Datei: ‚API-DEU.DOK‘ (10 Seiten) Stand: 14. Juli im Jahre des CPC 2023

Datei zur Dokumentation der Programmierschnittstelle (API) des Future Operating System (FutureOS). Hier werden die für den Anwender oder Programmierer wichtige Funktionen des Betriebssystems beschrieben, die sich in allen ROMs befinden, also jederzeit erreichbar sind. Alle OS Funktionen werden in folgender Form beschrieben:

1. Kurzbeschreibung: In einem Satz wird kurz die Funktionsweise des API Eingangs bzw. der OS Funktion beschrieben.

2. Label: Dieses Label wird für die OS Funktion im Source Code verwendet. In der mitgelieferten Label-Bibliothek (#E) mit den jeweils aktuellen Werten 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 Datei: #EQU-API.DEU.

3. ROM-Nummer: hier steht die logische Nummber des OS ROMs (A-D) in der die entsprechende OS Funktion zu finden ist. Manche OS Funktionen / API Einsprünge kommen in mehreren ROMs an identischen Adressen vor. Dann sind hier auch mehrere ROM Nummern angegeben. Da die ROM Nummer logisch zu verstehen ist, sollte man sie nicht automatisch mit dem physikalischen ROM Select gleichsetzen. Nur beim Einsatz von 256 KB ROM Karten stimmen logische und physikalische ROM Nummer oftmals überein.

4. Startadresse: gibt die Einsprung-Adresse der OS Funktion an. Falls diese in mehreren ROMs vorkommt, wird hier auch eine entsprechende Anzahl von Adressen angegeben.

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 wiedergegeben.

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 Funktion können unter FutureOS stets benutzt werden, da sie entweder in allen ROMs identisch vorhanden sind, oder im RAM liegen.

Es handelt sich dabei vor allem um OS Funktion, die das ROM Banking ermöglichen, bzw. um OS Funktion für die Verwaltung der Echtzeituhr.

Siehe auch FutureOS Homepage: http://www.FutureOS.de

OS ROM A, B, C ODER D SELEKTIEREN / EINBLENDEN

Kurzbeschreibung: Es wird das ROM A, B, C oder D des FutureOS eingeblendet.

Labels: OSRON_A, OSRON_B, OSRON_C, OSRON_D

ROM-Nummer(n): A, B, C und D (alle OS ROMs)

Startadressen:
&FF22 (OSRON_A), &FF58 (OSRON_B)
&FF8E (OSRON_C), &FFD6 (OSRON_D)

Einsprungbedingungen: Keine. Das ROM wird durch das Label selektiert.

Aussprungbedingungen: Das entsprechende FutureOS ROM (A, B, C oder D) wurde zwischen &C000 und &FFFF eingeblendet.

Manipuliert: Lediglich das obere ROM ab &C000.

Beschreibung: OSRON_A, OSRON_B, OSRON_C und OSRON_D dienen dazu, das FutureOS ROM A, B, C oder D einzublenden. Die Auswahl des ROMs wird dabei über die angesprungene OS Funktion (Label) selektiert. So müssen keine Parameter übergebenwerden. Weiterhin werden auch weder Register noch Flags verändert.

Da die OS Funktionen des FutureOS auf vier ROMs (als A, B, C oder D bezeichnet) verteilt sind muß der Anwender Sorge tragen, daß das richtige ROM eingeblendet ist, wenn man einen OS Funktion aufruft.

Dies kann entweder mit OSRON_A,B,C,D getan werden, oder es werden die nachfolgenden API Einsprünge verwendet.

Natürlich kann man ein FutureOS ROM auch 'per Hand' einblenden, siehe Handbuch.

Diese OS Funktionen benötigen 17 us und sind damit schnell.

Bitte Beachten: Die RAM-Variable AKT_ROM wird von dieser OS Funktion NICHT verändert.

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

Kurzbeschreibung: Es wird ein Sprung zu einer OS Funktion in einem definierten FutureOS ROM ausgeführt.

Label: ROM_A, ROM_B, ROM_C, ROM_D

ROM-Nummern: A, B, C und D (alle OS ROMs)

Startadressen: &FF00 (ROM_A), &FF06 (ROM_B), &FF0C (ROM_C), &FF12 (ROM_D)

Einsprungbedingungen: HL = Adresse der aufzurufenden OS Funktion.

Alles weitere hängt von der OS Funktion ab.

Aussprungbedingungen: Das entsprechende OS-ROM (A, B, C oder D) ist zwischen &C000 und &FFFF eingeblendet. Alles andere hängt von der eigentlichen OS Funktion ab.

Manipuliert: Register BC, es wird ein neues oberes OS-ROM (A-D) eingeblendet. Weitere Register könnten durch die Ziel-OS Funktion verändert worden sein.

Beschreibung: ROM_A, ROM_B, ROM_C und ROM_D dienen dazu, das FutureOS ROM A, B, C oder D einzublenden, um in diesem ROM eine OS Funktion anzuspringen.
Das entsprechende FutureOS ROM wird unabhängig von der bisherigen ROM Auswahl eingeblendet, und bleibt auch nach Ausführung der OS Funktion aktiv.
Die Zieladresse der OS Funktion, die angesprungen werden soll, wird im Register HL übergeben. OS Funktionn, die in HL Daten erwarten, können also nicht verwendet werden - dafür sind dann die API Eingänge ROM_?2? zuständig. Wobei ? = A, B, C oder D (siehe unten).
Was nach dem Ende / Rücksprung der OS Funktion geschieht hängt von ihr selbst ab. Das ROM, in dem sich die OS Funktion befindet, bleibt eingeblendet.
Die ROM-Auswahl und der Sprung zur OS Funktion geschieht in 8 ys, und ist damit sehr schnell.

Bitte Beachten: Sind der aufzurufenden OS Funktion in HL Werte zu übergeben, so sind die vier API Eingänge ROM_A..D ungeeignet, da sie bereits die Adresse der OS Funktion in HL erwarten. In so einem Fall sind die OS Funktionn ROM_?2? zu verwenden. Siehe folgende Seiten.

Die RAM-Variable AKT_ROM wird von dieser OS Funktion NICHT verändert.

API: AUFRUF OS FUNKTION IN ROM B, C oder D - RÜCKSPRUNG NACH ROM A

Kurzbeschreibung: Es wird eine OS Funktion in ROM B, C oder D aufgerufen, nach deren Beendigung wird ROM A eingeblendet.

Label: ROM_A2B, ROM_A2C, ROM_A2D

ROM-Nummern: A, B, C und D (alle OS ROMs)

Startadressen: &FF18 (ROM_A2B), &FF2A (ROM_A2C), &FF3C (ROM_A2D)

Einsprungbedingungen: IX = Adresse OS Funktion in ROM B, C oder D.

Weitere Parameter werden durch die OS Funktion bestimmt.

Aussprungbedingungen: Es ist FutureOS ROM A eingeblendet, alles weitere hängt von der aufgerufenen OS Funktion ab.

Manipuliert: ROM A aktiv. Anderes hängt von benutzter OS Funktion ab.

Beschreibung: Die API Eingänge ROM_A2B, ROM_A2C und ROM_A2D erlauben den Aufruf einer OS Funktion in ROM B, C oder D (unabhängig vom gerade aktiven ROM). Nach Beendigung der OS Funktion wird (wieder) ROM A eingeblendet.

Die Adresse der OS Funktion wird in Register IX übergeben. Die OS Funktion darf also in Folge dessen keine Parameter in IX verlangen.

Die Registerinhalte nach dem Rücksprung hängen ALLEIN von der aufgerufenen OS Funktion ab.

Sämtliche API Aufrufe der Art ROM_?2? (wobei: ? = A,B,C,D) dauern für sich genau 34 ys.

Bitte Beachten: OS Funktion darf in IX keine Daten verlangen!

API: AUFRUF OS FUNKTION IN ROM A, C oder D - RÜCKSPRUNG NACH ROM B

Kurzbeschreibung: Es wird eine OS Funktion in ROM A, C oder D aufgerufen, nach deren Beendigung wird ROM B eingeblendet.

Label: ROM_B2A, ROM_B2C, ROM_B2D

ROM-Nummern: A, B, C und D (alle OS ROMs)

Startadressen: &FF4E (ROM_B2A), &FF60 (ROM_B2C), &FF72 (ROM_B2D)

Einsprungbedingungen: IX = Adresse OS Funktion in ROM A, C oder D.

Weitere Parameter werden durch die OS Funktion bestimmt.

Aussprungbedingungen: Es ist FutureOS ROM B eingeblendet, alles weitere hängt von der aufgerufenen OS Funktion ab.

Manipuliert: ROM B aktiv. Anderes hängt von benutzter OS Funktion ab.

Beschreibung: Die API Eingänge ROM_B2A, ROM_B2C und ROM_B2D erlauben den Aufruf einer OS Funktion in ROM A, C oder D (unabhängig vom gerade aktiven ROM). Nach Beendigung der OS Funktion wird (wieder) ROM B eingeblendet.

Die Adresse der OS Funktion wird in Register IX übergeben. Die OS Funktion darf also in Folge dessen keine Parameter in IX verlangen.

Die Registerinhalte nach dem Rücksprung hängen ALLEIN von der aufgerufenen OS Funktion ab.

Bitte Beachten: OS Funktion darf in IX keine Daten verlangen!

API: AUFRUF OS FUNKTION IN ROM A, B oder D - RÜCKSPRUNG NACH ROM C

Kurzbeschreibung: Es wird eine OS Funktion in ROM A, B oder D aufgerufen, nach deren Beendigung wird ROM C eingeblendet.

Label: ROM_C2A, ROM_C2B, ROM_C2D

ROM-Nummern: A, B, C und D (alle OS ROMs)

Startadressen: &FF84 (ROM_C2A), &FF96 (ROM_C2B), &FFA8 (ROM_C2D)

Einsprungbedingungen: IX = Adresse OS Funktion in ROM A, B oder D.

Weitere Parameter werden durch die OS Funktion bestimmt.

Aussprungbedingungen: Es ist FutureOS ROM C eingeblendet, alles weitere hängt von der aufgerufenen OS Funktion ab.

Manipuliert: ROM C aktiv. Anderes hängt von benutzter OS Funktion ab.

Beschreibung: Die API Eingänge ROM_C2A, ROM_C2B und ROM_C2D erlauben den Aufruf einer OS Funktion in ROM A, B oder D (unabhängig vom gerade aktiven ROM). Nach Beendigung der OS Funktion wird (wieder) ROM C eingeblendet.

Die Adresse der OS Funktion wird in Register IX übergeben. Die OS Funktion darf also in Folge dessen keine Parameter in IX verlangen.

Die Registerinhalte nach dem Rücksprung hängen ALLEIN von der aufgerufenen OS Funktion ab.

Sämtliche API Aufrufe der Art ROM_?2? (wobei: ? = A,B,C,D) dauern für sich genau 34 ys.

Bitte Beachten: OS Funktion darf in IX keine Daten verlangen!

API: AUFRUF OS FUNKTION IN ROM A, B oder C - RÜCKSPRUNG NACH ROM D

Kurzbeschreibung: Es wird eine OS Funktion in ROM A, B oder C aufgerufen, nach deren Beendigung wird ROM D eingeblendet.

Label: ROM_D2A, ROM_D2B, ROM_D2C

ROM-Nummern: A, B, C und D (alle OS ROMs)

Startadressen: &FFBA (ROM_D2A), &FFCC (ROM_D2B), &FFDE (ROM_D2C)

Einsprungbedingungen: IX = Adresse OS Funktion in ROM A, B oder C.

Weitere Parameter werden durch die OS Funktion bestimmt.

Aussprungbedingungen: Es ist FutureOS ROM D eingeblendet, alles weitere hängt von der aufgerufenen OS Funktion ab.

Manipuliert: ROM D aktiv. Anderes hängt von benutzter OS Funktion ab.

Beschreibung: Die API Eingänge ROM_D2A, ROM_D2B und ROM_D2C erlauben den Aufruf einer OS Funktion in ROM A, B oder C (unabhängig vom gerade aktiven ROM). Nach Beendigung der OS Funktion wird (wieder) ROM D eingeblendet.

Die Adresse der OS Funktion wird in Register IX übergeben. Die OS Funktion darf also in Folge dessen keine Parameter in IX verlangen.

Die Registerinhalte nach dem Rücksprung hängen ALLEIN von der aufgerufenen OS Funktion ab.

Bitte Beachten: OS Funktion darf in IX keine Daten verlangen!

Lese ECHTZEITUHR aus M4 Karte oder DOBBERTIN/DXS ins RAM

Kurzbeschreibung: Datum und Zeit der Echtzeituhr der M4 Karte bzw. von Dobbertin / dxs ins RAM lesen.

Label: LUHR

ROM-Nummer: Dies OS Funktion befindet sich im System RAM des OS!

Startadresse: &BA66 - im RAM!

Einsprungbedingungen: Die OS Variable AKT_ROM muss die physikalische Nummer des gerade aktiven FutureOS ROMs enthalten.

Die Echtzeituhr von Dobbertin / dxs bzw. der M4 Karte muß unter der, in der RAM-Variable UHR_ROM angegebenen, physikalischen ROM-Nummer liegen. Siehe dazu auch Dateien ‚**#OS-VAR.DEU**' und ‚**#EQU-API.DEU**'

Aussprungbedingungen: Die acht Bytes Daten der Dobbertin / dxs RTC bzw. der M4 Karte stehen ab UHR_00 im RAM.

Das ROM aus der RAM Variable AKT_ROM wurde (wieder) eigeblendet.

Manipuliert: AF, BC, DE, HL und acht Bytes ab UHR_00

Beschreibung: LUHR liest die Daten der Dobbertin Echtzeituhr, des Nachbaus von dxs bzw. der M4 Karte ins RAM ab UHR_00. Aufbau der Daten: siehe Datei ‚**#OS-VAR.DEU**'. Die M4 RTC hat die höhere Priorität.

Beim Aufruf der OS Funktion muß die physikalische Nummer des gerade aktiven / aufrufenden OS ROMs in AKT_ROM angegeben sein, denn am Ende der OS Funktion wird dieses ROM wieder eingeblendet, und in diese Konfiguration zurück gesprungen.

Die logische ROM Nummer aller vier OS ROMs A, B, C, und D kann jederzeit aus dem Byte &C001 eines jeden ROMs gelesen werden. Dabei steht &0A für ROM A, &0B für ROM B, &0C für ROM D und schließlich &0D für ROM D.

Die physikalische ROM-Nummer (ROM-Auswahl / ROM-Select) der ROMs A, B, C oder D kann auch direkt aus JEDEM ROM gelesen werden:
Physikalische ROM-Nummer von ROM A aus &FF01 lesen.
Physikalische ROM-Nummer von ROM B aus &FF07 lesen.
Physikalische ROM-Nummer von ROM C aus &FF0D lesen.
Physikalische ROM-Nummer von ROM D aus &FF13 lesen.

Auf jedes dieser Bytes folgt im ROM das Byte &DF, dies ermöglicht folgenden Code um z.B. ROM C einzublenden:

```
LD BC,(&FF0D)  ;physikalische Nummer von ROM C
               ;dahiner folgt das Byte &DF
OUT (C),C      ;ROM C wird eingeblendet.
```

Die Dobbertin / dxs / M4 RTC wird am CPC wie ein ROM angeschlossen. Um Daten von ihr lesen zu können, muß die von der Uhr belegte ROM-Nummer eingeblendet werden. Ist die Uhr eingeblendet, dann ist das FutureOS natürlich ausgeblendet. Deshalb sind OS Funktionen erforderlich die zuerst die Uhr einblenden, die Zeit-Daten lesen, und dann wieder das FutureOS einblenden. Dies alles wird von LUHR erledigt. Das selbe gilt für die RTC der M4 Karte, die wahlweise verwendet wird.

Bitte Beachten: Vor Aufruf der OS Funktion LUHR muß diese mittels DOBIN (siehe ROM A) ins RAM kopiert worden sein, dies geschieht beim Start des FutureOS.

Sollten aber Zweifel an der Integrität des von den UHR-OS Funktion belegtem RAM bestehen, dann ist es ratsam DOBIN nochmals aufzufufen.

Siehe auch FutureOS Homepage: http://www.FutureOS.de