

Datei zur Dokumentation der Programmierschnittstelle (API) des Future Operating System (FutureOS) in Erweiterungs-ROMs. Hier werden die für Anwender und Programmierer wichtige Funktionen des Betriebssystems beschrieben, die sich in FutureOS ROM A und allen XROMs befinden.

Die OS Funktionen werden in folgender Form beschrieben:

1. Kurzbeschreibung: in einem Satz wird kurz die Funktionsweise der OS Funktion beschrieben.

2. Label: Dieses Label wird für die OS Funktion im Source Code verwendet. In der mitgelieferten Label-Bibliothek (#EQU-API.DEU) mit den jeweils aktuellen ROM-Adressen findet ebenfalls dieses Label Verwendung. In eigenen Programmen sollte man alle OS Funktionen (und Systemvariablen) stets mit diesen Labels ansprechen. Dadurch wird der Quellcode einheitlich. Ausserdem KÖNNTEN sich in zukünftigen OS Versionen die Startadressen der OS Funktionen ändern. Siehe #EQU-API.DEU.

3. ROM-Nummer: hier steht die logische Nummer des OS ROMs in der die entsprechende OS Funktion zu finden ist. Die in dieser Datei beschriebenen OS Funktionen kommen allerdings nur in ROM A vor und in allen Erweiterungs-ROMs (XROMs). Da die ROM Nummer logisch zu verstehen ist, sollte man sie nicht automatisch mit dem physikalischen ROM Select gleichsetzen.

4. Startadresse: gibt die Einsprung-Adresse der OS Funktion an.

5. Einsprungbedingungen: die Einsprungbedingungen der OS Funktion werden beschrieben und es wird sowohl auf Registerinhalte als auch auf RAM-Variablen eingegangen.

6. Aussprungbedingungen: die Aussprungbedingungen der OS Funktion werden beschrieben und es wird sowohl auf Registerinhalte als auch auf RAM-Variablen eingegangen.

7. Manipuliert: es werden alle manipulierten oder zerstörten Register und RAM-Variablen wiedergegeben. Manchmal werden auch manipulierte Pheripheriebausteine angegeben.

8. Beschreibung: es folgt eine vollständige Erklärung der Funktionen der beschriebenen OS Funktion.

9. Bitte Beachten: es werden einige wichtige Details kurz angeschnitten. Dies ist besonders wichtig, denn da alle OS Funktion kompromißlos auf Höchstgeschwindigkeit getrimmt worden sind, kann es bei falscher Handhabung zu Problemen mit der Systemstabilität kommen.

Die im folgenden beschriebenen OS Funktionen dienen der Verwaltung von Erweiterungs-ROMs für das FutureOS. Sie liegen alle im FutureOS ROM A und auch in allen XROMs. Die Adressen sind in allen ROMs gleich, so kann man direkt von einem ROM in ein anderes springen. Es handelt sich dabei vor allem um OS Funktionen, die das ROM Banking ermöglichen, bzw. um OS Funktionen für die Ausgabe von Zeichenketten.

Der automatisch Start von Applikationen

Bevor es an die Beschreibung der XROM Funktionen geht soll hier die XROM 'Initialisierung' bzw. der automatischs Start von Applikationen gezeigt werden.

Wenn ein XROM entsprechend konfiguriert ist, dann kann jeder enthaltene Teil bzw. jedes Programm beim Start des FutureOS automatisch ausgeführt werden. So kann beispielsweise ein Hintergrund-Bild automatisch geladen werden.

Ab dem Label INITXROM befindet sich in neueren XROMs eine OS Funktion, die beim System-Start vom FutureOS automatisch aufgerufen wird, wenn das ROM durch die Bytes &02, &09 ab Adresse &C000 als FutureOS XROM kenntlich gemacht wurden. Diese Funktion liest die vier Bytes ab Adresse &FEFC im XROM (siehe XROM Source-Code ab Label KBAA). Dort sind zuerst zwei Bytes enthalten, die die Kontrollbits für den Autostart von Applikationen (Apps) enthalten. Ab KBAA+&02 befinden sich dann zwei Bytes die die Autostartbits für max. 16 Applikationen enthalten. In beiden Fällen befindet sich das niederwertige Bit (und Byte) zuerst im Speicher.

Nach dem Lesen dieser vier Bytes wird getestet, ob die Kontroll-Bytes stimmen, falls nicht wird abgebrochen. Hierbei wird für jedes gesetzte Bit in den App-Bits das entsprechende Bit in den Kontroll-Bits geleert und somit bestätigt.

Ist ein App-Bit gesetzt, dann wird das Register IY mit dem Wert &9999 geladen, dieser Wert dient der App. als Erkennungsmöglichkeit, um zu prüfen, ob es sich beim aktuellen Aufruf um den System-Start handelt. In so einem Fall sollte die aufgerufene App. unbedingt mit einem RET zurückkehren. Bei normalen Aufrufen einer App. hingegen sollte diese mit einem Sprung nach TUR_E bzw. KLICK zurückkehren.

Nach dem Laden von IY mit &9999 wird die App. ausgeführt. Somit können beim Systemstart verschiedene Dinge erledigt werden, ohne sie per Hand ausführen zu müssen (z.B. Wallpaper, Bildschirmschöner etc.).

Der Aufbau der App.-Autostart-Bits ist hier gezeigt:

Konfigurations-Bits (16) zum Autostart von Applikationen

```
KBAA DW %1111111111111111 ; Kontrollbits, werden mit XOR verbunden
      0FEDCBA987654321 ; Nummer der Applikation, LSB = 1. App.
      DW %0000000000000000 ; Autostartbits, 0 = inaktiv, 1 = Autostart ein
```

Wenn man z.B. die erste App. automatisch starten will, dann sieht das so aus:

```
KBAA DW %1111111111111110 ; Kontrollbits, werden mit XOR verbunden
      0FEDCBA987654321 ; Nummer der Applikation, LSB = 1. App.
      DW %0000000000000001 ; Autostartbits, 0 = inaktiv, 1 = Autostart ein
```

In Bytes ab Adresse &FEFC: &FE, &FF, &01, &00

Weitere Beispiele:

- Autostart der 2. Applikation, ab &FEFC: &FD, &FF, &02, &00
- Autostart der 3. Applikation, ab &FEFC: &FB, &FF, &04, &00
- Autostart der 4. Applikation, ab &FEFC: &F7, &FF, &08, &00
- Autostart der 5. Applikation, ab &FEFC: &EF, &FF, &10, &00
- ...
- Autostart der 8. Applikation, ab &FEFC: &7F, &FF, &80, &00
- Autostart der 9. Applikation, ab &FEFC: &FF, &FE, &00, &01
- Autostart der 10. Applikation, ab &FEFC: &FF, &FD, &00, &02
- ...
- Autostart der 16. Applikation, ab &FEFC: &FF, &7F, &00, &80

API: OS FUNKTION IN ROM A, B, C ODER D AUFRUFEN

Kurzbeschreibung: Von einem XROM aus wird ein Sprung zu einer OS Funktion in einem FutureOS ROM ausgeführt.

Labels: ROM_X2A, ROM_X2B, ROM_X2C und ROM_X2D

ROM-Nummern: A und alle XROMs

Startadressen: ROM_X2A (&FD80), ROM_X2B (&FD99), ROM_X2C (&FDB2) und ROM_X2D (&FDCB)

Einsprungsbedingungen: IX = Zieladresse der OS Funktion in ROM A-D

Alles weitere hängt von der aufzurufenden OS Funktion ab.

Aussprungsbedingungen: Hängen von der aufgerufenen OS Funktion ab.

Manipuliert: Flags und die RAM-Variable AKT_ROM.

Weitere Register könnten durch die aufgerufene OS Funktion selbst verändert worden sein.

Beschreibung: Diese vier OS Funktionen sind ein Teil des API zum Aufruf von OS Funktionen (in FutureOS ROMs A - D) von einem XROM aus. Dabei wird vom XROM zuerst nach ROM A gesprungen und anschließend in die ROMs A - D verzweigt.

Die Einsprünge von ROM_X2A, ROM_X2B, ROM_X2C und ROM_X2D liegen nur in ROM A (nicht in den ROMs B, C oder D!). Und natürlich müssen sie sich auch in jedem XROM befinden (siehe Beispiel Source Code).

Die Auswahl des Ziel-ROMs (A-D) in dem die eigentliche OS Funktion, die man Aufrufen will, liegt wird durch Auswahl des Einsprungs selektiert: ROM_X2A für ROM A, ROM_X2B für ROM B, ROM_X2C für ROM C oder ROM_X2D für ROM D.

Die Einsprung- und Aussprung-Bedingungen entsprechen denen der aufgerufenen OS Funktion. Aber Achtung: Da die Zieladresse der OS Funktion in Register IX überbegen wird können keine OS Funktionen benutzt werden die in IX Parameter erwarten.

Nach Aufruf und Abarbeitung der OS Funktion kehrt man in das XROM zurück, von dem aus die OS Funktion aufgerufen wurde.

Bitte Beachten: Diese OS Funktionen müssen sich so auch in dem aufrufenden XROM befinden, siehe mitgelieferter Beispiel Source Code.

Beim Assemblieren ist darauf zu achten dir ROM Nummern korrekt zu setzen oder sie später anzupassen.

Die aufgerufene OS Funktion im Ziel-ROM darf RAM-Variable AKT_ROM nicht manipulieren!
Sonnst würde ins falsche XROM zurückgesprungen werden. Ein Absturz könnte die Folge
sein. Jedoch verändern die meisten OS Funktionen AKT_ROM nicht. :-)

Beispiel:

```
LD    IX,HOLE0ID ;Adresse der OS Funktion in IX laden.  
CALL ROM_X2B    ;Die OS Funktion liegt in ROM B. Also dort aufrufen!
```

TEXT AUS EINEM XROM AM BILDSCHIRM ANZEIGEN - MIT STEUERZEICHEN

Kurzbeschreibung: Dieser Satz von OS Funktionen dient dazu Text, der sich in einem XROM befindet in MODE 1 oder 2 mit Farben / Attributen auszugeben. Steuerzeichen werden dabei beachtet und haben ihre übliche Funktion.

Labels: TXR_2, TXR_2D, TXR_2I, TXR_2K, TXR_2U, TXR_GB, TXR_GG, TXR_BB und TXR_RR.

ROM-Nummern: Alle XROMs

Startadressen: TXR_2 (&FDE4), TXR_2D (&FDF5), TXR_2I (&FE07), TXR_2K (&FE19), TXR_2U (&FE2B), TXR_GB (&FE3D), TXR_GG (&FE4F), TXR_BB (&FE61) und TXR_RR (&FE73)

Einsprungsbedingungen: HL = Adresse des Textes (wie auch bei den normalen OS Funktionen für Textausgabe mit Steuerzeichen). Allerdings enthält das erste Byte des Textes die Gesamtlänge des Textes von 0 bis 255.

Aussprungsbedingungen: Der Text wurde am Bildschirm ausgegeben.

Manipuliert: AF, BC, DE, HL, AF', BC', DE', HL' und RAM Variable C_POS

Beschreibung: Das sich die OS Funktionen zur Textausgabe mit Steuerzeichen in FutureOS ROM A befinden kann Text aus anderen ROMs nicht direkt ausgegeben werden. Diese OS Funktionen kopieren den Text zuerst vom XROM in den Text-Puffer (&B000-&B7FF) und geben ihn anschließend am Bildschirm aus.

Dabei ist zu Beachten, dass das erste Bytes des Textes die Textlänge (in Zeichen: 0-255) enthalten muss. Wie üblich wird der Anfang im Register HL überbegen. Hier ein Beispiel:

```
LD    HL,TEXT    ;Zeiger auf Text
CALL TXR_2      ;Text in MODE 2 normal ausgeben
```

```
DB 18,"Das ist ein Test!!",&00 ;Textlänge und Text
```

Die folgenden OS Funktionen stehen zur Verfügung:

- &FDE4 TXR_2 - MODE 2 - Normal
- &FDF5 TXR_2D - MODE 2 - Durchgestrichen
- &FE07 TXR_2I - MODE 2 - Invertiert
- &FE19 TXR_2K - MODE 2 - Kursiv
- &FE2B TXR_2U - MODE 2 - Unterstrichen
- &FE3D TXR_GB - MODE 1 - Pen 1 / Pen 2
- &FE4F TXR_GG - MODE 1 - Pen 1
- &FE61 TXR_BB - MODE 1 - Pen 2
- &FE73 TXR_RR - MODE 1 - Pen 3

Die Auswahl der OS Funktion bestimmt über Bildschirm-Modus und Text-Attribute.

Bitte Beachten: Das erste Byte des Strings enthält die Anzahl der restlichen Zeichen. Steuerzeichen werden beachtet.

TEXT aus einem XROM am BILDSCHIRM ANZEIGEN - OHNE STEUERZEICHEN

Kurzbeschreibung: Dieser Satz von OS Funktionen dient dazu Text, der sich in einem XROM befindet in MODE 1 oder 2 mit Farben / Attributen auszugeben. Steuerzeichen werden dabei als Text-Zeichen dargestellt.

Labels: SXR_2, SXR_2I, SXR_2U, SXR_GB, SXR_GG, SXR_BB und SXR_RR

ROM-Nummern: Alle XROMs

Startadressen: SXR_2 (&FE85), SXR_2I (&FE96), SXR_2U (&FEA7), SXR_GB (&FEB8), SXR_GG (&FEC9), SXR_BB (&FEDA) und SXR_RR (&FEEB)

Einsprungsbedingungen: BC = Laenge des Strings + 1 (maximal 2000)

HL = Adresse des Textes (wie auch bei den normalen OS Funktionen für Textausgabe ohne Steuerzeichen)

Aussprungsbedingungen: Der Text wurde am Bildschirm ausgegeben.

Manipuliert: AF, BC, DE, HL, AF', BC', DE', HL' und RAM Variable C_POS

Beschreibung: Das sich die OS Funktionen zur Textausgabe in FutureOS ROM A befinden kann Text aus anderen ROMs nicht direkt ausgegeben werden. Hiermit werden Strings ohne Steuerzeichen ausgegeben.

Diese OS Funktionen kopieren den Text zuerst vom XROM in den Text-Puffer (&B000-&B7FF) und geben ihn anschließend am Bildschirm aus. Dabei ist zu Beachten, dass das Register BC Textlänge +1 enthält. Wie üblich wird der Anfang im Register HL überbegen. Hier ein Beispiel:

```
LD    HL,TEXT    ;Zeiger auf Text
LD    BC,22+1    ;Länge des Textes
CALL SXR_2      ;Text in MODE 2 normal ausgeben
```

```
DB "Das ist noch ein Test!" ;Das ist der auszugebende Text
```

Die folgenden OS Funktionen stehen zur Verfügung:

&FE85 SXR_2	-	MODE 2 - Normal
&FE96 SXR_2I	-	MODE 2 - Invertiert
&FEA7 SXR_2U	-	MODE 2 - Unterstrichen
&FEB8 SXR_GB	-	MODE 1 - Pen 1 / Pen 2
&FEC9 SXR_GG	-	MODE 1 - Pen 1
&FEDA SXR_BB	-	MODE 1 - Pen 2
&FEEB SXR_RR	-	MODE 1 - Pen 3

Die Auswahl der OS Funktion bestimmt über Bildschirm-Modus und Text-Attribute.

Bitte Beachten: Steuerzeichen werden als Text-Zeichen dargestellt.

Siehe auch FutureOS Homepage:

<http://www.FutureOS.de>