Hand Gesture-Based PC Control System Using Raspberry Pi for Human-Computer Interaction

Bhavesh P
3rd year, Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore-641112

Y Charan Ganesh
3rd year, Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore-641112

K Harshavardhan Reddy
3rd year, Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore-641112

Ch Madhav Dheeraj
3rd year, Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore-641112

G Veda Sai
3rd year, Computer Science and Engineering
Amrita School of Computing, Amrita Vishwa Vidyapeetham
Ettimadai, Coimbatore-641112

***Abstract*—In the realm of Human-Computer Interaction (HCI), this project introduces an innovative approach to control a personal computer (PC) using hand gestures, facilitated by a Raspberry Pi-based embedded system. The proposed system harnesses the power of gesture recognition technology to enable users to interact with their PCs intuitively, eliminating the need for traditional input devices such as keyboards or mice.**

I. Introduction

The system's architecture integrates a Raspberry Pi with Flex-sensors, enabling real-time processing of the live gesture feed to recognize and interpret flex sensor-driven hand gestures. By mapping these gestures to specific commands, the system achieves seamless and hands-free PC control, enhancing user experience and accessibility.

II. Ease of Use

The design prioritizes accessibility, cost-effectiveness, and ease of implementation, ensuring adaptability for a diverse user base. The scalability of the embedded system allows for future integration with a spectrum of applications, ranging from gaming to productivity tools. Emphasizing open-source principles, the project encourages collaboration and community-driven enhancements to propel the evolution of gesture-based HCI.

System evaluation encompasses rigorous testing under varying environmental conditions and user scenarios. Key metrics, including gesture recognition accuracy, latency, and overall user satisfaction, quantify the system's reliability and effectiveness. Results affirm the feasibility of the Flex Sensor-based PC control system, offering a practical and user-friendly alternative within the HCI domain.

Identify applicable funding agency here. If none, delete this.

III. Primary Objectives

- Setup the PICO Hardware, Micro-Python
- Loading the 'Board.py' into the PICO board.
- Connect the Flex Sensors to the ADC ports.
- Capture the sensor reading and return the corresponding string input to the PC.
- Interpret the string input obtained from sensor and activate the corresponding action.
- Deploy the optimized model on Raspberry Pi Pico for real-time gesture detection.
- Evaluate performance.

IV. Hardware Requirements

- Raspberry Pi Pico H
- Flex Sensors
- Breadboard
- Male to Female Jumper Wires
- Micro USB Cable
- UART cable
- Resistors(10 k-ohm)
- ADC board
- Glove

V. Software Requirements

- Visual Studio
- Pico-sdk
- Micro-python

VI. Methodology for sensor interfacing

A. Hardware Connections

To integrate the Flex-Sensors with the Raspberry Pi Pico, establish connections as per the GPIO Pins specified in Table 1.

Module	Raspberry Pi Pico GPIO Pin
VCC	Pin 36
GND	Pin 3
UART	Pin 1-2
SDA	Pin 19
SCL	Pin 20
ADC0	Pin 31

TABLE I
GPIO Connections

Module	ADC Board Pin
ADC	Pin A0,A1,A2,A3
VDD(2-5.5V)	VDD
Ground	GND
I2C SDA	SDA
I2C SCL	SCL

TABLE II
ADC Connections

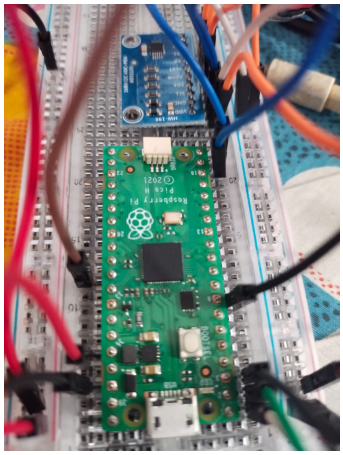

Fig. 1. Raspberry Pin Diagram

VII. Procedure for board Configuration

Interfacing a flex sensor to a Raspberry Pi Pico for hand gesture detection involves several steps, including connecting the flex sensor to the Pico, reading sensor values, and implementing logic to detect hand gestures. Here's a basic outline of the process:

A. Raspberry Pico Setup

- PIN 1-2 are used for configuring UART cable connections. For communication between board and PC.
- PIN 3,8 and 38 are used as GND.
- PIN 36 is used for Power supply.
- PIN 31 is used for analog inputs (ADC0).
- PIN 19-20 is used to connect Pico board to ADC board.

B. Connect the Flex Sensor

- Connect one end of the flex sensor to 3.3V on the Pico.
- Connect the other end of the flex sensor to GPIO pin on the Pico.
- Connect a 10k resistor between the GPIO pin and the ground (GND) on the Pico.
- +3.3V —— Flex Sensor —— GPIO Pin —— 10k Resistor —— GND

C. Software Configuration of Pico

- Write a program to read the analog value from the flex sensor. You can use the Pico SDK functions or MicroPython libraries for this.
- Convert the analog reading to a meaningful range (e.g., 0 to 100).
- Implement logic to detect hand gestures based on the flex sensor values. For example, you might define thresholds for different gestures.

D. Functionalities Implemented

Operations	Hand Gestures
Ctrl +	L + R
Browser	L + M
Win + M	L + I
Win + D	L + T
Explorer	R + M
Settings	R + I
Snapshot	R + T
Ctrl -	M + I
TM	M + T
Ctrl + T	I + T
Ctrl + N	L + R + M
Win Tab	L + R + I
Ctrl R	L + R + T
Virtual Desktop	L + M + I
Close (Win+Ctlr+F4)	L + M + T
Max	L + I + T
Emoji	R + M + I
Cut	R + M + T
Copy	R + I + T
Paste	M + I + T
Tab	I (Open)
Run	M (Open)
Select All	R (Open)
Clipboard	L (Open)
Select	All

TABLE III
Gesture Control

E. Representation Reference Table

Symbol	Interpretation
T	Thumb
I	Index
M	Middle
R	Ring
L	Little

TABLE IV

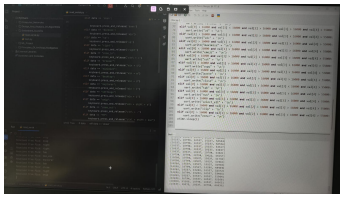

Fig. 2. Snap tool

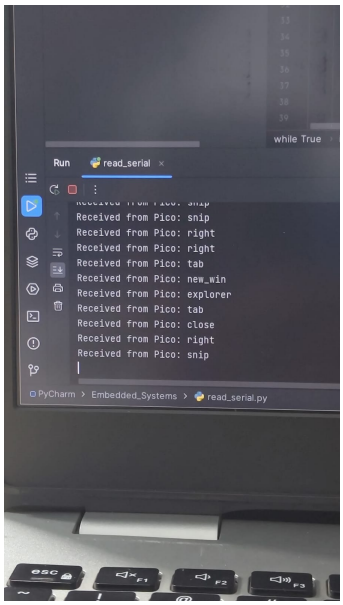

Fig. 3. Command Output Terminal

F. Implementing working of Snapshot Tool

- Fig 2. represents the activation of Snapshot tool in Windows PC using the gesture input of (R + T).
- Fig 3. denotes the command input and display in the output terminal in the CLI(Command Line Interface) of the program.
- Fig 4. denotes the input sensor readings obtained from the flex sensor during execution.

G. Implementing working of Opening a Browser

- Fig 5. represents the launch of Browser window in Windows PC using the gesture input of (L + M).
- Fig 6. denotes the command input and display in the output terminal in the CLI(Command Line Interface) of the program.
- Fig 7. denotes the input sensor readings obtained from the flex sensor during execution.

VIII.

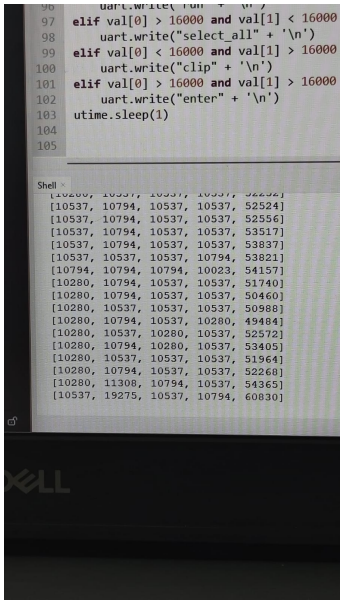

Fig. 4. Sensor Readings

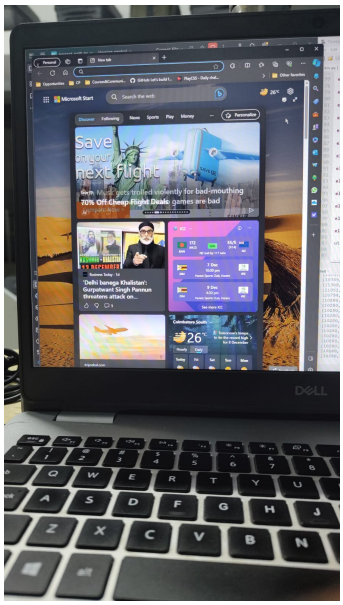

Fig. 5. Launching Browser

IX.

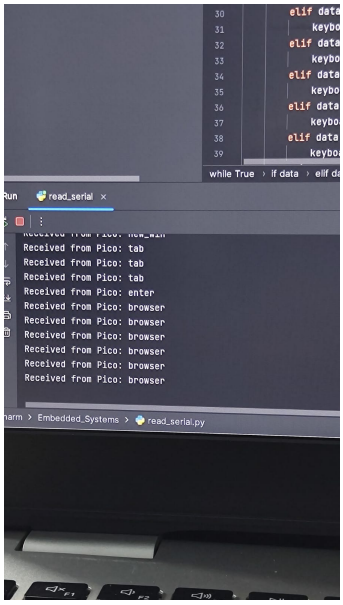

Fig. 6. Command Output Terminal

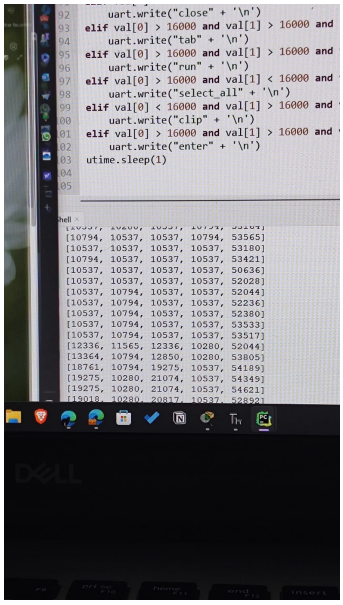

Fig. 7. Sensor Reading CLI

X. References

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6766835/
- https://www.irjmets.com/uploadedfiles/paper//issue$_{6}june_{2}022/26265/final/fin_{i}rjmets1655530772.pdf$
- https://circuitdigest.com/microcontroller-projects/raspberry-pi-flex-sensor