

# NFC Identification System for Fuel Management

R. Rajesh\*, G. Jesus Sandal Vinibha\*\*, K. Kalaimathi\*\*\*, P. Kamalakkanni\*\*\*\* & V. Kamatchi\*\*\*\*\*

\*Assistant Professor, Department of Electronics and Communication Engineering, SKP Engineering College, Tamil Nadu, INDIA.

\*\*UG Scholar, Department of Electronics and Communication Engineering, SKP Engineering College, Tamil Nadu, INDIA.

\*\*\*UG Scholar, Department of Electronics and Communication Engineering, SKP Engineering College, Tamil Nadu, INDIA.

\*\*\*\*UG Scholar, Department of Electronics and Communication Engineering, SKP Engineering College, Tamil Nadu, INDIA.

\*\*\*\*\*UG Scholar, Department of Electronics and Communication Engineering, SKP Engineering College, Tamil Nadu, INDIA.

**Abstract**—Air pollution in India is quite a serious issue with the major sources being fuel wood and biomass burning, fuel adulteration, vehicle emission and traffic congestion. The main reason of the pollution is vehicle emission particularly old vehicles. The simple and proper use of microcontroller and NFC (Near Field Communication) technology provides a simple solution to monitor the vehicle fuel consuming level to monitor the vehicle lifecycle for recycling the vehicle. It is interface with high speed fuel dispenser which is convenient for consumer to operate. In our system the NFC card will be placed to the every vehicle and the petrol pump have NFC reader and customer to place the NFC card into reader by the fuel station which will help the petrol company can monitor the fuel consumed level of the vehicle. In short we provide secure system for fuel distribution.

**Keywords**—Database; Data Logger; Data Storage PC; Dispense Controller; Fast Model Predictive Control (F-MPC); Intelligent Transportation System; Keyboard; Mifare 1K Card.

**Abbreviations**—Fast Model Predictive Control (F-MPC); Near Field Communication (NFC).

## I. INTRODUCTION

TODAY, lots of companies have their own gas station, to supply company cars. All of these companies seek for better identification system for fuel dispensing control. Older systems were not reliable enough. For example, in system with paper coupons there is real threat of fake coupons, which are difficult to distinguish from original. Furthermore, there is no clear insight into daily, weekly or monthly fuel dispensing neither how many vouchers are remaining in circulations, etc. [Clarck, 1]. For better recording and control of the fuel dispensing process, the system has to fulfill next requirements:

- provide clear insight into fuel consumption in real time,
- provide an overview of previous refills,
- allow dispensing without operator presence,
- enable fuel amount set-up for each car separately, etc..

This paper is proposing the system that meets these challenges. The system is based on NFC technology [Coskun et al., 2] with Mifare 1K 13.56MHz contactless cards as cars identifier [3]. Mifare technology ensures unique identification number for any Mifare card, worldwide. According to Mifare standard, higher security could be achieved by activating the 48 bits key. Mifare 1K card could support even longer key combining more memory sectors and its keys [3]. The system

is realized and mounted at the petrol station of the Ministry of the Interior, Republic of Montenegro. Each car from car fleet gets the identification card. Only with this card, the user can refuel the car. Instead paper voucher for 10, 20, 30, 50 or 100 litres, appropriate fuel quantity is approved for each card.

The system brings new quality in gas handling:

- cost savings,
- eco-friendly (eliminate usage of paper),
- faster administration (no need to collect paper-based
- vouchers and sent them to the issuer) and
- flexibility

At the end of the paper we propose some promising ways to achieve these goals.

## II. THE SYSTEM DESCRIPTION

The identification system we implemented consists of:

- Mifare 1K cards,
- Dispense controllers,
- Keyboards,
- Data logger,
- EIA232/EIA485 interface and
- Data storage PC with supporting software and database

- Fast Model Predictive Control (F-MPC)
- Intelligent transportation system.

The system includes 6 dispense controllers for fuel dispensing devices. Controllers are connected to data logger via EIS232 interface. The data storage PC and the logger are connected over EIA232/EIA485 interfaces. We choose this interface for three reasons:

- the connection distance exceeds 1km, much more
- then wire Ethernet limitation of 100m,
- the wireless connection was out of questions for data
- security reasons,
- there was few telephone line already protracted.

**2.1. Dispense Controller**

Dispense controller performs the following function in the system:

- reads Mifare 1K cards,
- turns on and off fuel dispensing devices,
- measures the amount of fuel dispensed,
- writes a messages on the LCD display and
- communicates with the data logger.

Dispense controller is realized using ATmega16 microcontroller [6, 7]. It performs card reading in accordance with ISO 14443A standards [Lekic et al., 8; Saha, 9; Lekic et al., 12]. Shows dispense controller input/output interface to fuel dispensing device. Operational amplifiers attenuate common mode voltages of signals SIG1, SIG2 and NOZZLE and pass filtered signals SIG1b, SIG2b and NOZZLEb to controller processor. The signal NOZZLE gives the state of the fuel dispensing nozzle: in use or hang up. Signals SIG1 and SIG2 comes from the fuel flow transducer. Additionally, the U3D amplifier translates TURNb signal from TTL level to +12V. The microcontroller turns on/off electromotive fuel pump via this U3D output stage [13; 14].

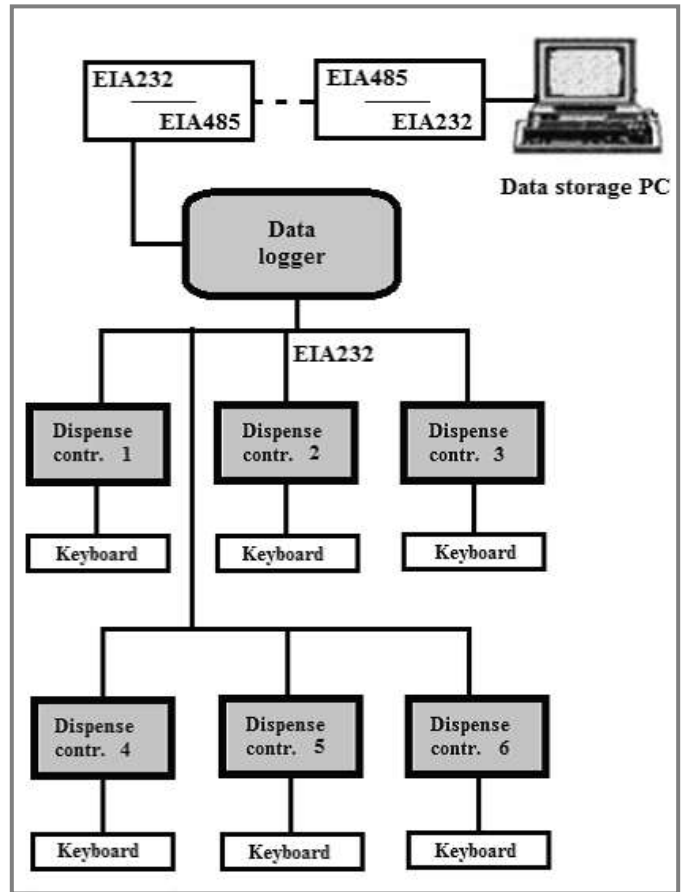


Figure 3: The Block Diagram of the System

Dispense controller has the LCD display, which shows information, like:

- Whether the user has the privileges to dispense fuel.
- Remained fuel quantity
- Current dispensing, etc.

Every controller is connected with one mechanical keyboard. Before any fuel dispensing, the user has to input car mileage by this keyboard. shows controller and keyboard, mounted on the fuel dispensing device [15; 16].

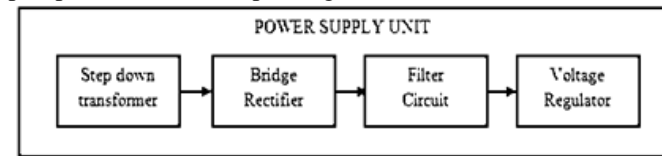


Figure 1: Power Supply Unit

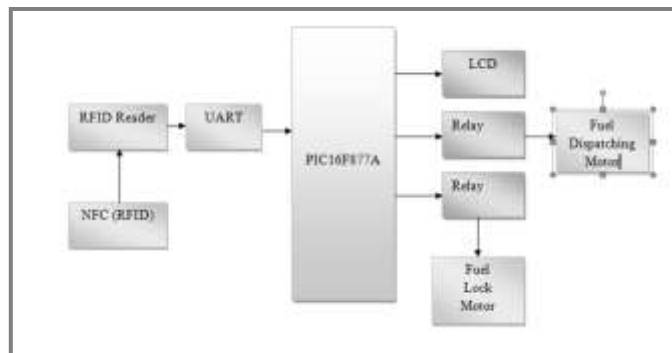


Figure 2: Block Diagram NFC Identification System

**2.2. Data Logger**

The data logger is central part of the system. The logger controls dispense controllers and ensure system operability. Figure 4 shows block diagram of the data logger.

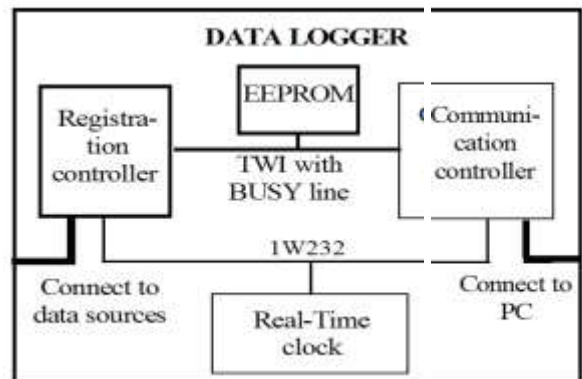


Figure 4: The Block Diagram of the Data Logger

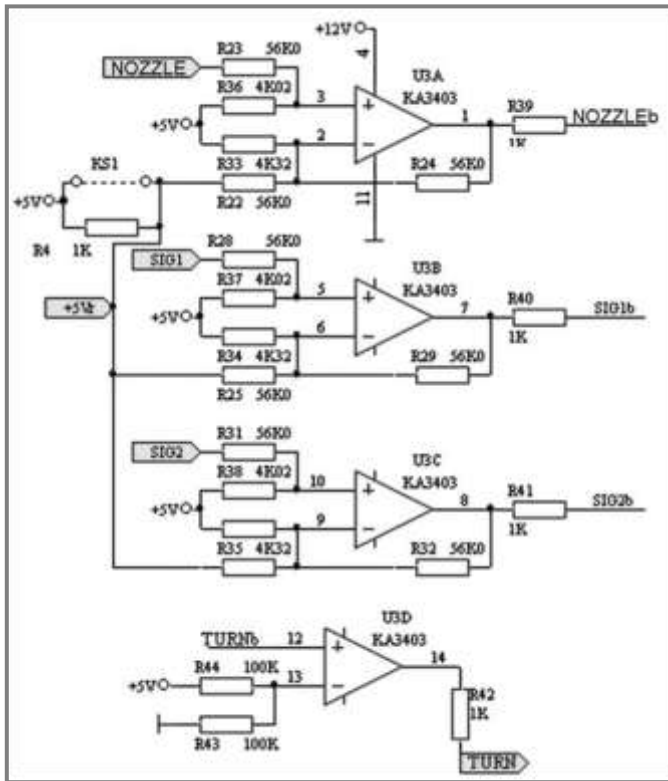


Figure 5: Dispense Controller Input/Output Interface to Fuel Dispensing Device



Figure 6: Dispense Controller and Keyboard Mounted on the Fuel Dispensing Device

The main components of the logger are: registration controller, communication controller, EEPROM and Real-Time clock.

Synchronisation (1octet)
Data logger address(1octet)
Command/Response(1octet)
DS=data size(2octet)
Data(Dsoctets)
CRC16(2octets)

Figure 7: Data Packet Structure in Main Communication

The first octet is used for synchronization, and denotes data packet start. The second octet contains logger address. It is useful in systems with more logger. The Command/Response octet represents the PC command or the logger response. The fourth and fifth octets contain number of following data octets. The last two octets are CRC16, used for data packet integrity check.

Second mode of data exchange between the logger and the PC is reports transfer. Reports transfer is performed in only one direction, from the logger to the PC. The communication controller sends these data as plain text, without PC request and without CRC16. The reports can be divided into next two categories:

- basic reports and
- detailed reports.

The communication controller always sends the basic reports. These reports contain information of each refuelling and other events, like the logger reset, dispense controller reset, exceed fuel dispensing limit, etc. The basic reports are formatted as bill text which can be directly printed



Figure 8: The Bill for One Refueling

The detailed reports provide information of all activities in the system. These reports are very useful for system maintenance. The detailed reports can be enabled or disabled. The data storage PC receives the reports and writes them to the log files. In the case of system failure, these log files can be used to recover the data. Possible system failures are data corruption caused by any reason, like software bug, incorrect data processing or hardware dysfunction [Lew, 10]. These log files are not intended for automatic data recovery. However, we built the software for this purpose, but we had no chance to prove and verify it in practice yet. Having in mind, that this kind of recovery is off-line operation, the software can be improved at the time we need it [11].

More details on the data logger can be found in [13].

The PC software in this system serves to:

- Create Folders file, Configuration file and Privileges file in the logger's EEPROM;
- Updates records into Privileges file;

- Retrieving data from Transaction file, formatting them and writing into database;
- Provide various reports of card privileges, performed fuel dispensing per day, cumulate fuel dispensing, etc.

The PC software updates privileges records for all cards, usually once per month in order to refresh fuel quotes. If necessary, the software can update the record of only one card.

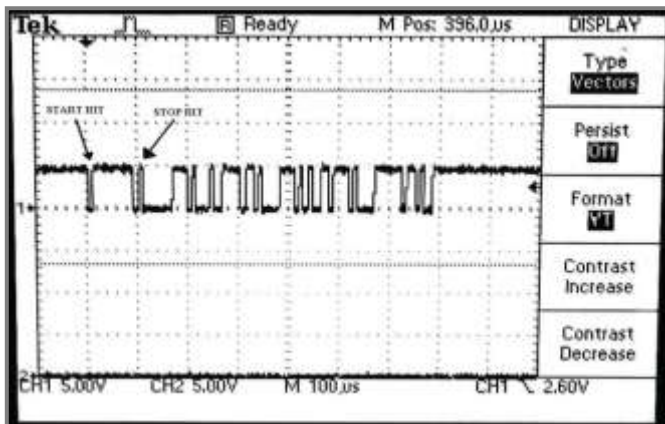
### 2.3. 1W232 Communication Protocol

Communication between microcontrollers in logger is achieved by 1W232 protocol. The 1W232 is our proposal of new simple one wire inter-chip communication protocol. In each one wire protocol only one wire is used for data exchange [Hersent et al., 4; Chabanne et al., 5]. In SensorPath and Dallas 1 Wire protocol bit value is determined by pulse duration. For reliable bit detection, these protocols define huge timing tolerances. Thanks to that, SensorPath and Dallas 1 Wire data may be received by devices without quartz stable clock [13], [14].

Main properties of our 1W232 protocol are:

- Data are transmitted in the form of RS232C signal (log 0=0V, log 1=5V).
- Multiple microcontrollers may share the same one wire bus.
- There is bus arbitration. In case of collision, the most priority message will be transferred.
- Communication can be tracked by simple protocol analyser.
- The timing must be precise with tolerances less than 3%. This is the only drawback in comparison with other one wire protocols.

Figure 9 shows 1W232 signal captured by scope. Figure 10 shows 1W232 bus with multiple microcontrollers connected. There is no Master on the bus. Any microcontroller can star communication.



1Wire microcontroller port may be in one of two states:

- 0 = port is set as output and delivers low voltage level,
- 1=Z= port is set as input (it is in Z state).

DATA line represents the 1 wire bus. When bus is free, DATA line is pulled-up, thanks to RP resistor. Any microcontroller can establish logical zero level on DATA line. However, logical 1 is obtained only when all microcontrollers sets Z state.

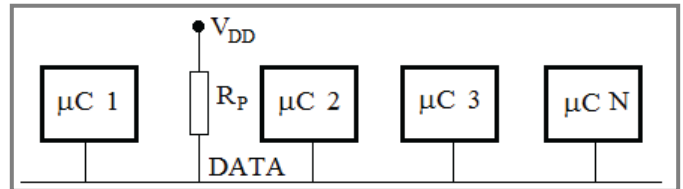


Figure 11: 1W232 Bus

The bus arbitration is solved without particular arbiter. To get bus, each microcontroller waits for free bus, actually it waits for logical “1” lasting appropriate time. After that it sends the message priority octet. One microcontroller can use various priorities depending on message urgency. For example, it can be allowed that

- first microcontroller uses priority levels 1, 5 and 7,
- second - 2, 4 and 6 and
- third - 0, 3 and 8.

No two microcontrollers are allowed to use the equal priority. Figure 12, 13 and 14 shows signals on DATA line, for different priority octets. Figures are captured on data logger 1W232 bus, by scope. After setting any bit on the bus, microcontroller reads back bus state and checks logical level. If it reads opposite logical level, microcontroller leaves the bus and switches to receiving mode. In case of two or more microcontrollers start transmission simultaneously, collision will occur. The collision is the state when microcontroller sets one level on the bus, but reads back opposite level. The lower priority microcontroller will detect collision during first octet, leave the bus and switch to receiving mode. Higher priority microcontroller won't detect collision and will continue with data packed transmission.

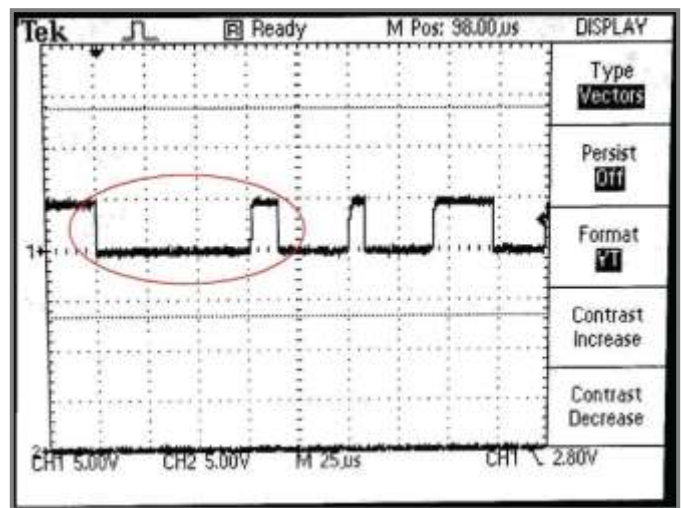


Figure 12: Signal of the Highest Priority Octet

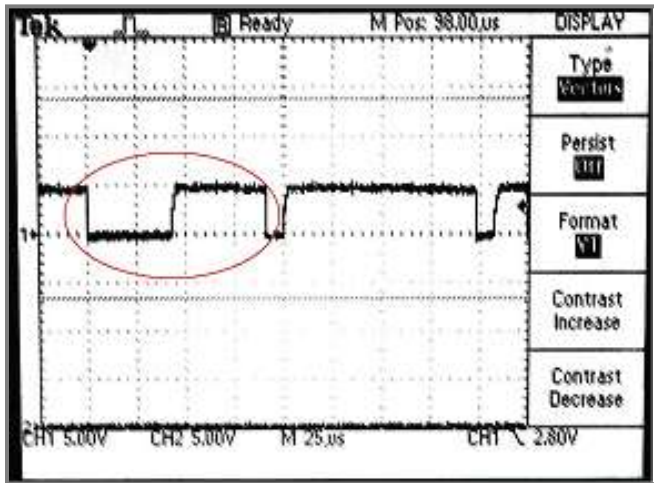


Figure 13: Signal of Moderate Priority Octet

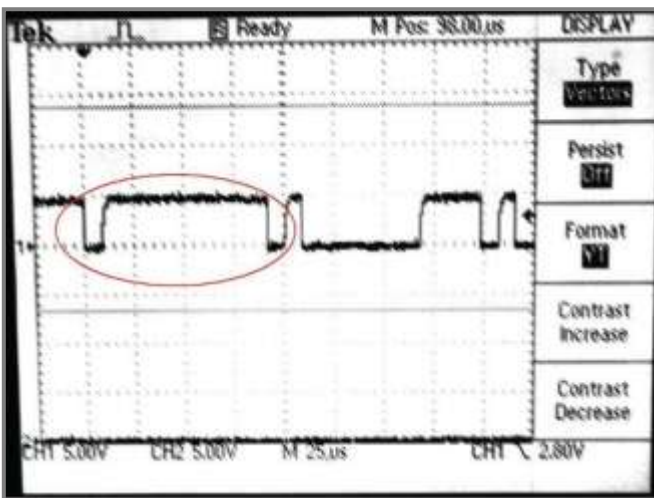


Figure 14: Signal of the Lowest Priority Octet

Protocol analysers are used to record communication between two or more sides, for debugging purpose. Thanks to 1W232 signal structure, taken from RS232C standard, 1W232 protocol analyser can be very simple. It could be realised with PC and extremely simple hardware (Figure 14). This hardware just inverts 1W232 signal, so the PC can capture it by RS232 serial port. Analyzers for other protocols could be quite complex and expensive



Before fuel dispensing starts, the user has to enter a car mileage (Figure 15), in accordance with mileage counter in his car. This data is later used to calculate the car fuel consumption per 100km.



Next, user enters identification card into controller field. The controller reads card identification number and sends it to the data logger. The data logger performs binary search through the Privileges file, for card identification number. If the logger finds it, sends card privileges record to the dispense controller. The controller displays: car plate, card ID, type and amount of fuel. Otherwise, if the logger didn't find card ID, the controller shows message: "UNKNOWN CARD". Upon reading the card and obtaining information of card privileges, the user pulls up fuel dispensing nozzle. The controller informs the logger about that. If the nozzle is regularly pulled up, the logger commands to the controller to turn on fuel dispensing device. During refuelling controller shows the amount of fuel issued.



System has built in protection against misuse and possible hardware failures. This is realised with predefined timeouts (in configuration file) for each refuelling step. Some timeouts are: time from card reading to start of dispensing, the longest pause in refuelling, overall time to complete refuelling, etc. If user exceeds any timeout, the controller will show message "FUEL DISPENSING LIMIT EXCEEDED" and turns off fuel dispensing device.



Dispensing can be continued by repeating the complete procedure, only. After fuel dispensing has been completed, user hangs up nozzle. The controller understands it as the end of the refuelling and sends adequate message to the logger. Based on this message, the logger records the fuel dispensing in transaction file and updates the card privileges:

- available fuel amount,
- number of transaction and
- date and time of the last change of the record.

Next, the logger sends updated card privileges file to the controller which displays it on the LCD. The message "ENTER A MILEAGE" returns on the controller's display, after a certain period of time, defined in Configuration file.

### III. FURTHER IMPROVEMENTS

In this system, users input car mileage data. The problem is, humans have limited time, attention and accuracy - all of which means they are not very good at capturing data. To override this weakness, we plan to design the microcontroller device which will be installed in each car. Among other purposes, this device will be used to automate mileage data transfer. The device will count odometer pulses and write it as mileage in Mifare card by NFC interface. In this improved system, human only brings the card from car to dispenser reader. The other solution we consider is the installation of long range UHF reader combined with UHF tags embedded in cars. We expect this will simplify usage of the system. The best system is the user invisible system. An UHF reader can read tags up to 5m with 32mW power emission. This is considerably below allowed 1W power according to safety standards [17; 18; Kitchen, 19]. Each of these possible improvements will reduce human interaction with system. Such system will meet concept of IoT, where things capture and process data without human.

## IV. CONCLUSION

This paper describes NFC system for automation company petrol station. The system is based on Mifare cards, dispense controllers, data logger, data storage PC with supporting software and database. There are two special solutions described in more details: the data logger realization with 2 microcontrollers and 1W232 communication protocol. The system functionality description is illustrated with the images captured during regular use of the system. At the end of paper further improvements are proposed.

## REFERENCES

- [1] S. Clarck (2011), "Webpage on NFC World".
- [2] V. Coskun, K. Ok & B. Ozdenizci (2011), "Near Field Communication (NFC): From Theory to Practice", 1st ed., English: Wiley, 2011.
- [3] MFRC531 ISO/IEC 14443 Reader IC, Product Data Sheet, Rev. 3.4, 2010.
- [4] O. Hersent, D. Boswarthick & O. Elloumi (2012), "The Internet of Things: Key Applications and Protocols", 2nd ed., English: Wiley.
- [5] H. Chabanne, P. Urien & J.F. Susini (2013), "RFID and the Internet of Things", 1st ed., English: Wiley-ISTE.
- [6] 8-bit Microcontroller with 16KB In-System Programmable Flash, ATmega16, Atmel Corporation, 2010.
- [7] ISO/IEC FDIS 14443 Identification Cards - Contactless Integrated Circuit Cards - Proximity Cards.
- [8] N. Lekic, Z. Mijanovic & R. Stojanovic (2011), "The Identification System for Monitoring of Teaching Activities", *EUROCON*, Lisbon.
- [9] G.K. Saha (2008), "Understanding Software Reliability Concepts", *IEEE Reliability Society Newsletter*, Vol. 54, No. 3.
- [10] K.S. Lew (1998), "Software Complexity and its Impact on Software Reliability", *IEEE Transactions on Software Engineering*, Vol. 14, Pp. 1645-1655.
- [11] Two-wire Serial EEPROM 256K (32,768 x 8) - AT24C256B, Atmel Corporation 2004.
- [12] N. Lekic, Z. Mijanovi & R. Stojanovi (2007), "The Data Logger Realized with Three Microcontrollers", *Applied Electronics*, Pp. 121-124.
- [13] Digital Out Temperature Sensor with Single-Wire SensorPath™ Bus, Texas Instruments, 2005.
- [14] AVR318: Dallas 1-Wire® master, Atmel Corporation, 2004.
- [15] LAP-321000C Logic Analyser. [Online]. Available: [http://studio16.booc2.co.za/index.php?option=com\\_content&view=article&id=80&Itemid=83](http://studio16.booc2.co.za/index.php?option=com_content&view=article&id=80&Itemid=83)
- [16] Beagle I2C/SPI Protocol Analyzer.
- [17] Assessment of inadvertent ignition of flammable atmospheres by radiofrequency radiation. Guide, BS 6656:2002, 2002.
- [18] Explosive atmospheres - Part 0: Equipment - General requirements, EN 60079-0:2009 6.6.1, 2009.
- [19] R. Kitchen (2001), "RF and Microwave Radiation Safety", 2nd Edition, Newnes.