

# 深井泵自动控制器

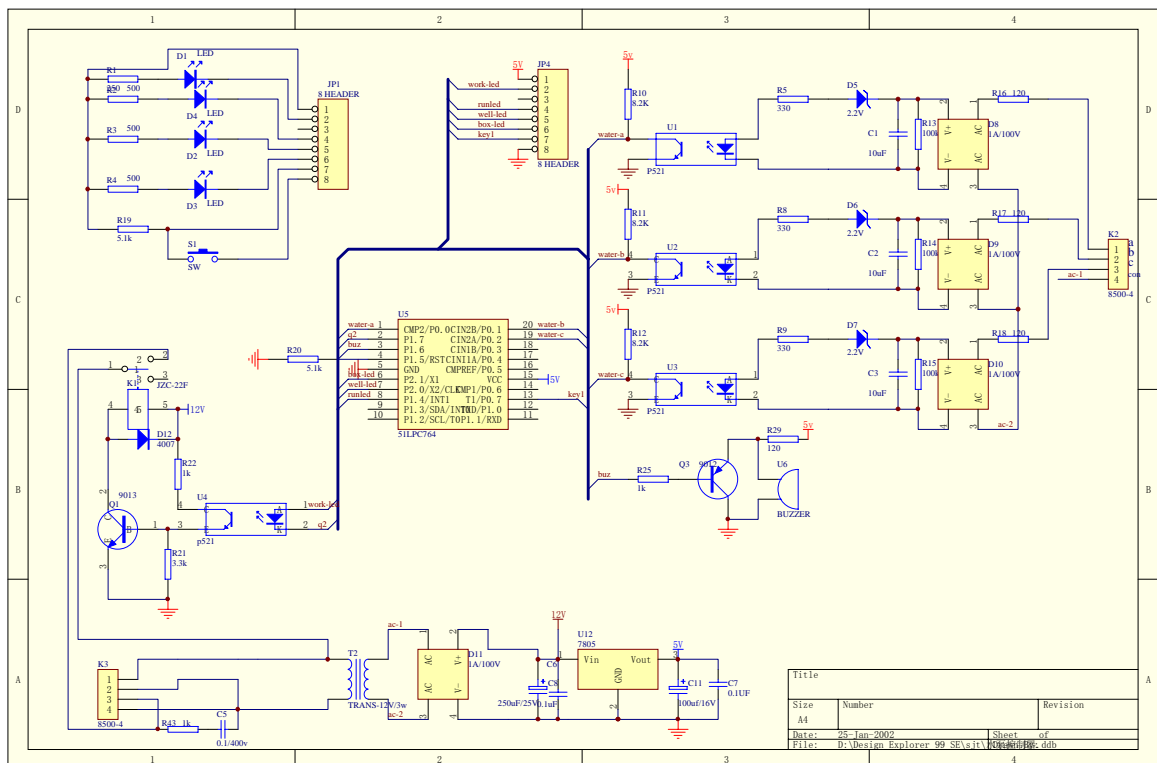
浙江义乌 吴志强

## 一、前言

环境污染，水资源紧张，农村打井取水情况较为普遍。现有的三极管电路和浮球开关水位控制器故障率较高，三极管电路构成的控制元件常出现的故障是抗干扰能力差，会使三极管误动；以浮球开关为形式的控制元件，由于浮球的连接线经常会磨断，以及开关触点的腐蚀和氧化，故障率也较高；且水井水量不足，水泵长时间空抽，容易毁坏电机。

## 二、电路设计

控制器电路如图所示，其核心部件采用了 PHILIPS 公司的 51LPC 系列 P87LPC762（如果有 P87LPC760 则可充分利用单片机的资源）OTP 单片机，51LPC 系列单片机是 PHILIPS 公司开发的一种 80C51 改进型 MCU，内置 2KB OTP 程序存储器和 128 字节 RAM，内置 RC 振荡器，看门狗，比较器等功能单元，具有集成度高，可靠性高，功耗低的特点。（详细资料请参考《PHILIPS 51LPC 系列单片机原理及应用设计》北京航空航天大学出版社出版）

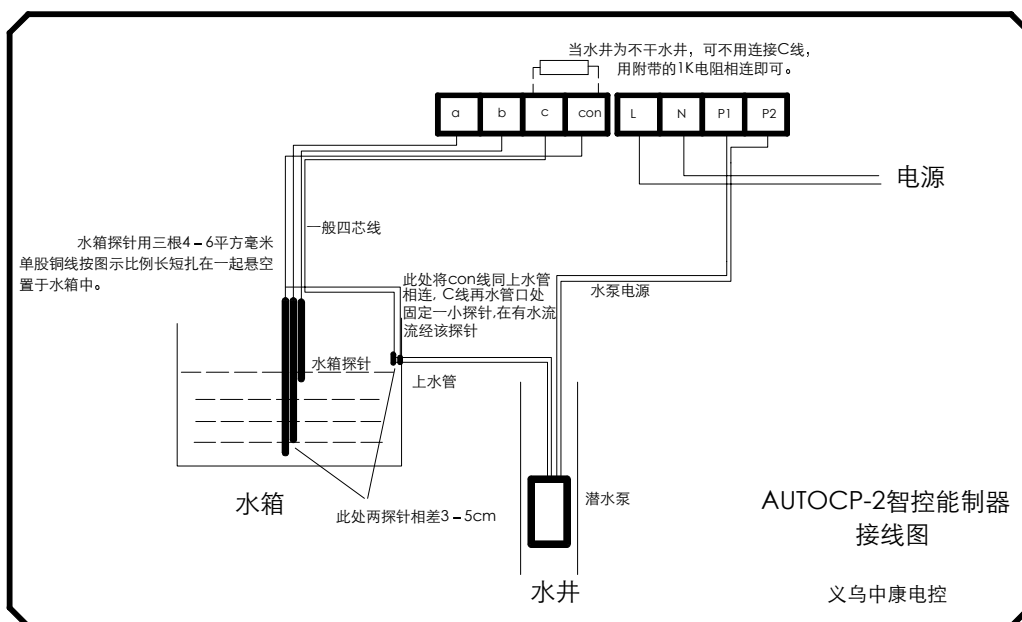


该控制器是在 P87LPC762 的外围加上水位采样、驱动、按钮显示、无音源蜂鸣器、电源回路等几部分组成。农村的水箱一般都在±0.00 起始 12 米以上，而控制器一般安装在底层，以致采样回路传输线相当于一付庞大的天线，极易受干扰，为提高采样回路的抗干扰和探极电解抗腐蚀性，采用 3—6mA 交流电流环进行水位检测，最低的探点“con”为交流 12V 电源接入点，另外三个（a，b，c）为水位探点，若水位达到相应的位置，交流 12V 电源与该探点接通，其输出信号经 R16 限流，D8 整流桥全波整流，C1 滤波和保证交流过零时 U1 光耦导通，R5 限流，D5 限流和抗干扰，驱动光耦给 762 一个开关信号。输出信号经光耦 U4 隔离，三极管 Q1 放大驱动继电器 K1，使得水

泵得电工作，二极管 D12 为续流二极管，R43, C5 抑制火花，发光二极管 D1 指示电机在工作状态，即显示同驱动复用。水箱、水井、运行得 LED (D2、D3、D4) 经限流电阻直接由 762 的 I/O 驱动。为了利用 762 的资源 and 节约成本，告警音用不带音源得峰鸣器来产生，由 buz 口输出约 2KHz 的方波驱动三极管，使峰鸣器 U6 发声。

### 三、工作原理

控制器的接线图如图 2 所示，当控制器上电时，对 LED 和 BUZZ 进行自检，由 762 的上电标志 POF 使 BUZZ 自检时发长音。延时 15 秒后，当水箱水位低于下限 a 时，控制器判为开机状态，水泵得电工作抽水，直致到上限水位 b，控制器判为水满输出关机；当水箱水位由 a—b 过程中时，并无水 (c 点判定) 则停机，延时半小时再自开机，且电机连续工作不超 1 小时 (约)，如此循环，直致到上限水位 b 时关机。



刚开机 5 秒内，进水口 c 检测不到进水，装置进入告警状态。

为防止探极回路失效，水泵连续工作 1 小时后进入停机状态。

对采样回路有容错和采样异常告警功能。有水时水位 led 不点亮，无水时点亮。

### 四、软件设计

P87LPC762 OTP 单片机内置 2KB 程序存储器中固化了系统应用程序，系统采用 SE-764 仿真器仿真，程序在 MedWin 调试器上采用 C51 语言调试通过，应用程序包括：主程序、T0—50ms 定时中断程序、键盘中断函数、时标函数、峰鸣器驱动函数等。相应的程序见附后。

1. 主程序：控制器上电后，首先初始化看门狗，对 LED 进行检查，初始化 T0、T1，BUZZ 检查，键盘中断初始化，使能中断，开定时器 T0，进入低功耗等待。

2. T0—50ms 定时中断程序：T0 定时初值重装，喂狗，装置是否在告警状态，“是”进入告警程序，“否”每中断一次采样一次，采样 15 次后，对采样结果处理，如采样成功，启动采样分析程序，否则从采样出错处理程序中退出；如探极正确，启动抽水处理程序，否则从探极出错程序中退出。中断 20 次启动时标函数和亮灭运行灯。

3. 峰鸣器驱动函数：峰鸣器的 2KHz 方波和发声间歇均有 T1 定时中断函数控制，且在上电时检查上电标志 POF

位，POF 为 1 则发长音，清 POF 和置位 UP\_POF 使上电时延时 15 秒输出控制命令。

/\*\*\*\*\*\*  
 /\*\*\*\*\*

文件名: autocp2.c  
 主题: autocp-2 型智能水泵控制器程序  
 Ver 1.2  
 修改日期: 2001/12/21  
 修改内容: 将 1 型的井水位检测由井内改到水箱的出水口处,  
 刚开机时延时 5 秒检测.

\*\*\*\*\*  
 /\*\*\*\*\*

```
#include<reg764.h>
#define uchar unsigned char

#define TH0_VAL      0X3C      /*定时器 0 的重载值*/
#define TL0_VAL      0Xd2
#define TH1_VAL      0X35      /*定时器 1 的重载值*/
#define TL1_VAL      0X35
#define NO_WORK      30        /*水泵停机时间 30 分*/
#define WORK         60        /*水泵连续工作时间 60 分*/
#define IN_CONT      15        /*连续采样次数 15 次*/
#define STAR_DELAY   8         /*开机延时检测时间 8X15/20=6 秒*/
#define STAR_BOXC    5         /*开机检测 C 点无水计数值*/

uchar data h;                //采样出错计数
uchar data l;                //探极出错计数
uchar data i;                //采样次数,采样滤波用
uchar data temp;            //采样值

uchar bdata temp1;
sbit boxa = temp1^0;        //水箱 a 点采样值
sbit boxb = temp1^1;        //b 点
sbit boxc = temp1^2;        //水箱 c 点采样值

uchar bdata flag;
sbit WELL = flag^0;        //水井水位标志
sbit SET_KB = flag^1;      //键盘中断标志
sbit MOTOR = flag^2;       //水泵工作标志
sbit ERR = flag^3;         //采样出错标志
sbit ERR_IN = flag^4;      //探极出错标志
sbit errsign = flag^5;     //装置告警标志
sbit UP_POF = flag^6;      //开机上电标志
```

```

sbit MOTOR_STAR = flag^7;           //刚开机标志
sbit motor_led = P1^7;              //电机驱动
sbit box_led = P2^1;               //水箱水位显示
sbit well_led = P2^0;              //水井水位显示
sbit runled = P1^4;                //运行灯
sbit speak = P1^6;                //峰鸣器输出

typedef struct {                    //定义存储时间的结构
    uchar hour,min,sec;
}timestruct;
timestruct curtime;                //存放当前的时间

void input(void);                  //采样函数
void output_led(void);            //采样分析函数
void con_out(void);               //判断输出函数
void warning(void);               //告警函数
void errin_con(void);             //采样出错函数
void errled_con(void);            //探极出错函数
void delay(unsigned int);         //延时函数
void init(void);                  //初始化函数

/*****
名称:WDT_FEED
用途: 喂狗程序
说明:请参照 87LPC764 的喂狗方法说明
*****/
void WDT_FEED(void){
    WDRST = 0X1E;
    WDRST = 0XE1;
}

/*****
名称:INIT_WDT
用途: 看门狗初始化程序
说明:
*****/
void INIT_WDT(void){
    WDCON = WDCON|0X17;           //开启看门狗定时器(溢出时间约为 2.1 秒)
    WDT_FEED();                  //设置完后喂狗一次
}

```

/\*\*\*\*\*\*

函数名: kb\_int

功能描述: 延时, 清键盘中断标志, SET\_KB 置位, 如满足上水条件, 启动电机

参数: 无

返回: 无

\*\*\*\*\*/

```
void kb_int(void) interrupt 7 using 1{
    EKB = 0;                //禁能键盘中断
    TR0 = 0;
    AUXR1 = AUXR1&0x7F;    //清键盘中断标志
    if((P0^7==0) && (motor_led==1) && (boxc==1)){ //低电平有效
        SET_KB = 1;        //置位手动标志
    }
    TR0 = 1;
    EKB = 1;
}
```

/\*\*\*\*\*\*

函数名: input

功能描述: 对 P0 进行采样, 如果连续 5 次一样, 则 ERR=0;

输入参数: 无;

返回参数: 无;

\*\*\*\*\*/

```
void input(void){                //采样滤波
    static uchar K=5;            //连续采样相同次数
    if(temp==(P0&0x0f)){
        K--;
        if(K==0){                //连续 5 次相同
            ERR = 0;
            temp1 = temp;
            h = 0;
        }
    }else{                        //采样值不同
        temp = (P0&0x0f);
        K = 5;
        ERR = 1;                  //置位采样出错标志
    }
}
```

/\*\*\*\*\*\*

函数名: output\_led

功能描述: 对采样值进行分析, 如果探极有问题, 则 ERR\_IN=1;

如无问题则点亮相应的水位 led

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```
void output_led(void){
    if((boxb==0) && (boxa==1)){
        ERR_IN = 1;           //置位探极出错标志
    }else{
        ERR_IN = 0;           //清探极出错标志和探极出错计数器
        l=0;
        box_led = ~boxa;      //无水点亮
    }
}
```

/\*\*\*\*\*\*

函数名: warning

功能描述: runled 闪,进入装置告警状态;

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```
void warning(void)
{
    EA = 0;           // 禁能总中断
    EKB = 0;          //关闭键盘中断
    motor_led = 1;    //关机
    MOTOR = 0;
    well_led = 1;     //灭显示灯
    box_led = 1;
    errsign = 1;      //置位装置告警标志
    TR1 = 1;          //启动 T1 峰鸣器控制中断函数
    EA = 1;           //使能总中断
}
```

/\*\*\*\*\*\*

函数名: errin\_con

功能描述: 采样出错记数, 在两次内不作反映, 3 次以上停机, 6 次以上告警死循环

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```
void errin_con(void){
```

```

    h++; //采样出错计数
    if((MOTOR==1) && (h>4)){ //水泵工作,出错 4 次,关机
        motor_led = 1;
        MOTOR = 0;
    }
    if(h>8){ //8 次以上则进入告警状态
        warning();
    }
}

```

\*\*\*\*\*

函数名: errled\_con

功能描述, 探极出错, 重新采样, 记数, 连续 6 次告警, 15 次停机;

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```

void errled_con(void){
    l++; //探极出错计数
    if((MOTOR==1) && (l>8)){ //水泵工作,8 次以上关机
        motor_led = 1;
        MOTOR = 0;
    }
    else if(l>15){ //15 次进入装置告警
        warning();
    }
}

```

\*\*\*\*\*

函数名: INIT\_TMR

功能描述, 定时器初始化

输入参数: 无

返回参数: 无

\*\*\*\*\*

```

void INIT_TMR(void){
    TMOD = 0X21; //定时器 0 置为模式 1
                //定时器 1 置为模式 2

    TH0 = TH0_VAL; //定时器 0 初值
    TL0 = TL0_VAL;
    TH1 = TH1_VAL; //定时器 1 初值
    TL1 = TL1_VAL;

    ET0 = 1; //开放 T0/T1 中断使能.*
    ET1 = 1;
}

```

}

/\*  
\*\*\*\*\*  
\*/

名称:CHK\_LED

说明: 用于检查 LED 的显示.

/\*  
\*\*\*\*\*  
\*/

void CHK\_LED(void){

```
    speak = 1;
    motor_led = 1;
    well_led = 0;
    box_led = 0;
    runled = 0;
    delay(350);
    well_led = 1;
    box_led = 1;
    runled = 1;
    delay(100);
```

}

/\*  
\*\*\*\*\*  
\*/

名称:CHK\_BUZ

用途: 用于检查峰鸣器,.

说明:

/\*  
\*\*\*\*\*  
\*/

void CHK\_BUZ(void){

```
    EA = 1;                //使能总中断
    P1M2 = P1M2|0x40;     //P1.6 设为上拉输出
    TR1 = 1;              //开启峰鸣器振荡输出
    delay(250);
    TR1 = 0;              //停 T1
    EA = 0;                //禁能总中断
```

}

/\*  
\*\*\*\*\*  
\*/

名称:INIT\_KBI

说明:键盘初始化子程序

/\*  
\*\*\*\*\*  
\*/

void INIT\_KBI(void){

```
    KBI = 0x80;           //P0^7 键盘中断允许
    P0M2 = P0M2&0X7f;    //P0.7 仅为输入
    P0M1 = P0M1|0X80;
    EKB = 1;              //使能键盘中断
```

}



\*\*\*\*\*

函数名: delay

功能描述: 延时

参数: uchar x

返回: 无

\*\*\*\*\*/

```
void delay(unsigned int x){
    uchar y;
    while(x--){
        for(y=0; y<250; y++)
            {;}
    }
}
```

\*\*\*\*\*

功能: second\_tick

描述: 每秒钟执行一次函数功能, 时间更新

参数: 无

返回: 无

\*\*\*\*\*/

```
void second_tick(void){
    static unsigned char motor_work;           //水泵工作时
    static unsigned char motor_nowork;        //水泵不工作时
    curtime.sec++;
    if(curtime.sec>=15){                       //开机上电延时 15s
        UP_POF = 0;
    }
    if(curtime.sec>59){
        curtime.sec = 0;
        curtime.min++;
        if((motor_led==1) && (MOTOR==1)){     //停机状态
            motor_nowork++;                   //停机计时
            if(motor_nowork>=NO_WORK){        //延时到,开机
                motor_led = 0;
                MOTOR_STAR = 1;              //刚开机置位
            }
        }else{
            motor_nowork = 0;
        }
        if(motor_led==0){                     //电机工作
            motor_work++;                     //工作计时
            if(motor_work>=WORK){             //电机连续工作时间到,停机.

```

```
        motor_led = 1;
    }
}else{
    motor_work = 0;
}
if(curtime.min>59){
    curtime.min = 0;
    curtime.hour++;
    if(curtime.hour>23){
        curtime.hour = 0;
    }
}
}
}
```

\*\*\*\*\*

函数名: time0

功能描述, 50ms 中断一次,采样一次,采样满 20 次后,对采样结果进行处理;

如采样成功, 启动采样分析程序, 否则从采样出错处理程序重退出

如探极正确, 启动抽水处理程序, 否则从探极出错程序中退出。

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```
void time0(void) interrupt 1 using 1{
    static unsigned char second_cnt=20;           //正常运行时标
    static unsigned char err_cnt=5;              //装置告警时标
    TR0 = 0;                                     //重装初值
    TH0 = TH0_VAL;
    TL0 = TL0_VAL;
    TR0 = 1;
    WDT_FEED();                                 //喂狗
    if(erssign==1){                             //装置告警
        err_cnt--;
        if(!err_cnt){
            err_cnt = 5;
            runled = ~runled;
        }
    }else{                                     //装置正常
        i++;                                    //采样计数
        input();                               //采样函数
        if(i==IN_CONT){                       //采样次数到
            i = 0;
            if(ERR==0){                       //采样成功
```

```

        output_led();                //采样分析
        if(ERR_IN==0){              //探极正确
            if(UP_POF==0){          //上电延时到
                con_out();          //控制输出
            }
        }else{errled_con();}        //探极出错处理
    }else{errin_con();}            //采样出错处理再采样
}
second_cnt--;                      //记数值减
if(!second_cnt){                  //如果经过 1 秒
    second_cnt = 20;              //设置记数值
    runled = ~runled;
    second_tick();                //时标函数
}
}
}
}

```

\*\*\*\*\*

函数名: time1

功能描述, 生成峰鸣器间歇性的 2K 方波

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```

void time1(void) interrupt 3 using 1{
    static unsigned int buz_cnt=1000;
    bit l;
    if(!l){                          //2K 方波
        speak = ~speak;
    }else{                             //间歇时间
        speak = 1;
    }
    buz_cnt--;                          //记数值减 1
    if(!buz_cnt){
        if((PCON&0X10)==0x10){        // 检查上电标志
            PCON = PCON&0XEF;        //清上电标志
            UP_POF = 1;                //置位开机延时标志
            buz_cnt = 2000;            //设置记数值
        }else{
            buz_cnt = 1000;           //设置记数值
            l = ~l;
        }
    }
}
}

```

}

/\*\*\*\*\*\*  
 \*\*\*\*\*

主函数

\*\*\*\*\*  
 \*\*\*\*\*/

```
void main(void){
    EA = 0;                //禁能总中断
    INIT_WDT();           //看门狗初始化
    CHK_LED();            //led 检查
    INIT_TMR();           //定时器初始化
    CHK_BUZ();            //峰鸣器检查
    INIT_KBI();           //键盘初始化
    EA = 1;                //使能总中断
    TR0 = 1;              //开启定时器 T0
    for(;;){              //低功耗等待
        PCON = 0x01;
    }
}
```

/\*\*\*\*\*\*  
 \*\*\*\*\*

函数名: star\_monent

功能描述: 刚开机,延时,判断有无水,无水告警

输入参数: 无;

返回参数: 无;

\*\*\*\*\*  
 \*\*\*\*\*/

```
void star_monent(void){
    static uchar star_delay,boxc_con;
    if(MOTOR_STAR==1){    //刚开机时的延时
        star_delay++;
        if(boxc){
            boxc_con++;
        }else{
            boxc_con=0;
        }
        if(star_delay>=STAR_DELAY){ //达到开机延时
            if(boxc_con>=STAR_BOXC){
                warning();
                boxc_con=0;
            }else{
                boxc_con=0;
                MOTOR_STAR=0;
                star_delay=0;
                well_led=1;
            }
        }
    }
}
```

```

    }
}
}else{ //不是刚开机
    if(boxc==1){ //井无水
        motor_led=1; //停机
        well_led=0; //无水点亮
    }
}
}
}

```

\*\*\*\*\*

函数名: con\_out

功能描述: 对采样值进行分析, 水箱 B 点无水条件下, 如果水箱 A 点无水, 启动水泵, 延时 5 秒检测出水口, 如无水, 告警, 有水则正常抽水  
水箱水位在 AB 之间时保持抽水状态, 到过 B 点停机

输入参数: 无;

返回参数: 无;

\*\*\*\*\*

```

void con_out(void){
    if(boxb==1){ //水未满
        if(SET_KB==0){ //手动未按下
            if(MOTOR==0){ //关机状态
                if(boxa==1){ //水箱无水,
                    motor_led=0; //开机
                    MOTOR=1; //电机运行标志
                    MOTOR_STAR=1; //刚开机置位
                }
            }
            if(motor_led==0){
                star_monent(); //刚开机延时 5 秒
            }
        }else{ //手动按下
            SET_KB=0;
            motor_led=0; //开机
            MOTOR=0; //手动, 停止条件: 水箱满或水井无水
            MOTOR_STAR=1;
        }
    }else{ //水满, 关机
        motor_led=1;
        MOTOR=0;
    }
}
}

```

## 五、讨论

民用深井泵的功率一般都小于 1KW (单相), 水泵的工作电流在水泵空抽和抽水状态时无明显的边界量, 故无法用水泵的工作电流来判断井内是否有水。对于功率较大的深井泵则可用工作电流边界量来判断井内是否有水, 则可省掉水位 C 点采样回路, 同时还可利用该机电流采样回路 (V/F 法或  $\Sigma - \Delta$  ADC) 进行电机过载 (堵转) 保护, 则可使该控制器功能更为完善, 同时也充分利用了 P87LPC762 的潜能。

## 六、后记

**编辑寄语:** 作者自己购买了 SE-764 仿真器、MP-764 编程器及西尔特万用编程器等开发工具, 在业余条件下用专业的手段自行开发了“深井泵自动控制器”。首先, 我为他立志向上的求学精神所感动; 其次, 作者在很多网站上表现得十分乐于助人足以说明他的爱心可嘉; 其三, 作者将自己设计的作品完全公开, 其无私奉献的思想境界能够帮助后学先进的年轻人。同时, 对所有的初学者来说, 这无疑是一个很好的学习机会。通过对一个简单的控制项目的学习和调试, 如能达到学好 PHILIPS P87LPC762 单片机和 C51 高级语言的目的, 那又何乐而不为呢? 学习单片机应用开发技术关键在于实践, 特别是配合实验板来学习效果尤其明显。请有兴趣的网友自行与作者联系购买实验板, 如果在实验中遇到困境请与作者联系解答, 我想作者一定会十分热心地帮助您解决问题。如果您有更好的设计思想也请与作者和我联系, 让我们一起合作来完善这个作品。千万不要认为这是一个小控制器, 只要我们认真完善, 但也能充分体现一个人的设计思想, 那么一定有可能会成为一个好作品。(周立功)

作者简介: 吴志强, 浙江义乌人

详细地址: 浙江省义乌市桥东三区 13 栋 1 号 Zip: 322000

E-mail: [ywwzqjk@sina.com](mailto:ywwzqjk@sina.com)