

# PC-BASED AUTOMATIC TIME MANAGER

■ ADEEB RAZA

Many articles have been published in EFY for controlling devices through a PC's parallel port or line printer (LPT) port. Here is another project for controlling a school bell or a scheduled shift-timing alarm of a factory automatically. The load connected to the output of the project can be an electric bell or a hooter. The timing can be programmed for every day schedule as per individual requirement.

A Windows-based program developed in Microsoft Visual Basic pro-

gramming language is used for controlling an electric bell or hooter through an electromagnetic relay. It has been designed for Windows XP.

The graphics have been included so that the master clock shows the real clock time on the PC's screen. At the same time, it shows alarm clock timing for triggering the timer circuit for activating the bell or hooter.

The alarm timings, 'on'/'off' controls and their activation sequence can be set by the user as per his requirement through 48 numbers of settings—that is, 24 settings for alarm time display on the screen and 24 for enabling or disabling the

timer circuit. The alarm time as well as bell or hooter 'on'/'off' timer can be set by simple clicking the bigger rectangular box

and a small square box on the top of the rectangular box, respectively. The program output user interface is shown here in the screenshot.

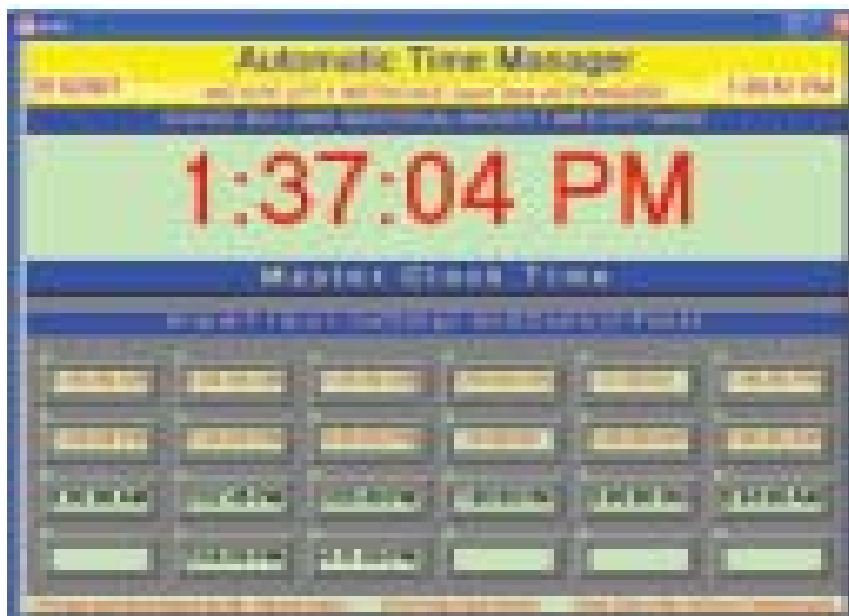
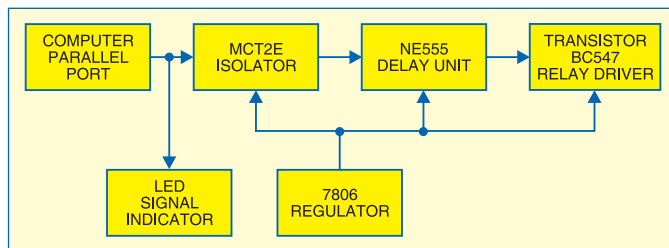
## Hardware interface

Fig. 1 shows the block diagram of the automatic time manager (ATM) and Fig. 2 shows the circuit diagram of the interfacing unit. Here, we are concerned with only two connections: pin 2 and a ground pin of the parallel port. Pin 2 corresponds to the first data bit 'D0' in 8-bit data bus of the parallel port.

The whole unit is powered by a 6V DC regulated supply. LED1 is the power-'on' indicator.

IC MCT2E isolates the PC from the main interfacing unit in case any short-circuit occurs. Thus, it acts as a safety device and protects the PC.

IC LM555 (IC2) timer is configured in monostable mode of operation. A low transition at its pin 2 will trigger the timer. IC2 timer can be triggered



## PARTS LIST

### Semiconductor:

IC1	- 7806 6V regulator
IC2	- 555 timer
IC3	- MCT2E optocoupler
LED1-LED3	- 5mm light-emitting diode
T1	- BC547 npn transistor
D1-D5	- 1N4007 rectifier diode

### Resistors (all 1/4-watt, ±5% carbon):

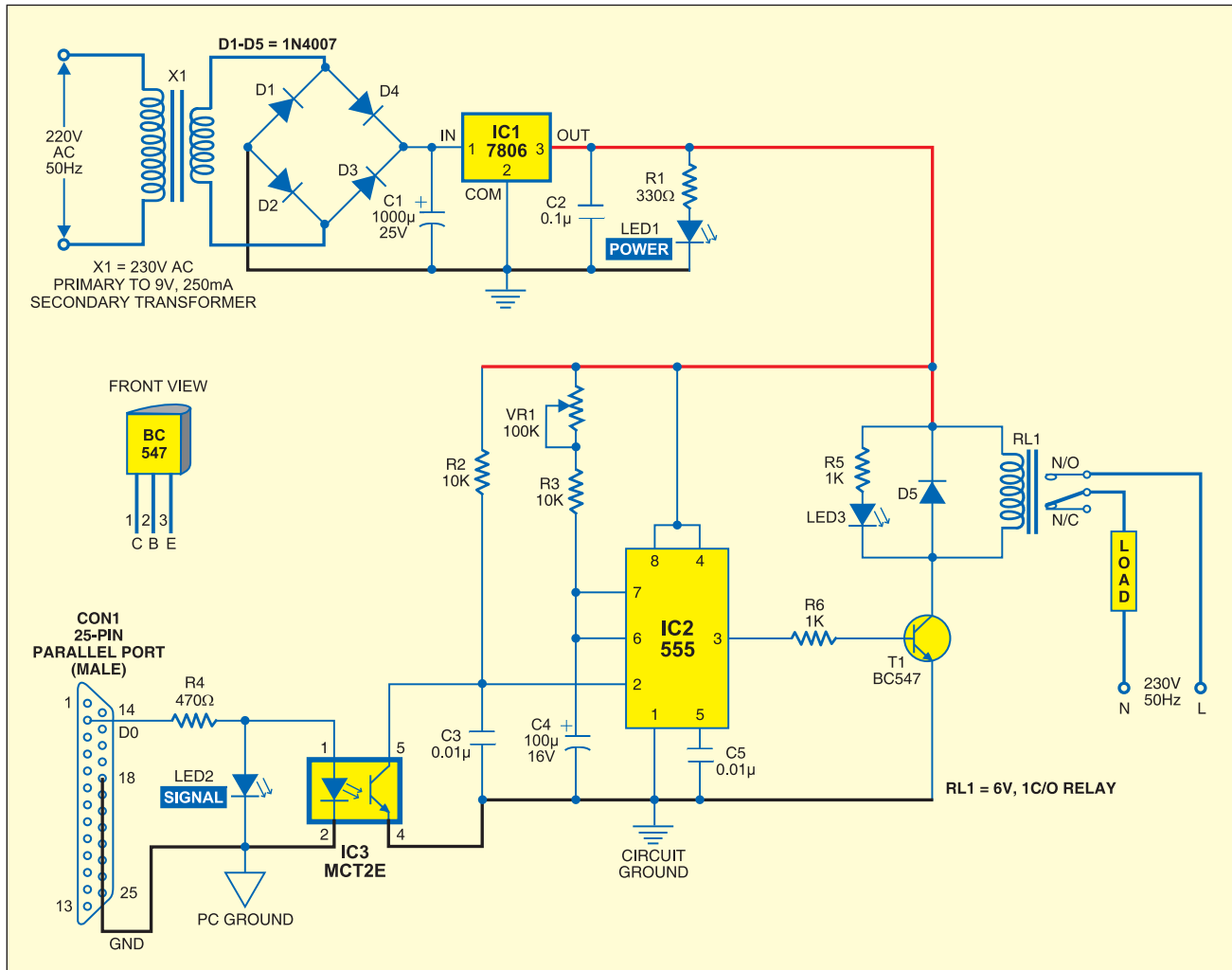
R1	- 330-ohm
R2,R3	- 10-kilo-ohm
R4	- 470-ohm
R5, R6	- 1-kilo-ohm
VR1	- 100-kilo-ohm

### Capacitor:

C1	- 100µF, 25V electrolytic
C2	- 0.1µF ceramic
C3, C5	- 0.01µF ceramic
C4	- 100µF, 16V electrolytic

### Miscellaneous:

X1	- 230V AC primary to 9V, 250mA secondary transformer
CON1	- 25-pin D-type parallel-port male connector
RL1	- 6V, 1C/O PCB-mounted relay



only if the alarm time is set to '1' in enable condition in the software. If the alarm time is set to enable condition, you will get a one-second long beep sound from the PC's speaker and a high pulse at pin 2 of the parallel port of the PC. A high pulse output at pin 2 of the LPT port will cause momentary glowing of signal LED2 and conduction of transistor inside optocoupler IC MCT2E. The conduction of this transistor sends a low pulse to pin 2 of 555 timer IC2.

When timer IC2 gets triggered, its output pin 3 goes high and holds the relay for 30 seconds. Output pin 3 of timer IC2 is connected to relay driver BC547. The load is controlled through the relay contacts to switch 'on' or 'off.'

The load-'on' time duration can be set through a 100-kilo-ohm potentiom-

eter (VR1) upto 60 seconds or more. The load (school bell or hooter) is connected to the pole of the relay. When the relay energises, the pole comes in contact with the N/O, and connects 230V AC across the load. This turns on the load for a brief predetermined time set through 555 timer circuit.

An actual-size, single-side PCB layout for the PC-based automatic time manager circuit is shown in Fig. 3 and its component layout in Fig. 4.

### Software program

In the program, master clock time or real-time clock is represented by 'Time,' alarm times are represented by AT1 through AT24, and timer enable/disable for alarm times are represented as T1 to T24, respectively. AT1 through AT24, and T1 through T24, are stored

in a file name called 'HTIME.' This data is fixed and continuously compared with the actual time of master clock.

Whenever master clock time is equal to alarm time and alarm time is set to enable condition, you will get a one-second long beep sound from the PC's speaker and a high pulse at pin 2 of the 25-pin D-type parallel port connector of the PC. This pulse is used to control the load connected across the relay in the interfacing circuit.

The logic of the ATM program is very easy:

Compare Master Clock Time and Set Alarm Time

If T1=1, Alarm Time No.1 is enabled. If T1=0, Alarm Time No.1 is disabled.

A sample program to display

the first alarm timer setting is given below:

```
Label4.Caption = AT1
If AT1 = Time then T = 1
If T + T1 = 2 then pat = 1 'This will give a beep
sound through PC speaker'
AEPData = Val(pat) 'Set PAT= 0 or PAT=1'
```

Here, when PAT= 0, the load will be in 'off' condition, and when PAT=1, the load will be in 'on' condition. The statement 'AEPData (Data)' will give a high output at pin 2 of the LPT port.

For details on LPT port, you may refer to 'Device Control Through PC's Parallel Port Using Visual Basic' project published in EFY's March 2005 issue.

### Testing of software and hardware interface

1. Install Microsoft Visual Basic 6 on your computer.
2. Fabricate the interface circuit on the PCB.
3. Connect the male connector pin 2 (D0) and ground (pins 18 through 25) to female connector on the PC.
4. Launch Visual Basic from the desktop and develop the application as shown in the screenshot. Save the project file with extension '.vbp'
5. Run the program
6. Set master clock time and alarm time.
7. Click ATM setting to know more about setting and help.

Alternatively, you can use 'ATMINSTALL.zip' file included in the EFY-CD. Unzip ATMINSTALL.zip file and you will get a folder name ATM-Win-Installer. In this folder, you will get a setup program. Clicking 'setup' file will install this project automatically in 'Program Files' folder of Windows

with the name 'Project1' folder along with other relevant files and drivers.

You can also create a Win-Installer program as follows: Open Microsoft Visual Studio folder. In the folder 'VB98,' you will get another folder 'Wizards,' which contains another folder 'PDWizard.' In folder 'PDWizard,' you will get a program file named 'PDCMDLN.EXE.' Clicking the PDCMDLN.EXE will create Win-install software for any of your Visual Basic projects.

In case you have additional software and services installed in your PC with Windows XP, you may observe a problem that pin 2 of the parallel port is high every five seconds. To solve the problem, there is a 'SuperProserver' services in the system under 'Control Panel / Administrative Tools' which has to be disabled before running the program.

**EFY.** All the relevant files of this project have been included in this month's EFY-CD. ●

