

実行 ===== 「[rtk_kernel for Raspberry Pi](#)」の実行手順。

シリアルコンソール

Raspberry Piとパソコンとは、シリアル通信で接続する。Raspberry Piのシリアル通信関係のコネクタピンは下記。尚Raspberry Pi側は3.3V信号なので、シリアル-USB変換ケーブルなどが必要。詳しくは[関連記事\(1.\)](#)を参照。

Raspberry Pi ピン名称	ピン番号	シリアル信号
GND	6	GND
GPIO 14 (TXD)	8	RX
GPIO 15 (RXD)	10	TX

- 通信仕様は下記
ボーレート :115200、データ長 :8bit、パリティ :なし、ストップビット :1、フロー制御 :なし

起動SDカード準備

T-Kernel/T-Monitorプログラムの起動は、[u-boot](#)を利用。T-Monitor単体でも起動はできるのだが、2019/10/21現在 SDカードアクセスは未サポートの為)

ファイル容量は4GByteもあれば十分なのでFATフォーマットされたマイクロSDカードを準備し、下記の合計7のファイルを書き込む。

u-boot関連ファイルの書き込み

GitHubのhttps://github.com/jr4qpv/rpi_u-boot_jtag_binsから、各Raspberry Piの機種用の下記4つのファイル入手し、SDカードのルートに書き込む。詳しくは、[関連記事\(1.\)](#)を参照。

1. bootcode.bin 2. start.elf 3. u-boot.bin 4. config.txt

T-Kernel関連ファイルの書き込み

T-Kernel関連のファイルを、[コンパイル手順](#)で作成した下記3つのファイルを、SDカードに書き込む。

1. tmonitor.bin 2. rominfo-rom.bin 3. kernel-rom.bin

T-Kernelの起動

前記で作成した起動用SDカードを Raspberry Piにセットして電源起動すると、シリアル通信でu-bootの起動メッセージが表示される。カウントダウンするので、その間にキー入力するとu-bootコマンドが入力できる。

続いて、下記コマンドを実行してT-Kernelを起動。

```
u-boot> fatload mmc 0 8000 tmonitor.bin
u-boot> fatload mmc 0 4000 rominfo-rom.bin
u-boot> fatload mmc 0 30000 kernel-rom.bin
u-boot> go 8000
```

起動メッセージが下記のように表示されたら、無事T-Kernelが起動。

```
U-Boot> go 8000
## Starting application at 0x00008000 ...

T-Kernel Version 2.02.00 for RPi2 r0.60 [Thu Oct 10 16:58:26 JST 2019]

ConsoleIO - OK
userinit(0x80000) code not found.
usermain start.
T2EX >>
```

この例では、アプリの実行コードが認識できないので、サンプルアプリ(簡易シェル)の入力待ちとなっている。

u-bootの調整

毎回手動でコマンド入力するのは面倒なので、下記コマンドでT-Kernelを自動起動するようにする。詳しくはu-bootのhelpやドキュメントを参照。

u-bootのコマンド入力待ちから、下記コマンドを入力

```
u-boot> setenv tk_bootcmd 'fatload mmc 0 8000 tmonitor.bin; fatload mmc 0
4000 rominfo-rom.bin; fatload mmc 0 30000 kernel-rom.bin; go 8000'
u-boot> setenv bootcmd 'run tk_bootcmd'
u-boot> setenv bootdelay 2
u-boot> saveenv
```

デバック中など自動でT-Kernelまで起動せずにu-bootのコマンド待ちにしたい場合は、下記コマンドを実行。

```
u-boot> setenv bootdelay -1
u-boot> saveenv
```

電源投入でu-bootのコマンド待ちとなり、bootコマンド入力で、bootcmd変数登録のコマンドが実行されるのでT-Kernelが起動。

自動起動画面例

□Raspberry-Pi 2□で起動した時の画面はこんな感じ。



```
COM1 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)

U-Boot 2016.09-rc1-00469-gdc09431 (Aug 18 2016 - 11:23:43 +0900)

DRAM: 128 MiB
RPI 2 Model B (0xa21041)
MMC: bcm2835_sdhci: 0
reading uboot.env
CACHE: Misaligned operation at range [07ff01d4, 07ff01d8]
In: serial
Out: lcd
Err: lcd
Net: Net Initialization Skipped
No ethernet found.
starting USB...
USB0: Core Release: 2.80a
scanning bus 0 for devices... 3 USB Device(s) found
       scanning usb for storage devices... 0 Storage Device(s) found
       scanning usb for ethernet devices... 1 Ethernet Device(s) found
Hit any key to stop autoboot: 0
reading tmonitor.bin
42104 bytes read in 25 ms (1.6 MiB/s)
CACHE: Misaligned operation at range [00008000, 00012478]
reading rominfo-rom.bin
673 bytes read in 16 ms (41 KiB/s)
CACHE: Misaligned operation at range [00004000, 000042a1]
reading kernel-rom.bin
122880 bytes read in 34 ms (3.4 MiB/s)
## Starting application at 0x00008000 ...

T-Kernel Version 2.02.00 for RPi2 r0.60 [Thu Oct 10 16:58:26 JST 2019]

ConsoleIO - OK
userinit(0x80000) code not found.
usermain start.
T2EX >> ?
test    [arg] ...
ref     [item]
call    addr [p1 p2 p3]
T2EX >> 
```

イーサネットLAN経由でのブート

毎回SDカードに書き込むのも面倒なので、u-bootではTFTPサーバからイーサネット通信でのファイル読み込みも出来る。詳しくは下記記事参照。

[🔗🔗Raspberry Pi u-bootでTFTPサーバからイーサネット経由のファイル読み込み🔗](#)

関連記事

1. [🔗🔗Raspberry Pi u-bootを動かす](#)
2. [🔗🔗Ubuntu 18.04/16.04 LTS Server u-bootをコンパイルする](#)

yt-kernel

From:

<https://www.yokoweb.net/dokuwiki/> - 新石器Wiki

Permanent link:

https://www.yokoweb.net/dokuwiki/develop/rtk_kernel/rtk_kernel-rpi/execution/start?rev=1578374488

Last update: **2020/01/07 14:21**

