SMART GLOVE

Shawn D'Souza
Flat no XF 8
Sunset Boulevard
Altinho Mapusa
Goa – India
S.R.I.E.I.T.
+91-9158804277
shawnelec@gmail.com

Aniruddh Gawade
C-2-D
All India Radio Colony
B.B. Borkar Road
Porvorim Goa – India
S.R.I.E.I.T.
+91-9423889600
aniruddhgawade@gmail.com

Siddesh Gad
R.H.6
Sayyadri Enclave A
Durgabhat Upper Bazar
Ponda Goa – India
S.R.I.E.I.T.
+91-9923247149
sidgad777@gmail.com

***Abstract*— Our project is designed to help the unfortunate people who cannot speak to communicate with us. The project consists of a glove equipped with flex sensors that will be worn by the person. A portable screen in the form of a badge will also be worn by the person. In case the need arises to communicate, the person will have to just make the signs corresponding to ASL (American Sign Language) symbolizing the various characters to form words and those words will be displayed on a graphical LCD.**

The system works by sensing the voltage drop across the flex sensors. This is then fed to the ADC input of the microcontroller. Here sampling is done and the digital values are stored (when unit is used for the first time), corresponding to each character. In this way the microcontroller learns the characters. When the sign is done again, these values are compared against the reference values in order to obtain the characters and pattern matching is done. The characters are transmitted to another microcontroller. These characters or words are sent to the display driver. The display driver then outputs the character or word on the portable screen.

The Smart Glove is multi-faceted. There are added facilities to control the mouse pointer of a computer via an accelerometer mounted on the glove and also to use it to input text to the computer just like a keyboard! The receiver unit is connected to the serial port of the computer and PS2 port. The user just switches between the text and mouse mode to use it as a mouse or keyboard. The right and left click is done by moving the index and middle finger just like an ordinary 'click' action!

When the user changes the mode to wireless control, he/she will be able to control his/her home appliances with only the glove. The glove acts as a master remote control where each character or combination of characters will represent a specific appliance. This too can be changed or set by the user. The slave unit is made up of another microcontroller that is wired to the appliance via a driver IC and relays.

This unit also can act as a control for a robot in the house in robot control mode. The robot can vary from a vacuum cleaner to a robotic wheel chair which the paralyzed or disabled person may sit on! Just by the movement in the hand, the robot is moved forward backward left or right. We have used the accelerometer mounted on the glove as input. In addition to this, the robot is also capable of storing the path traced by the user and following it automatically when told, without the user even controlling it!

Keywords—
Smart Glove: This refers to the glove with flex sensors mounted on each finger and accordingly linked to a microcontroller with a keypad for control. It has 4 basic modes: sign to text conversion, remote control of appliances, mouse control and text input for computer.

Multiple Sampling: This refers to the technique of taking many samples at an instant of time and then analysing each sample to find how much of a desired pattern is found in those samples. We then find the average to decrease the noise levels.

Cornering Algorithm: The name which was given to the pattern matching algorithm that we designed for this project. Here we match the characters to find greater than 80% probable match with an initial error factor. If many matches are there then we decrease the error factor and narrow down the match till we find a correct match each time taking note of the probability each time.

I. INTRODUCTION

In today's world, communication plays a vital role in each one's life. Vocal communication helps in speeding up information transfer. People who have voice are considered lucky since this precious God given gift is bestowed on them. What about those who are not so lucky to be blessed with a voice? How would this particular person communicate with the practical world?

Consider a scenario in which a person who cannot speak is waiting for bus at the bus stop and wants to ask about the bus timings. How would he communicate in sign language with another person who does not understand sign language? This sparked off the idea behind our project: to design a portable language translator that would convert this sign language into an understandable language (English). Along with this came the thought of making a universal controller for the same person who wears the smart glove to ease his/her life and help empower him/her. So we thought of four basic modes to be followed.

- Sign to text mode
- Keyboard and mouse mode
- Device/appliance control mode
- Robot control mode

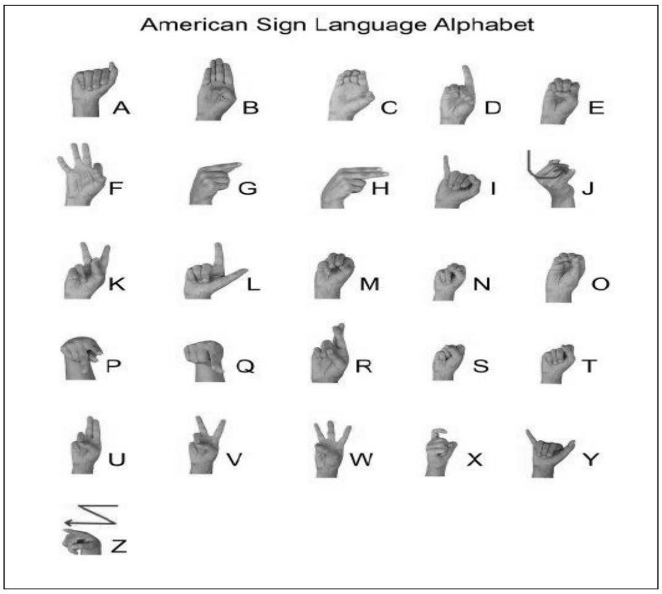


Figure 1: The American Sign Language

In the normal sign to text converter mode, we have the smart glove and a portable display which will be worn by the handicapped person who cannot speak. On this glove, we mount flex sensors that sense the bends in the fingers of that person. Proportional to the amount of bend, the resistance of the flex sensor increases. The flex sensors are connected to the microcontroller ADC input port.

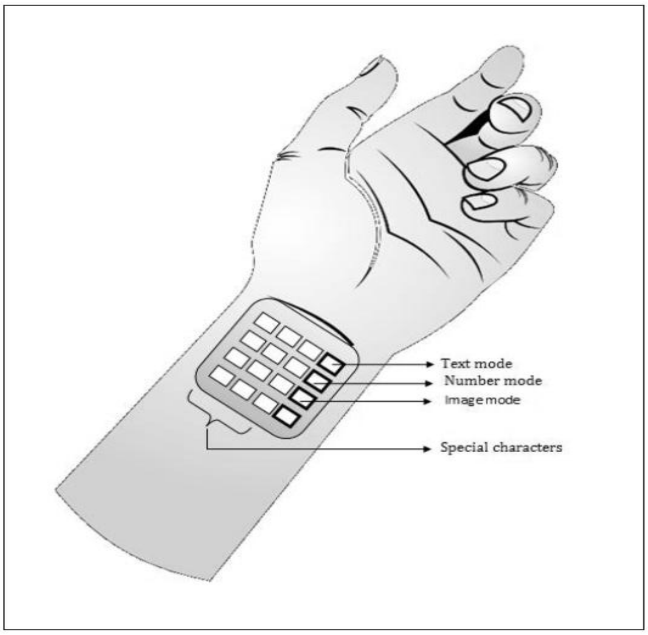


Figure 2: The Mounting of the keypad

This change in resistance causes a change in voltage across it. The analog voltage is then converted into digital form by the ADC present in the microcontroller. Here the digital values are stored for future pattern matching. When a sign is made, pattern matching is done and if the probability of match is high, it is taken as that character. This is then displayed on the portable LCD screen.

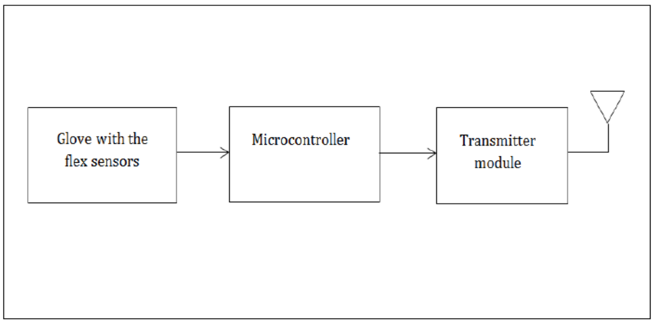


Figure 3: Block diagram of the transmitter unit

When the mode is changed to 'mouse and keyboard', the glove acts as a remote keyboard and mouse. The receiver is connected to the computer's serial port. When the glove is waved in the air, an accelerometer present on the glove senses the hand motion and accordingly an analog voltage is generated. Based on this, the mouse pointer is moved on the screen. Right or left 'click' is done the normal way with the index and middle finger. To act as a keyboard, the user has to just make the signs and the character will be sent to the computer. In the robot control mode, the user tilts his/her hand in order to drive the robot in the desired direction just like he/she did in order to control the mouse.

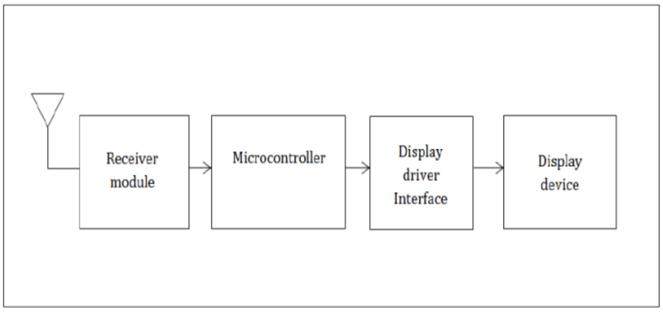


Figure 4: Block diagram of the receiver unit

I. HARDWARE

Atmgea 32 is the microcontroller chosen for the project. A slave (ATmega 32) is used as the display driver interface for the graphic LCD and a master (ATmega 32) is used as the interface for reading the inputs form the flex sensors in the glove. Subsequent slave (Atmega 32) microcontrollers can be used and configured to control appliances or control the robotic wheel chair for example.

The ADC input of the master is used to read the analog values form the flex sensors and accelerometer and convert them into digital values. For this project we have used the CC2500 wireless module which has a range of about 30 meters and 0 to 255 different channels. Each mode is assigned a separate channel of operation. The main master controller knows which channel to address for which mode. For driving the graphic LCD, an embedded graphic library (compatible with any KS0108B based (128 x 64 pixels) LCD driver) was designed. Some of the hardware used is described below.

A. *Flex Sensors:*
Flex sensors are sensors that change in resistance depending on the amount of bends in the sensors. They convert the change in bend to electrical resistance. More the bend, more the resistance value. They are usually in the form of thin strip from 1”-5” long that vary in resistance. They can be made unidirectional or bidirectional. They can be of various sizes and range lies between 1Ω-20kΩ, 20kΩ-50kΩ and 50kΩ-200kΩ.

B. *Microcontroller ATmega 32:*
This is a 40-pin microcontroller with 4 general purpose I/O ports. This 8-bit microcontroller can give a throughput of 16 MIPS at 16 MHz

C. *Graphic LCD 128x64 (128G064E):*
This graphic LCD has a 20 pin interface with 128 pixels spanning the x axis and 64 pixels spanning the y-axis, giving a total resolution of 8192 pixels. Each character is plotted in a 64 pixel grid (8 x 8 pixels). This gives 8 rows of (16 characters per row) characters that can be displayed on the LCD.

D. *CC2500 wireless transceiver module:*
The CC2500 is a 2.4GHz multi-channel transceiver RF module. It operates with a 2.4 GHz carrier and the data is modulated with FSK (Frequency Shift Keying) modulation. It’s designed specifically for high speed data reception and transmission.

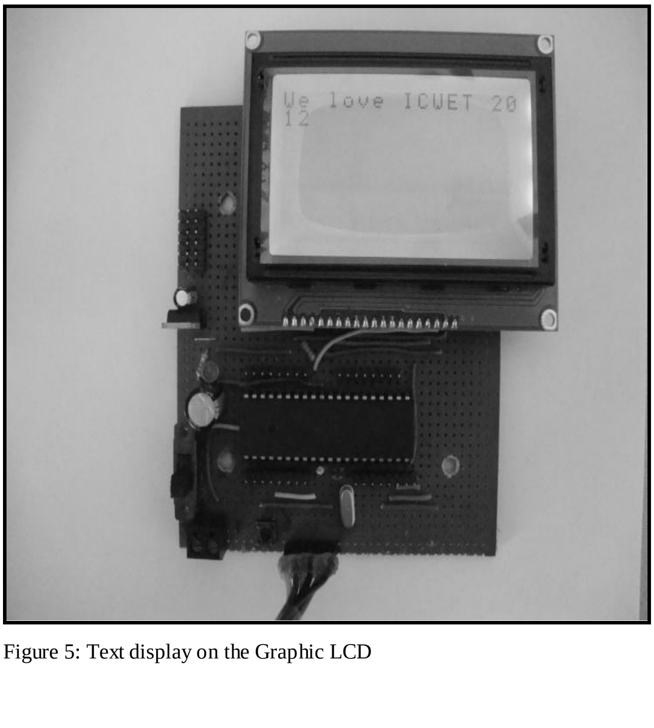


Figure 5: Text display on the Graphic LCD

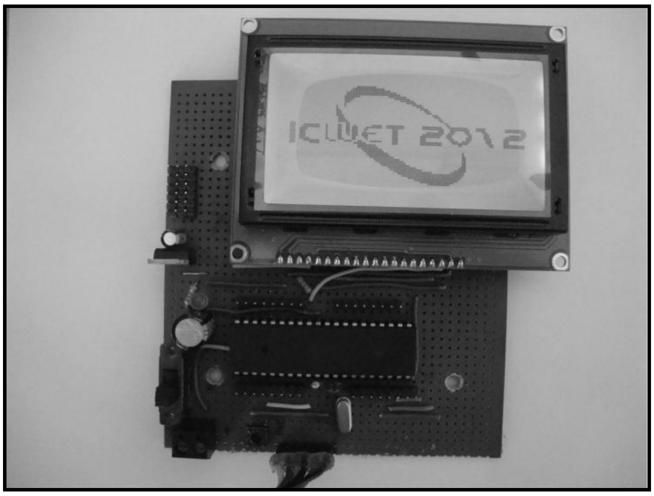


Figure 6: Image display on Graphic LCD

E. *4 x 4 keypad:*
The 4 x 4 keypad is divided into 4 rows and 4 columns. At each cross point, a switch is connected. To each column, a high pulse is applied in succession while the rest are held at high impedance (or input). When each column is made high, all rows are scanned, and the corresponding row which reads a high is the row at which the button is pressed. The column is known by the one which is made high. Hence knowing the corresponding row and column gives us exactly which button is pressed on the keypad.

F. *Accelerometer ADXL 335:*
The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

G. *I.C. EM84510F (PS2 mouse controller):*
The EM84510 Scrolling Mouse Controller is specially designed to control PS/2 mouse device. This single chip can interface three key-switches four photo-couples plus z-axis direct to 8042 controller. EM84510 can receive command and echo status or data format which are compatible with IBM PS/2 mode mouse. Key debouncing circuit is provided to prevent false entry and improve the accuracy. The EM84510 mouse controller provides noise immunity circuits to eliminate these noises in order to reduce energy consumption.

II. SOFTWARE

The Glove Transmitter Unit (for the sign to text or number mode)

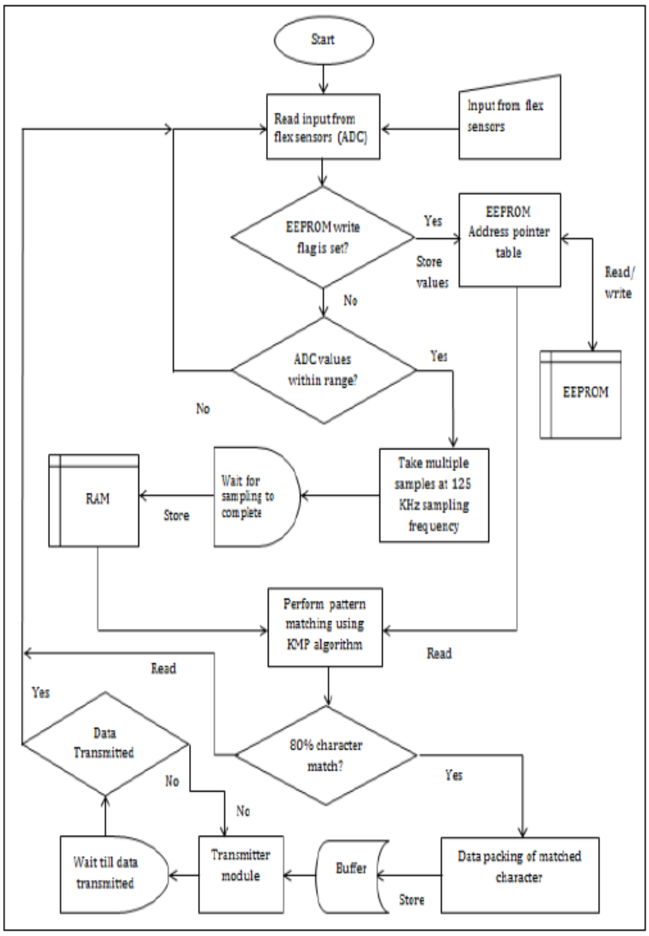


Figure 7: The Glove Transmitter Unit Flow Chart

The system works on the real time samples taken and pattern matched with the already stored samples. Multiple samples are taken and the probability of an approximate match is calculated. Taking the probability of multiple samples reduces error and increases the probability of success of a match. For example the probability of finding a person wearing a red shirt in a sample space of 1000 is higher than the probability of finding a red shirt in a sample space of just 100. When the user first wears the glove, the unit does not know the bend values of his/her fingers as nothing is stored in the EEPROM. Hence the unit first checks if there are values stored in the EEPROM before proceeding with pattern matching. If samples are already stored in the EEPROM, the EEPROM write flag is set. Hence the unit can proceed with the pattern matching. If the flag is reset (i.e. 0) then the unit asks the user to store the samples corresponding to each letter before proceeding.

For real-time pattern matching, the unit checks if the ADC sampled values are within a preset range. If these values lie within that range, the unit then starts to take multiple samples of the voltage drop values. The ADC has a sampling frequency of around 125 KHz. This frequency is enough to give us around 100 samples in about a millisecond. We have to wait till sampling is complete and hence a small delay is introduced.

These ADC real-time samples are temporarily stored on the RAM. Next the KMP algorithm is executed and the values are read from the EEPROM and matched against the values stored in the RAM. If there is a 80% or more probability of a character match, then that character is returned as the matched character. For reading the values from the EEPROM, a special page table is created which contains the character address and the pointer to the physical EEPROM address location. A ± 30 difference between the patterns matched is tolerated. This is done so as to compensate for the errors in sampling and physical sensor defects. Then we apply the 'Cornering' algorithm to find the correct match.

This character is packed along with its checksum and a flag byte. This is implemented in as a data structure consisting of flag (char), the data (int) and checksum (int). Before transmission, the data is buffered. This is done because at some point of time, the sampling may be faster than the data transmission or the character may be returned earlier than expected or the data transmitted may be slow. To deal with all these conditions, a buffer is inserted which ensures that data is transmitted with a smoothly. The data is then transmitted via the SPI port.

The Receiver Unit (for the sign to text or number mode)

At the receiver's end, the data is read via the SPI port. The read data from the register is buffered. The data is then unpacked and the packed is checked for a valid flag and checksum. If the checksum is good then the data is returned. Each character (data) is sent to the display driver software. There is a special font array bitmap file that contains the representations (in bits) of the character on the graphic LCD.

A character match of the ASCII value against the font array index is done. The font array is arranged so that the ASCII value of each character corresponds to the respective bit value in that index location of the array. Since the graphic LCD is divided into 8 x 8 pixels for each character, the font array is read for 8 consecutive locations, with each location containing 8 bits data. (1bit is allocated to each pixel). Once the character bitmap is returned, the data is sent to the Graphic LCD data port for display. The data is placed on the port, subsequently making the EN pin high in order to read it and low while changing data on the port.

III. THE CORNERING ALGORITHM

The main algorithm that is followed in this project is the Knuth Morris Pratt string matching algorithm combined with the 'cornering algorithm' which is specially designed by us for this project. The cornering algorithm was designed to help in efficient pattern matching taking into consideration the real time ADC values of the flex sensors and the error rates along with the noise input in the unit. The algorithm is explained in the following steps:

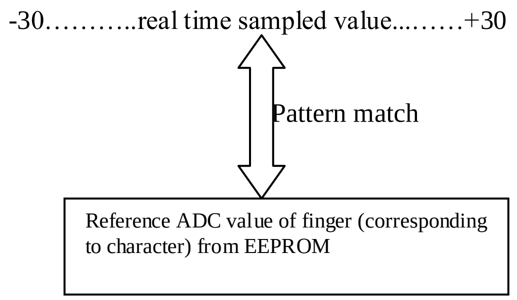
- **STEP 1:** Multiple sampling is done and an average is taken so as to reduce the noise irregularities in the input.

Sample[i][j] stores a sample value for a single five finger sample. ‘i’ ranges from 0 to 5 and ‘j’ ranges from 0 to n number of samples taken for that finger samples. Hence,

Sample[i][j] = ($\sum$ ADC values for that finger) / n

- **STEP 2:** Take a wide error factor window say ±30 and find out which reference character sample matches the real time sampled values with an 80% or more probability using Knuth Morris Pratt Algorithm. If there is only one match then the character is the match and the function will return the character’s ASCII value.

If not then all the other 80% or higher probability character matches are stored and are used for the remaining steps.



Error window

- **STEP 3:** If there is a tie of two or more 80% or higher probability match characters, determine a heuristic value dynamically based on the least distance of a real-time character sample from the reference value considering all the five samples at a time.

Character_sample_distance [i] = Realtime_Value[i] - Reference_Value [i]

Total Distance[j] = $\sum$ abs{Character_sample_distance [i] }

Pchar = Indexof_Character (min{ Total Distance[j]})

Heuristic Value:

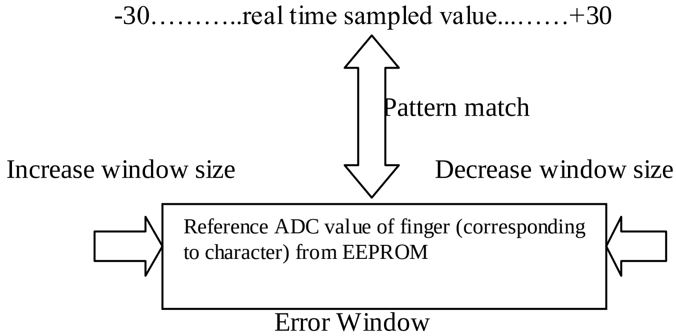
H_value (Pchar)= (min{ Character_sample_distance [i] , max {Character_sample_distance [i]})

If a tie exists between two or more characters, then we choose randomly any one and compute the heuristic value. The entire ‘80% or more probably matched character sample’ indices are stored. Then we proceed to step 4.

- **STEP 4:** The error factor window is decremented by a heuristic value depending on which side the match is more probable (i.e. either towards the +ve or –ve side of the reference) and these real time character samples are again pattern matched against the corresponding reference values with the new error factor window. The maximum samples (all five finger samples) that fit inside the window give us the probability of a strong match.

P_Window = Reference_value + H_ value (+ve end)

N_Window = Reference_value - H _value (-ve end)



Error Window

If (a single character match is found) and (it is the only character with the least distance)
Then it is returned as the character
Else if (there are more characters that tie for the least distance) go to STEP 5
Else if (there is a tie in matched characters) then return -1

- **STEP 5:** Take the next character in line with least distance. Calculate the heuristic value and Repeat STEP 4. Store all the pattern match probabilities (or window fit) of the characters.

- **STEP 6:** Repeat STEP 5 till the proper match is found. Compare the probability with the previous character. The character which gives the highest probability (most samples fitting in the window) is returned as the character.

  If there is a tie of two or more matched characters then return -1

Here there is no match or the match is not accurate. Hence we exit from the loop and do not return any character back. When we return -1, implies that a match is not possible from the samples.

The aim is to try to fit most of the real-time samples into the narrowest window with least error factor and with a success probability of 0.8 or more (80% or higher). So we corner down on a particular value and increase or decrease the window so that we can get the most accurate match.

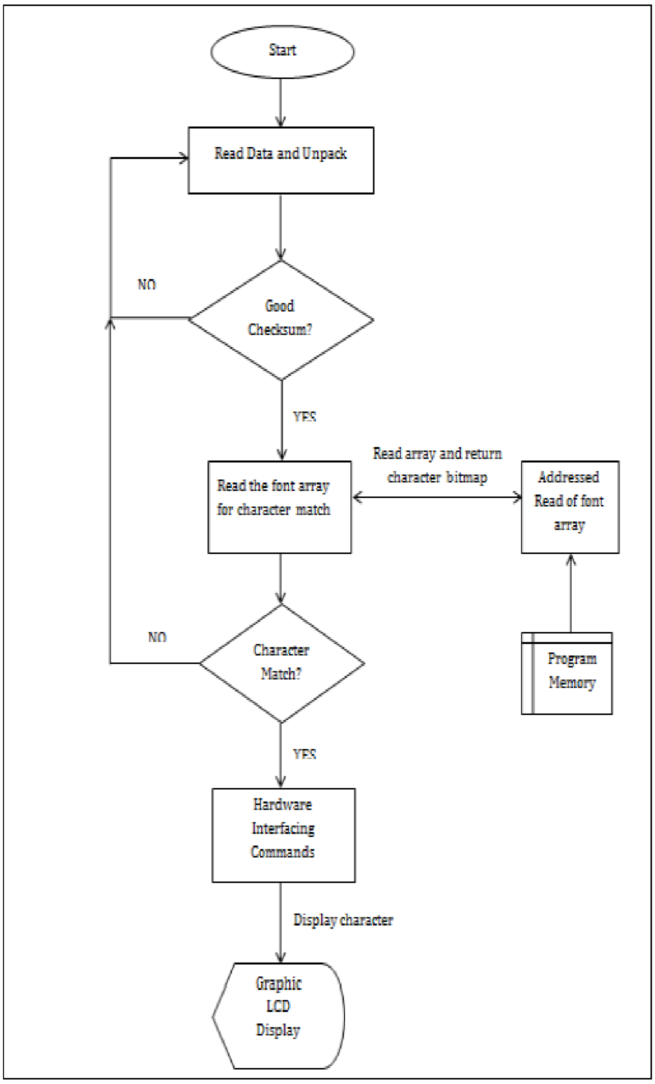


Figure 8: The Receiver Unit Flow Chart

IV. OTHER APPLICATIONS

A. Mouse and keyboard mode:

For the keyboard mode, the characters received by the unit are transmitted via the serial port to the computer. On the computer is installed small software that accepts the character and displays it on the computer screen. For the unit to act as a mouse, an accelerometer installed in the glove converts the movement of the hand into analog voltages. These voltages are fed to the ADC input of the microcontroller in the glove. The digital values are then data packed with a checksum and sent wirelessly to the receiver unit.

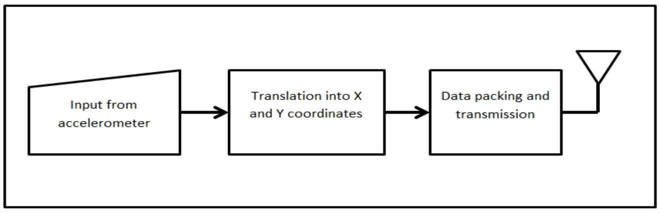


Figure 9: The Glove Transmitter Unit for mouse mode (Block diagram)

The receiver unit consists of a microcontroller whose output is linked to the I.C. EM84510F (PS2 mouse controller) X-Y input pins. A small code is written for the microcontroller to convert these analog voltage values to the corresponding X-Y motion. The PS2 mouse controller then controls the mouse pointer on the screen like a normal mouse. For the right click and left click, the flex sensors on the index and middle finger are employed. An analog threshold value is set for the left and right 'click' buttons in microcontroller of the transmitter module. When it is crossed a click code signal is sent to the receiver. The receiver accordingly controls the PS2 controller to emulate a right 'click' or left 'click' of a mouse.

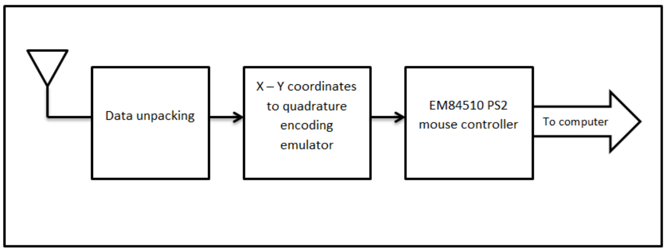


Figure 10: The Receiver Unit (Block Diagram)

B. Wireless remote control:

In this mode, the glove acts as a wireless control for the appliances. For the first use, the user makes the signs corresponding to each device he wishes to control. These signs are mapped in the EEPROM corresponding to each device. Hence when the sign is made the next time, it is pattern matched just like the in text mode and transmitted to a receiver unit.

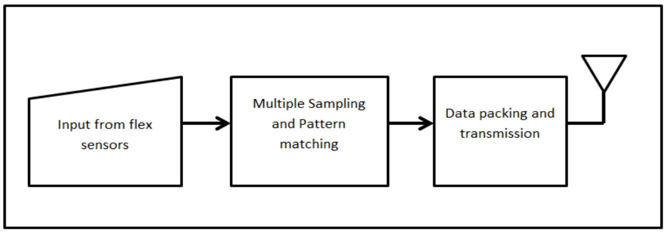


Figure 11: The Glove Transmitter Unit for Appliance Control (Block diagram)

Instead of displaying characters on the LCD, the unit will control the user's appliances. On the receiver's end, the microcontroller ports are configured accordingly to act as normal outputs for controlling the appliances. These are linked to a ULN2003 driver I.C. to drive relays to control the user's loads/appliances that operate of 230VAC.

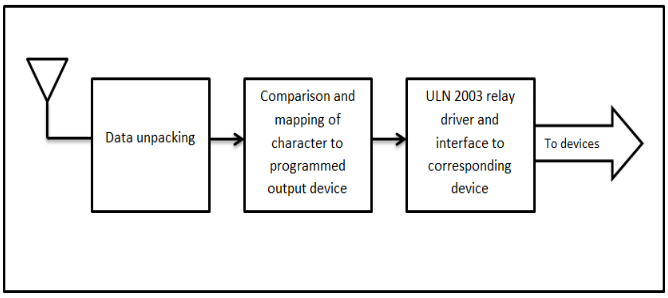


Figure 12: The Receiver Unit for Appliance Control (Block diagram)

C. Software and Hardware Tools:
The whole project is done in embedded C (for AVR microcontrollers). Here we use AVR Studio 4 in order to program and AVRdude to burn the code into the microcontroller with a USBtiny ISP programming device.

For using the smart glove as a virtual keyboard, a small Visual Basic Script is written for interfacing it with the computer. For the mouse control, the Operating System's driver software recognizes the EM84510 as a 'standard PS2 mouse controller' directly and hence there is no need for designing special interfacing software.

V. SYSTEM FEATURES

A. *Self-Learning: Embedded database creation*
Here we have used the EEPROM memory as a database or 'hard disk' for storing the sampled values at real-time. The user wears the glove and makes signs corresponding to a letter. As the signs are made, the digital values after conversion are mapped in the EEPROM. A pointer table which contains the address pointer and an index assigned to each character is framed. This pointer points to a specific start address within the EEPROM memory. Since we have five 8-bit values to be stored (corresponding to the 5 fingers), five consecutive locations are assigned on the EEPROM for each character. Hence the pointer to each location is incremented by five to read the reference sample corresponding to each character.

B. *Pattern Matching Algorithms:*
Pattern Matching is the act of checking some sequence of tokens for the presence of the constituents of some pattern. In our case of pattern recognition, the match usually need not has to be exact, it can be partial also. Using the KMP algorithm and the 'Cornering' algorithm we make pattern matching more efficient.

C. *Easy, Fast and Efficient translation:*
Portable sign language translator provides an ease in the flow of communication and behaves as an interface between sign language and Standard English language. Here users (who cannot speak) just have to wear a glove and provide input to the device by simply doing signs. Sampling for analog to digital conversion is done at a frequency of 125 KHz (8 microseconds for each sample). This is fast enough to cope with the speed of the person making the sign. Multiple sampling and real time pattern matching improves the translation accuracy.

Other features:

- Character displayed vividly on the display.
- With machine learning, the user can train the unit to adapt to his/her own hand.
- Faster and better character recognition by using multiple sampling and the 'cornering' algorithm.
- Portability: can be used anywhere
- Provision for other language support
- Provision for image support
- Battery operated and safe to use

VI. SYSTEM ANALYSIS

- The bend resistance offered by the flex sensor ranges from 45KΩ to 125KΩ. This gives a potential difference across the flex sensor of around 2.5V to 3.676V respectively. (When connected with a voltage divider with 45KΩ to Vcc and the flex sensor to Gnd.)
- This gives an 8-bit ADC value range from 128 to 188 respectively. (when the Aref pin is tied to Vcc)
- For graphic display, we require around 2KB or more of RAM for proper operation and display to the graphic LCD.
- For wrist movement sensing an accelerometer will enhance the performance if mounted in the palm of the hand glove.
- Character match is much faster and more accurate if multiple sampling techniques are adopted. The 'Cornering Algorithm' designed by us performs better than the KMP giving a match time of 100ms with high accuracy over 500ms for KMP with lesser accuracy.
- Image display is only possible if the bitmap is read straight from the flash memory of the microcontroller instead of storing is as 'data' on the RAM. If stored on the RAM, since each image is 1KB, the microcontroller runs out of RAM space fast, and this leads to loss of the important data values from the RAM (as those values are overwritten)
- Each character is of 8 x 8 pixels. This gives 16 characters per line and 8 such lines.
- The graphic display is divided into two halves of 64 x 64 pixels this giving a combined resolution of 128 x 64 pixels. Each half is controlled by a separate KS0108 chip. Hence we have CS1 and CS2 for selecting each half of the display.

VII. WORKING MODES

A. *Text and Number mode:*
In this mode, the user has to just make the signs in ASL and these signs will be translated into its corresponding character and is displayed on the screen. In number mode, the unit will accept signs corresponding to each number and will display the corresponding number done by the user. The text and number mode is selected by a simple button present on the keypad. Special symbols are input by special keys present on the keypad itself.

B. *Mouse and keyboard mode:*
In this mode, the user has the flexibility to connect the receiver unit to the computer and make it act as a combined mouse and keyboard. The glove has to be worn by the person and he/she has to just make the signs corresponding to ASL. The respective characters will appear on the computer screen. Hence this unit acts as a keyboard. By just pressing a button on the keypad, the user can toggle between using the glove as a mouse and a keyboard. Since the glove is also equipped with an accelerometer, the user will just have to move his/her hand in the air and the mouse pointer moves correspondingly on the

screen. The right and left click is done by moving the index and middle finger of the hand just like clicking a mouse in the air!

C. *Wireless remote control:*
The user also has the facility to control his/her home appliances by doing some keyword signs for each respective device. The user has the flexibility to train the unit to respond to a particular sign for that device. The number of devices is limited to the number of output line available in the unit. All the modes are toggled with the help of a convenient keypad placed on the glove.

D. *Robot control mode:*
Here the user can control a robot with the help of the Smart Glove. The accelerometer mounted in the glove senses the acceleration/movement of the hand in a particular direction and we get varying voltages as output. The ADC converts these voltages into digital values and accordingly the acceleration in that direction is measured by the master microcontroller in the glove unit. This helps to interpret the movement of the hand and the microcontroller sends commands to the robot to move in that direction. For example, if the hand is moved forward, the robot also moves forward. A slight tilt in the hand will cause the robot to turn right or left depending on which side the hand was turned.

VIII. CONCLUSION

This project aims at helping humankind; especially those who cannot speak. This portable translator makes it a handy device to be used on the go. Using this device enables clear communication between people and helps the person who doesn't know sign language to understand its meaning and reciprocate in a better way. The very fact that this unit can actually learn the signs makes it more flexible to adjust to different people. This device can also be used as an aid to enhance the learning of sign language for students and elders alike.
In addition to all this, the Smart Glove provides a lot of other useful functions that will surely make the life of such unfortunate people a lot better and easier. The mouse control and text input combined with a full-fledged flexible and user programmable remote control makes the Smart Glove a great contraption.

IX. ACKNOWLEDGEMENTS

The credit for this project goes to the sincere and timely guidance received from various people. We wouldn't be able to proceed smoothly with the design, without the constant encouragement and support received from our well-wishers.

We wish to convey our sincere gratitude to our guide Mrs. D.S. Vidhya, Asst. Professor, IT Department, SRIEIT, Shiroda who encouraged and helped us in completion of the design phase.

Our heartfelt gratitude goes out to our Dearest Parents for their invaluable encouragement, timely guidance and inspiration that made it possible to come this far with the project. Not to forget the principal Sr. Monica of Xavier Academy (School for the Differently abled) for her valuable information she offered on the use of sign language and what is needed in the project. Her valuable suggestions and information helped us to design the product as per how the actual user may want it.

X. REFERENCES

Datasheets (pdf version):
[1] Flex sensor 4.5 inches
[2] Atmega 32 microcontroller
[3] KS0108B Graphic LCD driver
[4] LCD-128G064E Graphic LCD datasheet
[5] CC2500

E-books:
[1] Avr Studio Software Tutorial by Sepehr Naimi
[2] Tuxgraphics AVR C-programming tutorial (http://tuxgraphics.org/electronics)
[3] C Programming for Microcontrollers- Quick Start Guide: The Next Generation by Joe Pardue

Websites:
[1] http://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm
[2] http://www.cmi.ac.in/~kshitij/talks/kmp-talk/kmp.pdf
[3] http://www.ics.uci.edu/~eppstein/161/960227.html
[4] http://extremeelectronics.co.in/avr-tutorials/interfacing-graphical-lcd-with-avr-mcu-%E2%80%93-part-i
[5] http://extremeelectronics.co.in/avr-tutorials/interfacing-graphical-lcd-with-avr-mcu-%E2%80%93-part-ii
[6] http://extremeelectronics.co.in/avr-tutorials/interfacing-graphical-lcd-with-avr-mcu-%E2%80%93-part-iii