Speaking at the Speed of Sight

Morse-Code Powered Blink Communication

Around the world, many suffer from Locked-in syndrome: a rare neurological disorder where a person is completely aware but cannot move most of their muscles. They can, however, use the muscles that control blinking. I have strived to create an app that can sense eye blinks to allow communication. My app allows someone to blink in Morse Code, interpreting the blinks using a simple AI model, which allows the user to type with their eyes.

For anyone with Locked-in syndrome or similar conditions, communication is a problem. Without speaking as an option, or even something like writing, what could work? Oftentimes, others don't even realize that people with conditions like this are conscious! Many have created solutions, and some work well. However, they are often very expensive and require elaborate hardware. It really depends on the individual. Can I create a simple, easy, and most of all, FREE solution for this?

As I was thinking of ways to solve this problem, the first thought that entered my mind was to create an app. I have always loved coding, and this seemed like a great project. As I looked into it more, I found that something similar had been done. If you look at blinkcommunicate.com, you can see that others have had similar ideas. However, I hope to create something innovative and useful. I want my version of this fix to be free and easy to access, be easy to use without a learning curve, and to function well without difficulty.

After I had decided my goals, I began to plan the app. I sketched out the user interface in my notebook, figuring out how I wanted it to work. Then, I dived into research to figure out which coding language to learn. I settled on Dart, a programming language developed by Google. The best way to use Dart is to program with Flutter-an open-source UI software development kit. With Flutter, you can create apps for mobile, web, and desktop, as it is very versatile.

Then, I had to decide how I would teach myself this language. I only had experience with Python, Javascript, and HTML- this was a whole new ball game. I decided on a simple solution: AI. I would <u>not</u> use AI to write the app. However, I would use it to teach me. Google gemini, a free AI model, is easy to customize: you can create "Gems" with special instructions. I created one with these base instructions: "You are a programmer with extensive knowledge. You are amazing at guiding through projects but never programming for them for whomever

you are assisting. You strictly teach new concepts and suggest paths for creation, but you never write code outside of demonstrations for learning. You are excellent at answering questions and guiding others through their projects. You stay away from generating code for whomever you are assisting, instead guiding them through it, so they can write it themselves."

As is clear, I wanted to create this app myself. Whenever I stumbled across a feature I didn't know how to use, I would ask the AI to teach me. That, along with many internet forums and the Flutter and Dart docs, allowed me to slowly learn enough to program what I needed.

Before I could really start programming, I needed a device that I could test my code on. Due to the fact that you need a MacBook to run code on iphones, I needed to install an android emulator. I spent quite a while figuring out how to install one successfully on my very old computer. Finally, I was ready.

Programming itself is a simple loop of writing, testing, and debugging. I must have gone through hundreds of iterations! Every time I added a new feature, I would boot up the emulator and test it. If it didn't work, which happened most of the time, I would try and figure out what was wrong, then fix it. If that didn't work, I'd do it all over again.

Finally, I was finished. Only one step was left: testing. I chose five words, then started the program. Sitting in a well-it room, directly in front of the screen, I blinked out the sequences, noting how many letters were missed. The total accuracy was approximately 87%. My five words were HI, SOS, YES, HELP, and FLUTTER. The longer the word, the less accurate the translation, due to both human and AI error. I found that if you are in a well-lit room–with your face in the center of the screen–it can interpret your blinks with high accuracy; only occasionally missing a blink.

After hours and hours of programming, searching the internet for explanations, testing, and troubleshooting, I have a working prototype. The app accesses your camera to take a video of your face. It sends each frame to a simple AI model that analyzes it. The AI model gives us data about your face. The program then interprets that data, deciding how long your eyes were closed. That is converted to long and short blinks, which are then interpreted using morse code to form letters and words.

The UI is quite nice looking, and I am very happy with it. I managed to program a home screen that shows the video it's taking, the letters and phrases you've blinked, and buttons to clear the screen. On the menu bar, you can go to the customization menu, which allows you to add dot-dash sequences of your own and completely decide what sequence

means what.

This menu, actually, is a very important part of my app. Without it, you'd have to memorize morse code and be stuck with that. That wouldn't make the app too easy to use for the average person. So, of course, I knew I had to add the feature to customize what sequences mean what. On the menu, you have a list of the existing signs. You can delete any of them. Then, there's a button to add a sign. You simply write the sequences of dots and dashes, then write what they mean.

My goal with this entire project was to provide an easily accessible app that can help people communicate through blinking. I can confidently say I have created a functioning app. As I tested it, the results showed that the AI could distinguish between a 'dot' and a 'dash' with high accuracy. While I haven't made it possible to download for anyone yet, that is plausible in the future, allowing for a solution other than expensive medical tech. While this is probably not something someone might want to use to communicate permanently, it can be used in tight spots when other options simply won't work. All in all, I accomplished what I wanted to, even with many limitations.

I had a myriad of limitations as I did this project. For a significant portion of my research, I didn't even have the basic materials to begin until I was fortunately gifted a laptop. Once I finally had the hardware, I faced a steep uphill climb: I had absolutely no prior experience with the Flutter language, and I was navigating this complex technical world entirely on my own, as no one in my family has a background in programming. These barriers, combined with a very limited window of time, meant that every breakthrough required double the effort and a lot of independent troubleshooting.

It took quite an amount of troubleshooting to set the emulator up, which took away from my time to program. This, along with silly little errors, took hours to debug without help. While these limitations meant I couldn't implement every 'extra' feature I dreamed of, they forced me to be resourceful and truly master every single line of code that made this prototype possible.

With more time, I plan to add Text-to-Speech to give users an audible voice and predictive text to reduce the physical fatigue of blinking. I also aim to optimize the AI for better tracking in low-light environments, which it currently struggles in.

In my opinion, while this app is mostly done, the research and engineering needed for this topic is nowhere near finished. Beyond my software, other vital areas of research for this problem include high-end Eye-Tracking hardware and Brain-Computer Interfaces (BCI).

While these are currently expensive or invasive, I hope we can one day bridge the gap between high-cost medical technology and free, accessible mobile tools for everyone.

One day, I am sure, we will. For now, though, we should just keep dreaming and doing! Everyone deserves the right to communicate. We should strive to give that to them. We should continue this important work. It matters so very much.