1245 热敏打印机芯为对象, 由 AVR 单片机 ATMEGA16 作为主控制芯片, FAN8200 作为热敏打印机走纸系统步进电机驱动芯片, 通过串口与上位机进行通讯。并制定给出了计算机控制程序系统的控制流程及相应的字库, 致使该微型打印机实现了系统优、硬件简、成本低、应用广阔的前景^[1-3]。

1 系统设计整体结构

基于采用日本精工 LTP1245 高速热敏打印机。LTP1245 由走纸单元、加热单元及控制单元构成。^[4] 系统硬件结构是 AVR 单片机主控电路、驱动走纸、热敏头、检保电路等组成, 如图 1 所示。

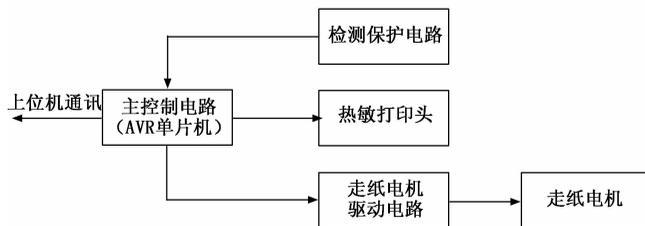


图 1 系统硬件结构框图

表 1 为 LTP1245 打印机芯规格参数。

1.1 走纸单元

LTP1245 热敏打印机芯走纸电机采用 2 相 4 拍步进电机, 其规格参数如表 2 所示, 激励时序如表 3 所示, 驱动脉冲波形如图 2 所示。电机以顺时针的方向旋转时, 是正向进纸^[4]。

收稿日期: 2019-12-13; **修回日期:** 2020-01-20。

基金项目: 国家自然科学基金项目(60575052); 江苏省第四期职业教育教学改革课题(ZYB145); 国家规划重点资助课题(HAG05056)。

作者简介: 刘 芙(1973-), 女, 江苏涟水人, 大学本科, 高级讲师, 主要从事控制工程、人工智能、物联网方向的研究。

表 1 LTP1245 打印头规格参数

项目	规格
打印方法	热敏行点打印
每行点数	384 点
同时打印点数	192 点
解像度	8 点/mm
打印速度	200 点 行/秒(25.0 mm/s)(于 5 V)
	450 点 行/秒(56.25 mm/s)(于 7.2 V)
	500 点 行/秒(62.5 mm/s)(于 8.0 V)
打印宽度	48 mm
打印纸宽	58 mm
打印头温度检测	热感应器
热写头检测	光电
缺纸检测	光电
操作电压范围:	
Vp (打印头和马达驱动电压)	4.2~8.5 V
Vdd (打印头逻辑电压)	4.5~5.5 V
电流消耗:	平均:1.8A (于 5.0 V)2.6A (于 8.0 V)
打印头驱动驱动电压(Vp)	最高: 2.1A (于 5.0 V)3.3A (于 8.0 V)
马达驱动电压 (Vp)	最高:0.46 A
打印头逻辑电压(Vdd)	最高:0.01 A

表 2 LTP1245 走纸电机规格参数

项目	规格
拍数	4 拍
驱动方式	双极斩波器
激励	2 相
每相绕组电阻	14 Ω ±10%
额定电压	4.2~8.5V
额定电流	0.23A/相
最大电流	0.46A
驱动频率	50~1 000 pps

表 3 LTP1245 走纸电机激励时序

信号名称	激励顺序			
	第 1 步	第 2 步	第 3 步	第 4 步
A/	低电平	高电平	高电平	低电平
B	高电平	高电平	低电平	低电平
A	高电平	低电平	低电平	高电平
B/	低电平	低电平	高电平	高电平

激励顺序: 表 3 中是 LPT1245 进纸按正常的方向, 当马达按相位的顺序激励时, 第 1 步, 第 2 步, 第 3 步, 第 4 步, 第 1 步, 第 2 步……, 相反, 要使马达按相反的方面旋转, 则激励马达的顺序为: 第 4 步, 第 3 步, 第 2 步, 第 1 步, 第 4 步, 第 3 步……

步进电机时序:

1) 停止步骤: 为了停止马达, 给一个相位的单个激励周期, 也就是最后的一个打印步骤。

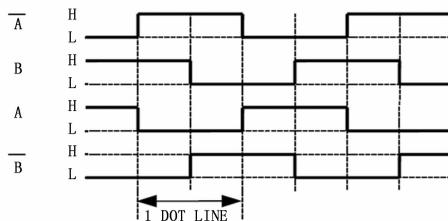


图 2 LTP1245 走纸电机驱动脉冲波形

2) 暂停状态: 在暂停状态, 为了防止步进电机过热, 不要激励步进电机, 即便在步进电机没有被激励的情况下, 仍然会有一个压力保持纸张不会滑动。

3) 起始步骤: 重新从步进电机的步进电机立即切换到打印状态。当重新启动电机的悬浮状态, 在单一步骤的输出和步骤相同的相位信号停止然后开始正常的打印顺序。

1.2 热敏头单元

LTP1245 热敏头如图 3 所示: 由 384 个加热单元及加热驱动构成。每一点线的打印数据按位由低到高在同步脉冲时钟以串行方式传输的数据, 然后将门锁止, 通过以下方式注册的门锁信号 (见图 4)^[5]。输入热敏头加热到写激励信号的 DST 反过来根据 printdata 存储在锁存寄存器, 刺激加热装置, 在热敏打印纸上产生点行^[6]。LTP1245 支持 6 个分区的打印。表 4 显示了 DST 信号与加热单元的关系。

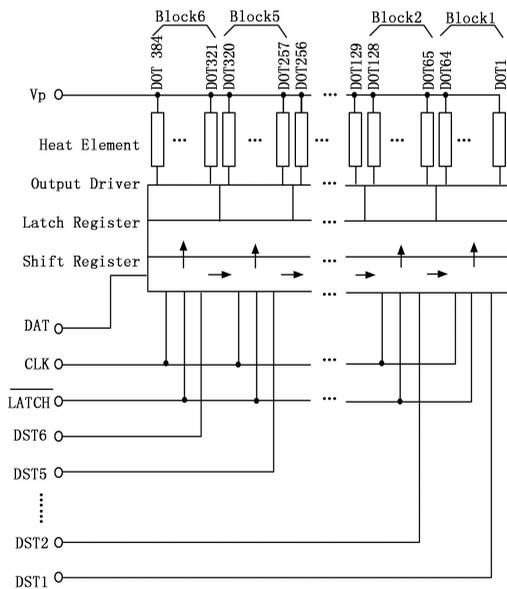


图 3 热敏头框图

表 4 分区和加热单元

分区数	加热单元数	点数/DST
1	1~64	64
2	65~128	64
3	129~192	64
4	193~256	64
5	257~320	64
6	321~384	64

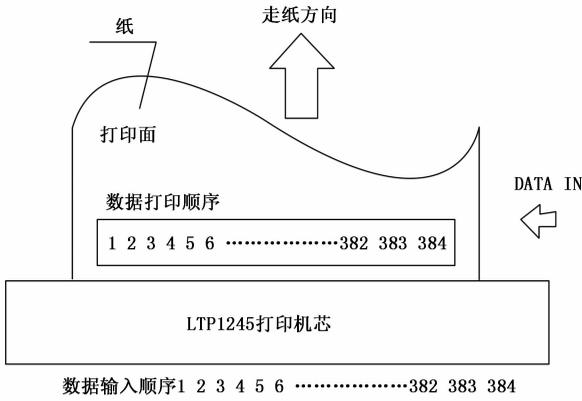


图 4 LTP1245 数据打印方式

LTP1245 热敏头工作时序如图 5 所示, 各脉冲信号的典型时间如表 5 所示。

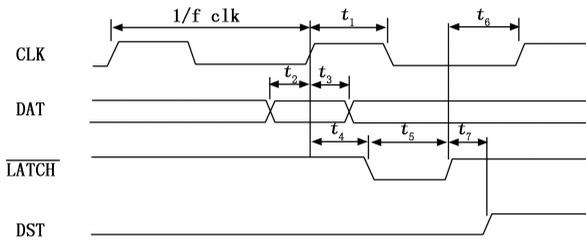


图 5 LTP1245 热敏头工作时序图

表 5 LTP1245 时序典型时间

项目	代号	典型时间
CLK 频率	Fclk	<4 MHz
CLK 脉冲宽度	T1	>80 ns
DAT 设置时间	T2	>50 ns
DAT 保持时间	T3	>50 ns
LATCH 设置时间	T4	>120 ns
LATCH 脉冲宽度	T5	>120 ns
LATCH 保持时间	T6	>120 ns
DST 设置时间	T7	>120 ns

LTP1245 为 384 点打印宽度, 即 48 个字节, 由于目前使用的都是双层热敏纸, 所以打印脉冲为 2 个脉冲/点, 即每点行 24 个字节。数据传输时 24 个字节数据按位由低到高依次传送。

根据 LTP1245 热敏头工作时序, 热敏打印头初始化时同步脉冲 CLK 为低电平, 锁存脉冲 LATCH 为高电平, 加热激励脉冲 DST 为低电平^[7]。打印时首先需要将打印数据传送到热敏打印头, 同步脉冲 CLK 输出 1 个脉冲信号作为传输开始, 在 CLK 为低电平时数据的第一个位放到信号线 DAT 上, 经过延时 t_2 后同步脉冲 CLK 变为高电平, 保持 t_3 时间后当前位数据传送完毕, CLK 变为低电平, 循环反复将所有 24 字节数据传送完毕, 在传送最后一位时在 CLK 为高电平时锁存脉冲 LATCH 变为低电平并保持 t_5 时间,

将数据锁存到热敏打印头, 然后加热激励脉冲 DST 变为高电平将数据加热打印到热敏打印纸上^[8]。

LTP1245 光电检测电路如图 6 所示, 当热敏头压纸杆打开时 HS 的信号为高电平, 当缺纸时 PS 的信号为高电平, 正常工作时均为低电平。

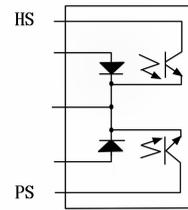


图 6 LTP1245 光电检测

1.3 主控制电路单元

为了实现快速打印, 要求处理器给打印头传送数据的速度要快。这就要求选用高性能、高速度的单片机做主控制器——AVR 单片机, 该芯片是高性能, 低功率 8 位 AVR 微处理器与先进的 RISC 结构, 131 指令, 多数指令执行时间为一个唯一时钟周期, 32 位 8 位普遍运行的计数器, 充分的静态操作, 16 MIPS 运行在 16 兆赫下, 硬件乘算器以仅二时钟周期发挥程序迭行和数据记忆; 16k 字节在系统可编程程序的闪光; 删掉生活: 10 000 次; 任意起动编码区与锁位的独立; 编程通过在芯片 Boot 程序起动编码系统, 真实地进行同时读和写操作; 512 字节 EEPROM, eraselife: 100 000 次, 片内模拟比较器等特殊的处理器特点; 上电复位和可编程电源故障检测, 在芯片标定 RC 振荡器、芯片/芯片外的中断源, 6 睡眠模式: 空闲模式、ADC 的噪声抑制模式, 节能模式, 断电模式下, 待机模式; 32 个可编程的 I/O 口^[9]。

主控制电路如图 7 所示。各控制引脚功能如下:

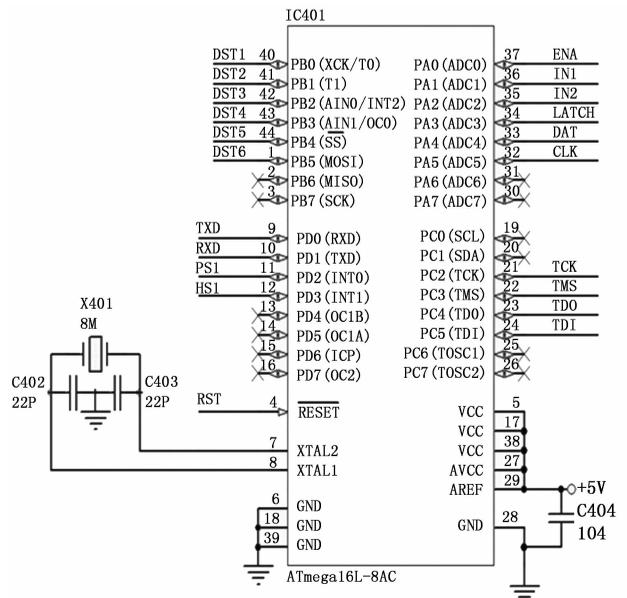


图 7 主控制电路图

- 1) RST: 复位
- 2) TXD, RXD: 串口 USART。
- 3) DST1~DST6: 热敏头打印 6 个控制区的加热接口。
- 4) ENA: 走纸电机使能, 当打印时 ENA 为高电平, 使步进电机驱动器 FAN8200 输出驱动脉冲。
- 5) IN1, IN2: 步进电机驱动脉冲输入, 工作时序如图 5 所示。
- 6) LATCH: 热敏打印头数据锁存脉冲, 当一点行数据由 DAT 传送到热敏打印头后, 该引脚由高电平变为低电平, 将数据存入热敏打印头。
- 7) DAT: 热敏打印头数据输入线。
- 8) PS1: 缺纸信号, 当热敏打印头缺纸时该信号为低电平。
- 9) HS1: 打印头压板杠杆解除信号, 当热打印头压板杠杆解除信号为低电压。
- 10) CLK: 同步信号脉冲的热打印头, 打印数据传输信号中的脉冲通过 DAT 热打印头的配合。

2 系统计算机控制程序设计

2.1 系统整体工作流程

本热敏打印机的计算机控制程序设计主要是通过串口接收上位机数据, 并判断数据类型, 如果是命令字, 则转去执行相应的命令程序。如果是打印字符, 则转换成相应的点行数据输送到热敏打印机的移位寄存器, 然后加热打印并走纸一定步数。打印的字符可以是 16 * 16 点阵的汉子字符, 也可以是 8 * 8 点阵的 ASCII 码^[10]。整体工作流程如图 8 所示。

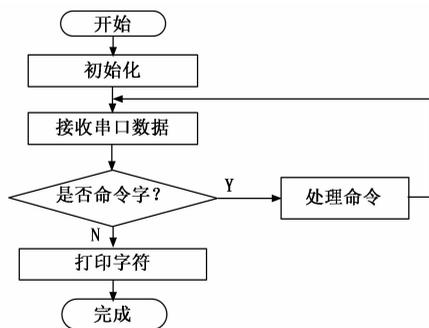


图 8 系统整体工作流程

2.2 系统整体工作流程

根据图 5 中工作时序的要求, 数据加载的工作流程如图 9 所示。在单片机模拟时钟 CLK 的配合下将点阵寄存器中的数据由低位到高位依次通过数据线 DAT 输出数据到热敏打印头内部的移位寄存器中。LTP1245 一点行需要 24 个字节数据, 故需要移位 24 个字节。工作流程如图 9 所示。

计算机 C 语言程序如下:

```

/* 数据加载函数 */
void Loaddata()
{
    uchar i,j,data; //i:左移位的次数;j:传送字节数;data:要传送
  
```

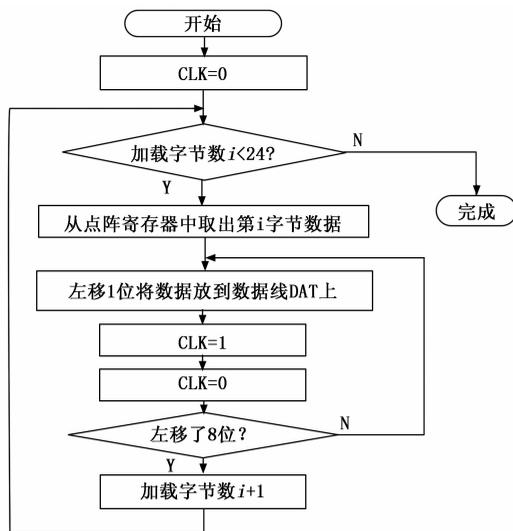


图 9 数据加载工作流程

的字节

```

CLK=0; //时钟为低电平
for (j=0;j<24;j++) //一点行总共需要传送 24 个字节
{
    data=Pointbuf[j]; //从点阵寄存器中取出要传送的字节数据
    for (i=0;i<8;i++) //循环左移 8 次,
    {
        if (data & 0x80)
            DAT=1; //若此位为 1,数据线为高电平
        else
            DAT=0; //若此位为 0,数据线为低电平
        CLK=1;
        delay_us(1);
        CLK=0;
        delay_us(1);
        data = data<<1; //左移 1 位
    }
}
  
```

图 10 为数据打印的工作流程: 首先将从串口接收的打印字符存入字符缓存区, 判断字符的类型。由于 ASC 码是 8 * 8 点阵, 所以每个字符有 8 个字节数据; 中文字符为 16 * 16 点阵, 共有 32 个字节, 汉字点阵的排列顺序采用“田字格”方法, 将一个汉字分成左上、左下、右上、右下 4 个部分, 每个部分就是一个 8 * 8 点阵数据。确定字库后如果是 ASC 码字符, 则将一行打印字符数据依次取第一行点阵, 重新编成一行点阵数据。将这一行点阵数据传送到热敏打印头并锁存。由于采用双层热敏打印纸, 每一打印行最多 24 个字节字符数据, 打印时先给 DST1、3、5 区加热然后走纸 1 点行, 再次加热 DST2、4、6 区, 走纸 1 点行, 完成一点行打印。取第二行点阵数据松热敏打印头打印, 循环直到一行字符打印完毕。如果是汉字字符, 则将一行打印字符数据依次取第一行点阵, 重新编成一行点阵数据。

将这一行点阵数据传送到热敏打印头并锁存。^[11]由于采用双层热敏打印纸, 每一打印行最多 24 个字节字符数据, 打印时先给 DST1、3、5 区加热然后走纸 1 点行, 再次加热 DST2、4、6 区, 走纸 1 点行, 完成一点行打印。取第二行点阵数据松热敏打印头打印, 循环直到一行字符打印完毕^[11-12]。

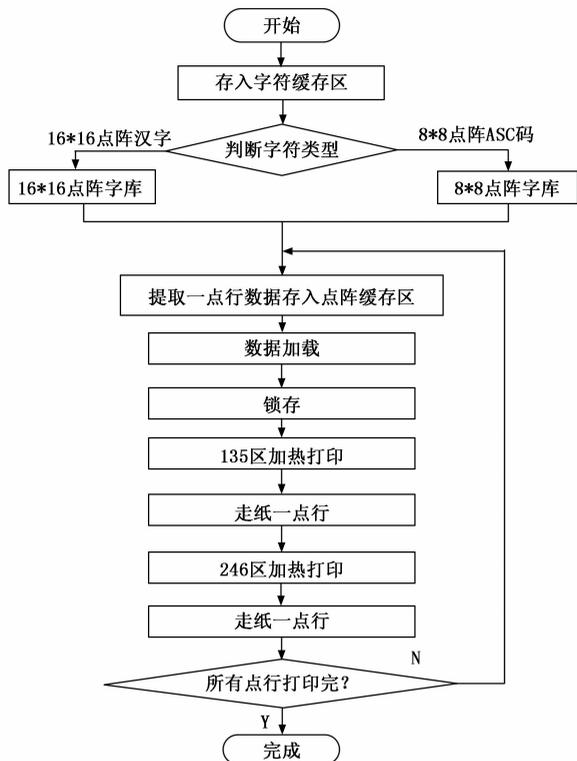


图 10 数据打印工作流程

打印 8 * 8 点阵 ASCII 码的计算机 C 语言程序:

```

/* 打印 8 * 8 点阵字符 */
void PrintASC()
{
    uchar i,j; //i: 打印点行数;j: 每点行打印字符字节数
    uint data; //加载的数据
    Pointbuf_clear(); //清打印点阵存储区
    for (i=0;i<8;i++) //打印 8 点行
    {
        for (j=0;j<24;j++) //每点行 24 个字节
        {
            data=Codebuf[j]-0x20; //转换字符,由 ASCII 码//0X20 开始
            data=data * 8+i; //每个 ASCII 码为 8 * 8 点阵
            Pointbuf[j]=ASCCODE[data]; //存入点阵存储器
        }
        Loaddata(); //数据加载
        latch_set(); //锁存
        dst135(); //135 区加热打印
        Motor_run(); //走纸 1 点行
        Dst246(); //246 区加热打印
        Motor_run(); //走纸 1 点行
    }
}
    
```

```

}
}
    
```

2.3 走纸电机工作流程

根据 LTP1245 走纸电机激励时序及驱动芯片 FAN8200 的工作时序, 走纸电机工作流程如图 11 所示。开始打印时 FAN8200 的使能端 ENA (ENA=ENA1=ENA2) 变为高电平, 输入端 IN1=1, IN2=1 走第一步, 接下来 IN1=0, IN2=1 走第二步, IN1=0, IN2=0 时走第三步, 第四步 IN1=1, IN2=0。由于 LTP1245 的走纸电机是 2 相 4 拍步进电机, 所以走四步后重新循环, 由第一步开始。当一点行打印结束后使能端 ENA 变为低电平, 停止步进电机的驱动输出。下面是计算机 C 语言的打印机走纸电机驱动程序^[13]。

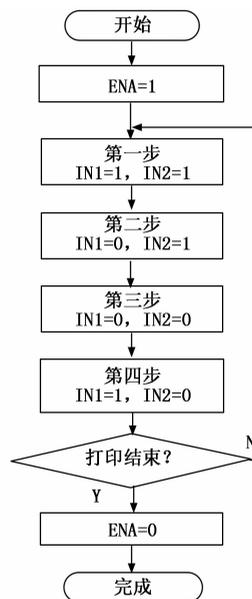


图 11 走纸电机工作流程

/* 打印机走纸电机驱动程序 */

```

void Motor_run()
{
    ENA=1;
    switch(Step)
    {
        case 0:
            IN1=1;
            IN2=1;
            Step=1;
            break;
        case 1:
            IN1=0;
            IN2=1;
            Step=2;
            break;
        case 2:
            IN1=0;
            IN2=0;
            Step=3;
            break;
        case 3:
            IN1=1;
            IN2=0;
            Step=4;
            break;
    }
}
    
```

