International Journal of Smart Home
Vol. 10, No. 5 (2016), pp. 305-310
http://dx.doi.org/10.14257/ijsh.2016.10.5.28

A Novel Man-Machine Command System

GUO Hong-Tao

North China University of Water Resources and Electric Power,
Zhengzhou, Henan 450045, China
htguo2000@163.com

Abstract

In embedded system, man-machine command system is the main way for people to use the system. The command in man-machine system often needs upgrade. It is a hard work to upgrade the old command with the new version of the command. A novel man-machine command system proposed in this paper can be used to upgrade the command system in a smooth way. The method comprises: split the embedded system into different operating states, provide two command versions (old version and new version), and attach different privileges according to system operating states and command version. According to the current operating status of the system and the privilege of the command, it is determined whether or not to execute the command and whether or not prompt information about the command to be upgraded should be shown to the user. By using the method proposed in this paper embedded software can be a smooth upgrade. The old version command can continue to use, and the progress of the new version command replacing old version command will be a smooth progress.

Keywords*: Man-machine command; Smooth upgrade; System state*

1. Introduction

Functions of embedded software are often configured manually through the man-machine command [1][2]. Man-machine commands can be transferred via the serial port protocol or TELNET protocol or SNMP protocol sent by network management system [3-4].

Generally speaking the configuration information is sent to the embedded system under general form of the command. A collection of all the command hierarchy can be represented by a data structure of tree. And the current general configuration information can be saved in the form of text or binary code [5-7]. The hierarchy relation of the information is not good for people to directly see.

Another aspect, with the use of the system configuration information continuously enhancing and expanding, there might be a need to have a new command CMD2 to replace the old command CMD1. Namely CMD2 is the new version of CMD1. People's habits are not easy to change to the new command, so it is difficult to achieve a smooth upgrade software system.

In this article a novel man-machine command system is proposed to make the upgrade of software system smoother.

2. A Novel Man-Machine Command System

The novel man-machine command system comprising:

Set different privilege levels to different commands considering the operating state of different commands and the different version of the commands. Whether or not a command will be executed, depends on the current operating status and privileges of this command.

Furthermore, the operating status include: starting up status and the operating status.

The privilege level information includes: visibility, and the ability of executable;

Visibility means: to get help for that command through online help system;

For example, at the command line, type HELP or '? followed by the keyword of the command to see the help for that command and its associated parameters. The privilege of the command determines whether the command can be seen, and whether the parameters corresponding to the command can be seen, and whether the upgrade information of the command can be seen.

The executable ability means: This function of the command stand for can be accomplished by delivering the command to system.

furthermore, the command privilege information includes at least the following categories:

Privileges P1: The command is visible and executable in the system startup phase and the operating status.

Privileges P2: The command is executable and not visible in the system startup phase.

Privileges P3: The command is visible and not executable in the system operating status.

Privileges P4: The command is not visible and not executable in the system operating status.

Furthermore, set the upgrade prompt of the command, comprising: this command will be abolished; the command has been abolished; alternative command of the command.

Furthermore, when the command does not need to upgrade, the privileges of the command will be set to privilege P1. When the command needs to be upgraded, the privileges of the command will be set to privilege P2. That is, the command can run normally, but the help information of the command could not be provided by the command-line help system. In addition, for the command with privilege P2, prompt information will show after the command was successful executed. In general prompt information includes: prompts the user the command will be abolished and the like. And the privilege P2 is only used for system startup phase.

Furthermore, when the command needs to be upgraded, its privilege can also be set to P3. That is, when user wants get help information byhelp system, he will get the information that this command is abolished and the new version of the command will also be given to the user. And if the user enters the command, the command will not be executed, but same help prompt information for the command will be given to user. This permission P3 is used only for the system in running stage.

Where, P2, P3 privileges can be used to set to the command, which will be discontinued, but is still saved in the configuration file of the embedded software. Namely on the one hand, in the command line cancels the execution permission for the command, on the other hand, at boot time the command can be executed according to the saved configuration file. With privilege P2 and privilege P3 the embedded software can be upgraded smoothly. After the embedded software has been upgraded, the configure file will be saved in the new form with the new version of the command, discard the use of old version of the command.Then the privilege of the old version of the command will be set to P4. That is, if the command is entered, it will not be executed successfully whether the system is in startup phase or in normal operation phase.

It should be noted that, under the P3 and P4 two privileges, help information system gives difference prompts. The former refers to the visible, namely help information can be get through help system; the latter refers to the invisible, namely help information can only be get by the by-product of the command execution.

Furthermore, the command script is stored in XML (eXtensible Markup Language) format in the XML script file. The XML script file contains privileges of commands and the upgrade message for the commands.

If a command has a new version, when the configuration is saved, command will be saved in new version format. In the system boot process all the commands in the configuration file will be executed, and the command's privilege will be got. If a command in the configuration file has a new version, the command will be save in the formation of new version to replace the old formation of old version.

Furthermore, by using the method of dynamic loading configuration scripts, the new version of configuration scripts can be packaged into a new version of embedded software. When the new version of embedded software is used to upgrade the embedded device, the new version of configuration scripts can be used to upgrade the configuration scripts file. In the method proposed in this paper, configure scripts is save as a XML script configuration file. The command configuration file can also be dynamic loaded into memory of embedded device independently to replace the old version command configuration file stored in the embedded device.

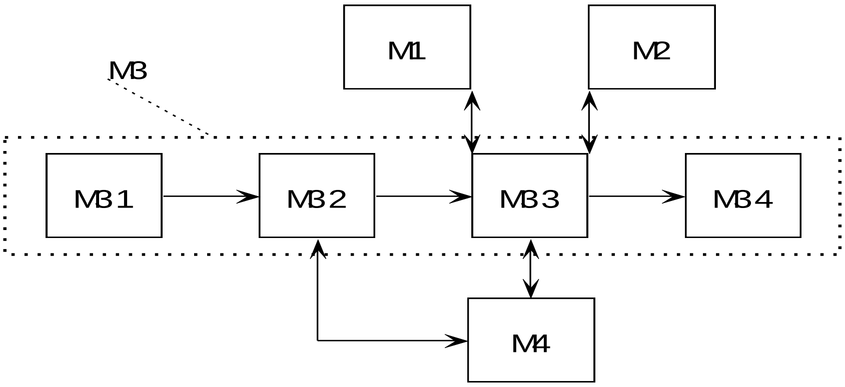

Figure 1. Logical structure of the system

The Logical structure of the man-machine command system propose in this paper is showed as following in Figure 1. The system includes interactive human-machine interface module M1, configuration file reader module M2, command parsing module M3 and execution module M4. The command parsing module can be further divided into: module M31 to read XML configuration script file, the command tree control module M32, interpretation module M33, database module M34.

Wherein each module functions as follows:

Module M1: responsible for reading and user input and giving feedback to user.

Module M2: responsible for reading information from the configuration file and writing information to the configuration file.

Module M3: responsible for parsing the semantics of the configuration script gotten from the command line or the configuration file.

Module M4: responsible for executing command and providing feedback of the execution of the command to command parsing module.

Module M31: responsible for reading XML configuration script file from the command script file and updating the XML configuration script file with the new command version. The privilege of the command, and the prompt information for update are stored in the XML script file.

Module M32 is used to generate command tree based on information read from the XML script file; and to provide for the interpretation of the lexical command module based on parsing.

Module M33: is used to explain the semantics of the command being executed according to the content of the command tree control module, and then to request M4 to execute the command. After the command is executed, the relevant information will be stored configuration information database.

Module M34 is used to store configuration information of the running system.
As it is showed in Figure 1, the process of the man-machine interface system includes the following steps:

Step 1: start-up phase XML script file reader module M2 reads the command structure information from XML configuration script file, and the command structure information will be transferred to module M32.

Step 2: command received from the user command line interface or read from a command script file will be sent to Module M33.

Step 3: module M33 will do job of command input lexical analysis, semantics parsing with the command structure information got from module M32. And the module M33 will determine whether the command can be executed according to the privilege of the command stored in the command tree and the current running state of the system. If the command can be executed, a message including the parameters and privilege information of the command will be sent to the execution module M4. If the command cannot be executed, the feedback information will be transferred to the user according to the privilege information stored in the command tree.

Step 4: execution module M4 will execute the command after M4 receive message sent in Step 3 from M33. And the results of the execution will be performed back to the interpretation module M33;

Step 5: to upgrade the software running in the embedded system, reboot of the embedded system is often needed. In the boot process all the command in the command script file will be executed. If there is a command to be updated, the command will be stored in the new version formation after the boot is done.

Step 6: command configuration file upgrade without system reboot is often called online configuration upgrade. During the process of online configuration upgrade, the configuration is loaded to the running system, and the command tree can be upgraded by a command entered from user interface command line. And during the upgrade process of the command tree, it will be locked to prevent abuse.

For example, when setting up command authority, there is no help information and no feedback information for the execution in the initial start-up phase. During software upgrade, if we need to replace the command CMD1 with command CMD2, assume that the software versions are V1, V2, V3, then privilege Pl can be set to command CMD1 in version Vl , privileges P2 + P3 can be set to command CMD1 in V2 version, and privilege P4 can be set to command CMD1 in version V3. In version V3the configuration file does not exist in the command, and the system will not actually execute the command, but the user will be prompted to upgrade later substitute command.

The processing procedure is showed in Figure 2.

Step 201: read string configuration commands lexical, syntactic analysis.

Step 202: resolve a configuration command to be processed.

Step 203: get privilege of the command resolved in the above step.

Step 204: determine whether the command is a command running for help, if it is step 205 will be the next step, otherwise Step 207 will be the next step.

Step 205: whether or not the command has the privilege to show the help information will be determined. If the command has the privilege, the Step 206 will be the next step. And in other cases, Step 202 will be the next step.

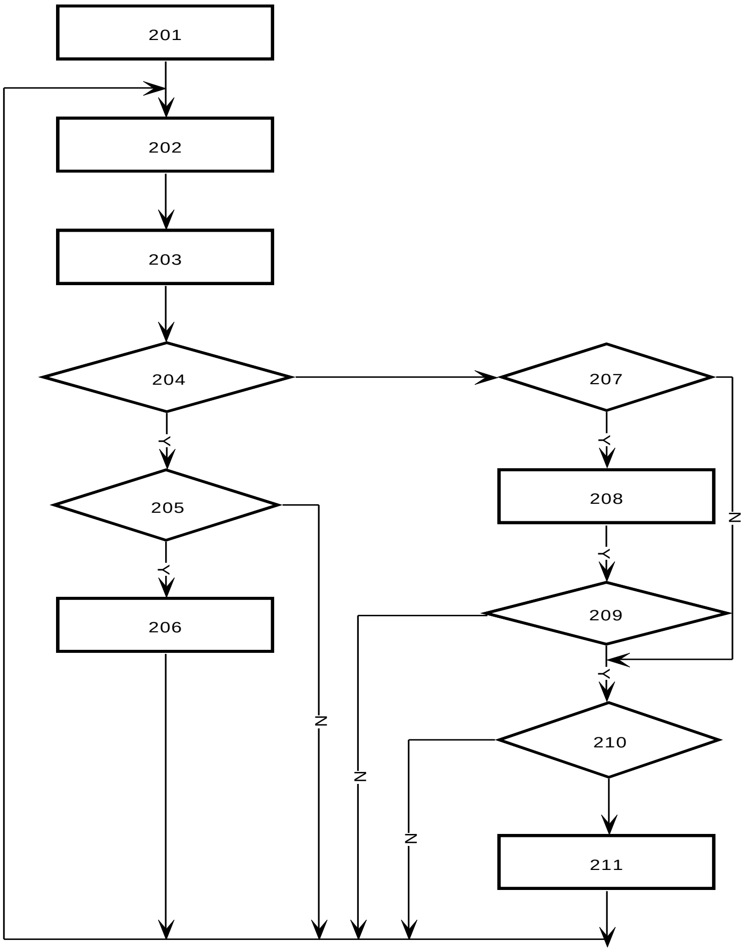

Figure 2. processing procedure of the system

Step 206: display help information about the command, and the return to Step 202;

Step 207: judge whether or not the command has the privilege to be executed. If the command has the privilege, Step 208 will be the next step. If the command do not have the privilege, Step 210 will be the next step.

Step 208: execute the command and then move to the next Step 209;

Step 209: it is determined whether the program is in the boot phase. And if it is in boot phase, feedback for the prompt to upgrade will not be transferred to the user and step will move to the 202. If it is not in boot stage, move to Step 210;

Step 210: judge whether or not the command has a prompt for new version of the command. If not then return to Step 202, if there is a new version command, next step will be Step 211;

Step 211, give the new version command information to the user, and then next step will be Step 202.

3. Conclusions

This method has been verified using C language under embedded Linux system. By using the method proposed in this paper, different privileges have been set to the command, according the system status and the version of the command. And in the execution phrase of the command, command execution will act in different way. Thus the system proposed in this paper can provide a user-friendly man-machine interface for user to upgrade the system smoothly.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under grant No.U1304606, and the Science Technology Development Project of Henan Province under grant No.132102210555. It was also supported by Foundation of Henan Educational Committee under grant No 14A520010.

References

[1] K. Qian, H. Yang and J. Niu, "Developing a gesture based remote human-robot interaction system using kinect," International Journal of Smart Home, vol. 7, no. 4, **(2013)**, pp. 203-208.
[2] Y. Kim and W. C. Yoon, "Generating task-oriented interactions of service robots," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 8, **(2014)**, pp. 981-994.
[3] S. Watanabe, "Experimental verification of power-limiting brake: Energy-saving train operation assistance," IEEE Transactions on Industry Applications, vol. 134, no. 8, **(2014)**, pp. 767-775.
[4] C. Rascon, "Integration of the multi-DOA estimation functionality to human-robot interaction," International Journal of Advanced Robotic Systems, vol. 12, **(2015)**.
[5] J. Cai and Y. Li, "Human-machine interaction system of sEMG based on DBN," Huazhong Keji Daxue Xuebao (Ziran Kexue Ban)/Journal of Huazhong University of Science and Technology (Natural Science Edition), vol. 43, **(2015)**, pp. 74-77.
[6] P. Ka, "Development system of script command line", C.N. Patent CN1967475A, **(2007)** May 23.
[7] W. Tao, "Method for command line interface authority classification and system", C.N. Patent CN101110702 (A), **(2008)** Jan 23.