# Experiment No. : 1

# Human Machine Interaction
# To understand the trouble of interacting with machines - Redesign interfaces of home appliances.

# Experiment No. 1

1. **Aim:** To understand the trouble of interacting with machines - Redesign interfaces of home appliances like microwave oven, land-line phone, fully automatic washing machine.

2. **Objectives:** From this experiment, the student will be able to
   - To understand the importance of human psychology in designing good interfaces.
   - To encourage students to indulge into research in Machine Interface Design

3. **Outcomes:** The learner will be able to
   - Apply HMI in their day – to – day activities.
   - To analyze the local and global impact of computing on individuals, organizations, and society.
   - An ability to recognize the need for, and an ability to engage in life-long learning.

4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

Human-machine interface (HMI) is a component of certain devices that are capable of handling human-machine interactions. The interface consists of hardware and software that allow user inputs to be translated as signals for machines that, in turn, provide the required result to the user. Design is concerned with how things work, how they are controlled, and the nature of the interaction between people and technology. When done well, the results are brilliant, pleasurable products. When done badly, the products are unusable, leading to great frustration and irritation. Or they might be usable, but force us to behave the way the product wishes rather than as we wish.

Two of the most important characteristics of good design are ***discoverability*** and ***understanding***.

- ***Discoverability:*** Is it possible to even figure out what actions are possible and where and how to perform them?

- ***Understanding:*** What does it all mean? How is the product supposed to be used? What do all the different controls and settings mean?

With complex devices, discoverability and understanding require the aid of manuals or personal instruction. We accept this if the device is indeed complex, but it should be unnecessary for simple things.

**Fundamental Principles of Interaction**

1. **Affordances:** Convey the rules by leaving visual clues. To make sure that the appropriate actions are perceivable, and non-accessible ones are not invisible. By just the appearance of any object, its functionality must be clear to the user.
Example: by looking at the handles, we should know how the door opens.

2. **Signifiers:** This is a physical form of showing the functionality to the user, such as a sound, printed word or image.
Example: writing word "PUSH" on a door is a clear way to tell the user that door will open when pushed.

3. **Feedback:** The effect of every action. A feedback in any form is very critical to the user.
Example: washing machine example, the user did not get any kind of feedback from the system. That made the user assume the system is faulty. Every single user action has to be acknowledged immediately.

4. **Constraints**: Prevent the user from making mistakes. Instead of having an option for the user to make a mistake and then forgiving them, make sure your user can never make a mistake.
Example, you want your user to enter a date. Show a pick-n-click calendar instead of a textbox. This will eliminate all possibilities of syntax mismatches

5. **Mapping:** Mapping gives the set of possible relations between objects. There should be Control-display compatibility between the objects visible. Mapping gives the natural relationship between controls and displays. **E**xample: mapping of stove controls to elements.

**Conceptual Model**

- **Conceptual models** define a good design as the communication between the designer and the user. The designer must be able to explain the entire product to the user by just appearance of it.

These models are very critical for a good user experience. If the end product does not map to the mental images of the user, the product is not a success.

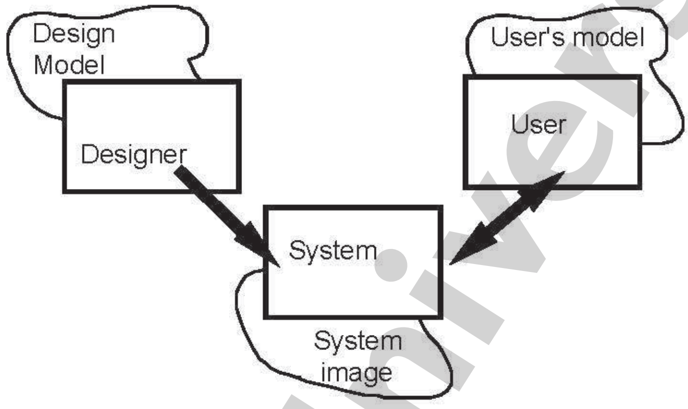



**Fig. 1: Conceptual Model for developing any product**

- ✓ The **affordance** induces a thought of action.
- ✓ **Signifiers** affirm that thought.
- ✓ **Mapping** his action to a consequence provides a feedback.
- ✓ Good **feedback** encourages learning.
- ✓ **Repeated learning** creates a **mental model** in the user about system.

## 6. Procedure:

Select any home appliances like microwave oven, land-line phone, fully automatic washing machine and understand the trouble of interacting with that machine. Comment on design of that machine as good or bad design based on whether interaction principles are matching with users mental model or not. Redesign the interface for mention the change in design and reason.

## 7. Conclusion:

People have "mental models" of how things work, built from different interaction principles. It is responsibility of the designers to first understand the mental model and then try to make a product that will agree to it.

## 8. Viva Questions:

- What is human machine interface?
- What you mean by psychopathology?
- What are characteristics of good design?
- Which are different fundamental principles of Interaction?
- What is conceptual model and mental model>

# Human Machine Interaction

# Experiment No. : 2

# To design a system based on user-centered approach.

# Experiment No. 2

1. **Aim:** To design a system based on user-centered approach

2. **Objectives:** From this experiment, the student will be able to,
   - To bring out the creativity in each student – build innovative applications that are user friendly.
   - To motivate students to apply HMI in their day – to – day activities

3. **Outcomes:** The learner will be able,
   - To analyze user models and develop user centric interfaces
   - To analyze the local and global impact of computing on individuals, organizations, and society.
   - To engage in life-long learning development and higher studies.
   - To understand, identify, analyze and design the problem, implement and validate the solution including both hardware and software.

4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

*"Computer System Design is the process that consists of a user-oriented approach where all meaningful activities are systematically framed together and utmost importance is given to the person who is going to use that system"*

Following are the activities involved in design process :

- ✓ Understanding user requirements.
- ✓ Understanding purpose of the user.
- ✓ Understanding the work environments.

In Human Machine Interface, we should make a conscious effort to consider humanity. Here the major focus is to design system based on user centered approach. The other alternate is system-centered approach in which importance is given on performance of system.

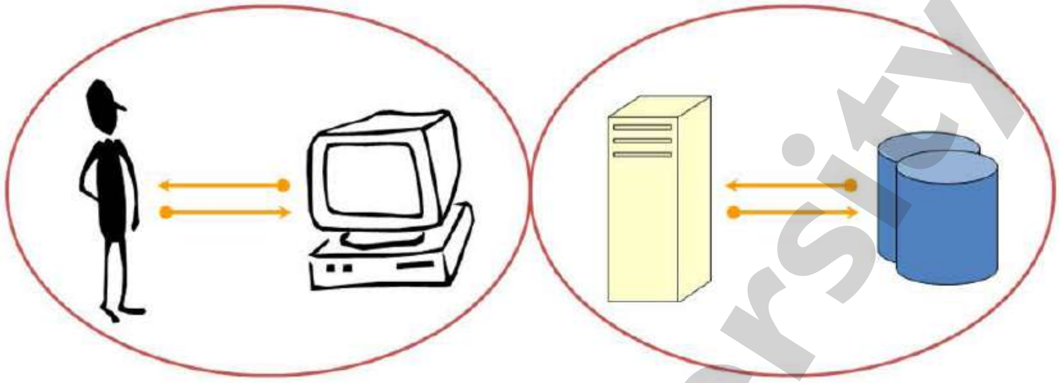



Fig. Difference between User- Centered and System-Centered Approach

The success of any product or digital design is totally dependent on how it meets the requirements of stakeholders and the product itself. The designer should always understand the user. It is not the case that the designer should understand who the user is but the experience level of the user should be totally accommodated. The deep knowledge of the user can not only be achieved by analyzing the numbers obtained in quantitative study during market research but also the deep knowledge **(behavior, attitude, and aptitude)** about the user will be obtained in various qualitative research techniques.

Qualitative research helps us understand the domain, context and constraints of a product in different and more useful ways than quantitative research does. It also quickly helps us identify patterns of behavior among users and potential users of a productmuch more quickly and easily than would be possible with quantitative app roaches.

6. **Procedure:**

Know your client: Select anyone category of user and develop application understanding the user who will be using your system. Comment on the category of user selected and specific features given for the users and identify what kinds of interfaces will they like and why?. Compare with existing system analyze and rate them.

a. **Children (4-5 years of age):** An application to teach math.

Perform analysis of children behavior e.g. their preferences, interests etc

b. **Teenagers**: Design a digital diary for young teens to help them overcome various social pressures they deal with during their teen years. The diary should also be like a self help tool which would help them deal with incidents like bullying, peer pressure, etc.. This is an open project and you can think in any direction to make the children sail through their teen years while trying to discover life around them.

Perform analysis of teenagers e.g. their problems, interests, needs, etc

c. **Older generation:** Folks from the older generation has been very wary of using their credit card on the Internet. They have various concerns when it comes to paying their bills. Also because of their old age, it will be beneficial for them to use the internet and pay their phone, electricity, gas, etc. bills

Analysis of old people e.g. their nature, interests, needs, etc

d. **Rural people:** ATVM for train ticketing in rural area

Perform analysis of rural people e.g. their problems, interests, needs, language etc

e. **Mentally disabled:** Design the interface of a game for mentally disabled children. Analysis of mentally disabled e.g. their behavior, problems, interests…

## 7. Conclusion:

User-centered approach helps to study detailed, composite user archetypes that represent distinct groupings of behaviors, attitudes, aptitudes, goals, and motivations observed and identified for any user.

## 8. Viva Questions:

- What are steps required for goal directed design process?
- What do you mean by skill spectrum of users?
- Difference between qualitative and quantitative research.
- What does Persona means?

# Human Machine Interaction

## Experiment No. : 3

## To understand the principles of good screen design by heuristic evaluation.

# Experiment No. 3

1. **Aim:** To understand the principles of good screen design by heuristic evaluation.

2. **Objectives:** From this experiment, the student will be able to,

   - To stress the importance of a good interface design
   - To understand the importance of human psychology in designing good interfaces.
   - To motivate students to apply HMI in their day – to – day activities.

3. **Outcomes:** The learner will be able,

   - To understand and apply principles of a good interface design.
   - To analyze the local and global impact of computing on individuals, organizations, and society.

4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

   The design goals in creating a user interface are described below.

   1. **Aesthetically Pleasing:** Visual appeal is provided by following the presentation and graphic design principles
      - Provide meaningful contrast between screen elements.
      - Create groupings.
      - Align screen elements and groups.
      - Provide three-dimensional representation.
      - Use color and graphics effectively and simply.

   2. **Clarity:** The interface must be clear in visual appearance, concept, and wording.
      - Visual elements should be understandable, relating to the user's real-world concepts and functions.
      - Metaphors, or analogies, should be realistic and simple.

- Interface words and text should be simple, unambiguous, and free of computer jargon.

3. **Compatibility**: Provide compatibility with the following:
   - ***The User:*** Design must be appropriate and compatible with the needs of the user or client. "Know the user" is *the* fundamental principle in interface design.
   - ***The task and job:*** The organization of a system should match the tasks a person must do to perform the job. The structure and flow of functions should permit easy transition between tasks. The user must never be forced to navigate between applications or many screens to complete routine daily tasks.
   - ***The product:*** Compatibility across products must always be considered in relation to improving interfaces, making new systems compatible with existing systems. As a result it reduces the necessity for new learning as user is aware of the system.

4. **Comprehensibility:** A system should be understandable, flowing in a comprehensible and meaningful order.
5. **Configurability:** Permit easy personalization, configuration, and reconfiguration of settings.
   - Enhances a sense of control.
   - Encourages an active role in understanding.

6. **Consistency**: A system should look, act, and operate the same throughout.
   - Similar components should:
     - ☐ Have a similar look.
     - ☐ Have similar uses.
     - ☐ Operate similarly.
   - The same action should always yield the same result.
   - The function of elements should not change.
   - The position of standard elements should not change.
7. **Control:** The user must control the interaction.
   - Actions should result from explicit user requests.
   - Actions should be performed quickly.
   - Actions should be capable of interruption or termination.
   - The user should never be interrupted for errors.

8. **Directness:** Provide direct ways to accomplish tasks.
    - Available alternatives should be visible.
    - The effect of actions on objects should be visible.
9. **Efficiency:** Minimize eye and hand movements, and other control actions.
    - Transitions between various system controls should flow easily and freely.
    - Navigation paths should be as short as possible.
    - Eye movement through a screen should be obvious and sequential.
10. **Familiarity:**
    - Employ familiar concepts and use a language that is familiar to the user.
    - Keep the interface natural, mimicking the user's behavior patterns.
    - Use real-world metaphors.
11. **Flexibility**: A system must be sensitive to the differing needs of its users, enabling a level and type of performance based upon:
    - Each user's knowledge and skills.
    - Each user's experience.
    - Each user's personal preference.
    - Each user's habits.
    - The conditions at that moment.
12. **Forgiveness:**
    - Tolerate and forgive common and unavoidable human errors.
    - Prevent errors from occurring whenever possible.
    - Protect against possible catastrophic errors.
    - When an error does occur, provide constructive messages.
13. **Predictability**
    - The user should be able to anticipate the natural progression of each task.
        - ☐ Provide distinct and recognizable screen elements.
        - ☐ Provide cues to the result of an action to be performed.
    - All expectations should be fulfilled uniformly and completely.
14. **Recovery**
    - A system should permit:
        - ☐ Commands or actions to be abolished or reversed.
        - ☐ Immediate return to a certain point if difficulties arise.
    - Ensure that users never lose their work as a result of:

- ☐ An error on their part.
- ☐ Hardware, software, or communication problems.

**15. Transparency**

- Permit the user to focus on the task or job, without concern for the mechanics of the interface.

## 6. Procedure:

Identify 5 different websites catering to one specific goal (eg. Goal – on-line shopping and 5 different websites – ebay, amazon, flipkart, zovi, myntra) and perform a competitive analysis on them to understand how each one caters to the goal, the interactions and flow of the payment system and prepare a report on the same. Consider any 8 HCI principles and prepare the following table evaluating the websites.

| Sr. No. | Principles | Poor | Average | Good | Very good | Excellent |
|---|---|---|---|---|---|---|
| 1 | **Aesthetically pleasing** | | | | | |
| 2 | .. | | | | | |

## 7. Conclusion:

The design goals in creating user interface are fundamental to the implementation of all effective interfaces including GUI and Web ones. These principles are general characteristics of the interface and they apply to all aspects.

## 8. Viva Questions:

- Difference between Graphical User Interface and Web User Interface.
- Which are different interaction styles?
- What are characteristics of GUI?

# Human Machine Interaction

## Experiment No. : 4

## To calculate screen complexity of existing Graphical User Interface and redesign the interface to minimize the screen complexity.

# Experiment No. 4

1. **Aim:** To calculate screen complexity of existing Graphical User Interface and redesign the interface to minimize the screen complexity.

2. **Objectives:** From this experiment, the student will be able to,

    - To stress the importance of a good interface design
    - To motivate students to apply HMI in their day – to – day activities.

3. **Outcomes:** The learner will be able,

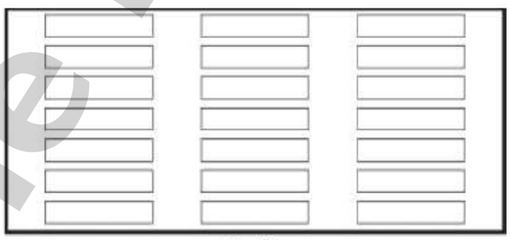
    - To understand and apply principles of a good interface design.
    - To analyze the local and global impact of computing on individuals, organizations, and society.
    - To engage in life- long learning and match the industry requirements in HMi domains

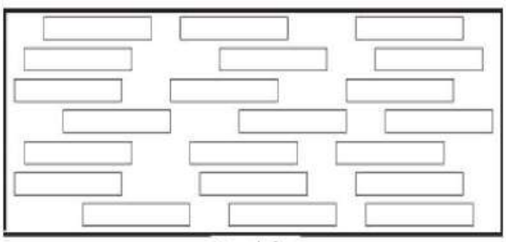
4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

***Simplicity***, illustrated in Figure, is directness and singleness of form, a combination of elements that results in ease of comprehending the meaning of a pattern. The opposite pole on the continuum is ***complexity.***

To achieve simplicity one needs to optimize the number of elements on a screen, within limits of clarity. And minimize the alignment points, especially horizontal or columnar.


Simplicity


Complexity

**Method for Measuring Complexity:**

1. Draw a rectangle around each element on a screen, including captions, controls, headings, data, title, and so on.
2. Count the number of elements and horizontal alignment points (the number of columns in which a field, inscribed by a rectangle, starts).
3. Count the number of elements and vertical alignment points (the number of rows in which an element, inscribed by a rectangle, starts).
4. Calculate number of bits required by horizontal (column) alignment points and number of bits required by vertical (row) alignment points by applying following formula for calculating the measure of complexity.

$$C = -N \sum_{n=1}^{m} p_n \log_2 p_n$$

   ***C***, complexity of the system in bits

   ***N***, total number of events (widths or heights)

   ***m,*** number of event classes (number of unique widths or heights)

   ***pn***, probability of occurrence of the nth event class (based on the frequency of events within that class)

5. Calculate overall complexity by adding the number bits required by horizontal alignment points and vertical alignment points.

**Guidelines for reducing complexity:**

1. The way to minimize screen complexity is to reduce the number of controls displayed. Fewer controls will yield lower complexity measures.
2. Optimize the number of elements on a screen, within limits of clarity.
3. **Alignment:** Minimize the alignment points, especially horizontal or columnar. Fewer alignment points will have a strong positive influence on the complexity calculation. When things don't align, a sense of clutter and disorganization often results. In addition to reducing complexity, alignment helps create balance, regularity, sequentiality, and unity.

6. **Procedure:**

1. Calculate Screen Complexity for existing Graphical User Interface (GUI).
2. Redesign the Screen by applying various guidelines to lower the complexity of selected Graphical User Interface (GUI) to achieve simplicity.

7. **Conclusion:**

Good alignment is related to shorter screen search times and higher viewer preferences for a screen. Misalignments and uneven spacing, no matter how slight, can create bothersome unconscious disruptions to our perceptual systems.

8. **Viva Questions:**

- What is need of alignment and grouping?
- How alignment helps in reducing complexity of screen.

# Human Machine Interaction

# Experiment No. : 5

# To design web user interface based on Gestalt Theory.

# Experiment No. 5

1. **Aim:** To design web user interface based on Gestalt Theory

2. **Objectives:** From this experiment, the student will be able to

    - To stress the importance of a good interface design.
    - To understand the importance of human psychology in designing good interfaces
    - To motivate students to apply HMI in their day – to – day activities

3. **Outcomes:** The learner will be able to

    - To analyzes human responses and develop guidelines for the best user experience.
    - To understand, identify, analyze and design the problem, implement using current techniques and skills.
    - To engage in continuing professional development and higher studies.

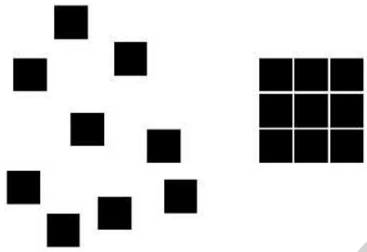
4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

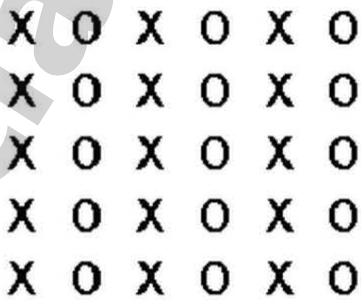
Gestalt is a German word meaning "essence or shape of an entity's complete form", and this single definition may be one of the most important rules of design. More often than not, designers tend to focus on the web design's details rather than the overall look. They focus on curved edges, shadows, fonts…etc. all that is good but may not really make any difference if the client doesn't like the design at first glance, what most people don't understand is that the brain first sees the overall shape of any design, then starts to focus on and see the details

1. **Proximity:** Proximity is similar to common regions but uses space as the enclosure. When elements are positioned close to one another, they are seen as part of a group rather than as individual elements. This is especially true when the elements in the group are closer to each other than they are to any elements outside the group. The picture below represents **arrangement of boxes on the**

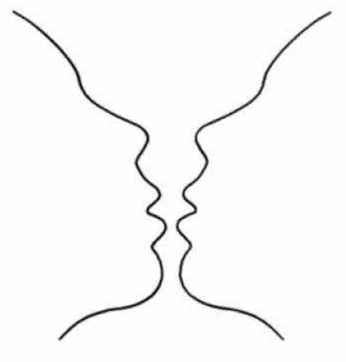
**left are not close** enough to have proximity, while the group on the right is perceived as a single whole element



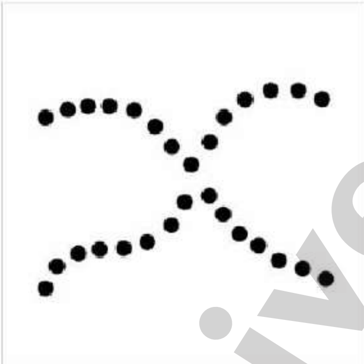
2. **Similarity:** From the name of this principle it is easy to understand that this will be used in the case of those items that are alike. These objects will be perceived as making part of a group and their similarity could consist of the size, color, form, length, and so on. People find it easy to navigate web page thanks to the similarity principle, which could be used either to make the most important elements stand out or maybe to direct the attention of your audience towards certain links, icons or images. In the picture below, we quickly perceive rows of circles and X's rather than the individual letters.




3. **Closure:** Closure refers to the tendency to perceive the whole in objects, filling in the necessary missing bits of information. For example, the below picture is perceived as two faces or a vase rather than two symmetrical lines.



4. **Continuity:** Continuity refers to the inclination to see objects as continuous, a smooth progression rather than parts. For example, in the below picture, we see two lines intersecting rather than a series of small dots.



6. **Procedure:**

Design/Redesign web user interface based on Gestalt theories and comment on the principle applied and justify. Also analyze one image in which Gestalt principle is applied and comment.

**Example:**

Take a look at old IBM logo:

You recognize the letters as an I, a B, and an M, no problem there. But they aren't letters at all; the whole thing is a compilation of bright blue horizontal lines arranged to create the perception of a set of letters.

Gestalt Property used here is **Closure.**

Closure means that we "close" objects that are themselves not complete; not only completing the figure in our perception, but perceiving the figure as having an extra element of aesthetic design; we look for a simple, recognizable pattern.

## 7. Conclusion:

At its simplest, gestalt theory describes how the mind organizes visual data. The stronger the clarity of form, the more effective the design will be.

## 8. Viva Questions:

- Define Perception.
- Discuss how perception is biased by experience, goals and current context.
- Which are different Gestalt Principles?

# Human Machine Interaction

## Experiment No. : 6

## Implementation of Menus for Graphical System

# Experiment No. 6

1. **Aim:** Implementation of Menus for Graphical System

2. **Objectives:** From this experiment, the student will be able,

    - To stress the importance of a good interface design.
    - To bring out the creativity in each student – build innovative applications that are user friendly.

3. **Outcomes:** The learner will be able to

    - To apply the core concepts and implementation guidelines of Human Computer Interaction to improve them
    - To understand, identify, analyze and design the problem, implement using current techniques and skills.
    - To engage in continuing professional development and higher studies.

4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

A system contains large amounts of information and performs a variety of functions. It should also provide some means to tell people about the information it possesses or the things it can do. This is accomplished by displaying listings of the choices or alternatives the user has at appropriate points while using the system. These listings of choices are commonly called ***menus.*** Menus are a major form of navigation through a system and, if properly designed, assist the user in developing a mental model of the system.

Providing the proper kinds of graphical menus to perform system tasks is also critical to system success. The best kind of menu to use in each situation depends on several factors. The following must be considered:

- ☐ The number of items to be presented in the menu.
- ☐ How often the menu is used.
- ☐ How often the menu contents may change.

## a. Menu Bar

- A menu bar consists of a collection of descriptions that serve as headings or titles for a series of actions on an associated pull-down menu.
- The menu is typically arrayed in a horizontal row at the top of a window. Occasionally a menu bar is referred to as a collection of menu *titles.* In reality it is a menu in itself, and it is appropriate to simply refer to it as a *menu*. A menu bar is the starting point for many dialogs. Menu bars often consist of a series of textual words, as represented in fig.


**Fig: Menu bar Composed of Text**

- All primary windows must have a menu bar.
- All menu bars must have an associated pull-down menu containing at least two choices.
- Do not allow the user to turn off the display of the menu bar.
- Position choices horizontally over the entire row at the top of the screen, just below the screen title.
- A typical bar is composed of no more than about seven or eight choices.
- The menu item descriptions must clearly reflect the kinds of choices available in the associated pull-down menus.
- Use mixed-case letters to describe choices.
- Use single-word choices whenever possible.
- Order choices left-to-right with:
  - ☐ Most frequent choices to the left.
  - ☐ Related information grouped together.
- Left-justify choices within the line.
- Help, when included, should be located at the right side of the bar.
- Indent the first choice one space from the left margin.
- Leave at least three spaces between each of the succeeding choices
- Keyboard cursor:
  - ☐ Use a reverse video, or reverse color, selection cursor to surround the choice.

- ☐ Cover the entire choice, including one blank space before and after the choice word.


**b. Pull-Down Menu**

- Selection of an alternative from the menu bar results in the display of the exact actions available to the user.
- These choices are displayed in a vertically arrayed listing that appears to pull down from the bar.
- Hence, these listings, as illustrated in fig, are typically referred to as *pull-downs*. Other identification terms may be used, such as d*rop-downs*.

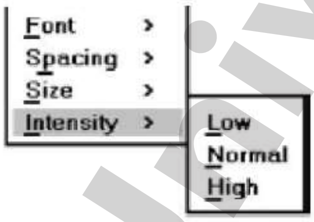

**Fig: Menu bar pull – down**

- Display all possible alternatives.
- Gray-out or dim items that cannot be chosen due to the current state of an application.
- Position the pull-down directly below the selected menu bar choice.
- Must contain a minimum of two choices.
- Restrict to no more than 5 to 10 choices, preferably 8 or less.
- Use mixed-case, headline-style words to describe choices.
- Do not:
  - ☐ Identify a menu item by the same wording as its menu title.
  - ☐ Change the meaning of menu items through use of the Shift key.
  - ☐ Use scrolling in pull-downs.
  - ☐ Place instructions in pull-downs.
- Place frequent or critical items at the top.
- Left-align choice descriptions.
- Leave the menu bar choice leading to the pull-down highlighted in the selected manner (reverse video or reverse color).

- Physically, the pull-down menu must be wide enough to accommodate the longest menu item description and its cascade or accelerator indicator.
- Align the first character of the pull-down descriptions under the second character of the applicable menu bar choice.

**c. Cascading Menu**

- A cascading menu is a submenu derived from a higher-level menu, most typically a pull-down.
- Cascading menus are located to the right of the menu item on the previous menu to which they are related, as illustrated in fig.




- Place an arrow or right-pointing triangle to the right of each menu choice description leading to a cascade menu.
- Separate the indicator from the choice description by one space.
- Display the indicator in the same color as the choice descriptions.
- Position the first choice in the cascading menu immediately to the right of the selected choice.
- Leave the choice leading to the cascading menu highlighted.

**d. Pop-up Menus**

- Choices may also be presented to the user on the screen through *pop-up* menus, vertically arrayed listings that only appear when specifically requested.
- Pop-up menus may be requested when the mouse pointer is positioned over a designated or *hot* area of the screen (a window border or text, for example) or over a designated icon. In look, they usually resemble pull-down menus, as shown in figure.

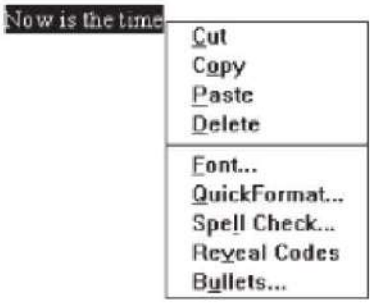



**Fig: Pop-up Menu**

e. **Tear-off Menus**

- A tear-off menu is a pull-down menu that can be positioned anywhere on the screen for constant referral.
- As such, it possesses all the characteristics of a pull-down. It may also be called a *pushpin, detachable,* or *roll-up* menu.
- Its purpose is to present alternatives or choices to the screen user that are needed infrequently at some times and heavily at other times.

## 6. Procedure:

Design an application which consists of different types of menus. Apply and explain general menu design guidelines applied for formatting, ordering, phrasing, selecting choices, and navigating menus for application which is designed.

## 7. Conclusion:

Menus are effective because they utilize the more powerful human capability of recognition rather than the weaker capability of recall. Working with menus reminds people of available options and information that they may not be aware of or have forgotten.

## 8. Viva Questions:

- Which are different kinds of menus?
- Discuss guidelines for menus choices.
- What are different structures of menus

# Human Machine Interaction

# Experiment No. : 7

# Implementation of Different Kinds of Windows.

# Experiment No. 7

1. **Aim:** Implementation of Different Kinds of Windows.

2. **Objectives:** From this experiment, the student will be able,

   - To stress the importance of a good interface design.
   - To bring out the creativity in each student – build innovative applications that are user friendly.

3. **Outcomes:** The learner will be able to
   - To apply the core concepts and implementation guidelines of Human Computer Interaction to improve them
   - To understand, identify, analyze and design the problem, implement using current techniques and skills.
   - To engage in continuing professional development and higher studies.

4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

A window is an area of the screen, usually rectangular in shape, defined by a border that contains a particular view of some area of the computer or some portion of a person's dialog with the computer. It can be moved and rendered independently on the screen. A display may contain one, two, or more windows within its boundaries.

People's tasks must be structured into a series of windows. The type of window used will depend on the nature and flow of the task. There are two types of windows: Primary Window and Secondary Windows.

**Primary Window**: The *primary* window is the first one that appears on a screen when an activity or action is started. A primary window should contain constantly used window components such as frequently used menu bar items and controls (for example, control bars) used by dependent windows

**Secondary Windows**

Secondary windows are supplemental windows. A secondary window is typically smaller than its associated primary window and smaller than the minimum display resolution. Secondary windows may be ***dependent*** upon a primary window or displayed ***independently*** of the primary window.

**Dependent Secondary Window**

- ✓ It can only be displayed from a command on the interface of its primary window.
- ✓ It is typically associated with a single data object, and appears on top of the active window when requested.
- ✓ It is movable, and scrollable.
- ✓ If necessary, it uses the primary window's menu bar.
- ✓ In general, dependent secondary windows are closed when the primary window closes, and hidden when their primary window is hidden or minimized.

**Independent Secondary Window**

- ✓ It can be opened independently of a primary window.
- ✓ An independent secondary window can typically be closed without regard to the state of any primary window unless there is an obvious relationship to the primary window.

**TYPES OF SECONDARY WINDOWS**

1. **Dialog Boxes:** Dialog boxes are used to extend and complete an interaction within a limited context. Dialog boxes are always displayed from another window, either primary or secondary, or another dialog box. They may appear as a result of a command button being activated or a menu choice being selected, or they may be presented automatically by the system when a condition exists that requires the user's attention or additional input.

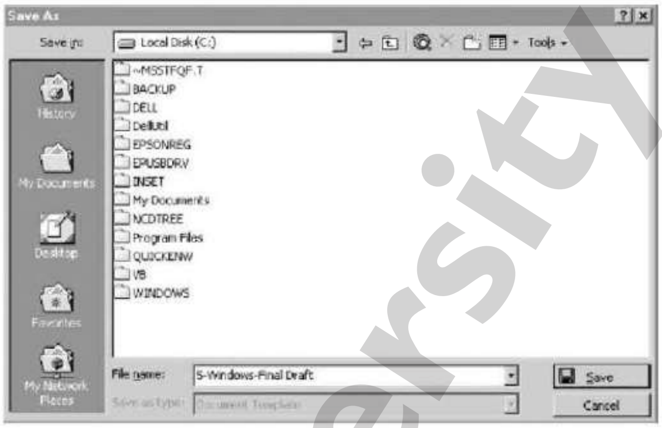



**Fig: Microsoft Windows Dialog Box**

**2 Property Sheets and Property Inspectors:** The properties of an object in an interface can be displayed in a variety of ways, for example, the image and name of an icon on the desktop. Secondary windows provide two other techniques for displaying properties, *property sheets* and *property inspectors.*

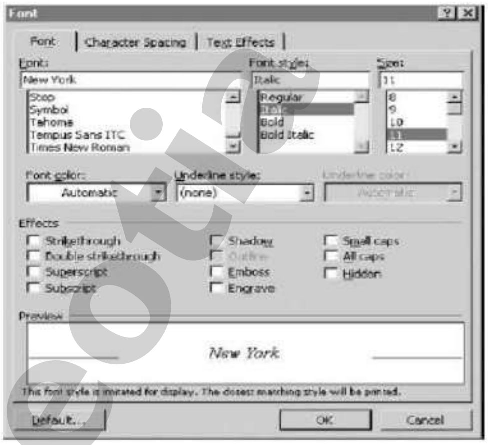




**Fig: Microsoft Windows Property Sheet Fig: Microsoft Windows Property Inspector**

**3. Message Boxes:** Use for displaying a message about a particular situation or condition. If a message requires no choices to be made but only acknowledgement, include an ok button and optionally a help menu. If the message requires the user to make a choice, include a command button for each option. Include OK and Cancel buttons only when the user has the option of continuing or stopping the action.

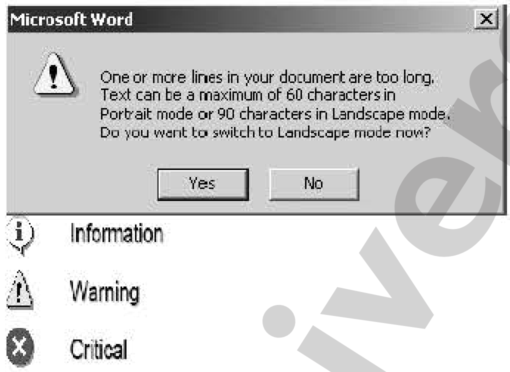



**Fig: Microsoft Windows Message Box**

4. **Palette Windows:** Palette windows are modeless secondary windows that present a set of controls. Palette windows are distinguished by their visual appearance, a collection of images, colors or patterns. The title bar for a palette window is shorter and includes only a close button.


**Fig: Microsoft Windows Palette Window**

5. **Pop-up Windows:** Use pop-up windows to display:

- Additional information when an abbreviated form of the information is the main presentation.
- Textual labels for graphical controls.
- Context-sensitive Help information.

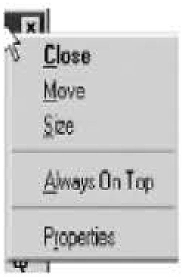


Fig: Microsoft Windows pop-up window

## 6. Procedure:

For every window designed for the application explain:

- **Purpose**
- **Description**
- **Components**
- **Kind window**

## 7. Conclusion:

Using windows it is much easier to switch between tasks and to maintain one's context, since one does not have to reestablish one's place continually. In addition,Windows provide access to more information than would normally be available on a single display of the same size. Overwriting, or placing more important information on top of that of less importance at that moment.

## 8. Viva Questions:

- Difference between dependent secondary window and independent secondary window.
- Explain different types of secondary windows.
- What are Palette windows?
- Which are different windows presentation styles.
- Explain modal and modeless property of secondary window.

# Human Machine Interaction

# Experiment No. : 8

# To understand how to design appropriate icons.

# Experiment No. 8

1. **Aim:** To understand how to design appropriate icons.

2. **Objectives:** From this experiment, the student will be able,

    - To stress the importance of a good interface design.
    - To bring out the creativity in each student – build innovative applications that are user friendly.

3. **Outcomes:** The learner will be able to
    - To apply the core concepts and implementation guidelines of Human Computer Interaction to improve them
    - To understand, identify, analyze and design the problem, implement using current techniques and skills.
    - To engage in continuing professional development and higher studies.

4. **Hardware / Software Required:** Any tool or technology can be used for implementation e.g., VB, DOTNET, JAVA, PHP, etc.

5. **Theory:**

Icons are most often used to represent objects and actions with which users can interact with or that they can manipulate. These types of icons may stand alone on a desktop or in a window, or be grouped together in a toolbar. A secondary use of an icon is to reinforce important information, a warning icon in a dialog message box, for example.

**Categories of Icon**

- ***Icon:*** Something that looks like what it means
- ***Index:*** A sign that was caused by the thing to which it refers
- **Symbol:** A sign that may be completely arbitrary in appearance

**Kinds of Icon**

- ***Resemblance:*** An image that looks like what it means
- ***Symbolic:*** An abstract image representing something
- ***Exemplar:*** An image illustrating an example or characteristic of something

- ***Arbitrary:*** An image completely arbitrary in appearance whose meaning must be learned
- ***Analogy:*** An image physically or semantically associated with something

**Characteristics of Icons**

Icons should be:

- familiar
- clear and legible
- simple
- consistent
- direct
- efficient
- discriminable

**Choosing Icons**

Icon design is an important process. Meaningful and recognizable icons will speed learning and recall and yield a much more effective system. Poor design will lead to errors, delays, and confusion. Looks different from all other icons.

- Is obvious what it does or represents.
- Is recognizable when no larger than 16 pixels square.
- Looks as good in black and white as in color.

**Icon Size**

Supply in all standard sizes.

- 16 × 16 pixels.
  - 16- and 256-color versions.
- 32 × 32 pixels
  - 16- and 256-color versions.
- 48 × 48 pixels
  - 16- and 256-color versions.
- Use colors from the system palette.
- Use an odd number of pixels along each side.
  - Provides center pixel around which to focus design.
- Minimum sizes for easy selection:
  - With stylus or pen: 15 pixels square.

- With mouse: 20 pixels square.
- With finger: 40 pixels square.

- Provide as large a hot zone as possible.

**Choosing Images**

- Use existing icons when available.
- Use images for nouns, not verbs.
- Use traditional images.
- Consider user cultural and social norms.

**The Design Process of Icons**

- **Define purpose:** To begin the design process, first define the icon's purpose and use. Have the design team brainstorm about possible ideas, considering real-world metaphors.
- **Collect, evaluate, and sketch ideas:** Start by designing on paper, not on the computer. Ask everyone to sketch his or her ideas.
- **Draw in black and white:** Many icons will be displayed in monochrome. Color is an enhancing property; consider it as such.
- **Test for expectation, recognition, and learning.** Choosing the objects and actions, and the icons to represent them, is not a precise process, and will not be easy. So, as in any screen design activity, adequate testing and possible refinement of developed images must be built into the design process. Icon recognition and learning should both be measured as part of the normal testing process.
- **Test for legibility.** Verify the legibility and clarity of the icons in general. Also, verify the legibility of the icons on the screen backgrounds chosen. White or gray backgrounds may create difficulties. An icon mapped in color, then displayed on a monochrome screen, may not present itself satisfactorily. Be prepared to redraw it in black and white, if necessary.
- **Register new icons in the system's registry.** Create and maintain a registry of all system icons. Provide a detailed and distinctive description of all new icons.

6. **Procedure:**

Identify separate lines of business, e.g., medical, greeting cards, law etc. Design an application using proper guidelines for icons. Comment on design of icons and their relevance in the system.

7. **Conclusion:**

Icons are the most often used to represent objects and actions with which users can interact with or that they can manipulate.

Experiment No: HMI/L/01

## Name of the Experiment:

Machine interface with manual switch.

## Objective:

To interface with a machine*(robot)* using a mechanical switch.

## Theory/Process:

A mechanical switch "switches" the flow of current in a circuit. When the contacts of a switch are closed, the switch creates the closed path for current flow and hence load consumes the power from source. When the contacts of a switch are open, no power will be consumed by the load. A mechanical switch is the simplest form of machine interfacing. I There are a few terms we need to know about mechanical switches. The ***pole*** represents the number of individual power circuits that can be switched. The number of ***throws*** represents the number of states to which current can pass through the switch. There are a few classifications of mechanical switches we should be familiar about. They are as follows:

1. Method of actuation:
   - Manual Switch
   - Limit Switch
   - Process Switch
2. Number of contacts:
   - Single contact switch
   - Multi contact switch
3. Number of poles and throws:
   - Single Pole Single Throw *(SPST)* switch
   - Single Pole Double Throw *(SPDT)* switch
   - Double Pole Double Throw *(DPDT)* switch
4. State based switches:
   - Momentary switch
   - Locked switch *(Latch configuration)*
5. Types of operation:
   - Push button
   - Toggle switch
   - Rotary switch

In the case of the robot, which is represented by a LED and Resistor block here, an SPST switch is used to switch the flow of current ON or OFF. The source of power here is a +24V DC battery. This arrangement is known as a kill-switch, which is used to kill the power in the case of any unlikely event, or to conserve power when not in use.

## Block Diagram/Circuit Diagram:

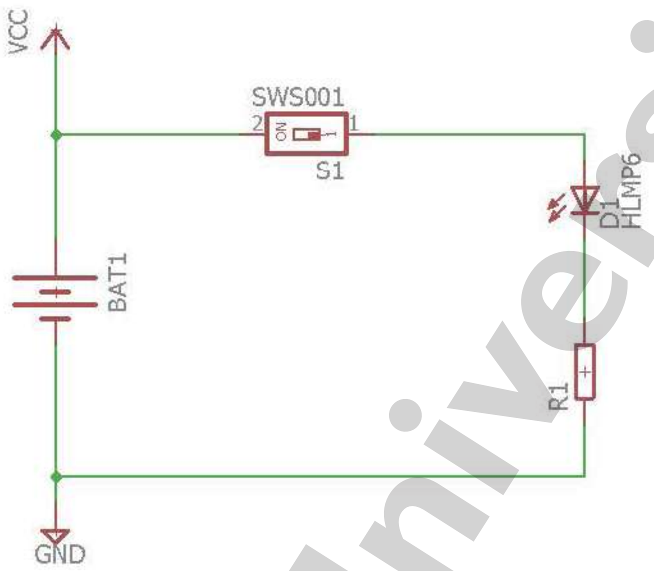



*Figure 1: The circuit diagram showing the interfacing of a switch with the robot subsystem(shown here as LED and resistor)*

## Output:

When the switch is turned ON, the current flows from the battery to the robot (LED+Resistor), effectively turning the robot ON. When the switch is thrown OFF, the flow of current is suddenly stopped, and the robot turns OFF. This hypothesis is further proven by the test results presented below. The data shows the state of the switch, and the state of the LED.

| Switch State | LED state |
| --- | --- |
| ON | ON |
| OFF | OFF |

## Discussion:

Not all switches are suitable for all types of all types of work. The selection of a switch depends on the load the switch is going to bear. In our case, the current rating of the battery was 10 Amps. Therefore, we had to use a switch which would be able to switch a load greater than 10 Amps. Also, we used a simple SPST switch for the only reason that this power circuitry is just to be turned ON/OFF. There is no multiple level selection involved here. Had we used a system where we need to switch between two different voltage levels, we would need a SPDT switch. We have used a toggle switch here. A latch type switch would allow for better functioning and easier operation.

## Name of the Experiment:

Machine interface using touch pad.

## Objective:

Interfacing with a machine *(robot)* using a touch pad.

## Theory/Process:

A touch pad is a popular machine interfacing device for industries and homes. It is an LCD screen on which a resistive/capacitive touch sheet is overlaid. This touch screen responds to the touch of a stylus or a finger, and sends out a signal, corresponding to the cartesian position of the stylus/finger on the touch screen. Touch pads are being used to control many industrial machines such as CNCs, and industrial robots, since they save space by incorporating the input device *(touch screen)* right on top of the output device*(LCD panel)*.

Today, with the advent of wireless communications, we are being able to use touch pads to a far greater extent than imagined before. A touch pad is present in the smartphones, which can communicate wirelessly with devices around our house, such as lights, fans, refrigerators, microwave ovens, etc. Therefore, using our phone as an interfacing device, and Bluetooth wireless communication as our method of communication, we can control an entire system.

We have used a smartphone, a bluetooth based app, and the robot, to perform this experiment. The basic mode of operation is that we use the smartphone, where we have a clear UI to turn the robot ON/OFF. When we touch in the appropriate area, a signal is communicated over to the robot via Bluetooth, and that actuates the power switch ON/OFF.

## Block Diagram/Circuit Diagram:

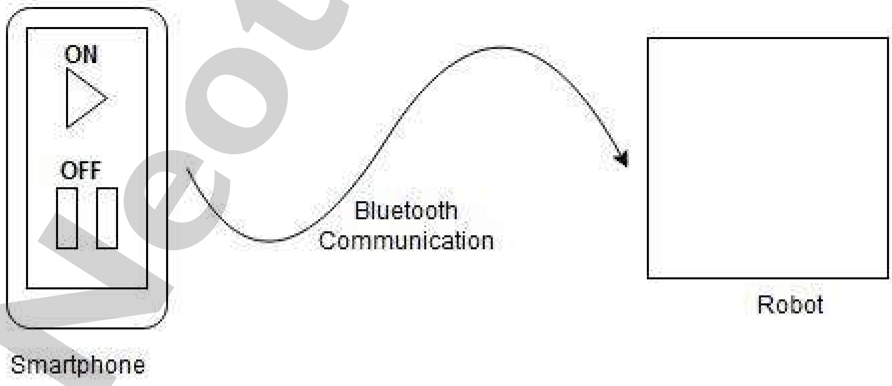



*Figure 1: The basic operation of the bluetooth app based robot interface*

## Output:

When we touch the appropriate area on the bluetooth app, it sends a signal via bluetooth to the main controller on the robotic system. This signal is caught and processed by the robot, and acted upon accordingly. This means that we can turn the robot on by pressing the triangle on the smartphone, and turn the robot off by pressing the two rectangles on the smartphone This is further illustrated in the comparision table of the states of the bluetooth app, and the states of the robot below:

| Bluetooth App | Robot |
|---|---|
| ON(Triangle is pressed) | ON(LED lights up) |
| OFF(Rectangles are pressed) | OFF(LED shuts down) |

**Discussion:**

There are a few precautions we need to take when we are performing this experiment:

- At no point should the high DC voltage of the battery be directly fed into the microcontroller, this could damage the entire circuitry.
- Ideally, the inputs should be optoisolated, to prevent any such disaster from affecting the main processing board.
- The bluetooth app should be properly paired with the robot. Failure to do so will result in a failed experiment.
- The bluetooth communication is range sensitive. This is a class B bluetooth device, which means we may expect proper communications up till 10 meters.

Experiment No: HMI/L/03

## Name of the Experiment:

Noise controlled machine interfacing.

## Objective:

To interact with a robot using audio signals as input.

## Theory/Process:

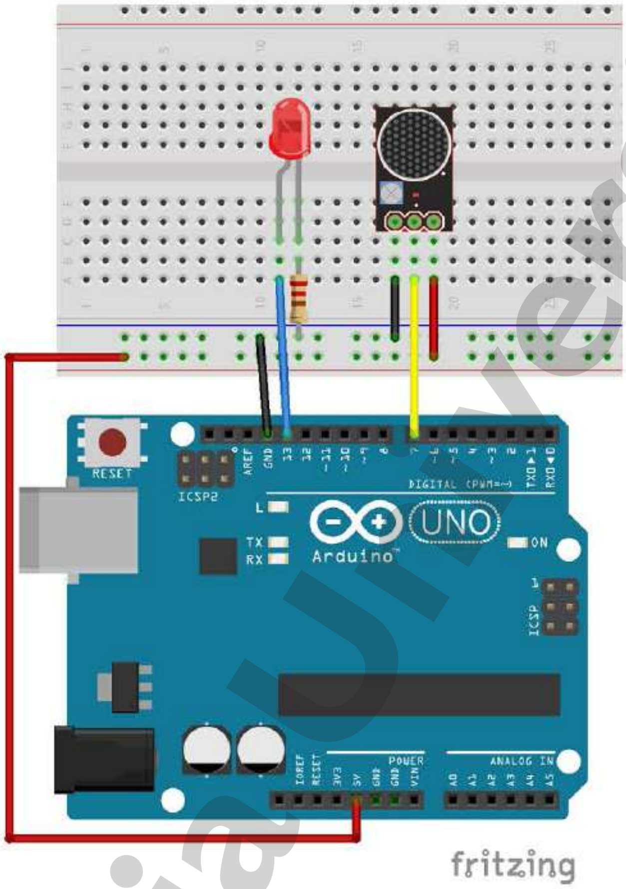
Humans can interact with machines by many different methods – Touch, gestures, and voice. In the earlier experiments, our mode of interaction has strictly been physical touch, i.e., we have physically interacted with the robot, in order to actuate it. Now we will look at the use of audio signals as commands to a machine. An audio signal is caused by the compression and rarefaction of the air from the source to the destination. This wave can be read by a microcontroller if we use any piezoelectric device to correspond to the changes in the air, which we call an audio signal. Now, even if we use the piezoelectric material to convert the audio signal to an al\nalog voltage, we need to digitize the signal and store it, in order to perform any action on it. In this case, we make use of an ADC, or an Analog-to-Digital Converter. An ADC samples the analog sound wave into digital signals, which can be stored and operated on by a microcontroller.

Also, we need to isolate a voice signal from the ambient noise of the environment. This is done by filtering the audio sample using an analog/digital filter. A filter will only allow frequencies of a certain level pass through, blocking the other frequencies. Filters can be of three types, as explained below:

- **Low pass filter:** A low pass filter allows frequencies that are lower than the cut-off frequency, or a set threshold frequency to pass through.
- **High pass filter:** A high pass filter allows frequencies that are higher than the cutoff frequency to pass through it, effectively blocking the other frequencies.
- **Band pass frequency:** A band pass filter is a mixture of a low pass filter and a high pass filter, in that it allows a band of frequencies to pass through. A band of frequencies generally mean that there are two cutoff frequency points – one for the highest frequency that the filter will allow to pass through, and one for the lowest frequency that the filter will allow to pass through. The filter will block the rest of the frequencies.

Since the range of the human voice is 10Hz to 10KHz, we will make use of a band pass filter that has a band equal to the aforementioned frequency band. Our main aim is for the arduino to detect a clap, and turn an LED on for one second. This kind of circuit is called a "clapper" circuit, since we activate the LED by clapping.

## Block Diagram/Circuit Diagram:




**Code:**

```
int ledPin=13;
int sensorPin=7;
boolean val =0;
void setup(){
  pinMode(ledPin, OUTPUT);
  pinMode(sensorPin, INPUT);
  Serial.begin (9600);
}
void loop (){
  val =digitalRead(sensorPin);
  Serial.println (val);
  // when the sensor detects a signal above the threshold value,
LED flashes
```

```
  if (val==HIGH) {
    digitalWrite(ledPin, HIGH);
  }
  else {
       digitalWrite(ledPin, LOW);
 }
}
```

**Output:**