RAL-KCM3MB1 User's Guide

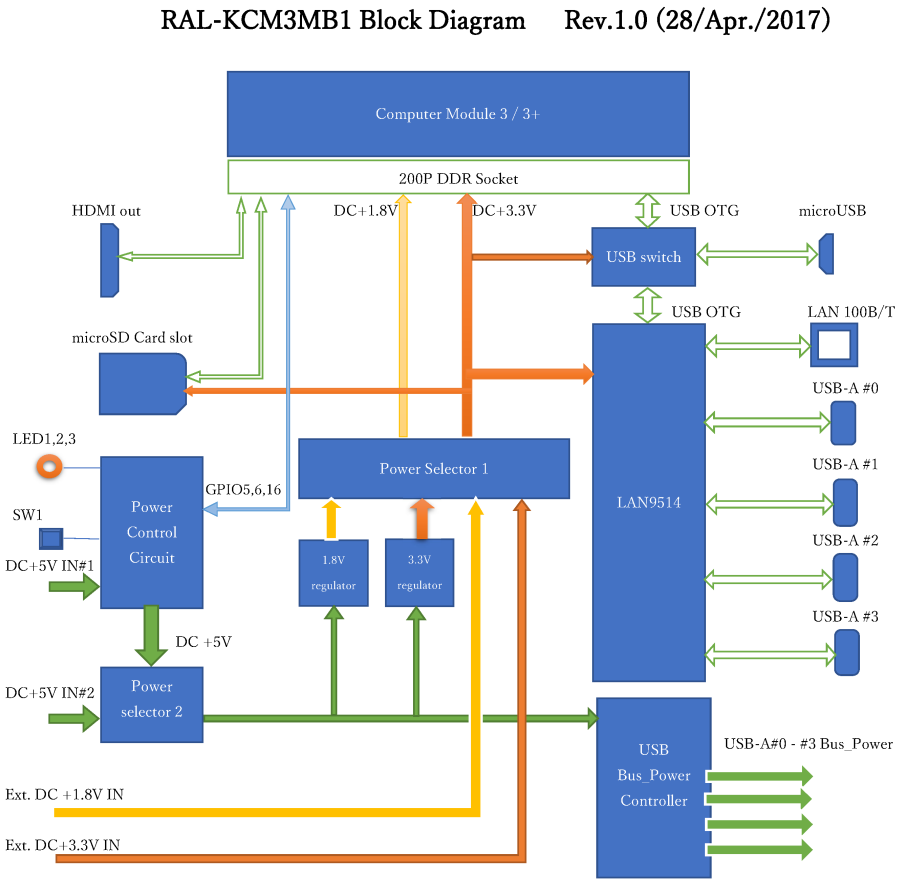
Rev. 2R1 3/Nov./2020

1.概要

RAL-KCM3MB1(以下CM3MB1と表記)はRaspberry piを組みこんだAudio機器(Network Player)を実現するために開発されたComputerModule 3/3+のためのキャリボードです。ボード上にはAudio機器として使いやすいように電源ON-OFFのためのスイッチやアクセス/シャットダウン表示LED、低ノイズの3.3V,1.8Vレギュレータ、DAC接続用のI2S専用コネクタだけではなくI2C専用コネクタやSPI専用コネクタも備えています。それらの特徴によりAudio機器に要求される低ノイズ、操作性の良さ、機器への組み込みやすさを実現しています。また、Audio機器らしい外観と機能を備えた専用エンクロージャも提供されており、Raspberry piのハードウェアに詳しくなくともGadget工作なしで「ラズパイオーディオ」を楽しむことができます。

2.構成　Block Diagram

CM3MB1の構成は下図を参照して下さい。



3.外部接続

LAN　100M/10Mbps(100Base-T/10Base-T)のLANケーブルを接続して下さい。

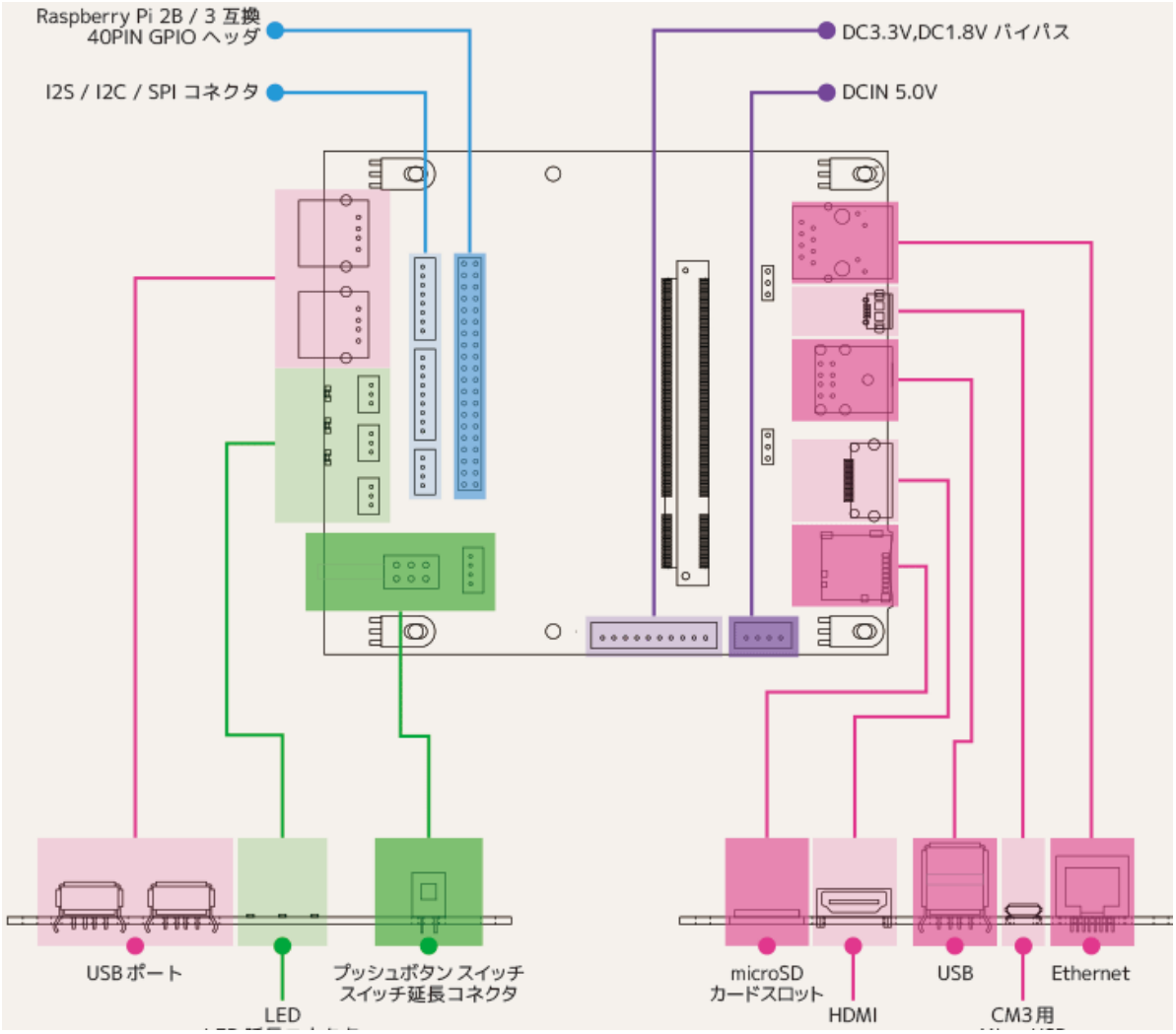
電源　DC+5V/3A(4個のUSB-AポートにBus_Power給電のUSBデバイスを接続していない場合はDC+5V/1AでもCM3は動作します。

USB-A　USB2.0(High Speed 400M/bpd, Full Speed(23Mbps), Loe-Speed(1Mbps)対応のデバイスを接続することができます。各デバイスに最大500mAのBus_Power(DC+5V)を供給可能ですが500mAを超えると過電流保護回路が動作します。

電源プッシュスイッチ　CM3MB1上のSW1を使用せずに外部にスイッチを増設することができます。

LED　CM3MB1状のLEDを使用せずに外部にLEDを増設することができます。

各コネクタの位置は下図を参照してください。



4. 基板上接続　コネクタおよび設定ポスト

4-1)　コネクタ

記号	名称	規格/種別	機能
CN1	microSD Card Socket	micro SD	CM3/3+ Light使用時に使用。eMMC搭載のCM3/3+使用時は無効。Push-Push方式。
CN2	micro B	USB 2.0	eMMC搭載のCM3/3+使用時、eMMCへの書込に使用。
CN3	HDMI	HDMI 1.4	HDMI Display 接続用出力コネクタ
CN4	LAN	100Mbps	LAN接続用コネクタ。
CN5	USB -A Host	USB 2.0	2ポート(2段スタック)。USB-DACやStorage,HID Device.
CN6	USB -A Host	USB 2.0	Wireless Dongleなどを接続するために使用。
CN7	USB -A Host	USB 2.0	Bus_Power出力は各ポートあたり500mAMax。

4-2)　設定ポスト

記号	名称	信号電圧	方向	機能
J1	CM3 Socket	3.3V	IN/OUT	DDR2 SODIMM connector for CM3/3+
J2	IO pin header	3.3V/5V	IN/OUT	Raspberry pi B 互換40Pヘッダ
J3	I2S	3.3V	OUT	PCM Audio 信号出力
J4	I2C	3.3V	IN/OUT	I2C 制御信号(Masterのみ)
J5	SPI	3.3V	IN/OUT	SPI制御信号(Masterのみ)
J6	eMMC WE	CN2 VBUS	IN	CM3/３＋上のeMMC書込み許可を設定
J7	LAN9514 GPIO	3.3V	IN/OUT	LAN9514コントローラのGPIO信号
J8	LAN LED Control	3.3V	IN/OUT	LAN アクセスLED使用/未使用を設定
J9	DC +5V In(Main Power)	5V	IN	KCM3MB1全体の主電源。
J10	Power Setting	5/3.3/1.8V	IN/OUT	CM3/3+用の3.3V,1.8Vの供給元を設定
J11	Status LED	5V	OUT	microSD/eMMCへのアクセスLED点灯信号
J12	Power LED	5V	OUT	CM3/3+および周辺回路の電源LED点灯信号
J13	Shutdown LED	5V	OUT	Shutdown Process実行中を示すLED点灯信号
J14	Ext. Power Switch	5V/3.3V	IN/OUT	KCM3MB1上の電源スイッチ信号
J15	LAN SEEPROM Enable	3.3V	IN/OUT	IEEE Global Address用ROM書込み用ヘッダ

4-3)　詳細

4-3-1) J1

名称　CM3/3+ Computer Module Socket

コネクタ形式　DDR2 SODIMM 200P

信号配列　　Raspberry Pi Compute Module 3+ のpage

www.raspberrypi.org/products/compute-module-3-plus/
からdatasheetをdownloadして参照してください。

4-3-2) J2

名称　Raspberry pi B　互換40Pヘッダ

コネクタ形式　FFC-40BMEP1B

信号配列

Raspberry pi 3 B+のpage

https://www.raspberrypi.org/documentation/usage/gpio/README.md
を参照して下さい。

4-3-3) J3

名称　　I2S Digital Audio (Linear PCM) 信号コネクタ

コネクタ形式/適合コネクタハウジング　JST　P8B-PH-K-S　8ピンポスト /　PHR-8

端子番号	信号名	レベル	方向	内容
1	Vcc 3.3	DC +3.3V	Out	KCM3MB1から供給されるIO_Drive用電源。
2	GND			
3	I2S SDIN	DC 3.3V	In	PCM date 入力端子。40Pの#38(G20)と接続されている。
4	GND			
5	I2S BCLK	DC 3.3V	Out	Bit Clock 出力。40Pの#12(G18)と接続されている。
6	I2S LRCLK	DC 3.3V	Out	LR Clock 出力。Sampling周波数を示す。40Pの#36(G19)と接続されている。
7	I2S SDOUT	DC 3.3V	Out	PCM Data 出力端子。40Pの#40(G21)と接続されている。
8	GND			

4-3-4) J4

名称　I2C　制御ライン信号コネクタ。マスタとしてのみ動作。

コネクタ形式/適合コネクタハウジング　JST P4B-PH-K-S 4ピンポスト / PHR-4

端子番号	信号名	レベル	方向	内容
1	Vcc 3.3V	DC +3.3V	Out	KCM3MB1から供給されるIO_Drive用電源。
2	SDA1		In/Out	I2C1 Data Line。40Pの#3(G2)と接続されている。 1.8KでVcc 3.3VにPull up済。
3	SCL1		Out	I2C1 Clock Line。40Pの#5(G3)と接続されている。 1.8KでVcc 3.3VにPull up済。
4	GND			

4-3-5) J5

名称　SPI制御ライン信号コネクタ。　マスタとしてのみ動作。

コネクタ形式/適合コネクタハウジング　JST　P9B-PH-K-S　9ピンポスト /　PHR-9

端子番号	信号名	レベル	方向	内容
1	Vcc 3.3	DC +3.3V	Out	KCM3MB1から供給されるIO_Drive用電源。
2	SCLK	DC +3.3V	Out	SPI Clock Line. 40Pの#23(G11)と接続されている。
3	MOSI	DC +3.3V	Out	SPI Data 出力Line。40Pの#19(G10)と接続されている。
4	MISO	DC +3.3V	In	SPI Data 入力Line。40Pの#21(G9)と接続されている。
5	CE0_N	DC +3.3V	Out	SPI Device Select Line #0。40Pの#24(G8)と接続されている。
6	CE1_N	DC +3.3V	Out	SPI Device Select Line #1。40Pの#26(G7)と接続されている。
7	N.C.			何も接続されていない。
8	N.C.			何も接続されていない。
9	GND			

4-3-6) J6

名称　CM3/CM3 + ON-BOARD eMMC Write Enable 設定ポスト

コネクタ形式/適合コネクタハウジング　HTK FFC-3AMEP1B 3ピンポスト / HTK DIC Z-208

端子番号	信号名	レベル	方向	内容
1	V_Bus_CN2	DC+5V	In	CN2(Micro USB)コネクタにUSBホスト機器接続時 DC+5Vが供給される。
2	eMMC WE		In	CM3/CM3 +のON-BOARD eMMCに対する書込許可 (Write Enable)制御入力。eMMC書込み実行時には 本コネクタの#1,#2間にショートプラグを挿入して 電源をONにする。
3	GND			CM3/CM3+、CM3 Lite/CM3 + Liteの通常使用時は 本コネクタの#2,#3間にショートプラグを挿入して おくこと。Liteモデルの場合はmicro SDカードから Bootすることができる。

4-3-7) J7

名称　LAN9514 GPIO 入出力コネクタ

コネクタ形式/適合コネクタハウジング　JST P5B-PH-K-S 5ピンポスト(未実装) / PHR-5

端子番号	信号名	レベル	方向	内容
1	LAN_GPIO3	3.3V	In/Out	LAN9514 ChipのGPIO3端子に直結。
2	LAN_GPIO4	3.3V	In/Out	LAN9514 ChipのGPIO4端子に直結。
3	LAN_GPIO5	3.3V	In/Out	LAN9514 ChipのGPIO5端子に直結。
4	LAN_GPIO6	3.3V	In/Out	LAN9514 ChipのGPIO6端子に直結。
5	LAN_GPIO7	3.3V	In/Out	LAN9514 ChipのGPIO7端子に直結。

4-3-8) J8

名称　　LAN コネクタLED ON-OFF設定ポスト

コネクタ形式/適合コネクタハウジング　HTK FFC-3AMEP1B / HTK DIC Z-208

端子番号	信号名	レベル	方向	内容
1	VCC3R3	DC +3.3V	In	CM3MB1上で生成された+3.3V電源。
2	LED Enable		In	#1との間をショートプラグで短絡するとLANコネクタの LEDが点灯、#3との間をシュートプラグで短絡すると LEDは点灯しない。
3	N.C.	Open	-	どこにも接続されておらず未使用、Open。

4-3-9) J9

名称　電源入力コネクタ。

コネクタ形式/適合コネクタハウジング　JST B4B-XH-A 4ピンポスト / XHP-4

端子番号	信号名	レベル	方向	内容
1	DC+5V_In	DC+5V +/-5%	In	CM3MB1基板の動作電力入力 4個のUSB-AコネクタのBus_Power を含め3A(1A+ 0.5Ax4).
2	DC +5V_In	DC+5V +/-5%		基板上パターンで#1に接続されている。
3	Power_GND	0V		動作電力入力に対するReturn Path。
4	Power_GND	0V		動作電力入力に対するReturn Path。 基板上パターンで#3に接続されている。

4-3-10) J10

名称　電源選択設定コネクタ(基板上の3.3Vレギュレータ、1.8Vレギュレータを設定します)

コネクタ形式/適合コネクタハウジング　JST P14B-PH-K-S 14ピンポスト / PHR-14

端子番号	信号名	レベル	方向	内容
1	GND			
2	VCC_5	DC+5V	Out	電源ON-OFF制御FETの出力。
3	GND			
4	VCC_5	DC+5V	Out	電源ON-OFF制御FETの出力。
5	VCC_5_In	DC+5V	In	基板上のVCC DC+5V電源供給ライン(3.3Vレギュレータ、1.8Vレギュレータへの入力)への入力。USB-AコネクタのV_Bus電源供給ラインには接続されていません。
6	GND			
7	VCC_5_In	DC+5V	In	基板上のVCC DC+5V電源供給ライン(3.3Vレギュレータ、1.8Vレギュレータへの入力)への入力。USB-AコネクタのV_Bus電源供給ラインには接続されていません。
8	GND			
9	VCC_3.3_Out	DC+3.3V	Out	基板上の3.3Vレギュレータ出力。
10	VCC_1.8_Out	DC+1.8V	Out	基板上の1.8Vレギュレータ出力。
11	GND			
12	VCC_1.8_In	DC+1.8V	In	CM3モジュールへの1.8V供給入力。
13	GND			
14	VCC_3.3_In	DC+3.3V	In	基板上のVCC DC3.3V電源およびCM3モジュール への3.3V供給ラインへの入力。

設定方法

(1)　電源入力コネクタJ9#1,#2から供給され、CM3MB1上の電源スイッチ(SW1)を経由した
DC+5Vを使用する場合は本コネクタの#2と#7、#4と#5をそれぞれ接続して下さい。

(2)　本基板上の電源スイッチを使用せずに外部のDC+5V電源を使用する場合は直接#5と#7に
接続して下さい。ただし、その場合はShutdownコマンドによる電源OFFは使用できません。

(3)　基板上の3.3Vレギュレータの出力を使用する場合は#9と#14を接続して下さい。
外部の3.3V電源を使用する場合は#14に直接接続して下さい。

(4) 基板上の1.8Vレギュレータの出力を使用する場合は＃10と#12を接続して下さい。
外部の1.8V電源を使用する場合は#12に直接接続して下さい。ただし3.3V電源との
投入、切断シーケンスに注意する必要があります。

(5) 標準添付のPHR-14ハウジングは#2と#7、#4と#5、#9と#14、#10と#12をそれぞれ
接続しています。

4-3-11) J11

名称　Status LED設定用コネクタ

コネクタ形式/適合コネクタハウジング　JST　P3B-PH-K-S　3ピンポスト /　PHR-3

端子番号	信号名	レベル	方向	内容
1	LED駆動電流源	DC+5V	Out	LEDを点灯させるための電流供給源 DC+5V10mA(500_Ohm)
2	LED_Anode	DC+2V	In	基板上のLED1(Status LED)を点灯させるための 入力。点灯させるためには#1と#2を接続しておく必要があります。CM3のGPIO16をLogic1にすることによりLED1が点灯します。
3	External_LED	DC+2V	In	外部のLEDを使用する場合は#1に外部LEDのAnode を#3にCathode(K)を接続して下さい。

4-3-12) J12

名称　Power LED設定用コネクタ

コネクタ形式/適合コネクタハウジング　JST　P3B-PH-K-S　3ピンポスト /　PHR-3

端子番号	信号名	レベル	方向	内容
1	LED駆動電流源	DC+5V	Out	LEDを点灯させるための電流供給源 DC+5V10mA(500_Ohm)
2	LED_Anode	DC+2V	In	基板上のLED2(Power LED)を点灯させるための 入力。点灯させるためには#1と#2を接続しておく必要があります。電源スイッチにより VCC_5 が供給されるとLED2が点灯します。
3	External_LED	DC+2V	In	外部のLEDを使用する場合は#1に外部LEDのAnode を#3にCathode(K)を接続して下さい。

4-3-13) J13

名称　Shutdown LED設定用コネクタ

コネクタ形式/適合コネクタハウジング　JST P3B-PH-K-S 3ピンポスト / PHR-3

端子番号	信号名	レベル	方向	内容
1	LED駆動電流源	DC+5V	Out	LEDを点灯させるための電流供給源 DC+5V10mA(500_Ohm)
2	LED_Anode	DC+2V	In	基板上のLED3(Shutdown LED)を点灯させるための 入力。点灯させるためには#1と#2を接続しておく必要があります。CM3のGPIO5をLogic1にすることによりLED1が点灯します。
3	External_LED	DC+2V	In	外部のLEDを使用する場合は#1に外部LEDのAnode を#3にCathode(K)を接続して下さい。

4-3-14) J14

名称　外部電源スイッチ接続用コネクタ

コネクタ形式/適合コネクタハウジング　JST P4B-PH-K-S 4ピンポスト / PHR-4

端子番号	信号名	レベル	方向	内容
1	POWER_ON		In	#1と#2の間を短絡すると電源スイッチを押す 動作と同じ働きをします。電源をON、OFF する場合に使用します。#3と#4の短絡動作 と同時に行うために外部電源スイッチは 2回路2接点のモーメンタリ スイッチを 使用して下さい。
2	GND		。	
3	VCC_3.3	DC+3.3V	Out	
4	POWER_DOWN		In	#3と#4の間を短絡すると電源スイッチを押す 動作と同じ働きをします。電源をON、OFF する場合に使用します。#1と#2の短絡動作 と同時に行うために外部電源スイッチは 2回路2接点のモーメンタリ スイッチを 使用して下さい。

4-3-15) J15

名称　LAN9514 EEPROM設定コネクタ

コネクタ形式/適合コネクタハウジング　HTK FFC-12BMEP1B 12ピンポスト / HTK DIC Z-208

LAN9514の動作設定用EEPROMに設定値を書き込むために使用するコネクタです。通常使用時は1-2、3-4、5-6、7-8、9-10、11-12間をすべてショートプラグを挿入しておいて下さい。
IEEE Global Addressの上位3バイトは0x00C0D0(RATOC Systems)に設定されています。

5. LED

5-1) LEDの制御

CM3MB1上には動作状況表示用にPower LED(LED2), Status LED(LED1), Shutdown LED(LED3)の3個のLEDとそれぞれに対応した外部LED用のコネクタ(J11,J12,J13)が用意されています。
これらのLEDのON-OFF(点灯、消灯)は下記のような条件で行われます。

(1)Power LED　CM2MB1にJ9を通してDC+5Vが供給され、電源スイッチ(SW1)により基板上にDC+5Vが供給されると点灯します。
電源スイッチが再び押された場合やShutdownによるPower OFFコマンドの実行により基板上にDC+5Vが供給されなくなると消灯します。

(2)Status LED　OSやSoftwareがmicroSDもしくはCM3上のeMMCにアクセス実行中に点灯します。点灯の制御はGPIO16によって行われます。

(3)Shutdown LED　電源スイッチが押されOSがGPIO6を通してそれを検出し受け付けたことを示すために点灯し、Shutdownプロセスが終了後15秒間待機したのち消灯します。

5-2) LED制御コマンド　/boot/config.txt への追加

Status LED、Shutdown LEDおよび電源コントロール回路の制御を行うためには下記の2行を**/boot/config.txt**に追加しておく必要があります。OSなどの自動updateによりこれらの追加行が消されてしまわないようにするためには
/boot/userconfig.txtというfileを作成しこれらの行を記述しておきます。
/boot/config.txtの最後に**include userconfig.txt**という1行を追加しておいて下さい。

```
/boot/userconfig.txtに下記の2行もしくは3行を追加。

• dtoverlay=pi3-act-led,gpio=16
• dtoverlay=gpio-poweroff,gpiopin=5

もし下記の行が記述されていなかった場合には追加。
• include userconfig.txt
```

Volumio 2.671(26-Nov.\2019) 以降のversionの**config.txt**には既にこの行が含まれていますので追加する必要はありません。 install時に**/boot/config.txt**の内容をText工ディタで確認しておいてください。

6. 電源OFFリクエストの検出および外部電源スイッチの増設について

電源ONで動作中に電源スイッチを押すとOSに対してShutdown とPoweroffコマンドの実行をリクエストします。リクエストが受け付けられなかった場合は一定期間待機した後、電源OFF操作をハードウェアで実行します。リクエストの検出はGPIO6により行われ、Poweroff指令はGPIO5により行われます。PoweroffコマンドへのGPIO5の登録は前章5.の **config.txt**への記述追加で実施することができますが、GPIO6による電源スイッチONの検出とShutdownコマンドリクエストはPythonで作成した簡単なProgramを**/etc/rc.local**にScriptを記述しておくことによって常駐、実行させます。

Note)　外部電源スイッチの増設について

CM3MB1を専用筐体(NWT01_en)以外の筐体に組み込み、基板上のSW1を使用せずに新たに電源スイッチを増設する場合は2回路2接点(2極単投もしくは双投)のモーメンタリスイッチを使用して下さい。LEDを内蔵した照光式スイッチを使用する場合の結線例は下表をご覧下さい。照光のための内蔵LEDはJ12の#1.#3に接続しPower LEDとして使用します。

参考用に使用したスイッチは NKK Switches LP01-25CH1CKNS1KM 2極双投モーメンタリ
LED発行色は緑色です。

LP01 Switch terminal name	CM3MB1 Connector and pin number
LED(+)	J12: #1
	J12: #2
LED(-)	J12: #3
Normally Open #1 (N.O.1)	J14: #1
Common #1 (COM.1)	J14: #2
Normally Open #2 (N.O.2)	J14: #3
Common #2 (COM.2)	J14: #4

注)　スイッチのN.C.1およびN.C.2、J12:#2には何も接続せずOPENとしておいて下さい。

7.CM3 Lite / CM3+ Lite boot用microSDカードの作成

モジュール上にeMMCを搭載していないCM3 Light / CM3+ Lightを使用する場合はmicroSDカードに**Raspberry pi OS**や**Volumio**、**MooDeAudio**などの**.img**ファイルをあらかじめ書き込んでおきCN1のmicroSDカードスロットに入れ電源をONにします。microSDカードへの書込みはRaspberry pi 3用のmicroSDカードの作成手順と同様にWindows10 PCやMacOS上で行うことができます。CM3MB1で使用するためには上記5,6のLEDと電源スイッチのために**config.txt**や**rc.local**などへの追加が必要です。

Step.1　microSDカードのFormat

Windows PCやMacでSD Associationのサイト(www.sdcard.org)からSD Formatterをdownload、installした後、displayの表示にしたがってmicroSDカードをフォーマットして下さい。詳しい使用方法は同サイトを参照してください。

Step.2 *.imgファイルのdownloadと解凍、作成

Raspbian OSやVolumioなどのOSイメージが圧縮されたファイルをそれらのサイトからdownloadした後解凍して***.img**ファイルを作成して下さい。

Step.3 microSDカードへの書込み

Win32Diskimager(Windows PC用,
SORCEFORGE www.sourceforge.net/projects/win32diskimager/からdownload)
もしくは
Etcher(Windows10 あるいはMacOS, www.balena.io/etcherからdownload)
を使用して*.imgをmicroSDカードに書き込んで下さい。

Step.4 config.txtの確認と記述追加

書込み終了後、explorer(Windows)もしくはfinder(MacOS)でmicroSDカードをチェックしてみて下さい。/bootフォルダ(explorerではmicroSDカードの名称がbootと表示されます)の**config.txt**をテキストエディタで読み出してみて下さい。
最終行のあたりに**include userconfig.txt**という記述が見当たらない場合は追加しsave(上書き)しておいて下さい。

Step.5 userconfig.txtの作成、sshの作成と/bootフォルダへの追加

同様にテキストエディタを使用してuserconfig.txtを作成し、bootフォルダに書き込んでおいて下さい。 **userconfig.txt**の内容はStatus LED、Shutdown LEDを有効にするための下記の2行です。

```
dtoverlay=pi3-act-led,gpio=16
dtoverlay=gpio-poweroff,gpiopin=5
```

Raspberry pi OSにcommandモードでアクセスするためには**ssh**というファイルをbootフォルダに入れておく必要があります。内容は空でも構いませんが、install記録のために作成日などを入れておくと後で便利です。

```
sshファイルの内容の例
  ssh sample for RAL-CM3MB1 01/Apr./2017
```

Step.6　CM3MB1にmicroSDカードをSetする。

　userconfig.txtと**ssh**の書込み終了後、microSDカードをPCあるいはMacから取り出し、CM1MB1のmicroSDカードスロットに挿入してください。また、J6の#2,#3にショートプラグがセットされていることを確認しておいて下さい。
当然ですがCM3 Lite/CM3 + LiteはJ1のDDRソケットに確実にセットしておいて下さい。

Step.7　その他の準備

　最初の設定作業を行うためにはHDMIコネクタにディスプレイ、USBポートにUSBキーボードを接続しておいたほうがスムーズに行えます。LANの接続と電源アダプタも忘れずに接続しておいて下さい。

Step.8　CM3MB1の電源をONにする。

　以上の準備が完了したことを確認した後CM3MB1の電源をONにして下さい。初回はOSがFile Systemのセットアップを行うため、HDMIディスプレイにuser,pwを入力を求めるprompt messageが表示されるまで数分かかる場合もあります。
/boot/config.txt, /boot/userconfig.txtの記述(上記(4),(5)参照)に間違いがなければ3個並んだLEDの中央Status LEDが何度も点滅を繰り返しているはずです。
しばらくすると**user**、**password**の入力を求められますので**Volumio**をinstallした場合はどちらも**volumio**, **volumio**と入力してください。**Volumio**が起動します。

Step.9　sambaの設定

　Windows10PCからExplorerを使用してfileにアクセスできるように**samba**の設定を行っておきましょう。下記のように組み込まれているtext editor **nano**を使用します。

```
$ sudo nano /etc/samba/smb.conf
 最後部に下記を追加してCntl+oで上書きし、Cntl+xで終了します。

 [volumio]
          comment = RAL-NWT01Plus volumio work folder
          path = /home/volumio
          read only = no
          guest ok = yes
          force user = volumio

    nanoを終了後
 $ sudo service smbd restart
 でsmbdを再起動させます。
```

　smbdを再起動させた後、同じLAN内にあるWindows10PCでExplorerを実行してみて下さい。
フォルダの指定はIPアドレス"192.168.xx.xx"を入力して下さい。
IPアドレス(192.168.xx.xx)がわからない場合は

```
$ ifconfig
  （注）Explorerのアドレス欄には￥￥に続けて表示されたIPアドレスを入力してください。
```

と入力すると表示されます。CM3MB1にVolumioをinstallした場合はCM3MB1のフォルダが表示され続いてhomeというフォルダをclickすると先ほど作成したvolumioというフォルダが表示されます。そのフォルダ(volumio)にWinowsPCの作業用フォルダに入れて置いたshtdwn_button_01.pyというPythonプログラムをcopyして下さい。

　ちなみにUSBやNASというフォルダも表示されます。もしCM3MB1のUSBポートにUSBメモリスティックやUSB HDD,SSD、USB Memory Card Readerなどが接続されていればそれらに格納されているフォルダやファイルが表示され、PCのUSBポートに接続している場合と同様にRead/Write/Copyを行うことができます。

　shtdwn_button_01.pyというPythonプログラムは電源pushボタンが押されたことを検出し、**shutdown -h now**を実行し10秒後に電源をOFFにします。このプログラムが**Volumio(Raspbian)**起動後に自動的にロードされるように設定を行っておく必要があります。
　そのための手順は

```
 $ sudo nano /etc/rc.local

 最終行のあたりにあるexit 0という行の前に下記を追加します。
 sudo python /home/volumio/shtdwn_button_01.py &
 追加後Ctrl-oで上書き、Ctrl-xでnanoを終了してください。**
```

Step.10　初期設定の完了。

以上でCM3 Lite/CM3 + Liteを使用する場合のmicroSDカードの作成と初期設定は終了です。
Volumioをinstallした場合は再起動したのち、同一LAN内のPCやTablet、Smartphoneの
ブラウザから **Volumio**の初期設定を行ってください。

8.CM3 / CM3+ eMMCへの書込みについて

モジュール上にeMMCを搭載したCM3 /CM3+ を使用する場合はそのeMMCメモリにOSやVolumioのSystem fileなどを書き込む必要があります(microSDカードスロットが無効になります)。
eMMCへの書込み手順は下記のStep.を参照してください。

Step.1　基本となるimage file(*.img)のdownloadおよび解凍やCM3 +上のeMMCへの書き込みを実行するためにWindow10PCを用意してください。microSDカードに書き込む場合と同様、WindowsPCに **SDFormatter** および **Win32DiskImager** のinstallも必要です。さらにCM3+上のeMMCにimageファイルを書き込むためにはWindws10にeMMCをmicroSDカードとして認識させるための**rpiboot.exe**という**DeviceDriver**が必要です。

詳しくは,
<raspberrypi.org/documentation/hardware/computemodule/cm-emmc-flashing.md>
の'Flashing the Compute Module eMMC'という記事を参照して下さい。その記事中のlinkをクリックして**RPiBoot.exe**をdownload,installしておいて下さい。

Step.2　CM3MB1のJ1(DDR socket)の左右の金属レバーを拡げてeMMC付きのCM3 +(CM3 4GBも同様です)を装着して下さい。左右のレバーが両方ともきちんとくぼみに収まり、CM3+もきちんとソケットに嵌っていることを確認して下さい。その後CM3MB1上のJ6(USB Boot)の1-2間に短絡プラグを差し込みeMMC Write Enableに設定 して下さい。

Step.3　CM3MB1のmicroUSBコネクタ(CN2)とWindows10_PCのUSB-Aポートを接続しCM3MB1の電源をONにして下さい。

Step.4　Windows10_PC上で**rpiboot.exe**を実行してください。command promptのWindowが表示され、CM3＋の接続待ちとなります。数秒後にCM3+上のeMMCがUSB Card Reader経由のMediaとして認識された後ドライブ名がアサインされます。explorerでドライブ名を確認しておいて下さい。

Step.5　アサインされたドライブに対して**SDFormatter**を実行し、microSDに見せかけたeMMCをフォーマットして下さい。

Step.6　Formatが終了したら**Win32DiskImager**を実行、書き込み先としてStep.4で確認したドライブ名を指定し、あらかじめdownload、解凍しておいたVolumioのimageをeMMCに書き込んで下さい。

Step.7　書込みが正常終了したらexplorerでアサインされたドライブをクリックしてみて下さい。ドライブにはbootというラベル(Volume名、Raspberry piでは/bootという名称のFAT32のフォルダ) が付けられておりその下のファイルが表示されます。そこにあらかじめPC上に作成しておいた(RAL HPからdownloadすることもできます) **ssh** (もしくは**SSH**)という名前の空の拡張子がないファイル(何かのtextがはいっていてもよい)をcopyしておきます。この**ssh****というファイルが**/boot**に存在しないと同一Networtk上のPCやMacからSSHを使用してloginできませんのでCM3+の初期設定などが難しくなります。**ssh**という名称のファイルはPC上ではテキストエディタ('メモ帳'など)を使って作成できますが、必ず'名前の変更'を選択して拡張子を削除しておいて下さい。

Step.8 同様にPC上でテキストエディタを使用してあらかじめPC上に**userconfig.txt**という名称のtextファイルを作成しておき、そのファイルを/bootフォルダにcopyして下さい。
userconfig.txtの内容は下記ですが、一つの文中にスペースは入れないで下さい。スペースがあるとセパレータとして認識され記述した内容が正しく実行されません。またタイプミスがあってもbootの実行は止まらずそれらを無視して続行されますので正確に記述して下さい。

```
dtoverlay=pi3-act-led,gpio=16
dtoverlay=gpio-poweroff,gpiopin=5
```

Windows10上のTextエディタで**boot/config.txt**を読み出し、最終行に**include userconfig.txt**という記述があることを確認して下さい。**Volumio 2.632**以降の**config.txt**にはこの記述が追加されていますので最新版をdownloadすれば追加する必要はありません。もしその記述がなければ最終行にTextエディタを使って**include userconfig.txt**という一行を追加しておいて下さい。

Step.9 以上の作業が終了したらCM3MB1の電源を切り、PCとの接続ケーブル(microB to USB-A)を外し、J6(USB Boot)の2-3間に短絡プラグを戻してDISableに設定 して下さい。

Step.10 LANケーブル、必要に応じてHDMIケーブル、キーボードなどを接続した後、電源をONにして下さい。Power_LEDが点灯しその横のAct_LEDが点滅して**Volumio(Raspbian)**がLoadされればStep.9までの操作が正しく実行されたことがわかります。最初はファイルシステムの構築などが行われるため数分かかる場合もありますが正常に**Volumio**が起動することを確かめて下さい。初回はHDMIディスプレイ(HD-TVでOK)とUSBキーボードを接続しておくことを推奨します。起動時いろんなMessageが次々と表示され最終的に画面の左上隅にUserとPasswordの入力prompt messageが表示されます。CM3MB1に接続されたHDMIディスプレイとキーボードでStep.11以降の作業を行う場合はUser,PWの初期値としてそれぞれ**volumio**, **volumio**を入力して下さい。
User,PWが認証され **Volumio***が起動したら**'Volumio'**というLogoマークが表示されます。**ifconfig** Enterと入力すればCM3MB1のIPアドレスが表示されます。

Step.11 ここまでのStepでCM3 +上のeMMCメモリにはVolumio2が正しくinstallされました。
続いて、CM3MB1のShutdownボタンを有効にするための設定や、時刻(Time Zone)の設定を行います。CM3MB1のShutdownボタンはCM3MB1をHeadless(CM3MB1にキーボード、マウス、HDMIディスプレイを接続しないで)で運用している場合電源をOFFにしたい場合にとても便利です。

Step.12 続いてWindows10PCからSSH経由でloginしたままの状態で時刻(Time Zone)の設定をしておきましょう。**Volumio**をinstallした後のTime ZoneはUTC(協定世界時)という国際度量衡局が原子時計を基準にして決めている時刻で日本標準時間とは9時間の時差があります(UTC+09:00)。このままでは同一Network上のNASやPC、Tabletなどと時刻が一致しませんのでlog listなどで不都合が出ることがあります。日本標準時間に合わせるためには下記のように行います。 Window10PCやMacから

```
 $ timedatectl
    現在の設定が表示されます。既にJST Time ZonがAsia/Tokyoに設定されている場合は年月日、時刻を確認
    しておいて下さい。
 $ sudo dpkg-reconfigure tzdata
    raspi-config実行時の「Time Zone 設定 Window」が表示されますのでカーソル移動キー(↓、↑キー)を操作
    してAsia/Tokyoを選択してReturnキーを押してください。

   再度
 $ timedatectl
   コマンドで日本標準時間(JST)の時刻、Time Zoneが「Asia/Tokyo」に設定されていることを確認して下さい。
   この設定は電源をOFFにしても保存されますので電源ONの都度設定する必要はありません。
```

Step.13　Windows10PCからExplorerを使用してfileにアクセスできるように**samba**の設定を行っておきましょう。

```
$ sudo nano /etc/samba/smb.conf
最後部に下記を追加してCntl+oで上書きし、Cntl+xで終了します。

[volumio]
   comment = RAL-NWT01Plus volumio work folder
   path = /home/volumio
   read only = no
   guest ok = yes
   force user = volumio

nanoを終了後
$ sudo service smbd restart
でsmbdを再起動させます。
```

smbdを再起動させた後、Windows10PCでExplorerを実行してみて下さい。フォルダの指定はIPアドレス"192.168.xx.xx"を入力して下さい。IPアドレス(192.168.xx.xx)がわからない場合は

```
$ ifconfig
 （注）Explorerのアドレス欄には￥￥に続けて表示されたIPアドレスを入力してください。
```

と入力すると表示されます。CM3MB1(Volumio)のフォルダが表示されhomeというフォルダをclickすると先ほど作成したvolumioというフォルダが表示されます。そのフォルダ(volumio)にWinowsPCの作業用フォルダに入れて置いた**shtdwn_button_01.py**というPythonプログラムをcopyして下さい。ちなみにUSBやNASというフォルダも表示されます。もしCM3MB1のUSBポートにUSBメモリスティックやUSB HDD,SSD、USB Memory Card Readerなどが接続されていればそれらに格納されているフォルダやファイルが表示され、PCのUSBポートに接続している場合と同様にRead/Write/Copyを行うことができます。

Step.14　**shtdwn_button_01.py**というPythonプログラムは電源pushボタンが押されたことを検出し、**shutdown -h now**を実行し15秒後に電源をOFFにします。このプログラムが**Volumio(Raspbian)**起動後に自動的にロードされるように設定を行っておく必要があります。そのための手順は下記を参照して下さい。

```
$ sudo nano /etc/rc.local

最終行のあたりにあるexit 0という行の前に下記を追加します。
sudo python /home/volumio/shtdwn_button_01.py &
追加後Ctrl-oで上書き、Ctrl-xでnanoを終了してください。
```

Step.15　ここまでの手順でCM3MB1に必要な設定は終了です。Volumioをシャットダウンし、再起動すれば設定した機能が動作するようになります。

9.外形寸法

110(W) x 124(D)　x 16.2(USB Type-A 2段スタックコネクタの基板上面からの高さ)