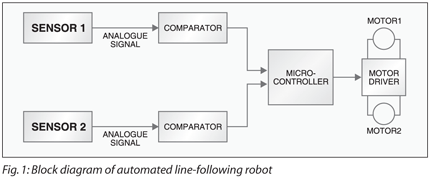
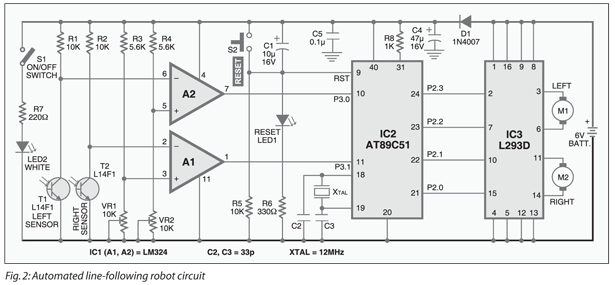
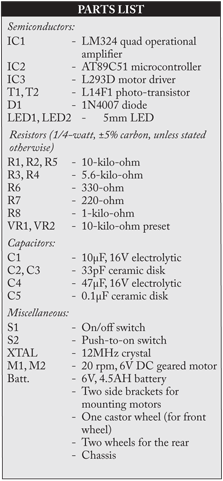
**Automated Line Following Robot**

**L**ine-following robots with pick-and-placement capabilities are commonly used in manufacturing plants. These move on aspecified path to pick the components from specified locations and place them on desired location. Basically, a line-following robot is a self-operating robot that detects and follows a line drawn on the floor. The path to be taken is indicated by a white line on a black surface. The control system used must sense the line and manoeuvre the robot to stay on course while constantly correcting the wrong moves using feedback mechanism, thus forming a simple yet effective closed-loop system.  
  
**Circuit description**  
  
Fig. 1 show the block diagram of the automated line-following robot. It consists of mainly four parts: two sensors, two comparators, one decision-making device and two motor drivers. The robot is built usingmicrocontroller AT89C51 (used as the decision-making device), motor driver L293D, operational amplifier LM324 (comparator), phototransistor (sensor) and a few discrete components.In the circuit, the sensors (phototransistors) are used to detect the white strip on a black background. The sensor output is fed to the microcontroller, which takes the decision and gives appropriate command to motor driver L293D so as to move the motor accordingly.

***Sensor.*** The sensor senses the light reflected from the surface and feeds the output to the comparator. When the sensor is above the white background the light falling on it from the source reflects to the sensor, and when the sensor is above the black background the light from the source doesn’t reflect to it. The sensor senses the reflected light to give an output,which is fed to the comparator.

***Comparator.*** The comparator compares the analogue inputs from sensors with a fixed reference voltage. If this voltage is greater than the reference voltage the comparator outputs a low voltage, and if it is smaller the comparator generates a high voltage that acts as input for the decision-making device (microcontroller).

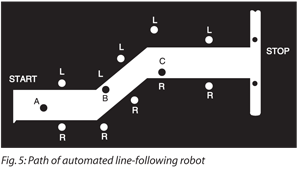
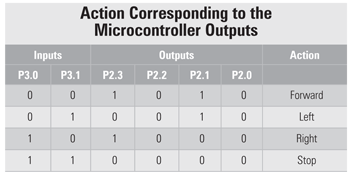
***Microcontroller.***The microcontroller is programmed to make the robot move forward, turn right or turn left based on the input coming from the comparator. The outputs of the microcontroller are fed to the motor driver.

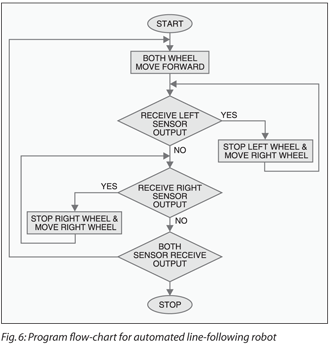
***Motor driver.*** The current supplied by the microcontroller to drive the motor is small. Therefore a motor-driver IC is used. It provides sufficient current to drive the motor.   
       
  
Fig. 2 shows the circuit of the automated line-following robot. When light falls on the phototransistor (say, T1), it goes into saturation and starts conducting. When no light falls on the phototransistor, it is cut-off. A white LED (LED2) has been used to illuminate the white path on a black background. Phototransistors T1 and T2 are used for detecting the white path on the black background.  
       
Collectors of phototransistors T1 and T2 are connected to the inverting inputs of operational amplifiers A2 and A1. The signal voltage at the inverting input of the operational amplifier is compared with the fixed reference voltage, which is formed by a potential divider circuit of 5.6-kilo-ohm resistor and 10-kilo-ohm preset. This reference voltage can be adjusted by changing the value of the 10-kilo-ohm preset.  
       
When sensor T2 is above the black surface, it remains cut-off as the black surface absorbs virtually all the light falling from LED2 and no light is reflected back. The voltage at the inverting input (pin 2) of operational amplifier A1 is higher than the reference voltage at its non-inverting input (pin 3) and therefore the amplifier output at pin 1 becomes zero.When sensor T2 is above the white line, the light gets reflected from the white surfaceto fall on phototransistor T2. Phototransistor T2 goes into saturation and conducts. The inverting input (pin 2) of operational amplifier A1 goes below the reference voltage at its non-inverting input (pin 3) of operational amplifier A1 and therefore output pin 1 goes high. This way, the comparator outputs logic ‘0’ surface

and logic ‘1’ for white surface.

Similarly, comparator A2 compares the input voltage from phototransistor T1 with a fixed reference voltage.The outputs of operational amplifiers A1 and A2 are fed to microcontroller AT89C51. The AT89C51 is an 8-bit microcontroller having 4 kB of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timers/ counters, a five-vector two-level interrupt architecture, on-chip oscillator and clock circuitry. A 12MHz crystal is used for providing the basic clock frequency. All I/O pins are reset to ‘1’ as soon as RST pin goes high. Holding RST pin high for two machine cycles while the oscillator is running resets the device. Power-on reset is derived from resistor R5 and capacitor C1. Switch S2 is used for manual reset. The microcontroller, based on the inputs from sensor T1 (say, left) and sensor T2 (say, right), controls the motor to make the robot turn left, turn right or move forward.  
       
Port pins P2.0, P2.1, P2.2 and P2.3 are connected to pins 15, 10, 7 and 2 of motor driver L293D. Port pins P2.0 and P2.1 are used for controlling the right motor, while port pins P2.2 and P2.3 are used for controlling the left motor. Three wheels can be used for this robot—one on the front and two at the rear. Front wheel can rotate in any direction as specified by the rear wheel. To make the robot turn left, the left-side motor should stop and the right-side motor should rotate in the clockwise direction. Similarly, to make the robot turn right, the right-side motor should stop and the left-side motor should rotate in clockwise direction. For forward motion, both the motors should rotate in clockwise direction.  
 **Working**

An actual-size, single-side PCB for the automated line-following robot is shown in [Fig. 3 (View as PDF)](http://www.electronicsforu.com/electronicsforu/circuitarchives/my_documents/my_files/fig3_actual_size.pdf) and its component layout in [Fig. 4 (View as PDF)](http://www.electronicsforu.com/electronicsforu/circuitarchives/my_documents/my_files/fig_4_component_layout.pdf). Fig. 5 shows the path of the line-follower robot, where ‘L’ is the left sensor and ‘R’ is the right sensor.   
       
At the start, when the robot is at point ‘A,’ sensors T1 and T2 are above the black surface and port pins P3.0 and P3.1 of the microcontroller receive logic ‘0.’ As a result, the robot moves forward in straight direction.   
       
  
At point ‘B,’ a left turn is encountered, and the left sensor comes above the white surface, whereas the right sensor remains above the black surface. Port pin P3.0 of the microcontroller receives logic ‘1’ from the left sensor and port pin P3.1 receives logic ‘0’ from the right sensor. As a result, the left motor stops and the right motor rotates, to make the robot turn left. This process continues until the left sensor comes above the black background. Similarly, at point ‘C,’ where a right turn is encountered, the same procedure for right turn is executed. When both the sensors are at the white surface, the robot should stop. The output of the microcontroller (IC2) depends on the inputs received at its port pins P3.0 and P3.1 as shown in table.

**Software**  
  
The source program for the project is written in Assembly language and assembled using Metalink’s ASM51 assembler, which is freely available on the Internet for download. It is well commented for easy understanding and works as per the flow-chart shown in Fig. 6. The hex file ‘robot.hex’ is to be burnt into the microcontroller.

|  |
| --- |
| **ROBOT.ASM** |

|  |  |
| --- | --- |
| $MOD51                                   ORG 0000H                                   LJMP MAIN                                   ORG 0030H MAIN:                          SETB P3.0 ;Input for left sensor                                   SETB P3.1 ;Input for right sensor AGAIN:                         JB P3.0,NEXT                                   JB P3.1,GO                                   CLR P2.0                                   SETB P2.1                                   CLR P2.2                                   SETB P2.3                                   SJMP AGAIN GO:                             CLR P2.0                                   SETB P2.1 | CLR P2.2                                     CLR P2.3                                     SJMP AGAIN NEXT:                         JB P3.1,GO1                                                                   CLR P2.0                                     CLR P2.1                                     CLR P2.2                                     SETB P2.3                                     SJMP AGAIN GO1:                             CLR P2.0                                     CLR P2.1                                     CLR P2.2                                     CLR P2.3                                     SJMP AGAIN HERE:                            SJMP HERE                                     END |