Design of Simulator for Alarm Modules

Smita Kashid[1]
Electronics and Telecommunication
Dept , FRCRIT , Vashi.
Mumbai University, India
smita.kashid6@gmail.com e-mail

Prakarsha Saxena[2]
Electronics and Telecommunication
Dept , FRCRIT , Vashi.
Mumbai University, India
prakarshasaxena@gmail.com

Smita Chopde[3]
Electronics and Telecommunication
Dept, FRCRIT , Vashi.
Mumbai University, India
chopde.smita@gmail.com

Abstract:- For easy, comprehensive testing and evaluation of error conditions in a real time environment, now-a-days use of a simulator on software based platform for generating these conditions as well as testing before it is actually implemented on a commercial scale ,that would be manifolds of the current scenario .The inputs are assimilated , conditioned , in an testing environment are processed according to a set of parameters specified and designed according to the designer or the user of application. The objective of a simulator is to create a uncomplicated migration from a hardware sourced system to software based simulator where concurrent processes and evaluations are done forehand to comprehend, ameliorate the simulator ,accordingly these implementations to be done in the hardware where the core would reside perform the key task managements. At, the testing unit, the system is ignorant whether it is an actual system performing the operations and implementing the tasks or a simulator performing mimicking the hardware and conducting the actual tasks on a software. It could be comprehended as if a mobile is simulated on desktop and the end user is being asked to use this mobile on computer with all the similar features that the user would have an access in an actual mobile. So, here in the system simulator will assist the testing as well as checking the efficiency of the system and study of the complex hardware system.

Keywords: *Simulator, hardware, testing, migration.*

*****

I. INTRODUCTION

The system needs to communicate all the tasks and processes executed simultaneously, and hence employing various measures for implementing these tasks and processes. This communication should be a bidirectional one, not a unilateral but also be a broadcasting type of communication ,where the collision of information/data packets should not take place as any loss of data can inflict a lot of system's mainstream in turn a failure at multiple levels which would not be a viable option. This communication should take place on a guided media which could withstand a lot of interference from the environment and successfully delivering the acquired data to the receiving end. The fundamental units of the system are: the testing unit and simulator (which contain the alarm modules [blocks]).

Here, the simulator will mimic the conditions as desired by the user or the implementer of the system. The system will first accomplish the tasks with the accurate results to be acquired at the first implementation. Then, various error conditions are enacted and the required output should be processed by the system. As the actual hardware system is not present at the tester end, it will be much easier for the user to emaciate multiple erroneous condition for each alarm module in multiple blocks. The output here obtained here, thus can help in improvising the system to be methodical and concise every single time the system is executed. This will further help to set realistic expectations of the tasks that will be implemented in this simulator designed system. The paper further subdivides about the [2] platform used for simulation, [3] working of system, [4] communication in the system via the guided link, [5] insight to thee hardware, [6] conclusions.

Migrating Software to Hardware on FPGAs Russell A Klein Rajat Moona Indian Institute of Technology Kanpur. [1]

II. PLATFORM USED IN THE SIMULATION OF THE SIMULATOR

Qt also known as /kjuːt/ "cute" .This is a cross-platform application framework which can be widely used for developing various application softwares that could be executed on multiple softwares as well as hardware platforms along with not even introducing a single variation or change within the underlying codebase. It is still being used as a native application with native capabilities in addition to speed. The current development is done by The Qt Company and the Qt Project under open-source jurisdiction, involving various individual developers along with firms working to advance Qt in all aspects and in accordance to the requirement of the application .Qt is accessible with both commercial and open source GPL 2.0, GPL 3.0, and LGPL 3.0 licenses.

Qt's main aim is for development application software along with graphical user interfaces (GUIs).Not only programs with GUI application but also programs without a GUI application can be enhanced. Use of various tools which can be stated as command-line tools and consoles for multiple servers. An illustration of a non-GUI program can be given using Qt is the Cutelyst web framework. GUI programs are created within the Qt which can have an aboriginal-looking interface, in this case Qt can be categorized as a widget toolkit.

Qt utilizes the standard C++ along with all it's extensions incorporating multiple signals and slots which disentangles the handling of events, which aids in conglomeration of both GUI and server applications which receive their own sets of event data and their processing should be done accordingly. Qt is designed to support many compilers, including the GCC C++ compiler and the Visual Studio suite. Qt is also capable of Qt Quick, which includes a declarative scripting language known as QML which permits using JavaScript to accomplish the logic of the code. With Qt Quick, rapid application development for mobile devices has become an easy task, despite the logic could also be written in native codes to attain the best possible outcome. Qt's usage within several other programming languages can be stated via various language bindings. It executes on vital desktop platforms .This has an extensive internationalization of support.

C++ GUI Programming with Qt4 (Prentice Hall Open Source Software Development) [2]

Blanchette, Jasmin; Summerfield, Mark (June 2006). "A Brief History of Qt". C++ GUI Programming with Qt 4 (1st Ed.). Prentice-Hall. [3]

III. MODELLING OF THE SYSTEM

The system encompasses a monitoring system, testing unit, physical connection via guided media here, RS485 cable. The simulator which will mimic the hardware is developed. The twenty eight alarm modules are designed as well as their input voltage or current limits. These are designed according to system designer's requirements.

The system can accordingly be designed as illustrated in the block diagram:

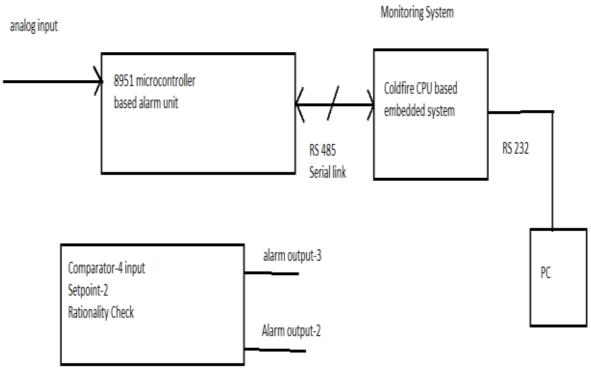


Fig1.System block diagram

The analog inputs are processed by the microcontroller (8951).The monitoring system is connected via RS485 serial link. The monitoring system consists of a coldfire CPU based embedded system. This monitoring system is also our designated simulator. A total of 28 modules can be stacked in a rack. The results are processed in the computer. Here, there is a requirement of a converter from RS485 to RS232. This procedure is incorporated within the system as the conventional data transfer in PCs take place via RS232 guided link.

The comparator compromises of four input voltage levels and two output signals one of them directed to display unit of system and the other testing and implementation of procedures according to their needful applications.

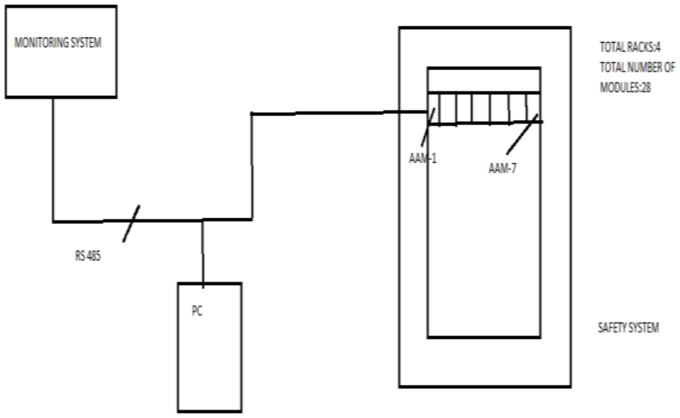


Fig2.Alarm modules and the communicating system

RS-485 and MODBUS Protocol:
The communication establishment is done using RS-485 communication cable which functions over the MODBUS protocol. Rs-485 is half duplex, balanced transmission line. It has ample number of advantages over the tradition RS-232 cable. RS-485 utilizes differential mode of communication consisting of about 32 drivers and 32 receivers with baud rate of 10M allowing transmission distances up to 1.2 km. In RS-485 there is based on the concept of single master governing multiple slaves. Only master has the authority to initiate a conversation with a message called as 'query' asking for a particular slave. If slave does not respond within the stipulated time the master will terminate the conversation.

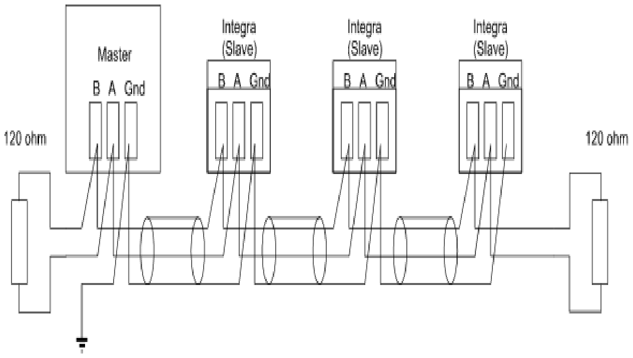
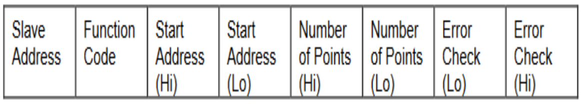


Fig3.Master and slave terminology in RS-485

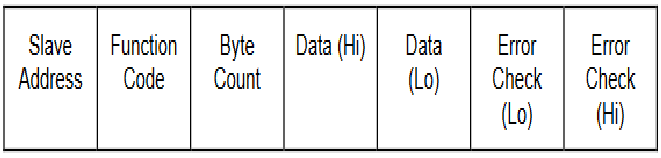
Master Query Format:



Slave Address	Function Code	Start Address (Hi)	Start Address (Lo)	Number of Points (Hi)	Number of Points (Lo)	Error Check (Lo)	Error Check (Hi)

Fig4.master query format

Slave response format:



Slave Address	Function Code	Byte Count	Data (Hi)	Data (Lo)	Error Check (Lo)	Error Check (Hi)

Fig5.Slave response format

"Modbus FAQ". Modbus. Modbus Organization, Inc. Retrieved 1 November 2012. [4]

RS-485 uses MODBUS protocol to establish a communication between to devices. It uses two serial transmission modes i.e. ASCII mode and RTU mode. For error correction and detection error checking methods, CRC checking, parity checking techniques are used. This makes the System bug free.

Working of MODBUS PROTOCOL at application layer:

MODBUS is an application layer and data link layer messaging protocol .At the physical layer, It uses variegated physical interfaces like RS-232,RS-485.This protocol is a serial communication protocol i.e it is used to transmit the information serially between the two electronic devices with the use of programmable Logic controllers(PLCs).It was first developed by Medicon nowadays called as Scheinder electric. MODBUS protocol is designed by considering the industrial applications in mind and is royalty free i.e. it is an open protocol Modbus protocol comprises of a single master and multiple devices connected to it are called as Slave. Masters communication is bidirectional while Slave communication is unidirectional i.e Slave can only answer to the master. MODBUS protocol has various versions including modbus RTU, Modbus ASCII, Modbus TCP/IP or modbus TCP, modbus RTU/IP, modbus over UDP, modbus plus etc.

MODBUS presence at data link layer:

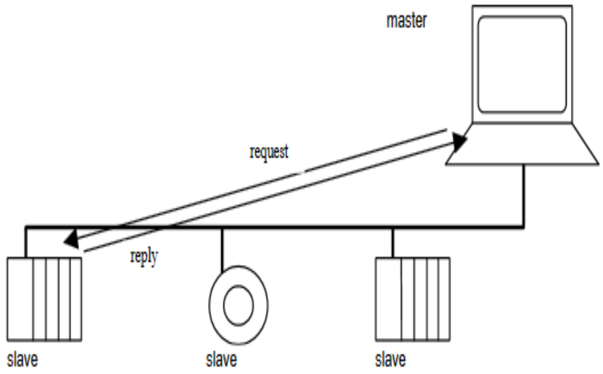


Figure 6(b) Unicast Mode

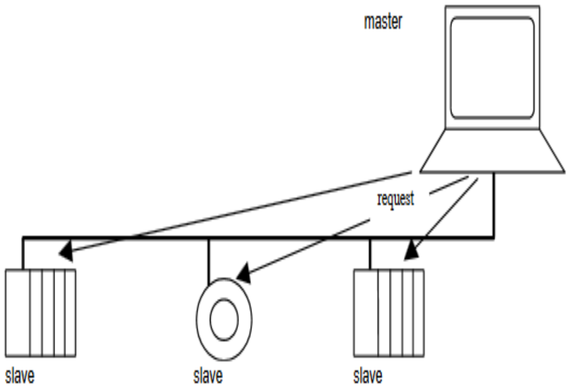


Figure 6(b) Broadcast Mode

Fig6.unicast and broadcast mode.

In this only master is connected to the bus at the same time, and one or more (limit is 247) slaves can be connected to the bus. The communication between master and slave can be established using two modes:

1 .unicast mode

2. broadcast mode.

Modicon Modbus Protocol Reference Guide PI–MBUS–300 Rev. [5]

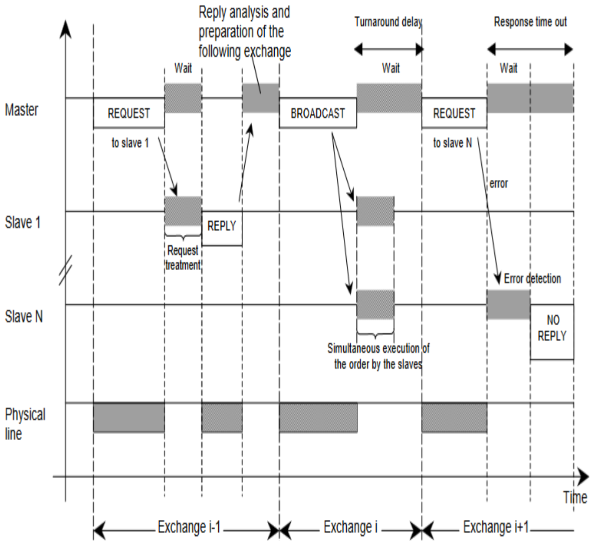


Fig7. Timing diagram of master and Slave

NS16c2552 IC TEXAS INSTRUMENT:

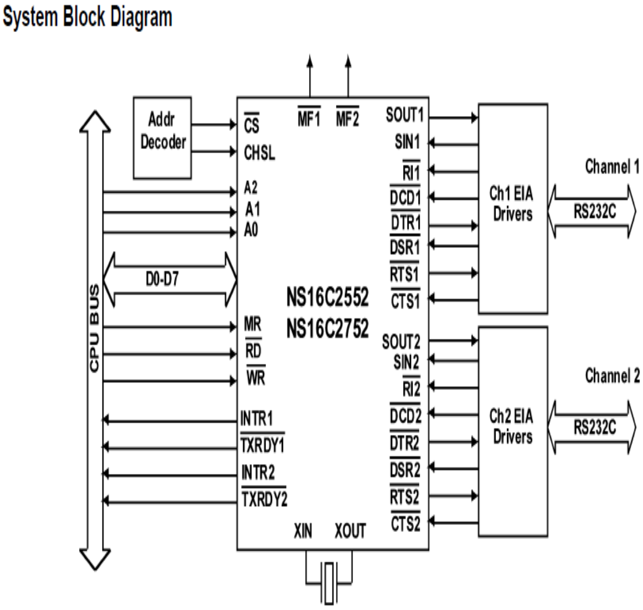


Fig8.Block diagram of system

It has innumerable features such as high data rate, dual independent UART (universal asynchronous receiver/transmitter) ,IrDA v1.0 wireless infrared encoder/decoder. It is driven by the IC MAX1480.This integrated circuit is provided with two serial channels which are completely independent. Each channel can be operated in FIFO(first input first output) Mode(on chip transmitter and receiver).DMA (direct memory access)controller is used to improve the CPU processing bandwidth i.e. the data transfer rate between DUART and CPU can be improved using DMA controller. Two different register sets are available for two different channels. Special registers are available in IC16c2552 are THR(transmit holding register), IER(interrupt enable register), LSR (light status register), MCR (modern control register), MSR (modern status register), SCR (SCRATCHPAD REGISTER).

The footprint as well as the functions of this integrated circuit are reconcilable with the PC16552D, while new features are added to the UART device. The NS16c2552 can be interchanged with NS16C2752. Various error conditions such as parity, overrun, framing, or break interrupt are provided.

MAX1480EA/MAX1480EC/MAX1490EA/MAX1490EB:
The driver

The MAX1480EA/MAX1480EC/MAX1490EA/MAX1490EB drivers which are comprehensive, electrically secluded, thus providing an effortless RS-485/RS-422 data communication interface solution within a composite microcircuit. The RS-485/RS-422 I/O pins furnish protection against ±15kV electrostatic discharge (ESD) shocks, without any other latch up conditions apparently being observed. It comprises of transceivers, optocouplers, as well as a transformer that yields a competent interface encompassed in a standard DIP (dual input package) package. A solitary +5V supply is delineated on the logic side that powers dual sides of this interface.

The MAX1480EC/MAX1490EB are prescribed as the reduced-slew-rate drivers which curtail EMI(electromagnetic induction) and diminish reflections caused due to irringly terminated cables, allowing error-free data transmission at data rates up to 160kbps. The MAX1480EA/MAX1490EA driver slew rate is not limited, allowing transmission rates up to 2.5Mbps.The MAX1480EA/MAX1480EC are designed for half-duplex communication, while the MAX1490EA/MAX1490EB feature full-duplex communication.

NS16C552V Texas Instrument(Datasheet). [6]

MAX1480A/MAX1480B/MAX1480C/MAX1490A/
MAX1490B Texas Instrument (Datasheet) [7]

IV. CONCLUSION

The migration from hardware to software is attained via Qt cross platform application. This in turn helps to reduce the system’s infrastructure, cost of manufacturing the system and the space occupied by the system. The communication link was established using the RS-485 possessing various innumerable features and advantages over the conventional RS-232 cable.

While migrating from hardware to software requires a comprehensive knowledge about the system hardware and software should be pre requisite. A proper design should be known pre handedly and the outcomes to be expected are thoroughly tested under various circumstances which includes multiple error condition.

Qt is a supportive cross platform which helps in development of the simulator which enacts as a hardware as per the required designing of the actual real time system.

ACKNOWLEDGEMENT

We would also be grateful to Mr.Tahir Khan for making familiar with the overview of the project.

The opportunity to develop such a testing and simulating this system helped gain an insight about how previous technology can still be existent and usefully harnessed in a well-developed commercial project within a reputed organization.

V. REFERENCES

[1] Migrating Software to Hardware on FPGAs Russell A Klein Rajat Moona Indian Institute of Technology Kanpur

[2] C++ GUI Programming with Qt4 (Prentice Hall Open Source Software Development)

[3] Blanchette, Jasmin; Summerfield, Mark (June 2006). "A Brief History of Qt". C++ GUI Programming with Qt 4 (1st Ed.). Prentice-Hall.

[4] "Modbus FAQ". Modbus. Modbus Organization, Inc. Retrieved 1 November 2012.

[5] Modicon Modbus Protocol Reference Guide PI–MBUS–300 Rev. j

[6] NS16C552V Texas Instrument(Datasheet)

[7] MAX1480A/MAX1480B/MAX1480C/MAX1490A/ MAX1490B Texas Instrument (Datasheet)