

Real time Hand Gesture Recognition using AVR Microcontroller

M. Shakunthaladevi
Dept. of Embedded System technologies
SKP Engineering College
Tiruvannamalai, India

R. B. Revathi.
Assistant Professor, ECE
SKP Engineering College
Tiruvannamalai, India

Abstract—The hand gestures provide a natural and intuitive communication modality for human-computer interaction. Efficient human computer interfaces (HCIs) have to be developed to allow computers to visually recognize in real time hand gestures. This project deals with real time hand gesture recognition using AVR Microcontroller. The existing system uses Digital camera for hand gesture recognition. The camera was used as an input device and the object need to present in front of the camera for capturing gesture. It limits mobility and cannot use camera in dark area as well as cost is very high. The camera specifications significantly affect GRS (Gesture Recognition System) characteristics. The proposed system, based on the input signal from the MEMS (Micro Electro Mechanical System) accelerometer sensors measure the acceleration such as tilt, shock & vibrate. It is reliably and accurate detect. Here, the tilt motion information produced by the human subjects and transmitted to the AVR microcontroller i.e. ATmega32A. The sign sequence and template matching algorithm developed to identify the gesture in a sequence. Based on the algorithm developed the microcontroller display the position of the hand in a display unit. We conclude that the recognition algorithm based on sign sequence and template matching as presents in this project can be used for non-specific users hand gesture recognition.

Index Terms---Gesture recognition, hand gesture, AVR microcontroller

I. INTRODUCTION

The objective of the project is to design a light-weight, wearable, gesture recognition system which is capable of recognizing hand gestures, based on the input signals from MEMS 3- axes accelerometers, ADXL335. Based on the algorithm developed the microcontroller displays the position of the hand in a display unit. Most of existing systems in the gesture recognition follows image-based approaches. Mostly cameras were used as input

devices. Object needs to be present in front of the cameras for capturing gestures, which limits the mobility. The proposed recognition system is implemented based on an inertial

measurement unit based on MEMS accelerometer sensors. The real time hand gesture recognition system which is capable of recognizing Hand Gestures of three axis such as X-axis, Y-axis and Z-axis of each gesture of human hand such as normal, up, down, left, right by using AVR microcontroller i.e.(ATmega32a). Based on the input signals from the MEMS 3-axis accelerometer will be transmitted to ADC i.e. (analogue to digital converter), which is inbuilt in ATmega32a in AVR developer board. Based on the algorithm developed, AVR Micro-controller will display the hand position of the display unit with the help of serial communication i.e., rs232.

II. BLOCK DIAGRAM

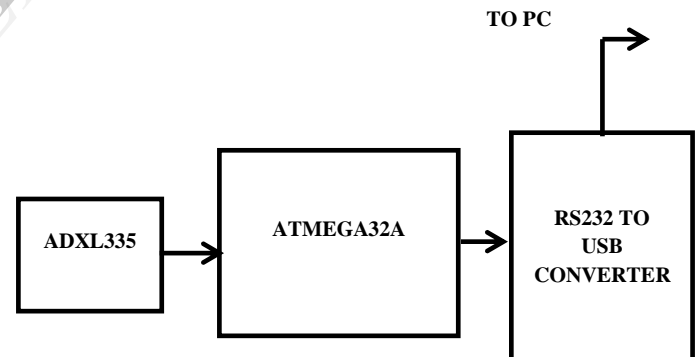


Fig. 1. Block Diagram

III. HARDWARE COMPONENTS

The hardware components used in the development of the project are

- A) ADXL335 MEMS 3-Axis Accelerometer,
- B) Atmega32A development board ,
- C) USB to Serial converter,
- D) AVR-ISP programmer.

A. ADXL335 MEMS 3-Axis Accelerometer

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. At midpoint when acceleration is 0 g output is typically 1/2 of supply voltage. If a supply voltage is 3V, then output is 1.5 V. Output sensitivity typically is 300 mV/g. The connecting ADXL335 to AVR developer board is simple. The accelerometer can be supplied from the 3.3 V output at AVR developer board, however then midpoint voltage and sensitivity is different from specified at datasheet. I used 3.0 V supply voltage, it came from voltage regulator LM317. Outputs of X, Y, Z axis are connected directly to analog inputs of AVR.



Fig. 2. ADXL335 MEMS 3-Axis Accelerometer

B. Atmega32A development board

ATmega32A is an 8-bit high performance microcontroller of Atmel's Mega AVR family. Atmega32 is based on enhanced RISC (Reduced Instruction Set Computing) architecture with 131 powerful instructions. Most of the instructions execute in one machine cycle. Atmega32 can work on a maximum frequency of 16MHz. The important features of ATmega32A microcontroller are

- High-performance, Low-power AtmelAVR8-bit Microcontroller.
- Advanced RISC Architecture
- 131 Powerful Instructions – Most Single-clock Cycle Execution
- 32 × 8 General Purpose Working Registers
- Up to 16MIPS Throughput at 16MHz
- High Endurance Non-volatile Memory segments
- 32Kbytes of In-System Self-programmable Flash program memory
- 1024Bytes EEPROM
- 2Kbytes Internal SRAM
- Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
- Data retention: 20 years at 85°C/100 years at 25°C(1)
- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program.

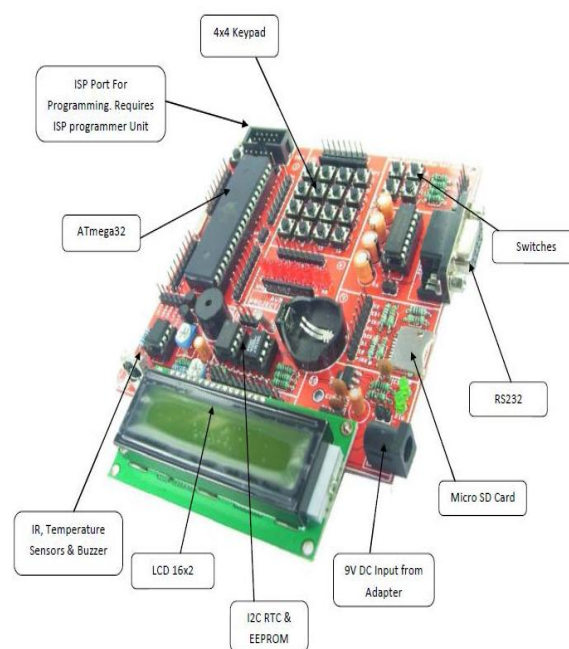


Fig. 3. Atmega32A development board

C. USB to Serial converter

Data communication is a general technique for two endpoints to communicate with each other. Data is sent from one endpoint and received by the other endpoint and vice versa. Serial communication describes the scenario where the data bits are transmitted sequentially (one bit at a time). Since present day laptops don't have serial pin, a USB to Serial converter is used.

D. AVR-ISP programmer

The in-system programming (ISP) programming method is functionally performed through SPI, plus some twiddling of the Reset line. As long as the SPI pins of the AVR are not connected to anything disruptive, the AVR chip can stay soldered on a PCB while reprogramming. All that is needed is a 6-pin connector and programming adapter. This is the most common way to develop with an AVR. The Atmel AVR ISP mkII device connects to a computer's USB port and performs in-system programming using Atmel's software.

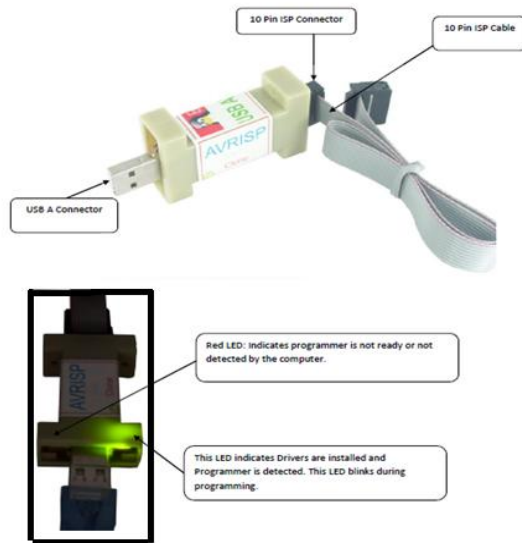


Fig. 5. AVR-ISP programmer

IV. SOFTWARES USED

The softwares used in development of the projects are 1) AVR Studio5.1. This is an IDE used to develop the program using C. The code is compiled and debugged here. Once the code is compiled without error a hex file is generated which is loaded into the microcontroller. 2) AVR-ISP mkII clone software which is Graphical User Interface that is used to program the microcontroller. It works along with the AVR-ISP programmer. 3) Hyperterminal, which is a software that is used as a display unit. It enables serial communication between PC and the microcontroller. The data sent from the microcontroller is displayed on the screen.

V. ALGORITHM DEVELOPMENT

Two algorithms are used in this project. One is to recognise the hand gesture and other is to program the microcontroller.

a. Algorithm for Gesture Recognition

The algorithm used for Gesture recognition is developed based on Sign Sequence and Template Matching Algorithm.

The Gestures we are going to recognize are left, right, front and back movement of the arm. The acceleration and its sign are previously measured for the gestures in all three axes are stored as a look up table. The look up table is given. Now, whenever the arm is moved, the acceleration values of the three axes are found and compared with the look up table.

Based on the comparison results the hand gesture can be recognized. This method can be used for non-specific users.

Fig.9.Look up table for hand gesture recognition

b. Algorithm for Programming the microcontroller

- Step 1:- Get the values from the accelerometer and convert it into digital values for all the three axes and store it in a variable.
- Step 2:- From those values find the acceleration and its sign for all three axes.
- Step 3:- Compare the values of the acceleration with the look up table to recognize the hand gesture.
- Step 4:- Display the output to a PC monitor using serial communication.

c. Calibration of the accelerometer

The three axis accelerometer actually measures the acceleration at three different axes X, Y and Z. Those acceleration values are used to recognise the hand gesture.

$$\text{acc}_x = (\text{adc}_x/1024.0*5.0-1.586)/0.3$$

$$\text{acc}_y = (\text{adc}_y/1024.0*5.0-1.586)/0.3$$

Gestures	X	Y	Z
Normal	X-normal	Y-normal	Z-normal
Left	X-left	Y-left	Z-left
Right	X-right	Y-right	Z-right
Front	X-front	Y-front	Z-front
Back	X-back	Y-back	Z-back

Table.1.Lookup table

VI. RESULT ANALYSIS

It was found that using acceleration alone to recognise the hand gesture is quite difficult. So, the angle between any two acceleration values is used for hand gesture recognition. The angle between accelerations of any two axis is found by the following formula.

$$\text{angle}_{xy} = \text{atan2}(-\text{acc}_y, -\text{acc}_x) * 57.2957795 + 180$$

$\text{acc}_x \Rightarrow$ acceleration at X axis

$\text{acc}_y \Rightarrow$ acceleration at Y axis

$\text{angle}_{xy} \Rightarrow$ angle between acc_x and acc_y

It was also found that using only two axis i.e. x and y we can only find the hand gesture only in that axis. The angle between those axes is used to recognise hand gesture in those axes. For recognising other gestures the values of acceleration of x and z axes should be used. Combining both the values we can also find the gestures such as bottom-left, bottom-right, top-left, top-right. Total of eight gestures can be identified using these values. The hand gesture recognition system shown in below figures.

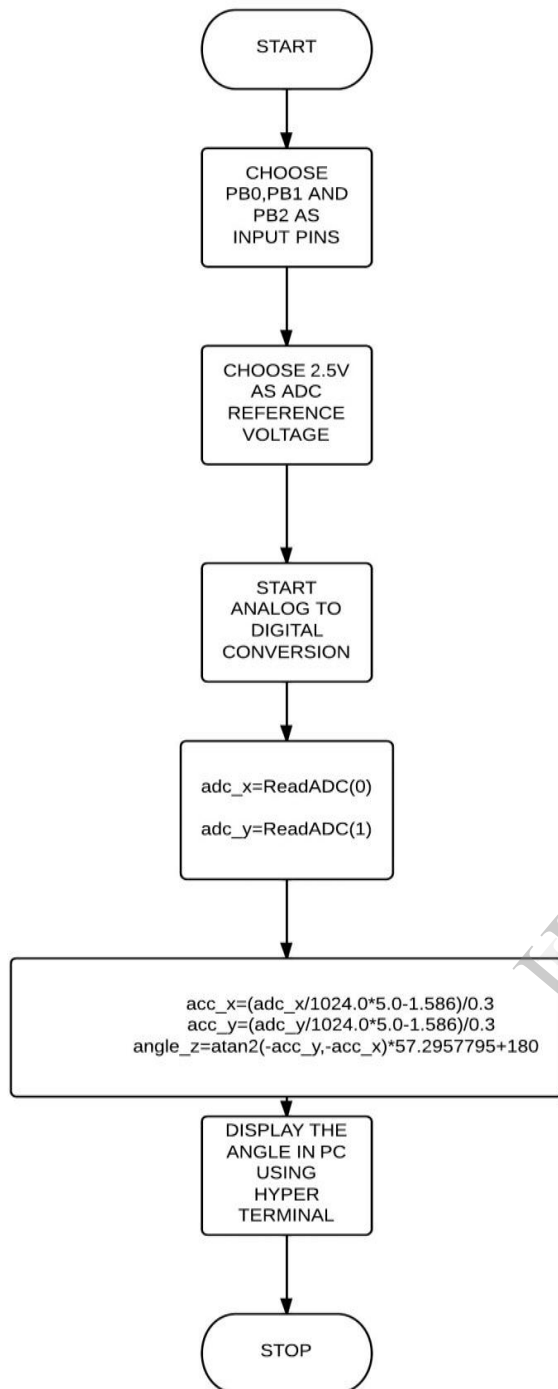


Fig.10.Flow chart of the program

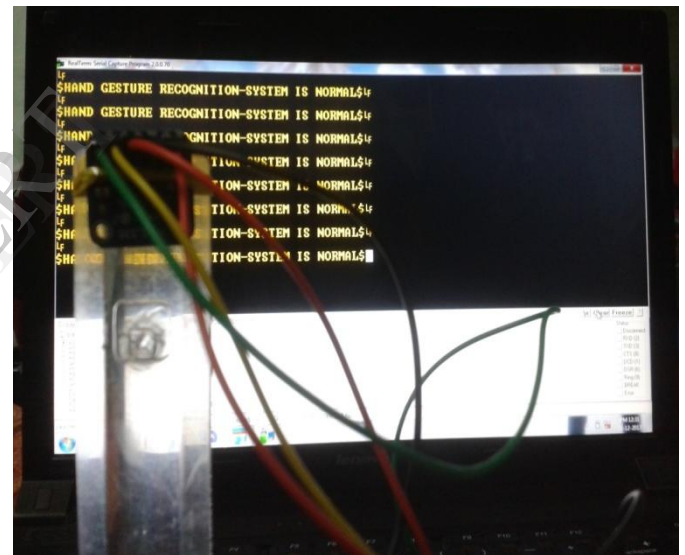


Fig.11.Hand gesture recognition system is normal

Now we can see the position of the hand is normal condition. by using this we can implement in many robotics applications. For example we can control the car without steering to move all the direction which is convenient for the movement by using real time hand gesture recognition. Now we can see all the position of the hand in the display unit in below figures.

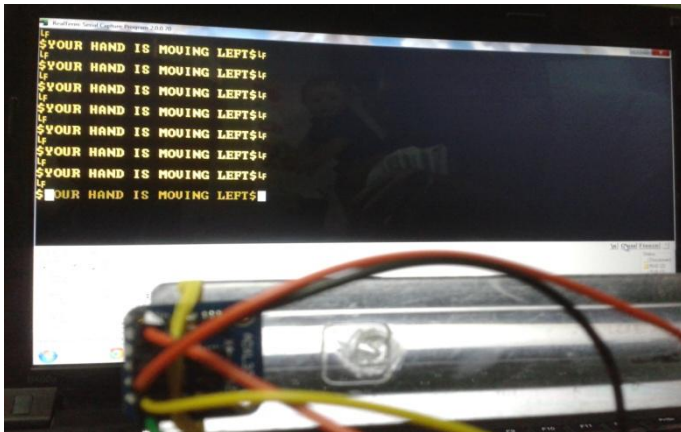


Fig.12.Hand is moving left



Fig.15.Hand is moving front

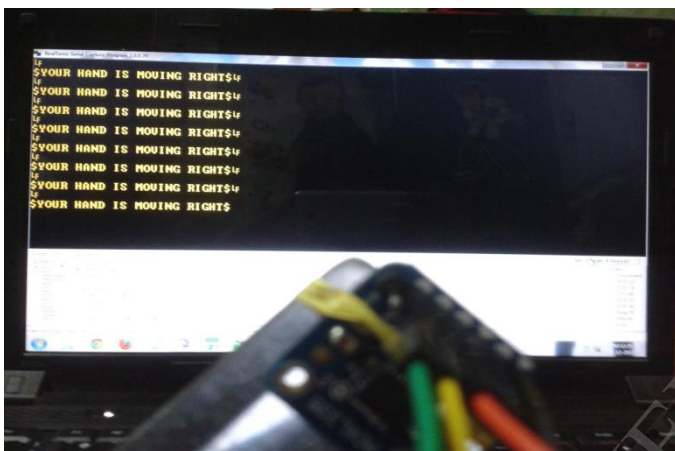


Fig.13.Hand is moving right



Fig.16.Hand is moving front-right

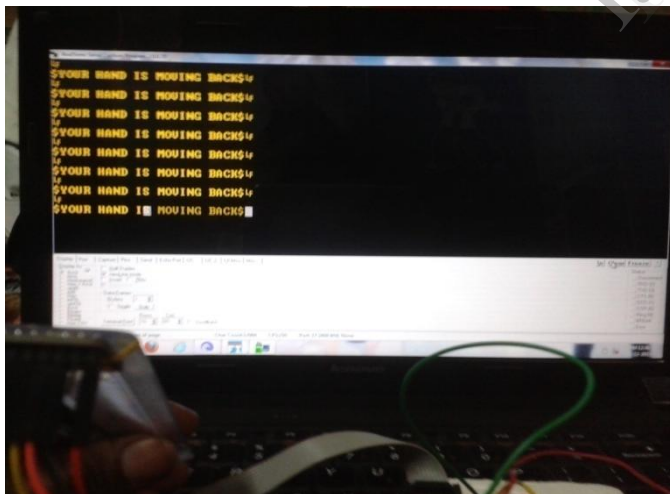


Fig.14.Hand is moving back

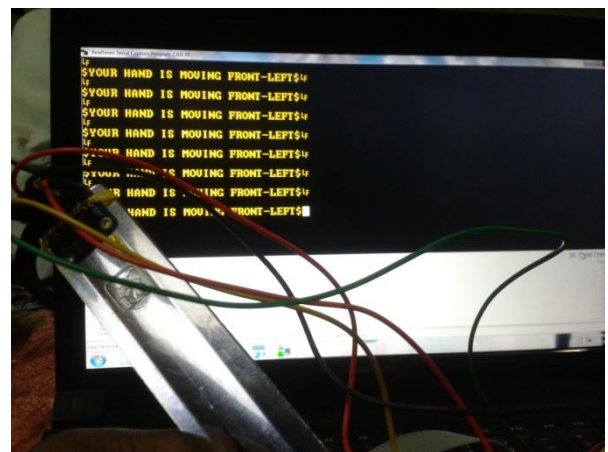


Fig.17.Hand is moving front-left

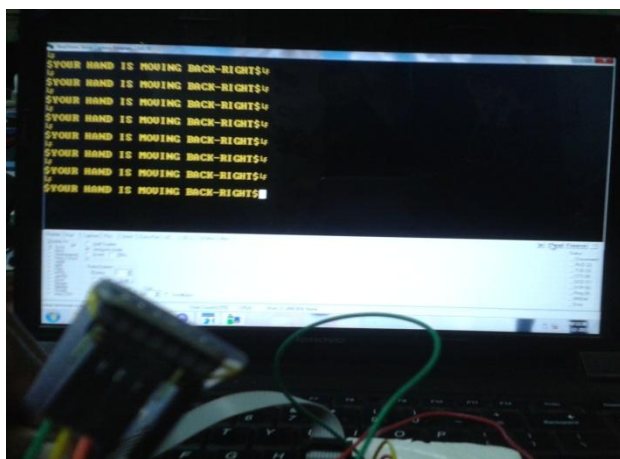


Fig.18.Hand is moving back-right

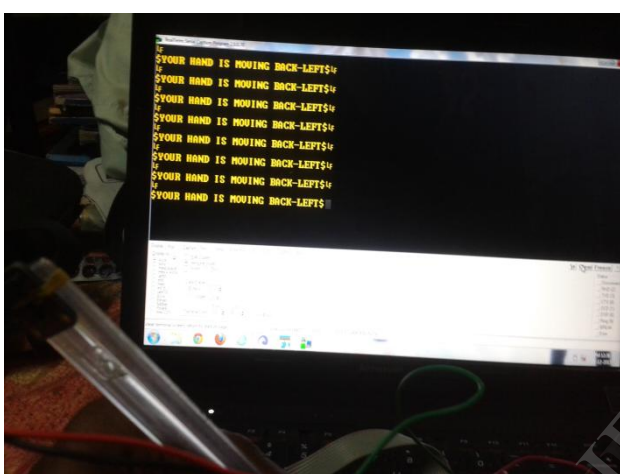


Fig.19.Hand is moving back-left

VII. CONCLUSION

Gestures as an input modality can provide efficient and flexible interaction between human and computer. Gesture control has few advantages compared with more traditional modalities: gestures are silent, they require no eye focus and, for certain tasks, they can feel more natural than, for instance, pressing a key on the keyboard. Inspired by the proliferation of MEMS accelerometer sensors built-in personal electronic devices, such as smartphones, music players and game consoles, this paper presented a novel gesture recognition system based solely on accelerometer data. In this paper the acceleration data is utilized to recognize the hand gestures. The algorithm used for gesture recognition is developed based on Sign Sequence and Template Matching Algorithm. Up to eight hand gestures are recognized in this project. In future work the hand gesture recognition system is to be used in Human-Machine interface such as controlling a flying machine using the hand gesture etc.

REFERENCES

- [1] Nasser H. Dardas and Nicolas D. Georganas, "Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques", IEEE transactions on instrumentation and measurement, vol. 60, no. 11, November 2011.
- [2] L. Yun and Z. Peng, "An automatic hand gesture recognition system based on Viola-Jones method and SVMs," in Proc. 2nd Int. Workshop Comput. Sci. Eng., 2009, pp. 72–76.
- [3] W. Chung, X. Wu, and Y. Xu, "A real time hand gesture recognition based on Haar wavelet representation," in Proc. IEEE Int. Conf. Robot. Biomimetics, 2009, pp. 336–341.
- [4] L. Juan and O. Gwon, "A comparison of SIFT, PCA-SIFT and SURF," Int. J. Image Process. (IJIP), vol. 3, no. 4, pp. 143–152, 2009.
- [5] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in Proc. IEEE ICCV, 2009, pp. 492–497.
- [6] Y. Ren and C. Gu, "Real-time hand gesture recognition based on vision," in Proc. Edutainment, 2010, pp. 468–475.
- [7] N. Dardas, Q. Chen, N. Georganas, and E. Petriu, "Handgesture recognition using bag-of-features and multi-class support vector machine," in Proc. 9th IEEE Int. Workshop HAVE, Phoenix, AZ, Oct. 16–17, 2010.
- [8] L. Shi, Y. Wang, and J. Li, "A real time vision-based hand gestures recognition system," in Proc. 5th ISICA, 2010, pp. 349–358.
- [9] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in Proc. IEEE Conf. CVPR, 2009, pp. 2929–2936.