COMPUTERISED ELECTRICAL EQUIPMENT CONTROL

■ V. MARIYAPPAN

Controlling electrical devices from a PC is great fun. Here is a Windows-based program written in 'C' language for controlling up to eight devices from the PC's parallel port termed as printer port (LPT). The program accepts the input in decimal numbers and outputs at the data output pins of the PC's parallel port for controlling the connected devices.

PC's parallel port

The parallel port is made up of three ports, namely, data port, status port and control port. It is found on the back of the PC as a D-type, 25-pin female connector. Here, we are concerned only with data lines D0 through D7 terminated at pins 2 through 9.

The data port is a write-only port, which means it can be used only to output data. Pins 18 through 25 of the connector are grounded. Control port is read/write capable, which means it can be used both for outputting and inputting some data to/from the external hardware. Status port is a read-only port, which means it can be used only to read data from the external hardware.

Table below shows pin details of the standard parallel port (SPP) and their traditional usage. The base address of the first parallel port (LPT1) is 378 (hex) or 888 (decimal). The data port of the parallel port can be accessed at its base address. The status port can be accessed at base address + 1, i.e., 0379 hex (or 889 decimal). The control port can be accessed at base address + 2, i.e., 037A hex (or 890 decimal). In case you are using LPT2 port, then substitute the base address of LPT2 as 0278 (hex) in place of 0378 (hex).

PARTS LIST

Semiconductor:

IC1-IC4	- CD4013 D-type flip-flop
IC5-IC12	- MCT2E optocoupler
IC13	- 7805 5V regulator
T1-T8	- BC548 npn transistor
D1-D8	- 1N4007 rectifier diode
BR1	- 1A, bridge rectifier

Resistors (all ¼-watt, ±5% carbon, unless stated otherwise):

R1-R8	- 100-ohm
R9-R17	- 10-kilo-ohm

Capacitors:

C1	- 1000µF, 25V electrolytic
C2	- 0.1µF ceramic

Miscellaneous:

RL1-RL8	- 5V, 100-ohm, 1C/O relay
S1	- Push-to-on switch
X1	- 230V AC primary to 9V, 250mA secondary transformer
	- 25-pin D-type parallel-port male connector

Pin Details of the Parallel Port

Pin number	Traditional use	Port name	Read/Write	Port address	Port bit
2-4	Data out	Data port	W	Base	D0-D2
5-9	Data out	—	W	Base	D3-D7
1	Strobe	Control port	R/W	Base+2	C0
14	Auto feed	—	R/W	Base+2	C1
16	Initialise	—	R/W	Base+2	C2
17	Select input	—	R/W	Base+2	C3
15	Error	Status port	R	Base+1	S3
13	Select	—	R	Base+1	S4
12	Paper end	—	R	Base+1	S5
10	ACK	—	R	Base+1	S6
11	Busy	—	R	Base+1	S7

Fig. 1 shows the circuit for interfacing the PC's parallel port to the devices to be controlled. The parallel port outputs the control signals generated by the software. The control signals are not continuous but a single clock pulse. For every 'on' or 'off' control, only a single clock pulse is sent from the parallel port to the circuit.

Data pins D0 through D7 of the parallel port are connected to pin 1 of optocouplers IC5 through IC12 via resistors R1 through R8, respectively. Optocouplers ensure complete isolation of the parallel port's data pins from the relay driver circuit.

Each optocoupler consists of an infrared light-emitting diode (LED) and an npn phototransistor. When a high going pulse is available on the data pin, the internal LED drives the phototransistor of optocoupler MCT2E and it provides a clock pulse to the corresponding flip-flop (IC CD4013) section.

IC CD4013 is a dual D-type flip-flop with independent set, clear and clock inputs and a single output. It accepts data when its clock pin is low and transfers it to the output on the positive-going edge of the clock. The high Q output of the flip-flop drives the corresponding transistor to energise the relay and switch on/off the device.

The flip-flops are set up for toggle mode by connecting their D inputs to $\overline{Q}$ outputs. Set inputs of all the flip-flops are grounded. Switch S1 is used to reset the flip-flops manually.

Fig. 2 shows the circuit of the power supply. The AC mains is stepped down by transformer X1 to deliver secondary output of 9V at 250 mA. The transformer output is rectified by full-wave bridge rectifier BR1, filtered by capacitor C1 and regulated by IC13 to

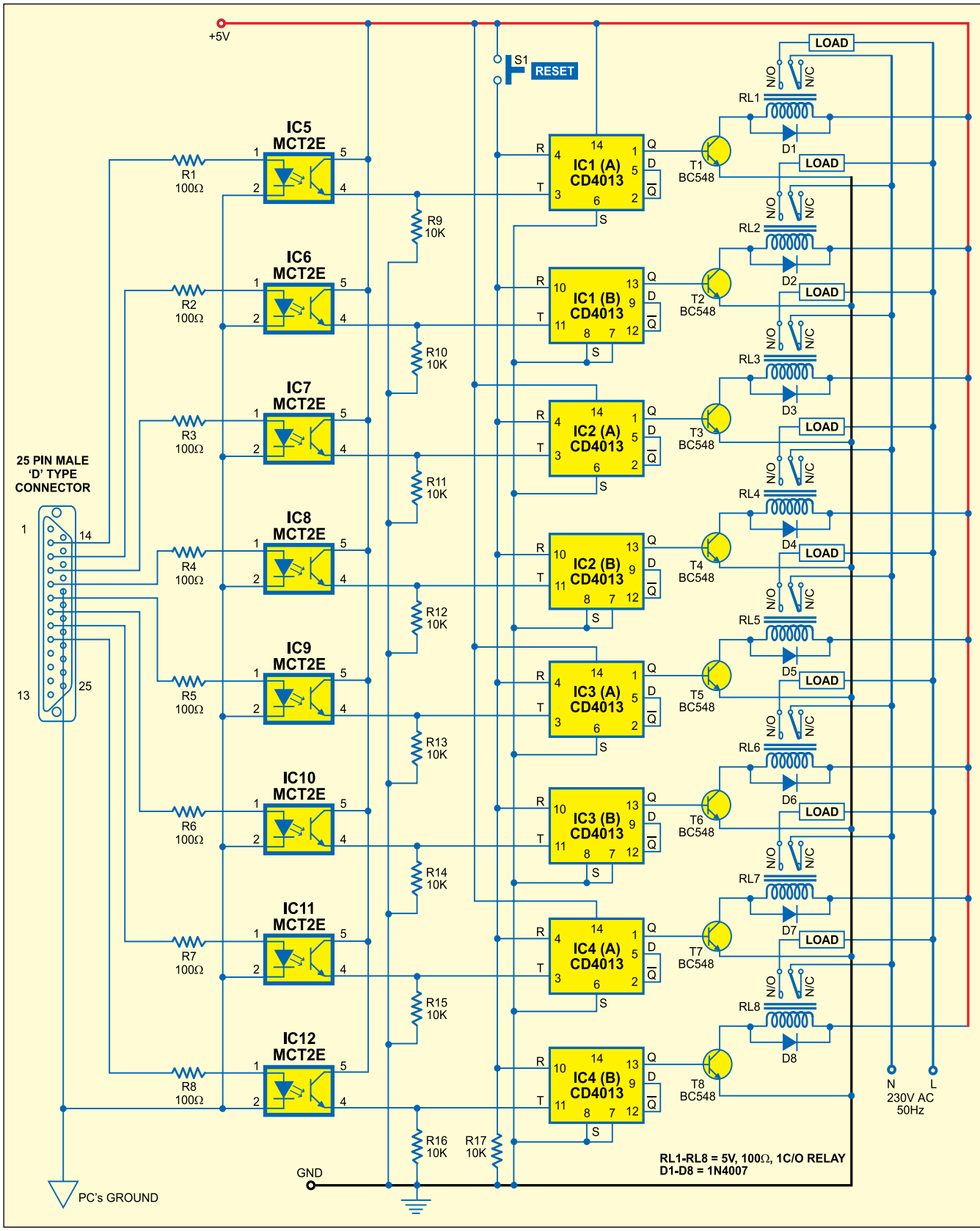

Fig. 1: Circuit diagram for computerised electrical equipment control

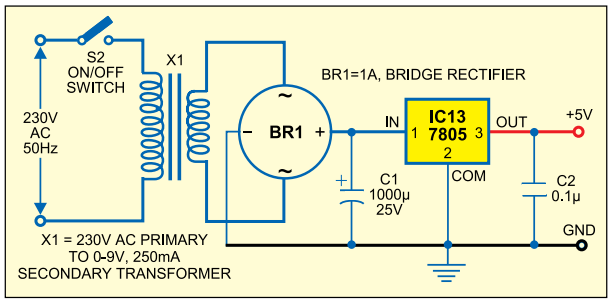

Fig. 2: Power supply circuit

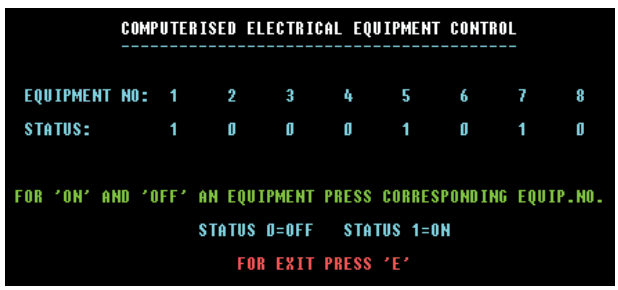

Fig. 3: Screenshot of the main screen for computerised electrical equipment control

provide regulated 5V supply. Capacitor C2 bypasses any ripple in the regulated output.

Before attempting to control the devices, the program (Control.exe) is to be run. A screen as shown in Fig. 3 appears which shows the current status of each device (on=1, off=0). To switch on a device, press the corresponding device number on the keyboard. This makes a high control pulse available on the data pin of the respective parallel-port pin and the toggle flip-flop switches from 'off' state to 'on' state to energise the relay and switch on the device. The status of the corresponding device on screen changes automatically.

Similarly, to turn the device 'off,' press the device number on the keyboard. This again makes a high control pulse available on the data pin and the corresponding toggle flip-flop switches from 'on' state to 'off' state to de-energise the relay and turn the device 'off.'

The program can be terminated simply by pressing the 'E' key, but before that you should turn all the devices 'off.' If you try to terminate the program without shutting down any of the devices, the message "Please shutdown all the equipment" will appear on the screen for a short period followed by the main screen.

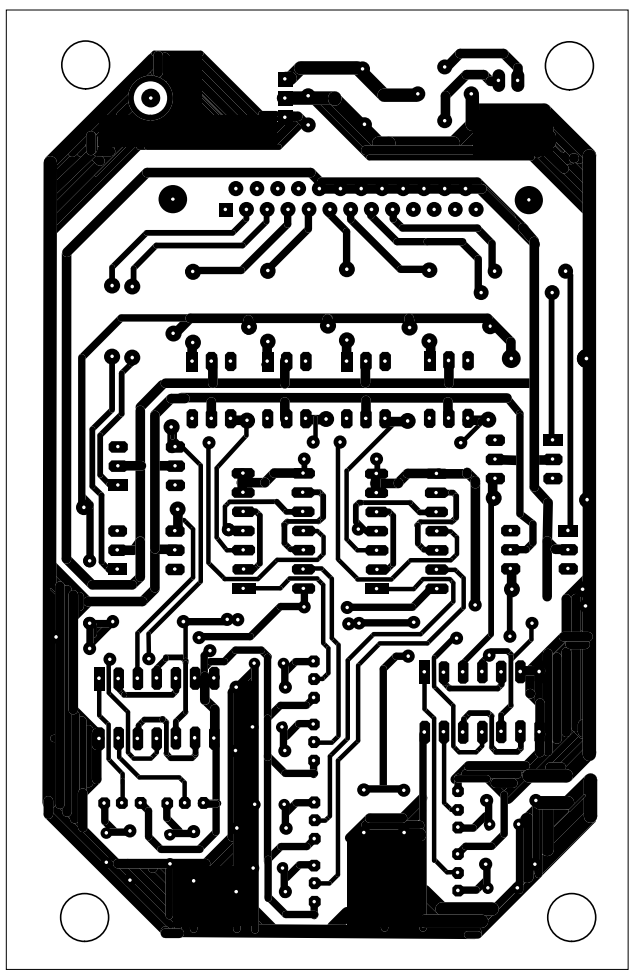

Fig. 4: Combined actual-size, single-side PCB layout for computerised electrical equipment control and power supply circuits

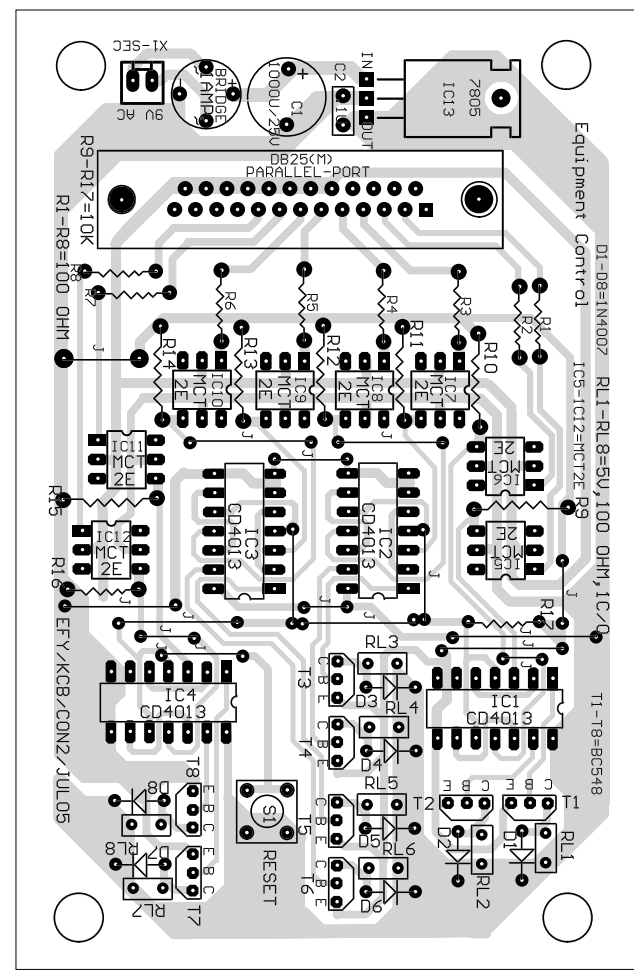

Fig. 5: Component layout for the PCB

In case you press any invalid key (not used in the software), the message "Invalid key pressed wait 2 seconds" is displayed followed by the main screen.

The combined single-side PCB layout for the equipment control and power supply circuits is shown in Fig. 4 and and its component layout in Fig. 5.

Download source code: http://www.efymag.com/admin/issuepdf/Equipment%20Control.zip

CONTROL.C

```
/*COMPUTERISED ELECTRICAL EQUIPMENT CONTROL*/
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
void tone(void);
int p=0x0378;
char ex[23]={"Created by V.MARIYAPPAN"};
int j;
char ex1[34]={"For Further Details & Improvements"};
int k;
char ex2[40]={"Contact: Email-marietech2003@yahoo.co.in"};
int l;
char ex3[23]={"Programming Language: C"};
int m;
int u[10];
int i;
static a,b,c,d,e,f,g,h;
char no;
clrscr();
textcolor(15);gotoxy(20,6);
cprintf("COMPUTERISED ELECTRICAL EQUIPMENT CONTROL");
textcolor(11);gotoxy(20,7);
cprintf("----------------------------------------");
textcolor(11);gotoxy(10,10);
cprintf("EQUIPMENT NO: 1 2 3 4 5 6 7 8");
textcolor(11);gotoxy(10,12);
cprintf("STATUS: %d %d %d %d %d %d %d %d",a,b,c,d,e,f,g,h);
textcolor(10);gotoxy(9,16);
cprintf("FOR 'ON' AND 'OFF' AN EQUIPMENT PRESS CORRESPONDING EQUIP.NO.");
textcolor(11);gotoxy(28,18);
cprintf("STATUS 0=OFF STATUS 1=ON");
textcolor(12);gotoxy(32,20);
cprintf("FOR EXIT PRESS 'E'\n");
no=getch();
switch(no)
 {
 case '1':
 a=!a;
 tone();
 outportb(p,1);
 delay(500);
 outportb(p,0);
 break;

 case '2':
 b=!b;
 tone();
 outportb(p,2);
 delay(500);
 outportb(p,0);
 break;

 case '3':
 c=!c;
 tone();
 outportb(p,4);
 delay(500);
 outportb(p,0);
 break;

 case '4':
 d=!d;
 tone();
 outportb(p,8);
 delay(500);
 outportb(p,0);
 break;

 case '5':
 e=!e;
 tone();
 outportb(p,16);
 delay(500);
 outportb(p,0);
 break;

 case '6':
 f=!f;
 tone();
 outportb(p,32);
 delay(500);
 outportb(p,0);
 break;

 case '7':
 g=!g;
 tone();
 outportb(p,64);
 delay(500);
 outportb(p,0);
 break;

 case '8':
 h=!h;
 tone();
 outportb(p,128);
 delay(500);
 outportb(p,0);
 break;

 case 'e':
 if((a|b|c|d|e|f|g|h)==1)
 {
 clrscr();
 textcolor(10);gotoxy(20,12);
  cprintf("PLEASE SHUT DOWN ALL THE EQUIPMENTS");
 sound(200);
 delay(500);
 nosound();
 delay(3000);
 break;
 }
 else
 {
 clrscr();
 for(j=0;j<23;j++)
 {
 textcolor(10);gotoxy(20+j,12);
 cprintf("%c",ex[j]);
 sound(3000+j);
 delay(30);
 nosound();
 }
 for(m=0;m<23;m++)
 {
 textcolor(10);gotoxy(20+m,13);
 cprintf("%c",ex3[m]);
 sound(1800+m);
 delay(30);
 nosound();
 }
 for(k=0;k<34;k++)
 {
 textcolor(10);gotoxy(20+k,14);
 cprintf("%c",ex1[k]);
 sound(2000+k);
 delay(30);
 nosound();
 }
 for(l=0;l<40;l++)
 {
 textcolor(10);gotoxy(20+l,15);
 cprintf("%c",ex2[l]);
 sound(2500+l);
 delay(30);
 nosound();
 }
 printf("\n\n\n\nPress any key");
 getch();
 outportb(p,0);exit(0);
 }

 case 'E':
 if((a|b|c|d|e|f|g|h)==1)
 {
 clrscr();
 textcolor(10);gotoxy(20,12);
  cprintf("PLEASE SHUT DOWN ALL THE EQUIPMENTS");
 sound(200);
 delay(500);
 nosound();
 delay(3000);
 break;
 }
 else
 {
 clrscr();
 for(j=0;j<23;j++)
 {
 textcolor(10);gotoxy(20+j,12);
 cprintf("%c",ex[j]);
 sound(2500+j);
 delay(30);
 nosound();
 }
 for(m=0;m<23;m++)
 {
 textcolor(10);gotoxy(20+m,13);
 cprintf("%c",ex3[m]);
 sound(3500+m);
 delay(30);
 nosound();
 }
 for(k=0;k<34;k++)
 {
 textcolor(10);gotoxy(20+k,14);
 cprintf("%c",ex1[k]);
 sound(3000+k);
 delay(30);
 nosound();
 }
 for(l=0;l<40;l++)
 {
 textcolor(10);gotoxy(20+l,15);
 cprintf("%c",ex2[l]);
 sound(3500+l);
 delay(30);
 nosound();
 }
 printf("\n\n\n\nPress any key");
 getch();
 outportb(p,0);exit(0);

 }
 default:
 clrscr();
 sound(500);
 delay(100);
 nosound();
 textcolor(11);gotoxy(30,12);
 cprintf("INVALID KEY PRESSED");
 textcolor(11);gotoxy(33,14);
 cprintf("WAIT 2 SECONDS");
 delay(3000);
 break;
 }
 main();
 }
 void tone(void)
 {
 sound(1000);
 delay(100);
 nosound();
 }
```