**COMPONENTS**

**BLOCK DIAGRAM**

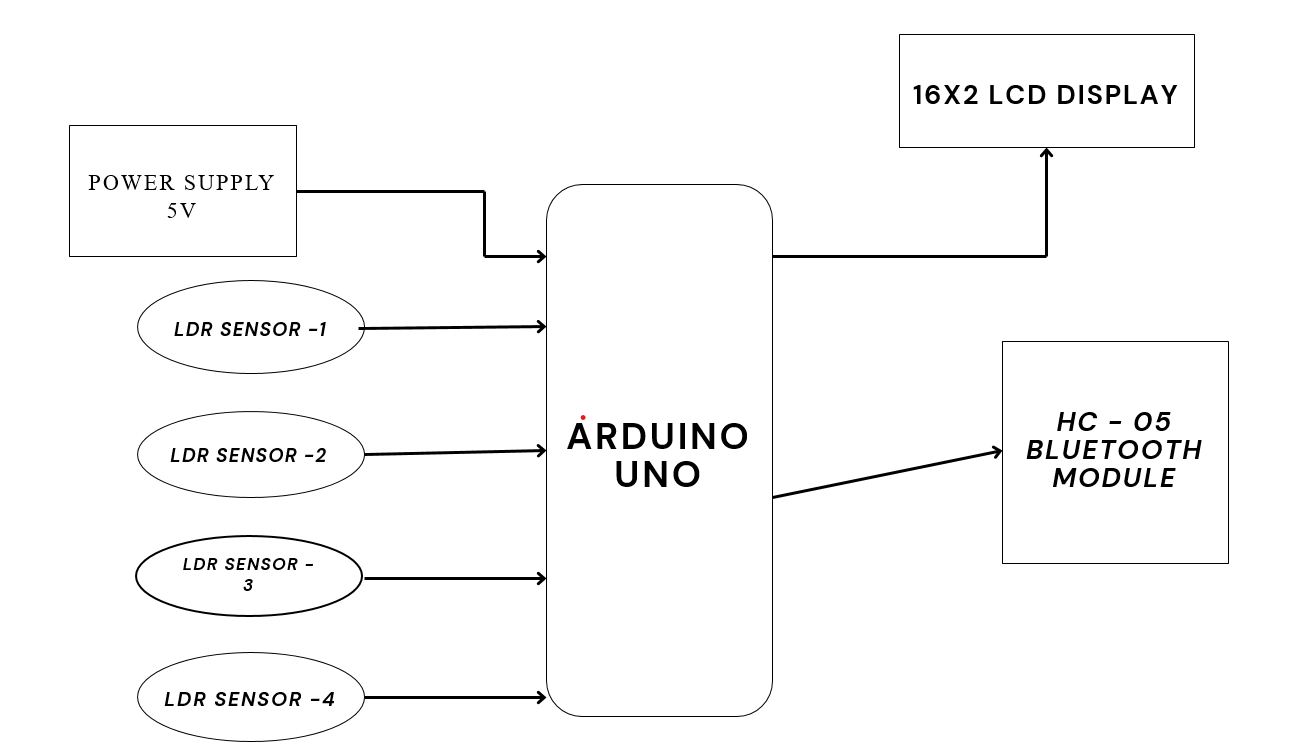


Fig.4.1 Block diagram of Hand Gesture Recognition System

[**ARDUINO UNO**](http://www.jtagelectronics.com/?p=75)

The most common version of Arduino is the Arduino Uno. This board is what most people are talking about when they refer to an Arduino. The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. There are different revisions of Arduino Uno, below detail is the most recent revision (Rev3 or R3).

The Arduino Uno is a microcontroller board based on the Atmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Microcontroller : Atmega328

Operating Voltage : 5V

Input Voltage (recommended): 7-12V

Input Voltage (limits) : 6-20V

Digital I/O Pins : 14 (of which 6 provide PWM output)

Analog Input Pins : 6

DC Current per I/O Pin : 40 mA

DC Current for 3.3V Pin : 50 mA

Flash Memory Flash Memory                         : 32 KB (Atmega328) of which 0.5 KB used by bootloader

SRAM : 2 KB (Atmega328)

EEPROM : 1 KB (Atmega328)

Clock Speed : 16 MHz

Length : 68.6 mm

Width : 53.4 mm

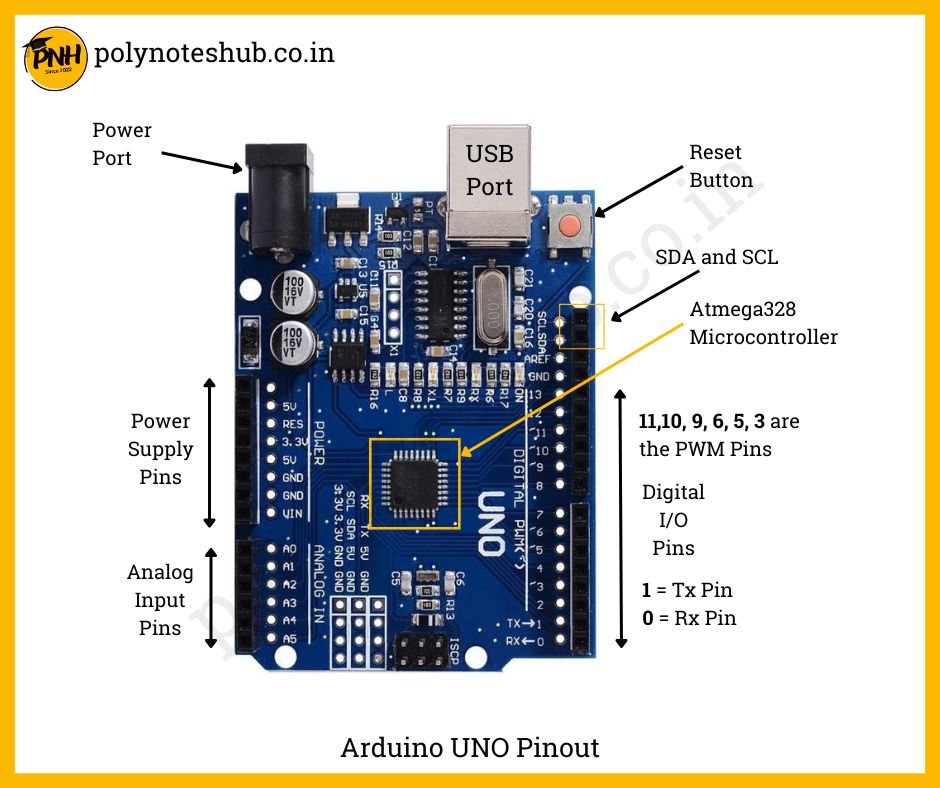


Fig.4.2 Arduino UNO

1. **USB Plug & External Power Supply Plug**

Every Arduino board needs a way to be connected to a power source. The Arduino Uno can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack. The power source is selected automatically. The USB connection is also how you will load code onto your Arduino board. Please on my other post on how to program with Arduino can be found in Installing and Programming Arduino.

**NOTE:** The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V. However, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

1. **Voltage Regulator**

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don’t hook up your Arduino to anything greater than 20 volts.

1. **Power Pins**
   1. Voltage In Pin – The input voltage to the Arduino board when it’s using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
   2. 5V Pin – This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 – 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. It’s not recommended.3.3V Pin – A 3.3volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
2. **Ground Pins**

There are several GND pins on the Arduino, any of which can be used to ground your circuit.

1. **IOREF Pin**

This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

1. **Input and Output Pins**

Each of the 14 digital pins on the Uno can be used as an input or output. They operate at 5 volts. These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED). Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-5k Ohms. In addition, some pins have specialized functions.

1. **Serial Out (TX) & Serial In (RX)**

Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the Atmega8U2 USB-to-TTL Serial chip.

1. **External Interrupts**
2. Pins 2 and 3 can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
3. PWM – You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11). These pins act as normal digital pins, but can also be used for something called Pulse- Width Modulation (PWM). Think of these pins as being able to simulate analog output (like fading an LED in and out).
4. SPI – Pins 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). SPI stands for Serial Peripheral Interface. These pins support SPI communication using the SPI library.
5. Analog Input Pins – Labeled A0 through A5, each of which provide 10 bits of resolution (i.e.,1024 different values). These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read. By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF Pin (Stands for Analog Reference. Most of the time you can leave this pin alone). Additionally, some pins have specialized functionality:
6. TWI – Pins A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
7. **Reset Pin**

Bring this line LOW to reset the micro controller. Typically, used to add a reset button to shields which block the one on the board.

1. **LED Indicators**

Power LED Indicator – Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit. On-Board LED – There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it’s off. This useful to quickly check if the board has no problem as some boards has a pre-loaded simple blinking LED program in it.

TX & RX LEDs – These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we’re loading a new program on to the board).

**Reset Button:** Pushing the reset button temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times.

* 1. **BLUETOOTH (HC-05)**

HC‐05 module is an easy use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate**)** 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Blue core 04‐External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).

**BLUETOOTH MODULE HC-05**

The Bluetooth module HC-05 is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS, PC to your embedded project, etc. Just go through the datasheet for more details.

**Software Features:**

1. Slave default Baud rate: 9600, Data bits:8, Stop bit:1, Parity: No parity.
2. Auto‐connect to the last device on power as default.
3. Permit pairing device to connect as default.
4. Auto‐pairing PINCODE:”1234” as default.

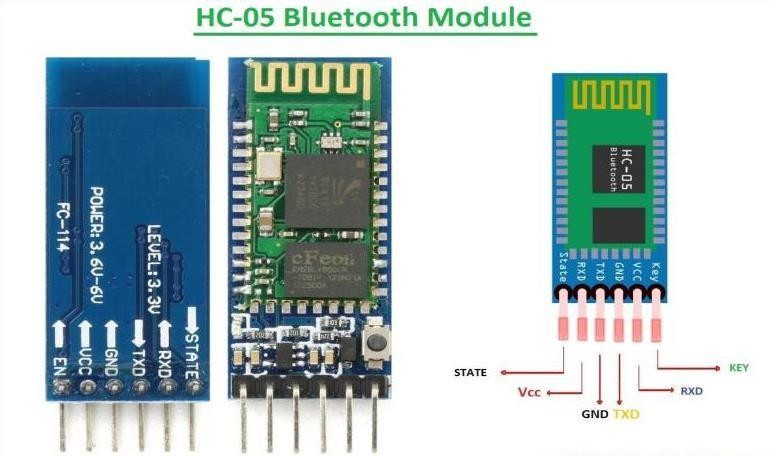


Fig.4.3 Bluetooth Module (HC-05)

The HC-05 Bluetooth Module has 6pins. They are as follows:

**ENABLE:** When enable is pulled LOW, the module is disabled which means the module will not turn on and it fails to communicate. When enable is left open or connected to 3.3V, the module is enabled i.e., the module remains on and communication also takes place.

**Vcc:** Supply Voltage 3.3V to 5V

**GND:** Ground pin

**TXD & RXD:** These two pins acts as an UART interface for communication

**STATE:** It acts as a status indicator. When the module is not connected to **/** paired with any other Bluetooth device, signal goes Low. At this low state, the led flashes continuously which denotes that the module is not paired with other device. When this module is connected to/paired with any other Bluetooth device, the signal goes high. At this high state, the led blinks with a constant delay say for example 2s delay which indicates that the module is paired.

* 1. **LIGHT DEPENDENT RESISTOR (LDR)**

A Light Dependent Resistor, is a type of passive electronic component that changes its resistance in response to the intensity of incident light. They are made of semiconductor materials that exhibit a decrease in resistance as the light level increases. This property makes them highly useful in various applications, such as light-sensitive switches, ambient light sensors in electronic devices, and even in photography equipment to control exposure settings. LDRs are an integral part of many circuits where the response to changing light conditions is required for automatic control or monitoring.

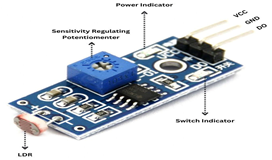


Fig.4.4 Light Dependent Resistor (LDR)

It works on the principle of photoconductivity whenever the light falls on its photoconductive material, it absorbs its energy and the electrons of that photoconductive material in the valence band get excited and go to the conduction band and thus increasing the conductivity as per the increase in light intensity. Also, the energy in incident light should be greater than the bandgap gap energy so that the electrons from the valence band got excited and go to the conduction band.

**16×2 LCD DISPLAY**

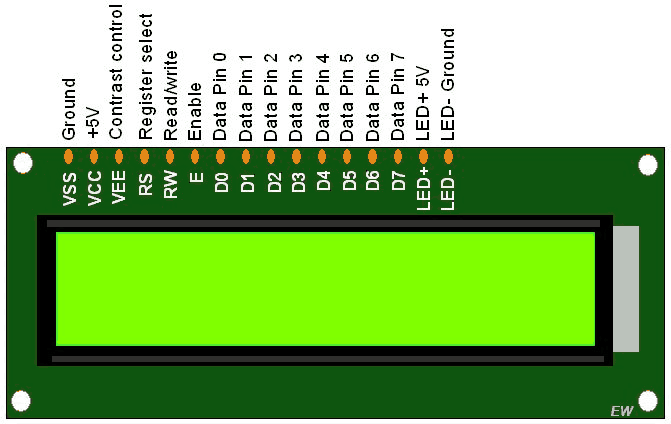
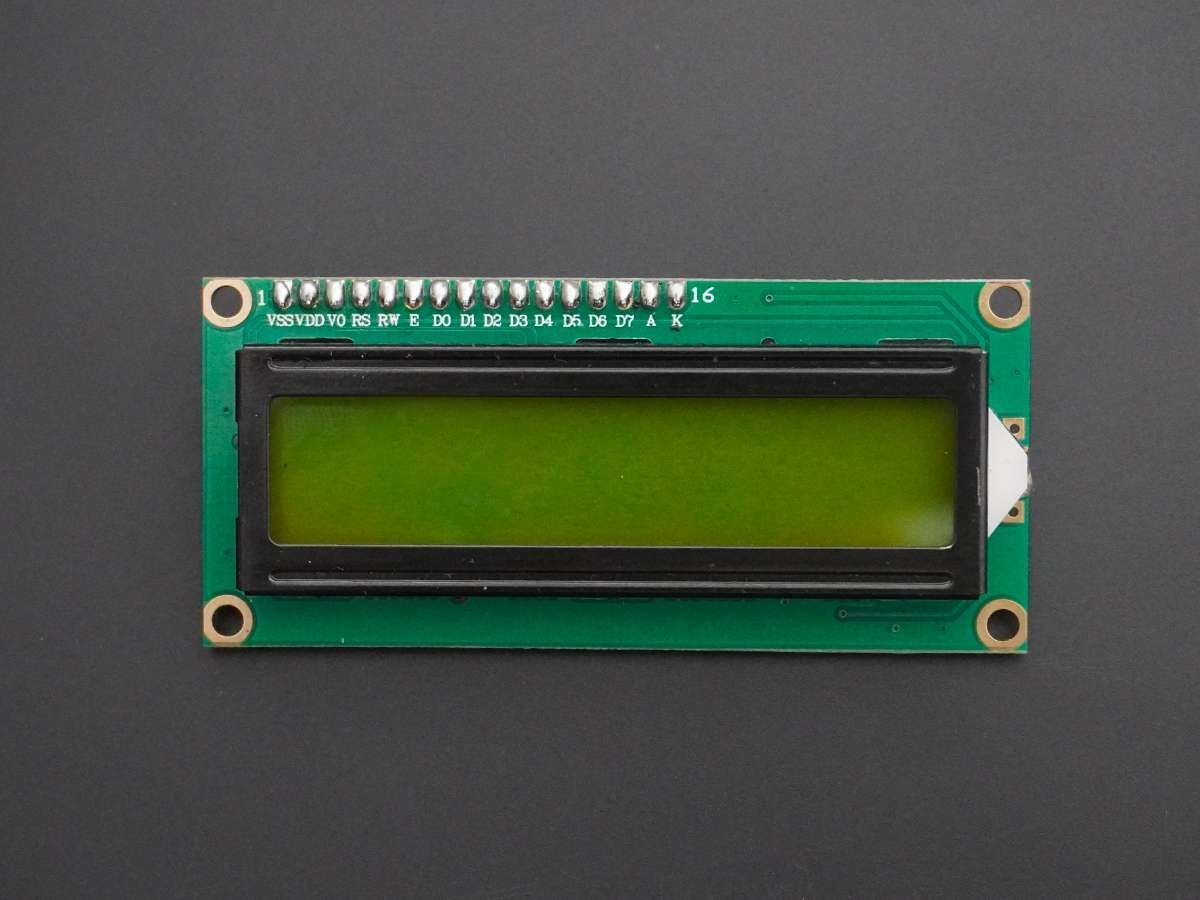


Fig.4.5 16×2 LCD Display

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

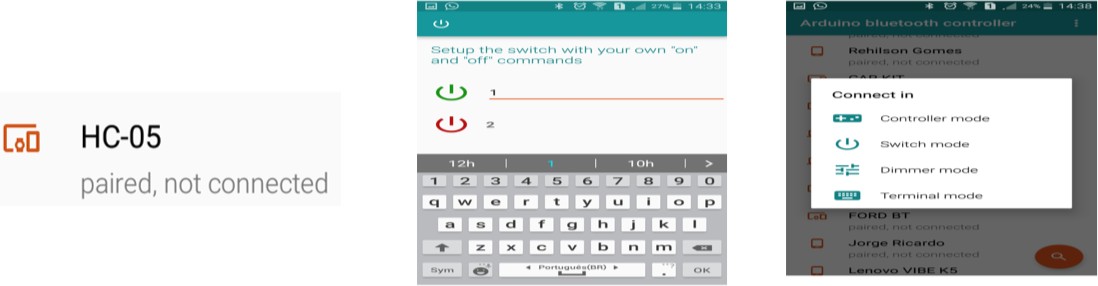
 **THE ANDROID APPLICATION**

Fig.4.6 The Android Application

1. Download the android app to your android device.
2. Open the app when the download finishes and search for the module on the Bluetooth device list.
3. Tap HC-05 and if it requests the password or pin code, type 0000 or 1234.
4. Select the “Switch Mode”.
5. Go on the configurations and tap on the “On” icon to configure the “message” to be sent as “1”.
6. Configure the “Off” icon to “2” and tap the upper-left corner icon to leave configuration.
7. Touch the app switch to turn on and off the Arduino LED attached.
8. Open the Serial Monitor on Arduino IDE and observe the messages being printed as well as you turn on and off the App switch.

**DESCRIPTION OF FUNCTIONALITY**

The hand gesture recognition system is designed to detect specific hand movements and convert them into meaningful actions using a combination of sensors, microcontrollers, and wireless communication. Below is a detailed description of each functional step:

1. Gesture Detection

- LDR Sensors (Light Dependent Resistors) are placed on a wearable device to detect changes in light intensity caused by finger movements.

- When a user moves their fingers, the light levels around the LDRs change, providing a signal that indicates a gesture.

1. Data Acquisition and Processing

- The sensor readings are continuously transmitted to an Arduino microcontroller.

- The Arduino processes the raw sensor data, identifying patterns that match predefined gestures stored in its code.

- Each recognized pattern corresponds to a specific command or message.

1. Wireless Data Transmission

- Once a gesture is identified, the Arduino sends the corresponding command or message to an HC-05 Bluetooth module.

- The Bluetooth module transmits the data wirelessly to a connected mobile application on an Android device.

1. Message Interpretation and Output

- The mobile application receives the data from the Bluetooth module.

- It compares the incoming command with a library of stored messages to determine the appropriate response.

- The app then performs two key functions:

- Text-to-Speech Conversion: Converts the message into audible speech, allowing users to hear the output.

- Screen Display: Displays the message on the mobile screen for visual feedback and clarity.

1. Power Supply and Portability

- The entire system is powered by a battery, ensuring portability and enabling the user to operate it anywhere without the need for a fixed power source.

- This makes the system ideal for mobile applications, wearable devices, and assistive technology.

1. User Interaction

- The user interacts with the system by performing specific hand gestures.

- In response, the system provides real-time feedback, allowing the user to communicate or control devices wirelessly.

**PICTURES**

