DRISHTI

A Revolutionary Concept

PROJECT NAME: Hand Gesture Controlled LCD Display

MENTORS:

- Nancy Jikadra
- Vansh Khaitan
- Jaydev Rajaiya

TEAM MEMBERS:

- Vishwa Paragbhai Lathigara
- Rishita Jangid
- Raghav Lathi
- Arihant Hirawat
- Ashish Jha
- Krishna Patel
- Mudra Bhedi

INDEX

AIM

We are working on a Hand Gesture Controlled LCD Display project and our aim is to help disabled people by comprehending the meanings of various gestures and conveying their pov to the common mass.

It basically works using various python libraries, OPEN CV and MEDIAPIPE which detects user's hand through a total of 21 landmarks starting with 0 to 20.The final code with implemented conditions is thus uploaded to Arduino which sends the instructions to LCD Display and giving the result as per the conditions given.

PYTHON

- Python is a popular programming language.
- Python is used in various domains such as web development, data analysis, artificial intelligence, scientific computing, machine learning and more.
- Python is a high-level, versatile and widely used programming language known for its simplicity and readability.
- Created by Guido van Rossum and first released in 1991, Python has gained immense popularity among programmers due to its clear and concise syntax, extensive standard library, and active community support
- Python is an interpreted language, meaning code is executed line by line by the Python interpreter, without the need for compilation. This allows for rapid prototyping and debugging.

NUMPY Library

- NumPy is a Python library used for working with arrays
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

PANDAS Library

- Pandas is a popular Python library for data manipulation and analysis.
- It provides data structures like DataFrame and Series, enabling efficient handling of structured data. With powerful functions for reading, writing, and transforming data, pandas simplifies tasks like cleaning, filtering, and aggregating data.
- It integrates well with other libraries and tools, making it an essential tool for data scientists and analysts. By offering a wide array of functionalities, pandas has become a go-to choice for data wrangling, exploration, and preparation tasks in various domains, from finance and economics to machine learning and scientific research.

MATPLOTLIB Library

- Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
- Matplotlib is a solution for Python users who need to visualize their data to make essential statistical conclusions.
- It is a complete plotting package useful for Python and NumPy users. This article will assist you in comprehending the matplotlib library, which is frequently used in the industry.
- Matplotlib includes a wide range of graphical tools and is simple to use.

OPENCV Library

- OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc.
- It can process images and videos to identify objects, faces, or even the handwriting of a human.
- When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in our Arsenal i.e whatever operations one can do in Numpy can be combined with openCV.
- In opencv we have learned about:
    1. Reading an image
    2. Extracting the RGB values of a pixel
    3. Extracting the Region of Interest (ROI)
    4. Resizing the Image
    5. Rotating the Image
    6. Drawing a Rectangle
    7. Displaying text
- For drawing rectangle in open cv it takes five argument
    1. Image
    2. Top-left corner coordinates
    3. Bottom-right corner coordinates
    4. Color (in BGR format)
    5. Line width
- For drawing a text in opencv it takes 7 argument
    1. Image
    2. Text to be displayed
    3. Bottom-left corner coordinates, from where the text should start
    4. Font
    5. Font size
    6. Color (BGR format)
    7. Line width
- The steps to read and display an image in OpenCV are:
    1. Read an image using imread() function.
    2. Create a GUI window and display image using imshow() function.
    3. Use function waitkey(0) to hold the image window on the screen by the specified number of seconds, o means till the user closes it, it will hold GUI window on the screen.
    4. Delete image window from the memory after displaying using destroyAllWindows() function.

MEDIAPIPE Library

MediaPipe is an open-source framework developed by Google that focuses on building pipelines for perceptual computing tasks, including hand tracking. It provides pre-built components and pipelines for various tasks, saving developers the effort of building these pipelines from scratch.

For hand detection using MediaPipe:

- **Hand Landmarks Model**: MediaPipe provides a pre-trained hand landmarks model that can detect 21 keypoints (landmarks) on the hand.
- **Landmark Tracking:** The model tracks these landmarks in real-time as the hand moves, allowing you to obtain precise hand pose information.
- **Gesture Recognition**: With the landmarks' positions, you can recognize various hand gestures and movements, such as pinching, thumbs-up, and more.
- **Integration**: MediaPipe offers a well-documented API that makes it easy to integrate hand tracking into your projects. You can use the provided Python API to access the hand landmarks and build applications.

While OpenCV provides a more manual approach to hand detection involving image processing and contour analysis, MediaPipe simplifies the process by offering a pre-trained model and an API for real-time hand tracking. The choice between OpenCV and MediaPipe depends on the complexity of our project, our familiarity with the libraries, and whether you want a more hands-on approach or a quicker integration using a pre-built solution.

Steps to use media pipe for :

1. **Capture Video:** Initialize the webcam or camera using MediaPipe to capture a stream of video frames.
2. **Hand Detection and Landmarks:** From the detected landmarks, calculate distances between specific pairs of landmarks (e.g., distance between thumb and index finger tips) and angles between triples of landmarks (e.g., angle between thumb, index finger, and middle finger). These measurements capture the hand's pose and shape.
3. **Gesture Classification:** Set threshold values for distances and angles for each gesture's landmark configuration. These thresholds define the range of acceptable values for each gesture. Compare the calculated distances and angles with the predefined thresholds for each gesture. If they fall within the specified range, classify the current hand pose as that gesture.
4. **Gesture State Management:** Track the state of each recognized gesture over multiple frames to handle transitions and prevent rapid fluctuations in recognition. For instance, require a gesture to be consistently detected for a certain number of frames before triggering an action.
5. **Gesture Triggering and Actions:** Once a gesture is classified and its state is stable, trigger a corresponding action. For instance, display a message or initiate a command based on the

recognized gesture. To avoid repeated triggering, establish a cooldown period between successive activations of the same gesture.

6. **Continuous Monitoring:** Continuously update the hand landmarks and perform the gesture recognition process for each new video frame.

ARDUINO

Arduino is an open-source electronics platform that consists of both hardware and software components. It was designed to make it easy for individuals, particularly those without extensive technical backgrounds, to create and experiment with a wide range of electronic projects. Here's a brief overview:

- **Hardware:** The Arduino platform primarily revolves around microcontroller boards, which are small, programmable devices that can interact with the physical world through various sensors, actuators, and other components. These boards are equipped with input/output pins, digital and analog, which allow users to connect and control external devices.
- **Software**: Arduino provides an integrated development environment (IDE), which is a software application used to write, compile, and upload code to the Arduino boards. The programming language used is a simplified version of C++ with additional libraries that make it straightforward to interact with the hardware. Users can write code to control LEDs, motors, sensors, and more.
- Arduino's open-source nature means that its designs, schematics, and software are freely available to the public.In summary, Arduino is a user-friendly electronics platform.

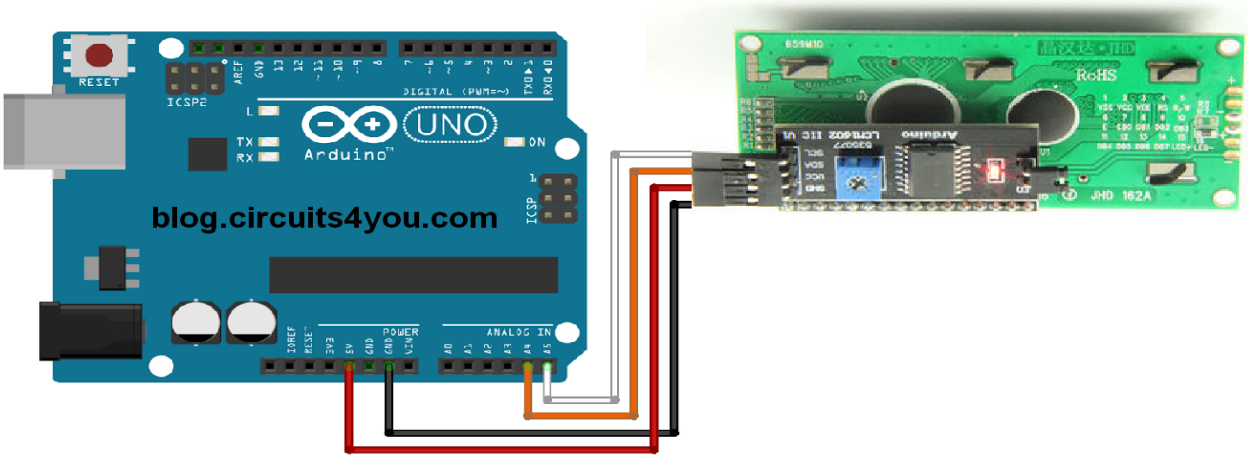

FIG: ARDUINO AND LCD DISPLAY CONNECTION

ARDUINO Pins

1. Digital Pins: Digital pins on an Arduino can be used for both input and output operations. They can read or write binary values (0 or 1). These pins are often used to interface with digital sensors, control LEDs, and communicate with other digital devices. They are labeled with 'D' followed by a number (e.g., D2, D7).

2. Analog Pins: Analog pins on an Arduino can be used to read analog voltage levels. These pins are mainly used for connecting analog sensors that provide continuous voltage outputs, such as light sensors or temperature sensors. The Arduino's analog pins are labeled with 'A' followed by a number (e.g., A0, A3). However, it's important to note that most Arduino boards have a limited number of true analog pins, and additional analog inputs may be achieved using techniques like analog-to-digital conversion (ADC).

3. Power Pins: Power pins provide the necessary voltage and ground connections to power the Arduino board and any connected components. The main power pins include:

- VCC (Voltage Common Collector): This is the supply voltage for the board, typically 5V or 3.3V, depending on the board model.
- GND (Ground): This is the common ground reference for the entire circuit.
- VIN: This pin allows you to supply an external voltage for powering the board when using an external power source.

LCD Display

- A 16x2 LCD (Liquid Crystal Display) is a commonly used alphanumeric display module that can display 16 characters in each of its two rows, making a total of 32 characters. It's widely used in various electronics projects, embedded systems, and devices for displaying information in a clear and easily readable format.

- 16 Characters by 2 Rows: The LCD display consists of 16 characters in each of its two rows, forming a 2x16 configuration.

- The 16x2 LCD display module usually interfaces with microcontrollers using a parallel connection of data lines and control signals. Common control signals include Register Select (RS), Read/Write (R/W), and Enable (EN).

- The data lines (often referred to as D0 to D7) are used to send character data to the display.

- When using an Arduino microcontroller, you can easily interface with a 16x2 LCD using libraries like the "LiquidCrystal" library. These libraries simplify the process of sending commands and data to the LCD.

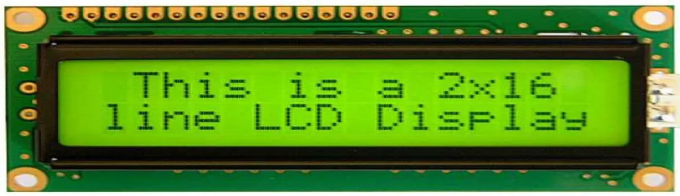

Fig: LCD DISPLAY

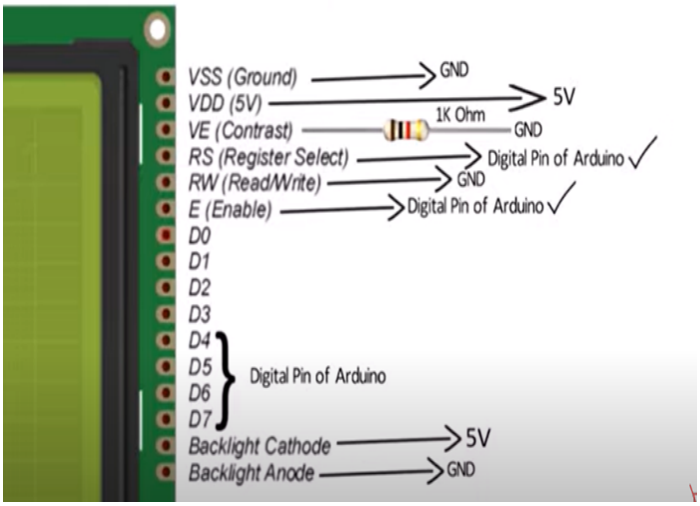

FIG: LCD PINS

I2C Module

Connecting an I2C module with an LCD and Arduino can simplify the wiring and reduce the number of required pins, as I2C (Inter-Integrated Circuit) communication uses only two wires (SDA and SCL) to communicate with multiple devices on the same bus.

1. I2C LCD Module Preparation -
   The I2C LCD module should have a small circuit board attached to the back of the LCD screen. This board contains an I2C expander chip that communicates with the Arduino.
   The module usually has a 16-pin header. Look for pins labeled SDA (Serial Data Line) and SCL (Serial Clock Line) on the module.

2. Wiring -
   Connect the SDA pin of the I2C module to the SDA pin of the Arduino.
   Connect the SCL pin of the I2C module to the SCL pin of the Arduino.
   Connect the VCC pin of the I2C module to the 5V pin of the Arduino.
   Connect the GND pin of the I2C module to the GND pin of the Arduino.

3. Library Installation -
   In the Arduino IDE, go to "Sketch" > "Include Library" > "Manage Libraries."
   Search for "LiquidCrystal I2C" and install the library.
   Include the library in your Arduino sketch.

Serial Library (Arduino IDE)-

The Serial library in Arduino provides functions and methods for establishing and managing serial communication between an Arduino board and external devices, such as a computer, another Arduino, sensors, or actuators. It allows you to exchange data, commands, and information .

Liquid Crystal I2C Library (Arduino IDE)-

The "LiquidCrystalI2C" library in the Arduino IDE is an extension of the standard LiquidCrystal library that allows you to control I2C (Inter-Integrated Circuit) based LCD displays with fewer pins compared to traditional parallel connection. I2C is a communication protocol that enables devices to communicate over a shared bus using only two wires: a clock line (SCL) and a data line (SDA).

The "LiquidCrystalI2C" library is especially useful when you have limited pins available on your Arduino board or when you want to simplify the wiring of your project. This library abstracts the low-level I2C communication and provides a simple interface to control the LCD display.

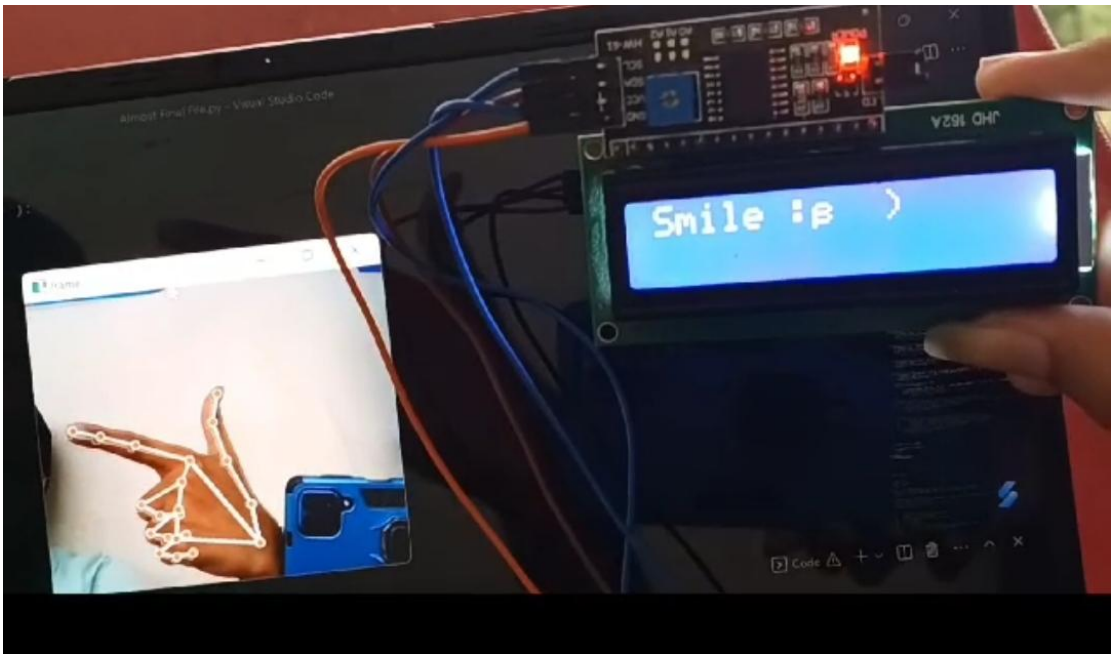

FIG: FINAL OUTPUT

Timeline

MEETS	DATE	TASKS	REFERENCE
Meet 1	9/7/23	1. First online phase common meet 2. Introduction	-
Meet 2	10/7/23	1. First AI ML combine online meet 2. Introduction to AI ML topics 3. Started learning basic concepts of python 4. Started learning numpy and pandas	1. numpy 2. pandas
Meet 3	12/7/23	1. Started learning matplotlib	1. matplotlib
Meet 4	15/7/23	1. Learned about jupyter 2. Learned about Kaggle 3. Started practicing above concepts on titanic database	1. jupyter notebook 2. kaggle
Meet 5	19/7/23	1. Online phase for project started 2. Started learning opencv	1. intro 2. Opencv
Meet 6	20/7/23	1. Progress in learning opencv and mediapipe 2. Different ideas regarding project were brought up	1. Opencv
Meet 7	22/7/23	1. Coding for hand detection started using mediapipe and opencv	1. Hand Tracking Project 2. Python Hand detection

Meet 8	25/7/23	1. Code analysis and debugging 2. Basic gesture detection coding started	1. Hand gesture recognition 2. Hand Sign Detection
Meet 9	25/7/23	1. Coding for Indian emergency sign language detection started	1. Hand Sign detection 2 2. Sign Language Detection
Meet 10	27/7/23	1. Coding for emergency hand signs completed	1. Real Time AI Sign language detection
Meet 11	27/7/23	1. Started learning about Arduino IDE, Arduino coding and LCD display	1. arduino IDE 2. arduino 3. LCD display
Meet 12	30/7/23	1. Installation of Arduino IDE 2. Started learning how to integrate python code with arduino	1. installation 2. python code
Meet 13	2/8/23	1. First offline meet 2. Interaction with team mates and mentors	-
Meet 14	5/8/23	1. Hardware implementation started 2. LED blink internal and external3. LED operation through keyboard letters	-

Broader Perspective

- We have implemented the code through various methods to detect different gestures using python coding.

- We have done the project according to the problem statement, but it can be further customized for various applications.

- Different methods of detection of gestures have different accuracy.

- We can also add audio for gestures.It can further be used in gaming applications and various gestures can be used to make objects move or control volume and brightness,etc.