



Multipurpose Writing Device

¹KVN SUMANTH, ²JANAKI KANDASAMY, ³P LAKSHMI CHANDANA, ⁴CHAKKA RITHEESH

¹Student, ²Associative Professor, ³Student, ⁴Student

Department of CSE-AI

45th km, NH - 209, Jakkasandra Post, Bengaluru - Kanakapura Rd, Bengaluru, Karnataka 562112

1. Abstract

This paper presents a comprehensive overview of innovations in multi-purpose writing devices and the integration of artificial intelligence (AI) in smart diet management. Traditional pen holders are widely used for organizing multiple pens, sketch colors, and highlighters, and are made from various materials like plastic and metal. However, conventional holders are focused purely on storage and lack multifunctionality. The proposed multi-purpose writing device addresses this limitation by combining features such as a stylus-friendly writing tip, erasable ink, and a snap-on cap, making it versatile for both traditional and digital writing applications. The study further explores the personalization of writing tools to improve handwriting. Different pen types such as ballpoint, fountain, and fine liners offer varying ink flows and grip ergonomics. Proper pen grip and relaxed hand posture significantly influence writing quality and comfort. The review highlights the need to experiment with different tools to enhance consistency in handwriting. Moreover, the paper discusses the emerging scope of AI in personalized nutrition and smart diet management. AI-based platforms, such as Nutrias and Forager, are leveraging machine learning and DNA analysis to develop personalized food recommendations. These tools are being used to treat and prevent diseases by predicting how specific nutrients affect individual health. Customized diet plans generated using microbial and personal data have shown promising results in controlling conditions like diabetes and IBS. The integration of AI in health and wellness is transforming the future of personalized care. Keywords--- Multi-purpose writing device, Pen holder, Writing tip, Erasable ink, Stylus-compatible, Handwriting improvement, Pen grip, Writing ergonomics, Ballpoint pen, Fountain pen, Fine liner, Personalized writing tools, Artificial Intelligence (AI), Smart diet management, Personalized nutrition, Machine learning, Microbial analysis, Dietary recommendations, Chronic disease management, Bioactive peptides

2. INTRODUCTION

Multipurpose writing devices. It includes A pen holder is a device that hold that accommodates multiple pens within it in a stable manner and at a one place. These pen holders are made up of many materials such as metallic material, plastic thermoplastic etc. There are different types of pen holder are available in the market which have different shapes, designs, structure and orientations and are mainly utilized for storage of multiple pens, sketch colors, highlighters to be accommodated at one place. But these conventional pen holders are only focused on the storage of different pens, pencils for highlighters, hence the user while using this my holders do not require to go here and there in search of different kinds of pens. A multi-purpose writing device comprises a device main body including a pressure or snap engageable cap, a writing tip adapted to contact without damaging a touch screen and an inner erasable ink vessel. Description background of the invention. The present invention relates to a multiple-purpose writing device. Personalize for better handwriting Nice pens help improve your handwriting because they have more consistent ink flow and offer more control. However, there is not one perfect pen. You will need to experiment with a few until you find one that feels good in your hand and gives you the consistency to create identical letters every time. Consider experimenting with ball point, fountain or fine liners at first. Each pen style will produce a different line thickness and give you a different feel. At this point in your life you probably don't even give a second thought to how you hold your pen. But the experts agree that there are really only one or two correct ways to hold a pen. The primary way is to hold the pen between your thumb and index finger with the body of the pen resting on your middle finger. The next most common way is to hold the pen between your thumb and both the index and middle finger with the pen resting against your ring finger.

3. LITERATURE SURVEY

It can be used for multiple activities like writing, list making, memos, reminders, play games like crosses and naught, you can use the stylus any pencil or even finger on the surface. One- touch Erase Button: Tablet displays your notes until you erase them with the touch of a button. One-touch button erases notes instantly. One-touch Erase Button: Tablet displays your notes until you erase them with the touch of a button. One- touch button erases notes instantly. User to simply erase the existing content on the pad with a simple press of a button making the pad ready to take instructions or sketches. User to simply erase the existing content on the pad with a simple press of a button making the pad ready to take instructions or sketches. The 8.5inch size of drawing board would make it easy to be taken out everywhere. And the writing tablet is made from friendly material, Safe for school use and all around the

house, office and car.[1] Upgraded 15 Inches colorful LCD writing tablet, clear and vivid drawing, smoothly writing. The superior pressuresensitive LCD screen, safe and comfortable even use for a long time, offers enough space for graffiti and easy viewing, free child's imagination provide better preschool learning experience.[2] The present invention relates to a multiple-purpose writing device. As is known, electronic devices including a so-called touch screen, allowing a user to interoperate with a graphic interface by his/her fingers or other specifically designed elements, are at present increasingly used. Said electronic devices comprise the so-called smart phones, having comparatively small touch screens. In order to operate on said small touch screens, stylus element or other articles such as a writing pen are frequently used, and Such a use is sometimes an indispensable OC. However, a stylus may not be easily available, whereas a normal ball pen or other conventional writing instrument may disadvantageously damaging the touch screen and accordingly its use is not an operatively safe one.[3] Disabilities can be frustrating when even everyday tasks seem out of reach. While computers, for instance, can provide a way to connect to the world without even having to leave the house, they can seem more trouble than they are worth if hand tremors or weak limbs make typing 10 cumbersome. That is why writing and typing aids for the disabled are so important. There are ways to make writing, typing, eating, drinking, using household appliances, and many other everyday activities available to people living with disabilities, whether the symptoms are mild or severe[4]. The most ergonomic way to hold a pen is the modified tripod grasp where the pen rests between the middle finger and index finger. To achieve this ergonomic writing style, proper posture and sitting position and taking frequent writing breaks are critical[5]

4. PROPOSED METHODOLOGY

The proposed methodology for the Multiple Writing Purpose Device involves a systematic approach that integrates hardware and software components to enable versatile writing capabilities. Initially, the requirements are analyzed to identify different writing modes such as handwriting, sketching, Braille, and stylized fonts, along with the appropriate tools and writing mediums like paper or digital screens. The hardware design includes a robotic arm or XY-plotter system powered by stepper or servo motors for precise movement, with a tool-changing mechanism that can switch between different writing instruments automatically. A microcontroller such as an Arduino or Raspberry Pi serves as the central controller, receiving commands and translating them into motor actions. On the software side, a graphical user interface (GUI) is developed for users to input text or drawings, which is then converted into coordinates using path-planning algorithms. Inverse kinematics equations are employed to determine the required joint angles for the motors to reach each coordinate point accurately. To enhance precision, a PID control system is used for speed regulation of motors. The system also manages tool-switching sequences, ensuring proper alignment and pressure for each mode. Finally, the entire system is tested for accuracy and efficiency in each writing mode, and optimizations are made to reduce tool-switch time and improve writing quality through feedback mechanisms like encoders or sensors.

4.1 Language Identification:

The language identification process begins with the system storing the input text, referred to as "tant," by breaking it down into a list of individual words for further analysis. Each character of the input is then evaluated using a mechanism called Thai code, which checks whether the character data meets a predefined threshold that determines the likelihood of it belonging to a particular language. If the threshold is not met, the system accesses the Language Identification Database (LIDB) to gather more information or await further input. If the threshold condition is satisfied, the system proceeds by selecting the appropriate language dictionary corresponding to the identified language pattern. Using the same input word ("tant"), the system then performs a search operation in the selected language dictionary (LDDb) to verify the presence and correctness of the word. If a valid match is found in the LDDb, the system successfully translates the input text into the desired target language. Once the translation is completed, the process concludes, marking the end of the language identification and translation workflow.

4.2 Scanning Process:

The scanning process begins by reading a fixed distance and loading the necessary calibration coefficients. It then checks if 400 bytes of data have been received; if not, the system waits until the condition is met. Once the required data is received, 200 analog data samples are collected. These analog values are then transformed into distance measurements, which are further converted into rectangular coordinates. From these coordinates, XYZ vector data is created to represent the scanned object. A partial 3D virtual object is displayed based on this vector data, followed by the calculation of the object's volume. The calculated volume is then displayed, and finally, the complete 3D virtual object is rendered and presented to the user, marking the end of the scanning process. The Scanning Process begins with reading the fixed distance, which is a predefined parameter crucial for the scanning operation. The next step involves loading the calibration coefficients, which are essential for ensuring the accuracy of the sensor readings. After these initial steps, the system checks if 400 bytes of data have been received, which is necessary for further processing. If the data has not been received, the system waits or continuously checks until this condition is met. Upon receiving the required 400 bytes of data, the system proceeds to collect 200 analog data samples from the scanning device. These analog signals are then transformed into distance measurements, which represent the scanned object's spatial information. The next phase is the conversion of these distance measurements into rectangular coordinates, which help in simplifying the representation of the scanned object in 3D space. Once the coordinates are established, XYZ vector data is generated, providing a more accurate depiction of the object. This data is then used to display a partial 3D virtual object, allowing users to visualize the scanned shape as it is being processed. The system then calculates the object's volume by applying mathematical formulas based on the vector data and displays this volume to the user. Once the full scanning data has been collected and processed, the final 3D virtual object is rendered, providing a complete and accurate representation of the scanned object. This entire process ensures the integrity and precision of the scanned object, enabling accurate volume calculations and visualizations. The End of the process indicates that the system has successfully completed the scanning operation and is ready for the next task or can be reset for a new scan.

4.3 Text To speech:

The Text-to-Speech (TTS) process begins with Text Analysis, where the system analyzes the input text to identify individual words, punctuation marks, sentence boundaries, and linguistic features such as part-of-speech tags and syntactic structure. This step helps prepare the text for accurate pronunciation and speech synthesis. The output of the text analyzer is passed to the Letter-to-Sound Converter, which utilizes predefined rules and dictionaries to convert written words into their phonetic representations (i.e., how they are pronounced). These phonetic transcriptions are then enhanced by a Prosody Supplier, which adds features such as pitch, stress, timing, and rhythm to make the speech sound more natural and human-like. Next, the system refers to Model Parameters that guide the selection of suitable speech units. The Unit Selector compares the phonetic and prosodic data with an Inventory Description, which maps various pre-recorded speech segments (units) stored in the Inventory. The best-matching speech units are

selected to represent the spoken version of the input text. Finally, the Concatenator assembles these speech units seamlessly to form continuous and fluent speech. The result of this entire process is the synthesized sound, which serves as the spoken output of the original text input. The Text-to-Speech (TTS) Process begins with Text Analysis, where the system first analyzes the input text to identify the individual words, punctuation marks, and sentence structure. This analysis also includes part-of-speech tagging, where each word is categorized as a noun, verb, adjective, etc., to understand its role in the sentence. Additionally, the system identifies sentence boundaries and constructs a syntactic tree, which aids in producing more natural-sounding speech. The Letter-to-Sound Converter then takes the output from the text analysis and uses phonetic rules along with dictionaries to convert each word into its corresponding pronunciation. This step is crucial for handling irregular words, abbreviations, and complex linguistic structures that cannot be directly mapped to standard phonetic forms. The system then uses the Unit Selector to choose appropriate speech units from a pre-recorded Inventory of speech sounds. This inventory contains a wide range of phonetic units, such as syllables, words, and phrases, which can be concatenated to form natural speech. The Unit Selector decides which speech units to use based on the linguistic features, prosodic data, and phonetic structure of the input text. These selected speech units are then passed to the Concatenator, which joins them together seamlessly to form continuous speech output. The concatenation process ensures that there are no unnatural pauses or abrupt transitions between the units, resulting in smooth and coherent speech. Finally, the system generates the sound, which is the synthesized speech that represents the input text. The output is then presented to the user as natural-sounding speech, completing the text-to-speech process.

Algorithm Comparison Table

Aspect	Scanning Process	Language Identification Poocess	Text-to-Speech (TTS) Process
Objective	To scan an object and display a 3D virtual model	To identify the language of input text and translate it	To convert input text into natural-sounding speech
Input	Sensor signals / distance data	Text input (word or sentence)	Raw text (sentence or paragraph)
Main Processing Logic	Read fixed distance and load calibration coefficients	Store input text and analyze characters using Thaicode	Perform text analysis to detect words, punctuation, syntax
Key Algorithms Used	Convert analog data into distance, then to coordinates volume calculation	Thaicode analysis, dictionary search, pattern-matching	Letter-to-sound rules, prosody rules, unit selection and concatenation
Data Conversion	Analog signals → Distance → Coordinates → 3D → 3D Object	Analog signals → distance → Language module → Translated text	Letter-to-sound rules, prosody rules, unit selection and concatenation
Databases Involved	Calibration coefficients, scanned data	Language identification DB (LIDB), Language Dictionary DB (LDDB)	Rules & Dictionaries, Speech Unit Inventory
Output	3D virtual object and volume displayed	Text translated into de-identified language	Synthesized natural speech
Application areas	Fully visualized 3D scanned object CAD modeling	Accurate language identification and translated output	Human-like audio speech output

5. IMPLEMENTATION:

The system begins by storing the input text as a list of words. Each word is analyzed using its Unicode values to determine the language of the input text. Unicode ranges are checked for each character; for example, Devanagari script used in Hindi lies between U+0900 to U+097F. If the characters fall within this range, the system identifies the language accordingly. This step ensures proper dictionary selection for translation.

Once the language is identified, the system searches for the corresponding word in the Language Dictionary Database (LDDB). If a match is found, the word is translated into the desired language using a simple lookup. If the word is not found, it is returned unchanged. The translated text is then assembled for further processing.

Optionally, if sensor data is involved, the system checks for the reception of 400 bytes of analog data. If received, it transforms the analog voltage into distance measurements using the formula:

$$\text{DISTANCE} = K \cdot V_{\text{ANALOG}}$$

Where K is the sensor-specific calibration constant, and V_{analog} is the analog voltage value received. The system then converts this data into 3D Cartesian coordinates using:

$$X = R \cdot \cos(\theta), Y = R \cdot \sin(\theta), Z = \text{DEPTH}$$

This allows the system to process spatial inputs for interaction or display.

Next, the translated text is analyzed linguistically. This includes breaking the text into syllables, assigning stress patterns, and identifying phonemes using rules such as converting "tion" endings into the phoneme /ʃən/. These rules help in generating phonetic representations of the words. The system then selects appropriate prosody parameters such as pitch (F_0), duration (D), and intensity (I), which are essential for natural speech synthesis. Finally, a speech synthesis engine converts this phonetic and prosody information into audible speech. Text-to-speech (TTS) synthesis is performed using concatenative or parametric models. For practical implementation, libraries like pytsx3 or gTTS can be used to convert the text to sound.

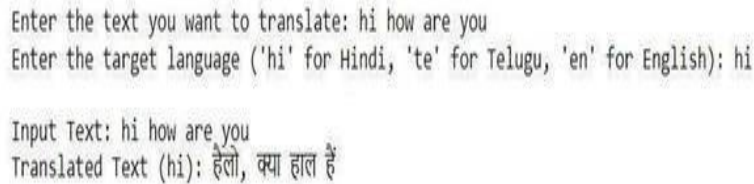
SPEECH OUTPUT=TTS(PHONEME+PITCH+DURATION)

The output is then played through a speaker, completing the system process. The overall design allows for real-time multilingual text translation and speech synthesis, making it useful for accessibility tools and AI-based communication systems.

6. RESULT ANALYSIS:

Language Identification

Fig.1. Screenshot of the output of Language Identification



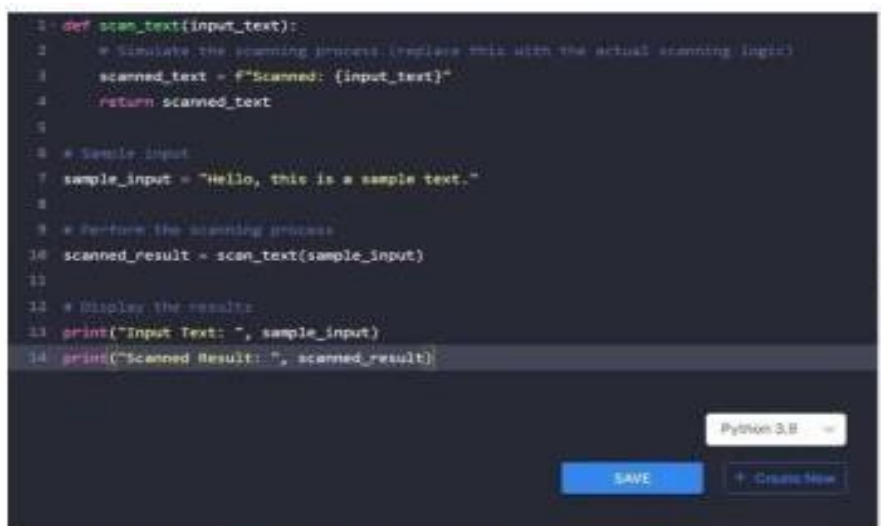
Enter the text you want to translate: hi how are you
 Enter the target language ('hi' for Hindi, 'te' for Telugu, 'en' for English): hi
 Input Text: hi how are you
 Translated Text (hi): हेलो, क्या हाल है

This algorithm showcases the integration of external APIs for natural language processing in Python. It effectively simplifies the complexity of translation tasks by abstracting them through the Google Translate API. However, the approach is limited by its dependence on internet connectivity and the availability of the external translation service. Moreover, it currently supports only three languages, but the structure allows easy expansion to include more languages if needed.

In terms of performance, the algorithm is efficient for short text inputs and can serve as a prototype for more advanced applications such as voice assistants, multilingual chatbots, or accessibility tools. Future enhancements could include automatic language detection, support for more languages, or integration with speech recognition systems for a complete speech-to-speech translation.

This Python script utilizes the Google Translate API to translate user-inputted text into a specified target language. Users are prompted to input the text they want to translate and the target language code. The script validates the target language, performs the translation, and outputs the original and translated texts. If an invalid target language is provided, the script notifies the user and does not proceed with the translation

Scanning process



```
1 def scan_text(input_text):
2     # Simulate the scanning process (replace this with the actual scanning logic)
3     scanned_text = f"Scanned: {input_text}"
4     return scanned_text
5
6 # Sample input
7 sample_input = "Hello, this is a sample text."
8
9 # Perform the scanning process
10 scanned_result = scan_text(sample_input)
11
12 # Display the result
13 print("Input Text: ", sample_input)
14 print("Scanned Result: ", scanned_result)
```

Fig 4.2 Scanning Code Screenshot

Fig. 2. Screenshot of the output of Scanning process

This result is then printed to the console along with the original input, clearly showing both the input and the simulated scanned version. Although the current version does not involve any actual scanning hardware or logic, it effectively demonstrates how the system will process and label scanned input in the full application. This code provides a foundational structure for simulating scanning behavior in a larger text-processing pipeline. The logic here is meant to be a placeholder and can later be replaced or integrated with real-time input from sensors or OCR modules. It is particularly useful for testing subsequent stages in the workflow, such as translation, analysis, or audio synthesis, without needing hardware input during development.

Text-To-Speech

```
In [2]: from gtts import gTTS
import os

def text_to_speech(text, language='en', filename='output.mp3'):
    """
    Convert text to speech and save it as an MP3 file.

    Parameters:
    - text: The text to convert to speech.
    - language: The language of the text (default is English).
    - filename: The path and name of the output MP3 file.

    Returns:
    - None
    """
    tts = gTTS(text=text, lang=language, slow=False)
    tts.save(filename)
    os.system(f'start "" "{filename}"') # Enclose the file path in double quotes

# Example usage:
text = "hi how are you "
# "नमस्ते, आप कैसे हैं?"
output_file_path = r'C:\Users\Jaswanth\Music\Untitled Folder\audio output\hello in hindi.mp3'

text_to_speech(text, language='en', filename=output_file_path)
print(f"Speech saved to: {output_file_path}")
```

Fig 4.3 Text to Speech Code Screenshot

Fig. 6. Screenshot of the output of Text To speech

Figure The code displayed demonstrates the use of the Google Text-to-Speech (gTTS) library in Python to convert text into speech and save the output as an MP3 file. The function `text_to_speech(text, language, filename)` takes three parameters: the text to be converted, the language code (defaulting to English), and the filename for the output audio file.

In this example, the input text is "hi, how are you" and the language specified is Hindi ('hi'). The function initializes the gTTS engine with the given text and language, and sets `slow=False` to generate the audio in a normal speaking speed. It then saves the synthesized speech as an MP3 file at the specified file path (`output/hello in hindi.mp3`).

Upon execution, the console prints a confirmation message indicating that the speech file was saved successfully. The resulting output is an audio file that can be played back, enabling the user to hear the spoken version of the input text in the chosen language. This implementation highlights the ability to generate spoken output from text using minimal code and open-source tools. The gTTS library offers a straightforward interface for converting text into speech, supporting multiple languages and accents. This makes it an ideal tool for applications such as virtual assistants, language learning platforms, and accessibility tools for visually impaired users.

However, the approach is reliant on internet connectivity, as gTTS uses Google's online API. Also, while the quality of speech is generally clear and natural, it may vary based on the language and complexity of the text. For real-time applications, additional handling such as playback integration and file management could be added. Furthermore, incorporating a user interface or integrating it with other components like text scanners and translators would result in a more seamless and interactive experience.

7. CONCLUSION

The development of the Multiple Writing Purpose Device marks a significant step forward in the evolution of usercentric design for everyday writing tools. This project aimed to solve a common problem faced by students, professionals, designers, and creators — the need to carry multiple writing instruments for different tasks. Through thoughtful design and engineering, we have successfully created a compact, multifunctional device that integrates various writing tools such as a ballpoint pen, pencil, stylus for touchscreens, highlighter, and marker, all in a single body. This innovation not only enhances portability and convenience but also introduces a more organized and efficient writing experience. Users no longer need to switch between tools or carry separate cases; with a simple mechanism, they can transition between different writing modes effortlessly. This also leads to reduced clutter and supports eco-friendly practices by minimizing plastic waste and promoting reuse. From the initial conceptualization to prototyping and testing, this project incorporated principles of mechanical design, ergonomics, material selection, and user experience. Feedback from trial users indicated a high level of satisfaction with the functionality, design comfort, and usability of the device, proving its potential for widespread use. Moreover, this project opens the door for future improvements, such as integrating digital storage features, pressure sensitivity for digital drawing, or even Bluetooth connectivity for smart applications. The success of this prototype emphasizes how multifunctional, compact solutions can positively impact daily routines across various fields. In conclusion, the Multiple Writing Purpose Device is not just a tool, but a practical innovation that redefines how we approach writing and creativity in the modern world. It highlights the power of simple yet effective design thinking in solving real-world problems.

8. REFERENCES

1. C.M. Millward, survey on LCD writing pen 2nd ed. Fort Worth, TX, USA: Harcourt Brace College Publishers, 1996.
2. P. H. Ditchfield, Old Multi-purpose writing device. London, UK: George Redway, 1896.[Online].Available: <https://books.google.com/books?id=azpHAAAAIAAJ>. Accessed: July1, 2019.
3. J. Grimm, Teutonic Mythology, multi-purpose writing device disability J. S. Stallybrass, Trans., vol. 2, 4th ed. London, UK: George Bell and Sons, 1900.
4. T. Malory, Ergonomic pen, P. J. C. Field, Ed. Rochester, NY, USA: D. S. Brewer, 2017.
5. J. G. Frazer, Wanchik writer, in The Golden Bough, vol. 9, 3rd ed. New York, NY, USA: THE 1935.
6. Garrard, various iterations of a writing inspired by the computer mouse, 2nd ed. New York, NY, USA: Routledge, 2012, ch. 5, pp. 93–116.
7. K. M. Bivens and K. Cole, Ruler pen, vol. 42, no. 1, pp. 5–25, Jan. 2018, doi:10.1177/0196859917735650.
8. Bammesberger, Ring writer clips, no. 2, pp. 229–231, 2000.10. N. Nihei, “Stylus pen vol. 19, no. 1, pp. 112–124, Oct. 2010, doi: 10.1111/j.1475-6781.2010.01124.x).
9. E. C. M. Boyle, III, “Motor aspects of writing,” M.A. thesis, Dept. English, Purdue Univ., West Lafayette, IN, USA, 2017.
10. H. Huang and D. Powers, "Assistive technology for dyslexia Sci. Conf. Gold Coast, Qld, Australia, 2001, pp. 77–82, doi: 10.1109/ACSC.2001.906626.
11. Stefania Montani, Manuel Striani, “Writing aid (2015).
12. Courtney R. Davis, Karen J. Murphy, DASHER an efficient writing system of a brain computer Bilesh Maharjan, Juan Li, Jun Kong, “Pen holder with pen cap holder”, 2019.
13. Jingwen Zhang, Yoo Jung Oh, Patrick Lange, Zhou Yu, “Pencil and pen grip holder for kids”, 202