

Event-System

Die Basis-Firmware enthält ein kleines Event-System, mit der eine lose Kopplung zwischen AddOn und dem BiDiB-Bus zur Versendung von Nachrichten und das Starten bestimmter Makros realisiert werden kann.

Ereignisverwalter

Zentraler Teil ist der Ereignisverwalter mit dem Modul `event_handler` in der Basis-Firmware. Das Modul wertet das für Events zuständige Fifo: `event_fifo` aus und agiert entsprechend.

Zum Auslösen eines Ereignisses wird das Fifo: `event_fifo` lediglich mit dem gewünschten Ereigniscode gefüllt. Per Taskverwaltung wird dann der „Event Handler“ informiert.

Ereignisse können die Werte 0x00 bis 0xFF annehmen.

Makro-Ereignisse

Der Bereich von 0x00 bis 0x7F ist für die interne Makroverwaltung reserviert.

Taster Ereignisse

Der Bereich von 0xF0 bis 0xFF ist für die Taster auf der Basis-Platine reserviert.

Keyboard-Ereignisse

Unter Keyboard-Ereignissen wird das Verstellen von **externen** Tastern oder anderen Schaltelementen verstanden.

Keyboard-Ereignisse liegen zwischen `IN_DOWN0` (ab 0x80) und `IN_DOWN15` sowie `IN_UP0` und `IN_UP15`. Die Definition ist in der Definitionsdatei `keyboard.h` enthalten.

Diese Ereignisse können frei verwendet werden und lösen eine `MSG_LC_KEY`-Nachricht bei eingeschaltetem „Feature“ `FEATURE_CTRL_INPUT_NOTIFY` aus.

Zusätzlich wird für die Ereignisse `IN_DOWN0` bis `IN_DOWN15` das korrespondierende Makro gestartet, wenn das „Feature“ `FEATURE_CTRL_MAC_START_MAN` aktiviert ist.

Ereignis auslösen

Allein durch das Füllen des Fifos wird das entsprechende Ereignis ausgelöst:

```
const unsigned char in_code_table[] PROGMEM = {  
    // Scan line 0  
    IN_DOWN0,      // Bit 0  
    IN_DOWN1,      // Bit 1  
    IN_DOWN2,      // Bit 2  
    IN_DOWN3,      // Bit 3  
    IN_DOWN4,      // Bit 4  
    IN_DOWN5,      // Bit 5  
    IN_DOWN6,      // Bit 6  
    IN_DOWN7,      // Bit 7  
    // Scan line 0  
    IN_UP0,         // Bit 0  
    IN_UP1,         // Bit 1  
    IN_UP2,         // Bit 2  
    IN_UP3,         // Bit 3  
    IN_UP4,         // Bit 4  
    IN_UP5,         // Bit 5  
    IN_UP6,         // Bit 6  
    IN_UP7,         // Bit 7  
};
```



```
unsigned char in_code;  
in_code = pgm_read_byte(&in_code_table[i]);  
cr_fifo_put(&event_fifo, in_code);
```

Im Beispiel nimmt den größten Teil die Umsetzungstabelle des Port-Bits in den Ereigniscode in Anspruch.

From:

<https://forum.opendcc.de/wiki/> - BiDiB Wiki

Permanent link:

<https://forum.opendcc.de/wiki/doku.php?id=softwarebausteine:event-system>

Last update: **2014/09/01 12:34**

