REMOTE-CONTROLLED DIGITAL AUDIO PROCESSOR

■ **KULAJIT SARMA**

These days most audio systems come with remote controllers. However, no such facility is provided for normal audio amplifiers. Such audio controllers are not available even in kit form. This article presents an infrared (IR) remote-controlled digital audio processor. It is based on a microcontroller and can be used with any NEC-compatible full-function IR remote control.

This audio processor has enhanced features and can be easily customised to meet individual requirements as it is programmable. Its main features are:

1. Full remote control using any NEC-compatible IR remote control handset
2. Provision for four stereo input channels and one stereo output
3. Individual gain control for each input channel to handle different sources
4. Bass, midrange, treble, mute and attenuation control
5. 80-step control for volume and 15-step control for bass, midrange and treble
6. Settings displayed on two 7-segment light-emitting diode (LED) displays and eight individual LEDs
7. Stereo VU level indication on 10-LED bar display
8. Full-function keys on-board for audio amplifier control
9. All settings stored on the EEPROM
10. Standby mode for amplifier power control

Circuit description

Fig. 1 shows the block diagram of the remote-controlled digital audio processor. The system comprises Atmel's AT89C51 microcontroller (IC1), TDA7439 audio processor from SGS-Thomson (IC4) and I²C bus compatible MC24C02 EEPROM (IC5). The microcontroller chip is programmed to control all the digital processes of the system. The audio processor controls all the audio amplifier functions and is compatible with I²C bus. All the commands from the remote control are received through the IR sensor. The audio amplifier can also be controlled using the on-board keys.

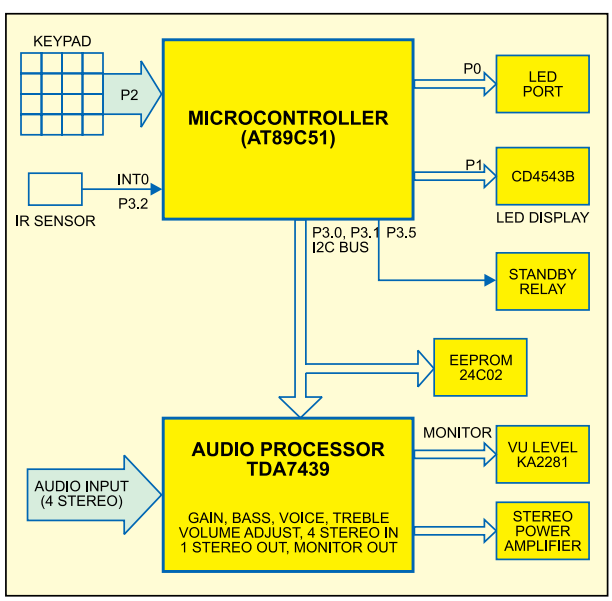

Fig. 1: Block diagram of the remote-controlled digital audio processor

Microcontroller. The function of the microcontroller is to receive commands (through port P3.2) from the remote handset, program audio controls as per the commands and update the EEPROM. A delay in updating the EEPROM is deliberately provided because normally the listener will change the value of a parameter continuously until he is satisfied.

The 40-pin AT89C51 microcontroller has four 8-bit input/output (I/O) ports.

Port 0 is used for indicating through LEDs the various functions selected via the remote/on-board keys.

PARTS LIST

Semiconductors:

IC1	-	AT89C51 microcontroller
IC2, IC3	-	CD4543 7-segment decoder/driver
IC4	-	TDA7439 audio processor
IC5	-	MC24C02 I²C EEPROM
IC6	-	KA2281 2-channel level meter driver
IC7	-	TSOP1238 IR receiver module
IC8	-	7809 9V regulator
IC9	-	7805 5V regulator
IC10	-	LM317 variable regulator
T1	-	BC558 pnp transistor
T2, T3, T5	-	BC547 npn transistor
T4	-	BD139 pnp transistor
BR1	-	W04M bridge rectifier
D1-D6	-	1N4004 rectifier diode
DIS1, DIS2	-	LTS543 7-segment display
DIS3	-	10-LED bargraph display
LED1-LED8	-	Red LED
LED9	-	Green LED

Resistors (all ¼-watt, ±5% carbon):

R1	-	8.2-kilo-ohm
R2-R24, R40-R49	-	1-kilo-ohm
R25, R28, R50, R53	-	10-kilo-ohm
R26, R29, R30, R34	-	2.7-kilo-ohm
R27	-	100-ohm
R31, R35	-	5.6-kilo-ohm
R32, R33	-	4.7-kilo-ohm
R36-R39	-	22-kilo-ohm
R51	-	220-kilo-ohm
R52	-	2.2-kilo-ohm

Capacitors:

C1, C2	-	33pF ceramic disk
C3, C10	-	10µF, 16V electrolytic
C4-C6, C39-C41	-	100nF ceramic disk
C7	-	4.7µF, 16V electrolytic
C8, C9	-	2.2µF, 16V electrolytic
C11, C20	-	5.6nF polyester
C12, C19	-	18nF polyester
C13, C18	-	22nF polyester
C14, C17	-	100nF polyester
C21-C28	-	0.47µF polyester
C29-C32	-	4.7µF, 25V electrolytic
C33, C34	-	10µF, 25V electrolytic
C35	-	1000µF, 25V electrolytic
C36	-	4700µF, 25V electrolytic
C37, C38	-	0.33µF ceramic disk
C42	-	470µF, 25V electrolytic

Miscellaneous:

X1	-	230V AC primary to 12V, 1A secondary transformer
RL1	-	9V, 160Ω, 2 C/O relay
X_{TAL}	-	12MHz crystal
S1-S7	-	Push-to-on switch
S8	-	On/Off switch
Remote	-	Creative's remote (NEC-compatible format)

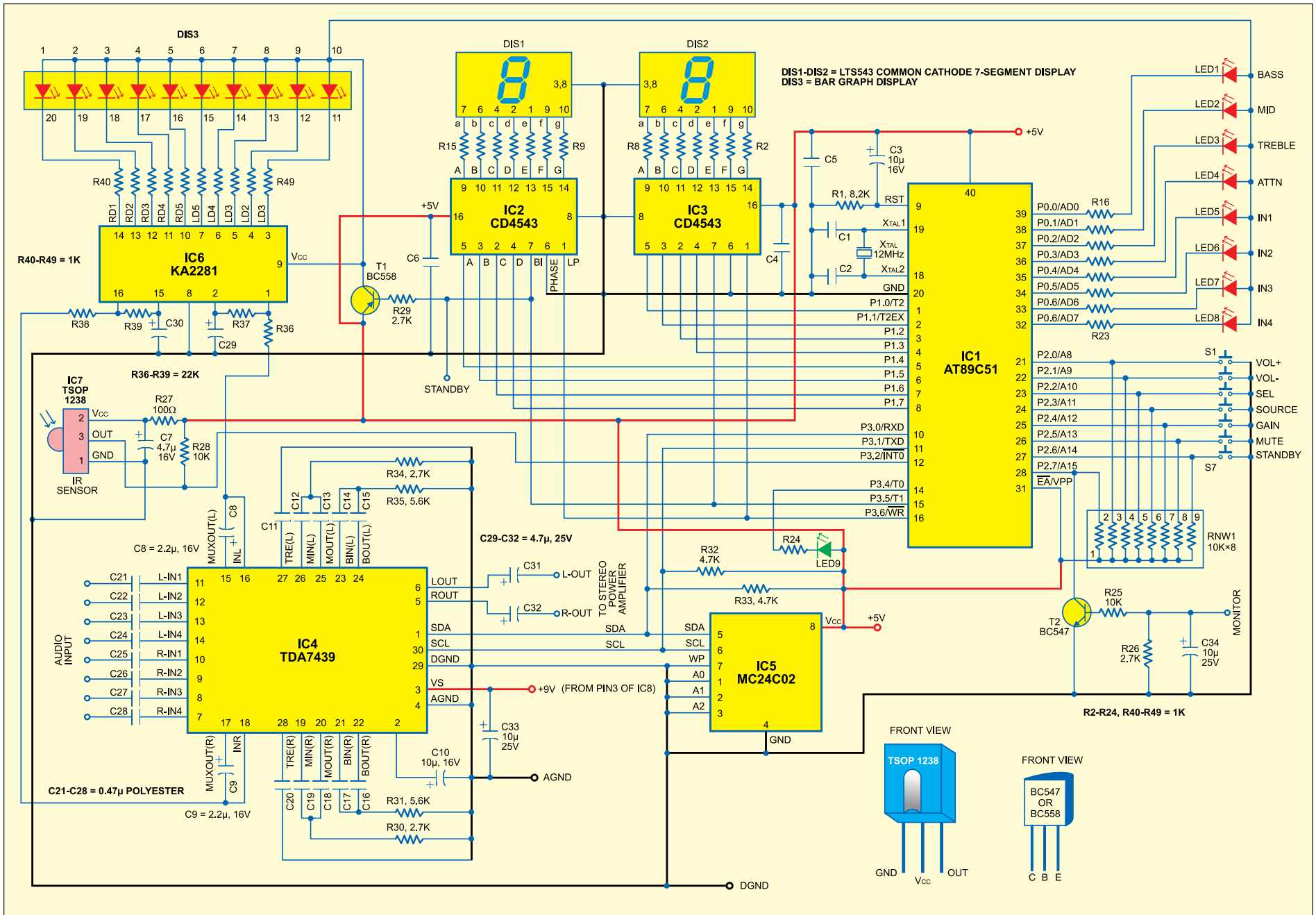

Fig. 2: Circuit diagram of the remote-controlled digital audio processor

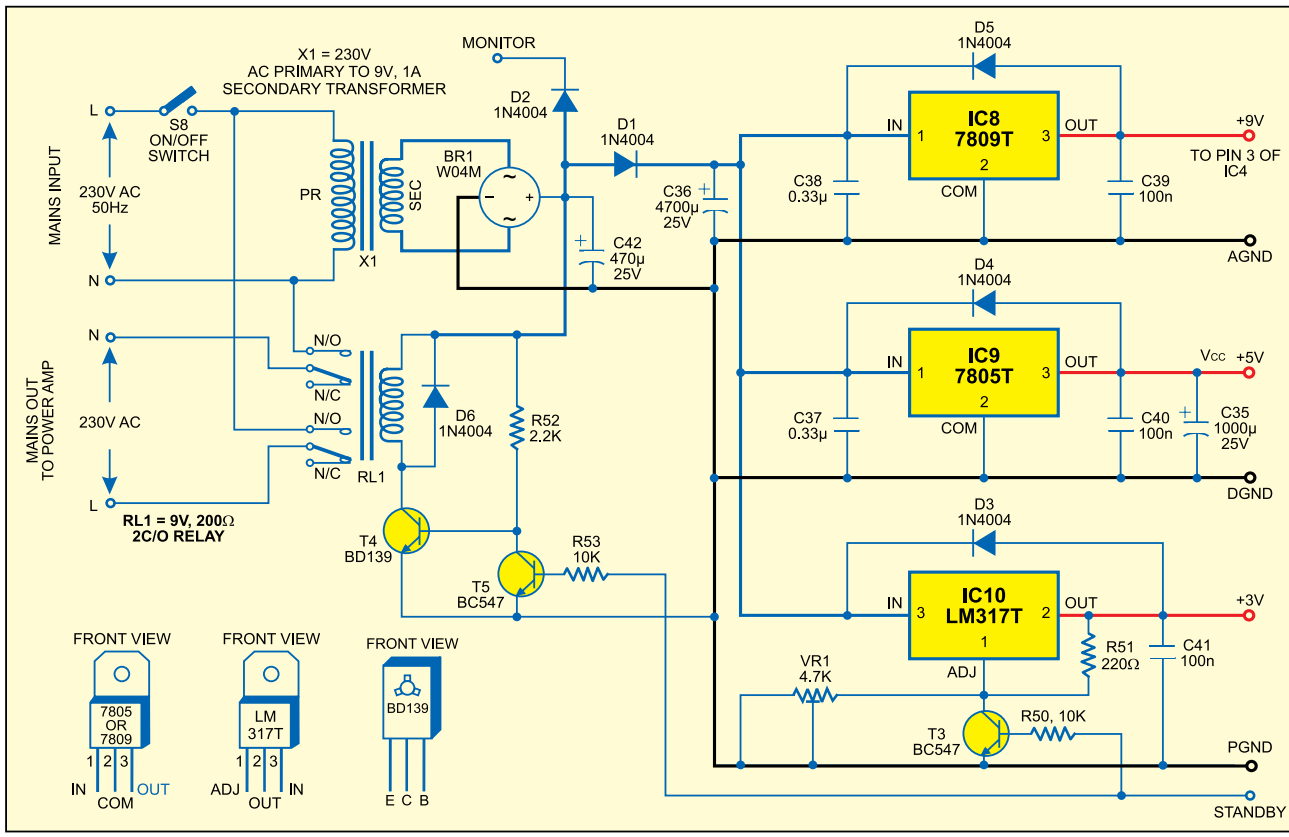

Fig. 3: Power supply

Port 1 drives the 7-segment display using 7-segment latch/decoder/driver IC CD4543.

Port 2 is pulled up via resistor network RNW1 and used for manual key control.

Pins P3.0 and P3.1 of the microcontroller are used as serial data (SDA) and serial clock (SCL) lines for the I²C bus for communicating with the audio processor (TDA7439) and EEPROM (MC24C02). These two lines are connected to pull-up resistors, which are required for I²C bus devices. P3.2 receives the remote commands through the IR receiver module. Pin P3.4 is used for flashing LED9 whenever a remote command is received or any key is pressed.

The microcontroller also checks the functioning of the memory (MC24C02) and the audio processor (TDA7439). If it is not communicating with these two ICs on the I²C bus, it flashes the volume level on the 7-segment displays.

Memory. IC MC24C02 is an I²C-bus compatible 2k-bit EEPROM organised as 256×8-bit that can retain data for more than ten years. Various parameters can be stored in it.

To obviate the loss of latest settings in the case of power failure, the microcontroller stores all the audio settings of the user in the EEPROM. The memory ensures that the microcontroller will read the last saved settings from the EEPROM when power resumes. Using SCL and SDA lines, the microcontroller can read and write data for all the parameters.

For more details on I²C bus and memory interface, please refer to the MC24C02 datasheet. Audio parameters can be set using the remote control handset or the on-board keys as per the details given under the 'remote control' section.

Audio processor. IC TDA7439 is a single-chip I²C-bus compatible audio controller that is used to control all the functions of the audio amplifier. The output from any (up to four) stereo preamplifier is fed to the audio processor (TDA7439). The microcontroller can control volume, treble, bass, attenuation, gain and other functions of each channel separately. All these parameters are programmed by the microcontroller using SCL and SDA lines, which it shares with the memory IC and the audio processor.

Data transmission from the microcontroller to the audio processor (IC TDA7439) and the memory (MC24C02) and vice versa takes place through the two-wire I²C-bus interface consisting of SDA and SCL, which are connected to P3.0 (RXD) and P3.1 (TXD) of the microcontroller, respectively. Here, the microcontroller unit acts as the master and the audio processor and the memory act as slave devices. Any of these three devices can act as the transmitter or the receiver under the control of the master.

Some of the conditions to communicate through the I²C bus are:

1. Data validity: The data on the SDA line must be stable during the high period of the clock. The high and low states of the data line can change

only when the clock signal on the SCL line is low.

2. Start and Stop: A start condition is a high-to-low transition of the SDA line while SCL is high. The stop condition is a low-to-high transition of the SDA line while SCL is high.

3. Byte format: Every byte transferred on the SDA line must contain eight bits. The most significant bit (MSB) is transferred first.

4. Acknowledge: Each byte must be followed by an acknowledgement bit. The acknowledge clock pulse is generated by the master. The transmitter releases the SDA line (high) during the acknowledge clock pulse. The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains low during the high period of this clock pulse.

To program any of the parameters, the following interface protocol is used for sending the data from the microcontroller to TDA7439. The interface protocol comprises:

1. A start condition (S)

2. A chip address byte containing the TDA7439 address (88H) followed by an acknowledgement bit (ACK)

3. A sub-address byte followed by an ACK. The first four bits (LSB) of this byte indicate the function selected (e.g., input select, bass, treble and volume). The fifth bit indicates incremental/non-incremental bus (1/0) and the sixth, seventh and eighth bits are 'don't care' bits.

4. A sequence of data followed by an ACK. The data pertains to the value for the selected function.

5. A stop condition (P)

In the case of non-incremental bus, the data bytes correspond only to the function selected. If the fifth bit is high, the sub-address is automatically incremented with each data byte. This mode is useful for initialising the device. For actual values of data bytes for each function, refer to the datasheet of TDA7439.

Similar protocol is followed for sending data to/from the microcontroller to MC24C02 EEPROM by using its chip address as 'A0H'.

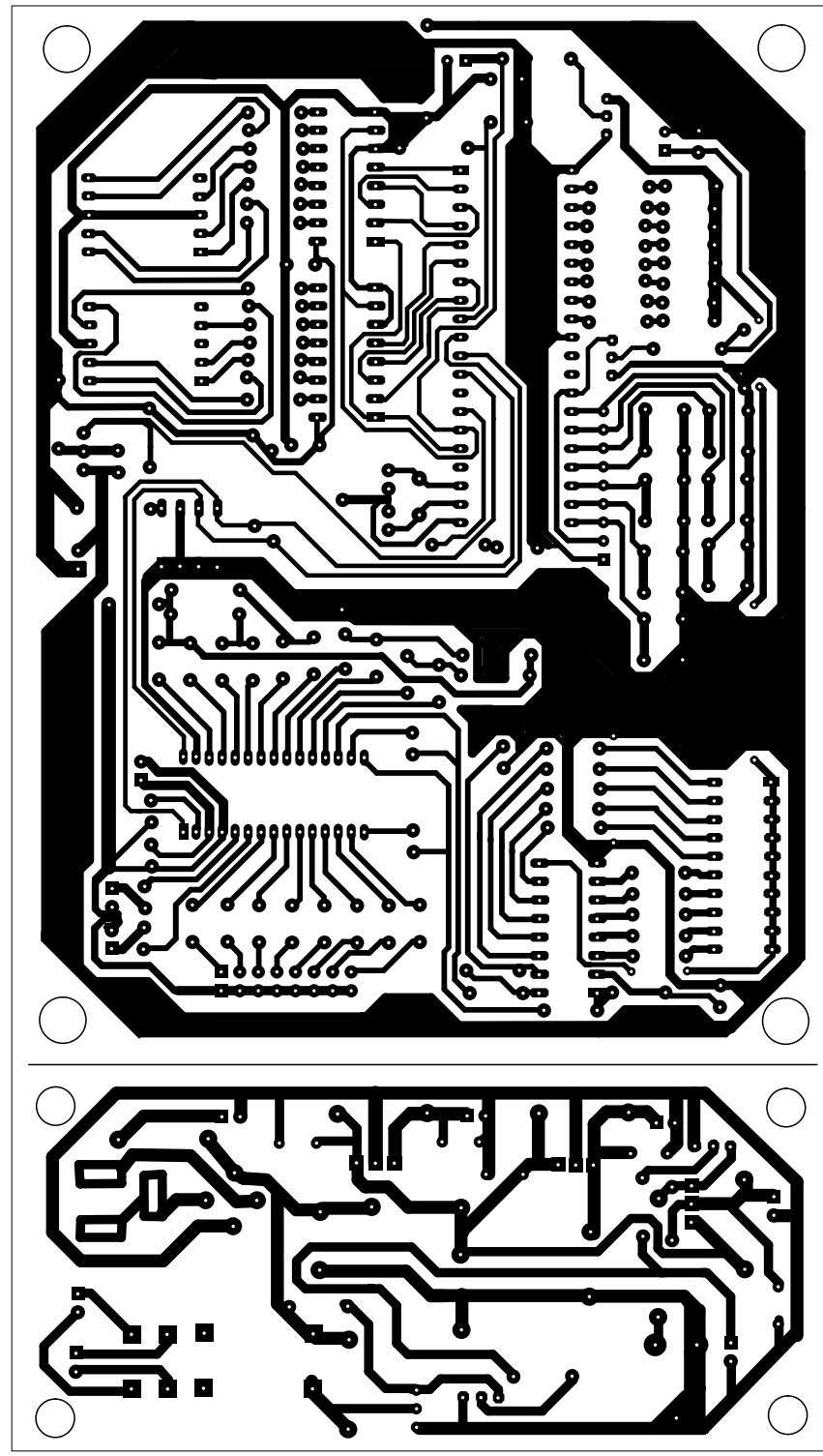

Fig. 4: Combined actual-size, single-side PCB for the remote-controlled digital audio processor (Fig. 2) and power supply (Fig. 3)

Power supply. Fig. 3 shows the power supply circuit for the remote-controlled digital audio processor. The AC mains is stepped down by transformer X1 to deliver a secondary output of 9V AC at 1A. The transformer output is rectified by full-wave bridge rectifier BR1 and filtered by capacitor C42. Regulators IC8 and IC9 provide regulated 5V and 9V power supplies, respectively. IC10 acts as the variable power supply regulator. It is set to pro-

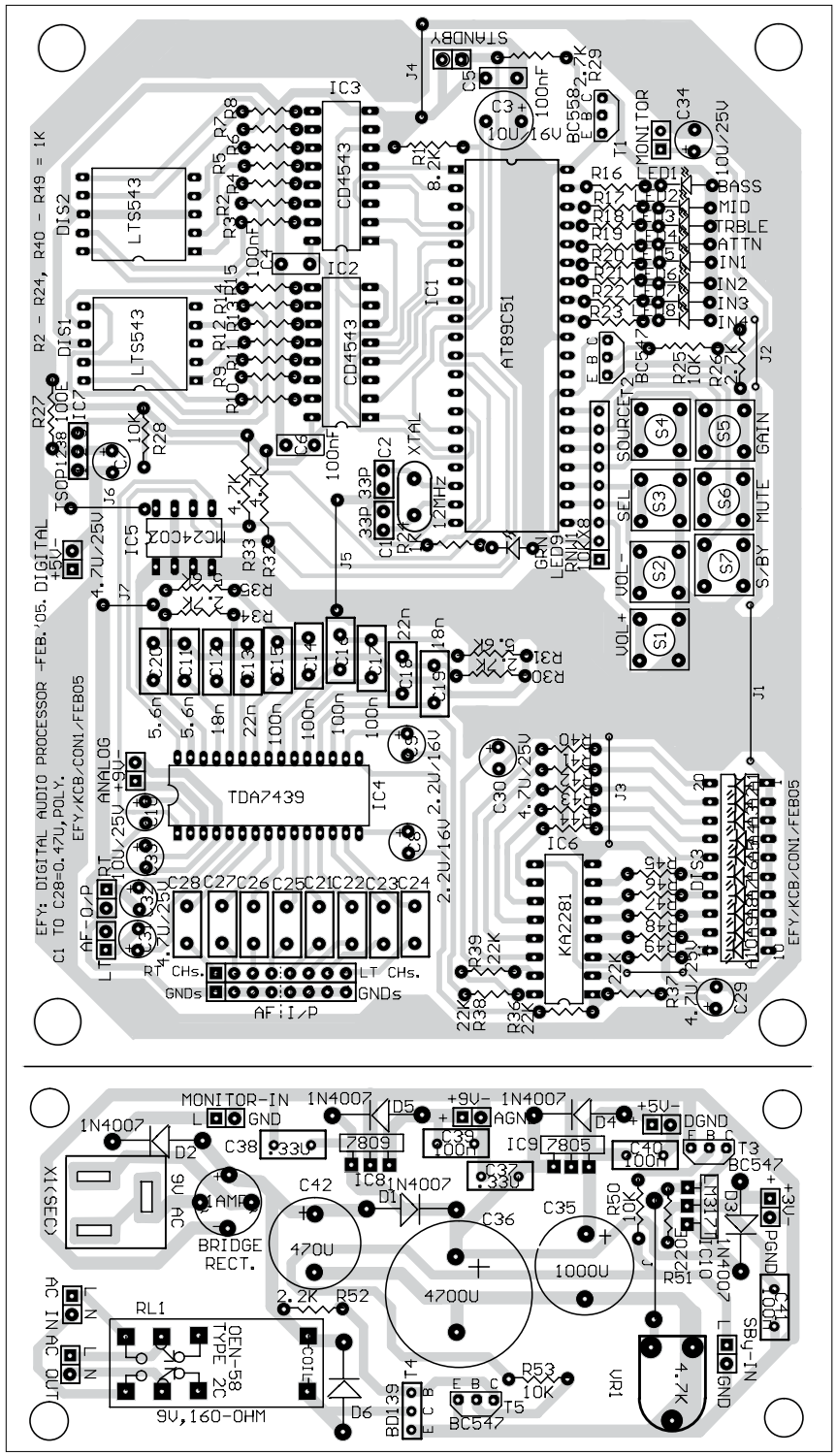

Fig. 5: Component layout for the PCB of Fig. 4

vide 3V regulated supply by adjusting preset VR1. Capacitors C39, C40 and C41 bypass any ripple in the regulated outputs. This supply is not used in the circuit. However, the readers can use the same for powering devices like a Walkman.

As capacitors above 10 µF are connected to the outputs of regulator ICs, diodes D3 through D5 provide protection to the regulator ICs, respectively, in case their inputs short to ground. Relay RL1 is normally energised to provide mains to the power amplifier. In standby mode, it is de-energised. Switch S2 is the 'on'/'off' switch.

Software

The software was assembled using Metalink's ASM51 assembler, which is freely available for download. The source code has been extensively commented for easier understanding. It can be divided into the following segments in the order of listing:

1. Variable and constant definitions
2. Delay routines
3. IR decoding routines
4. Keyboard routines
5. TDA7439 communication
6. MC24C02 communication
7. I²C bus routines
8. Display routines
9. IR and key command processing
10. Timer 1 interrupt handler
11. Main program

On reset, the microcontroller executes the main program as follows:

1. Initialise the microcontroller's registers and random-access memory (RAM) locations.

2. Read Standby and Mute status from the EEPROM and initialise TDA7439 accordingly.

3. Read various audio parameters from the EEPROM and initialise the audio processor.

4. Initialise the display and LED port.

5. Loop infinitely as follows, waiting for events:

- Enable the interrupts.
- Check the monitor input for AC power-off. If the power goes off, jump to the power-off sequence routine.
- Else, if a new key is pressed, call the DO_KEY routine to process the key. For this, check whether the NEW_KEY bit is set. This bit is cleared after the command is processed.
- Else, if a new IR command is received, call the DO_COM routine to process the remote command. For this, check whether the NEW_COM (new IR command available) bit is set. This bit is cleared after the command is processed.

• Jump to the beginning of the loop.

6. Power-off sequence. Save all the settings to the EEPROM, and turn off the display and standby relay.

Since the output of the IR sensor is connected to pin 12 (INT0) of the microcontroller, an external interrupt occurs whenever a code is received. The algorithm for decoding the IR stream is completely implemented in the 'external interrupt 0' handler routine. This routine sets NEW_COM (02H in bit memory) if a new command is available. The decoded command byte is stored in 'Command' (location 021H in the internal RAM). The main routine checks for NEW_COM bit continuously in a loop. Timer 0 is exclusively used by this routine to determine the pulse timings.

Decoding the IR stream involves the following steps:

1. Since every code is transmitted twice, reject the first by introducing a delay of 85 milliseconds (ms) and start timer 0. The second transmission is detected by checking for no-overflow timer 0. In all other cases, timer 0 will overflow.

2. For second transmission, check the timer 0 count to determine the length of the leader pulse (9 ms). If the pulse length is between 8.1 ms and 9.7 ms, it will be recognised as valid. Skip the following 4.5ms silence.

3. To detect the incoming bits, timer 0 is configured to use the strobe signal such that the counter runs between the interval periods of bits. The value of the counter is then used to determine whether the incoming bit is '0', '1' or 'Stop.' This is implemented in the RECEIVE_BIT routine.

4. If the first bit received is 'Stop,' repeat the last command by setting the NEW_COM bit.

5. Else, receive the rest seven bits. Compare the received byte with the custom code (C_Code). If these don't match, return error.

6. Receive the next byte and compare with the custom code. If these don't match, return error.

7. Receive the next byte and store in 'Command.'

8. Receive the next byte and check whether it is complement value of 'Command.' Else, return error.

9. Receive 'Stop' bit.

10. Set NEW_COM and return from interrupt.

Other parts of the source code are relatively straightforward and self-explanatory.

Remote control. The micro-controller can accept commands from any IR remote that uses NEC transmission format. These remote controllers are readily available in the market and use µPD6121, PT2221 or a compatible IC. Here, we've used Creative's remote handset.

All the functions of the system can be controlled fully using the remote or the on-board keys. By default, the display shows the volume setting and LEDs indicate the channel selected. LED9 glows momentarily whenever a command from the remote is received or any key is pressed.

Function adjustments are detailed below:

1. Volume: Use Vol+/Vol- key to increase/decrease the volume. The volume settings are shown on the two-digit, 7-segment display. Steps can be varied between '1' and '80.'

2. Mute and Standby: Using 'Mute' and 'Standby' buttons, you can toggle the mute and standby status, respectively. If 'Mute' is pressed, the display will show '00.' In 'Standby' mode, the relay de-energises to switch off the main amplifier. All the LEDs and displays, except LED9, turn off to indicate the standby status.

3. Input Select: To select the audio input source, press 'Channel' key until the desired channel is selected. The LED corresponding to the selected channel turns on and the input gain setting for that channel is displayed for five seconds. Thereafter, the volume level is displayed on the 7-segment display.

4. Input Gain set: Press 'Gain' key. The LED corresponding to the channel will start blinking and the gain value is displayed. Use Vol+/Vol- key to increase/decrease the gain for that channel. Note that the gain can be varied from '1' to '15.' If you press 'Gain' key once more, and no key is pressed for five seconds, it will exit the gain setting mode and the volume level is displayed.

5. Audio: Press 'Audio Set' (Menu) key to adjust bass, middle, treble and attenuation one by one. Each time 'Audio Set' key is pressed, the LED corresponding to the selected function turns on and the function value is displayed. Once the required function is selected, use Vol+ and Vol- to adjust the setting. Bass, middle and treble can be varied from '07' to '7.' Values '0' through '7' indicate 'Boost' and '00' through '07' indicate 'Cut.' Attenuation can be varied from '0' to '40.'

Construction

The circuit can be easily assembled on any PCB with IC base. Before you install the microcontroller, memory and audio processor in their sockets and solder the IR receiver module, make sure that the supply voltage is correct. All parts, except the audio processor (TDA7439), require 5V DC supply. The audio processor is powered by 9V DC.

Download source code: http://www.efymag.com/admin/issuepdf/Audio%20Processor.zip ●