Liberate:ATCS - Adaptive Typing & Control System

https://liberate-atcs.vercel.app/

- ***By Team Lumen - Aaromal A , Deeraj P Menon***

Liberate:ATCS muscle-twitch-based control system that translates subtle muscle movements into control signals through an adaptive typing interface enabling hands-free assistive control & accessibility for people with limited mobility.

1. Problem Statement

Individuals with severe physical impairments—such as ALS, muscular dystrophy, or paralysis—often lose the ability to communicate effectively or control mobility devices due to limited voluntary movement. Existing assistive technologies are often slow, intrusive, or require residual motor control that these users may not possess and expensive. There is a critical need for a non-invasive, intuitive and affordable system that enables these individuals to regain autonomy in communication and mobility using the limited muscle activity they can still control.

2. Proposed Solution

We propose **Liberate:ATCS**, an adaptive control system that detects subtle muscle twitches and translates them into precise control signals through a typing interface for assistive applications. Using a non-invasive sensor interface, the system enables hands-free operation of devices such as communication interfaces and electric wheelchairs. By interpreting muscle activity through signal processing and intelligent control algorithms, Liberate:ATCS empowers individuals with limited mobility to communicate and navigate their environment independently, restoring both accessibility and dignity.

3. Technical Overview

The project is divided into hardware and software, we aim to minimize as much unnecessary hardware for portability & accessibility for people with limited mobility whereas software is made more minimal/simple providing a intuitive interface users are already familiar with

a. Project Architecture & Working

- **Signal Acquisition and Processing (ADXL345, ESP32 Microcontroller)** :In our prototypic version we currently use a ADXL345 digital accelerometer that records muscle twitches as 3-axis accelerations, which are encoded into precise twitches ie the control signals (as binary 1 for twitch) taking the net value and then filtering out unnecessary accelerations by baseline calibration (ie using a twitch & rest baseline values) and filtering algorithms. We can scale it to use different error prone sensors at the same time like EMG and IR reflexive sensors according to user needs/situation and affordability. An ESP32 32 board is used to process the signals received from the ADXL345 which offers both serial and wireless communication for further steps. It can still minimize it by using a small and compact microcontroller.

- **Data Transmission (Serial/Wireless such as WiFi):** Data/Control signals processed in the microcontroller can be sent serially/wirelessly from ESP32 to the Adaptive typing interface via WiFi, which is used to interact with the interface.
- **Adaptive Keyboard Interface (Python PyQT5 as desktop interface):** A Python based PyQt5 interface (desktop interface) provides a moving highlight bar which spans across the rows of a adaptive keyboard interface (incl QWERTY/MORSE/Command mode, AI suggestions, SOS key, Mobility arrow keys, Speak Keys for easy accessibility), allowing selection of rows and keys through subsequent muscle twitches ie control signals received from the microcontroller. We can scale the desktop interface into a tab display screen which can be attached near to the user or as an application user could install in his/her system/other devices with screens and enhance the interface by implementing AI on interface that learns and adapts to the user's muscle signal patterns, facilitating efficient - error prone & fast - hands-free communication.

b. Product Design

- **Non-Invasive Wearable Design**: The system can be designed as a wearable device including the microcontroller & necessary sensors like ADXL345 with a compact battery in it, ensuring user comfort and ease of use without surgical interventions.
- **Display - Adaptive Typing Interface**: Features an interface that learns and adapts to the user's muscle signal patterns implemented on an external tab screen or user's system as an application he/she can install, facilitating efficient hands-free communication.

4. Current Status

This project is built as a part of an 8hr hardware hackathon **- Buildathon:Hertz25'** held at CUSAT, India. We were able to build a **prototype version** of the idea/product we intend to implement. Which includes the **ADXL345 accelerometer** which can be attached to the user's desirable/mobile muscle group to sense its twitches, connected with a **ESP32** through a push button which process and filter the accelerations to the 80% accurate twitch control signals using filtering algorithms, send serially as encoding 1 to a twitch to the python based PyQt5 interface in one of our system. The processing signal logic we developed, put together as a **practical efficient algorithm to encode accelerations to twitch control signals (1s)** utilising filter algorithms such as debouncing filters and baseline calibration, is uploaded to the arduino first time, which records & stores the threshold/baseline values of twitch/rest from the initial readings in the flash rom of it permanently which can be reset later ie used to filter unnecessary twitches by baseline comparison during subsequent runs. The python based **PyQt5interface provides QWERTY mode for now including A-Z keys, Speak key (using a offline text to speech engine pyttsx3) & a SOS key for emergency, and a highlight bar to navigate through rows and keys using twitch control signals.**

5. Potential Impact

1. Real-World Applications

a. Social Impact

- Empowers people with severe physical disabilities (e.g., ALS, spinal injuries, muscular dystrophy) to communicate, navigate devices, and control their environment hands-free! contributing to the future of assistive tech.
- Acts as a cost-effective, open-source, intuitive & more controlled wearable alternative to commercial systems like Intel ACAT, EyeGaze, or Tobii, which are often expensive, region-locked or for a specific control.
- A new level of handfree human-computer interaction, Not being limited to the differently abled, imagine a world where people could interact with a computational device with just his muscles or even thoughts, we provide a minimal stepping stone for that future of **hands-free computation**.

b. Industrial Use-Case

- Can be extended to **low-effort gesture control systems** for environments where hands-free operation is required (e.g., sterile medical labs, hazardous manufacturing).
- Variants could be deployed for **robotic teleoperation** using minimal physical effort.

c. Commercial Potential

- Open-source foundation could lead to a modular assistive tech platform that allows companies or NGOs to build custom accessibility tools.
- Could evolve into a **wearable product line** for input control using subtle muscle movements (e.g., gesture keyboards, alternative controllers).

2. Academic Learning & Skill Building

a. Interdisciplinary Integration: Combines electronics, signal processing, machine learning, human-computer interaction, and assistive technology design.

b. Research Opportunities: Opens doors for research in neuromuscular interface design, adaptive user interfaces, and intelligent accessibility systems. Can be a base for academic papers, capstone projects, or collaborative research with medical/rehabilitation institutions.

c. Personal skill-growth: Hands-on mastery of tools like Arduino,, PyQt, real-time plotting, serial communication, and calibration algorithms.

3. Future Improvements & Scalability

a. Software Improvements

- Allow user-defined modes (typing, smart home control, web browsing).
- Make it mobile-compatible (e.g., Android app version via Bluetooth communication).
- Machine learning models can improve accuracy of signal classification (adaptive to each user's physiology).
- Replace current thresholding methods with dynamic pattern recognition or RNNs/CNNs.
- Add voice synthesis, eye-tracking integration, or cloud backup for personalized profiles.

b. Hardware Improvements

- Port system to wearable devices using ESP32, RP2040, or custom PCBs.
- Add support for other bio-signals: EMG, EEG, or optical muscle sensors.

For more project overview, visit our project portfolio: https://liberate-atcs.vercel.app/

contact for any inquiries: aaromalonline@gmail.com