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

R. ABDUL HAKEEM\(^1\)  
Research Scholar, Dept of ECE, VIF College of Engineering and Technology, Hyderabad, A.P.-INDIA,  
Email: hakeemabdul587@gmail.com

K. RAVINDER\(^2\)  
Asst Prof, Dept of ECE, VIF College of Engineering and Technology, Hyderabad, A.P.-INDIA.  
Email: koraniphd814@gmail.com

WAJID HUSSAIN\(^3\)  
HOD, Dept of ECE, VIF College of Engineering and Technology, Hyderabad, A.P.-INDIA.  
Email: wajidhussain452@gmail.com

Abstract: This paper presents a novel electronic system for stock control and container tracking in a storage warehouse environment. The system is based on the use 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 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 identified as location tag is placed at each and every storage cell. Once a container is placed into an available storage cell, the pair of ID tags (on the container and in the cell) appeared immediately on the 1-Wire communication network, hence allowing the central controller (PC) to update the latest space utilization and the container/stock associated with it. This intelligent way of tagging the containers with their particular contents to any available storage cells, offers a new alternative to the conventional storage and stock management system where a particular container must be placed into its pre-specified storage space.

Keywords: iButton, Identification Tag, 1-Wire Communication, Automated Storage System, Stock Management.

I. INTRODUCTION

Efficient and effective stock control and tracking is a hallmark of a successful company. Many companies and industries have heavily invested into getting the best stock and inventory management systems. Various studies have shown that the effectiveness and performance of an inventory relies on the accuracy of the goods information at any given time, which includes 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 regulate a stock level, to forecast market demand and material controlling in the warehouse.

There are two commonly known technologies which are most widely used in the inventory systems nowadays. They are based on the use of barcode and Radio Frequency Identification (RFID). The barcode-based systems though cheap to implement are easily susceptible to wear and tear damages. Their performance could Using RFID technology in management inventory system is not completely without its problems either. Issues related to electro-magnetic interference, signal distortion, absorption and deflection between RFID tag and reader are the most common problems there. Other factors like vibrations and static charges can also negatively affect the performance of RFID tags. The durability of the RFID tags can pose another problem: it is deteriorated when exposed to moisture, which is often present in storage areas [4-6]. This paper describes a novel implementation of warehouse stock control and tracking system based on the advanced silicon identification technology product (so-called iButton) and the 1-Wire communication protocol [7]. It elaborates on both the hardware and software implementations that provide a user friendly environment with high reliability among other features.

II. SYSTEM OVERVIEW

To achieve the goal the following additional technical arrangements are implemented in the storage warehouse. Each of the storage cells/shelf positions is equipped with an individual position identification tag iButton. All the storage cells are linked through the 1-Wire communication network to the central host computer. However only cells with containers in them have a closed loop and thus seen
by the computer. Those cells that are not occupied are disconnected and thus are not seen by the PC. Every storage container is also attached with its own iButton with data on the container content, owner, storage instructions, expiry date, loading/unloading history, etc. Each placement of a container to an available storage space automatically leads to connection of both the position and container tags to the 1-Wire network linked to the central computer. The network connection is configured according to the Micro LAN architecture. Whenever the container is placed onto the available shelf space, the pair of iButton information records (container content, etc. and shelf location) are immediately captured by the host computer through the network and updated in the central database (as well as in the tag memories if required).

III. IBUTTON AND 1-WIRE COMMUNICATION PROTOCOL

The iButton is a computer chip enclosed in a 16mm stainless steel micro-can (Fig.1). It has the “lid” for 1-Wire data contact and the “base” for ground contact. Each of these contacts is connected to the silicon chip inside. The two contacts are separated by a polypropylene grommet. Every iButton has a laser-programmed ROM containing a 64-bit unique device ID, which includes 6-byte serial number, 1-byte family code and 1-byte Cyclic Redundancy Check (CRC) verification field (Fig. 2).

In addition to ROM, iButton provides read-write memory of various capacity (depending on the chip type) and 1-Wire interface communication circuitry. The operation of the chip is backed by the built-in long-life lithium battery. The 1-Wire protocol provides a very efficient half-duplex bidirectional communication and power supply transmission on a single data line [7]. Fig. 3 [9] shows how a Master device (central host PC) and one of the Slaves (e.g., iButtons of DS199X series [9]) can be connected on the 1-Wire MicroLAN network. Each device attached 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 1. Network architecture**

**Figure 2. iButton 64-bit lasered ROM structure**

**Figure 3. 1-Wire hardware connection**

Accessing an iButton through the 1-Wire port involves four steps: Initialization, ROM Function Command, Memory Function Command and Transaction/Data. Initialization. At the beginning of the communication session, the bus Master needs to know which 1-Wire devices are available and ready to operate. This is accomplished by the bus Master transmitting a Reset pulse. The Presence pulses are transmitted by the Slaves if any of them are present on the network. This informs the Master about the devices on the bus, and whether they are ready to communicate. ROM Function Command. This command allows a particular device present on the 1-Wire bus to be selected on the basis of its unique 64-bit serial identification (ROM) number. After one or more 1-Wire Slave(s) are detected by the bus Master, an 8-bit long ROM function command is issued for inspecting the device by the bus Master.

Memory Function Command. Reading and Writing into device memory as well as performing specific functions on the 1-Wire device are done at this stage. Only the active device (selected with the previous ROM command),
An Intelligent Warehouse Stock Management and Tracking System based on Silicon Identification Technology and 1-Wire Network Communication

responds to the Memory Function commands. Transaction/Data. The actual data transaction progresses either from the Master to the Slave or the other direction. For communication purpose, 1-Wire protocol provides four types of signaling. Initialization (contains Reset pulse and Presence pulse), Write 0, Write 1 and Read Data. All the signals are generated by the bus Master except for the Presence pulse, which 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 know that the DS199X is on the bus and is ready to operate.

Figure 4. Initialization sequence

After the initialization sequence is completed, the bus master issues one of the four ROM function commands (Read ROM, Match ROM, Skip ROM, Search ROM) to the slave iButton, followed by read-write memory function command and finally the actual data transaction starts between the Master and the Slave. Fig. 5 [7] illustrates the timing for the Master to Read Data (one bit) from the Slave.

Figure 5. Read data timing diagram

The cycle begins with the Master pulling 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 in the idle state. If however it needs to send 0, the Slave pulls the line to Low. After 15ms the Master samples the data line and the level it detects is a value for the bit. Write 1 / Write 0 (Fig. 6, 7) are implemented in a similar fashion as the Read 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 Slave device then samples the line after a window of 60ms.

Figure 6. Write 1 timing diagram

Figure 7. Write 0 timing diagram

By means of these cycles, data is sent bit by bit and later assembled into bytes in the Master or Slaves. Bytes are sent starting with the least significant bit. The Master can 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 used as a development platform in the laboratory environment) is shown in Fig.8. It includes devices utilizing various iButtons, USB to 1-Wire adapter, 1-Wire network tools and host PC [9-10]. In the real world the 1-Wire connection line can go as far as 450m in length depending on the type of the employed network topology [10]. The system software 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. The system implementation has been chosen due to their low-cost, reliability, technological acceptability and rapid development time.

a) Low-cost. The cost of the project involves only the purchase price of the iButtons and 1-Wire connecting devices, as the software tools are freely available from their respective providers on the internet. The cost-effectiveness is further improved with the higher scale of the system (increasing number of storage locations/containers for larger-scaled inventories), as the cost-per-unit for the devices are significant lower with larger quantity of order.

b) Reliability. iButtons have established a good reputation for their durability and reliability. The iButton product range has been wear-tested for 10-year durability [9]. This feature is particularly significant for systems operating in a harsh environment, e.g.[11].

c) Technological Acceptability. There are 165 million iButtons currently being utilized in a wide range of applications worldwide, while the Microsoft .Net Framework and MySQL Database system are the industry 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 PC, iButtons and other 1-Wire devices thus facilitating system integration.

Figure8. System prototype

V. DESIGN SOLUTIONS

While 1-Wire communication protocol and iButtons are widely employed in various applications, a study has shown that 1-Wire communication protocol is not able to determine the physical sequence of the devices on the network. This is due to the use of wired-AND property of the 1-Wire bus in identifying the IDs of the devices on the network. The procedure allows reliable identification of the IDs of all devices on the network. However it is not capable to determine a sequential order of the devices, hence losing their location information on the network. One of the possible solutions is to use the chain function communication concept as described in Fig.9 [12]. There should be two additional pins on each slave device (location tag) so as to facilitate the sequence location identification. These two pins are: an input (active-low EN) to enable a device to respond during the discovery, and an output (active-low DONE) to inform the next device in the chain that the discovery of the previous device is done. The active low DONE signal of one device is also connected to the active-low EN input of the next device, and so on (Fig. 8). In this chain operation only the active-low EN input of the first device in the 1-Wire network is hardwired to GND. However this chain discovery method is not suitable for the stock management and goods tracking system, where the chain needs to be broken many times whenever there are requests for storage and retrieval of the containers or the stocks in the warehouse. This leads to wiring complications, special electronic tags having four terminals (IO, GND, EN and Done), and requires the use of new network function command Conditional Read ROM instead of the normal Read ROM.

Figure9. 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 pair of iButtons. With the functionalities of the TMEX C API verified, the prototype hardware implementation was expanded into multiple pairs of iButtons connected to the host computer. The MySQL database system was added 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 block diagram is shown in Fig. 10. The tracking process begins when a stock container is being pushed into the storage location where both iButtons (the stock container one and storage location tag) physically contact the drive wire. This means that the paired IDs are now linked to the 1-Wire network and thus - to the host computer.

![Figure10. Hardware block diagram](image)

Figure10. Hardware block diagram

![Figure11. A pair of iButtons in contact with drive line](image)

Figure11. A pair of iButtons in contact with drive line

Fig.11 shows an example of how contact can be established between the storage location’s iButton, stock container’s iButton and the host computer via a conducting metal strip acting as 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 computer via 1-Wire network when 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., stored in the container’s tag.

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

![Figure12. Switching circuitry](image)

Figure12. Switching circuitry

![Figure13. Tag connections](image)

Figure13. Tag connections

VII. REFERENCES


[14] 1-Wire® software resource guide device description