Middle-East Journal of Scientific Research 20 (10): 1247-1251, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.20.10.114113

# Character Comparsion Using Gyroscope for Visible Challenged People

*S. Arulselvi*

Department of ECE,
Bharath University, Chennai, India

**Abstract:** In this paper we are going to design an embedded system with the help of human logic. This embedded system is about designing a creative pen-style hardware and analysis software for the recognition of simple air written characters wirelessly. Motion through air is converted as a set of points in space and to recognize a character a unique software algorithm verifies the pattern of motion points against a stored form and a display shows the character recognized in the process. In this project, the hardware and software system will be designed to recognize 10 numerals at a very high recognition rates, between 90-100%. Use of Gyroscope is used for the motion recognition and essential coding form the unique embedded system for specific purpose, especially for visible challenged people.

**Key words:** Embedded system · Gyroscope · Motion recognition

## INTRODUCTION

In any mobile phone, to write, people have two clear choices at the moment-they either use a keypad or a touch screen stylus. How about creating a third option that allows people to write in air? The concept is simple. The entire phone should be moved through the air to write. So making a "3" shape in the air sees that letter appear on the screen. The idea has been made possible by the growing use of accelerometers inside devices that can track the movement being carried out with the device-something used so effectively in the Nintendo and the iPhone. In fact the project uses the same accelerometer LIS302DL that was used in highly successful "Apple iPhone". In those days, many people get discouraged with current phones and their small keys. As phones get smaller, this frustration will only grow.

The project is about designing a creative pen-style hardware and analysis software for the recognition of simple air written characters wirelessly.

The hardware has a 3-dimensional acceleration sensor, a microcontroller with I2C serial communication port and does not need any touching screen or keypad. Motion through air is converted as a set of points in space and to recognize a character a unique software algorithm verifies the pattern of motion points against a stored form and a display shows the character recognized in the process. In this project, the hardware and software system will be designed to recognize 10 numerals at a very high recognition rates, between 90-100%. Although the database is quite small, it clearly demonstrates the usefulness of the acceleration-based air-written character recognition system without any touch screen or keypad.

The project creates a useful application with this. Once you typed a number in air, just make a simple tap. The system recognizes this and alerts you after the specified number of minutes through a vibrating motor like the one used in mobile phones. And data will be transmitted wireless with current accelerometers the system can only recognize a single character at a time with a pause before starting the next one. As accelerometer technology advances joined letter writing should become possible. Once the data is received from transmitter wireless the recognized number will be seen in HyperTerminal window in PC (Personal Computer).

**Related Works:** Akl and S. Valaee 2010 [1] gives the details about an accelerometer-based gesture recognition system that uses only a single 3-axis accelerometer to recognize gestures, where gestures here are hand movements. The work of this paper is built upon a preliminary version of our gesture recognition system. A dictionary of 18 gestures is created for which a database of 3780 traces is built by collecting data from 7 participants. Some of the gestures defined in the dictionary are taken from the gesture vocabulary identified by Nokia.

**Corresponding Author:** S. Arulselvi, Department of ECE, Bharath University, Chennai, India.

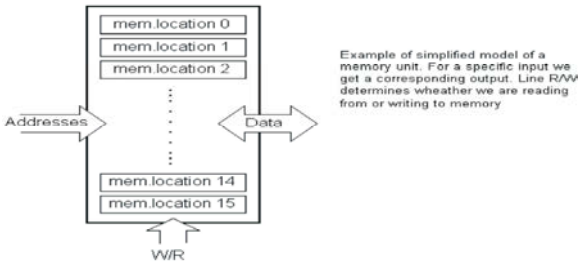
J. Liu, L. Zhong, J. Wickramasuriya 2009 [2] this paper gives the most recent gesture recognition system that is solely accelerometer-based is the uWave. uWave is a user-dependent system that supports personalized gesture recognition. uWave functions by utilizing only one training sample, stored in a template, for each gesture pattern. The core of th e uWave is dynamic time warping (DTW) and the system's database undergoes two types of adaptation: positive and negative adaptation. However, uWave's database adaptation resembles continuous training and in some cases, if thorough examination of templates is ignored, removing an older template every other day might lead to replacing a very good representative of a gesture sequence, which is best avoided.

T. Pylvänäinen 2005 [3] paper presents the Accelerometer-based gesture recognition system using continuous hidden Markov models (HMMs) has been developed. However, the computational complexity of statistical or generative models like HMMs is directly proportional to the number as well as the dimension of the feature vectors. Therefore, one of the major challenges with HMMs is estimating the optimal number of states and thus determining the probability functions associated with the HMM. Besides, variations in gestures are not necessarily Gaussian and perhaps, other formulations may turn out to be a better fit.

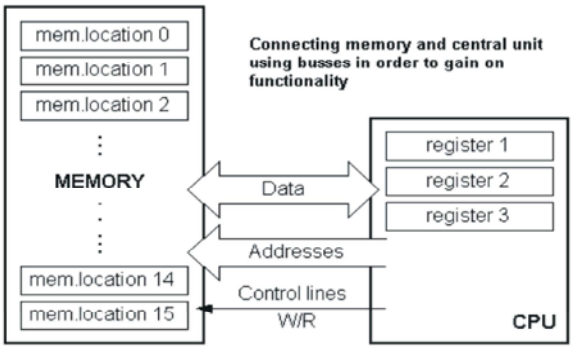
**System Architecture Design**

**Pic Microcontroller (16F877A):** A general definition of embedded systems is: embedded systems are computing systems with tightly coupled hardware and software integration, which are designed to perform a dedicated function. In some cases, embedded systems can function as standalone systems. One class of embedded processors focuses on size, power consumption and price. Therefore, some embedded processors are limited in functionality, i.e., a processor is good enough for the class of applications for which it was designed but is likely inadequate for other classes of applications. Real-time systems are defined as those systems in which the overall correctness of the system depends on both the functional correctness and the timing correctness. The timing correctness is at least as important as the functional correctness.

**Memory Unit:** Memory is part of the microcontroller whose function is to store data. For a certain input we get the contents of a certain addressed memory location and that's all. Two new concepts are brought to us: addressing and memory location. Memory consists of all memory locations and addressing is nothing but selecting one of them. This means that we need to select the desired memory location on one hand and on the other hand we need to wait for the contents of that location. Besides reading from a memory location, memory must also provide for writing onto it. This is done by supplying an additional line called control line. We will designate this line as R/W (read/write). Control line is used in the following way: if r/w=1, reading is done and if opposite is true then writing is done on the memory location.




Block diagram of memory unit




**Central Processing Unit:** Let add 3 more memory locations to a specific block that will have a built in capability to multiply, divide, subtract and move its contents from one memory location onto another. The part we just added in is called "central processing unit" (CPU). Its memory locations are called registers.

Registers are therefore memory locations whose role is to help with performing various mathematical operations or any other operations with data wherever data can be found. Look at the current situation. We have two independent entities (memory and CPU) which are interconnected and thus any exchange of data is hindered, as well as its functionality.

**Bus:** That "way" is called "bus". Physically, it represents a group of 8, 16, or more wires. There are two types of buses: address and data bus. The first one consists of as many lines as the amount of memory we wish to address and the other one is as wide as data, in our case 8 bits or

the connection line. First one serves to transmit address from CPU memory and the second to connect all blocks inside the microcontroller.

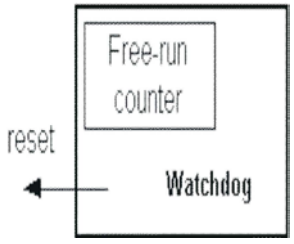
**Representation of Bus**

**Input-Output Unit:** Those locations we've just added are called "ports". There are several types of ports: input, output or bidirectional ports. When working with ports, first of all it is necessary to choose which port we need to work with and then to send data to, or take it from the port. When working with it the port acts like a memory location. Something is simply being written into or read from it and it could be noticed on the pins of the microcontroller.

**Serial Communication:** As we have separate lines for receiving and sending, it is possible to receive and send data (info.) at the same time. So called full-duplex mode block which enables this way of communication is called a serial communication block. Unlike the parallel transmission, data moves here bit by bit, or in a series of bits what defines the term serial communication comes from. After the reception of data we need to read it from the receiving location and store it in memory as opposed to sending where the process is reversed. In order for this to work, we need to set the rules of exchange of data. These rules are called protocol. Data goes from memory through the bus to the sending location and then to the receiving unit according to the protocol.

**Timer Unit:** The timer block this can give us information about time, duration, protocol etc. The basic unit of the timer is a free-run counter which is in fact a register whose numeric value increments by one in even intervals, so that by taking its value during periods T1 and T2 and on the basis of their difference we can determine how much time has elapsed. This is a very important part of the microcontroller whose understanding requires most of our time.

**Watchdog:** One more thing is requiring our attention is a flawless functioning of the microcontroller during its run-time.

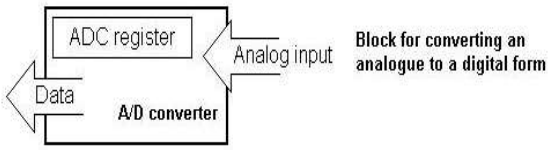



Watch dog time

Suppose that as a result of some interference (which often does occur in industry) our microcontroller stops executing the program, or worse, it starts working incorrectly. Of course, when this happens with a computer, we simply reset it and it will keep working. However, there is no reset button we can push on the microcontroller and thus solve our problem. To overcome this obstacle, we need to introduce one more block called watchdog.

This block is in fact another free-run counter where our program needs to write a zero in every time it executes correctly. In case that program gets "stuck", zero will not be written in and counter alone will reset the microcontroller upon achieving its maximum value. This will result in executing the program again and correctly this time around.

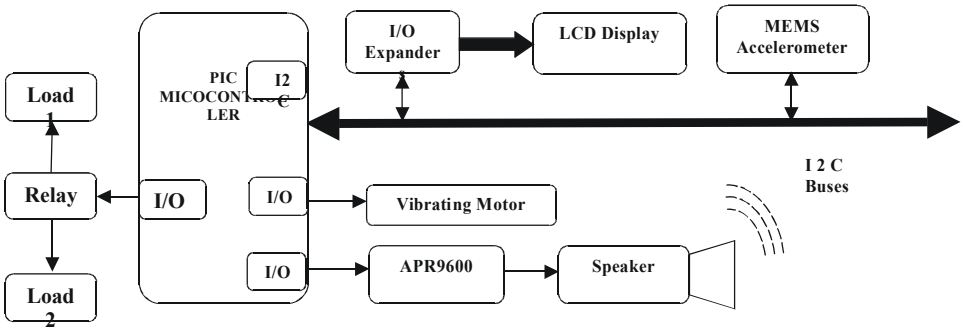
**Analog to Digital Converter:** As the peripheral signals usually are substantially different from the ones that microcontroller can understand (zero and one), they have to be converted into a pattern which can be comprehended by a microcontroller.




**Analog to Digital Converter:** This task is performed by a block for analog to digital conversion or by an ADC. This block is responsible for converting an information about some analog value to a binary number and for follow it through to a CPU block so that CPU block can further process it.

**MEMS Accelerometer:** An accelerometer is a device for measuring acceleration and gravity induced reaction forces. Single-and multi-axis models are available to detect magnitude and direction of the acceleration as a vector quantity. Accelerometers can be used to sense inclination, vibration and shock. They are increasingly present in portable electronic devices.

Modern accelerometers are often small micro electro-mechanical systems (MEMS) and are indeed the simplest MEMS devices possible, consisting of little more than a cantilever beam with a proof mass (also known as seismic mass). Mechanically the accelerometer behaves as a mass-damper-spring system; the damping results from the residual gas sealed in the device. As long as the Q-factor is not too low, damping does not result in a lower sensitivity.




Under the influence of gravity or acceleration the proof mass deflects from its neutral position. This deflection is measured in an analog or digital manner. Most commonly the capacitance between a set of fixed beams and a set of beams attached to the proof mass is measured. This method is simple and reliable; it also does not require additional process steps making it inexpensive. Integrating piezo resistors in the springs to detect spring deformation and thus deflection, is a good alternative, although a few more process is needed. For very high sensitivities quantum tunneling is also used; this requires specific fabrication steps making it more expensive. Optical measurement has been demonstrated on laboratory scale.

Another, far less common, type of MEMS-based accelerometer contains a small heater at the bottom of a very small dome, which heats the air inside the dome to cause it to rise. A thermocouple on the dome determines where the heated air reaches the dome and the deflection off the center is a measure of the acceleration applied to the sensor.Most micromechanical accelerometers operate in-plane, that is, they are designed to be sensitive only to a direction in the plane of the die. By integrating two devices perpendicularly on a single die a two-axis accelerometer can be made. By adding an additional out-of-plane device three axes can be measured. Such a combination always has a much lower misalignment error than three discrete models combined after packaging. Micromechanical accelerometers are available in a wide variety of measuring ranges, reaching up to thousands of g's. The designer must make a compromise between sensitivity and the maximal acceleration that can be measured.

**MEMS Accelerometer LIS302DL:** The LIS302DL is an ultra compact low-power three axes linear accelerometer. It includes a sensing element and an IC interface able to provide the measured acceleration to the external world through I2C/SPI serial interface. The sensing element, capable of detecting the acceleration, is manufactured using a dedicated process developed by ST to produce inertial sensors and actuators in silicon. The IC interface is manufactured using a CMOS process that allows to design a dedicated circuit which is trimmed to better match the sensing element characteristics. The LIS302DL has dynamically user selectable full scales of ±2g/±8g and it is capable of measuring accelerations with an output data rate of 100Hz or 400Hz. A self-test capability allows the user to check the functioning of the sensor in the final application [4]. The device may be configured to generate inertial wake-up/free-fall interrupt signals when a programmable acceleration threshold is crossed at least in one of the three axes. Thresholds and timing of interrupt generators are completely programmable by the end user on the fly. The LIS302DL is available in plastic Thin Land Grid Array package (TLGA) and it is guaranteed to operate over an extended temperature range from-40°C to +85°C. The LIS302DL belongs to a family of products suitable for a variety of applications:

- Free-Fall detection
- Motion activated functions
- Gaming and Virtual Reality input devices
- Vibration Monitoring and Compensation

**Feature:**

- Three axes
- SPI/I2C digital interface
- Innovative embedded functionalities
- Two highly flexible and programmable interrupt request outputs
- Programmable thresholds and timing of interrupt signals

**Software Design:** In this project we are using Embedded C as a programming language. The hardware tools are simulated by MPLAB IDE (Integrated Development Environment). It converts Embedded C language into opcode formation for the microcontroller (PICKIT2). A

source code editor is a text editor program designed specifically for editing source code of computer programs by programmers. A compiler is a computer program that transforms source code written in a programming language into another computer language. The most common reason for wanting to transform source code is to create an executable program.

A special program used to find errors (bugs) in other programs. A debugger allows a programmer to stop a program at any point and examine and change the values of variables. MPLAB IDE is a free, integrated toolset for the development of embedded applications employing Microchip's PIC® and dsPIC® microcontrollers. MPLAB IDE runs as a 32-bit application on MS Windows. MPLAB is easy to use and includes a host of free software components for fast application development and super-charged debugging. MPLAB IDE also serves as a single, unified graphical user interface for additional Microchip and third party software and hardware development tools. MPLAB IDE is an integrated toolset for the development of embedded applications employing Microchip's PIC microcontrollers. The MPLAB IDE runs as a 32-bit application on Microsoft Windows. Both Assembly and C programming languages can be used with MPLAB IDE.

## CONCLUSION

In this PIC based on intelligent control system of the embedded system for visually handicapped people, in this system we are using gyroscope with plays an important role in recognising the characters through the motion of the hand and also this analog value is sent to the PIC microcontroller where it is converted to digital format. The digital format compares with the code which is damped in the controller and if the comparison is satisfied (90-100%) the value is sent to the speaker and which gives us the output. The future enhancement of the project can be upgrading the hardware say PIC microcontroller can be upgraded to ARM11 microcontroller and also specific purpose MEMS sensor can be used to particular application.

## REFERENCES

1. Yang, A., S. Iyengar, S. Sastry, R. Bajcsy, P. Kuryloski and R. Jafari, 2008. Distributed segmentation and classification of human actions using a wearable motion sensor network, in Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. Workshops (CVPRW '08), Jun, pp: 1-8.
2. Zhang, X., X. Chen, W.W. hui, J.Y. hai, V. Lantz and K.W. Qiao, 2009. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors, in Proc. 13th Int. Conf. Intell. UserInterfaces (IUI '09), New York, pp: 401-406.
3. Pylvänäinen, T., 2005. Accelerometer Based Gesture Recognition Using Continuous HMMs, in Pattern Recognition and Image Analysis. New York: Springer Berlin/Heidelberg, pp: 639-646.
4. Liu, J., L. Zhong, J. Wickramasuriya and V. Vasudevan, 2009. U Wave: Accelerometer-based personalized gesture recognition and its applications, Pervasive Mobile Comput., 5(6): 657-675.