

# μ ST-SH2

## μ ST-USB の使用方法

2.1 版 2023 年 10 月 02 日

### 目次

<b>1. 概要</b> .....	<b>1</b>
1.1 Linux について.....	1
1.2 USB について.....	1
<b>2. μ ST-USB について</b> .....	<b>2</b>
2.1 μ ST-USB の概要.....	2
2.2 μ ST-USB の接続.....	3
2.3 デバイスドライバ.....	4
<b>3. μ ST-USB の動作</b> .....	<b>5</b>
3.1 μ ST-USB の動作環境.....	5
3.2 μ ST-USB の設定.....	6
3.3 USB デバイスの使用.....	7
<b>4. 保障とサポート</b> .....	<b>13</b>

## 1. 概要

本アプリケーションノートは USB 拡張オプションボード『 $\mu$ ST-USB』を  $\mu$ ST-SH2 用 Linux で使用方法について述べます。

$\mu$ ST-USB は、ISP1161A (NXP 社) USB ホスト・スレーブコントローラを搭載した USB 拡張オプションボードです。 $\mu$ ST-USB は、USB ホスト・スレーブデバイス両対応で、フルスピード (12M) とロースピード (1.5M) デバイスに対応しています。

$\mu$ ST-SH2 と  $\mu$ ST-USB を組み合わせることにより、Linux USB ホストコントローラとして動作させることができます。

本アプリケーションノートでは、 $\mu$ ST-SH2 用 Linux を使用して、USB メモリの動作方法について説明します。

※ 本アプリケーションノートを実行するには、『 $\mu$ ST-SH2 Linux 開発キット』がインストールされている必要があります。

### 1.1 Linux について

Linux とは 1991 年に Linus Torvalds 氏によって開発された、オープンソースの UNIX 互換オペレーティングシステムです。

Linux はオープンソース、ロイヤリティフリーという特性から、世界中のプログラマたちにより日々改良され、今では大手企業のサーバーや、行政機関などにも広く採用されています。

また、Linux の特長として CPU アーキテクチャに依存しないということがあげられます。これは、GNU C コンパイラの恩恵にもよるものですが、数多くのターゲット (CPU) に移植されており、デジタル家電製品を中心に非 PC 系製品にも採用されるようになりました。

Linux の詳細については、一般書籍やインターネットから多くの情報を得られますので、それらを参考にしてください。

### 1.2 USB について

USB (Universal Serial Bus) は Compaq (現 Hewlett-Packard)、Intel、Microsoft、NEC の四社により策定された PC と周辺機器とのデータ転送方式の一つです。USB は現在バージョン 3.0 が最新版となります。

USB は通信をすべてホスト側で制御するようになっており、ホストと各デバイス間の通信では必ずホストから各デバイスに対して転送要求を行います。USB は、1 台のホストからハブを介することによって、最大 127 個のデバイスを接続することができます。

USB の詳細については、インターネット上に規格書が公開されておりますので、そちらをご覧ください。

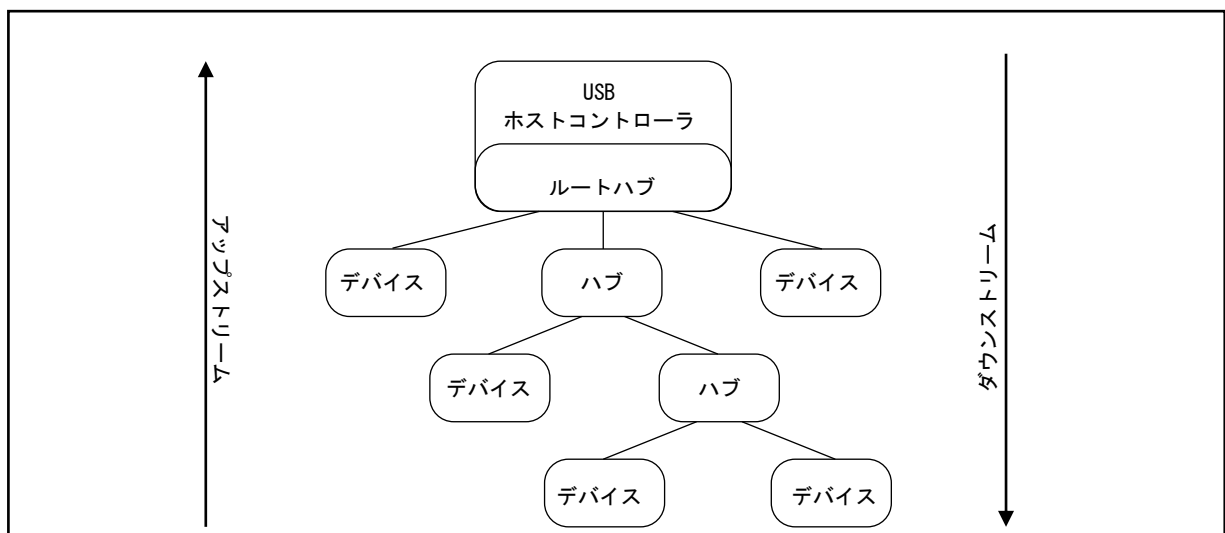


Fig1.2-1 USB のバスターノロジー

## 2. $\mu$ ST-USB について

### 2.1 $\mu$ ST-USB の概要

$\mu$  ST-USB は USB2.0 の Full Speed (12Mbps) と Low Speed (1.5Mbps) に対応した USB 拡張オプションボードです。

$\mu$  ST-USB は弊社 CPU ボード『 $\mu$  ST-SH2』と組み合わせることにより、LinuxUSB ホストコントローラとして動作することができます。

$\mu$  ST-SH2 用 Linux には  $\mu$  ST-USB 用ホストデバイスドライバが組み込まれており、USB マスタレージ規格に準拠する USB フラッシュメモリを動作させることができます。

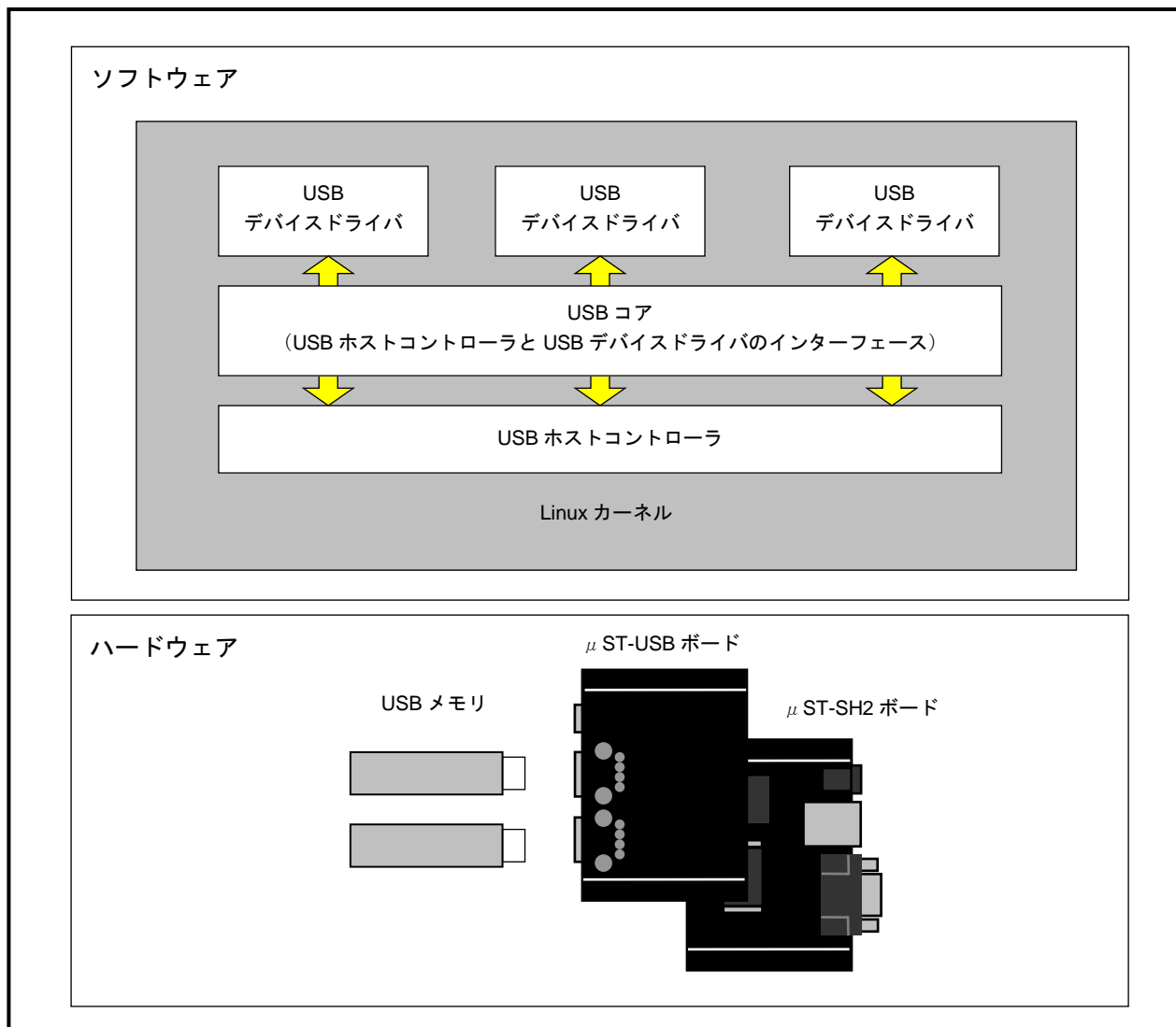


Fig 2.1-1  $\mu$  ST-SH2 の概要

## 2.2 $\mu$ ST-USB の接続

$\mu$  ST-USB は  $\mu$  ST-SH2 と組み合わせることにより、LinuxUSB ホストコントローラとして動作させることができます。下図に  $\mu$  ST-USB および  $\mu$  ST-SH2 を使用したときの USB メモリとの接続例を示します。

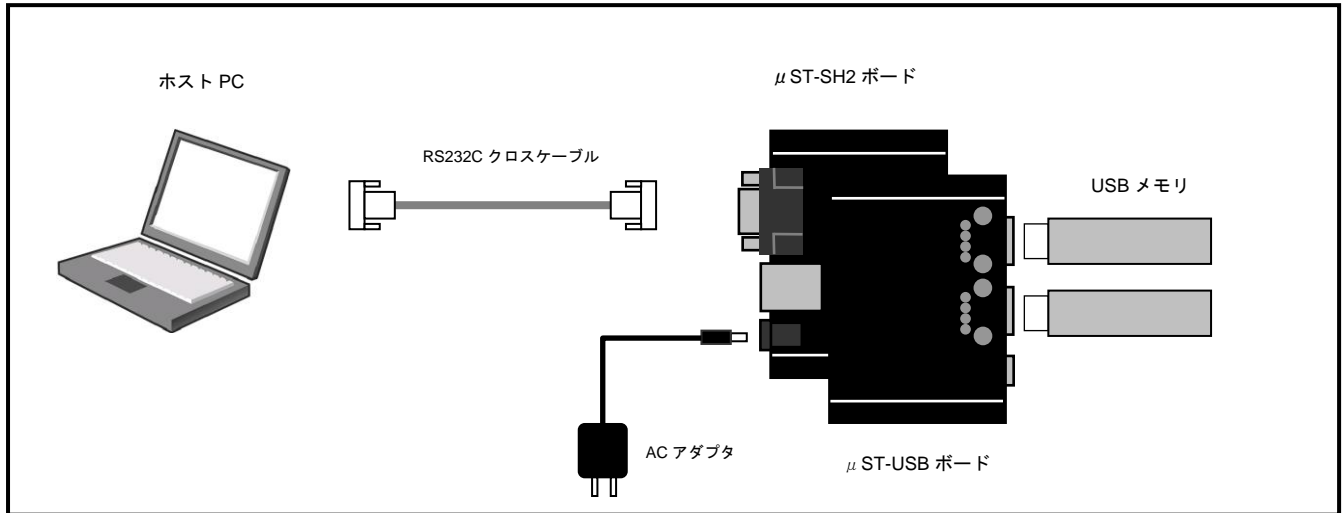


Fig 2.2-1 USB メモリとの接続

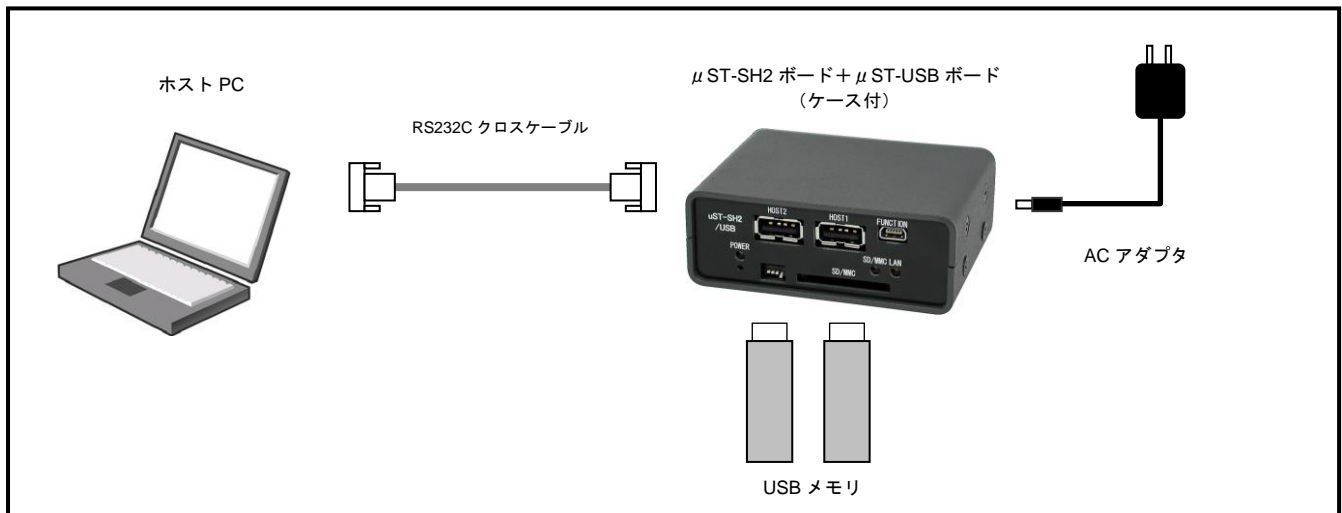


Fig 2.2-2 USB メモリとの接続 (ケース付)

### 2.3 デバイスドライバ

Linux の USB ドライバは3階層に分かれています。最上位層は各 USB ドライバを制御する USB デバイスドライバ、中間層は各 USB デバイスドライバのロード・アンロードや USB デバイスドライバと最下位層の HCD との仲介を行う API を提供する USB コア、最下位層は実際 USB 転送などのハードウェアの制御を行う HCD (ホストコントローラドライバ) となります。

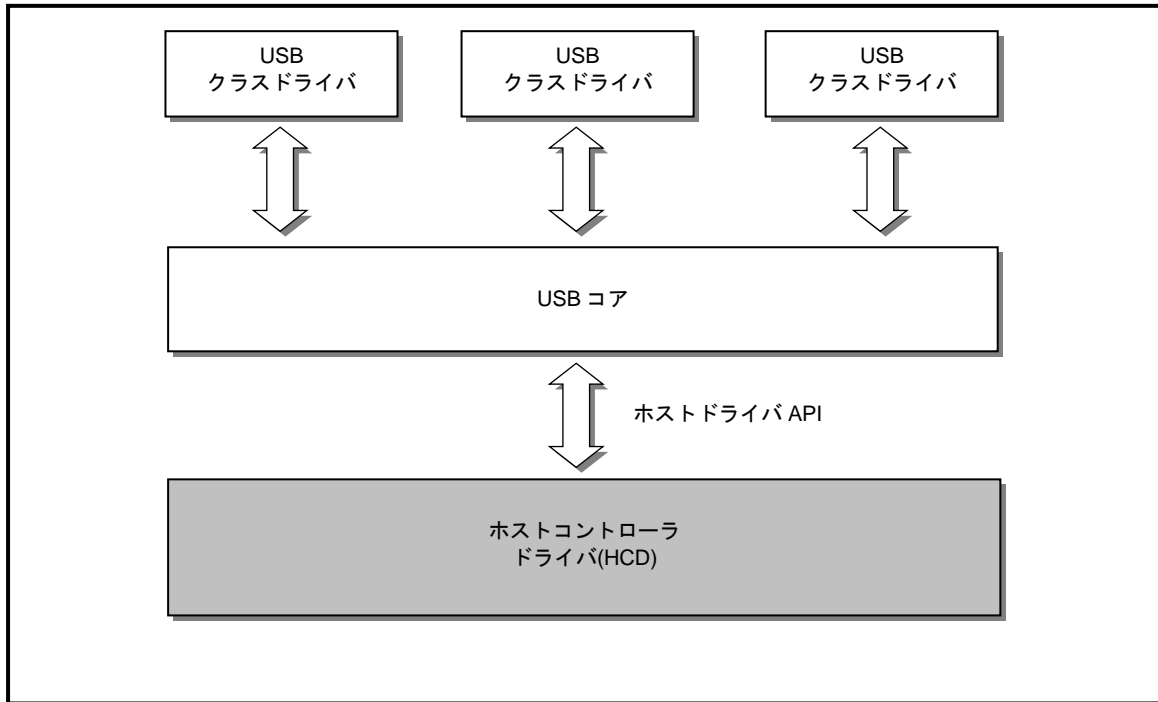


Fig 2.3-1 Linux USB ドライバ

```
Linux カーネルソースディレクトリ
|--arch
| |--sh
|   |--boards
|     |--mach-ustsh2
|       |--usb.c           :   ISP116x コンフィグレーションデータ
|                           :
|--drivers
  |--usb
    |--host
      |-- isp116x-hcd.c     :   ISP116x ホストコントローラ
      |-- isp116x.h        :   ISP116x ホストコントローラヘッダファイル
      :
      :
```

Fig 2.3-2 LinuxUSB ホストコントローラデバイスソースファイル構成

### 3. $\mu$ ST-USB の動作

$\mu$  ST-USB と  $\mu$  ST-SH2 を使用して、USB デバイスを動作させる手順について説明します。

#### 3.1 $\mu$ ST-USB の動作環境

$\mu$  ST-USB の動作を確認するためには  $\mu$  ST-SH2 を含め以下の環境が必要です。

- ホスト PC

$\mu$  ST-SH2 用 Linux では PC をコンソール端末として使用しますので、Linux の起動を確認するためにはシリアルポートが使用可能な PC が必要となります。ホスト PC では、ハイパーターミナル等のターミナルソフトウェアを動作させます。

- 電源

$\mu$  ST-SH2 本体に必要な電源は DC5V $\pm$ 5% です。AC アダプタを用意してください。

- シリアル

$\mu$  ST-SH2 と PC をシリアルで接続する場合、付属の RS232C クロスケーブル (RxD、TxD、RTS、CTS、GND) をご使用ください。

- USB メモリ

Linux 上での USB ホストの機能を検証するために USB メモリを使用します。

Table 3.1-1  $\mu$ ST-USB 動作環境

使用機器等	環 境
USB オプションボード	$\mu$ ST-USB
Linux ボード	$\mu$ ST-SH2
ホスト PC	PC/AT 互換機
ホスト OS	Windows7/XP (WindowsXP 推奨) ※1
ソフトウェア	ターミナルソフト
ドライブ	DVD-ROM 読み込み可能なドライブ
シリアルポート	1 ポート
USB メモリ	USB マスストレージクラス対応 ※2
RS232C ケーブル	クロスケーブルを使用
電源	DC5V $\pm$ 5% 1A 以上 (AC アダプタ)

※1 WindowsVista につきましては VMware Player のサポート対象外のため、動作環境の対応 OS に含まれておりません。

※2  $\mu$  ST-SH2 用 Linux 上では USB ハードディスクなど容量が大きいストレージデバイスを接続するとディレクトリ探索 (ls 等のコマンド) が非常に遅いという現象が見られます。

## 3.2 $\mu$ ST-USB の設定

$\mu$  ST-SH2 用 Linux のために  $\mu$  ST-USB ボードの設定を行います。

※ ケース付き  $\mu$  ST-SH2 +  $\mu$  ST-USB ボードは出荷時設定が以下の設定になっています。

$\mu$  ST-USB のチップセレクトを CS4 に設定します。

$\mu$  ST-USB の JSW1 スイッチを『CS4』に選択します。

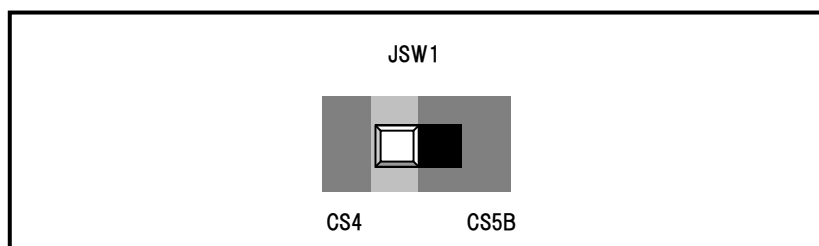


Fig 3.2-1 JSW1 の設定

### 3.3 USB デバイスの使用

$\mu$  ST-SH2 用 Linux では USB デバイスとしてストレージデバイスを扱うことができます。  
以下では、USB メモリの使用方法について説明します。

#### $\mu$ ST-USB ホストコントローラ対応 Linux カーネル

$\mu$  ST-USB を使用するには、 $\mu$  ST-USB ホストコントローラ対応 Linux カーネルを起動する必要があります。

$\mu$  ST-SH2 用 Linux カーネルはカーネルパラメータに『ustusb』を渡すことにより  $\mu$  ST-USB ホストコントローラ対応 Linux カーネルを起動することができます。

$\mu$  ST-SH2 では U-Boot の環境変数『bootargs』の値を設定することにより、カーネルパラメータを指定します。カーネルパラメータとして『console=ttyS0,115200 ustusb』を使用します。

環境変数『bootargs』の設定

```
=> setenv bootargs console=ttyS0,115200 ustusb
```

Table3.3-1 カーネルパラメータ

カーネルパラメータ	値	概要
console	ttyS0,115200	コンソールの指定を行います。 コンソールとして ttyS0 シリアルデバイスをボーレート 115200bps で使用することを示しています。
ustusb	-	$\mu$ ST-USB ホストコントローラ対応 Linux カーネルを起動します。

- 『2.2  $\mu$  ST-USB ボードの接続』にしたがって、 $\mu$  ST-USB と  $\mu$  ST-SH2 を接続し、ホスト PC のシリアルポートとを接続します。
- $\mu$  ST-SH2 の電源を投入しホスト PC のコンソール上で何らかのキーを入力します。  
ブートローダの自動起動がキャンセルされ、コンソールの起動ログが表示されます。

```
U-Boot 2010.03 ustsh2-1.0 ( 6月 16 2010 - 13:09:33)

CPU: SH2
BOARD: ALPHAPROJECT  $\mu$ ST-SH2
DRAM: 32MB
FLASH: 8MB
In: serial
Out: serial
Err: serial
Net: sh_eth
Hit any key to stop autoboot: 0 ← 何かキーを入力する
=>
```

- 環境変数『bootargs』の値を『console=ttyS0,115200 ustusb』と設定します。

```
=> setenv bootargs console=ttyS0,115200 ustusb ←
=>
```



- ④ フラッシュ ROM から Linux システムを起動します。

『bootm a0100000』を実行してください。

```
=> bootm a0100000 ←  
## Booting kernel from Legacy Image at a0100000 ...  
Image Name: Linux-2.6.33.1  
Created: 2010-06-16 2:23:39 UTC  
Image Type: SuperH Linux Kernel Image (gzip compressed)  
Data Size: 2552311 Bytes = 2.4 MB  
Load Address: 0c001000  
Entry Point: 0c002000  
Verifying Checksum ... OK  
Uncompressing Kernel Image ... OK  
以下省略 . . .  
  
Welcome to Buildroot  
buildroot login:
```

## USB デバイスファイルシステム

USB デバイスファイルシステムは USB の状態を /proc ファイルに出力します。主にデバッグ用ツールとして用いられます。

- ① 『2.2  $\mu$ ST-USB ボードの接続』にしたがって、 $\mu$ ST-USB と  $\mu$ ST-SH2 を接続し、ホスト PC のシリアルポートを接続し、 $\mu$ ST-USB ホストコントローラ対応 Linux カーネルで Linux を起動します。

```
=> setenv bootargs console=ttyS0,115200 ustusb ←入力
=> bootm a0100000 ←入力
## Booting kernel from Legacy Image at a0100000 ...
   Image Name:   Linux-2.6.33.1
   Created:      2010-06-16  2:23:39 UTC
   Image Type:   SuperH Linux Kernel Image (gzip compressed)
   Data Size:    2552311 Bytes =  2.4 MB
   Load Address: 0c001000
   Entry Point:  0c002000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
以下省略 . .
```

- ②  $\mu$ ST-SH2 上で Linux の起動を確認し、root 権限でログインします。

```
Welcome to Buildroot
buildroot login: root ←入力
~ #
```


- ③ デバッグファイルシステムを『/proc/sys/debug』ディレクトリにマウントします。  
『mount -t debugfs none /proc/sys/debug』を実行してください。

```
~ # mount -t debugfs none /proc/sys/debug ←入力
~ #
```

- ④ 『/proc/sys/debug/usb/』ディレクトリの中身を確認します。

```
~ # ls /proc/sys/debug/usb ←入力
devices ←確認
~ #
```

- ⑤ USB ホストコントローラデバイスを確認します。

```
~ # cat /proc/sys/debug/usb/devices   
  
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 2  
B: Alloc= 0/900 us ( 0%), #Int= 0, #Iso= 0  
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS=64 #Cfgs= 1  
P: Vendor=1d6b ProdID=0001 Rev= 2.06  
S: Manufacturer=Linux 2.6.33.1 isp116x-hcd  
S: Product=ISP116x Host Controller  
S: SerialNumber=isp116x-hcd  
C:* #Ifs= 1 Cfg#= 1 Atr=e0 MxPwr= 0mA  
I:* If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub  
E: Ad=81(I) Atr=03(Int.) MxPS= 2 IvI=255ms  
~ #
```

『/proc/sys/debug/usb/devices』ファイルの書式の詳細については <http://www.linux-usb.org/USB-guide/book1.html> をご覧下さい。

## USB マスストレージデバイス

USB マスストレージデバイスは FDD や HDD など外部記憶装置をサポートします。USB マスストレージクラスは外部記憶装置をサポートするための USB 規格として制定されています。

Linux 上では USB マスストレージデバイスを SCSI デバイスとして認識します。

- ① 『2.2  $\mu$ ST-USB ボードの接続』にしたがって、 $\mu$ ST-USB と  $\mu$ ST-SH2 を接続し、ホスト PC のシリアルポートとを接続し、 $\mu$ ST-USB ホストコントローラ対応 Linux カーネルを使用して Linux を起動します。

```
=> setenv bootargs console=ttySC0,115200 ustusb ←入力
=> bootm a0100000 ←入力
## Booting kernel from Legacy Image at a0100000 ...
   Image Name:   Linux-2.6.33.1
   Created:      2010-06-16  2:23:39 UTC
   Image Type:   SuperH Linux Kernel Image (gzip compressed)
   Data Size:    2552311 Bytes =  2.4 MB
   Load Address: 0c001000
   Entry Point:  0c002000
   Verifying Checksum ... OK
   Uncompressing Kernel Image ... OK
以下省略・・・
```

- ②  $\mu$ ST-SH2 上で Linux の起動を確認し、root 権限でログインします。

```
Welcome to Buildroot
buildroot login: root ←入力
~ #
```

- ③  $\mu$ ST-USB の CN1 (USB コネクタシリーズ A) に FAT 形式でフォーマットされた USB メモリを挿入してください。接続されたデバイスファイルが『/dev/sda1』であることを確認します。

```
~ # usb 1-1: new full speed USB device using isp116x-hcd and address 2
scsi0 : usb-storage 1-1:1.0
scsi 0:0:0:0: Direct-Access   xxxxxxxx xxxxxxxx      8.07 PQ: 0 ANSI: 2
sd 0:0:0:0: [sda] 1986558 512-byte logical blocks: (1.01 GB/969 MiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1 ←入力
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] Attached SCSI removable disk
```

USB フラッシュ  
メモリ挿入時の  
メッセージ

- ④ FAT 形式で USB メモリをマウントします。

マウント時にエラーメッセージが出力されなければ、通常のディスクとして扱うことができます。

```
~ # mkdir /mnt/usb1 ←入力
~ # mount -t vfat /dev/sda1 /mnt/usb1 ←入力
```

※ 上記の方法でマウントできない場合はオプション『-o iocharset=cp932』を追加してお試しください。

コマンドは『mount -t vfat -o iocharset=cp932 /dev/sda1 /mnt/usb1』となります。

※ USB メモリを抜く際は必ずアンマウントを実行してください。

```
~ # umount /dev/sda1 ←入力
```

- ⑤  $\mu$  ST-USB の CN2 (USB コネクタシリーズ A) に二つめの FAT 形式でフォーマットされた USB メモリを挿入してください。

接続されたデバイスファイルが『/dev/sdb1』であることを確認します。

```
~ # usb 1-2: new full speed USB device using isp116x-hcd and address 3
scsi1 : usb-storage 1-2:1.0
scsi 1:0:0:0 Direct-Access xxxxxxxx xxxxxxxx BC01 PQ: 0 ANSI: 0 CCS
sd 1:0:0:0: [sdb] 1982464 512-byte logical blocks: (1.01 GB/968 MiB)
sd 1:0:0:0: [sdb] Write Protect is off
sd 1:0:0:0: [sdb] Assuming drive cache: write through
sd 1:0:0:0: [sdb] Assuming drive cache: write through
sdb: sdb1 ←入力
sd 1:0:0:0: [sdb] Assuming drive cache: write through
sd 1:0:0:0: [sdb] Attached SCSI removable disk
```

USB フラッシュ  
メモリ挿入時の  
メッセージ

- ⑥ FAT 形式で USB メモリをマウントします。

マウント時にエラーメッセージが出力されなければ、通常のディスクとして扱うことができます。

```
~ # mkdir /mnt/usb2 ←入力
~ # mount -t vfat /dev/sdb1 /mnt/usb2 ←入力
```

※ 上記の方法でマウントできない場合はオプション『-o iocharset=cp932』を追加してお試しください。

コマンドは『mount -t vfat -o iocharset=cp932 /dev/sda1 /mnt/usb1』となります。

※ USB メモリを抜く際は必ずアンマウントを実行してください。

```
~ # umount /dev/sdb1 ←入力
```

## 4. 保障とサポート

弊社では最低限の動作確認をしておりますが、Linux および付属ソフトウェアの性能や動作を保証するものではありません。  
また、これらのソフトウェアについての個別のお問い合わせ及び技術的な質問は一切受け付けておりませんのでご了承ください。  
個別サポートをご希望されるお客様には、別途有償サポートプログラムをご用意しておりますので、弊社営業までご連絡ください。

Linux など付属する GPL ソフトウェアのソースコードは、ユーザ登録をすることにより弊社ホームページより全てダウンロードすることができます。

また、これらのソフトウェアは不定期にバージョンアップをおこない、ホームページ上で公開する予定です。

ユーザ登録は弊社ホームページにて受け付けております。ユーザ登録をしていただきますと、バージョンアップや最新の情報等を E-mail でご案内させていただきますので、是非ご利用ください。

弊社ホームページアドレス <https://www.apnet.co.jp>

## 参考文献

- 「LINUX デバイスドライバ 第3版」 Alessandro rubini, Jonathan corbet, Greg Krooah-Hartman 著  
山崎康宏、山崎邦子、長原宏治、長原陽子 訳／オライリージャパン
- 「詳解 LINUX カーネル 第3版」 Daniel P. Bovet, Marco Cesati 著 高橋弘和 監訳  
岡島順治朗、田宮まや、三浦広志 訳／オライリージャパン
- 「USB ハード&ソフト開発のすべて」 CQ 出版社

その他 各社データシート

<http://www.usb.org/>

USB Implementers Forum ホームページ。各種 USB 規格書をダウンロードできます。

<http://www.linux-usb.org>

Linux USB プロジェクト ホームページ。USB の Linux への移植を行っています。

## 謝辞

Linux、SH-Linux の開発に関わった多くの貢献者に深い敬意と感謝の意を示します。

## 著作権について

- ・本文書の著作権は、株式会社アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは、一切禁止します。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書の内容については万全を期して作成いたしました。万が一不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。
- ・本文書の内容に基づきアプリケーションを運用した結果、万が一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

## 商標について

- ・ SuperH は、ルネサス エレクトロニクス株式会社の登録商標、商標または商品名称です。
- ・ Linux は、Linus Torvalds の米国およびその他の国における登録商標または商標です。
- ・ Windows® の正式名称は、Microsoft® Windows® Operating System です。
- ・ Microsoft、Windows は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。
- ・ Windows® XP、Windows® Vista、Windows® 7 は、米国 Microsoft Corporation の商品名称です。

本文書では下記のように省略して記載している場合がございます。ご了承下さい。

Windows® XP は WindowsXP もしくは WinXP

Windows® Vista は WindowsVista もしくは WinVista

Windows® 7 は Windows7 もしくは Win7

- ・ その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト

〒431-3114

静岡県浜松市中央区積志町 834

<https://www.apnet.co.jp>

E-Mail: [query@apnet.co.jp](mailto:query@apnet.co.jp)