UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Faculty of Electrical Engineering

FRAMEWORK OF DISTRIBUTED CONTROLLING VIA SMARTPHONE FOR INDUSTRIAL AUTOMATION APPLICATION

Phua Ging Sheng

Master of Science in Mechatronic Engineering

2013

© Universiti Teknikal Malaysia Melaka

FRAMEWORK OF DISTRIBUTED CONTROLLING VIA SMARTPHONE FOR INDUSTRIAL AUTOMATION APPLICATION

PHUA GING SHENG

**A thesis submitted
in fulfillment of the requirements for the degree of
Master of Science in Mechatronic Engineering**

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2013

© Universiti Teknikal Malaysia Melaka

DECLARATION

I declare that this thesis entitle "Framework of Distributed Controlling via Smartphone for Industrial Automation Application" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	[illegible]
Name	:	PHUA GING SHENG
Date	:	3 MAC 2014

© Universiti Teknikal Malaysia Melaka

ABSTRACT

Programming language that is used in industrial is platform specific, when hardware is updated or replaced by a newer hardware, the entire binary software needed to be changed and modified. Software that has been written is not able to run on different hardware platform without recompiling the source code. Besides that, the same code that is executed on machine is not able to be executed on computer for any hardware simulation or testing. Machine running code is not compatible with computer running code, thus, different types of source code are needed for computer simulation and real time platform respectively. This research was focused on extending Smartphone and Java platform into automation environment where users are able to monitor and control the targeted system regardless of any location by utilizing internet. The main task of this research had been working on creating new framework of distributed controlling and computing via smartphone. Platform independent frameworks of classes that implement internet protocol were designed to provide interaction over client server paradigm and finally all frameworks were then integrated over a system which allows access to automation control, manipulate and monitor via smartphone over wireless internet network protocol. Through this approach, Smartphone and Java platform's extendibility and adaption in different environment with Java features inherited were proven successfully. As a result, Industrial automation can use Java to produce systems that perform more reliably, improve programmer productivity, time efficiency and even provide binary compatibility across hardware platforms. Using Java, when comes to hardware simulation, the same application class files that run on the target machine can also execute on a desktop computer for development and debugging due to its platform independent ability. It simplifies remote monitoring and controlling in several ways where its internetworking based language allows it to run on any web browser or any other networking platform. In the event of a hardware fault, a developer could also write a dynamic Java classes that monitor a set of sensors on the hardware for maintenance and troubleshooting. Not only can dynamic troubleshoot the behavior of the hardware, but it can also be used to patch existing code or add additional functions without stoppage or modification of machine. The ability to change a single source file, recompile it and then dynamically download only the changed class file and link it with a running JVM provides added flexibility for software maintenance. Hardware independence ability allows developers to simulate the run-time environment on a desktop computer, greatly reducing development time and debugging is required only on a single version of source code, once for all platforms. This reduces the project overhead and maintenance by greatly simplifying project build configurations, reducing coding errors and simplifying bug tracking and change requests.

© Universiti Teknikal Malaysia Melaka

ABSTRAK

Bahasa pengaturcaraan yang digunakan dalam industri adalah platform tertentu, apabila perkakasan dikemaskini atau digantikan dengan perkakasan baru, perisian keseluruhan perlu diubah suai. Perisian yang telah ditulis tidak dapat berfungsi di platform yang berbeza tanpa dibina semula. Selain itu, kod yang sama yang dilaksanakan pada mesin tidak dapat dilaksanakan pada komputer untuk mana-mana perkakasan atau simulasi ujian. Kod mesin tidak serasi dengan kod komputer, oleh itu , pelbagai jenis kod sumber diperlukan untuk simulasi komputer dan platform sebenar.Kajian ini adalah untuk melanjutkan telefon pintar dan platform Java di dalam automasi dan pengguna boleh memantau dan mengawal apa-apa di mana-mana dengan hanya menggunakan internet. Tumpuan utama kajian ini adalah mereka rangka kerja baru untuk pengawal teragih dan pengkomputeran melalui telefon pintar. Kelas-kelas rangka kerja platform berdikari yang berasaskan protokol internet telah direka untuk menyediakan interaksi paradigma klien pelanggan. Semua rangka kerja dan perisian telah diintegrasikan ke atas sistem yang membolehkan akses kepada kawalan, pantauan, dan manipulasi automasi menggunakan telefon pintar melalui rangkaian protokol internet tanpa wayar. Melalui pendekatan ini, peranti tertanam isirong dan kesesuaian platform Java dan penggunaannya dalam persekitaran dan bidang yang berbeza dengan ciri-ciri Java diwarisi telah berjaya dibuktikan. Hasilnya, automasi perindustrian boleh menggunakan Java untuk menghasilkan sistem yang lebih tepat , meningkatkan produktiviti pengaturcara, kecekapan masa dan juga menyediakan keserasian binari di seluruh platform perkakasan. Kod aplikasi yang berjalan pada mesin sasaran juga boleh digunakan pada komputer untuk pembangunan dan debugging kerana kemampuannya untuk berfungsi di platform yang berbeza. Sekiranya suatu kesalahan perkakasan, pemaju juga boleh menulis dinamik kelas Java yang memantau satu set sensor pada perkakasan bagi penyelenggaraan dan penyelesaian masalah. Ia juga boleh digunakan untuk menampal kod yang sedia ada atau menambah fungsi-fungsi tambahan tanpa pemberhentian atau pengubahsuaian mesin. Keupayaan untuk menukar fail sumber tunggal, susun semula dan kemudian dinamik turun hanya fail kelas berubah dan menghubungkannya dengan JVM berjalan menyediakan fleksibiliti tambahan untuk penyelenggaraan perisian. Keupayaan platform bebas membolehkan pencipta mengurangkan penyelenggaraan, memudahkan projek konfigurasi, mengurangkan kod kesilapan dan memudahkan pengesanan pepijat dan perubahan permintaan.

© Universiti Teknikal Malaysia Melaka

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians and practitioners. They have contributed towards my understanding and thought. I would like to take this opportunity to express my greatest appreciation to those people who have given their help, supports, and guidance during research.

In particular, I wish to express my sincere appreciation to my supervisors, Dr. Tay Choo Chuan and Associate Prof. Dr. Zulkifilie Bin Ibrahim for advices, critics, and guidance. Without their continued support and interest, this thesis would not have been completed.

Besides, I would like to thanks to the authority of UTeM for providing me with a good environment and facilities. My sincere appreciation also extends to all my colleagues and others who have provide assistance at various occasions. Their view and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. Thanks again.

© Universiti Teknikal Malaysia Melaka

LIST OF TABLES

© Universiti Teknikal Malaysia Melaka

LIST OF FIGURES

© Universiti Teknikal Malaysia Melaka

© Universiti Teknikal Malaysia Melaka

© Universiti Teknikal Malaysia Melaka

LIST OF ABBREAVIATION

JVM	-	Java virtual machine
API	-	Application programming interface
IEC	-	International electrotechnical commission
PLC	-	Programmable logic controllers
PC	-	Personal computer
HMI	-	Human machine interface
SCADA	-	Supervisory control and data acquisition
MTU	-	Master terminal unit
RTU	-	Remote terminal units
DCS	-	Distributed control system
DAQ	-	Data acquisition
LAN	-	Local area network
IDE	-	Integrated Development Environment
RF	-	Radio frequency
JIT	-	Just in time
OOP	-	Object-oriented programming
OS	-	Operating system
GUI	-	graphical user interface

© Universiti Teknikal Malaysia Melaka

RMI	-	Remote method invocation
JME	-	Java micro edition
RTOS	-	Real time operating system
JEE	-	Java enterprise edition
RISC	-	Reduced instruction set computing
JCA	-	Java cryptography Architecture
JCE	-	Java cryptography Extension
AES	-	Advanced encryption standard
IP	-	Internet protocol
TCP	-	Transmission control protocol
HTTP	-	Hypertext transfer protocol
IDC	-	International data cooperation
OSI	-	Open system interconnection
QoS	-	Quality of service

© Universiti Teknikal Malaysia Melaka

TABLE OF CONTENT

© Universiti Teknikal Malaysia Melaka

© Universiti Teknikal Malaysia Melaka

© Universiti Teknikal Malaysia Melaka

© Universiti Teknikal Malaysia Melaka

© Universiti Teknikal Malaysia Melaka

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter discusses the whole thesis briefly as an introduction while sets the scene, research context, problem context and establishes why it is an issue worthy of research. The research question, aims, objectives, scope and context of the research were clearly identified and stated throughout the chapter. At the end of this chapter, each chapter are outlined briefly with introduction, literature review, research methodology and modeling design, integration and result discussion, output analysis and discussion, conclusion and recommendation.

1.2 Problem statements

Currently, there are limited smartphone based solutions for real time distributed control system especially in industrial automation domain. User is needed on site to monitor and control the system. Smartphone controlling regardless of any location is more practical and convenient, thus it is a perfect candidate for automation control solution. Currently, there are also limited Java based solutions for real time distributed control system. Programming language that is used in industrial do not have garbage collector and is platform specific (Ann Wollrath, Jim Waldo, Roger Riggs, 2012). These platform do not have any automatically memory optimizing or memory management

© Universiti Teknikal Malaysia Melaka

mechanism, thus, this will cause redundant and excessive memory allocation in the memory storage. When hardware is updated or replaced by a newer hardware, the entire binary software needed to be changed and modified. Software that has been written is not able to run on different hardware platform without recompiling the source code. Besides that, the same code that is executed on machine is not able to be executed on computer for any hardware simulation or testing. Machine running code is not compatible with computer running code, thus, different types of source code are needed for computer simulation and real time platform respectively. Besides that, code patching and modification require stopping of machine which then will affect the production yields. User is not able to write a real-time dynamic code to diagnose hardware failures in case of emergency or faulty. Users are usually forced into purchasing one brand of hardware or software from specific manufactures due to the platform specific features of programming style that used in industrial. The machine running codes are all tightly-coupled where a small little change of software or hardware required the modification of whole system or even sometime the need of integrating parts that will burden the performance.

The current version (Java 6 update 26) of the Java includes a garbage collector which will dislocate any unused memory automatically and platform independent ability (John Ferguson Smart, 2008). Industrial automation can use Java to produce systems that perform more reliably, improve programmer productivity, time efficiency and even provide binary compatibility across hardware platforms. Using Java, when comes to hardware simulation, the same application class files that run on the target machine or PLC can also execute on a desktop computer for development and debugging due to its platform independent ability. Once the code is completed, the class files can be executed directly on the target hardware for final testing and quality assurance. It simplifies

© Universiti Teknikal Malaysia Melaka

remote monitoring and controlling in several ways where its internetworking based language allows it to run on any web browser or any other networking platform. In the event of a hardware fault, a developer could also write a dynamic Java classes that monitor a set of sensors on the hardware for maintenance and troubleshooting. Not only can dynamic troubleshoot the behavior of the hardware, but it can also be used to patch existing code or add additional functions without stoppage or modification of machine. The ability to change a single source file, recompile it and then dynamically download only the changed class file and link it with a running JVM provides added flexibility for software maintenance. When that technique is used, software in an entire factory can be upgraded remotely, thereby avoiding costly on-site maintenance requests. Hardware independence ability allows developers to simulate the run-time environment on a desktop computer, greatly reducing development time. Another benefit is that debugging is required only on a single version of source code, once for all platforms. This reduces the project overhead and maintenance by greatly simplifying project build configurations, reducing coding errors and simplifying bug tracking and change requests. Users will no longer be forced into purchasing one brand of hardware or software as successful systems can operate on a multiplicity of platforms (hardware and operating system).

1.3 Research Objectives

The main goal of the research presented in this thesis is to investigate issues and approach pertinent to integration of Java, Smartphone and hardware and research implementation issues associated with their development. The thesis has the following objectives:

© Universiti Teknikal Malaysia Melaka

- Extend and adapt Java platform into real time distributed control application environment with Java features inherited.
- Design three platform independent frameworks of component that implement internet protocol which provide interaction over client server paradigm.
- Integrate all frameworks and hardware over an application which allows access to automation control via smartphone over wireless internet network.

1.4 Scopes of Research

Java is a programming language and is typically compiled to byte code (class file) that can run on Java Virtual Machine regardless of computer architecture with platform independent ability. Java was used to create a platform independent framework of classes that implement internet protocol which provide interaction over client server paradigm. These frameworks are library or class file that are compatible with any computer regardless any type of operating system. A computer was served as a centralized server before connecting to other automation devices. Server side framework that used to establish connection between client and automation was compiled into a server. Client side framework that used to control and manipulates the system was developed and compiled into the Kernel of the smartphone. Controlling devices are not limited to only smartphone, but as long as targeted devices has a build in operating system, then are compatible with the software that created in this research. In the demonstration or in this research, only one smartphone and one server were used for modeling. These frameworks and software was then integrated over the automation system which allowed access to automation control via smartphone over wireless internet network.

© Universiti Teknikal Malaysia Melaka

At the end, there appeared only two simplified parts which are client side and server side. A custom circuit for mimicking industrial automation part has been developed in this project for demonstration. Devices or peripherals such as motor, lighting, sound wave and signal related system consists of both analogue and digital signal was integrated in this project. Through this approach, kernel embedded device and Java platform's extendibility and adaption in different environment were proven successfully. Java features that inherited and utilized in this research are limited to object oriented, multithreaded, platform independence, automatic memory allocation, distributed and dynamic networking. The figure 1.1 below illustrates all overall integrated system.

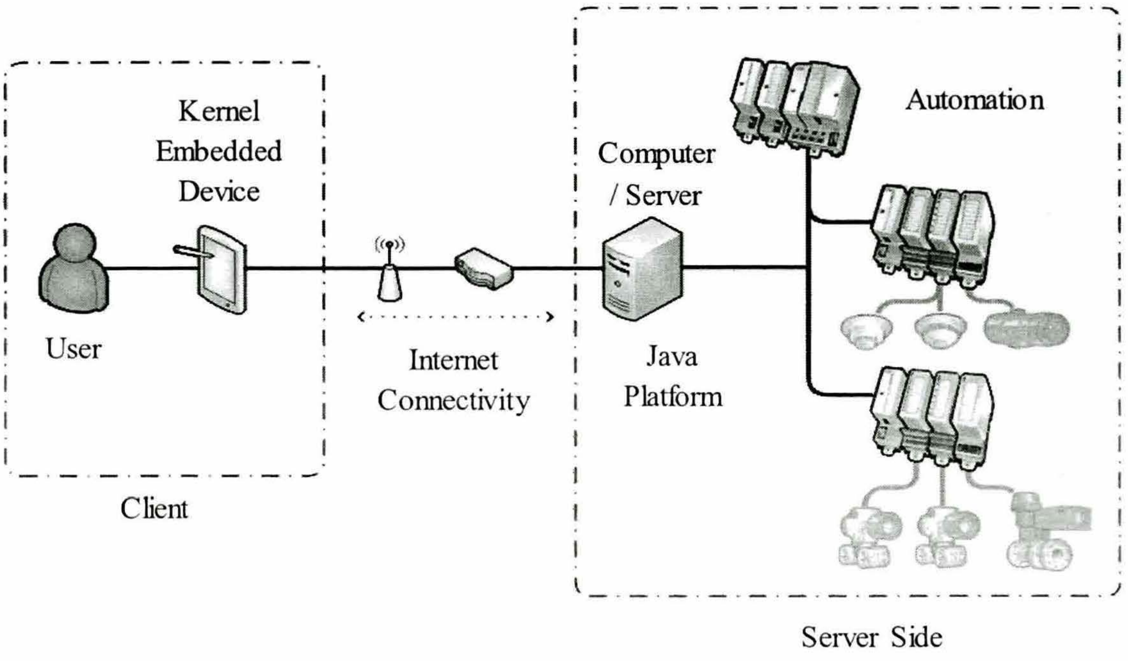

Figure 1.1 Simplified Single System Overview

© Universiti Teknikal Malaysia Melaka

1.5 Research Methodology

A research methodology is developed in order to achieve the research objectives. Initially a general flow was decided on how the user steps through that feature and determine the content and behavior that's needed for user to utilize this framework. For each feature, high-level task flow was diagrammed to map the necessary steps a user must take in order to complete a controlling task. As the flows are discussed among the team, details are identified and added to the diagram to show decision points, subtasks, etc. Once the team has a good idea of that features workflow and the necessary data for each step. The screen flow let the development know how many unique functionalities need to be created and also begin to identify some User Interface and controlling behaviors that helped to build consistency in the workflow, such as: certain types of errors send the user back to the originating page, or, search results of one can skip the list results page and send users to the details page. A good wireframes are invaluable in identifying common User Experience design patterns at both the page and widget level. From this information, our research team created a library of framework for reusable designs for any application, noting the appropriate interaction information for the behavior layer. All phases which include literature review, analysis, design, model development, validation, and simulation were used in order to achieve the research objective. Any future enhancements can then be plugged into the flow and evaluated for usability and consistency.

1.6 Thesis Contribution

One of the primary contributions of this research is for users that use this framework, This framework allow users to integrate the internetworking environment

© Universiti Teknikal Malaysia Melaka

with machine running devices and its ability to adapt on different hardware platforms without recompiling the source code. For the industrial automation industry, this has several benefits. First, as hardware is updated and newer hardware controllers replace older ones, no change in the binary software is necessarily required. For example, if a PLC that fully utilizes a 40-MHz Intel processor is upgraded to a 100-MHz Power PC processor, then the same binary class files can be used for both controllers, assuming that hardware ports and external interfaces are programmed for portability. Second, hardware independence allows developers to simulate the run-time environment on a desktop computer, greatly reducing development time. With the simple addition of low-level hardware simulation, the same application class files that run on the target PLC can also execute on a desktop computer for development and debugging. Once the code is completed, the class files can be executed on the target hardware for final testing and quality assurance. This saves valuable time during the development process, since standard software development environments may be used. Familiar debugging tools and development environments that are not tied to specific hardware platforms are easier for developers to learn and use; also, testing does not require the constant downloading and setup of the target PLC. Another contribution is that debugging is required only on a single version of source code, once for all platforms. This reduces the project overhead and maintenance by greatly simplifying project build configurations, reducing coding errors and simplifying bug tracking and change requests once this framework was implemented in user environment. The idea of remote controlling via smartphone has already proved to be effective in the computer server market, where high-end servers are remotely managed through a Web interface. Developers could also use dynamic downloading to diagnose hardware failures. For example, in the event of a hardware fault, a developer could write source code that monitor a set of sensors on the hardware,

© Universiti Teknikal Malaysia Melaka