***I.J. Education and Management Engineering* 2012, 2, 47-53**
Published Online February 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.02.08

Available online at http://www.mecs-press.net/ijeme

A Course Project of Chinese Character Text Editor for Embedded System Education

Ye Junjie[a], Yu Jianxin[b]

Department of Computer Science and Technology, Nanjing University, Nanjing, China

Abstract

Course project experiment plays a very important role in master student embedded system software education. The paper gives the design and implementation solution to a Chinese character text editor for embedded system, including requirement analysis, function design, task division, main process flow, Chinese character process, man-machine interface layout and test results. This course project can be used as a template for embedded system software experiment.

Index Terms: embedded system; text editing; Chinese character processing; VxWorks; WindML; course project

1. Foreword

One important aspect of embedded system (ES for short) education is course project experiment. It requires the students to make a product prototype with some design concept in lab, which is between verification experiment in school and industry project at scale and technology level [1]. Therefore, the current situation in China is that most college students lack enough knowledge about product development in IT industry. Even though they have studied many courses in institute, it is still a puzzle for them to make a course project. In other words, to guide them to gradually touch the course project and play an active part in this kind of experiment development, assistant and teacher of the course need several product prototype examples for teaching. The prototype examples should possess characteristic of wide applicability and adequate development scale. Out of this consideration, we developed an embedded Chinese character text editor. It is used as a course project paradigm in embedded software development curricula for master students. This text editor runs on ES experiment board, capable of ASCII text editing and Chinese character text editing.

This paper introduces design and implementation of embedded Chinese character editor, which is a utility on VxWorks operating system.

* Corresponding author.
E-mail address: [a]ye020510625@126.com; [b]yujianxn@nju.edu.cn

2. Requirement analysis

We analyzed a well-made text editor with graphic interface, which runs in the real ES environment[2][3]. After the analysis, we considered that the editor should have the following basic functions[4].

(1) It can create a new text under graphic interface; (2) User can input upper and lower letters, ASCII symbols, number, and Chinese characters; (3) It can write edited text into flash memory; (4) It can open a text file in flash memory, and the characters in file can be displayed on editing window in pages and lines sequence; (5) Page down and page up display function; (6) Find a specific string in text and highlight it. (7) By pushing keys on keyboard, user can move focus cursor to operating location in text file. (8) It can insert or delete a letter, a symbol or a Chinese character at the cursor location.

3. Task division

This embedded text editor program is divided into 7 tasks. Detailed information of those tasks is given as follows.

tEntry task: This is the entry function of editor. It is also the main control function of the editor and responds to any input from keyboard and touch screen. The task firstly initializes global variables, creates some semaphores and message queues. Then the task draws picture frame of Man-Machine Interface (MMI for short) on LCD. After that, this task creates other 6 tasks. Priority of this task is higher than that of tasks created by it, because input operations often need preferential process. Finally, it enters an infinite loop. In the loop, once it received a touch screen input or keyboard input, it sends corresponding message to related task.

tTextArea task: This task is responsible for text's content display. It opens a text file from flash memory, reading all data and loading them into memory. Then it sets up index for the text pages and calculates total page number. After that, it displays first page content in editing window of MMI. It is always in waiting states. If this task received messages from other tasks, including cursor moving, content deleting, content adding, page up or page down displaying, file saving, etc., it processes them at once according to concrete content of received message.

tNumArea task: Main function of this task is to respond to message trigged by number input button. It extracts ASCII code from the message which is from tEntry, and sends them to tTextArea task, which will insert the code into cursor place and display the text page including the code.

tEngArea task: This task responds to message trigged by English input button. It processes message sent from tEntry, and extracts ASCII letter code. Then this task sends ASCII code to tTextArea task, which will insert and display the code.

tChnArea task: It responds to message trigged by Chinese character input button. It processes Chinese phonetic code message sent from tEntry, completing Chinese character letter code search and input through phonetic retrieval function. Then this task sends Chinese character GB2312 code to tTextArea task, which will insert and display that Chinese character at current cursor place.

tSymArea task: It responds to message trigged by symbol input button. It processes key scan code message sent from tEntry and obtains ASCII symbol code within symbol conversion table. Then it sends ASCII symbol code to tTextArea, so as to insert and display the symbol code.

Fig. 1 shows the communication between the main 6 tasks. In this program the communication is mostly achieved by message queues. It also involves a semaphore so as to realize the synchronization between two tasks. In the Fig. 1, the semaphore is marked by dashed line, while the message queues are marked by real lines.

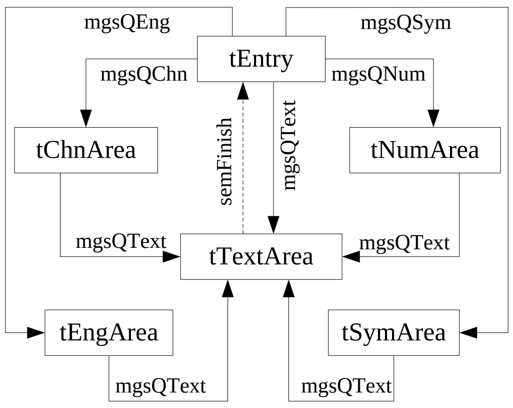


Fig 1. IPC flow chart for the main 6 tasks

4. Data structure

Main data structure in the editor is character array FTEXT. This array is used to save the total text data read from .txt file from flash memory. User's any operation on the text, such as appending, deleting or inserting will change this array. Another main structure is an integer array PAGE_INDEX. It is used to record the first character byte location of a page in array FTEXT, and the location is expressed by subscript number in FTEXT array.

In Fig. 2, three main data structure variables are illustrated by UML class diagram. They are char_msg, tree_node, and find_list.

The data type char_msg is used in message queue among tasks by send_msg ( ) and recv_msg ( ) functions to deliver current input information which is from keyboard or touch screen, or to deliver what will be inserted in text. Character codes to be inserted into text are transmitted by this kind of message too. In data structure char_msg, the start field is used to record the type of message; the value field is used to record ASCII character code when ASCII character is transmitted. Once transmitting Chinese characters, both value and word fields are used.

Data structure variable tree_node is used to express nodes that make up Chinese character pinyin lookup tree. Any pinyin letter in a valid pinyin sequence string corresponds to a unique node of the lookup tree, and this pinyin letter is stored in key field of the node. In order to build that tree, function add_node( ) is called. When user inputs a Chinese character by pinyin method, function tree_search( ) will be called to search the tree and obtain several Chinese characters. In this case, start field records GB2312 code of the first Chinese character corresponding to that pinyin sequence, while number field records the number of Chinese characters corresponding to that pinyin sequence.

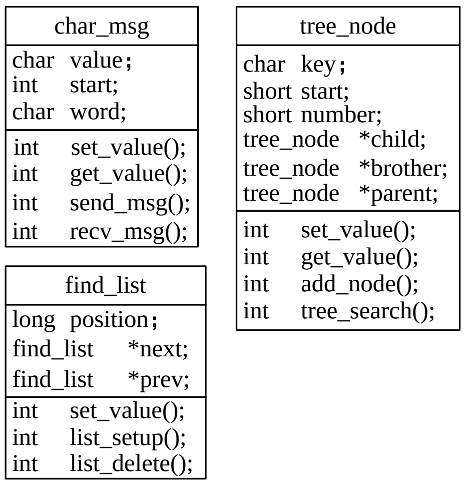


Fig 2. UML class diagram

Data structure variable find_list stores node information of double linked list. When a double linked list needs to be set, function list_setup( ) is called. When a string pattern is searched within text and a matched string position in text is found, then the position information is stored in a node of this kind of double linked list. This store operation is done for all matched position for each search action. Once work state returns to edit state from search state, it is necessary to delete the double linked list by calling list_delete( ) function.

5. Main process flow

Text editor entry function is tEntry( ). We use pseudo-code to describe process flow of tEntry function as follows.

```
void tEntry ( ){
Initialize global variable.
Initialize WindML.
Create font.
Set up key tree for Chinese character pinyin input method.
Input text file name for editing.
Draw MMI picture frame.
Create message queue and semaphore.
Spawn 6 tasks.
Wait for input until reset editor.
Delete all message queues, semaphores and 6 tasks.
Jump to start position of this function.
}
```

The most important function of text editor is tTextArea( ). That task is responsible for such operations as text display, page down or page up, insert, find, etc. Main operation flow chart of task function tTextArea( ) is illustrated by Fig. 3.

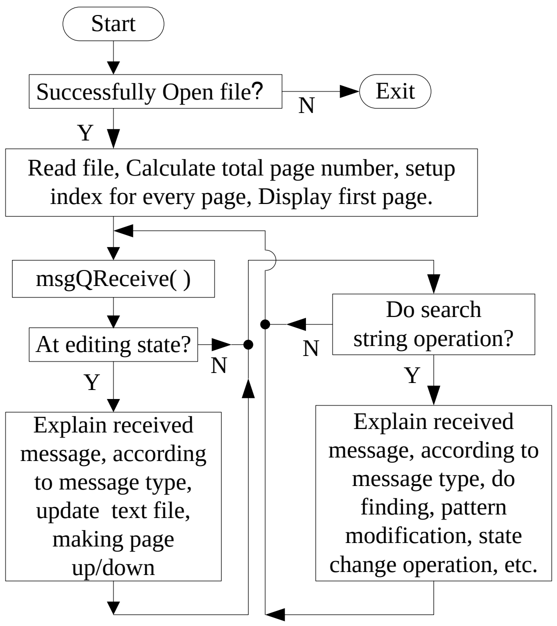

Start
Successfully Open file?
N
Exit
Y
Read file, Calculate total page number, setup index for every page, Display first page.
msgQReceive( )
At editing state?
N
Y
Do search string operation?
N
Y
Explain received message, according to message type, update text file, making page up/down
Explain received message, according to message type, do finding, pattern modification, state change operation, etc.

Fig 3. tTextArea task flow chart

When building character pinyin lookup tree, we use recursive-descent routine. The flow chart of recursive function add_node( ) is shown in Fig. 4.

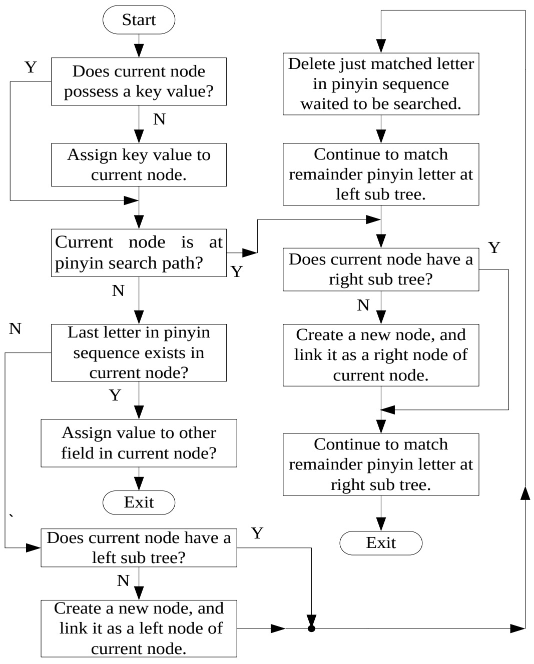

Start
Does current node possess a key value?
Y
N
Assign key value to current node.
Current node is at pinyin search path?
Y
N
Last letter in pinyin sequence exists in current node?
N
Y
Assign value to other field in current node?
Exit
Does current node have a left sub tree?
Y
N
Create a new node, and link it as a left node of current node.
Delete just matched letter in pinyin sequence waited to be searched.
Continue to match remainder pinyin letter at left sub tree.
Does current node have a right sub tree?
Y
N
Create a new node, and link it as a right node of current node.
Continue to match remainder pinyin letter at right sub tree.
Exit

Fig 4. add_node function flow chart

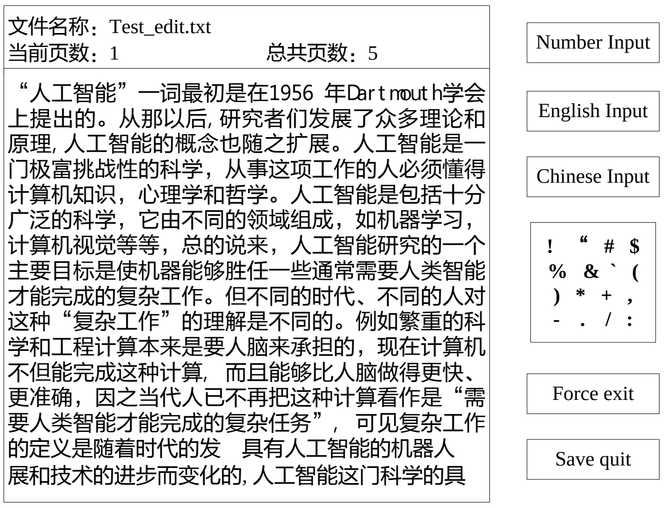
6. Chinese character process

In this course project, Chinese character matrix library HZK16 is used for Chinese character display within the edited text. In HZK16 library, every simplified Chinese character code is expressed in two bytes and accordant dot array matrix has 16 rows and 16 columns. HZK16 library is national simplified Chinese code standard published at 1980, supporting 6763 Chinese characters and 682 symbols, in which level one have 3755 Chinese characters sorted by phonetic sequence, level two have 3008 Chinese characters sorted by simplified radicals[5].

Apart from display, the editor program provides pinyin input method. For this reason, we constructed a key tree[6]. Every node in key tree corresponds to a key value. Nodes path that is from root node to any node along the key tree constitutes a pinyin code of Chinese character. If a node possesses last letter of one pinyin sequence, the first Chinese character's GB2312 code of that pinyin sequence in HZK 16 is saved in that node. Additionally, the number of this kind of Chinese characters is also saved in that node. After a pinyin key tree has been built up, the first GB2312 code of a group Chinese character which has the same input pinyin sequence can be retrieved by recursive-descent searching routine. The number of Chinese characters to this pinyin sequence can be retrieved as well.

7. MMI picture layout

We use components of WindML library to implement MMI for text editor. The embedded experiment board has a 640*480 TFT type LCD. WindML supports this kind LCD screen and touch screen[7]. So we wrote visual MMI interaction picture layout, which composes a convenient GUI for text operation. Fig. 5 gives the layout of MMI picture.


文件名称：Test_edit.txt
当前页数：1　　　　　　总共页数：5

“人工智能”一词最初是在1956 年Dartmouth学会上提出的。从那以后, 研究者们发展了众多理论和原理, 人工智能的概念也随之扩展。人工智能是一门极富挑战性的科学，从事这项工作的人必须懂得计算机知识，心理学和哲学。人工智能是包括十分广泛的科学，它由不同的领域组成，如机器学习，计算机视觉等等，总的说来，人工智能研究的一个主要目标是使机器能够胜任一些通常需要人类智能才能完成的复杂工作。但不同的时代、不同的人对这种“复杂工作”的理解是不同的。例如繁重的科学和工程计算本来是要人脑来承担的，现在计算机不但能完成这种计算， 而且能够比人脑做得更快、更准确，因之当代人已不再把这种计算看作是“需要人类智能才能完成的复杂任务”， 可见复杂工作的定义是随着时代的发　具有人工智能的机器人展和技术的进步而变化的, 人工智能这门科学的具

Fig 5. GUI picture of text editor

8. Test results

(1) Every input method is tested and confirmed normal. (2) In text display state, operations such as page down or page up displaying, cursor moving, character deleting all run normally. (3) Testing cursor moving and

locating by touch screen is successful. (4) String pattern searching function runs well, all positions of test sample string in the tested text file can be searched out quickly. (5) When save button is pushed, edited text can be well saved into flash memory as a .txt file. After saving, if lab assistant opens this .txt file once again, the latest changed content can be normally read and shown. (6) When editor runs in editing state and force exit button is pushed, the edited text will not save current content changes.

9. Conclusion

Text editing tool is a basic software tool in handheld device for example, mobile phone or PDA, etc. This course project of text editing possesses such functions within ES graphics interface as ASCII character input, Chinese character input, text edit, text display, etc. In our master course of embedded system software development, this course project can be used as a code paradigm in text editing tool education. The editor is programmed by ANCI C in Tornado IDE. Its total statements line number is about 2800. Education emphases of this course project are process oriented programming, multitask division, multitask scheduling by semaphores and messages, and Chinese character process. We can teach the editor’s inner structure and process in class room. Apart from this, in lab, based on this sample editor we are able to make students write an ES text editing tool from the beginning, or expand functions at current version, so as to train their independent ES software development ability.

References

[1] Wilson P. Paula Filho, “Process Issues in Course Projects,” St. Louis, Missouri, USA, pp. 629-630, May 2005.
[2] Andrew J. Ko, Htet Htet Aung, and Brad A. Myers, “Design Requirements for More Flexible Structured Editors from a Study of Programmers’ Text Editing,”, ACM New York, NY, USA, pp. 1557 – 1560, April 2005.
[3] Sharma, D. K. and Gruchacz, A. M., “The Display Text Editor TED: A Case Study in the Design and Implementation of Display-Oriented Interactive Human Interface,” IEEE Trans. Comm. COM-30, Jan.1982.
[4] Teresa L. Roberts, Thomas P. Moran, “Evaluation of text editors,” ACM New York, NY, USA, pp. 136–141, March 1982.
[5] GUO Hua, XU Long fei, and ZHANG Zhong, “Study on simplified and traditinoal Chinese processing and character libray expansion technique in embedded system,” computer Engineering and Design, vol.27, No. 3, Feb. 2006. (in Chinese)
[6] Li Fangjun, Jin Weidong, Xun Yonghong, and Liu Yong, “Realization the phonetic input method on embedded system,” Education and Science, issue 2, June. 2006. (in Chinese)
[7] Wind River Co., VxWorks Programmer's Guide, Tornado Online Manuals, 2003.