Design and Implementation of G-Code Interpreter Based on QT

Guang-he CHENG¹, Shuai ZHAO²,* and Rui-rui SUN¹

¹Shandong Computer Science Center (National Supercomputer Center in Jinan), Shandong Provincial Key Laboratory of Computer Networks, Jinan 250014, Shandong, China
²Qilu University of Technology, Jinan 250353, Shandong China
*Corresponding author

Keywords: Open CNC system, G code interpreter, SQLite database, Qt4.

Abstract. This paper introduces a design of G code interpreter based on Qt. Firstly, it introduces the overall structure of the G code interpreter design. It introduces the regular expression as the tool of lexical, grammatical and semantic analysis, and introduces the SQLite database which is used to store the processing data. And then it explains the realization of the interpretation of the G code. Under the Qt4 environment, we develop the graphics applications, it not only has the module interface friendly, simple operation, easy transplant and a good human-computer interaction, but also has very high engineering application value.

Introduction

Compared with the traditional numerical control system, the embedded open CNC system has the advantages of small volume, low cost, low power consumption and stable system[1]. In the embedded system software, Qt/Embedded as a cross-platform GUI toolkit, which directly through the Qt API and Linux I/O interaction, has a higher operating efficiency. The C++ class mechanism, who encapsulates the corresponding operation functions, plays a protective role to prevent the destruction of external data that caused by the instability[2].

The program written in Qt has good portability and wide adaptability. So in other different operating system platforms, We just need to recompile the code to re-use again, greatly reducing development time and complexity.

CNC machine tools usually use the G code to describe the machine tool processing information, such as tool path, the choice of coordinate system, the opening of the coolant and so on. The main function of the G code interpreter is to interpret the NC code as a data source that can be recognized by the CNC machine or machine tool drive system and then communicated to the servo system.

Taking into account the limitations of embedded itself, We need to interface to each function module design, in which the interpretation of CNC machining instructions part of the entire human-machine interface design focus. G code interpreter design quality, which directly affect the performance of numerical control system. This paper describes the NC code interpreter design and functional realization.

G-Code Interpreter Design

Whether it is hand-written or automatically generated NC code program, we need to first check the correctness of the program, and then it can enter the computer to ensure the accuracy of processing. In the numerical control system, the interpreter as a separate module, whose input is the G code file and whose output is the cutter compensation and interpolation program, can identify the data flow. The CNC machining instruction code interpretation function module includes the following four parts:

1) Lexical analysis sub-module: Using the GRETA regular expressions[3] to match the string format first, and then extracting each word from the string. Lexical check is based on our keyword table[4], which is to check the legal of keywords and the data behind the keyword format and scope is
correct. If it has the wrong problem which is passed to the error handling module, the error is recorded not real-time being to deal with and continue the behind step. As shown in Figure 1:

![Figure 1. Lexical analysis sub-module.](image)

(2) Grammar analysis sub-module: Based on the lexical analysis, through the split function in Qt, it divides a line of code by a space to obtain a series of words into the array split[]. These words are parsed to see if they conform to the NC programming rules and the machine actions could be carried out by the NC code. If an error occurs, the error will be passed to the error handling module, it will be recorded, not be real-time dealt with, and this module continues the following grammar check. As shown in Figure 2. In the grammar check include the following:

- Whether the same group of G code
- Parameter integrity and legality check

![Figure 2. Syntax analysis sub-module.](image)

(3) Extraction of processing information storage sub-module: After lexical analysis and grammar analysis, you can extract the effective processing of information that G code contains what are stored in the embedded database. The database can not only access and query data effectively, but also ensure the system has good overall performance by security measures such as security check and integrity check.

For the entire G code file, simply repeat (1) (2) (3) process, until the end of the file, you can parse the entire G code file. The basic information of a CNC defined instruction is defined with the class CNC_Data. The structure member corresponds to the field of the NC_Data table. During the system start up initialization phase, the function of Open_Library() is called and the NC_Data table is created. If the table already exists, the function is canceled. Operation and return a prompt message. Here's a partial implementation:

```cpp
// The G code interpreter interprets the data structure of the results
Class CNC_Data // G code decoding results
{
    int error_number; // according to the number to determine the type of error
    int error_Line;   // statistical error line number
    long line_number; // line number
```
int g_code_mode;                                 // G code
int g_code_count;                                 // Number of G codes
double start_x, start_y, start_z;                // starting point coordinates
double end_x, end_y, end_z;                     // end point coordinates
double u, v, w;                                     // Incremental coordinates
int m_code_mode;                                 // m code
int m_code_count;                                // M code number
double radius;                                       // Radius value
double spindle_Speed;                         // spindle speed
double feed_Rate;                                 // feed rate
int tool_Number;                                   // tool number

CNC_Data * data;

void Data_Base:: Open_Library (char * db_Name)
{
    char * info;
    char * sql;
    if (data_Base.Open (db_Name, & db)! = SQLITE_OK)
    {
        Show_Error_Message ( "Open DB Failure!"); // This function is defined in the function
        return; // open the database failed to exit the program
    }
    If (data_Base.exec (db, sql, NULL, NULL, & info)! = SQLITE_OK)
    {
        Show_Error_Message (info);
        ...
    }
}

Declare a function in a function show_Error_Message(QString &).

(4) the error handling module: The error appears in the interpretation process, but the interpreter
who cannot stop decoding work and wants the system to record the error location and type of error,
continues to explain the work until the program is completed and then print all the error message so
that Error handling. The system error handling implementation also follows this principle, the system
error handling only provides the error message printing and error location display and other
functions, and the error correction work left to the user to complete. If an error is detected in the G
code during lexical and grammar checks, an error number and the line number of the error are
returned, and the cause of the error is determined by the error line number and the error number.
Common errors are shown in Table 1:

<table>
<thead>
<tr>
<th>Error number</th>
<th>Explanation</th>
<th>Error type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>normal</td>
<td>Lexical error</td>
</tr>
<tr>
<td>1</td>
<td>An illegal character or an unsupported character appears</td>
<td>Lexical error</td>
</tr>
<tr>
<td>2</td>
<td>Data format is incorrect or missing data</td>
<td>Lexical error</td>
</tr>
<tr>
<td>3</td>
<td>Incompatible code in the same section</td>
<td>Grammatical errors</td>
</tr>
<tr>
<td>4</td>
<td>The command parameter is incorrect</td>
<td>Grammatical errors</td>
</tr>
</tbody>
</table>

G Code Interpreter Implementation

The G code interpreter designed in this paper mainly includes two processes: reading G code and
extracting processing information. During reading G code, it is mainly G code interpretation,
including lexical examination, grammar check, etc. And it will be interpreted to the data source that
a numerical control system can identify, that is the intermediate code, what will be stored in the embedded database SQLite according with the agreed manner. In the process of processing, it only need to read data from the database file in the intermediate code, and then the data will be sent to the corresponding compensation by the tool after the motion controller can be processed. NC program interpretation process is shown in Figure 3:

![Interpreter flow chart](image)

**Figure 3. Interpreter flow chart.**

**Human - Machine Interaction of G-Code Interpreter**

In the QT4 environment, with the using of signal and slot mechanism, the user simply click on the decoding button, and then call the relevant process function to complete the interpretation of G code. And then the relevant information back to the interface. If the NC program is successfully interpreted, the data information will be explained to the circular and linear interpolation command. If the NC program is unsuccessful, the corresponding error message will be sent back to the user interface through the error processing module. The user will modify the corresponding error according to the prompt. Specific ideas are as follows:

In the Analyze_NC class, it is responsible for the relevant processing of the NC program. If successful, the corresponding processed information will be transferred to the text display area to inform that the G code is interpreted correctly, and print that the processing is in progress; if unsuccessful, the error message will be sent to the text display area, and then tell us where there is an error and what kind of error, and then we manually start to modify the NC program.

In the main interface of the program, according to Analyze_NC class, create a class object, and then call:

```cpp
Analyze_NC * analyze_NC;
Analyze_NC = new Analyze_NC;
Analyze_Btn = new QPushButton (tr ( "parse"), this);
Connect (analyze_Btn, SIGNAL (clicked ()), analyze_NC, comfileSlot ()); // Click here button to
//achieve the specific function, so that it can compile and explain.
```
Summary
In this paper, the NC G code interpreter is designed under the Qt4 software platform. The interpreter translates the NC code into the data structure which can be recognized by the numerical control system and sends it to the corresponding motion controller to realize the actual machining. The use of Qt portability, G code interpreter through the windows platform development, and then compiled under other platforms, greatly saving development time.

Acknowledgement
This work was supported by the National Science and Technology Support Program (2014VAF09B00) in Shandong Province, NC generation of innovative application of mechanical products demonstration project and the research &Shandong Province independent innovation and transformation of special(2014CGZH0804), NC generation of bus control system in the field of wood processing machinery and industrial application demonstration.

References