MOTION CONTROLLED PASSWORD RECOGNITION USING MEMS ACCELEROMETER

S.Malathi[1], S.Naseema[2], P.Pavithra[3], A.Pradeepa[4]
[1]Assistant professor, Department of Electronics and Communication Engineering, T.J.S Engineering College, Peruvoyal, Chennai
[2, 3,4] UG Scholar, Department of Electronics and Communication Engineering, T.J.S Engineering College, Peruvoyal, Chennai

malathi.nec@gmail.com
fathimadawod555@gmail.com
pavithrapannerselvamr@gmail.com
deepa29anandh@gmail.com

ABSTRACT

In this work relates to security mechanisms for compact electronic devices and prevents un authorized access to electronic devices such as ATM machine, laptop computers, Cellular telephones, and the like which can be activated and deactivated simply by moving the device correctly according to a preselected "password" . A preselected password is easy to remember when compared to traditional passwords and Personal Identification Numbers (PINs). They may sometimes be forgotten and are not always confidential. The chip set includes a MEMS sensor which could be integrated within a device such as a laptop computer, a ARM LPC2148 microcontroller and Visual Basic where in the database of memory includes a set of coordinate values corresponding to authorized movement of the computer. Also includes Buzzer for indication of system and misuse.

I.INTRODUCTION

Motion controlled password recognition system presents an advanced authentication based on motion of the system.This may be integrated as a chipset attached to or otherwise integrated with the devices which may be electronic devices such as laptop computer, PDA's, a cellular phone or the like, or even other devices as subjected to theft and/or miss use such as fire arm. Unlike most of the previous authentication like biometric, cards to swipe it's entirely different and more reliable. Using machine motion techniques, in our approach logical patterns are obtained from a physical sensor attached to the system.

The chip set includes accelerometer or some other type of motion sensor responsive to movement of device and which provides signal corresponding to the movement of device (to processor or microcontroller or any other electronic circuitry), stored in memory, EEPROM which may be data base of values corresponding to predetermined motion or orientation electronic device. In operation, when accelerometer detects the motion of device and signals the processor, processor

access the data base and one of the two things happen: either the movement if the device corresponds to the values stored as motion password; or the movement of the device don't corresponds to motion password stored. Acceleration in 2 directions: up-down, forward backward of the system movement, which is obtained by the sensor, is used for authentication.

II. GENERAL BLOCK DIAGRAM OF OF MOTION CONTROLLED PASSWORD SYSTEM

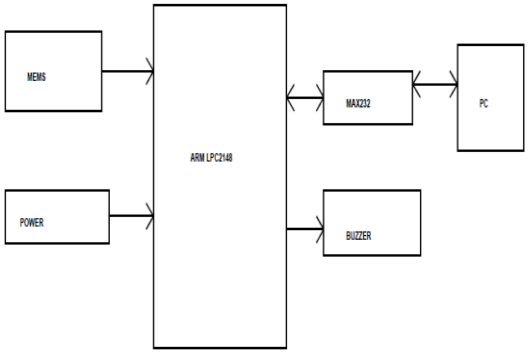

Fig.1 Block diagram of motion controlled password system

In the above circuit diagram the MEMS sensor is connected with analog to digital pin of ARM LPC2148 controller and it has 12 MHz oscillator with reset circuit.Since the MEMS sensor contains two axis it will give two analog signal with respect to the change in axis that is when changes in X axis changes then analog voltage in X-axis pin changes and that same way when Y axis changes the analog voltage in Y-axis will change. Internally the ARM LPC2148 contains two analog to digital converter which converts analog data into ten bit digital value.

Controlling ADCis all about manipulating ADC control register(AD0CR/AD1CR) and global data register(AD0GDR/AD1GDR).

For any adc to function properly,clk should be supplied for conversion and this clk must be stable.Adc in lpc2148 need clk frequency<=4.5MHZ.Burst mode is 0 for software controlled conversion and 1 hardware controlled conversion (using capture and match register).we prefer to use software controlled.

The most important register in adc is control register is AD0CR/AD1CR. Other than adc pin selection part we use to load AD0CR=0x00200600|(1<<ch);or AD1CR=0x00200600|(1<<ch);
Then start the conversion by setting appropriate bit…AD0CR=1<<24; or AD1CR=1<<24;

Wait untill the conversion is over checking done bit. while((AD1GDR&(1U<<31))==0); (or) while((AD0GDR&(1U<<31))==0);

Since result is stored in bit 6 to 15,we need to copy the register value to a variable and right shift the converted value and store it.Pclk/(1+clkdiv)<=4.5MHZ.Then the converted data will send to the PC via serial communication (UART) universal asynchronous receiver transmitter.The information is transmitted one binary bit at a time; as such it is a serial communication

method. These bits are grouped together in the form of '**Frames**' (a set format) for conveying one meaningful piece of data (e.g. character byte). UART is asynchronous because it doesn't require a transmitter provide clockto synchronize the transmission and receipt of data.Just because there is no clock signal per se, a start bit is added sent first to tell the receiver to listen out for data. The receiver monitors for a logic HIGH falling to logic LOW. The receiver synchronizes its own bus clock to that make up the word being sent, with bit zero, the least significant bit (LSB) being sent first. The bits are sent as pulses on the wire at specific time intervals, set at both ends of links to previously agreed values. The receiver looks at the voltage on the wire at these times; if it sees logic high, it records a binary digit 1 or 0 if the line is low. The receiver checks half way between the start and the end of the pulse to ensure it does not miss-read the voltage on the line during the brief interval while the voltage is rising or falling.LPC2148 ARM7 core supports two UART in it, UART0 and UART1. UART0 can be used as general purpose UART and also can support ISP Programming through it, whereas UART1 has additional modem support. Both have built in baud rate generator and 16-byte transmit and receive FIFOs. For UART0 the TxD Pin is at P0.0 and RxD Pin is at P0.1 and similarly for UART1 the TxD Pin is at P0.8 and RxD Pin is at P0.9

Most of microchips work on TTL or CMOS voltage level which can't be used to communicate over RS-232 protocol. In this case voltage or level converter is needed which can convert TTL to RS-232 and RS-232 to TTL voltage levels. The most commonly used RS-232 level converter is MAX3232 chip. This chip includes charge pump which can generate RS232 voltage levels (-10V and +10V) from 5V power supply. It also includes two receiver and two transmitters and is capable of full-duplex UART communication. RS232 communication enables point-to-point data transfer, which often used in data acquisition application and for data transfer between microcontroller and PC. The following table shows the registers need for serial communication in ARM core

III. GESTURE MOTION ANALYSIS

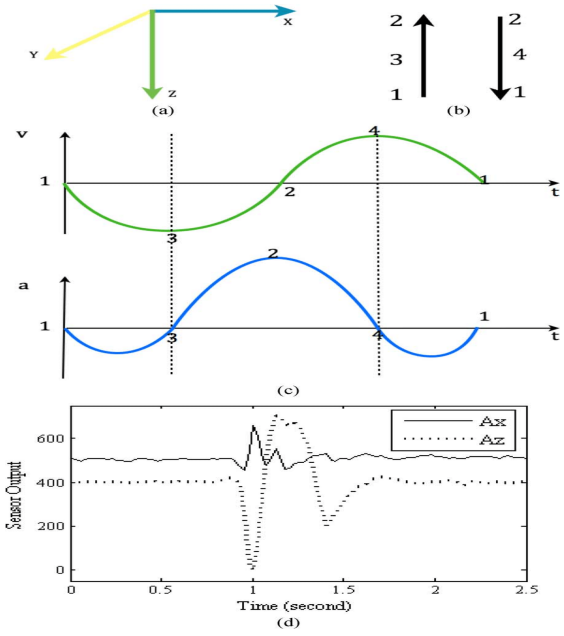

Fig.2 Gesture up motion analysis. (a) Coordinate system. (b) Gesture up motion decomposition. (c) Predicted velocity and acceleration in the z-axis of the gesture up. (d) Real acceleration plot of the gesture up. Solid

and dotted lines are accelerations on x- and z-axis, respectively.

Gesture motions are in the vertical plane, so the accelerations on x- and z- axes are adequate to distinguish each gesture. Therefore, the acceleration on y-axis is neglected to reduce computational requirement. We propose that the exact shape of the acceleration curves is not critical, but only the alternate sign changes of acceleration on the two axes are required to uniquely differentiate any one of the 7 gestures: up, down, left, right, tick, circle, and cross. This is the basis of the recognition algorithms discussed in this paper. For instance, the gesture up has the acceleration on z-axis in the order: negative —positive— negative (positive z direction points downward) and nearly has no acceleration on x-axis; for a circle gesture, on x axis: positive-negative-positive and on z-axis: negative-positive-negative-positive. Experiments showed that each of these gestures hasa special order of sign changes, and a kinematics analysis also proves this. A kinematic motion a hand goes through in performing a gesture could nonintuitive at time. For example, a simple up gesture can be decomposed into several acceleration and deceleration periods. As shown in Fig. 1(b), an up gesture is actual consist of motion from point 1 to point 2, and then back to point 1. The velocity at the starting point 1, midpoint 2 and end point 1 are all zeros. For the convenience of analysis, point 3 is the point between point 1 and point 2 where acceleration changes sign, and point 4 is the point between point 2 and point 1 where acceleration changes sign. Then the acceleration changes can be described as: 1 3: acceleration on z-axis is negative (since positive z direction is downward); velocity changes from zero to a maximum value at 3; acceleration at point 3 is zero. 3 4: acceleration on z-axis is positive; velocity changes from negative to positive and is maximum at point 4, where acceleration becomes zero. 4 1: acceleration on z-axis is negative; velocity changes from positive to zero. Also, acceleration and velocity become zero at point 1. The analysis above is illustrated by Fig. 1(c). Fig. 1(d) is the real acceleration plot for the gesture upin which the dotted line is the acceleration on z-axis and solid line is the acceleration on x-axis. From Fig. 1(d), we note that noise exists from sensor measured data. However, the noise does not influence the trend of the acceleration curves, and hence, the analysis of gestures based on the above method still works without adding computational burdens on a CPU by using a noise-filtering algorithm. Comparing the predicted acceleration pattern in Fig. 1(c) with the real acceleration plot in Fig. 1(d), it is concluded that the trend of the real acceleration is the same with the prediction. After analyzing the other gestures, it was found that they all have unique acceleration patterns for classification. Gesture down is similar to up but with changes in directions, left and rightand also similar, but the changes in motion axes information. Tick, circle and cross are more complex since they have accelerations on both x- and z- axes

simultaneously, but the accelerations on the two axes can be separated and decomposed, then the motion trend becomes similar to the above example. The uniqueness of each gesture trend makes the recognition algorithm possible, and the algorithms presented in this paper are based on this basic motion feature of the seven gestures.

IV. SENSING SYSTEM OVERVIEW

A. Sensor Description

The sensing system utilized in our experiments for hand motion data collection is shown in Fig. 2 and is essentially a MEMS 3-axes acceleration sensing chip integrated with data management and Bluetooth wireless data chips. The algorithms described in this paper were implemented and run on a PC. Details of the hardware architecture of this sensing system were published by our group in [19] and [20]. The sensing system has also been commercialized in a more compact form recently [21].

B. System Work Flow

When the sensing system is switched on, the accelerations in three perpendicular directions are detected by the MEMS sensors and transmitted to a PC via Bluetooth protocol.

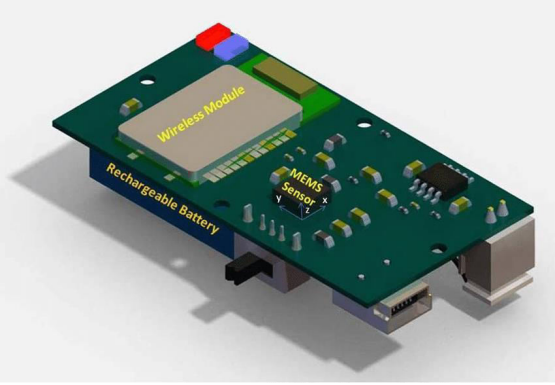

Fig. 3. Illustration of the components of the sensing system used for hand gesture recognition (the device shown as dimensions of(l=6cm,w=4.5cm,h=2.5cmcm).

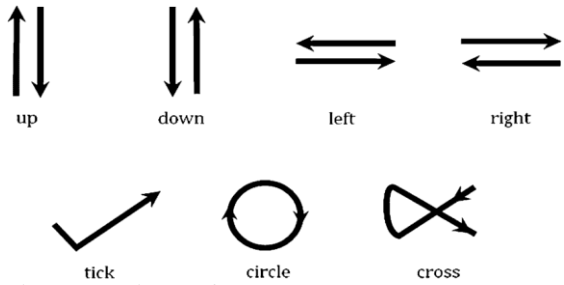

Fig. 4. Motions of seven gestures.

IV. GESTURE SEGMENTATION A. Data Acquisition To collect reliable hand gesture data for the sensing system, the experimental subject should follow guidelines below during the data acquisition stage: • The sensing devices hould be held horizontally during the whole data collection process (i.e., the x-y plane of the sensor chip in Fig. 2 pointing towards the ground). • The time interval between two gestures should be no less than 0.2 seconds so that the segmentation program can separate each one of the gestures in sequential order. • The gestures should be

performed as indicated in Fig. 4. B. Gesture Segmentation 1) Data Preprocessing: Raw data received from the sensors are preprocessed by two 2 processes: a) vertical axis offsets are removed in the time-sequenced data by subtracting each data points from the mean value of a data set; hence, a data set shows zero value on the vertical axes when no acceleration is applied; b) a filter is applied to the data sets to eliminate high-frequency noise data.

2) Segmentation: The purpose of the segmentation algorithm2) Segmentation: The purpose of the segmentation algorithm is to find the terminal points of each gesture in a data set of gesture sequence. The algorithm checks various conditions of all the data points and picks out the most likely data points as the gesture termination points. The conditions of determining the gesture terminal points in our algorithm are a) amplitude of the points (- coordinate value of a data point); b) point separation (the difference between the x-coordinates of the two points); c) mean value (mean of y-coordinates of points on left and right sides of a selected point); d) distance from the nearest intersection (quantifies how far is a selected point away from an "intersection point", i.e., a point where acceleration curve crosses from negative to positive or vice versa"); e) sign variation between two successive points. After examining all the points by checking these 5 different conditions, the terminal points can be generated for the motion data on each axis. Since, all these five conditions are checked separately on x- and z- axes acceleration data, two matrices are generated for each of gesture sequence data

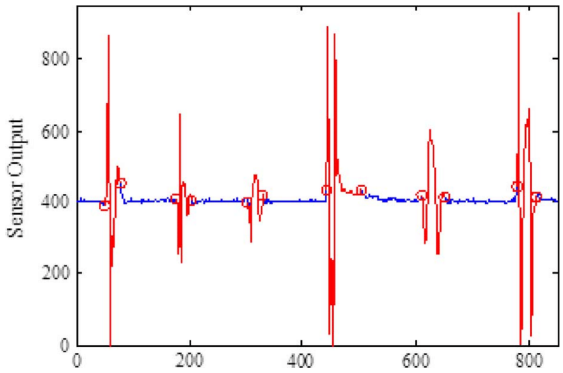

Fig. 5. Segmentation of a seven-gesture sequence in the order up-down-leftright-tick-circle-cross.

The element on the first row is the starting point of a gesture and the element in the second row in same column is the end points of the same gesture. Comparing the columns of the two matrices, if the pair of terminal points on one axis is close to a pair on the other axis, one pair of the terminal points will be eliminated. A final determination on if a given set of pairs of points are indeed terminal points is made by comparing the maximum acceleration between them with the mean value of the maximum accelerations between all pairs of points. If the former is too small, then that pair of points will be eliminated. As an example, the final terminal points for both x- and z- axes are denoted by circles in Fig. 5. After obtaining the terminal points of each gesture, the number of gestures becomes obvious since every gesture has one starting point and one end point, i.e., the number of the

columns of the final terminal points

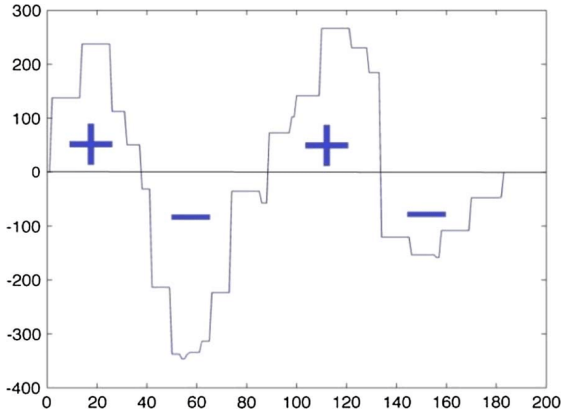

Fig. 6. Sign sequence generation.

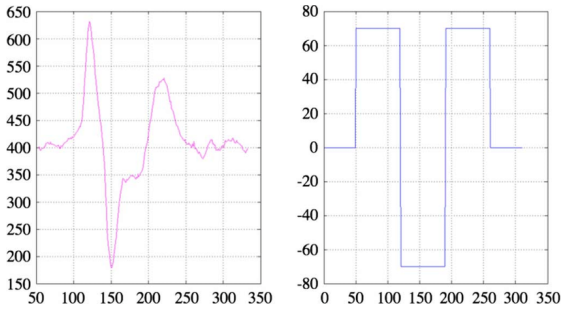

Fig. 7. Feature extraction transformation.

V.GESTURE RECOGNITION BASED ON VELOCITY INCREMENT

Due to the intensity variance of each gesture, an area sequence should be normalized before stored as training data or aligned so that their centers of masses coincide. This algorithm is expected to reduce the possibility of misalignment. The final step is to compare the velocity increment sequence by subtracting two area sequence vectors

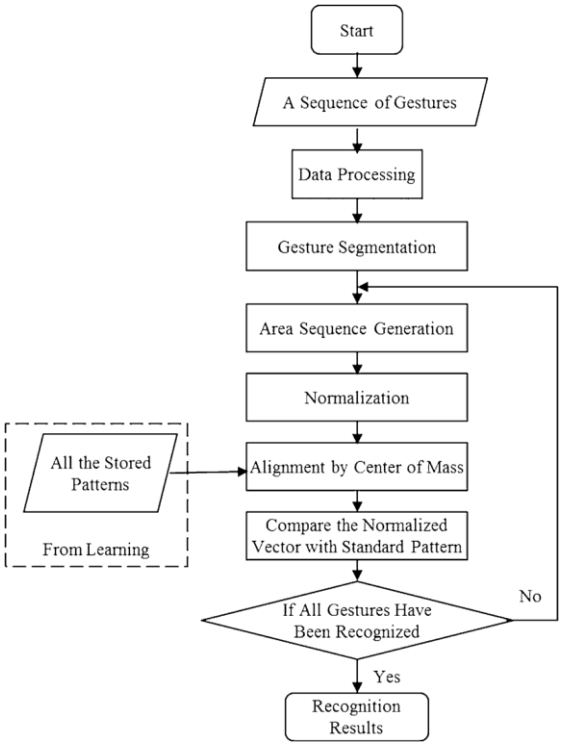

Fig. 8. Work flow chart

VI. EXPERIMENT RESULTS

Using machine motion techniques, in our approach logical patterns are obtained from a physical sensor attached to the system. The chip set includes accelerometer or some other type of motion sensor responsive to movement of device and which provides signal corresponding to the movement of device (to processor or microcontroller or any other electronic circuitry), stored in memory, VB which may be data base of values corresponding to predetermined motion or orientation electronic device. In operation, wh en accelerometer detects the motion of device and signals the processor, processor access the data base and one of the two things happen: either the movement if the device corresponds to the values stored as motion password; or the movement of the device don't corresponds to motion password stored. Acceleration in 2

directions: up -down, forward backward of the system movement, which is obtained by the sensor, is used for authentication.

VII.SNAP SHOTS

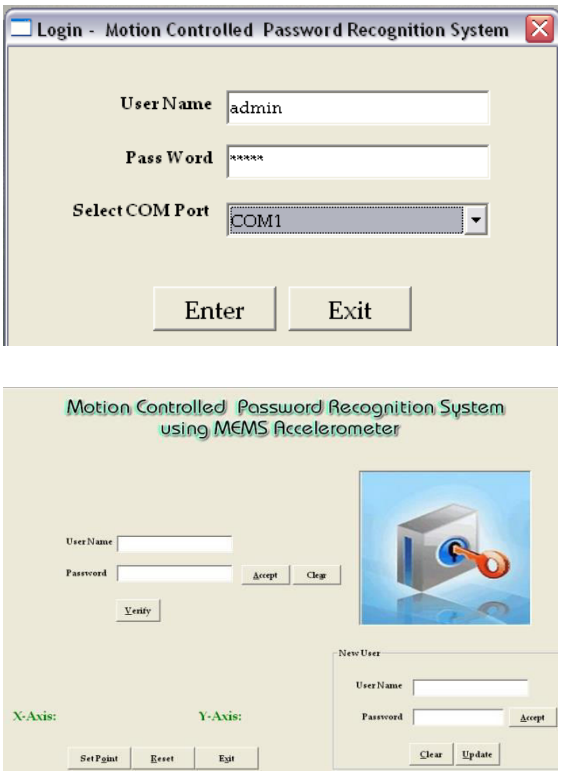

Fig.9 Visual basic application software

VIII.FIRMWARE IMPLEMENTATION

This project is implemented using following software's:

- KEIL compiler - for compilation part
- FLASH MAGIC – for dumping part

KEIL Compiler

For ARM7 processor, KEIL3 compiler is used for compilation. The compilation steps are as follows:

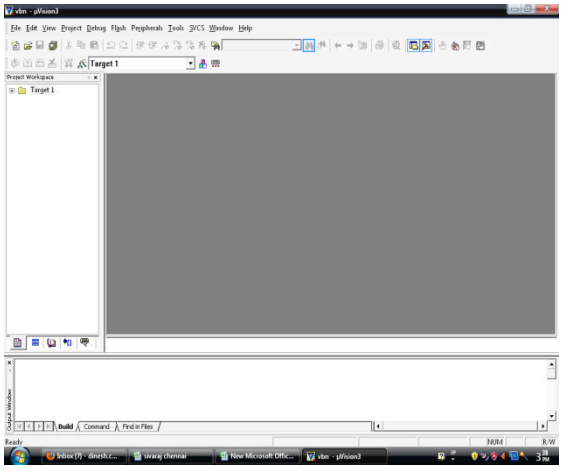

Fig.10 Picture of opening KEIL compiler

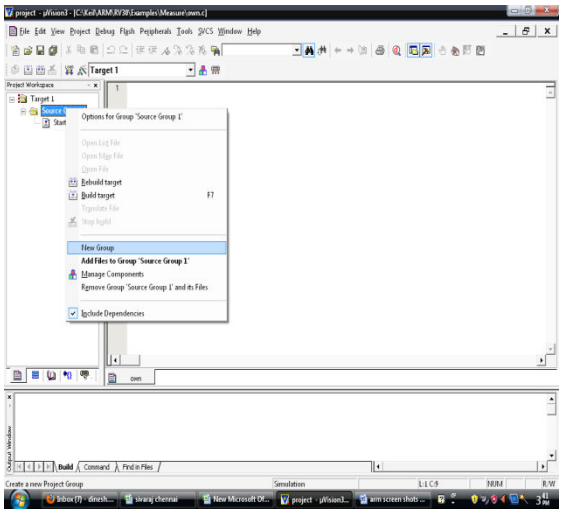

Fig.11 Picture of opening file for writing prog using KEIL compiler

DUMPING STEPS

The steps involved in dumping the program to microcontroller are shown below.

The main window of the FLASH MAGIC programmer is as show in the figure below which is self explanatory

1. Initially before connecting the program dumper to the microcontroller kit the window is appeared as shown above.

2. Select device option and click on Check Communication for establishing a connection.

3. Import the program which is '.hex' file from the saved location by selecting Browse option and clicking on 'Import Hex'.

4. After clicking on 'Import Hex' option we need to select START option for dumping the program into the microcontroller.

5. After the successful dumping of program it shows that completed.

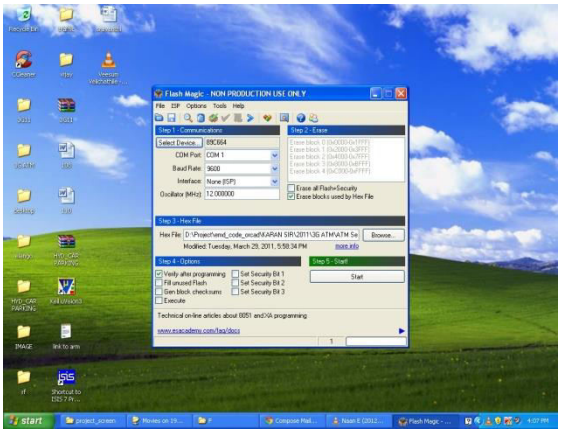

Fig.12 FLASH MAGIC window

IX. CONCLUSION

A motion controlled password recognition system using MEMS accelerometer has been implemented using ARM processor. This analyzes whether any motion has occurred or not and in the absence of a motion for a predetermined duration, forces the system to go into a low-power mode. By analyzing the motion pattern of the user and authentication can be provided to the user. Here direction of the motion has been assigned as a password to the system and the status of the system can be viewed with the help of application software which has been developed using high level graphical programming language (Visual Basic).

REFERENCES

[1] B.S. Kerner, C. Demir, R.G. Herrtwich, S.L. Klenov, H. Rehborn,M.Aleksic, and A.Haug,"Traffic state detection with floating car data in road networks," *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, Sept. 2005, pp. 44-49.

[2] Victor Olugbemiga Matthews,Emmanuel Adetiba,"Vehicle Accident Alert and Locator (VAAL)" International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 11 No: 02,April 2011.pp.38-44.

[3] Rajesh Kannan Megalingam, Ramesh Nammily Nair,Sai Manoj Prakhya Amrita VishwaVidyapeetham, Amritapuri, Clappana,"Wireless Vehicular Accident Detection and Reporting System" 2010 International Conference on Mechanical and Electrical Technology (ICMET 2010)pp-636-640,2010 IEEE.

[4] Douglas V. Hall, "Microprocessors and Interfacing Programming and Hardware'', Tata McGraw-Hill Publishers, II Edition, New Delhi -1999.

[5] Muhammad Ali Mazidi, Janice Gillespie Mazidi, "The 8051 Microcontroller Microcontroller and Embedded systems'', Pearson Education, 2nd edition,2000

[6]http://www.its.berkeley.edu/volvocenter/VREF/UCBSOA_VIIChristian.pdf

[7]http://www.theiet.org/factfiles/it/rfidpage.cfm?type=pdf

[8]http://www.itsc.org.sg/pdf/Journal%202004
/Section_Three04/Three_Water_2a.pdf