RF-based Motion-sensing Robot

VINEET NAYAK

We have seen many projects where a robot is controlled by a remote controller. Here we present a robot that can be moved in any direction just by the movement of your hand. This motion-sensing robot senses the motion of your hand and transmits corresponding instructions to control the robot through RF communication.

Fig. 2 shows block diagram of the motion-sensing robot. The movement of wireless controller is sensed by the accelerometer. The data from the accelerometer is compared by the comparator and corresponding encoded instructions are transmitted through the RF transmitter to control the robot. The robot receives the instructions and the microcontroller section drives the motors corresponding to the instructions received.

Fig. 1: Author's prototype

Test Points for Fig. 2

Test point	Details
TP0, TP4	Ground (GND)
TP1, TP5	5V
TP2	Input from Y-axis
TP3	Input from X-axis
TP6	0V when S3 is pressed
TP7	Input for PB0 of IC6 (D1)
TP8	Input for PB1 of IC6 (D2)
TP9	Input for PB2 of IC6 (D3)
TP10	Input for PB3 of IC6 (D3)

Circuit and working

Figs 3 and 4 show circuits of the wireless controller and the robot, respectively.

The wireless controller is built around encoder HT12E (IC1), operational amplifier LM339 (IC2), regulator 7805 (IC3), RF transmitter module (TX1), accelerometer module and a few discrete components.

EFY TESTED SUNIL KUMAR

IC2 has four op-amps that are wired as comparators. Pins 9 and 10 of IC2 are connected to X-axis output of the accelerometer. Pins 4 and 7 are connected to Y-axis output of the accelerometer. IC2 receives the data for X and Y axes from the accelerometer. This data is continuously compared with pre-defined values for each axis by comparators A1 through A4. The reference value is selected using VR1 through VR4.

If the received signal for any axis

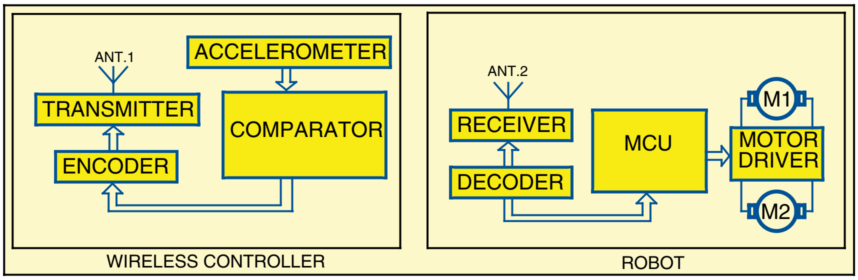

Fig. 2: Block diagram of motion-sensing robot

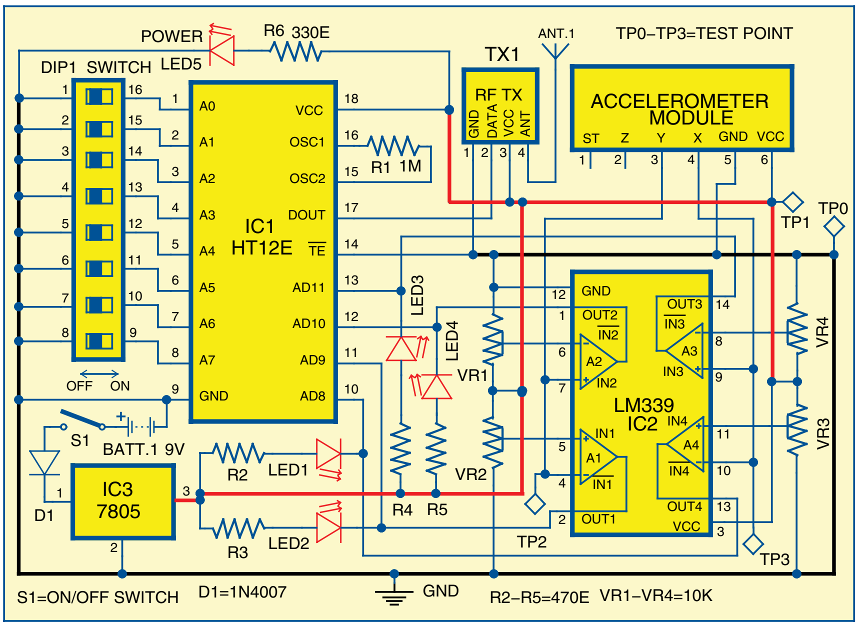

Fig. 3: Circuit of wireless controller

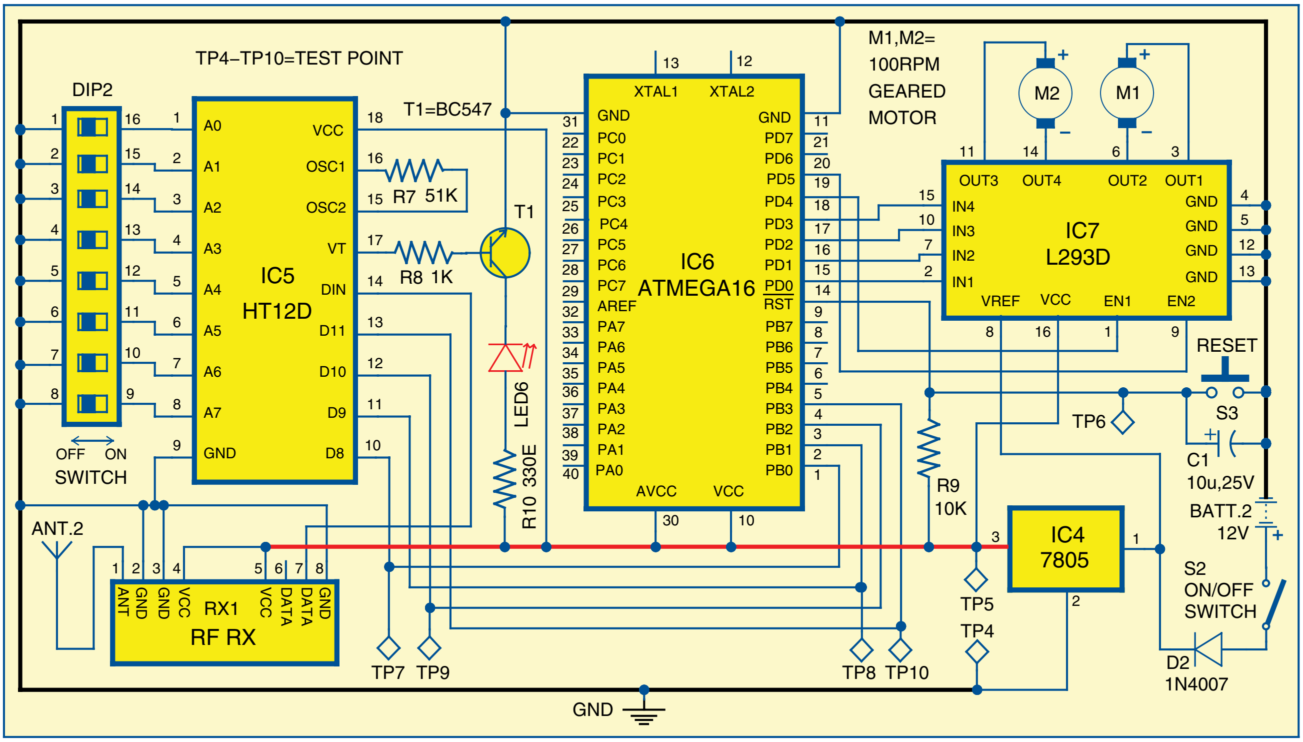

Fig. 4: Circuit for the robot

crosses the pre-defined voltage level, the corresponding comparator's output goes low. The same is also indicated by LED1 through LED4. The outputs of the comparators are fed to inputs AD8 through AD11 of IC1. The generated

PARTS LIST

Semiconductors:

IC1	- HT12E encoder
IC2	- LM339 comparator
IC3, IC4	- 7805, 5V regulator
IC5	- HT12D decoder
IC6	- ATMega16 microcontroller
IC7	- L293D motor driver
T1	- BC547 npn transistor
LED1-LED6	- 5mm LED
D1, D2	- 1N4007 rectifier diode
TX1	- ASK transmitter module
RX1	- ASK receiver module
	- Accelerometer module

Resistors (all ¼-watt, ±5 per cent carbon):

R1	- 1-mega-ohm
R2-R5	- 470-ohm
R6, R10	- 330-ohm
R7	- 51-kilo-ohm
R8	- 1-kilo-ohm
R9	- 10-kilo-ohm
VR1-VR4	- 10-kilo-ohm preset

Capacitors:

C1	- 10µF, 25V electrolytic

Miscellaneous:

DIP1, DIP2	- 16-pin DIP switch
S1, S2	- On/off switch
S3	- Push-to-on switch
M1, M2	- 12V, 100-rpm geared motor
BATT.1	- 9V battery (PP3)
BATT.2	- 12V, 4.5Ah battery
ANT.1-ANT.2	- 10 cm hook-up wire

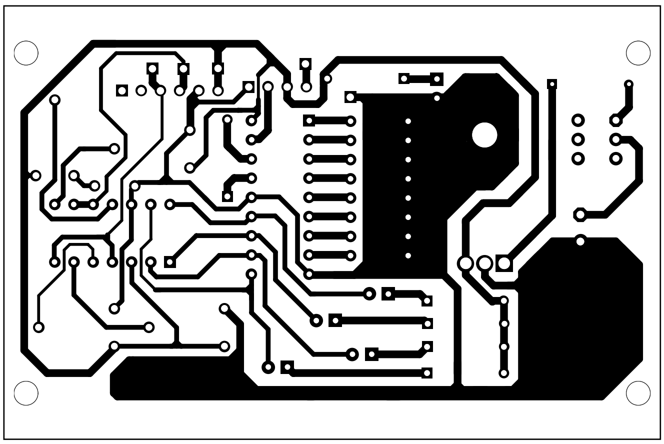
Fig. 5: An actual-size, single-side PCB for the wireless controller

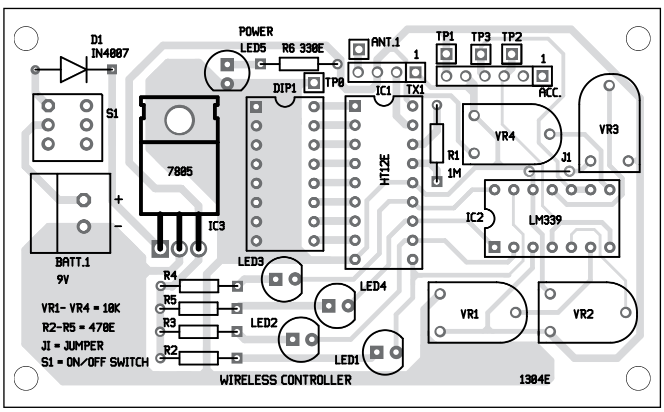

Fig. 6: Component layout for the PCB in Fig. 5

code is encoded through encoder IC1 and transmitted using the RF transmitter module TX1. Regulator IC3 provides regulated 5V supply to the circuit.

Accelerometer module. An accelerometer is an electromechanical device that measures acceleration of anything that it is mounted on. The accelerometer module used here is based on ADXL335 triple-axis accelerometer from Analog Devices. The sensor has a full sensing range of ±3g.

Robot. It comprises an

RF receiver module (RX1), regulator 7805 (IC4), decoder HT12D (IC5), microcontroller ATmega16 (IC6), motor driver L293D (IC7) and a few discrete components.

Microcontroller ATmega16 is the heart of the circuit. ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR-enhanced RISC architecture. It has 16 kB of in-system programmable flash program memory, 512 bytes of EEPROM, 1kB SRAM, 32 general-purpose input/output (I/O) lines, 32 general-purpose working registers, three flexible timers/counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented two-wire serial interface, an 8-channel 10-bit analogue-to-digital converter, an SPI serial port and six software-selectable power-saving modes.

Data transmitted from the wireless controller is received by the RF receiver module and decoded by IC5. Data reception is indicated by LED6. Decoded data is fed to port pins PB0 through PB3 of microcontroller IC6. Identical addresses for the encoder and the decoder can be selected through switches DIP1 and DIP2. The outputs from pins PD0 through PD3 of the microcontroller are fed to IN1 through IN4 of motor-driver IC7 to drive motors M1 and M2, respectively. Outputs OUT1 and OUT2 drive motor M1, and outputs OUT3 and OUT4 drive motor M2. Enable pins EN1 (pin 1) and EN2 (pin 9) are connected to port pins PD4 and PD5 of microcontroller IC6, respectively.

Regulator IC4 provides regulated 5V supply to the circuit. 12V battery supply is directly connected to pin 8 of motor driver IC7. Switch S3 is used for manual reset.

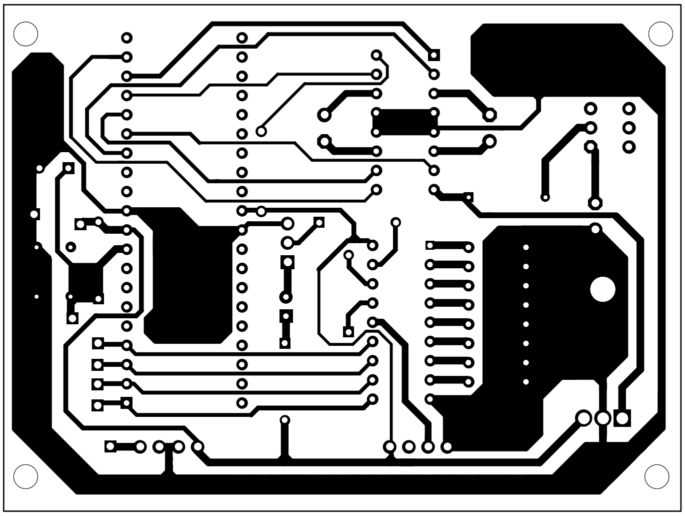

Fig. 7: An actual-size, single-side PCB for the robot

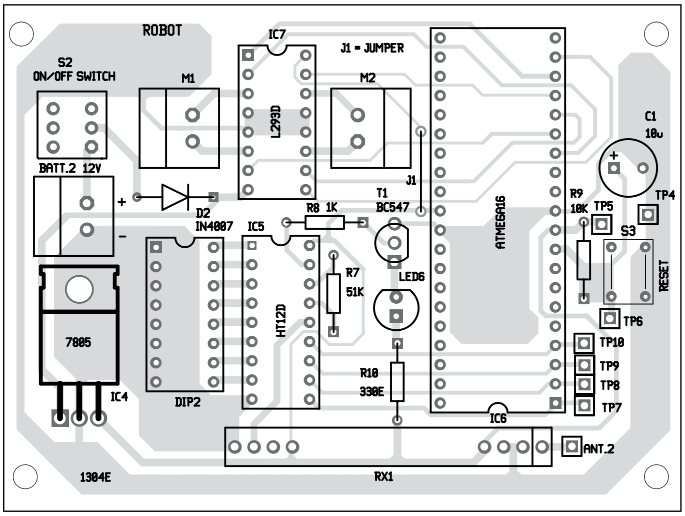

Fig. 8: Component layout for the PCB in Fig. 7

EFY Note

The source code of this project is included in this month's EFY DVD and is also available for free download on www.efymag.com website.

Construction and testing

Actual-size, single-side PCBs of wireless controller and robot are shown in Figs 5 and 7 and their component layouts in Figs 6 and 8, respectively. Assemble the circuit on a PCB as it saves time and minimises assembly errors. Carefully assemble the components and double-check for any overlooked error. Always use IC bases for ICs. Before inserting the microcontroller and other ICs, always check the correct supply voltage. Suitable connectors are provided on both the PCBs to connect geared motors and power supplies.

Hold the wireless controller in your hand and tilt it to move the robot in the desired direction. Controller's sensitivity can be adjusted using VR1 through VR4.

Align the controller in your hand such that when you tilt the remote control down in front (forward), both motors M1 and M2 rotate in forward direction (clock-wise). When you tilt the remote control up in front (backward), both motors M1 and M2 should rotate in reverse direction (anti-clockwise). Both the circuits are simple and will immediately work after proper assembly. In case of any difficulty refer the test point table and verify the voltage levels.

Software

The source program is written in 'C' language and compiled using AVR Studio to generate the Intel hex code. The generated hex code is programmed using a suitable programmer. It is well commented and easy to understand.

The device is shipped with CKSEL = "0001" and SUT = "10." The default clock source setting is therefore 1MHz internal R-C oscillator with the longest startup time. To activate the internal oscillator, fuse bytes of the microcontroller have to be programmed as follows:

Fuse low byte = D4
Fuse high byte = 99 ●

The author is a second-year B.Tech in electrical engineering from College of Engineering and Technology, Bhubaneswar. He likes working with electronic projects including microcontrollers