In this project we are going to control a robot wirelessly using hand gestures. This is an easy, user-friendly way to interact with robotic systems and robots. An accelerometer is used to detect the tilting position of your hand, and a microcontroller gets different analogue values and generates command signals to control the robot. This concept can be implemented in a robotic arm used for welding or handling hazardous materials, such as in nuclear plants. The author’s prototype is shown in Fig. 1.

Circuit and working

The block diagram of the wireless gesture-controlled robot is shown in Fig. 2. The circuit diagram of the transmitter section of the wireless gesture-controlled robot is shown in Fig. 3 and of the receiver section in Fig. 4.

**ATmega328.** ATmega328 is a single-chip microcontroller from Atmel and belongs to the mega AVR series. The Atmel 8-bit AVR RISC based microcontroller combines 32kB ISP Flash memory with read-while-write capabilities, 1kB EEPROM, 2kB SRAM, 23 general-purpose I/O lines, 32 general-purpose working registers, three flexible timers/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 10-bit A/D converter, programmable watch-dog timer with an internal oscillator and five software-selectable power-saving modes.

The device operates between 1.8 and 5.5 volts. It achieves throughputs approaching one MIPS per MHz. An alternative to ATmega328 is ATmega328p.

**ADXL335.** This is a complete three-axis acceleration measurement system. ADXL335 has a minimum measurement range of ±3g. It contains a poly-silicon-surface micro-machined sensor and signal-conditioning circuitry to implement open-loop acceleration measurement architecture. Output signals are analogue voltages that are proportional to acceleration. The accelerometer can measure the static acceleration of gravity in tilt-sensing applications as well as dynamic acceleration resulting from motion, shock or vibration.

The sensor is a poly-silicon-surface micro-machined structure built on top of a silicon wafer. Poly-silicon springs suspend the structure over the surface of the wafer and...
resulting in a sensor output whose unbalances the differential capacitor, deflection deflects the moving mass and out-of-phase square waves. Accelerometer fixed plates and plates attached to capacitor that consists of independent forces. Deflection of the structure provides resistance against acceleration. The capability to select a TE transmission medium upon receipt of a trigger signal. The HT1222 encoder further enhances the application flexibility of 212 series of encoders. The HT12D also provides a 38kHz carrier for infra-red systems.

**Transmitter.** The transmitter consists of ATmega328 microcontroller (IC2), ADXL335 accelerometer, HT12E encoder (IC4) and 433MHz RF transmitter module (TX1). In this circuit, two analogue outputs from ADXL335 pins (x, y) are connected with input pins (23, 24) of the microcontroller. Analogue signals are converted to digital signals through the microcontroller. Digital outputs from pins 16, 17, 18 and 19 of the microcontroller are directly sent to pins 13, 12, 11 and 10 of encoder IC4. This data is encoded and transmitted via RF module TX1.

**Receiver.** The receiver part consists of 433MHz RF receiver module (RX1), HT12D decoder (IC5) and L293D motor driver (IC6) to run the motors. Here, receiver module RX1 receives the transmitted signal, which is decoded by decoder IC to get the same digital outputs. Four outputs of IC6 drive two motors. The robot moves as per tilt direction of the accelerometer in the transmitter. The direction of the robot movement is as per logic listed in Table 1.

**Encoder (HT12E) and decoder (HT12D) ICs.** The 212 encoders are a series of CMOS LSIs for remote-control system applications. These are capable of encoding information that consists of N address bits and 12 N data bits. Each address/data input can be set to one of two logic states. Programmed addresses/data are transmitted together with header bits via an RF or infra-red transmission medium upon receipt of a trigger signal. The capability to select a trigger on HT12E or a data (DIN) trigger on HT12D decoder further enhances the application flexibility of 212 series of encoders.

**Software program**

The software program is written in Arduino programming language. We programmed a fresh ATmega328 microcontroller with the help of Arduino IDE 1.0.5 and an Arduino Uno board.

First, we have to load bootloader code into the microcontroller. For that, we used Arduino Uno for in-system programming (ISP) given in the IDE, by selecting File → Examples → Arduino ISP. Once the bootloader

---

**TABLE I**

<table>
<thead>
<tr>
<th>Movement of Robot and Decoder Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robot (accelerometer)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>Forward (-Y)</td>
</tr>
<tr>
<td>Backward (+Y)</td>
</tr>
<tr>
<td>Right (+X)</td>
</tr>
<tr>
<td>Left (-X)</td>
</tr>
</tbody>
</table>

---

provide resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass.

Fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor, resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

**L293D.** This is a 16-pin DIP package motor driver IC having four input pins and four output pins. All four input pins are connected to output pins of the decoder IC (IC5) and the four output pins are connected to DC motors of the robot. Enable pins are used to enable input/output pins on both sides of IC6.

**Encoder (HT12E) and decoder (HT12D) ICs.** The 212 encoders are a series of CMOS LSIs for remote-control system applications. These are capable of encoding information that consists of N address bits and 12 N data bits. Each address/data input can be set to one of two logic states. Programmed addresses/data are transmitted together with header bits via an RF or infra-red transmission medium upon receipt of a trigger signal. The capability to select a trigger on HT12E or a data (DIN) trigger on HT12D decoder further enhances the application flexibility of 212 series of encoders. The HT12D also provides a 38kHz carrier for infra-red systems.

**Transmitter.** The transmitter consists of ATmega328 microcontroller (IC2), ADXL335 accelerometer, HT12E encoder (IC4) and 433MHz RF transmitter module (TX1). In this circuit, two analogue outputs from ADXL335 pins (x, y) are connected with input pins (23, 24) of the microcontroller. Analogue signals are converted to digital signals through the microcontroller. Digital outputs from pins 16, 17, 18 and 19 of the microcontroller are directly sent to pins 13, 12, 11 and 10 of encoder IC4. This data is encoded and transmitted via RF module TX1.

**Receiver.** The receiver part consists of 433MHz RF receiver module (RX1), HT12D decoder (IC5) and L293D motor driver (IC6) to run the motors. Here, receiver module RX1 receives the transmitted signal, which is decoded by decoder IC to get the same digital outputs. Four outputs of IC6 drive two motors. The robot moves as per tilt direction of the accelerometer in the transmitter. The direction of the robot movement is as per logic listed in Table 1.

**Software program**

The software program is written in Arduino programming language. We programmed a fresh ATmega328 microcontroller with the help of Arduino IDE 1.0.5 and an Arduino Uno board.

First, we have to load bootloader code into the microcontroller. For that, we used Arduino Uno for in-system programming (ISP) given in the IDE, by selecting File → Examples → Arduino ISP. Once the bootloader
Fig. 5: An actual-size PCB layout of the transmitter circuit

Fig. 6: Component layout of the transmitter circuit

Fig. 7: An actual-size PCB layout of the receiver circuit

Fig. 8: Component layout of the receiver circuit

Fig. 9: Transmitter module mounted on the back of the palm

### Table II

<table>
<thead>
<tr>
<th>Test Point</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>TP0</td>
<td>0V (GND)</td>
</tr>
<tr>
<td>TP1</td>
<td>5V</td>
</tr>
<tr>
<td>TP2</td>
<td>3.3V</td>
</tr>
<tr>
<td>TP3</td>
<td>Train of pulses</td>
</tr>
<tr>
<td>TP4</td>
<td>0V (GND)</td>
</tr>
<tr>
<td>TP5</td>
<td>4.5V</td>
</tr>
<tr>
<td>TP6</td>
<td>Train of pulses</td>
</tr>
</tbody>
</table>

EFY Note

The source code of this project is included in this month’s EFY DVD and is also available for free download at source.efymag.com

in Fig. 6. An actual-size, single-side PCB layout of the receiver circuit is shown in Fig. 7 and its component layout in Fig. 8.

The transmitter section can be held in your palm or on the other side (refer Fig. 9). The receiver module is mounted on the robot.

Mount all components on the PCBs shown here to minimise assembly errors. Fix the receiver PCB and 4.5V battery on the chassis of the robot. Fix two motors, along with wheels, at the rear side of the robot and a castor wheel on the front. After uploading the main code into the microcontroller, remove it from the Arduino Uno board and insert it into the populated transmitter PCB.

Now, switch-on the power supplies in the transmitter as well as receiver circuits. Attach the transmitter circuit to your hand and move your hand forwards, backwards and sideways. Directions of the robot movement are given in Table I. The robot will stop if you keep your palm horizontal, parallel to the Earth’s surface.

For troubleshooting, first verify that voltages at various test points are as per Table II.

Aquib Javed Khan is pursuing B.Tech from Orissa Engineering College, Bhubaneswar. He is interested in mechatronics systems.