Handicap Assistance Device for Appliance Control using User-Defined Gestures

Dhairya Dand
VJTI
dhairyadand@gmail.com

Sisil Mehta
VJTI
sisilmet2000@gmail.com

Shashank Sabesan
VJTI
fierdor@gmail.com

Ankit Daftery
VJTI
ankitdaf@gmail.com

Abstract:- **Traditional input systems for interaction with machines include keyboards, joystick or the mouse. Those suffering from physical handicaps such as Carpel Tunnel Syndrome, Rheumatoid Arthritis or Quadriplegia may be unable to use such forms of input . In this paper, we propose a "Human Machine Interfacing Device" utilizing hand gestures to communicate with computers and other embedded systems acting as an intermediary to an appliance. Developments in field of communication have enabled computer commands being executed using hand gestures. Inertial navigation sensor like an accelerometer is utilized to get dynamic/static profile of movement to navigate the mouse on the computer or provide commands to appliances, thus accelerometer profiles are converted into wireless interactivity. The device involves non-tactile interaction with machines to manipulate or control them in accordance with hand gestures. The applications envisioned: interaction using gesture technology for effective communication empowering physically challenged to interact with machines and computing devices including 3-D graphic interactions and simulations**.

Keywords- **hand gesture technology; human machine interaction; remote screen management; accelerometer; wireless communication; Handicap assistive**

INTRODUCTION.

In the current era, embedded systems are being integrated into every aspect of our lives (eg. Microsoft Surface[1] and Automated Smart Homes) making it essential to move away from the conventional keyboard/mouse or keypad interface and delve into intuitive methods of interacting with the computers and other appliances around us.

Human hand gestures are a means of nonverbal interaction among people. They range from simple actions of pointing at objects and moving them around to the more complex ones that express our feelings or allow us to communicate with others. To exploit gestures in Human Machine Interfacing, it is necessary to provide the means by which they can be interpreted by computers or intermediate embedded systems enabling appliance control[2]. Though various efforts are being made in this direction [3] [4] they face issues of variable reliability, cost and convenience [5] and often need to provide samples to train the device [6]. There was an effort [7] in this direction which eventually lead to filing a patent [8] A careful study reveals that this device is inadequate for 3-D interactions. This paper describes our work in this direction mainly with the computer as the target device and highlights extending this system to an embedded platform as an intermediary to an appliance as a target device. The inertial sensor, the accelerometer along with a microcontroller, maps the orientation and position of the hand in 3-D space. This data is then transmitted to the end application where it can be interpreted and acted on, thereby simulating wireless interactivity.

The inspiration to develop such a device is manifold. Firstly, it aims at venturing into the field of physiological computing and permitting convenient interaction with the surrounding appliances through minimal hardware, as compared to complicated magnetic flux sensors or ultrasonics [4]. Also, the use of hand gestures for interaction [2] avoids the most prevalent injury due to continuous use of the keyboard and the mouse, the Carpal Tunnel Syndrome which occurs when the median nerve that runs from the forearm into the hand gets pressed at the wrist [9]. For the users suffering from Rheumatoid Arthritis[10] causing loss of function at the finger joints it would be easier to use simple hand gestures which don't require the use of fingers[11] but only hand movements to perform tasks on a computer. This technology is also of prime importance in dealing with 3-dimensional objects, effective teaching aid and for providing an alternative to the traditional joysticks in the gaming industry.

The authors have built such a device, a hand glove and demonstrated successfully at various technical events across India. This paper highlights the components used in the device and their interaction with the iMEMS system as applied to a computer (hardware and software). The hand

978-0-7695-3977-5/10 $26.00 © 2010 IEEE
DOI 10.1109/ICMLC.2010.18

gesture technology device, the hand glove, can be used to successfully perform the following activities on the computer screen with 100% success rate:

- Navigate through various Microsoft Windows menus
- Drag a window to a different location on the screen
- Move the cursor on screen
- Interact with on-screen elements.
- Maximize and minimize windows
- Perform Video editing
- Interact with 3-D objects/images on screen
- Gaming
- Control of external hardware such as robots.

The remaining paper has been organized in the following manner: Section II discusses the base design, section III discusses component selection and interfacing, section IV then briefly discusses the issues encountered while interfacing the components. Section V explains the algorithms and the various software involved in data interpretation at the computer end and section VI lists out the applications. The last section concludes the paper with possible further developments and references.

BASE DESIGN

A simple cotton glove is fitted with an accelerometer as shown in Figure 1(a) and Figure 1(b). This glove is then worn by the user and the desired actions are performed as shown in Figure 2. The accelerometer senses the orientation of the hand in space through an inbuilt capacitive system and feeds out electrical signals as unique voltages for each unique orientation. These voltages are converted to digital values by the microcontroller(Figure 3(a)) and then sent to the wireless XBEE(Figure 3(b)) transmitter-receiver pair, which passes on this data to the computer where it is manipulated by our program running in the background, to

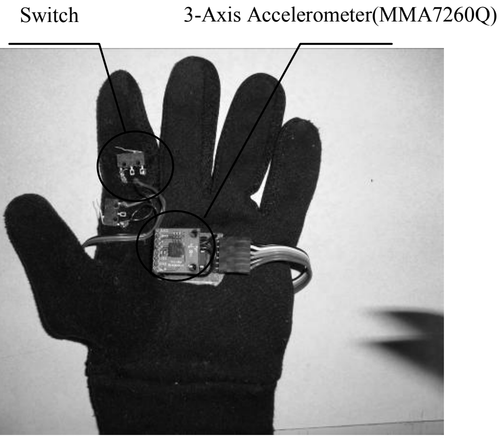

Figure 1(a):The Left Glove

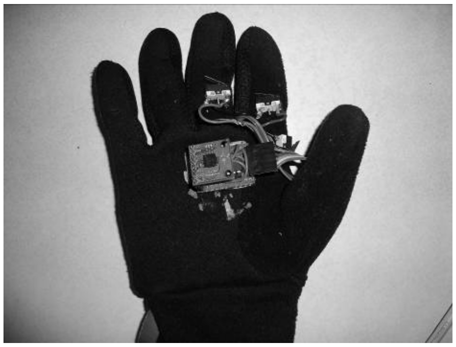

Figure 1(b): The Right Glove

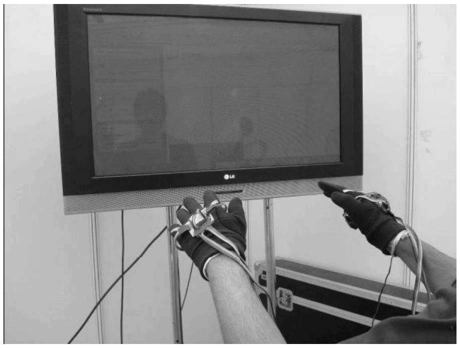

Figure 2: Use of Device

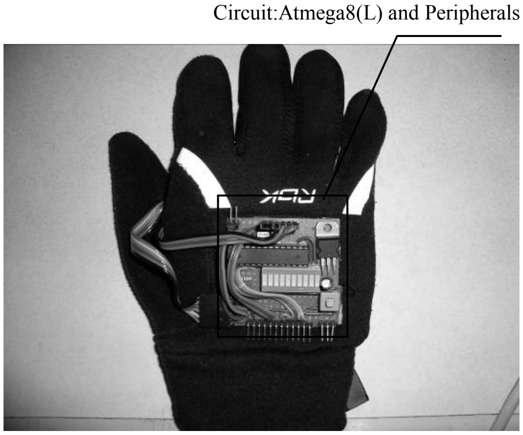

Figure 3(a): Right Glove, Main Circuit

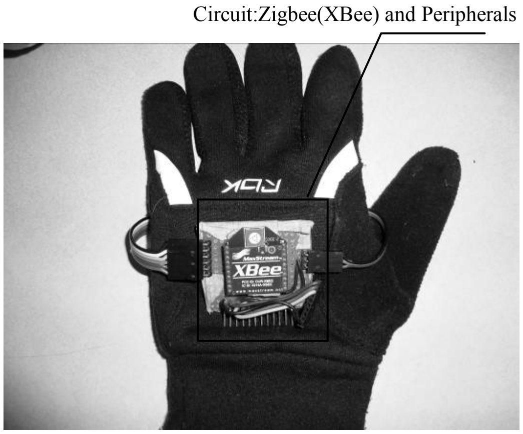

Figure3(b):Left Glove,Zigbee Circuit

I. COMPONENT SELECTION

A. *The Human-End*

The human dons the gloves that are fitted with the accelerometer module and the micro switches in the right as well as on the left hand. This means that mere movements of hand (which need to be predefined) will translate into corresponding interaction with on-screen elements.

B. *Accelerometer*

The accelerometer is the device which will generate corresponding electrical signals to control the on-screen elements. The accelerometer used here is the 3-Axis Accelerometer[12] (MMA7260Q) with an easy analog interface and running at a supply voltage of 3.3V, which makes it ideal for handheld battery powered electronics.

The accelerometer will experience acceleration in the range of +1g to -1g as the device is tilted from -90 degrees to +90 degrees. In order to determine the angle of tilt, θ, the A/D values from the accelerometer are sampled by the ADC channel on the microcontroller. The acceleration is compared to the zero-g offset to determine if it is a positive or negative acceleration. This value is then passed to the tilt algorithm[13]. When applied to all three axis, we are able to calculate the orientation of hand in three dimensional space.

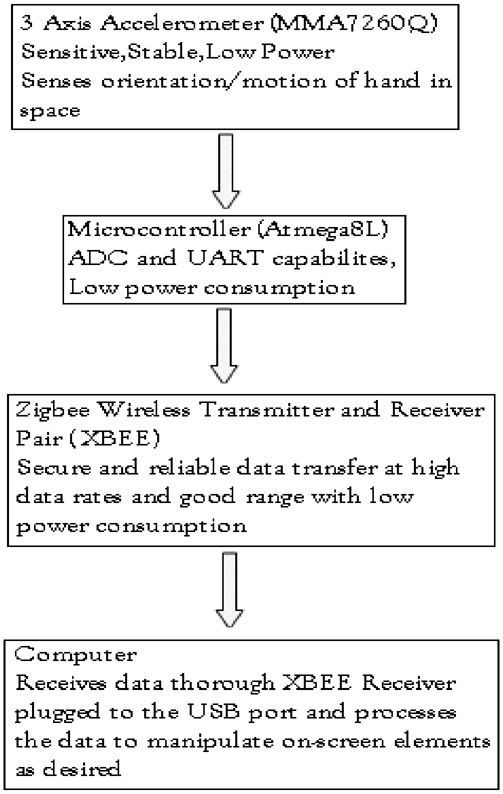
3 Axis Accelerometer (MMA7260Q)
Sensitive,Stable,Low Power
Senses orientation/motion of hand in space
Microcontroller (Atmega8L)
ADC and UART capabilites,
Low power consumption
Zigbee Wireless Transmitter and Receiver Pair (XBEE)
Secure and reliable data transfer at high data rates and good range with low power consumption
Computer
Receives data thorough XBEE Receiver plugged to the USB port and processes the data to manipulate on-screen elements as desired

Figure 4: General Data Flow in the Device

C. *Microcontroller*

The function of the microcontroller in this application is to act as an interpreter between the hand gestures and the end application. The ADC Port converts analog signals coming in from the accelerometer into corresponding 8-bit digital values. It then shifts out the result through the UART line to the device to be controlled. The ATmega8L[14] along with having all these features achieves throughputs approaching 1 MIPS per MHz by executing powerful instructions in a single clock cycle, allowing the system designer to optimize power consumption versus processing speed.

D. *XBEE and UART Data Flow*

The digital data is sent in the following order via the XBEE being operated in the transparent mode.

- Microswitches
- X-axis data
- Y-axis data
- Z-axis data

The XBEE[15] and XBEE -PRO OEM RF Modules operating in the ISM 2.4 GHz band were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The XBee receiver module connects into the USB hub of the computer or can be serially connected to other hardware. The advantage of USB over other data transfer standards is that USB is a widely spread and well accepted standard and it also provides plug and play thereby not requiring either the computer or the appliance to boot-up every time the device is connected to it.

The complete system, along with the data flow and interconnections are as shown in schematic diagram of Figure 4 and Figure 5.

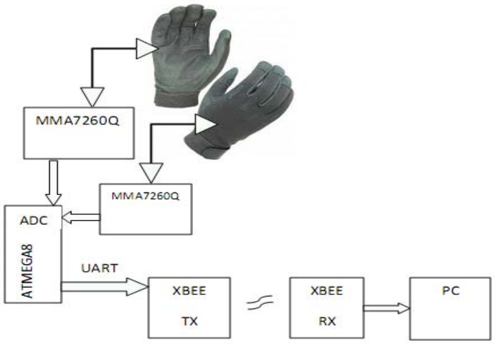

Figure 5: General Data Flow of the Circuit

II. INTERFACING ISSUES AND SOLUTIONS

Various problems arose during the development phase while interfacing. Some of them, along with their solutions are listed below.

- Selection of components had to be made keeping in mind the balance between the application requirements and the need for minimizing power consumption as well as weight, as the device was to be equipped on the user's hand.
- The major issues faced by us while interfacing included data communication between the Atmega8L and the computer and calibration of the on screen actions to the input received via the Zigbee.
- Selecting the right data type for carrying information was of utmost importance as it influences the entire data transfer process as well as transfer speeds. This was resolved by designating one data packet as four 8-bit frames. The frames contain button/click information, X Axis Voltage, Y Axis Voltage and Z Axis Voltage in that order.
- Another issue of great importance was security and reliability of communication and interference by other devices operating in the 2.4 GHz ISM band. The selection of the XBEE with capability for device identification resolves this issue.
- Scaling the data received by the computer to obtain the resulting motion on screen required a lot of tuning of the received data to obtain optimum sensitivity. The issue could be resolved by either hardware filters or software. The latter option was selected by us as it provided greater tuning and customizing ability without addition of any external hardware to the device.
- Selecting the correct sampling rate of data to attain optimum sensitivity and accuracy as well as to avoid picking up stray natural motions/vibrations of the hand is important. This issue was resolved by selecting an appropriate data transfer rate as well as putting in software checks to resolve the problem of stray natural motion.
- Another issue encountered was the question whether to totally replace the mouse input if present or to add to the existing mouse input stream, which was solved by selecting the latter, in order to provide a backup input.

V. ALGORITHM FOR DATA INTERPRETATION AND EXECUTION.

The serial data sent by the microcontroller via the UART is read by the software written in Visual Basic using the MSCOMM32.OCX module which interacts with the System Kernel [16]

However one may use any system programming language that can provide access to the core kernel libraries to interact with the mouse parameters.

The following settings are required for initialization –

1. Defining the Port no.
2. Setting up the baud rate (9600 bps is the default), setting up parity bits and other error correction parameters

The following are the essential functions that would be needed to be reused as modules

1. Constructor for the Mouse Pointer Initialization.
2. State functions for defining the mouse clicks and the associated events
3. Function for sending real time processed of the X, Y positional parameters to the kernel libraries

The overall code function can be summarized in the following -

A. Getting data from the Serial port

The data is sent in packets, each packet representing one single positional state. Each packet is 32 bits in size – consisting of 8 bits each of X, Y, Z and Button State information. The values of X,Y and Z range from 0 to 255 which are mapped from the voltage from the accelerometer.

B. Calibrating

Since the values of acceleration may vary with altitude as well as with various people using it. The displacement of the positional parameters due to unintentional motions is to be neutralized. For this, we take the first 50 set of packets as sampling packets – to these packets we separate out the individual values and take an average, the person wearing the glove is expected to rest his hands in the rest position. The normal value hence obtained is used for subsequent mapping of co-ordinates into proportionate displacement.

C. Error Detection

The sources of errors can be –

- Signal noise
- Intermittent connection at the hardware
- Packet loss

In order to detect errors in the data, we compare each individual packet data with the previous data and measure if the difference is within limits of the normal deviation expected (The estimate of the standard deviation is done at the Calibration stage) In case we have an abnormal deviation, we neglect the packet and compare the next packet with the last error-free packet. Since we have 300 packets arriving every second we can afford a packet loss up to 10 %

D. Mapping Dynamic Data onto Positional Parameters

In order to map the error free packet data onto the screen, we use the following formulae to come up with the proportionate placement of co-ordinates –

$$X_{onscreen}=(-1)*X * 0.4 * ((X * X) / X_{pix}) \qquad (1)$$

$$Y_{onscreen}=Y * 0.4 * ((Y * Y) / Y_{pix}) \qquad (2)$$

where X and Y are the co-ordinates from the error free packets, the X_{pix} and the Y_{pix} are the respective resolutions along the X and Y axis and $X_{onscreen}$ and $Y_{onscreen}$ are the resulting positions on the screen. The 0.4 constant is estimated and adjusted according to the aspect ratio of the screen.

The co-ordinates ($X_{onscreen}$ and $Y_{onscreen}$) hence obtained are passed onto the MouseMove() function which position the mouse co-ordinates to the respective position.

E. Processing Gestures

The raw coordinates obtained after the error correction are used in identifying gestures which are pre-defined. A button press on the glove puts it into the Gesture Mode; when in this mode, the co-ordinates are passed onto the Gesture Functions which constantly monitor and store previous co-ordinate data to recognize a pattern. These patterns can be customized and are to be pre-defined as required.

During the testing phase of the device some of the hand gesture patterns that we successfully used:

- Hand lift and drop motion to copy and paste.
- Left Hand vertical sweep motion to scroll
- Dual hand motion to turn a 3-D object.
- Right hand tilt motion to control the mouse.
- Left hand directional motion to control the arrow keys.
- Use of right directional motion and left hand tilt motion motion to play a video game(Call of Duty).
- Action button press along with Palm lift and drop motion to minimize and maximize windows respectively.
- Use of Right hand horizontal sweep to fast forward, reverse or pause a video.
- Action button press along with left hand lift and drop to change volume.

VI. EXTENDING THE SYSTEM TO EMBEDDED PLATFORM

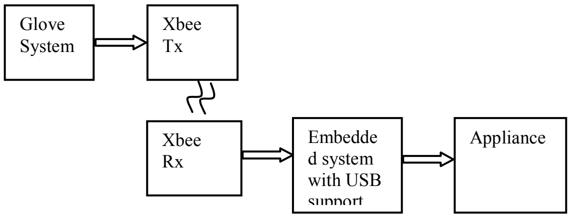

Figure 6: General Data Flow for Appliance as End Target

The incoming data from the glove can be passed to an embedded system acting as a blackbox interpreter. The embedded system can accept the action, x, y, and z data and convert it into appliance specific signals enabling gesture based control of the appliance. In order to use for multiple appliances, mode selection can be done using the action switches, with each action switch assigned to a particular appliance.

The TI MSP430 mixed signal processor [17] is an ideal blackbox interpreter for this application. It provides 12 channels of 10-bit A/D lines for high resolution gesture mapping. It provides software selectable dual internal reference level selection as well as external level selection. The Data Transfer Controller (DTC) attached to the A/D core eliminates the need for software intervention for data transfer on A/D conversion completion. The MSP430 can easily be interfaced with Zigbee hardware via the Universal Serial Communications interface, enabling real time transfer of data. The low Supply Voltage range of 1.8V to 3.6V and the Ultra Low Power consumption makes the MSP430 an ideal choice for a portable embedded platform.

Testing: This system was integrated with a wheelchair prototype to form a gesture controlled wheelchair .

The wheelchair prototype then moved in the direction perpendicular to the plane of the accelerometer.

The glove-system could be mounted on the palm as a glove or on the fore-arm as a wrist band and did not involve use of fingers. Also no action buttons were used since the system was tested with one application only.

VII. CONCLUSIONS AND APPLICATIONS

This proposed multi-functional portable device for better human – machine interaction using hand gestures can be applied in the following applications:

- Replace the mouse as a more convenient and natural interaction peripheral.
- Interacting with 3D objects on computer screen
- Easy control of Robots, Robotic Arms and Human Controlled Automation
- Easy Home Automation
- Effective Teaching / Animation / Design Aid
- Easy accessibility tool for people with disabilities
- When used with other inertial sensors (eg. gyros) the glove can be used to manipulate objects in 3 dimensions.
- It can be used extensively in the gaming industry for remote location manipulation.

VIII. FUTURE WORK.

We look forward to facilitate rich interactive features which would enable the users to interact and take portability to the next level. Use of smaller packages of the integrated circuits will scale down the size of the device to that of a watch, thereby improving the portability.

We are working on a Software Development Kit to ease the integration of our device into the software development tool-chain. It can be used by software developers to add hand gesture recognition to a variety of applications from games to educational software.

IX. REFERENCES

[1] Microsoft Surface http://www.microsoft.com/SURFACE/Default.aspx

[2] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 677-695, July 1997, doi:10.1109/34.598226

[3] Thomas B. Moeslund and Lau Nørgaard,2003, ”A Brief Overview of Hand Gestures used in Wearable Human Computer Interfaces”, Technical report: CVMT 03-02, ISSN: 1601-3646, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark

[4] Thomas G. Zimmerman , Jaron Lanier , Chuck Blanchard , Steve Bryson , Young Harvill, A hand gesture interface device, Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface, p.189-192, April 05-09, 1987, Toronto, Ontario, Canada

[5] Logitech Harmony One Advanced Universal Remote http://www.logitech.com/index.cfm/remotes/universal_remotes/devices/3898&cl=us,en

[6] Jiayang L., Wang Z., Zhong L., Wickramasuriya J., Vasudevan V., 2009, “uWave: Accelerometer-based personalized gesture recognition and its applications”, Pervasive Computing and Communications, IEEE International Conference on, Vol. 0 (2009), pp. 1-9

[7] Acceleglove by Anthrotronix http://www.acceleglove.com/ Patent # 6895305

[8] *Carpal Tunnel Syndrome* http://www.ninds.nih.gov/disorders/carpal_tunnel/detail_carpal_tunnel.htm

[9] Rheumatoid Arthritis http://www.nlm.nih.gov/medlineplus/rheumatoidarthritis.htm

[10] Tomoichi Takahashi , Fumio Kishino, Hand gesture coding based on experiments using a hand gesture interface device, ACM SIGCHI Bulletin, v.23 n.2, p.67-74, April 1991 [doi:10.1145/122488.122499]

[11] Koji-Tsukada,←-Michiaki-Yasumura, “UBI-FINGER: A SIMPLE GESTURE INPUT DEVICE FOR MOBILE”, Proceedings of APCHI 2002, Vol. 1, pp.388-400 (2002).

[12] FreeeScale MMA7260Q Data Sheet www.freescale.com/files/sensors/doc/fact_sheet/MMA7260QFS.pdf

[13] Atmega8L Data Sheet www.atmel.com/dyn/resources/prod_documents/doc2486.pdf

[14] Michelle Clifford and Leticia Gomez ,”Measuring Tile with Low-g Accelerometers”, Freescale Semiconductor, Application Note AN3107

[15] XBee Data Sheet www.mikrocontroller.net/attachment/29126/ZigBit_OEM_Module_Product_Datasheet.pdf

[16] Microsft Kernel Modules and Visual Basic Libraray www.msdn.microsoft.com/

[17] *MSP430 Overview and User Guide* *http://focus.ti.com/paramsearch/docs/parametricsearch.tsp?familyId=342&sectionId=95&tabId=1200&family=mcu*