

GR-MANGO プロデューサーミーティング

ライブラリ & サンプルプログラムのご紹介

ルネサス エレクトロニクス株式会社

GR-MANGO は Mbed OSに対応します

Mbed OS（エンベット オーエス）は、IoTデバイスのために特別に設計された無料のオープンソース組み込みオペレーティングシステムです。

セキュリティ、接続性、RTOS、センサーやI/Oデバイス用のドライバなど、Armマイクロコントローラをベースにしたコネクテッド製品を開発するために必要なすべての機能が含まれています。

<https://www.mbed.com/en/platform/mbed-os/>

GR-MANGO βボードで使用できるMbed OSライブラリ

GR-MANGO βボードは下記のライブラリが使用できます。

オフィシャルのmbed-osライブラリではご利用になれませんのでご注意ください。

mbed-os

Mbed OS 5.12.4にGR-MANGOを含むRZ/A2Mボードを追加したライブラリです。

https://github.com/d-kato/mbed-os/tree/feature_RZA2_base5.12.4

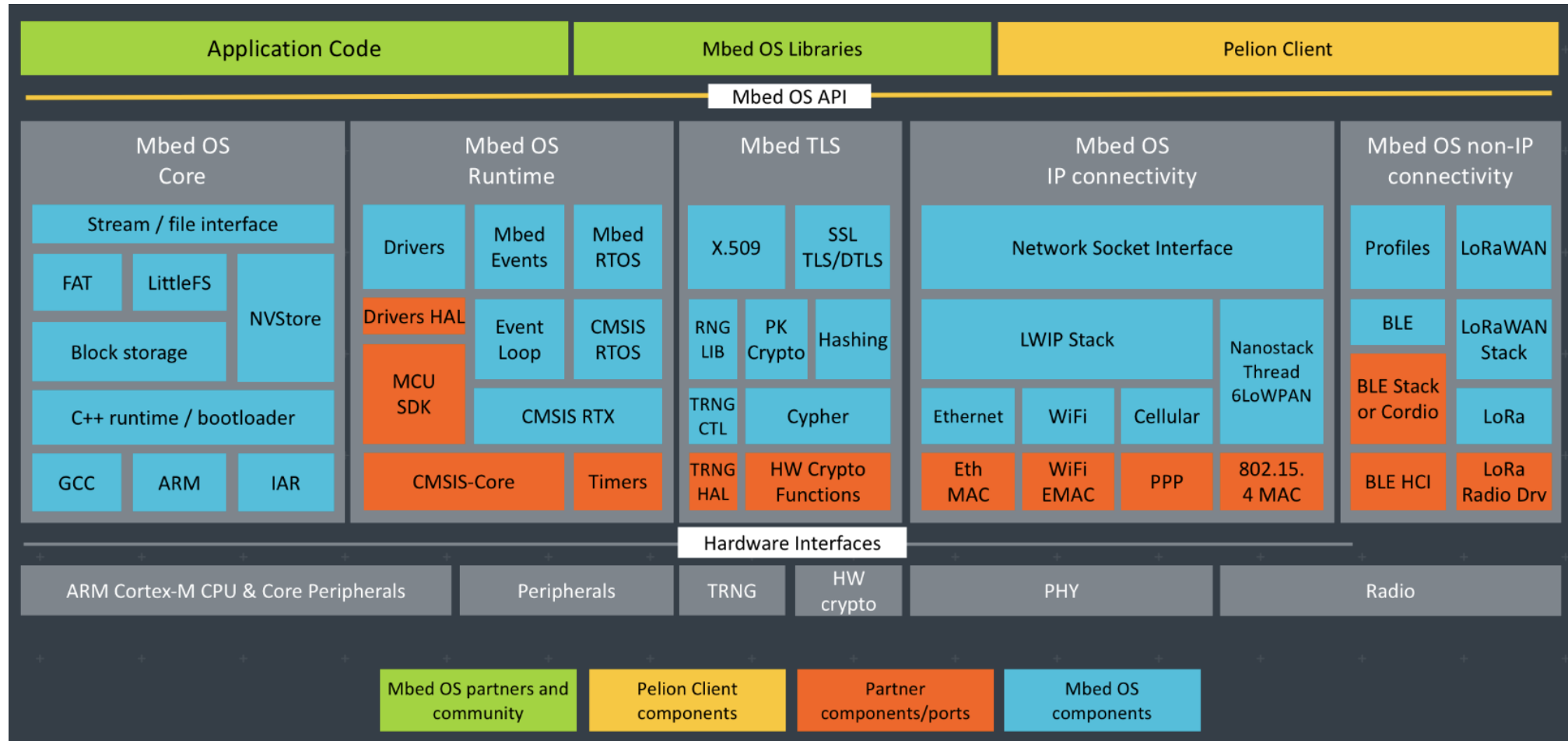
mbed-gr-libs

GR-PEACH、GR-LYCHEE、RZ/A2Mボード（GR-MANGO含む）用のライブラリ群です。

カメラ入力やLCD表示など、Mbed OSで定義されていないドライバなどが用意されています。

<https://github.com/d-kato/mbed-gr-libs>

アーキテクチャ概略図



出典 : <https://os.mbed.com/docs/mbed-os/v5.12/introduction/index.html>

対応ドライバー一覧

	ARMCC	GCC ARM	IAR Tool
CPU	✓	✓	✓
DMAC	✓	✓	✓
INTC	✓	✓	✓
MTU3a			
OSTM	✓	✓	✓
Watchdog Timer			
RTC	✓	✓	✓
USB Host*1			
USB Function	✓	✓	✓
NAND			
SDHI*3	✓	✓	✓
MMC			
I2C (RIIC)	✓	✓	✓
Sync Serial			
UART (SCIF)	✓	✓	✓
SPI (RSPI)	✓	✓	✓
SCI 9bit			
SCI Smart Card			

	ARMCC	GCC ARM	IAR Tool
I2S (SSIF) *1	✓	✓	✓
SPDIF*1	✓	✓	✓
CANFD			
Ether	✓	✓	✓
DRV(W)			
VDC6*1	✓	✓	✓
LVDS*1	✓	✓	✓
IMR-LS2			
JCU*1	✓	✓	✓
CEU*1	✓	✓	✓
MIPI*1	✓	✓	✓
VIN*1	✓	✓	✓
ADC	✓	✓	✓
PWM (GPT)	✓	✓	✓
SPEA			
HyperBus Ctrl	✓	✓	✓
Octa Memory Ctrl	✓	✓	✓
eSD Boot			

	ARMCC	GCC ARM	IAR Tool
eMMC Boot			
SPI Multi Boot (Serial Flash)	✓	✓	✓
SPI Multi Boot (OctaFlash)			
SPI Multi Boot (HyperFlash)			
OctaFlash Boot	✓	✓	✓
HyperFlash Boot	✓	✓	✓
Security Library			
Secure Boot			
DRP*1	✓	✓	✓

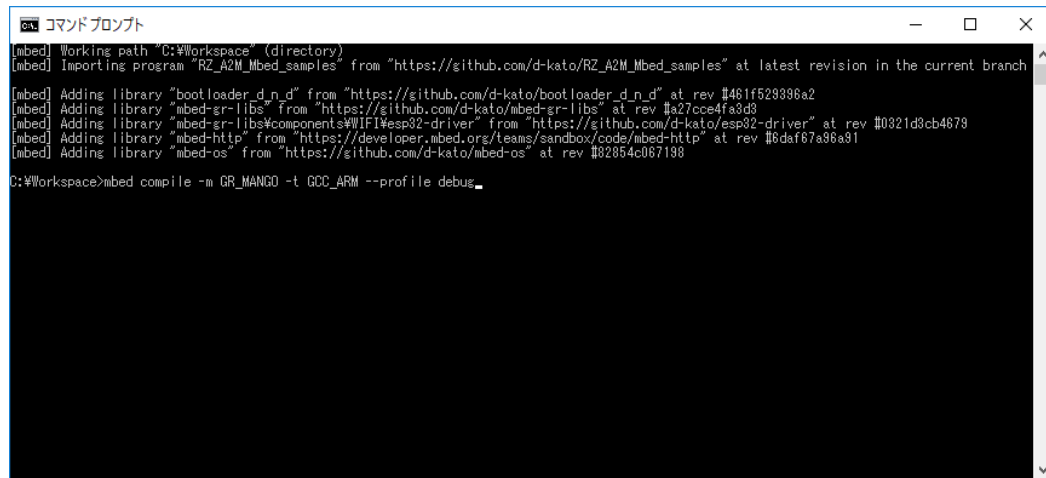
- *1: Mbed OS APIではなく、Mbed OS Librariesとして提供
- *2: Renesas Graphics Architecture
- *3: SDスピードクラスがご利用になれます。
UHSスピードクラスが必要なお客様は最寄りの営業担当者または代理店にお問い合わせください。

凡例
✓: 提供可能な状態

開発環境

Mbed CLI (CUI)、または、Mbed Studio (GUI) を使用できます。好みの開発環境を選択してください。
インストール方法、使い方については [こちら](#) を参照ください。

Mbed CLI



```
コマンドプロンプト
[mbed] Working path "C:\Workspace" (directory)
[mbed] Importing program "RZ_A2M_Mbed_samples" from "https://github.com/d-kato/RZ_A2M_Mbed_samples" at latest revision in the current branch

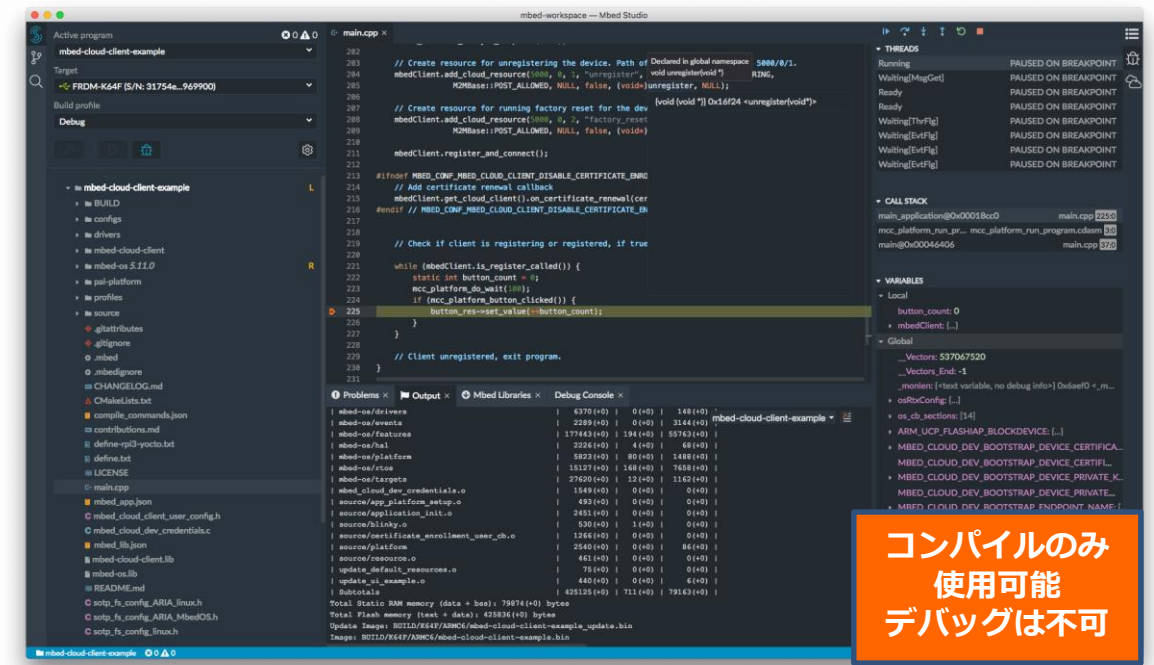
[mbed] Adding library "boot_loader_d_n_d" from "https://github.com/d-kato/boot_loader_d_n_d" at rev #461f529396a2
[mbed] Adding library "mbed-gr-lib" from "https://github.com/d-kato/mbed-gr-lib" at rev #a270ce4fa3d8
[mbed] Adding library "mbed-gr-lib%components%WIFI%esp82-driver" from "https://github.com/d-kato/esp82-driver" at rev #0321d3cb4678
[mbed] Adding library "mbed-http" from "https://developer.mbed.org/teams/sandbox/code/mbed-http" at rev #6daf67a98a31
[mbed] Adding library "mbed-os" from "https://github.com/d-kato/mbed-os" at rev #82954c087198

C:\Workspace>mbed compile -m GR_MANGO -t GCC_ARM --profile debug
```

ビルドコマンド

```
$ mbed compile -m GR_MANGO -t GCC_ARM --profile debug
```

Mbed Studio



コンパイルのみ
使用可能
デバッグは不可

サンプルプログラム

Mbed OS サンプルプログラム集

https://github.com/d-kato/RZ_A2M_Mbed_samples

MIPIカメラとDRPを使ったWebカメラサンプル

https://github.com/d-kato/RZ_A2M_WebCamera_MIPI

DRPサンプルプログラム集

https://github.com/d-kato/RZ_A2M_DRP_samples

オムロン 形2JCIE-EV センサ評価ボード デモ

https://github.com/d-kato/GR-MANGO_2JCIE-EV01-RP1_sample

オムロン 形D6T MEMS非接触温度センサ 映像出力デモ

https://github.com/renesas-rz/GR-MANGO_D6T_44L_06_sample

プログラムの書き込み方法

プログラムの書き込みはとても簡単です。バイナリファイルをdrag & dropするだけで書き込みが完了します。

先ほど紹介したGR-MANGOのサンプルは **Custom bootloader** に対応しています。これらのサンプルは **DAPLink** を使った書き込みと **Custom bootloader** を使った書き込みの両方をお試しいただけます。

DAPLink

Mbedボードに搭載されているDAPLinkマイコンを介してバイナリファイルを書き込みます。

Custom bootloader New

DAPLinkを介さずに、RZ/A2MのUSB端子に直接バイナリファイルを送信します。

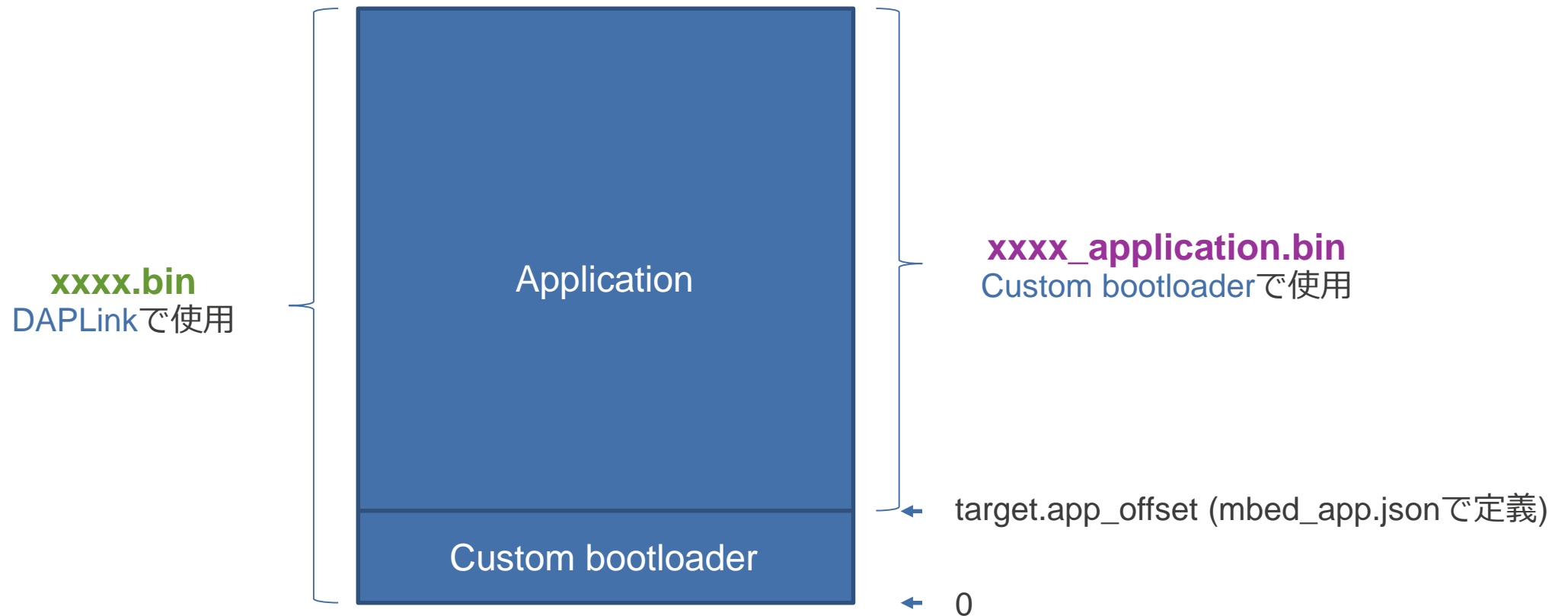
DAPLink書き込み
の約5倍の速度

初めて使う際はボードFlashの先頭にCustom bootloaderを書き込む必要があります。(DAPLink使用)

詳細は [こちら](#) を参照ください。

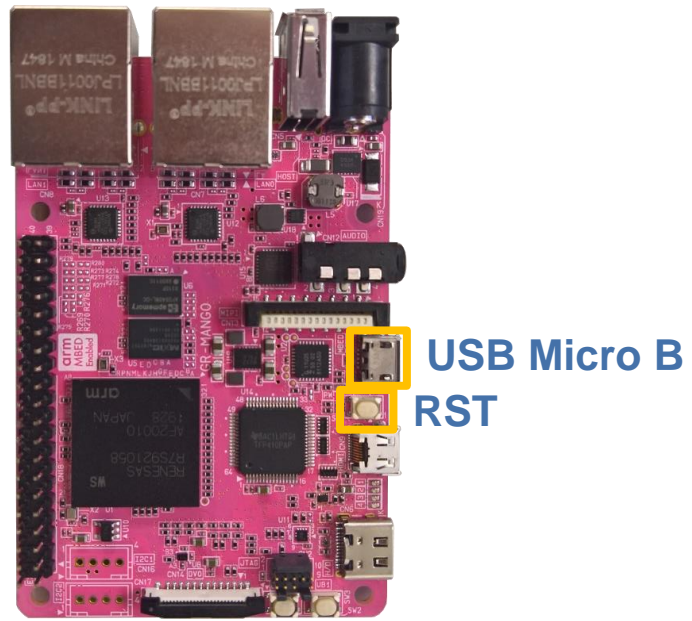
フラッシュレイアウト

Custom bootloaderに対応したサンプルをビルドすると「**xxxx.bin**」と「**xxxx_application.bin**」の2つのバイナリファイルが生成されます。



DAPLink の使い方

1. **USB Micro B** とPCを接続する。
2. PC上に **MBED** というドライブが見えるようになるので、「**xxxx.bin**」をコピーする。
3. 書き込みが終わったら **RST** (Resetボタン)を押す。



MBEDドライブが見えない場合

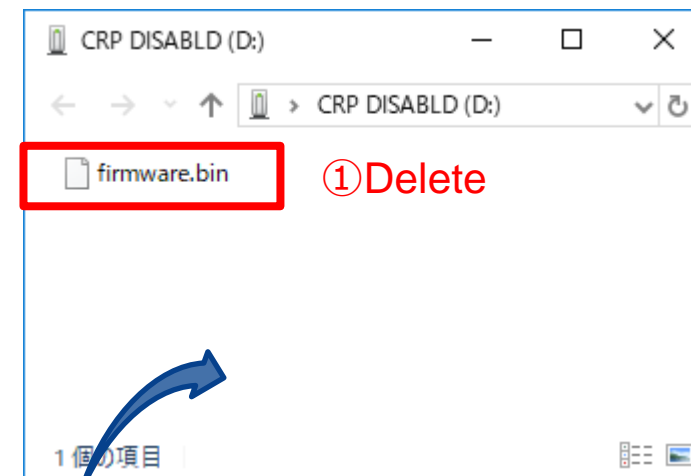
RST を押したままPCに接続すると **CRP DISABLD** ドライブが表示されます。**RST** が押されていないか確認してください。

Resetボタンを押していないのに **CRP DISABLD** ドライブが表示される場合はDAPLinkファームウェアが書き込まれていない可能性があります。

DAPLinkファームウェアを [こちら](#) より入手し、DAPLinkの更新を行ってください。

アップデート手順

1. **CRP DISABLD** ドライブの firmware.bin を削除する。
2. lpc11u35_gr_mango_20191212.bin をコピーする。

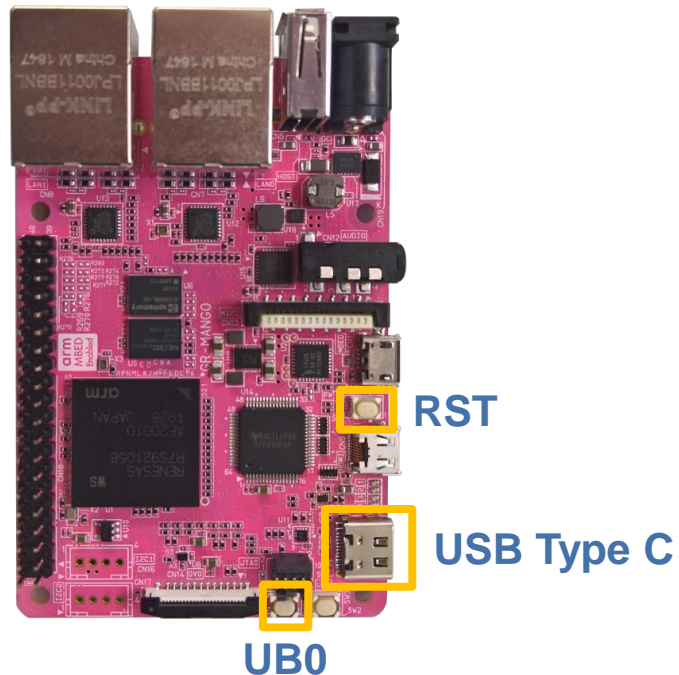


② Drag and drop

lpc11u35_gr_mango_20191212.bin

Custom bootloader の使い方

1. **USB Type C** とPCを接続する。
2. **UB0** を押しながら **RST** を押す。
3. PC上に **GR-MANGO** というドライブが見えるようになるので、「**xxxx_application.bin**」をコピーする。
4. 書き込みが終わったら **RST** を押す。



ターミナル出力先を USB Type C にする方法

「mbed_app.json」にてマクロ「`OVERRIDE_CONSOLE_USBSERIAL`」を追加することで、DAPLinkの代わりにUSB Type Cからターミナル出力を送信することができるようになります。

例) GR-MANGO_2JCIE-EV01-RP1_sample の場合 (赤字部分を追加)

```
{
  "target_overrides": {
    "*": {
      "platform.stdio-baud-rate": 115200,
      "platform.stdio-convert-newlines": true,
      "target.macros_add": ["MBED_CONF_APP_MAIN_STACK_SIZE=8192"]
    },
    "GR_MANGO": {
      "target.bootloader_img": "bootloader_d_n_d/GR_MANGO_boot.bin",
      "target.app_offset": "0x11000",
      "target.macros_add": ["OVERRIDE_CONSOLE_USBSERIAL"]
    }
  }
}
```

非キャッシュメモリの使い方

GR-PEACH、GR-LYCHEEでは、使用の有無に関わらず常に1MBの非キャッシュメモリ領域が確保されていました。
GR-MANGOでは、プログラムで使用された分だけ非キャッシュメモリ領域が確保されます。

変数やバッファを非キャッシュメモリとして配置する場合は、宣言時に「`__attribute((section("NC_BSS")))`」を追加します。キャッシュ制御については [こちら](#) を参照ください。

```
// キャッシュ有効
static uint8_t cache_buf[64];

// 非キャッシュメモリ __attribute の位置は前でも後ろでもOK
static uint8_t non_cache_buf_0[64] __attribute((section("NC_BSS")));
static __attribute((section("NC_BSS"))) uint8_t non_cache_buf_1[64];
```

OctaRAMの使い方

GR-MANGO βボードには8MBの OctaRAM が搭載されています。(正式版では16MBを予定)

変数やバッファをOctaRAMに配置する場合は、宣言時に「`__attribute((section("OCTA_BSS")))`」を追加します。

初期値ありメモリをOctaRAMに配置する場合は、「`__attribute((section("OCTA_DATA")))`」を使用します。

OCTA_BSS、OCTA_DATAはキャッシュ有効となります。キャッシュ制御については [こちら](#) を参照ください。

```
// OctaRAM 初期値なし __attribute の位置は前でも後ろでもOK
// キャッシュ制御することを考慮してaligned(32)としておく
static uint8_t octa_buf_0[64] __attribute((section("OCTA_BSS"),aligned(32)));
static __attribute((section("OCTA_BSS"),aligned(32))) uint8_t octa_buf_1[64];

// OctaRAM 初期値あり
static __attribute((section("OCTA_DATA"))) uint8_t octa_data = 0x01;
```


SDカードに高速アクセスするために

GR-MANGOには Non-NDA版SDドライバ が搭載されており、SDスピードクラスでの通信が可能です。

ですが、普通に使用していてもあまり速さを感じないかもしれません。実は、SDカードは一度にたくさんの通信を行うことでアクセスが速くなるデバイスなのです。

“**fopen()**” でファイルオープンするとファイルシステム内で **512byte** のバッファが自動的に割り当てられます。ブロックデバイス へのアクセスはこのバッファ単位で行われます。バッファサイズは “**setvbuf()**” で変更可能で、バッファリング無しに設定することもできます。

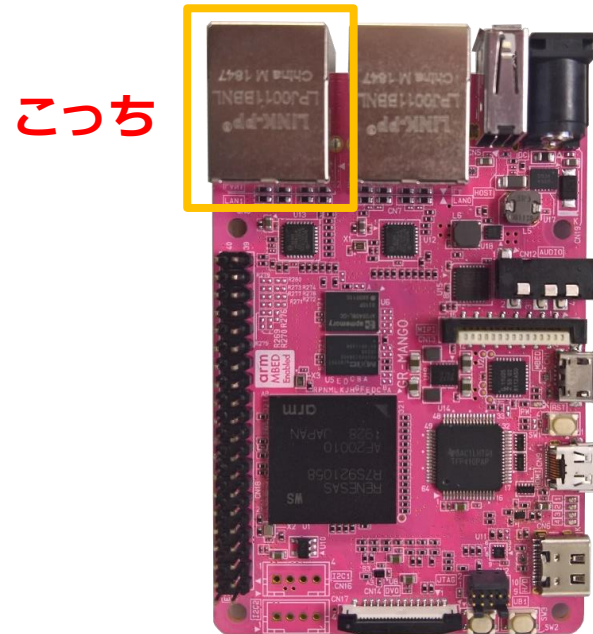
SDカードに高速アクセスするには、バッファリング無し(分割無し)に設定し、一度に大きな転送要求を行います。

```
SdUsbConnect storage("bd");           // SdUsbConnectを使えばファイルシステムへのマウントも自動実行
storage.wait_connect(STORAGE_SD);     // SDカードの挿入を待つ 引数省略でSD or USBのいずれかの挿入を待つ

FILE * fp = fopen("/bd/test.bin", "rb");
setvbuf(fp, NULL, _IONBF, 0);         // バッファリング無し(分割無し)
size_t read_size = fread(buf, sizeof(char), 4096, fp); // 一度に大きな転送要求を行う
```

Mbed OS で使用する Ethernet のコネクタ

Mbed OSで使用するEthernetのコネクタはこっち。



変更する際は以下ファイルのETHER_CH0_ENを1、ETHER_CH1_ENを0に変更してください。

`mbed-os\features\netsocket\emac-drivers\TARGET_RZ_A2_EMAC\r_ether_rza2\src\targets\TARGET_GR_MANGO\r_ether_rza2_config.h`

カメラやLCDを変更する方法

「mbed_app.json」の“camera-type”でカメラの変更、“lcd-type”でLCDの変更できます。GR-MANGOでは以下の設定が可能です。詳細は [こちら](#) を参照ください。

camera-type の値	説明
CAMERA_RASPBERRY_PI	Raspberry Pi カメラモジュール V2 (デフォルト設定)
CAMERA_RASPBERRY_PI_WIDE_ANGLE	Raspberry Pi カメラモジュール V2 広角設定
CAMERA_OV7725	GR-LYCHEEの付属カメラ

lcd-type の値	説明
RGB_TO_HDMI	GR-MANGOボード上HDMIに出力 (デフォルト設定)
TF043HV001A0	TF043HV001A0 (40-pin connector)
ATM0430D25	ATM0430D25 (40-pin connector)
FG040346DSSWBG03	FG040346DSSWBG03 (40-pin connector)

[Renesas.com](https://www.renesas.com)