# Warehouse Stock Management and Tracking System based on Silicon Identification Technology and 1-Wire Network Communication

S Javid [1], M Bhagyalakshmi Devi M.Tech. [2]
[1](*Electronics and Communication Engineering,. / JNTUK, India)*
[2](*Assoc. Professor Electronics and Communication Engineering, / JNTUK, India)*

***ABSTRACT :*** *This paper presents à completely unique electronic system for tock management and container tracking in a storage warehouse environment. The system relies on the utilization of advanced electronic identification tags and 1-Wire communication tools. Each warehouse container is affixed with a small electronic ID tag (called iButton). The tag is an electronic chip housed in a very durable 16mm button-shaped stainless-steel package (microcan). The chip provides a unique ID number read-write memory and 1-Wire interface communication circuitry. It is powered by a long lasting internal battery. The tag keeps information on storage loading/unloading history, goods contents, ownership, etc. Another iButton known as location tag is placed at each and every storage cell. Once a container is placed into an accessible cell, the pair of ID tags (on the container and within the cell) appeared instantly on the 1-Wire communication network, hence permitting the central controller (PC) to update the latest pace utilization and also the container/stock related to it. This intelligent way of tagging the containers with their particular contents to any available storage cells, offers a replacement different to the conventional storage and stock management system wherever a particular container should be placed into its pre-specified storage area..*

***Keywords*** *- iButton, identification tag, 1-Wire communication, automatic storage system, stock management.*

## I. Introduction

Efficient and effective stock management and tracking could be a hallmark of productive company. Several firms and industries have heavily invested into obtaining the simplest stock and inventory management systems. varied studies have shown that the effectiveness and performance of an inventory depends on the accuracy of the products information at any given time, which incorporates the storage locations, goods retrieval rate from the shelves, packing history, stock quantities, contents, etc. [1-3]. Such information is crucial in helping the management to manage a stock level, to forecast market demand and material dominant within the warehouse.

There are two normally renowned technol ogies that are most widely employed in the inventory systems these days. They are supported the utilization of barcode and frequency Identification (RFID). The barcode-based systems although cheap to implement are simply liable to wear and tear damages. Their performance might even be stricken by poor print quality and print distinction quantitative relation. Using RFID technology in management inventory system is not fully while not its issues either.

Problems connected to electro-magnetic interference, signal distortion, absorption and deflection between RFID tag and reader are the foremost common issues there. Alternative factors like vibrations and static charges may negatively have an effect on the performance of RFID tags. The sturdiness of the RFID tags will cause another problem its deteriorated once exposed to wetness, which is often gift in storage areas [4-6]. This paper describes a unique implementation of warehouse stock management and tracking system supported the advanced silicon identification technology product (so-called iButton) and therefore the 1-Wire communication protocol [7].It elaborates on both the hardware and software package implementations that offer a user friendly atmosphere with high reliability among alternative options.

## II. SYSTEM OVERVIEW

To achieve the goal the subsequent further technical arrangements are enforced within the warehouse. Each of the storage cells/shelf positions is provided with an individual position identification tag – iButton [7]. All the storage cells are connected through the 1-Wire communication network to the central host computer. But solely cells with containers in them have a control system and so seen by the computer. Those cells that don't seem to be occupied are disconnected and so don't seem to be seen by the computer.

Every storage container is additionally connected with its own iButton with data on the container content, owner, storage instructions, ending date, loading/unloading history, etc. Each placement of container to an accessible storage space mechanically results in association of position and container tags to the 1-Wire network connected to the central computer - Fig. 1.

The network association is designed according to the MicroLAN design[7]

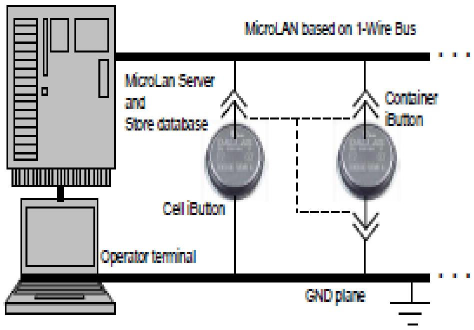


Figure 1. Network Architecture

Whenever the container is placed onto the available shelf space, the pair of iButton information records (container content, etc. and shelf location) area immediately captured by the host computer through the network and updated within the central database (as well as within the tag memories if required).

## III. IBUTTON AND 1-WIRE COMMUNICATION PROTOCOL

The iButton is a computer chip capsulated in a very 16mm stainless steel micro-can (Fig.1). It has the "lid" for 1-Wire data contact the "base" for ground contact. Each of those contacts is connected to the silicon chip inside.. The two contacts are separated by a plastic fastener. Every iButton contains a laser-programmed computer memory containing a 64-bit distinctive device ID, which has 6-byte serial number, 1-byte family code and 1-byte Cyclic Redundancy Check (CRC) verification field Fig. 2

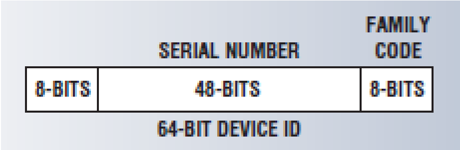


Figure 2. iButton 64-bit layered ROM structure

In addition to ROM, iButton provides read-write memory of various capability (depending on the chip type) and 1-Wire interface communication electronic equipment. The operation of the chip is backed by the inbuilt long life lithium battery. The 1-Wire protocol provides a very efficient half-duplex bidirectional communication and power supply transmission on single data line [7]. Fig. 3 [9] shows how a Master device (central host PC) and one amongst the Slaves (e.g., iButtons of DS199X series [9]) are often connected on the 1-Wire MicroLAN network.

Each device connected to the 1-Wire bus must have open-drain or three-state outputs, and the connection is done through a pull-up resistor of approximately 5kΩ. The 1-Wire system is able to transmit data at the rate of 14.4-16.3 Kbps in standard mode and up to 142 Kbps in overdrive mode.

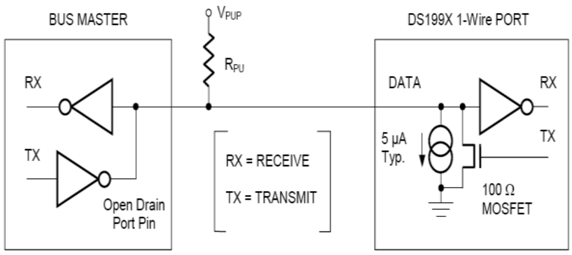


Figure 3. 1-Wire hardware connection

Accessing an iButton through the 1-Wire port involves four steps: Initialization, read-only memory Function Command, Memory Function Command and Transaction/Data.

Initialization:

At the start of the communication session, the bus Master must understand that 1-Wire devices are out there and prepared to work. This is often accomplished by the bus Master transmits a Reset pulse. The Presence pulses are transmitted by the Slaves if any of them are present on the network. This informs the Master regarding the devices on the bus, and whether or not they are able to communicate.

ROM Function Command:

This command permits a particular device gift on the 1-Wire bus to be elite on the basis on its distinctive 64-bit serial identification (ROM) number. once one or a lot of 1-Wire Slave(s) square measure detected by the bus Master, Associate in Nursing 8-bit long read-only memory operate command is issued for inspecting the device by the bus Master.

Memory Function Command:

Reading and Writing into device memory moreover as performing specific functions on the 1-Wire device are done at this stage. only the active device (selected with the previous read-only memory command), responds to the Memory Function commands.

Transaction/Data:

The particular information transaction progresses either from the Master to the Slave or the opposite direction For communication purpose, 1-Wire protocol provides four forms of signaling: low-level signaling (contains Reset pulse and Presence pulse), Write 0, Write 1 and read information. All the signals are generated by the bus Master except for the Presence pulse that is generated by the Slave device showing its availability to the Master. All transactions on the 1-Wire bus begin with an initialization sequence. The sequence (Fig. 4 [9]) consists of Reset pulse transmitted by the bus master followed by Presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus Master understands that the DS199X is on the bus and is prepared to work.

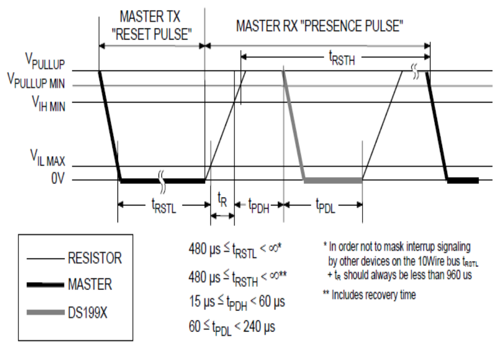


Figure 4. Initialization sequence

After the formatting sequence is completed, the bus master problems one amongst the four read-only memory perform commands (Read ROM, Match ROM, Skip ROM, Search ROM) to the slave iButton, followed by read-write memory perform command and eventually the particular information transaction starts between the Master and also the Slave. Fig. 5 [7] illustrates the timing for the Master to scan information (one bit) from the Slave.

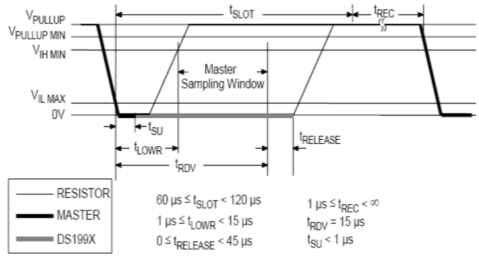


Figure 5. Read data timing diagram

The cycle begins with the Master pull the data line to Low. If the Slave wants to Write 1 on the line for the Master to read, it leaves the line High within the idle state. If but it needs to send 0, the Slave pulls the line to Low. After 15ms the Master samples the info line and therefore the level it detects could be a value for the bit.

Write 1 / Write 0 (Fig. 6, 7) are enforced in a very similar fashion because the scan data cycle [7]. It is initiated by the master pulling the data line to Low. The Master then either leaves the line High (for 1), or pulls it to Low (for 0). The lave device then samples the line after a window of 60ms.

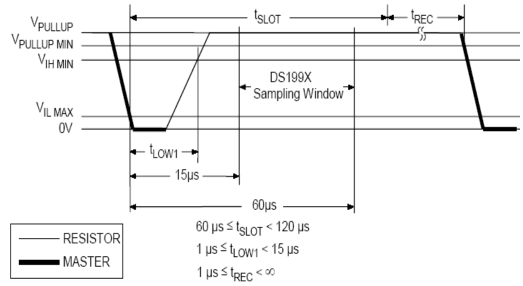


Figure 6. Write 1 timing diagram

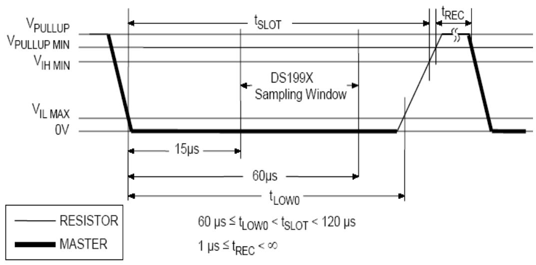


Figure 7. Write 0 timing diagram

By the means of those cycles, data is shipped bit by bit and later assembled into bytes within the Master or Slaves. Bytes are sent starting with the least significant bit. The Master will interpret the bytes received from the Slave as specific data, while the Slave uses them to perform specific functions.

## Iv. Prototype Design

Prototype hardware (which is employed as a development platform within the laboratory environment) is shown in Fig.8.It includes devices utilizing numerous iButtons, USB to 1- Wire adapter

1-Wire network tools and host computer [9-10]. In the real world the 1-Wire connection line will go as so much in length depending on the type of the employed network topology [10]. The system software package has been developed using Microsoft Visual Studio C++ software development kit. By utilizing .Net Framework, TMEX C API (by Maxim Integrated Products) and MySQL Database System.

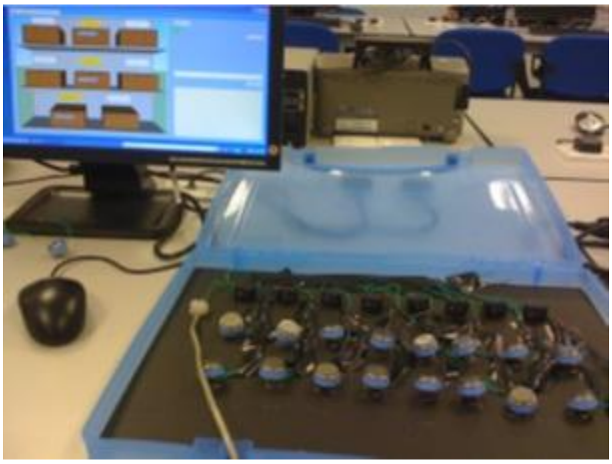

Figure 8. System prototype

The system implementation has been chosen attributable to their inexpensive, dependability, technological acceptableness and rapid development time.

a) Inexpensive. The value of the project involves only the purchase price of the iButtons and 1-Wire connecting devices, because the software tools are freely obtainable from their respective suppliers on the web. The cost-effectiveness is additional improved with the upper scale of the system (increasing variety of storage locations/containers for larger-scaled inventories), because the cost-per-unit for the devices are important lower with larger amount of order.

b) Dependability. iButtons have established a decent reputation for their sturdiness and dependability. The iButton product vary has been wear-tested for 10-year sturdiness [9]. This feature is especially important for systems operating during a harsh setting, e.g.[11].

c) Technological Acceptableness. There are 165 million iButtons presently being used during a wide range of applications worldwide, whereas the Microsoft .Net Framework and MySQL Database system are the business standards in software applications.

d) Rapid Development Time. Maxim Integrated Products provides the TMEX C API software [9] for rapid development of communication tools between the host computer, iButtons and alternative 1-Wire devices therefore facilitating system integration

## V. DESIGN SOLUTIONS

While 1-Wire communication protocol and iButtons are widely utilized in numerous applications, a study has shown that 1-Wire communication protocol isn't able to verify the physical sequence of the devices on the network.

This is due to the utilization of wired-AND property of the 1-Wire bus in identifying the IDs of the devices on the network. The procedure permits reliable identification of the IDs of all devices on the network. but it's unable to determine a consecutive order of the devices, thus losing their location information on the network.

One of the possible solutions is to use the chain perform communication idea as delineate in Fig.9 [12]. There should be 2 additional pins on every slave device (location tag) so as to facilitate the sequence location identification.

These 2 pins are: an input (active-low EN) to enable a device to reply the discovery, and an output (active-low DONE) to inform consequent device within the chain that the discovery of the previous device is completed. The active-low DONE signal of one device is additionally connected to the active-low EN input of consequent device, so on (Fig. 8). In this chain operation only the active-low nut input of the primary device within the 1-Wire network is hardwired to GND.

However this chain discovery methodology isn't appropriate for the stock management and goods tracking system, where the chain has to be broken over and over whenever there are requests for storage and retrieval of the containers or the stocks within the warehouse. This results in wiring complications, special electronic tags having four terminals ( IO, GND, EN and Done), and needs the utilization of latest network perform command Conditional browse fixed storage rather than the traditional Read ROM.

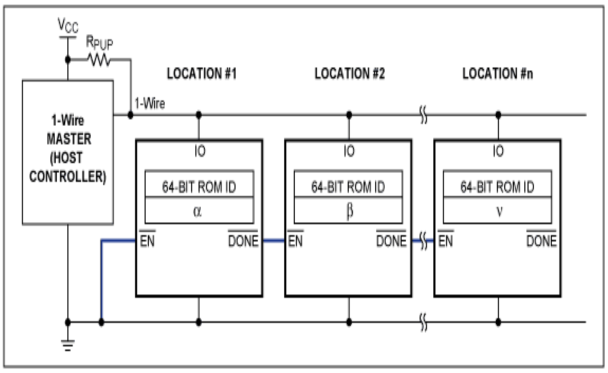


Figure 9. 1-Wire network using chain function

The current project commenced with developing the Graphical *User Interface* (GUI) for basic testing of the functionality and capability of TMEX C API to establish communication between the host computer and a try of iButtons. With the functionalities of the TMEX C API verified, the model hardware implementation was expanded into multiple pairs of iButtons connected to the host computer. The MySQL database system was additional into the software implementation on the later stage, with the aim of data-logging the activities of the inventory system for historical records.

## VI. SYSTEM PROTOTYPE IMPLEMENTATION

The hardware diagram is shown in Fig. 10. The tracking method begins once a stock container is being pushed into the storage location wherever each iButtons (the stock container one and storage location tag) physically contact the drive wire. This suggests that the paired IDs are now connected to the 1-Wire network and so - to the host computer.

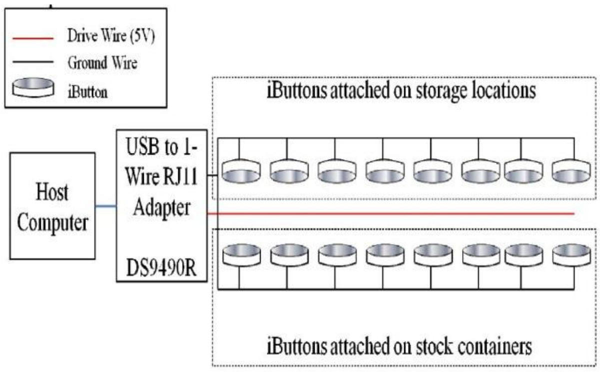


Figure 10. Hardware block diagram

Fig.11 shows an example of how contact will be established between the storage location's iButton, stock container's iButton and therefore the host computer via a conducting metal strip acting because the drive wire. For the sake of simplicity in the scope of the prototype development the connection of iButton pairs to the 1-Wire network was implemented using 2-way switches (Fig.12). A pair of iButtons (attached on storage location and stock container respectively) is connected to the host pc via1-Wire network once the 2-way switch is closed. The host PC is then able to gather information on the physical location of the relevant stock container based on the recognition of the 64-bit iButton ID, and also to link this information to data representing the container contents, ownership, packing history, etc., keep in the container's tag.

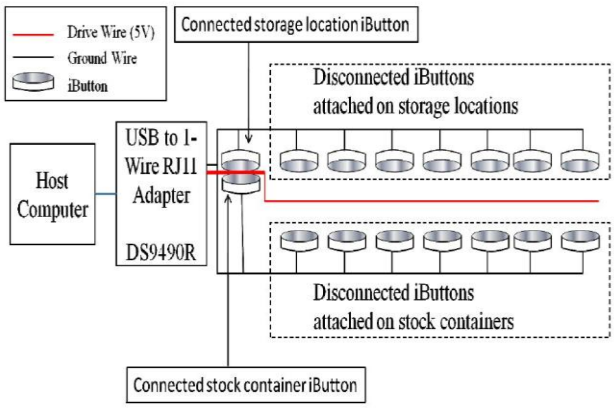


Figure 11. A pair of iButtons in contact with drive line

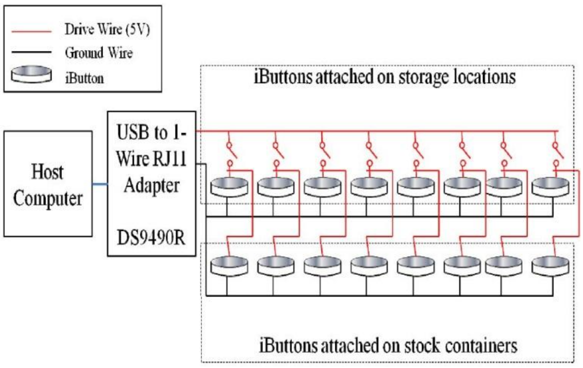


Figure 12. Switching circuitry

The snapshot of the hardware implementation of the connection circuitry of the prototype is shown in Fig. 13

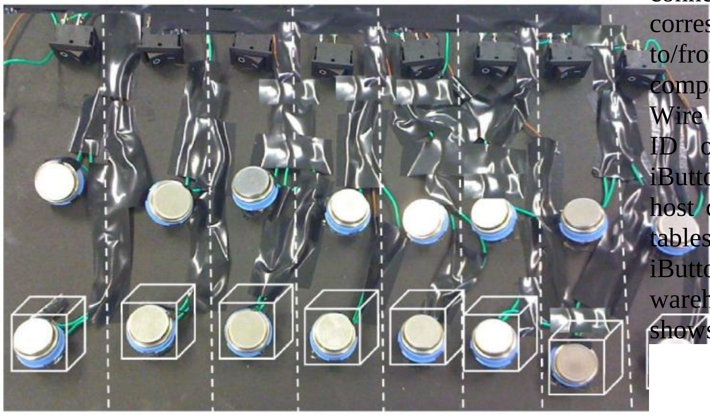

Figure 13. Tag connections

## VII. SOFTWARE

As mentioned earlier, the software of the project has been developed using three main components, which are Microsoft .Net Framework, TMEX C API and MySQL Database System.

The Microsoft .Net Framework [13] provides a platform for building various user applications, seamless and secure communication, and the ability to model a range of processes. It consists of [13]:

*Common Language Runtime* – provides an abstraction layer over the operating system;

*Base Class Libraries* – pre-built code for common lowlevel programming tasks; *Development frameworks and technologies* – reusable, customizable solutions for larger programming tasks. The GUI of the project is developed solely using Microsoft .Net Framework, which ensures compatibility with Microsoft Windows operating systems. TMEX C API [14] provides a list of functions that allow various operations to be performed on the 1-Wire network connected to the computer. The proposed system uses TMEX C API to retrieve the IDs and memory content from each iButton connected to the 1-Wire network.

MySQL is an open-source database system widely used in various industries [15]. MySQL Database System is used in the proposed system to store and retrieve stock and container location data as well as record loading/unloading activities (by monitoring connection/disconnection of pairs of iButtons on the 1-Wire network). The later objective is achieved by using a self-firing timer function with 50mS interval to poll the 1-Wire network continuously. The polling keeps track on the number of iButtons on the 1-Wire network during each cycle and stores the IDs of all present iButtons into a table in the MySQL database system. Therefore any increment/decrement in the number of iButtons as compared to the number obtained during the previous cycle indicates a connection/disconnection of a pair(s) of iButtons corresponding to loading/unloading a container(s) to/from the storage cell(s) respectively. By comparing the IDs of all iButtons present on the 1-Wire network to the table in MySQL database, the ID of the newly connected or disconnected iButtons can be identified by the software on the host computer. A record is then inserted into the tables of the storage location and stock container iButtons to keep track of the stocks and store the warehouse activity for reporting purposes. Fig. 14 shows the flow diagram illustrating the algorithm.

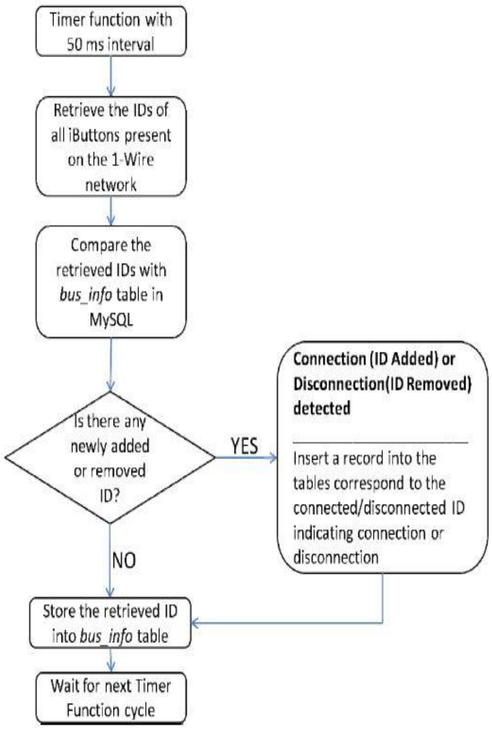


Figure 14. Network polling and iButtons ID retrieval/recording algorithm

By integrating various portions of the hardware and software including GUI, a user-friendly inventory management and control system has been created. Fig. 15 shows a screenshot when there are stock containers are present at storage locations 2, 3, 4, 5 and 7. By placing the mouse pointer over an occupied cell, the detail information related to the store location and container is displayed in the bottom-right corner of the screen. It includes the IDs of the storage location and stock container, as well as the content of the stock container. This feature provides the user a quick and intuitive interface to view the status and information of the entire inventory. The presented information can be easily enhanced to include any other relevant data (good ownership, final destination, storage requirements and expiry date, etc.)

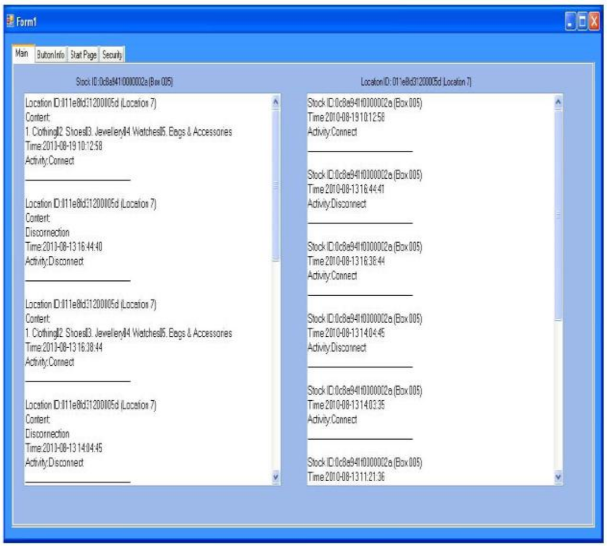

Figure 15. Storage overview window

Fig. 16 shows the system screenshot when the user clicks on a location to view all previous activities in the particular storage cell and stock container. Figure 16. Activities report window The software also provides a search function allowing the user to search the content of the entire warehouse using keywords. Fig. 17 shows the screenshot of the system when a match is found at the storage location 7. A security feature was also added to the system. It allows a user to specify the barring periods when all or some specific containers should not be taken from the shelves. This feature is valuable for inventories during non-operation time. Fig. 18 show the warning displayed when a disconnection occurs at storage location 5 during a specified range of time when loading/unloading activities are forbidden.

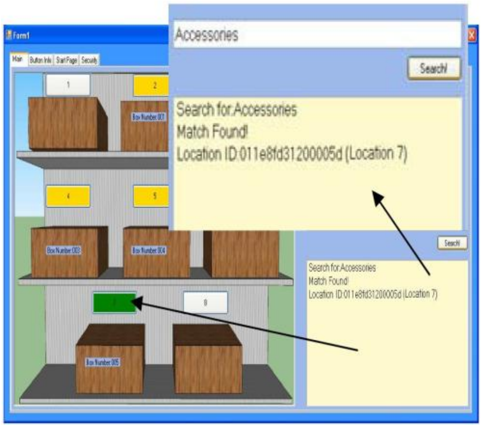


Figure 17. Search function implementation

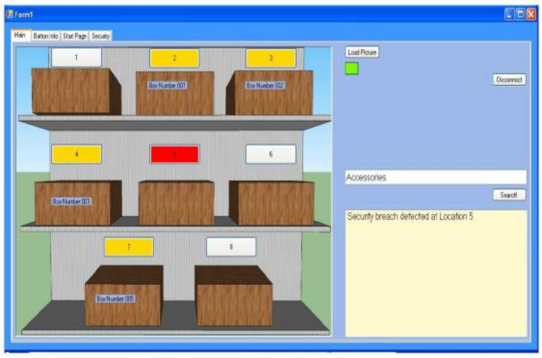

Figure 18. Security breach alert screen

## VIII. CONCLUSION

Wide variety systems based on application iButtons and 1- Wire network protocol have been developed and successfully implemented in various application areas world-wide. This paper proposes one more potentially very beneficial solution, where iButton electronic ID technology and 1-Wire communication protocol are employed for storage space allocation and container/load tracking. The paper summarizes the findings and results of the final year BE degree project by one of the authors (K.X.T.). The operational prototype of Intelligent Storage, Stock Control and Tracking System has been successfully developed and tried in course of this study. New outlook and functionalities of the warehouse inventory management system have been achieved with a reasonable ease of implementation and cost effectively while providing good reliability among other features. It is expected that the very encouraging outcome of the project would encourage further development in this promising area, thus leading to efficient industrial application.

## REFERENCES

[1] G. Relph, W. Brzeski and G. Bradbear, “Professional

inventory management" – Seminar: *Demand Chain Management in 21st Century*. [Online]: http://www.inventorymatters.co.uk.noisegate2.webhoster.co.uk//userFiles/profesional_inventory_management.pdf

[2] Y. Kang and S. B. Gershwin, "Information inaccuracy in inventory systems: stock loss and stockout," *IIE Transactions*, vol. 37, December 2004, pp. 843-859.

[3] G. W. Plossl, "Fundamentals of inventory management," in *Production and Inventory Control: Principles and Techniques*, Ed. Second Edition, Englewood Cliffs: Prentice Hall, 1985, pp. 16-29.

[4] C. Sellitto, S. Burgess and P. Hawking, "Information quality attributes associated with RFID-derived benefits in the retail supply chain," *International Journal of Retail & Distribution Management*, vol. 35, no.1, 2007, pp. 69-87.

[5] G.-D. Park, K.-S. Min and J.-C. Lee, "Durability design of passive tag antenna for logistic RFID system," *Asia-Pacific Microwave Conference- APMC'2008*, 2008, pp. 1-4.

[6] iButton Guard Tour Solutions. [Online]: http://www.maximic.com/products/ibutton/applications/dbimages/ppt/31.ppt

[7] iButton Standards [Online]: http://pdfserv.maximic.com/en/an/appibstd.pdf

[8] H. M. Dao, S. Demidenko and M. T. Chew, "An intelligent storage and retrieval system based on electronic ID technology and 1-Wire communication", *15th Electronics New Zealand Conference*, 2008, pp.133-137

[9] DS1992/DS1993 [Online]: http://www.maximic.com/quick_view2.cfm?qv_pk=2831

[10] Guidelines for Reliable Long Line 1-Wire® Networks [Online]: http://www.maxim-ic.com/appnotes/index.mvp/id/148

[11] M.-T. Chew, T.-H. Tham, and Y.-C. Kuang, "Electrical power monitoring system using Thermochron sensor and 1- Wire communication protocol", *4th IEEE International Symposium on Electronic Design, Test and Applications*, 2008, pp. 549-554.

[12] "Regain location information by leveraging the 1-Wire chain function", *Maxim Engineering Journal*, vol. 60, 2007, pp. 16-19 [Online]: http://pdfserv.maxim-ic.com/en/ej/EJ60.pdf

[13] Microsoft .NET Framework. [Online]: http://www.microsoft.com/net/

[14] 1-Wire® software resource guide device description [Online]: http://www.maxim-ic.com/appnotes/index.mvp/id/155

[15] MySQL [Online]: http://www.mysql.com/