Research and Implementation of Message Queue on Key Event Processing

Zhijian Yin and Linhan Dai

ABSTRACT

As electronic technology products continue to expand the control panel buttons, there are hundreds of buttons, and different types of buttons. In this paper, the advantages and functions of message queue in the processing of key events are analyzed in detail. Proposed bond between the control board and the control board to pass key information through SPI communication method, and implemented in Free ROTS(embedded operating systems) use a variety of different message queue management of key events, and in response to key events. The conclusion is that the main control board chip can use software to classify the key events by message queue.

INTRODUCTION

Electronic technology products increased peripheral button number, type diversification, lead to a traditional software consists of a main control chip control scan button event will greatly increase the task burden of MCU and software execution efficiency. In addition to the aid of hardware other specialized keyboard management chip scanning key events, and with main control MCU establish corresponding communication protocol, software also need to use more optimized way to manage key events. The way in which the use of message queues can not only greatly reduce the burden of MCU real-scan key event, but also to manage a variety of different types and key messages for different events.

Zhijian Yin, Linhan Dai, Jiangxi Science & Technology Normal University, Nanchang, Jiangxi, China
MESSAGE QUEUING IN THE ROLE OF KEY EVENTS

Message queue is a linear chain structure. When the message queue is empty, only the first team and the team tail these two pointers, they are front and rear. When the MCU receives key ID from slave machine through communication. For example, by mean of SPI communication, the task is scanned to the corresponding button ID changes, the message queue will be in a specific format and a particular priority to add new messages received at this time. As shown in Figure 1:

Message queues are created based on the message structure defined by the user, including the type of key, key state ID, keys, key values and other information, create a new node. By linear chain structure to add in a new node. Since the head of the queue (front) is only allowed to remove one end, while the tail (rear) is to allow only one end of the FIFO queue to delete determines the characteristics (FIFO), as shown in Figure 2:
When the button is released, the button event will be released from the queue in accordance with the first out of the way, the release of the button marked the queue node in the queue to delete. Node in the key structure of the message will also be followed by the team as shown in Figure 3:

Therefore, the message queue is used to deal with the key events, and only need to open a high priority task to monitor the key events in the embedded operating system. As long as the detected event is triggered, the new message queue will be added, when the button is released in response to a custom mode according to the corresponding treatment. So, It can reflect the advantages as follows: 1. Key management become more high speed and high efficiency, as long as the key data structure established, the same type of button can be given the same attributes, without keys on the each one defined its structure properties; 2. To save the memory, expansion and convenient buttons; 3. Facilitate the management, when a certain key types or attributes to change, only need to change the corresponding data structure can be. 4 According to the needs of users to deal with complex button events such as: combination of buttons, long press, even press, etc..

MESSAGE QUEUE KEY COMMUNICATION MESSAGES

Master control MCU through the communication to accept the key messages from the machine, the communication can be through the CAN bus, SPI, I^2C, etc.. At the same time, the key message can be given in the form of an array, the form of the message, the shape of the frame byte, and so on. In the circuit design, by other from chip in charge of each kind of different types of key scan can be the key event situation is sent to a main control buttons MCU through the CAN bus, of course, this MCU can also processing other data, or too wasteful MCU resources. And button control of the MCU through the collation of key data will be sent to a specific message in a specific manner to the main control chip.
THE KEY MESSAGE QUEUE AND DATA STRUCTURE CREATED

You should first establish the key message queue data structure shown in Figure 5:

```c
typedef struct {
    KeyStuct *Out;
    KeyStuct *In;
    KeyStuct *End;
    uint16_t NDatas;
    uint16_t MaxDatas;
    uint8_t (*ReadEmpty)();
    uint8_t (*WriteFull)();
    KeyStuct Buf[1];
} KeyQueue;
```

Figure 5. message queue data structure.

```c
typedef enum {
    KEY_SLIDER = 0, //Slider
    KEY_BUTTON,    //Common key
    KEY_FUNCS,     //Function keys
    KEY_NONE       //Pending button
} KEY_T;
```

```c
typedef enum {
    KEY_FREE = 0,     //Key release
    KEY_PRESS,        //Press the button
    KEY_LONG,         //Press
    KEY_S,
} KEY_S;
```

Figure 6. key data structure.

By writing `Q_Key_Msg_Create uint8_t (*Buf void, Buf Size uint32_t)` function, where “Buf” is the data, Buf Size is the size of the data set up an empty queue. Then define the data structures of various keys for different types of buttons, as shown in Figure 6:

MESSAGE QUEUE KEY EVENT RESPONSE

In Free ROTs open a higher task or is using a timer scanning the key event, when the detected key or value changes, this time written `uint8_t Q_Key_Msg_Write (void *Buf, Key Stuct *Data)` function, the buf is spatial data, the data is a key message of the structure body. Add a message to the queue. At the same time, the task of key events to make corresponding processing. First, it compiles the `uint16_t Q_Get_Key_Msg_Number (void) *Buf function is read key message size, if the size is not 0, said a key event is triggered; followed by writing `uint8_t Q_Key_Msg_Read (KeyStuct *Ret, void *Buf)` function, read out the message queue in the key message structure, at the same time released in the queue, the message node; finally to corresponding logic judgment, needle for different button type and combinations were custom and. The key data structure defines the key type, key state, the key
values of key ID. Key type slider, ordinary keys, function keys, pending button. The
state of key includes the long press, press and release. At the same time, each key
has by pressing the corresponding value and ID. The flow chart is shown in Figure 7:

![Flow Chart](image)

**Figure 7. message queue processing button flow chart.**

**SYSTEM TEST**

When the encoder rotates, the message queue receives the pulse message of the
encoder, and the program can increase or decrease the corresponding value
according to the logic judgment. As shown in Figure 8:

When the slider is sliding, the message queue receives the current value of the
slider in real time, and the program also increases or decreases the corresponding
value. As shown in Figure 9:

Keys are respectively in ordinary buttons and button combinations, message
queue according to the ID of the button, corresponding ordinary button press button
LED light, or by combinations of buttons makes a row of buttons on a LED light. As
shown in Figure 10:
CONCLUSION

Message Queuing in dealing with issues of key events enable efficient management of all types and combinations of keys. Through analysis of the key events to the message queue processing, according to the characteristics of the message queue extended is used to solve other problems. For example: the pulse signal processing and utilization the message queue communication platform for three-tier structure, MMORPG server software architecture is implemented.

REFERENCE