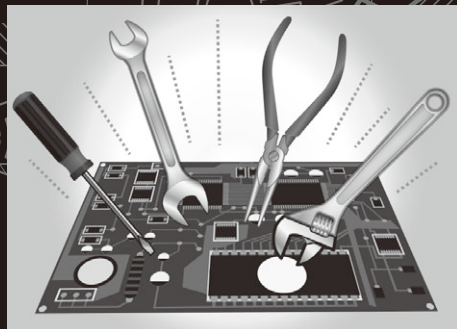


短期連載

組み込みOS チューニング・ テクニック

第2回 省メモリ設計への対応



鹿取 祐二

リスト1 RAM領域に確保されるコントロール・ブロックの個数
(カーネル・オブジェクトのID数)を定義するconfig.hの内容

```
#define CNF_MAX_TSKPRI 32 /* Task Max priority */

/* Maximum number of kernel objects */
#define CNF_MAX_TSKID 32 /* Task */
#define CNF_MAX_SEMID 16 /* Semaphore */
#define CNF_MAX_FLGID 16 /* Event flag */
#define CNF_MAX_MBXID 8 /* Mailbox */
#define CNF_MAX_MTXID 4 /* Mutex */
#define CNF_MAX_MBFID 8 /* Message buffer */
#define CNF_MAX_MPLID 4 /* Memory pool */
#define CNF_MAX_MPPID 8

/* Fixed size memory pool */
#define CNF_MAX_CYCID 4 /* Cyclic handler */
#define CNF_MAX_ALMID 8 /* Alarm handler */

/* Device configuration */
#define CNF_MAX_REGDEV (8)
/* Max registered device */
#define CNF_MAX_OPNDEV (16) /* Max open device */
#define CNF_MAX_REQDEV (16) /* Max request device */
```

2023年7月号特集「ラズパイPicoで1500行ゼロから作るOS」で取り上げたTry Kernelは、1500行で学習するため機能を絞り込んであります。ここで紹介するμT-Kernelは、Try Kernelの発展版と言うこともでき、多くの製品に導入されています。

本連載では、あるマイコン向けに作り込んだOSおよび動作プログラムを、他のマイコンに載せ替える際に、先人がどのような工夫を施しているのかを紹介합니다。

今回は、組み込み向けリアルタイムOSの省メモリ設計を紹介します。解説に当たり、例としてトロンフォーラムと筆者が公開している組み込みシステム向けのリアルタイムOS μT-Kernel 3.0⁽¹⁾のソースコードを取り上げます。(編集部)

RAM領域の削減

リアルタイムOSのソースコードにおいて、RAM領域に確保される変数の大半は各種のコントロール・ブロック(OSの内部情報を集めておく構造体)です。

リスト2 config_func.hで定義されているマクロ名

リスト1のID数を0に設定したい場合はconfig_func.hで定義されているマクロ名を0に設定する

```
#define USE_SEMAPHORE (1)
#define USE_MUTEX (1)
#define USE_EVENTFLAG (0) ← このように0にする
#define USE_MAILBOX (1)
#define USE_MESSAGEBUFFER (1)
#define USE_RENDEZVOUS (1)
#define USE_MEMORYPOOL (1)
#define USE_FIX_MEMORYPOOL (1)
#define USE_TIMEMANAGEMENT (1)
#define USE_CYCLICHANDLER (1)
#define USE_ALARMHANDLER (1)
#define USE_DEVICE (1)
#define USE_FAST_LOCK (1)
#define USE_MULTI_LOCK (1)
```

その代表例はTCB(タスク・コントロール・ブロック)です。これ以外にも各種カーネル・オブジェクトのコントロール・ブロックがRAM領域に確保されます。従って、RAM領域を節約するためには各種コントロール・ブロックの個数を削減することが一番の近道となります。

μT-Kernel 3.0のソースコードの場合、各種コントロール・ブロックの個数は、使用するカーネル・オブジェクトのID数と同じであり、その個数はconfigフォルダ内にあるconfig.hで定義します(リスト1)。

これらの値はシステムで使用する最低値にすることでRAM領域を節約できます。ただしシステムで全く使用しないとしても、0を指定することが許されません。0は、目的の機能を使用しないことを意味しているためです。

もし全く使用しないのであれば、configフォルダ内にあるconfig_func.hで定義されているリスト2のマクロ名を0に設定してください。当該の機能そのものを削除することになり、RAM領域だけでなく、プログラム・コードであるROM領域も節約が可能となります。