Remote-controlled Audio Processor Using Microcontroller

VINAY CHADDHA

EFY TESTED G.S. SAGOO

Most hi-fi audio CD systems nowadays are available with remote control. However, no such circuit is available for adding to the AF power amplifiers made as hobby projects. The design of such an add-on circuit along with remote control is presented here. It can be used by hobbyists as an attachment to their audio power amplifiers.

This design is based on audio controller TDA7315 from SGS-THOMSON and Motorola microcontroller MC68HC705KJ1. The microcontroller, after programming with the specific software code for the current application, has been renamed GVC-AUD-257. Other construction projects by the author using the same series of microcontrollers, which can be referred to for additional details, have been published in EFY in Jun. '97 (Set-Top Converter) and Apr. '99 (Caller ID). In addition to audio controller and microcontroller the circuit comprises the following standard parts that are used in any normal system:

1. Power supply
2. LED indicator panel to indicate status/keys pressed
3. Relay to switch on/off the supply to the power amplifier or the main unit

The following parts are used in remote systems for VCR, TV, etc:

1. Memory (24C02).
2. Remote handset with 12 keys
3. IR receiver for remote operation

Functionality of these parts is explained below with reference to Figs 1 and 3.

The author is the proprietor of GVC Systems, Noida. He can be approached through e-mail at gvc@vsnl.com

Description

Memory. 24C02 is an I2C bus compatible 2k-bit EEPROM, organised as 256 x 8 bits. It can retain data for a period of more than ten years, including the current settings of volume, treble, balance, bass, as well as the on/off status of the main amplifier unit or the relay status. The various audio parameters can be stored in just six bytes.

Mains power failure is quite common in India. This does not allow the last settings of volume, tone and balance to remain intact. To overcome this the microcontroller unit (MCU) must store all audio settings of the user in memory (EEPROM). The memory ensures that even after a power trip, the MCU will read the latest saved settings from the EEPROM.

Using two lines, SCL (serial clock) and SDA (serial data), the microcontroller can read and write six bytes for all the audio parameters. For more details on I2C bus and memory interfacing, please refer to Caller ID construction project in Apr. '99 issue of EFY.

At power 'on,' the last-saved audio settings are read by the MCU. In case memory IC cannot be read by the microcontroller, volume LED will blink three times to indicate the problem. The possible reasons could be either a bad memory IC, or a discontinuity/shorting of its tracks, or improper insertion of the IC in its socket.

Under the circumstances, the unit will still work, but it will not remember the last settings and will select the centre values of treble, balance and bass. Volume will be set at 50 per cent of the maximum value and the relay will be off. Loudness and mute will also remain in the off mode. A remote handset can be used to change the settings as desired.

Audio controller. TDA7315 is a single-chip I2C bus-compatible audio controller which is used to control all functions of the audio amplifier. Stereo audio input from the preamplifier is fed to the IC input. A microcontroller can

PARTS LIST

Semiconductors:	
IC1	- TDA7315 digitally controlled audio processor
IC2	- MC68HC705KJ1CP Motorola microcontroller (GVC-AMP-257)
IC3	- 24C02 I2C serial EEPROM
IC4	- 7805 fixed regulator +5V
IC5	- 7809 fixed regulator +9V
IC6	- µPD6121 infrared remote control
T1,T2,T3	- BC547 npn transistor
T4	- 2SC2001 npn transistor
D0-D7	- Red LED
D8, D10-D11	- 1N4007 rectifier diode
D9	- 8.2V,0.5W zener
D12	- IR LED
Resistors (all 1/4W, ± 5% metal/carbon film, unless stated otherwise)	
R1,R2,R15,R16	- 10-kilo-ohm
R3,R4,R14	- 1-kilo-ohm
R0,R5-R9, R12,R13	- 330-ohm
R10,R11	- 5.6-kilo-ohm
R17	- 2-ohm, 0.5W
R18,R19	- 200-kilo-ohm
Capacitors:	
C1	- 10µF, 16V electrolytic
C2-C4,C7, C19,C20	- 0.1µF ceramic disc
C15,C16,C9, C10,C11,C13	- 0.1µF polyester
C5,C6	- 33pF ceramic disc
C8	- 22µF, 25V electrolytic
C12,C14	- 2.7nF polyster
C17,C18	- 2.2µF, 16V electrolytic
C21,C22	- 220pF ceramic disc
C23	- 10µF, 50V electrolytic
Miscellaneous:	
RL1	- 12V, 150 ohm, SPST relay OEN Pt. No. 57DP-12-1C6
X_{TAL}	- 4MHz quartz crystal
Y	- 455kHz, ceramic resonator
	- Battery 1.5V, pencil cell
	- IR sensor module
	- Remote control handset, complete with keyboard
	- Connectors for audio input-output and power supply
	- 12V DC, 250mA battery eliminator

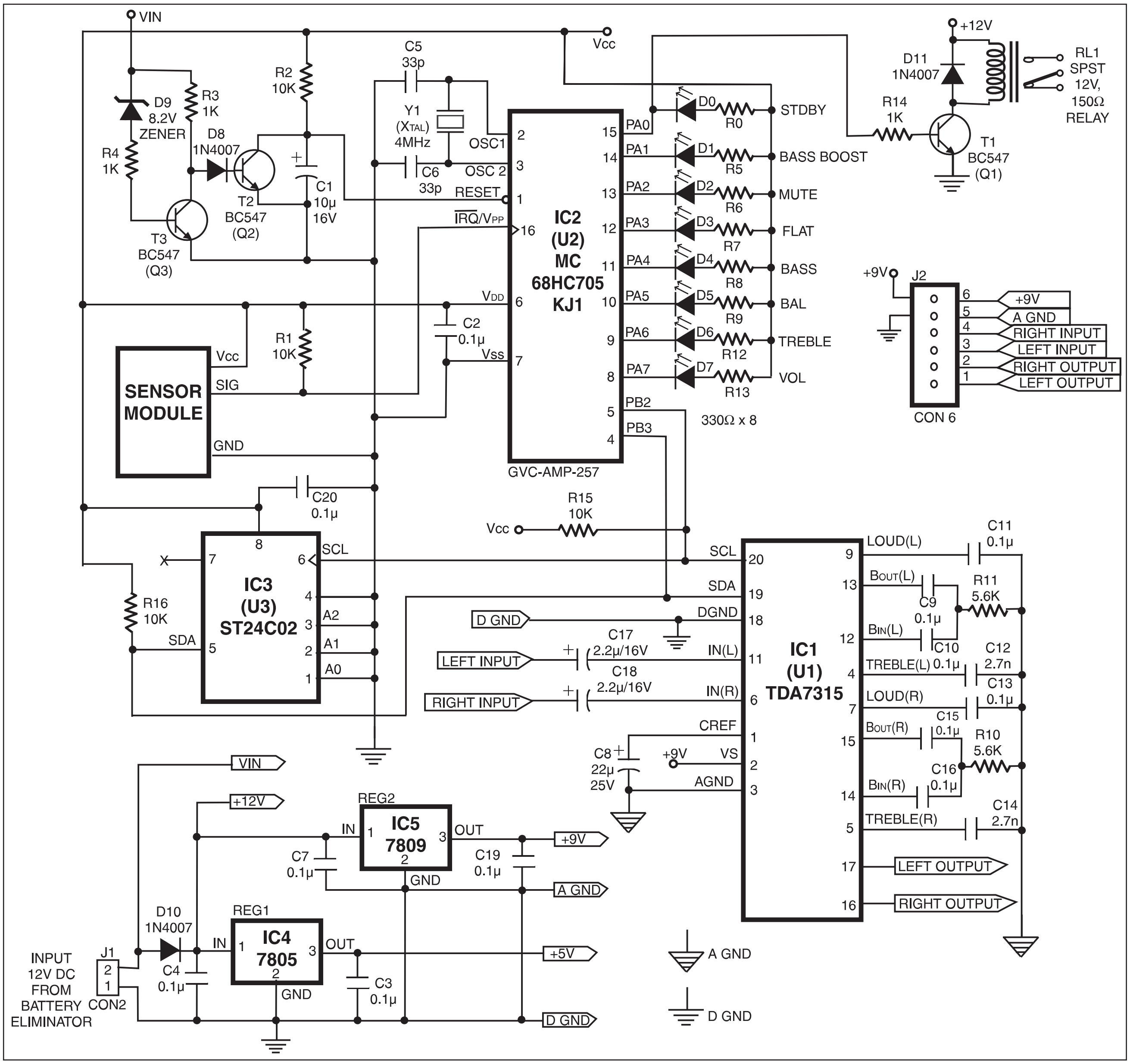

Fig. 1: Schematic diagram of remote-controlled audio processor using microcontroller

control volume, treble, balance, bass and loudness. All these parameters are programmed by microcontroller using SCL and SDA lines, which are the same lines as used for the memory IC, as mentioned earlier. The audio controller reference data is given in Table I.

To program any of the parameters, the following interface protocol is used for sending the data from the MCU to TDA7315. The interface protocol comprises:

- A start condition (S)
- A chip address byte (80H) followed by an acknowledgement bit (ACK)
- A sequence of data bytes, with each byte followed by an acknowledgement bit
- A stop condition

For explanation of start, acknowledgment and stop conditions of I2C protocol, readers may refer to Apr. '99 issue of EFY. A typical message format, comprising an address byte and two data bytes, is shown in Table II.

Address byte and audio parameters of TDA7315, showing the coding as well as weightage of bits representing audio parameters, are shown in Table III. For sending the address and complete functional parameters (six) of Table III, the

TABLE I
The Audio Controller Reference

Parameter	Value
Supply voltage	6 to 10V DC, 9V typically
Max input signal	2 V minimum
Total harmonic distortion	0.01 % typical, 0.1% maximum
Signal-to-noise ratio	106 db typical
Channel separation	103 db @ 1 kHz
Volume control 1.25 db step	0 to –78.75 dB
Bass and treble control 2 db step	+14 to –14 dB
Balance control 1.25 db step	0 to –38.75 dB
Mute attenuation	100 dB typical

TABLE II

	MSB …… TDA7315 …… ADDR…… LSB									MSB …… DATA……. LSB									MSB…… DATA…….. LSB									
S	1	0	0	0	0	0	0	0	ACK	X	X	X	X	X	X	X	X	ACK	X	X	X	X	X	X	X	X	ACK	P

S = start condition ACK = acknowledgement P = stop condition X = any hex digit

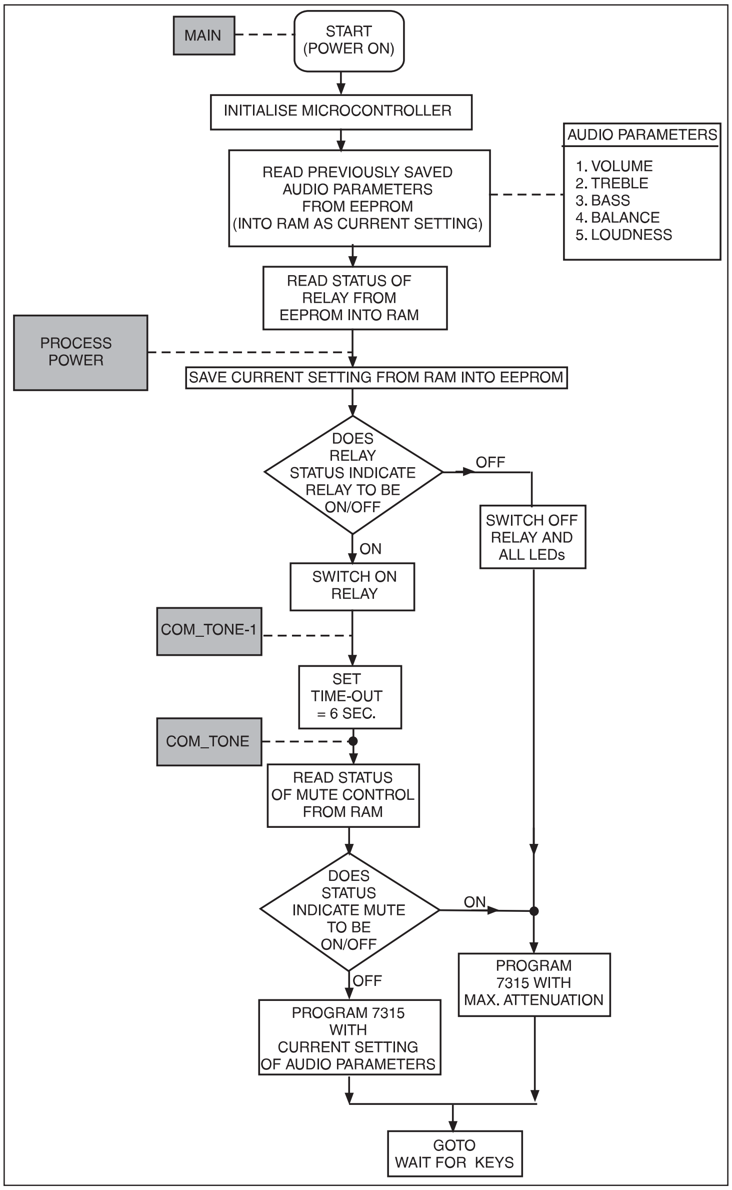

Fig. 2(a): Software flow chart (contd.)

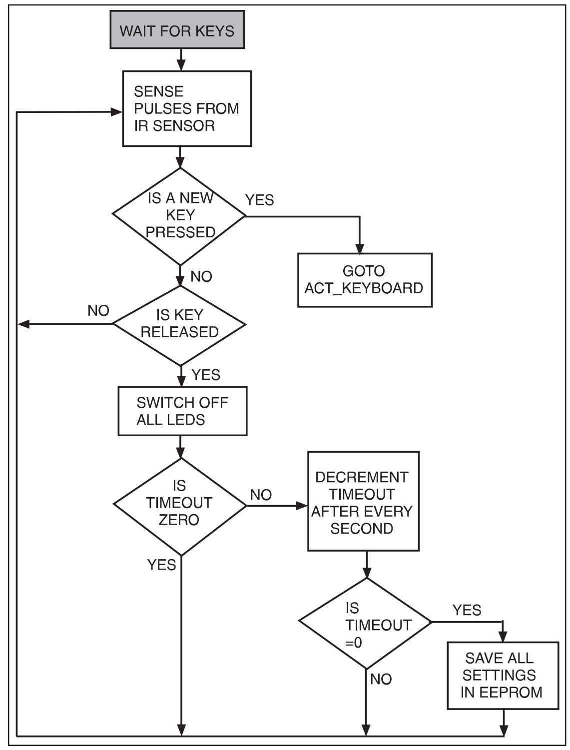

Fig. 2(b): Software flow chart (contd.)

microcontroller has to send seven bytes of data as well as start and stop condition bits through I2C bus. The acknowledgment bit, after receipt of every byte, is sent by the slave unit (TDA7315 in this case).

Details of each audio control byte are discussed below:

Volume-control byte. Any number from 00 to 63 (or 3FH) can be sent (programmed). Minimum attenuation is achieved with 00H. Each increment in number adds attenuation of 1.25 dB.

Speaker-control (L) byte. Any number from 128 to 159 (80H to 9FH) can be sent. Minimum attenuation for left channel is at 128(80H). Each increment in number adds attenuation of 1.25 dB. The maximum attenuation is typically 37.5 dB.

Speaker-control (R) byte. Any number, from 160 to 191 (0A0H to 0BFH), can be sent. Minimum attenuation for the right channel is 160(0A0H). Each increment in number adds an attenuation of 1.25 dB. The maximum attenuation is typically 37.5 dB.

Loudness byte. It has only two pos-

TABLE III

MSB							LSB	FUNCTION
1	0	0	0	0	0	0	0	TDA7315 Chip address = 80H
0	0	B2	B1	B0	A2	A1	A0	Volume control byte
1	0	0	B1	B0	A2	A1	A0	Speaker control (L) byte
1	0	1	B1	B0	A2	A1	A0	Speaker control (R) byte
0	1	0	X	X	L	X	X	Loudness byte
0	1	1	0	C3	C2	C1	C0	Bass control byte
0	1	1	1	C3	C2	C1	C0	Treble control byte

Bx = 10dB steps Ax = 1.25dB steps Cx = 2dB steps X = Don't care

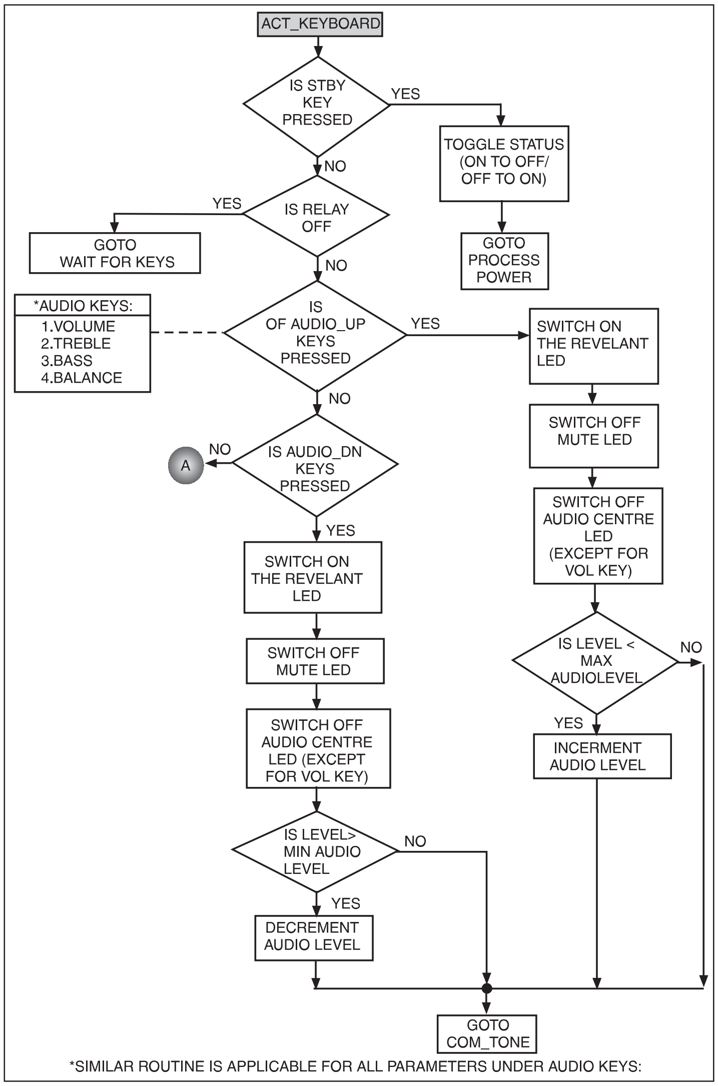

Fig. 2(c): Software flow chart (continued)

sible values. When 64(40H) is sent, it sets the loudness on and when 70(44H) is sent, it sets the loudness off.

Bass control byte. Any number, from 96 to 111 (60H to 6FH), can be sent. This can change the bass range from 14 dB gain to 14dB loss (attenuation).

Treble-control byte. Any number, from 112 to 127 (70H to 7FH), can be sent. This can change the treble value from 14 dB gain to 14 dB attenuation. Table IV gives the gain/attenuation for various values of bass and treble bytes.

TABLE IV

Bass byte	Treble byte	Gain(+) /Attn(-)
96(60H)	112(70H)	-14 dB
97(61H)	113(71H)	-12 dB
98(62H)	114(72H)	-10 dB
99(63H)	115(73H)	-8 dB
100(64H)	116(74H)	-6 dB
101(65H)	117(75H)	-4 dB
102(66H)	118(76H)	-2 dB
103(67H)	119(77H)	0 dB
111(6FH)	127(7FH)	0 dB
110(6EH)	126(7EH)	+2 dB
109(6DH)	125(7DH)	+4 dB
108(6CH)	124(7CH)	+6 dB
107(6BH)	123(7BH)	+8 dB
106(6AH)	122(7AH)	+10 dB
105(69H)	121(79H)	+12 dB
104(68H)	120(78H)	+14 dB

Thus to have volume at 10 dB attenuation with perfectly balanced sound, i.e. both speakers at equal level, loudness on and maximum treble and minimum bass, the message to be sent by the MCU to the TDA7315 on the I2C bus would be as follows:

START-80-08-80-A0-40-78-60-STOP

(*Note:* Normally, in the I2C interface data changes state only when the SCL signal is low. However, 'start' and 'stop' are special conditions, which indicate start of I2C activity and end of I2C activity respectively on the bus. In start condition, SDA goes from high to low when SCL is high. In a stop condition, SDA goes from low to high when SCL is high. Refer Apr. '99 issue of EFY for details.)

Microcontroller is programmed to check this IC (TDA7315) at power on. If the microcontroller cannot communicate with it through the I2C bus, it will indicate the error by flashing treble LED three times at power on. In this case audio values cannot be changed as communication between the microcontroller and the audio controller is not possible. However, other parts of the unit will still function. For instance, the remote unit can be used to switch on and switch off the relay.

Microcontroller. The function of microcontroller is to receive commands from the remote handset, program audio controllers as per the commands, and update the EEPROM if no new command is given from the handset for six seconds. However, for on/off and mute commands, it will update the EEPROM data immediately. The delay in updating the EEPROM is provided as normally the listener will change the value continuously till he is satisfied. The MCU will, however, not transfer all values (from RAM to EEPROM) as and when these are changed. Only when no key is pressed for six seconds, will the MCU assume that the listener is satisfied and save the audio values in the EEPROM.

This 16-pin microcontroller from Motorola (68HC705KJ1), used in this project, has a total of 11 I/O lines/pins. Two pins each are used for power supply and crystal while one pin is used for reset function. Balance lines are available for any programming functions. Out of 11 I/O lines, eight lines are exclusively used for LED outputs. Relay RL1 used for switching the mains supply to the output power amplifier shares the same line (P_{A0}), which is used for STDBY LED D0 output. Two lines are used for I2C bus which are connected both to the EEPROM as well as the audio controller. One line (IRQ) is connected to the IR sensor output. In fact, this pin serves the dual purpose of being used as an interrupt as well as an input pin whose status can be read by the MCU. The data sent from the remote handset is received at this pin.

The microcontroller also checks the functioning of the memory IC ST24C02 and the audio controller IC TDA7315. If it is not able to communicate with these two ICs on the I2C bus, it flashes the volume and treble LEDs respectively to indicate that communication is not possible with the respective ICs. Refer the assembly and testing section for more details.

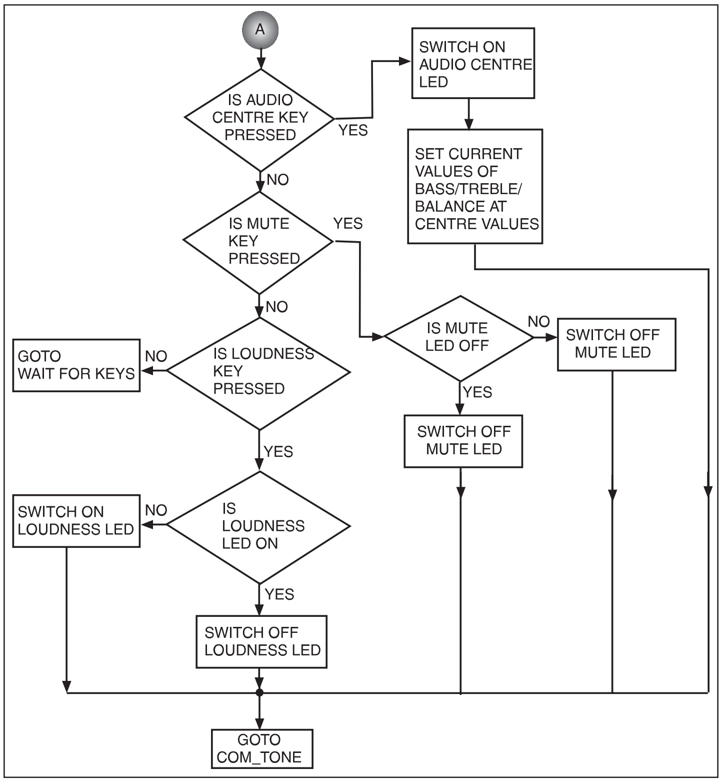

Fig. 2(d): Software flow chart (continued)

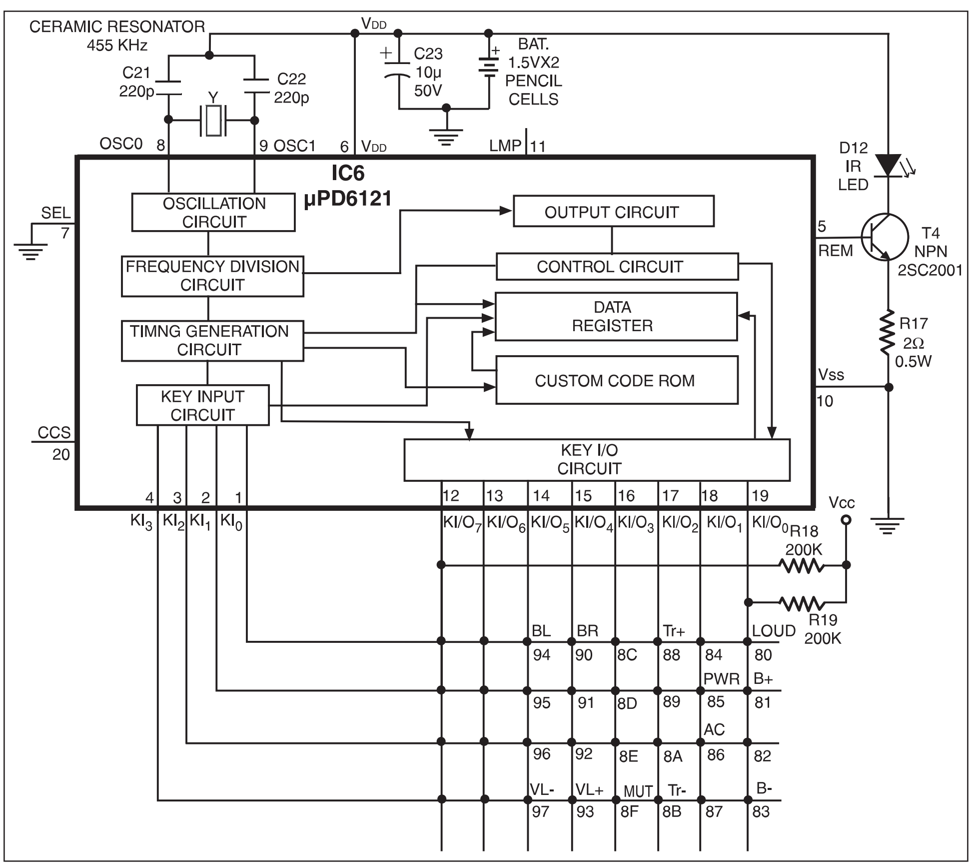

Fig. 3: Schematic diagram of remote control

Remote control (Fig. 3). Remote control is based on NEC µPD6121 customised remote transmitter IC. This IC is also used by a number of TV manufacturers. The handset sends data in a specified format whenever any key is pressed on the handset. The MCU reads the data and decides which key has been pressed. In this project, 12 keys have been used:

1. Loudness	2. Treble up
3. Balance right	4. Balance left
5. Bass up	6. Power on/standby
7. Audio centre	8. Bass down
9. Treble down	10. Mute
11. Vol up	12. Vol down

Standby, loudness and mute keys are toggle keys, i.e. if you press them once, they will change the current state. For instance, pressing the loudness key when loudness is already 'on' will cancel loudness.

Volume, treble, balance and bass keys are repeat keys. If you press them once and keep them pressed, the selected function keeps on repeating. For instance, if you press the vol up key and keep it pressed, the volume will keep on increasing until it is maximum.

The audio centre key is the selection key. Pressing it once will cause it to set all audio parameters in centre. Pressing it again will have no further effect. It will not cancel the audio centre mode, unlike the standby, mute or loudness keys.

Remote handset circuit diagram is similar to that used for the Set-Top Converter published in Jun. '97 issue of EFY. The only difference is that one of the two 200 kilo-ohm resistors is now connected to pin 12 in place of 13 of IC µPD6121. Also, the key names are different and only 12 keys are used instead of 21 in set-top converter.

Transmission code. The remote's transmission code consists of a 'leader pulse,' '16-bit customer code' and '16-bit data code.' The carrier frequency with 455kHz ceramic resonator is 38 kHz (1/12 f_0, where f_0 is resonator's fundamental frequency). The code used is based on pulse position modulation (PPM). The leader pulse consists of a 9ms carrier waveform (w/f) followed by 4.5ms off period. A 'logic 0' consists of 0.56ms of carrier w/f followed by 0.56ms of off period, and a 'logic 1' consists of 1.125ms of carrier w/f followed by 1.125ms of off period. Each code byte (starting with LSB)

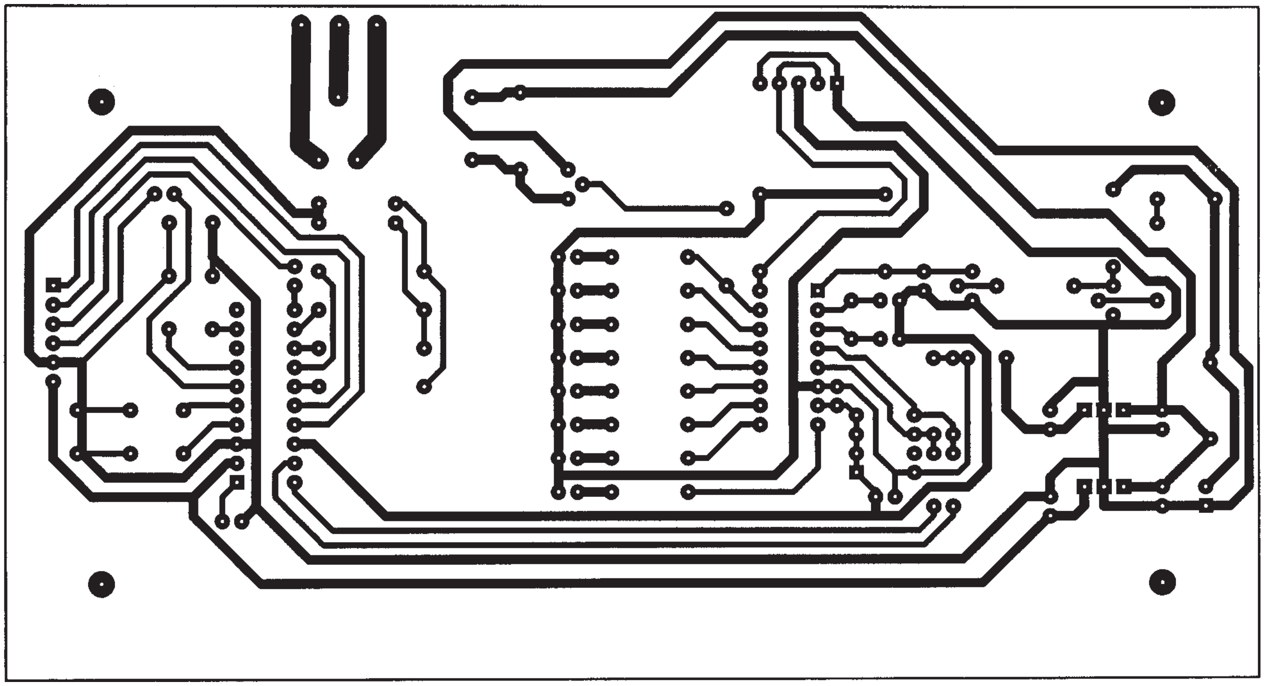

Fig. 4: Actual-size single-sided track layout for the schematic of Fig. 1

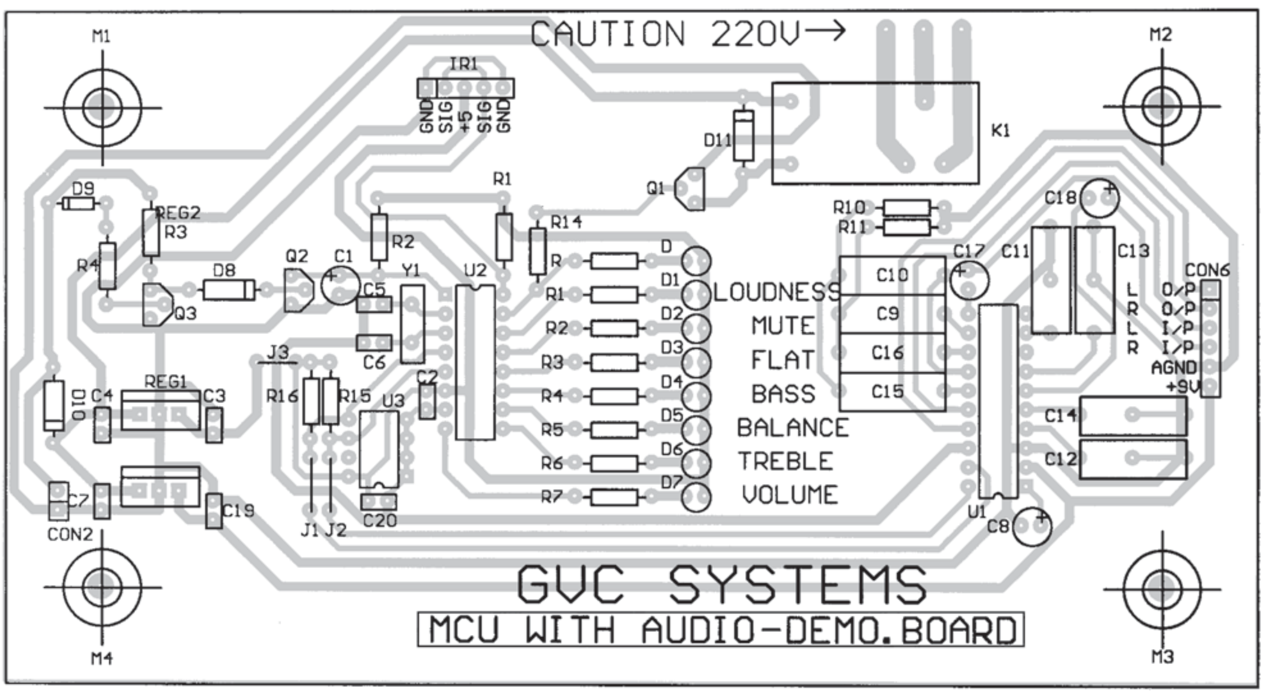

Fig. 5: Component layout for the PCB of Fig. 4

is followed by inverse code byte to give an extremely low-error rate.

The customer code (also referred to as custom code) of µPD6121 suffix G-001 can be configured as follows: The first byte (high-order byte) bits are all zeros (00000000) if no diode is placed between K I/O pins and CCS pin (cathode towards CCS). For instance, if a diode is placed across K I/O_1 and CCS then the higher order byte will become 01000000 (starting with LSB) or 02H. The low-order byte will be inverse (10111111) of the high-order byte, unless K I/O pins are pulled to Vcc. For instance, if only pin K I/O_5 is pulled high then the bit at that position only will not be reversed. Thus the low-order byte will become 10111011 (starting with LSB) or DD (Hex). With the above-mentioned diode and pull-up resistor, the 16-bit custom code would be 02DD (Hex). So, you can figure out how the custom code for the present remote control circuit has been arrived at. The key code for each key is annotated on the keys in Fig. 3. Grounding of SEL pin 7 results in bit D7 of key code to be set to 1. Hence, key at intersection of row 0 and column 0 would generate 80H as the key code. Complete remote code table for the present circuit configuration is given in Table V.

IR receiver module. This is a 3-pin device incorporating surface mount IC. Two pins are for +5V supply and ground while the third pin is for data output. IR receiver module receives the data sent by remote handset, amplifies, demodulates and converts it to MCU-compatible voltage format and outputs it on its data output pin. The MCU can decode this signal and act as per the key pressed.

Relay. This relay is used to switch off the main power amplifier circuit. Thus, when the unit is switched off from remote, supply to the amplifier circuit is switched off, and this saves electricity. Only the circuit in this construction article remains active and waits for any remote key operation. Relay can be used to control AC supply or DC supply of the amplifier as desired. This choice is left to the reader assembling the circuit. Relay and standby LED share the same pin from the microcontroller as stated earlier.

LED indicator panel. Eight LEDs are used in the circuit. Their functions are:

Standby. This LED is on when the unit is in standby mode (power amplifier off). When this LED is on, all other LEDs will be off and all keys, except the standby key, are rejected. Pressing the standby key, in this state, will switch the unit on.

Mute. This LED is on when the unit is in the mute mode. Pressing the mute key will toggle the state of LED and unit, i.e. from mute to sound or sound to mute. This LED is also switched off when the unit is in sound mode, i.e. when any key which changes the volume level is pressed (e.g. vol down, treble up/treble down, bass up/bass down, balance up/balance down, loudness and audio centre).

Loudness. This LED is on when the loudness mode is on. Pressing the loudness key will change the state of LED and unit—from loudness on to loudness off or loudness off, to loudness on. Pressing audio centre key will switch off this LED (if on) and set the loudness off.

Audio centre. This LED (marked flat) goes on when the audio centre key is pressed. All tone parameters are set at normal, i.e. zero-attenuation for speakers, with bass and treble at 0 dB (no gain/no attenuation) and loudness is switched off. Mute is cancelled if active. Just flat frequency response. Volume remains undisturbed. Changing treble/balance/bass/loudness level from the remote controller will switch off this LED.

Volume. This LED is on when the vol up/vol down key is pressed. The LED goes off as soon as the key is released.

Treble. This LED is on when the treble up/treble down key is pressed. The LED goes off as soon as the key is released.

Bass: This LED is on when the bass up/bass down key is pressed.

Balance: This LED is on when the balance left or balance right key is pressed. The LED goes off as soon as the key is released.

Power supply. The unit requires +5V regulated supply for all three major components—MCU/ EEPROM/IR Sensor. The audio controller requires +9V for the audio amplifier section. Supply design uses external +12V adaptor and two linear industry-standard regulators, 7805 and 7809. External +12V DC is also used for relay driving. So, make sure that the adaptor is capable of catering for the relay load easily. On full load, its voltage should not drop below +11V DC.

TABLE V

Remote Code Table

Custom code	00 7E
Power (PWR)	85
Mute (MUT)	8F
Treble up (TR+)	88
Treble down (TR-)	8B
Bass up (B+)	81
Bass down (B-)	83
Volume up (VL +)	93
Volume down (VL -)	97
Balance left (BL)	94
Balance right (BR)	90
Audio centre (AC)	86
Loudness	80

TABLE VI

Indication	Possible Fault And Remedial Action
All LEDs blink thrice but remote does not function	Micro controller OK. Check infra-red sensor connections or remote handset.
No LED blinks	Check supply voltage/crystal connection to MCU/pull up resistance and capacitor at reset pin of MCU.
Vol.LED blinks thrice	Replace 24C02, check supply, SCL/SDA lines to 24C02
Treble LED blinks thrice	Replace 7315, check supply SCL/SDA lines to TDA7315
Vol &Treble LED blink thrice	Check pull up resistance on SDA.Replace 24C02 and TDA7315. Check SCL and SDA lines

Assembly and Testing

Complete circuit comprising the MCU, the memory, the audio controller and the IR receiver, as also the power supply circuit, is shown in Fig. 1. Actual-size, single-sided PCB for the circuit of Fig. 1 is shown in Fig. 4. The component layout for the PCB is shown in Fig. 5. No PCB is shown for remote. However, suitable pre-assembled remote control for the project is proposed to be made available to readers through Kits‘n’Spares outlet.

The circuit is very simple, having very few components. Before installing the main ICs for MCU, memory, audio controller in their sockets and soldering the IR receiver module, make sure that supply voltage is correct. All parts, except the audio controller, require +5V as logic supply. The audio controller requires +9V supply for the audio control section. The remote can be operated with two 1.5-volt pencil cells.

The unit has built-in diagnostics, which will make the job of testing very easy. Use Table VI to test the unit and identify/rectify faults.

Software flow chart

For commercial reasons the software for the project, which is intellectual property of the author, is not included. However, logical flow charts of the software are given in Figs 2(a) through 2(d), which are quite self-explanatory. ❑