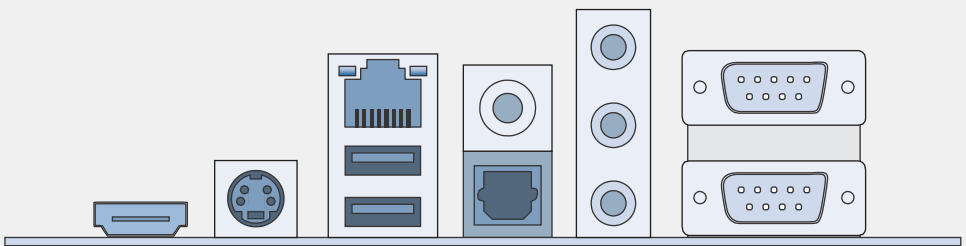
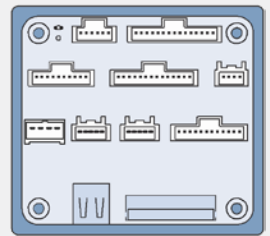
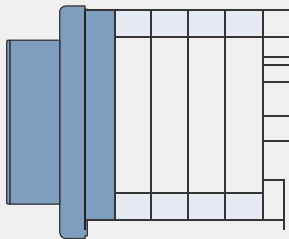
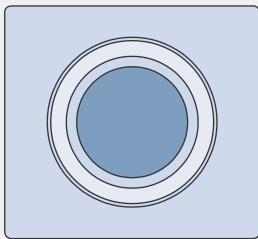


# VRmagic

## D3インテリジェントカメラ

### ユーザーガイド



株式会社 アド・サイエンス

〒273-0005 千葉県船橋市本町2-2-7 (船橋本町プラザビル)

TEL:047-434-2090 FAX:047-434-2097

<http://www.ads-img.co.jp/>

Issued April 2014



VRmagic

VRmagic GmbH

Augustaanlage 32  
68165 Mannheim  
Germany

Phone +49 (0)621 400 416 -20

Fax +49 (0)621 400 416 -99

info.imaging@vrmagic.com

www.vrmagic-imaging.com

## D3インテリジェントカメラ – ユーザーガイド

ドキュメントバージョン: 2.2

発行日: 2014年4月4日

対象製品: すべてのD3インテリジェントカメラ、SDKバージョン 4.2

仕様は予告なく変更する場合があります。誤記は不問とします。

このドキュメントは著作権によって保護されています。無断複写転載を禁じます。このドキュメントのどの部分も、VRmagicから承諾書を得ることなく、電子的、機械的な手段の如何にかかわらず、いかなる形でも何らかの目的で複写、転用することを禁じます。

Windows®はMicrosoft®の登録商標です。

DaVinci™はTexas Instrumentsの登録商標です。

# 目次

|          |                                 |           |
|----------|---------------------------------|-----------|
| <b>1</b> | <b>一般情報</b> .....               | <b>7</b>  |
| <b>2</b> | <b>安全</b> .....                 | <b>8</b>  |
| <b>3</b> | <b>概要</b> .....                 | <b>9</b>  |
| 3.1      | D3インテリジェントカメラのアーキテクチャ.....      | 9         |
| 3.2      | カメラモデル.....                     | 10        |
| 3.3      | インターフェースボード.....                | 13        |
| 3.3.1    | VRmCUEO3インターフェースボード.....        | 14        |
| 3.3.2    | VRmEIO3インターフェース評価用ボード.....      | 15        |
| 3.4      | 付属品.....                        | 19        |
| 3.4.1    | VRmCUEO3付属品キット.....             | 19        |
| 3.4.2    | VRmEIO3付属品キット.....              | 20        |
| <b>4</b> | <b>ソフトウェアのインストール</b> .....      | <b>21</b> |
| 4.1      | 概要.....                         | 21        |
| 4.1.1    | D3の開発環境.....                    | 21        |
| 4.1.2    | VRmagicソフトウェアパッケージ.....         | 21        |
| 4.2      | LinuxホストPCのセットアップ.....          | 22        |
| 4.2.1    | Ubuntu Linuxのインストール.....        | 22        |
| 4.2.2    | VRmagicソフトウェアパッケージのインストール.....  | 23        |
| 4.2.3    | VRmagicソフトウェアパッケージのアップデート.....  | 25        |
| 4.3      | クロスコンパイル環境の手動設定.....            | 25        |
| 4.3.1    | armhf Toolchainの手動設定.....       | 25        |
| 4.3.2    | 追加のarmhfライブラリのインストール.....       | 26        |
| 4.3.3    | TI EZSDKのダウンロード.....            | 27        |
| 4.3.4    | D3 EZSDKとTI EZSDKの手動インストール..... | 28        |
| 4.3.5    | VRmagic D3 EZSDKのアップデート.....    | 29        |
| 4.3.6    | Linuxカーネルソースのインストールとアップデート..... | 29        |
| 4.3.7    | PCにインストールされるコンポーネント.....        | 30        |

|          |  |           |
|----------|--|-----------|
| <b>5</b> | <b>ハードウェアのインストール.....</b>                | <b>31</b> |
| 5.1      | 外部センサーボードの接続.....                        | 31        |
| 5.2      | VRmCUEO3ボードの接続.....                      | 33        |
| 5.2.1    | イーサネット.....                              | 33        |
| 5.2.2    | GPIO.....                                | 34        |
| 5.2.3    | MicroSDカードスロット.....                      | 34        |
| 5.2.4    | シリアルコンソール、RS232、S-Video、UART1、CANバス..... | 34        |
| 5.2.5    | トリガーとストロボ.....                           | 35        |
| 5.2.6    | USB.....                                 | 35        |
| 5.2.7    | SPI、LED、ウォッチドッグ、電源/リセットボタン.....          | 36        |
| 5.2.8    | HDMI.....                                | 36        |
| 5.2.9    | 外部バッテリー.....                             | 36        |
| 5.2.10   | 電源.....                                  | 37        |
| 5.3      | VRmEIO3ボードの接続.....                       | 37        |
| 5.3.1    | 標準コネクタの接続.....                           | 37        |
| 5.3.2    | ピンヘッダーとSATAの接続.....                      | 39        |
| 5.3.3    | 電源の接続.....                               | 41        |
| <b>6</b> | <b>第1ステップ.....</b>                       | <b>42</b> |
| 6.1      | イーサネット接続のセットアップ.....                     | 42        |
| 6.1.1    | 2点間接続のセットアップ.....                        | 42        |
| 6.1.2    | イーサネット接続のチェック.....                       | 43        |
| 6.2      | ターミナルでのデバイスのアクセス.....                    | 44        |
| 6.2.1    | SSHを介したアクセス.....                         | 44        |
| 6.2.2    | シリアル通信(RS232)を介したアクセス.....               | 45        |
| 6.3      | カメラの初期設定パスワードの変更.....                    | 46        |
| 6.4      | カメラのIPアドレスの変更.....                       | 47        |
| 6.5      | ファイルとデータの交換.....                         | 48        |
| 6.5.1    | NFSまたはSamba/Windows Shares.....          | 48        |
| 6.5.2    | scpコマンド.....                             | 49        |
| 6.5.3    | MicroSDカード.....                          | 50        |
| 6.5.4    | USBフラッシュドライブ.....                        | 50        |
| 6.6      | CamserverとCamLab.....                    | 51        |
| 6.7      | カメラのGPIOへのアクセス.....                      | 52        |
| 6.8      | D3への追加ソフトウェアのインストール.....                 | 54        |
| 6.8.1    | VRmagicリポジトリの有効化.....                    | 54        |
| 6.8.2    | サードパーティのソフトウェア.....                      | 55        |

|          |                                      |           |
|----------|--------------------------------------|-----------|
| 6.9      | トリガーとストロボの使用法.....                   | 55        |
| 6.9.1    | トリガー入力.....                          | 56        |
| 6.9.2    | ストロボ出力.....                          | 57        |
| 6.10     | カメラのOn/Offの切り替え.....                 | 58        |
| <b>7</b> | <b>アプリケーションの開発.....</b>              | <b>59</b> |
| 7.1      | デモアプリケーションの解凍.....                   | 59        |
| 7.1.1    | ARMデモとC++ APIラッパー.....               | 60        |
| 7.1.2    | DSPデモ.....                           | 61        |
| 7.2      | ARMデモのコンパイルと実行.....                  | 61        |
| 7.3      | DSPデモのコンパイルと実行.....                  | 63        |
| 7.4      | .NETデモのコンパイルと実行.....                 | 65        |
| 7.4.1    | .NETサポートパッケージのインストール.....            | 65        |
| 7.4.2    | C#デモのコンパイルと実行.....                   | 65        |
| 7.5      | HALCON Embedded.....                 | 67        |
| 7.5.1    | HALCONライセンスの取得.....                  | 67        |
| 7.5.2    | HALCON HDevelopの取得.....              | 68        |
| 7.5.3    | D3上でのHALCONサポートのセットアップ.....          | 68        |
| 7.5.4    | D3でのHALCONインストールのテスト.....            | 69        |
| 7.5.5    | HALCON出力の視覚化.....                    | 69        |
| 7.5.6    | VRmagicによるHALCONサポートプロシージャの使用.....   | 71        |
| <b>8</b> | <b>カメラコンフィギュレーション.....</b>           | <b>73</b> |
| 8.1      | ビデオ出力の設定.....                        | 73        |
| 8.2      | USBコントローラー(USBホストまたはUSBデバイス)の設定..... | 75        |
| 8.2.1    | D3用のカスタムカーネルの作成.....                 | 76        |
| 8.2.2    | ホスト/ホストとしてのコンフィギュレーション.....          | 77        |
| 8.2.3    | ホスト/デバイスとしてのコンフィギュレーション.....         | 77        |
| 8.2.4    | デバイスのみとしてのコンフィギュレーション.....           | 79        |
| 8.3      | WiFi接続のセットアップ.....                   | 80        |
| 8.3.1    | WiFiサポート.....                        | 80        |
| 8.3.2    | WiFiインターフェースの開始と停止.....              | 80        |
| 8.3.3    | ネットワークコンフィギュレーションの変更.....            | 81        |
| 8.3.4    | ワイヤレスネットワークへの接続.....                 | 82        |
| 8.3.5    | WiFiアクセスポイントのスキャン.....               | 82        |
| 8.3.6    | エキスパートWiFiコンフィギュレーション.....           | 82        |

|           |                                  |            |
|-----------|----------------------------------|------------|
| 8.4       | 電源管理.....                        | 83         |
| 8.4.1     | CPU周波数とコア電圧の設定.....              | 83         |
| 8.4.2     | デバイスの電源切断.....                   | 84         |
| 8.5       | リアルタイムクロックの設定.....               | 86         |
| <b>9</b>  | <b>さらなるデバイスの接続.....</b>          | <b>87</b>  |
| 9.1       | SPIデバイス.....                     | 87         |
| 9.2       | I2Cデバイス.....                     | 88         |
| 9.3       | CANデバイス .....                    | 89         |
| <b>10</b> | <b>ファームウェアの更新と回復.....</b>        | <b>90</b>  |
| 10.1      | 一般的説明.....                       | 90         |
| 10.2      | カメラファームウェアのダウンロード.....           | 91         |
| 10.3      | イーサネットによるファームウェアの更新.....         | 91         |
| 10.4      | SDカードによるファームウェアの更新.....          | 94         |
| 10.4.1    | 更新用SDカードの作成.....                 | 94         |
| 10.4.2    | 更新プロセスのカスタマイズ.....               | 95         |
| 10.4.3    | 手動更新.....                        | 97         |
| 10.4.4    | 自動更新.....                        | 98         |
| 10.5      | SDカードによるデバイスの回復.....             | 98         |
| <b>11</b> | <b>補足.....</b>                   | <b>100</b> |
| 11.1      | 電気特性.....                        | 100        |
| 11.2      | VRmCUEO3インターフェースボードのピン配列.....    | 101        |
| 11.3      | VRmCUEO3インターフェースボードのケーブルプラン..... | 108        |
| 11.4      | VRmEIO3インターフェースボードのピン配列.....     | 113        |
| 11.5      | Linux簡易リファレンス.....               | 122        |
| 11.6      | Linuxデバイス.....                   | 123        |
| <b>12</b> | <b>索引.....</b>                   | <b>124</b> |

# 1 一般情報

このガイドはVRmagicのD3インテリジェントカメラプラットフォームに基づいたすべてのインテリジェントカメラに適用されるものです。このガイドを章ごとにたどってデバイスをセットアップして理解してください。このドキュメントのセクションが、特定のカメラモデルにのみ適用される場合は、各セクションの始めに示します。

## 使用されている記号

このガイドでは以下の記号や決まりを使用します:



### 警告!

避けられない場合は、多少の人身事故および/または物的損害やデバイスの損傷をもたらす可能性のある状況を示します。



### 注意

避けられない場合は、デバイスのちょっとした損傷、デバイスの不具合、データの損失をもたらす可能性のある状況を示します。



### 備考

デバイスに関連する特別な問題についての情報やデバイスの操作がより簡単になるような情報を提供します。

このような斜体の見出しは手順の始まりを示します:

1. この数は、従う手順の第一ステップを示します。以後のステップは順次、番号付けされます。
  - ▶ この矢印は、作業の予想される結果を示します。
- この記号は、このマニュアルの別の箇所や外部のドキュメントへの参照を示します。

## 2 安全

### バッテリー

VRmEIO3インターフェースボードは非充電式のCR2032リチウムコインセルを備えています。バッテリーは子供の手の届かない場所に置いてください。誤って飲み込むと最小2時間で大げがか死に至る場合があります。

バッテリーを交換する場合は同じタイプのバッテリー(CR2032)のみを使用してください。極性には常に気をつけてください。へこみや傷など外部に損傷のあるバッテリーは使用しないでください。

バッテリーは適用される国内法令と指針に従って処分されなければなりません。

### ESD防止

静電放電は、カメラに深刻で回復できない損傷をもたらす場合があります。ボードレベルカメラを扱う場合は、必ず国際規格(IEC 61340 -5 -1 (2007- 08))に従うESD防護領域で作業をしてください。

### カメラのハードウェア

このマニュアルに明示的に示しているもの以外は、カメラのどんな部分も分解しないでください。分解すると、損傷する場合があります、またどのような場合であっても製品の保証は無効になります。



### 3 概要

#### 3.1 D3インテリジェントカメラのアーキテクチャ

VRmagicのD3インテリジェントカメラプラットフォームは組込み型システムとして機能します。これは1ギガヘルツのARM Cortex-A8プロセッサと700MHzのデジタルシグナルプロセッサ(DSP)を擁する米国テキサスインスツルメンツ社の高集積DaVinci™デュアルコアCPUを備えています。両方のプロセッサは浮動小数点ユニット(FPU)を含んでいます。

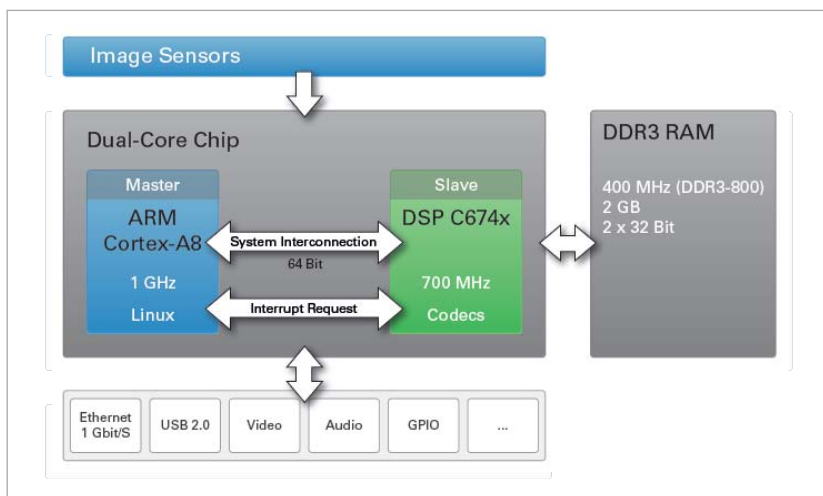


図1: インテリジェントカメラのアーキテクチャ(D3プラットフォーム)

ARMプロセッサはLinuxオペレーティングシステムを扱いますが、DSPは画像処理タスクに完全に利用可能です。ARMとDSPの間の通信はTIコーデックエンジンによってコントロールされます。

D3プラットフォームはカスタマイズされたLinux 2.6.37カーネルで動作します。ルートファイルシステムはUbuntu 12.04LTSのもので、Cortex-A8(ARMv7、Thumb2)とそのFPU(Neon)の命令セットにネイティブに対応しています。

カメラのDDR3メモリはいくつかの領域に分割されています: Linuxメモリ、共有メモリ(CMEM)、DSPメモリ、グラフィックス出力などのサブシステムのための専用領域です。イメージセンサーからの画像は常に共有メモリに書き込まれて、ARMとDSPの両方からアクセスできます。

## 3.2 カメラモデル

### シングルセンサーカメラVRmD3(F)C-X

シングルセンサーカメラは1つのイメージセンサーを備えています。センサーボードはカメラのベースユニットに直接取り付けられて、1つのコンパクトユニットを形成します。シングルセンサーカメラには、ボードレベルカメラ(OEMバージョン)、光学系付きボードレベルカメラ(COBバージョン)があります。

すべてのカメラはセンサーボード、2枚のCPUボード、インターフェースボードを備えています。イメージセンサーによっては、カメラは追加のFPGAボードを備える場合があります(VRmD3FC-Xモデル)。

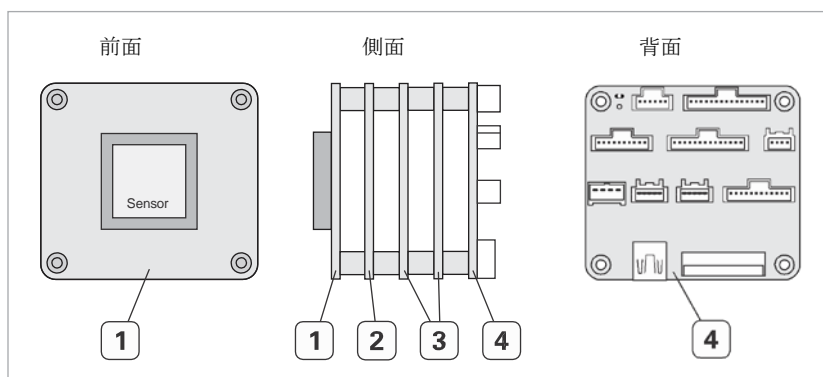


図2: CUEO3インターフェースボードとFPGAボードを備えたインテリジェントシングルセンサーカメラ(OEM)

- 1 イメージセンサーを備えたセンサーボード
- 2 FPGAボード(VRmD3FC-Xカメラのみ)
- 3 CPUボード
- 4 インターフェースボード(他のインターフェースボードが利用可能です)

## リモートセンサーカメラVRmD3C-X-E

リモートセンサーカメラは、ベースユニットとフレックスフォイルケーブルによってベースユニットに接続される外部センサーボードからなっています。リモートセンサーカメラには、ボードレベルカメラ(OEMバージョン)と光学系付きボードレベルカメラ(COBバージョン)があります。

ベースユニットは、センサー接続ボード、2枚のCPUボード、インターフェースボードからなっています。

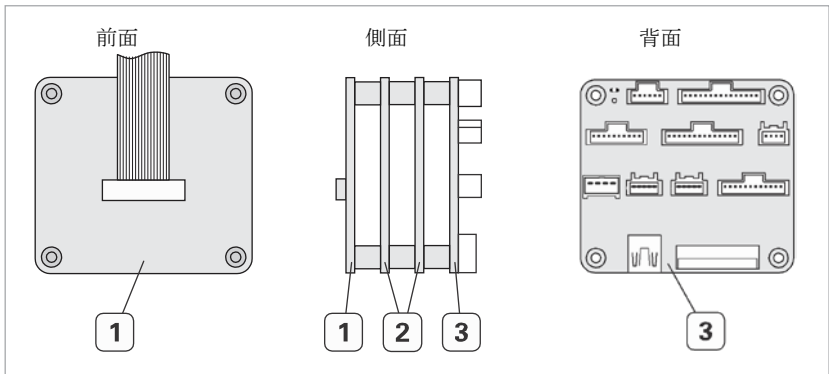


図3: CUEO3インターフェースボードを備えたインテリジェントリモートセンサーカメラのベースユニット

- 1 センサー接続ボード(外部センサーボードはフレックスフォイルケーブルで接続されます)
- 2 CPUボード
- 3 インターフェースボード(他のインターフェースボードが利用可能です)

## マルチセンサーカメラVRmD3MFC

マルチセンサーカメラは、ベースユニットとフレックスフォイルケーブルによってベースユニットに接続される4枚までの外部センサーボードからなっています。それらにはボードレベルカメラ(OEMヴァージョン)と光学系付きボードレベルカメラ(COBヴァージョン)があります。

ベースユニットは、センサー接続ボード、FPGAボード、2枚のCPUボード、インターフェースボードからなっています。

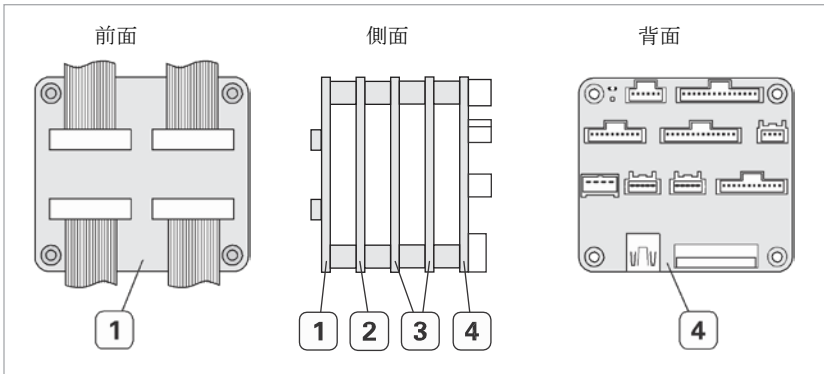


図4: CUEO3インターフェースボードを備えたインテリジェントマルチセンサーカメラのベースユニット

- 1 センサー接続ボード(外部センサーボードはフレックスフォイルケーブルで接続されます)
- 2 FPGAボード
- 3 CPUボード
- 4 インターフェースボード(他のインターフェースボードが利用可能です)

### 3.3 インターフェイスボード

現在、D3カメラは2種類のインターフェイスボードが利用可能です。

#### VRmCUEO3 Picoインターフェイスボード

このインターフェイスボードには、カメラスタックの他のボードと同じフォームファクタがあります。

これはミニチュアのコネクタと、SATAとJTAGを除くD3プラットフォームのすべての対応するインターフェイスを備えています。

→概要については13ページの3.3.1を参照。

#### VRmEIO3インターフェイス評価用ボード

このボードは、D3プラットフォームが対応する最大範囲のインターフェイスを備えています。VRmEIO3ボードは標準のコネクタとピンヘッダーを備えているので、すべてのインターフェイスを都合よく評価することができます。評価結果に基づいて、カスタムインターフェイスボードを注文することができます。

→概要については14ページの3.3.2を参照。

### 3.3.1 VRmCUEO3インターフェースボード

#### 備考

→詳しいピン配列については11.2を参照。かぎ括弧内の名称はボードのラベリングです。

コネクタとインターフェース

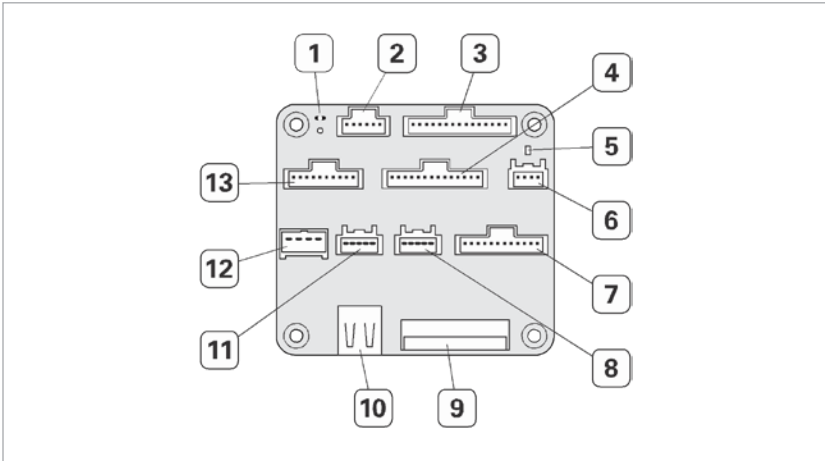


図5: VRmCUEO3インターフェースボード[かぎ括弧内の名称: ボードのラベリング]

- 1 [VBAT] リアルタイムクロック用3Vバッテリーコネクタ
- 2 [GPIO] 4 x GPIOまたは(SPDIF出力を含む)オーディオコーデック用McASPバス
- 3 [ETHERNET] ギガビットイーサネット
- 4 [SER SVID] RS232、S-Video、UART0 (RS232とシリアルコンソール)、UART1、CAN
- 5 ステータスLED
- 6 [TRIG + STRB] トリガーとストロボ
- 7 [SPI] コントロールLED、ウォッチドッグ、電源ボタン、リセット、SPI
- 8 [USB0] USBポート0
- 9 [RGB888] RGB888およびI<sup>2</sup>C 2線シリアルインターフェース
- 10 [HDMI] マイクロHDMI
- 11 [USB1] USBポート1
- 12 [5VDC] 電源入力(5V DC)
- 13 [uSD] SDIO (たとえばMicroSDカード用)

### 3.3.2 VRmEIO3インターフェース評価用ボード

#### **i** 備考

→詳しいピン配列については11.4を参照。かぎ括弧内の名称はボードのラベリングです。

コネクタとインターフェース(前面)

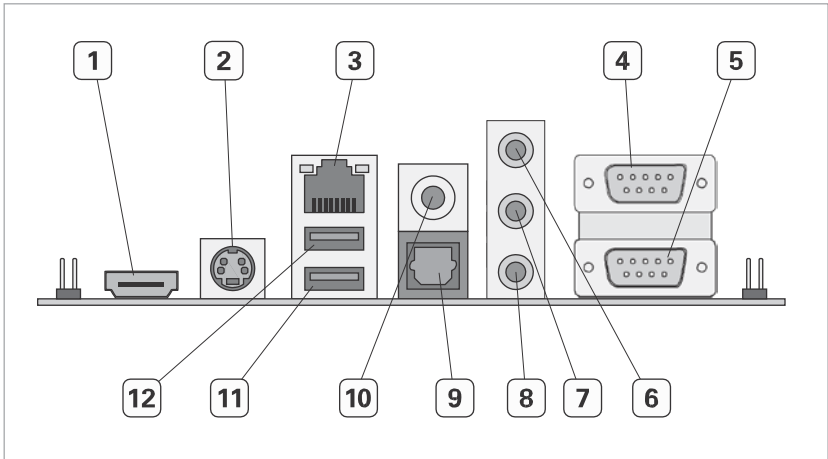


図6: VRmEIO3ボード、前面

- 1 HDMI (タイプAポート)
- 2 S-Video (標準4ピンポート)
- 3 ギガビットイーサネット(RJ45ポート)
- 4 UART0: RS232およびシリアルコンソール(DE-9 D-Subポート)
- 5 UART1: RS232/RS485 (DE-9 D-Subポート)
- 6 オーディオライン入力(3.5 mm 電話コネクタ)
- 7 オーディオライン出力(3.5 mm 電話コネクタ)
- 8 マイクロフォン入力(3.5 mm 電話コネクタ)
- 9 S/PDIF出力 光学(TOSLINKポート)
- 10 S/PDIF出力 同軸(RCA /シンチポート)
- 11 USB0ポート(タイプA)
- 12 USB1ポート(タイプA)

コネクタとインターフェース(上部)

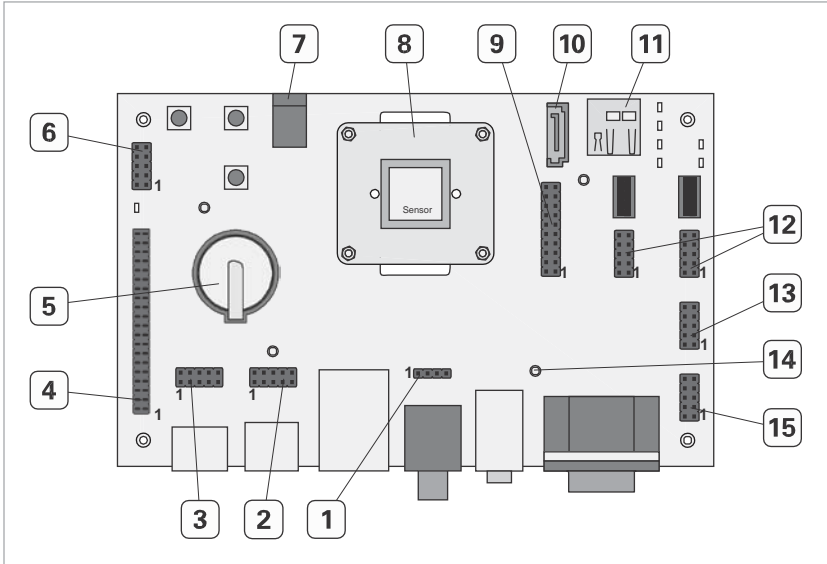


図7: VRmEIO3ボードのインターフェース、上部[かぎ括弧の名称: ボードのラベリング]

- 1 [I2C] I<sup>2</sup>C 2線シリアルインターフェース(1 x 4 ピンヘッダー)
- 2 [SPI] SPIバス(2 x 5 ピンヘッダー)
- 3 [muxed GPIO] 7GPIO (DaVinci、2 x 5 ピンヘッダー)
- 4 [VIDEO I/O] RGB888/GPIO (2 x 20 ピンヘッダー)
- 5 リアルタイムクロック用3 Vコインバッテリー(CR2032)
- 6 [TRIG & STR] トリガーとストロボ(2 x 5 ピンヘッダー)
- 7 [5V DC IN] 電源入力(5 V DC、3 A)
- 8 インテリジェントカメラスタック
- 9 [JTAG debug] JTAG (2 x 10 ピンヘッダー)
- 10 [SATA] SATA2 AHCI 3 Gbit/秒
- 11 [microSD] MicroSDカードスロット(最大 32 GB) SDXC、ブート可能
- 12 [GPIO\_0, GPIO\_1] 16 GPIO (GPIO\_0、GPIO\_1) I<sup>2</sup>C経由(2 x 5 ピンヘッダー)
- 13 [CAN] CANバス(2 x 5 ピンヘッダー)
- 14 グラウンドピン(4 x ボード全体)
- 15 [SYS panel] システムパネル(2 x 5 ピンヘッダー)



コントロールとインディケータ(上部)

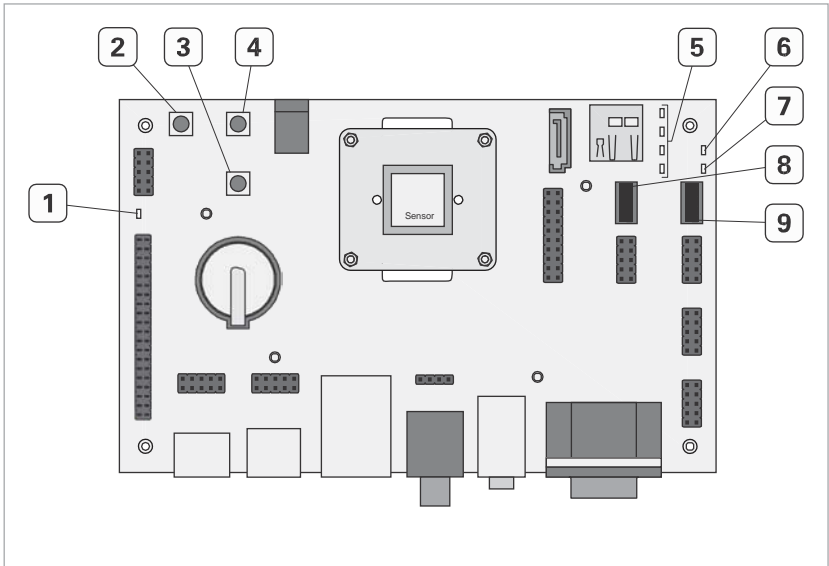


図8: VRmEIO3ボードのコントロールとインディケータ、上部[かぎ括弧の名称: ボードのラベリング]

- 1 ストロボLED (ストロボシグナルにより発光) STRB
- 2 [TRIG] 手動トリガーボタン
- 3 [PWRBTN] 電源ボタン
- 4 [RESET] リセットボタン
- 5 4つのユーザー定義可能LED(下から上へ: ULED1~ULED4、GPIO 148~151によってコントロール)
- 6 ステータスLED (緑) CTRL
- 7 ウォッチドッグLED (赤) WD
- 8 [OPT1] DIPスイッチ
- 9 [OPT2] DIPスイッチ

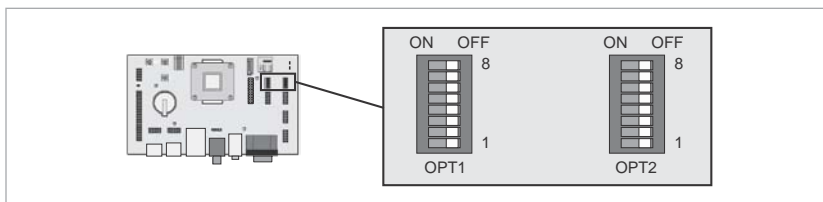


図9: VRmEIO3インターフェースボードのDIPスイッチOPT1とOPT2のレイアウト

#### OPT1 | DIPスイッチ 1 ... 8

|     |     |                              |                          |
|-----|-----|------------------------------|--------------------------|
| 1-3 | OFF | OFFのままにしてください! 保守点検にのみ使用します! |                          |
| 4   |     | 機能なし                         |                          |
| 5   | ON  | USB0 port = ホスト(初期設定)        | OFF USB0 port = デバイス     |
| 6   | ON  | USB1 port = ホスト(初期設定)        | OFF USB1 port = デバイス     |
| 7   | ON  | [VIDEO I/O] = ビデオ入力          | OFF [VIDEO I/O] = ビデオ出力  |
| 8   | ON  | [VIDEO I/O] = GPIO           | OFF [VIDEO I/O] = ビデオI/O |

#### 備考

##### USBコントロールの機能

カメラのUSBポートのUSB0とUSB1は初期設定でUSBホストとして設定されています。USBコンフィギュレーションを変更するにはディップスイッチの設定だけでは十分ではありません。カメラのLinuxカーネルを再コンパイルする必要があります(→セクション8.2を参照)。

#### OPT2 | DIPスイッチ 1 ... 8

|   |     |                                    |     |                                |
|---|-----|------------------------------------|-----|--------------------------------|
| 1 | ON  | [muxed GPIO]<br>ピン3...5 = GPIO有効化  | OFF | オーディオポート<br>オーディオ有効化           |
| 2 | ON  | デジタルオーディオ出力有効化<br>(1 = OFFの場合)     | OFF | アナログオーディオ出力有効化<br>(1 = OFFの場合) |
| 3 | OFF | OFFの位置にしてください! サービス用にのみ使用します!      |     |                                |
| 4 | ON  | [muxed GPIO]<br>ピン7...8 = GPIO有効化  | OFF | [CAN]<br>CANバス有効化              |
| 5 | ON  | [muxed GPIO]<br>ピン9...10 = GPIO有効化 | OFF | UART1ポート<br>UART1有効化           |
| 6 |     | 機能なし                               |     |                                |
| 7 | ON  | UART1 = RS232                      | OFF | UART1 = RS485                  |
| 8 |     | 機能なし                               |     |                                |

## 3.4 付属品

### 3.4.1 VRmCUEO3付属品キット

#### VRmCUEO3付属品キット

VRmCUEO3 OEMインターフェースボードを備えたD3カメラ用

- 1 国際対応ケーブル付きAC電源アダプター5 V / 15 W
- 1 電源ケーブル4ピン、150 mm (4ピンプラグメス ⇄ 電源アダプターレセプタクル)
- 1 イーサネットケーブルCAT6、3 m
- 1 RJ45ポートのあるイーサネットアダプターパネル
- 1 GBitイーサネットアダプターケーブル、150 mm (15ピンプラグメス ⇄ 10ピンプラグメス)
- 1 microSDカード 8 GB
- 1 USB microSDカードリーダー
- 1 microSDアダプターケーブル、150 mm (10ピンプラグメス ⇄ 10ピンプラグメス)
- 1 microSDアダプターパネル
- 1 Micro HDMIケーブル、19ピン、1 m
- 1 トリガーおよびストロボケーブル(4ピンプラグメス ⇄ 4シングルピンコネクタメス)
- 1 GPIOケーブル、500 mm (6ピンプラグメス ⇄ 6シングルピンコネクタメス)
- 1 S-Videoケーブル、2 m
- 1 スルモデムケーブル、3 m
- 1 S-Video、RS232、UART1、CANアダプターケーブル(14ピンプラグメス ⇄ S-Video、9ピンD-Subプラグオス、8シングルピンコネクタメス)用Serial/S-Videoケーブル、250 mm
- 1 USBアダプターケーブル、150 mm (5ピンプラグメス ⇄ USBミニレセプタクル)
- 1 USB-Aアダプターケーブル、150 mm (4ピンプラグメス ⇄ USB Aレセプタクル)
- 1 SPI、LED、ウォッチドッグ、電源/リセットボタン用SPIケーブル、500 mm (12ピンプラグメス ⇄ 12シングルピンコネクタメス)
- 1 USBワイヤレスアダプター
- 1 ジャンパーケーブル (オス ⇄ オス)



## 備考

### ケーブルプラン

VRmCUEO3付属品キットの詳細なケーブルプランは→セクション11.3にあります。

## 3.4.2 VRmEIO3付属品キット

### VRmEIO3付属品キット

D3 cameras with VRmEIO3インターフェース評価ボードを備えたD3カメラ用

- 1 国際対応ケーブル付きAC電源アダプター5 V / 15 W
- 1 スルモデムケーブル、3 m
- 1 USB-Serialアダプター、9ピン
- 1 イーサネットケーブルCAT6、3 m
- 1 microSDカード 8 GB
- 1 USB microSDカードリーダー
- 1 HDMIケーブル、2 m
- 1 S-Videoケーブル、2 m
- 7 リボンケーブル、0.9 m (ピンヘッダー 2 x 5 ピン ⇄ シングルピンコネクタメス)
- 1 リボンケーブル、0.9 m (ピンヘッダー 2 x 20 ピン ⇄ シングルピンコネクタメス)
- 1 リボンケーブル、0.5 m (ピンヘッダー 1 x 4 ピン ⇄ シングルピンコネクタメス)
- 1 USBワイヤレスアダプター
- 1 ジャンパーケーブル (メス ⇄ メス)
- 1 ジャンパーケーブル (オス ⇄ メス)

## 4 ソフトウェアのインストール

### 4.1 概要

#### 4.1.1 D3の開発環境

##### **D3オペレーティングシステム(ファームウェア)**

D3インテリジェントカメラは、D3ファームウェアに付属するARM用Linux 12.04 LTSに基づいていて、すでにカメラにインストールされています。カメラのアプリケーションは、D3にインストールされて実行されます。けれども、このカメラアプリケーションは、LinuxホストPCの上で開発されてクロスコンパイルされる必要があります。

##### **LinuxホストPCの開発環境**

D3インテリジェントカメラアプリケーションのための開発環境はUbuntu Linux 12.04 LTS (32ビット)のホストPCを必要とします。以下のセクションは、D3アプリケーションのためにLinuxホストPCをセットアップしてクロスビルドする環境を構築するのに必要な情報を提供します。

#### 4.1.2 VRmagicソフトウェアパッケージ

##### **VRmagic Custom Ubuntu**

便利なようにVRmagicはホストPC用にUbuntu 12.04 LTS (32ビット)のISOイメージを提供しています。これにはたとえばVRmagic Easy Installerのようないくつかのアプリケーションとコンフィギュレーションが事前にインストールされています。

##### **VRmagic Easy Installer**

VRmagic Easy Installerは、以下に説明するすべてのVRmagicソフトウェアコンポーネントをインストールして設定するのに便利です。

##### **VRmagic Linux PCカメラランタイムとCamLabアプリケーション**

カメラランタイムによってLinuxホストPCはたとえばVRmagic CamLabアプリケーションを使用してVRmagicカメラにアクセスすることができます。CamLabによって、イーサネット上で使用できるすべてのVRmagicカメラの検出、試験、設定が可能です。

## VRmagic Linux PC Camera SDK

Linux PC Camera SDKは、ホストPC上で実行されるカメラアプリケーションの開発に必要です。したがってこのSDKは、D3で直接実行されるアプリケーションをコンパイルするには必要ありません。けれども、Linux PC Camera SDKは、最初にホストPC上でカメラアプリケーションをデバッグして、後でD3用にそれをクロスコンパイルするのに役立ちます。

## VRmagic Linux D3 Camera SDK

Linux D3 Camera SDKは、D3で実行されるアプリケーションを開発するのに必要なライブラリとヘッダーファイルを含んでいます。さらに、デモソースコード、ドキュメント、D3カメラファームウェア、D3をアップデートするためのツールを含んでいます。また、ARM CPUとDSP用のクロスコンパイル環境をセットアップして設定するためのスクリプトも含んでいます。より正確には、これはメタパッケージで、以下のコンポーネントをインストールします。

- multiarch-armhfクロスコンパイル環境  
このクロスコンパイル環境は、LinuxホストPC上でD3のARM CPUで動作させるアプリケーションをコンパイルするのに必要です。現在のところUbuntu Linux 12.04 LTS用のARMHF EABIツールチェーンに対応しています。
- VRmagic D3 EZSDK  
これはさらにD3のC674x DSPを使用するアプリケーション用のクロスコンパイル環境でもあります。VRmagic DSPデモのコンパイルやDSPアプリケーションの開発にはこれをコンパイルする必要があります。  
D3 EZSDKもLinuxカーネルソースに含まれています。
- D3ファームウェア
- D3ファームウェアアップデートツール

## 4.2 LinuxホストPCのセットアップ

### 4.2.1 Ubuntu Linuxのインストール

#### VRmagic Custom Ubuntu

VRmagic Custom Ubuntuは、Ubuntu Linux 12.04 LTS (32ビット)をカスタマイズしたもので、使用開始が容易になります。インストールの環境は試験済です。もちろん、通常のUbuntu Linux 12.04 LTS (32ビット)を使用することもできます。

#### 仮想マシン

D3環境の基礎であるホストPCを仮想マシンでセットアップすることを推奨します。この目的にはWindows版VMwareが適切ですが、他の仮想マシンや実際のPCを使用することもできます。

以下では、開発用PCにVMware Player仮想マシンがインストールされるものと仮定します。詳しくはメーカーの仮想マシンの説明書を参照してください。

#### VMware仮想マシンへのVRmagic Custom Ubuntuのインストール:

1. VRmagic Custom Ubuntu image from [www.vrmagic-imaging.com](http://www.vrmagic-imaging.com)からVRmagic Custom Ubuntuイメージをダウンロードします。
2. VRmagic Custom Ubuntuを仮想マシンたとえばVMwareにインストールします。
3. 仮想マシンのデスクトップ統合を改善するためにVMware Toolsをインストールします。
4. 仮想マシンの設定で、**Virtual Machine Settings > Network Adapter**によってネットワークアダプターを**bridged**に設定します。CamLabを介してカメラにアクセスし、**vrm-eth-updater**を介してカメラをアップデートするのにこれは必須です。
  - ▶ これで仮想マシンに動作するVRmagic Custom Ubuntuをインストールしたことになります。

#### 備考

##### ホストPCとD3カメラ

以下では、「ホストPC」はUbuntu Linux 12.04 LTSでのことで、仮想マシンまたは実際のPCのインストールされたものです。D3カメラのことを指す場合は、常にテキストで示します。

## 4.2.2 VRmagicソフトウェアパッケージのインストール

### VRmagic Easy Installer

VRmagic Easy Installerは、VRmagic Custom Ubuntuに事前にインストールされていて、必要なソフトウェアコンポーネントのインストールを支援するものです。Ubuntu 12.04 (32ビット)の別のインストールを使用している場合は、VRmagic Easy Installer Debianパッケージを[www.vrmagic-imaging.com](http://www.vrmagic-imaging.com)からダウンロードしてホストPCにインストールします。

#### 備考

##### D3シリアルナンバーとパスワード

以下の手順ではカメラのシリアルナンバーと対応するパスワードが必要です。いずれもカメラに添付されています。

**VRmagic Easy Installerを使用したVRmagicソフトウェアパッケージのインストール:**

1. ホストPCにログインします。
  - ▶ **vrm-easy-installer**アイコンはデスクトップとランチャーバーにあります。
2. インストールを開始するには、**vrm-easy-installer**アイコンをダブルクリックします。
  - ▶ ターミナルウィンドウが表示されます。ルートパスワードの入力を促されます。
3. Linuxユーザーパスワードを入力します。
  - ▶ ウェルカムスクリーンが表示されます。
  - ▶ 以下では、ホストシステムにどのVRmagicソフトウェアパッケージをインストールしたいか尋ねられます(→セクション4.1.2)。
  - ▶ 次に**multiarch-armhf**クロスコンパイル環境をセットアップして設定するかどうか尋ねられます。
  - ▶ 最後に**D3 EZSDK (DSPクロスコンパイル環境とカーネルソース)**をセットアップするかどうか尋ねられます。完全なインストールでは、**TI EZSDK**インストールファイルがホームディレクトリに存在しなければなりません(→セクション4.3.3)。
4. D3カメラのシリアルナンバーと対応するパスワードを促されたら入力します。
  - ▶ シリアルナンバーとパスワードが正しい場合は、インストール手順が開始します。インストール中に、外部サーバーからソフトウェアがダウンロードされます。
  - ▶ すべての質問に「Yes」と答えた場合は、VRmagic Easy Installerは以下のタスクを実行します。
    - VRmagicリポジトリを**apt sources.list**に追加
    - すべてのホストPCソフトウェアコンポーネントをインストール
    - D3 SDKをホストPCにインストール
    - ARMクロスコンパイル環境を設定
5. すべてのタスクが問題なく完了したら<Return>を押すように促されます。
6. ステップ3で、D3 EZSDKの設定を選択した場合はEasy Installerは**vrm-d3-ezsdk-installer**を立ち上げてDSPクロスコンパイル環境を設定します。詳しくは画面上の指示に従うか→セクション4.3.3と4.3.4を参照してください。



## 4.2.3 VRmagicソフトウェアパッケージのアップデート

### ■ 備考

#### ソフトウェアパッケージのアップデート

VRmagic Easy InstallerでVRmagicソフトウェアパッケージをインストールしたばかりなら、このセクションをスキップして、ソフトウェアのアップデートをしたい場合にのみ適用してください。

以下の指示は、VRmagic Easy Installerが前に実行された場合、特にホストPCのsources.listにVRmagicリポジトリのapt-getソースが追加されている場合にのみ適用されます。

#### VRmagic Easy Installerを使用したVRmagicソフトウェアパッケージのアップデート

1. ホストPCのターミナルウィンドウを開きます。
2. パッケージリストをアップデートするには`sudo apt-get update`を実行します。
3. VRmagic Easy Installerをアップデートするには  
`sudo apt-get install vrm-easy-installer`を実行します。
4. Easy Installerを→セクション4.2.2で説明したように動作させます。

## 4.3 クロスコンパイル環境の手動設定

このセクションとサブセクションは、→セクション4.2.2のすべてのステップを実行してクロスコンパイル環境を拡張するか維持したい場合に適用されます。たとえば、さらに開発ライブラリをインストールしたりVRmagic D3 EZSDKをアップデートしたりする場合です。

### 4.3.1 armhf Toolchainの手動設定

D3カメラのARM CPU用にLinuxホストPC上でアプリケーションをコンパイルするのにarmhf EABIクロスコンパイル環境が必要です。armhfアーキテクチャ用に追加の開発ライブラリをインストールするには、以下に説明するようにパッケージ管理システムを設定します。

### ■ 備考

multiarch-armhfクロスコンパイル環境をインストールして設定するのにVRmagic Easy Installerを使用する場合は(→セクション4.2.2)、このセクションをスキップできます。

### *armhf toolchainの手動設定:*

1. ホストPCのターミナルウィンドウを開きます。
2. debianパッケージvrmagic-d3-toolchain-addonsをインストールします。
3. Armhfパッケージのインストール用にパッケージ管理システムを設定するために `sudo vrm-configre-multiarch-armhf-support`を実行します。
4. →セクション4.3.2にあるように開発ライブラリをarmhfクロスコンパイル環境にインストールします。

これでホストPC上のARMクロスコンパイル環境で作業ができます。



### 備考

VRmagicデモをクロスコンパイルするのに必要なデペンデンシー便宜のために、VRmagic ARMデモをクロスコンパイルするのに必要なすべてのデペンデンシーを自動的にインストールするのに `vrm-install-demo-dependencies-armhf` スクリプトを使用することができます。それは `/opt/vrmagic/sdk-*/D3/development_kit/bin`にあります。

### 4.3.2 追加のarmhfライブラリのインストール

以下の指示は、multiarch-armhfクロスコンパイル環境をたとえばVRmagic Easy Installerでインストールして設定する場合を仮定しています(→セクション4.2.2)。

### *D3環境の追加ソフトウェアのインストール:*



### 注意

armhfパッケージをパッケージ管理システムで手動でインストールしないでください。armhfパッケージをインストールするのにホストPCのパッケージ管理システム(たとえばaptitude)を使用しないでください。ホストPCのUbuntuインストールが破損する場合があります。以下のステップに従ってください。

1. ホストPCのターミナルウィンドウを開きます。
2. D3環境用の追加のarmhfパッケージを `/usr/arm-linux-gnueabi/vrmagic/`のクロスビルド環境に `vrm-multiarch-armhf`スクリプトを使用してルートユーザーとしてインストールします。
3. コマンドオプションを見るにはターミナルに `vrm-multiarch-armhf help`とタイプします。

- たとえばパッケージ `libbsd1.2-dev` をダウンロードしてインストールするにはターミナルで以下のコマンドを入力します:  
`vrm-multiarch-armhf download_install libsd1.2-dev`

### 4.3.3 TI EZSDK のダウンロード

D3 の DSP コプロセッサを使用するアプリケーションを作成するには、米国テキサスインスツルメンツ社 (TI) によって提供されるいくつかのソフトウェアパッケージが必要です。TI は DM8148 用の TI EZSDK ですべての必要なパッケージのコレクションを提供しています。D3 の armhf アーキテクチャでこのソフトウェアを動作させるために、VRmagic はいくつかのパッケージを変更して、これらの変更されたパッケージを debian パッケージ `vrmagic-linux-d3-ezsdk` 内に収めています。

完全な `vrmagic-linux-d3-ezsdk` をインストールして設定するには、米国テキサスインスツルメンツ社の TI EZSDK がコンピュータ内に存在していなければなりません。ライセンスの制限があるので、以下に示すように TI EZSDK インストーラファイルを自分でダウンロードする必要があります。

#### *TI EZSDK のダウンロード:*

- <http://www.ti.com/tool/LINUXEZSDK-DAVINCI> にアクセスします。
- `ezsdk_dm814x-evm_5_05_02_00_setuplinux` という名称の DM814x EZSDK 5.05 インストールファイルをダウンロードします。
  - ▶ 登録するか既存の TI アカウントでログインするかを尋ねられます。
- インストールファイル `ezsdk_dm814x-evm_5_05_02_00_setuplinux` を Ubuntu Linux ホスト PC のホームディレクトリにコピーします。

#### 4.3.4 D3 EZSDKとTI EZSDKの手動インストール

debianパッケージvrmagic-linux-d3-ezsdkはスクリプトvrm-d3-ezsdk-installerを提供します。これはTI EZSDKとVRmagicパッチをインストールして設定するのに使用することができます。

#### i 備考

VRmagic Easy Installerを使用していて、TI EZSDKのインストールを完了したいだけの場合は、以下の手順のステップ3からの開始で十分です。

TI EZSDKを含むVRmagic D3 EZSDKをすでにインストールしている場合は、このセクションにある手順全体をスキップできます(→セクション4.2.2)。

#### *VRmagic D3 EZSDKとTI EZSDKの手動インストール:*

1. ホストPCのターミナルウィンドウを開きます。
2. Debianパッケージvrmagic-linux-d3-ezsdkをホストPCにインストールします。
3. TI EZSDKをダウンロードします (→セクション4.3.3)。
4. VRmagic D3 EZSDKを設定するには、  
`sudo vrm-d3-ezsdk-installer <YOUR/PATH/TO>/ezsdk_dm814x-evm_5_05_02_00_setu linux`を実行します。
5. 画面の指示に従います。
  - ▶ D3 EZSDKはopt/vrmagic/vrm-d3-ezsdkにインストールされます。
  - ▶ TI EZSDKはopt/ti/ti-ezsdk-5\_05\_02\_00にインストールされます。

### 4.3.5 VRmagic D3 EZSDKのアップデート

VRmagic D3 EZSDKの旧バージョンを新バージョンにアップデートするには以下の手順で行います。

#### 備考

TI EZSDKのアップデートは不要です

VRmagic D3 EZSDKは/opt/ti/ti-ezsdk-5\_05\_02\_00の変更されないTI EZSDKへのシンボリックリンクを含んでいるのでVRmagic D3 EZSDKを更新する場合に以前にインストールされたTI EZSDKを再インストールする必要はありません。

#### VRmagic D3 EZSDKのアップデート:

1. ホストPCのターミナルウィンドウを開きます。
2. `sudo rm -rf /opt/vrmagic/vrm-d3-ezsdk`を実行して古いD3 EZSDKを含んでいる/opt/vrmagic/vrm-d3-ezsdkディレクトリを削除します。
3. 新規D3 EZSDKをインストールするには  
`sudo vrm-d3-ezsdk-installer --no-ti-ezsdk`を実行します。
  - ▶ これでホストPC上に更新されたDSPクロスコンパイル環境ができます。

### 4.3.6 Linuxカーネルソースのインストールとアップデート

現在、D3ファームウェアのLinuxカーネルソースはVRmagic D3 EZSDKと共に出荷されています。D3 EZSDKをインストールすると、カーネルソースはホストPCの/opt/vrmagic/vrm-d3-ezsdk /board-support/linux-DM8148-\*ディレクトリにあります。READMEファイルにはコンパイルの説明もあります。

- ▶ VRmagic D3 EZSDKのインストール方法については→セクション4.3.4を参照してください。
- ▶ VRmagic D3 EZSDKのアップデート方法については→セクション4.3.4を参照してください。

### 4.3.7 PCにインストールされるコンポーネント

Ubuntu Linuxシステムには以下のディレクトリとサブディレクトリが作成されます。

| <b>/opt/vrmagic/sdk-*/D3/</b> (D3プラットフォーム用SDK; * = SDKバージョンナンバー)      |   |
|---|---|
| <b>development_kit/</b>   | D3用VRmUsbCam DevKit<br>vrmagic-linux-d3-armhf-demos-src-*.zip<br>ARMデモアプリケーションを含むzipファイル<br>vrmagic-linux-d3-armhf-dsp-demos-src-*.zip<br>DSPデモアプリケーションを含むzipファイル |
| <b>bin</b>  |   |
| <b>dsp</b>  |   |
| <b>include</b>  | VRmagic APIインクルードファイル   |
| <b>lib</b>  | VRmagic API開発ライブラリ  |
| <b>vm_lib</b>   | VRmagicマシンビジョンライブラリ   |
| <b>docs</b>   | ドキュメント  |
| <b>/opt/vrmagic/firmware/</b> (D3カメラ用ファームウェア; * = SDKバージョンナンバー)       |   |
|   | vrmagic-linux-d3-firmware-*.zip<br>D3ファームウェア含むzipファイル   |
| <b>/opt/vrmagic/sdk-*/x86/</b> (Linux x86ホストPC用SDK; * = SDKバージョンナンバー) |   |
| <b>development_kit/</b>   | Linux x86ホストPC用VRmUsbCam DevKit   |
| <b>demos</b>  | APIの使用例   |
| <b>doc</b>  | ドキュメント  |
| <b>include</b>  | VRmagic APIインクルードファイル   |
| <b>lib</b>  | VRmagic API開発ライブラリ  |
| <b>vm_lib</b>   | VRmagicマシンビジョンライブラリ   |
| <b>wrappers</b>   | C++ラッパー   |

## 5 ハードウェアのインストール

### 5.1 外部センサーボードの接続



#### 備考

このセクションは、リモートセンサーカメラとマルチセンサーカメラにのみ適用されます。



#### 注意

カメラで起こりうる不具合

指示と図面にあるようにフレックスフォイルケーブルの接点の正しい方向を確認してください。これを間違えるとデバイスは機能しません。

ベースユニットへのフレックスフォイルケーブルの接続:

1. ベースユニットのコネクタのロックを引き抜きます (→31ページの図10)。
2. フレックスフォイルケーブルをボードから出て見えている接点に挿入します。
3. ロックを押し戻してケーブルを固定します。
4. ほかの外部センサーボードについても1~3のステップを繰り返します。

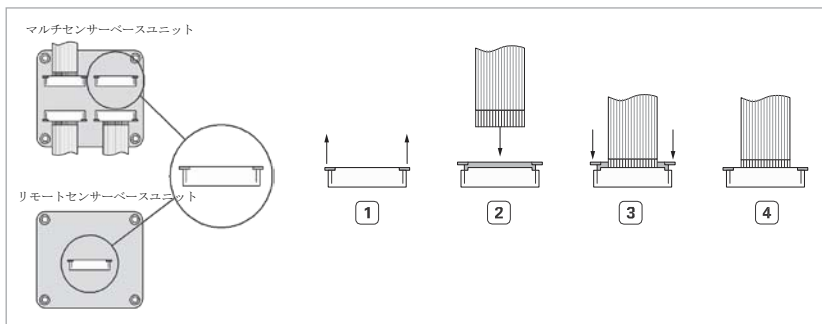


図10: ベースユニットへのフレックスフォイルケーブルの接続: 接点は見えませんが青いマーカースは見えます。

センサーボードへのフレックスfoilケーブルの接続:

1. センサーボードのコネクタのロックを引き抜きます (→32ページの図11)。
2. フレックスfoilケーブルをセンサーボードから出て見えている青印とセンサーボードに面している接点に挿入します。
3. ロックを押し戻してケーブルを固定します。
4. ほかの外部センサーボードについても1~3のステップを繰り返します。

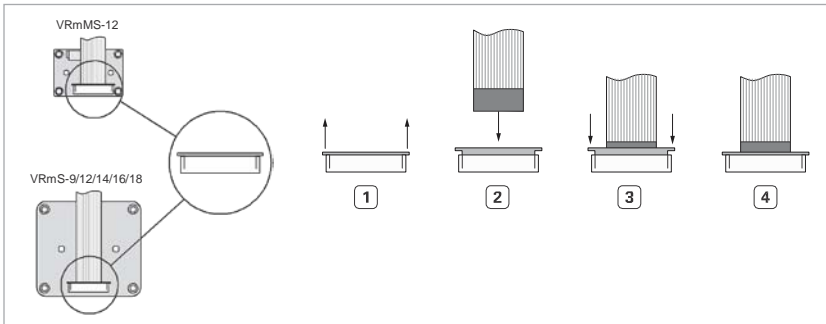


図11: センサーボードへのフレックスfoilケーブルの接続: 接点は見えませんが青いマーカーは見えます。



## 5.2 VRmCUEO3ボードの接続

カメラにアクセスするための最低条件は、機能するイーサネット接続と電源です。また、アプリケーションに応じて、他のインターフェース、たとえばシリアル接続、S-Video出力、トリガー/ストロボ、I/Oが必要になる場合があります。

### 備考

以下の説明はVRmCUEO3付属品キットを注文してあるものとします(→セクション3.4.1)。

→詳しいピン配列はセクション11.2にあります。

VRmCUEO3インターフェースボードでは、すべてのインターフェースはミニチュアコネクタで行われます。VRmCUEO3付属品キットには、片方が適合するメスミニチュアコネクタで、もう片方が別の規格かシングルピンコネクタのアダプターケーブルが含まれています。

### 5.2.1 イーサネット

#### イーサネットへの接続:

1. カメラの15ピンイーサネットコネクタに付属のイーサネットアダプターケーブルを接続します (→14ページの図5)。
2. イーサネットポートがはんだ付けされているパネルをイーサネットアダプターケーブルのもう一方の端に接続します。
3. 付属のイーサネットケーブルをイーサネットポートとローカルエリアネットワークに接続します。

### 備考

→イーサネットのセットアップの説明はセクション6.1にあります。

## 5.2.2 GPIO

### **GPIOの接続:**

1. 付属のGPIOケーブルをカメラの6ピンGPIOコネクタに接続します (→14ページの図5)。
2. アプリケーションに接続するのにケーブルのもう片方のシングルピンコネクタを使用します。→ピン配列については11.2を参照。

## 5.2.3 MicroSDカードスロット

### **MicroSDカードスロットの接続:**

1. 付属のMicroSDカードアダプターケーブルをカメラの10ピンSDIOコネクタに接続します (→14ページの図5)。
2. MicroSDスロットがはんだ付けされているパネルをMicroSDカードアダプターケーブルのもう一方の端に接続します。
3. 付属のMicroSDカードを挿入します。

## 5.2.4 シリアルコンソール、RS232、S-Video、UART1、CANバス

### **シリアルコンソール、RS232、S-Video、UART1、CANバスの接続:**

1. 付属のアダプターケーブルをカメラの14ピンのRS232/S-Video/UART1/CANコネクタに接続します (→14ページの図5)。
  - ▶ ケーブルのもう片方はいくつかのコネクタに分かれています。
2. RS232を介してカメラのシリアルコンソールにアクセスするには、付属のマルチモデムケーブルをアダプターケーブル上のオス9ピンD-Subプラグに接続します。マルチモデムケーブルのもう片方をUSBシリアルアダプターを使用してホストPCに接続します。
3. カメラのS-Video出力を見るには、付属のS-Videoケーブルをアダプターケーブル上のS-Videoレセプタクルに接続します。S-Videoケーブルのもう片方を出力デバイスに接続します。
4. UART1とCANバスを接続するには、アダプターケーブル上のシングルピンコネクタを使用します。→ピン配列についてはセクション11.2を参照。

## 5.2.5 トリガーとストロボ

### トリガーとストロボの接続:

1. 付属のトリガーおよびストロボケーブルをカメラの4ピンのトリガー/ストロボコネクタに接続します (→14ページの図5)。
2. アプリケーションに接続するのにケーブルのもう片方のシングルピンコネクタを使用します。→ピン配列についてはセクション11.2を参照。

## 5.2.6 USB

### USBの接続:

#### 備考

#### カメラのUSBコントローラーの機能

カメラのUSBポートのUSB0とUSB1は初期設定でUSBホストとして設定されています。USBデバイスをカメラに接続するには、大きな**USB A**レセプタクルの付いた付属のUSBアダプターケーブルを使用します。

カメラがUSBホストに接続される場合は、**USB ミニレセプタクル**の付いたUSBアダプターのみを使用します。この場合、カメラのUSBポートの1つをデバイスとして設定しなければなりません。

カメラのUSBポートのUSBコントローラーの機能(ホストまたはデバイス)を変更するには→セクション8.2を参照してください。

1. カメラにUSBデバイスを接続するには、**USB A**レセプタクルの付いた付属のUSBアダプターを使用します。アダプターケーブルをUSBホストとして設定されたカメラの5ピンUSBコネクタに接続します (→14ページの図5)。初期設定では、両方のUSBインターフェースはホストとして設定されます。
2. USBデバイスとしてのカメラを**USB ミニレセプタクル**の付いた付属のUSBアダプターケーブルを使用してUSBホストに接続します。アダプターケーブルをUSBデバイスとして設定されるカメラの5ピンUSBコネクタに接続します (→14ページの図5)。

## 5.2.7 SPI、LED、ウォッチドッグ、電源/リセットボタン

### **SPI、LED、ウォッチドッグ、電源/リセットボタンの接続:**

1. 付属のアダプターケーブルをカメラの12ピンSPI/LED/ウォッチドッグ/電源/リセットコネクタのラベルSPIに接続します (→14ページの図5)。
2. アプリケーションに接続するのにケーブルのもう片方のシングルピンコネクタを使用します。→ピン配列についてはセクション11.2を参照。

## 5.2.8 HDMI

### **HDMIの接続:**

1. カメラのラベルHDMIのMicro HDMIレセプタクルに接続するのに付属のMicro HDMIケーブルを使用します (→14ページの図5)。
2. Micro HDMIケーブルのもう片方を出力デバイスに接続します。

## 5.2.9 外部バッテリー

カメラの電源が切られている間に、カメラ内部のリアルタイムクロックに電力を供給するのに外部3Vバッテリーが必要です。VRmCUEO3ボードはバッテリーターミナルとして機能するラベルVBATの2個の銅パッドを備えています(→14ページの図5)。これによってバッテリーの接続に最大の融通性がもたらされます。



### **注意**

#### **極性と電圧の確認**

VRmCUEO3インターフェースボードの四角形の形状のパッドはバッテリーの正端子(+3V)に接続しなければなりません。丸い形状のパッドはバッテリーの負端子(GND)に接続しなければなりません。

### **外部バッテリーの接続:**

1. 1x2ピンヘッダー(2.54mm)、2本のワイヤ、または他の適切なタイプの接続を使用します。必要な場合はボードのバッテリーパッドにそれをはんだ付けします (→14ページの図5)。
2. 極性を確認して任意のタイプの3Vバッテリーを接続します。

## 5.2.10 電源

付属品キットにあるAC電源アダプターPHIHONG PSA 15R-050Pを使用します。代わりには同じ出力値(5 V DC、15 W)の電源を使用します。

### カメラを電源に接続:

1. 電源ケーブルをカメラの5VDCというラベルのある4ピン電源入力に接続します (→14ページの図5)。
2. AC電源アダプターをコンセントに接続します。
3. ACアダプターケーブルをカメラの電源ケーブルに接続します。
  - ▶ カメラが起動します。
4. ステータスLEDが点灯するのを待ちます (→14ページの図5)。
  - ▶ カメラは使用の用意ができます。

## 5.3 VRmEIO3ボードの接続

カメラにアクセスするための最低条件は、機能するイーサネット接続と電源です。また、アプリケーションに応じて、他のインターフェース、たとえばシリアル接続、S-Video出力、トリガー/ストロボ、I/Oが必要になる場合があります。

### 備考

以下の説明はVRmEIO3付属品キットを注文してあるものとします(→セクション3.4.2)。→詳しいピン配列はセクション11.4にあります。

### 5.3.1 標準コネクタの接続

#### イーサネットへの接続:

1. カメラのイーサネットポートに付属のイーサネットケーブルを接続します (→15ページの図6)。
2. イーサネットケーブルをローカルエリアネットワークに接続します。

### 備考

イーサネットのセットアップの説明は→セクション6.1にあります。

### **HDMIまたはS-Videoの接続:**

1. 付属のHDMI/S-VideoケーブルをカメラのHDMI/S-Videoレセプタクルに接続します (→15ページの図6)。
2. 付属のS-VideoケーブルをカメラのS-Videoレセプタクルに接続します (→15ページの図6)。
3. カメラのDVI/S-Video出力を見るには、DVI/S-Videoケーブルのもう片方を出力デバイスに接続します。

### **USBの接続:**

#### **備考**

#### **カメラのUSBコントロールの機能**

カメラのUSBポートのUSB0とUSB1は初期設定でUSBホストとして設定されています。USBコントロールの機能 (ホストまたはデバイス) を変更するには→セクション8.2を参照してください。

1. USBデバイスをカメラに接続するには、両側にUSB Aプラグのあるケーブルでカメラを接続します。初期設定では両方のUSBインターフェースはホストに設定されています。
2. カメラをUSBデバイスとしてUSBホストに接続するには、カメラのUSBポートの1つをデバイスとして設定します。それからカメラをUSBホストに接続します。

### **オーディオの接続:**

VRmEIO3ボードは、2つのデジタルオーディオ出力と3つのアナログオーディオ入力/出力を備えています(→15ページの図6)。デジタルまたはアナログオーディオのどちらかを使用できますが、同時に使用できるというわけではありません。

1. オーディオを使用できるようにするには、OPT2のディップスイッチ1をOFFに設定します (→18ページの図9)。
2. OPT2のディップスイッチ2をそれぞれONまたはOFFに設定することによって、デジタルかアナログオーディオを選択します (→18ページの図9)。
3. 必要に応じてオーディオケーブルを接続します (デジタル: TOSLINKかシンチ、アナログ: 3.5mm電話コネクタ)。

### シリアルポートの接続

VRmEIO3ボードは、2つのオスシリアルDE-9 D-Subポートを備えています(→15ページの図6)。上側ポート(UART0)は、RS232を介してカメラのシリアルコンソールにアクセスするのに使用されます。下側ポート(UART1)は、RS232またはRS422/RS485を介してカメラと外部デバイスとのシリアル通信に使用することができます。

1. RS232を介してカメラのシリアルコンソールにアクセスするには、付属のマルチモデムケーブルをUART0コネクタに接続します (→15ページの図6)。ケーブルのもう片方を付属のUSBシリアルアダプターを使用してホストPCに接続します。
2. シリアル接続を使用して外部デバイスをカメラに接続する場合は、OPT2のディップスイッチ5をOFFに設定してUART1を有効にします。それからOPT2のディップスイッチ7をそれぞれONかOFFに設定することによってRS485かRS232を選択します (→18ページの図9)。それからシリアルケーブルをUART1ポートに接続します。

### 5.3.2 ピンヘッダーとSATAの接続



#### 警告!

**装置が破損する可能性があります**

カメラが動作している間に、ピンヘッダーまたはSATAインターフェースの接続や切断をしないでください! どのピンもショートさせないでください!

ピンヘッダーまたはSATAコネクタでコンポーネントの接続または切断を行う前に、カメラに電源が入っていないことを確認してください。



#### 注意

**電力が不足する可能性があります**

ピンヘッダーの供給ピンは最大20mAの電流を送ります。この値を超えないようにしてください。さもなければ、他のカメラコンポーネントが十分な電力を受け取れなくなるか、AC電源アダプター自体が電源を切断する場合があります。外部デバイスを供給ピンに接続しないでください。

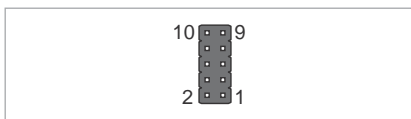


図12: ピンヘッダーのピンの順序。ピン1はボード上にラベルがあります。

### **JTAGの接続:**

互換性のあるJTAGエミュレータ(XDS510/560)を直接接続するのに20ピンヘッダーを使用することができます。I/O電圧は3.3 Vです。

1. TI20ピンコネクタを使用して、JTAGとラベルされた20ピンヘッダーのピン1をコネクタのピン1に合わせます (→16ページの図7)。
2. ピンヘッダーにメスソケットを押し込みます。
3. JTAGエミュレータを接続します。

### **他のピンヘッダーの接続:**

1. ピンヘッダー (2x5、2x20、1x4) に合うコネクタのある付属のケーブルを選びます。どのコネクタもピン1には小さい矢印▼が付いています。
2. ボードでは、どのピンヘッダーのピン1にも数「1」とラベル付けされています。
3. ピンヘッダーのピン1とケーブルコネクタの三角印▼を合わせます。
4. ピンヘッダーにコネクタを押し込みます。
  - ▶ 2 x 5ピンヘッダー: 茶色の線はピン1に接続されます。
  - ▶ 2 x 20ピンヘッダー: 赤色の線はピン1に接続されます。
5. 必要に応じてケーブルのシングルピンコネクタを接続します。
6. すべてのピンヘッダーの詳細なピン配列については→セクション11.4を参照してください。いくつかのピンヘッダーはディップスイッチOPT1とOPT2を使用して設定することができます(→18ページの図9)。

### **SATAデバイスの接続:**

1. 標準のSATAケーブルを使用して、SATAデバイスをカメラのSATAコネクタに接続します (→16ページの図7)。



### 5.3.3 電源の接続

付属品キットにあるAC電源アダプターPHIHONG PSA 15R-050Pを使用します。代わりには同じ出力値(5 V DC、少なくとも15 W)の電源を使用します。

#### カメラを電源に接続:

1. AC電源アダプターをコンセントに接続します。
2. ACアダプターケーブルをカメラの電源入力に接続します (→16ページの図7)。
  - ▶ カメラが起動します。
3. ステータスLEDが点灯するのを待ちます (→17ページの図8)。
  - ▶ カメラは使用の用意ができます。

## 6 第1ステップ

### ■ 備考

このマニュアルのLinuxコマンドにはルートユーザー権限を必要とするものがあります。ルートとしてログインしていない場合は、**sudo <command>**をタイプしてください。プレフィックス**sudo**によってLinuxユーザーパスワードの入力後にルート権限でシングルコマンドを実行できます。

### 6.1 イーサネット接続のセットアップ

#### DHCPサーバーのあるネットワーク

カメラは、DHCPサーバーからIPアドレスを取得するように設定されます。カメラはDHCPサーバーのあるネットワークに接続した場合は、イーサネット接続は動作するはずですが、→セクション6.1.2に進んでください。

#### DHCPサーバーのないネットワーク

DHCPサーバーを検出できない場合、カメラはIPv4LLアドレスへ戻ろうとします。この場合、カメラをコンピュータのネットワークインターフェース(2点間接続)に直接接続します。これにはホストコンピュータもIPv4LLアドレス(169.254.x.x)を必要とします。→セクション6.1.1に進んでください。

### ■ 備考

カメラにRS232を介してログインすることは常に可能です(→セクション 6.2.2)。こうすることによって、特定のネットワークインフラストラクチャに関連した問題を特定して修正することができます。

#### 6.1.1 2点間接続のセットアップ

### ■ 備考

このセクションは、ネットワークがDHCPサーバーを備えていない場合にのみ適用されます。

#### 2点間イーサネット接続のセットアップ:

1. ホストPCで**avahi-autoipd -D eth[n]**を実行して、169.254.x.xの範囲から適切なIPアドレスを取得します。
2. **Ifconfig**で現在のIPアドレスをチェックします。

## 6.1.2 イーサネット接続のチェック

イーサネット接続をチェックするには、VRmagic CamLabアプリケーションを使用します。CamLabはすべてのカメラパラメーターへのアクセスが可能なGUIアプリケーションです。

### VRmagic CamLabでのイーサネット接続のチェック:

1. Ubuntu LinuxホストPCデスクトップの**CamLab**アイコンをダブルクリックするか desktop or by executing camlab in the **Terminal**アプリケーションでcamlabを実行してCamLabアプリケーションを開始します。
  - ▶ カメラのステータスLEDが点灯すると(→セクション3.3)、カメラは**デバイス**リストに表示されます(モデル名とシリアルナンバー)。同じネットワークに他のVRmagicカメラが接続されている場合は、それらもリストに表示されます。

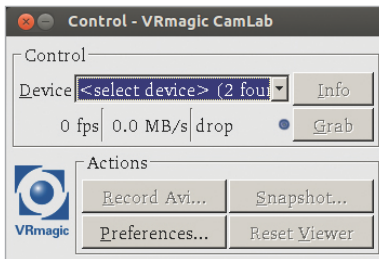


図13: CamLabウィンドウ

2. まだの場合は、リストからカメラを選択します。
  - ▶ カメラは接続されて使用の用意ができます。

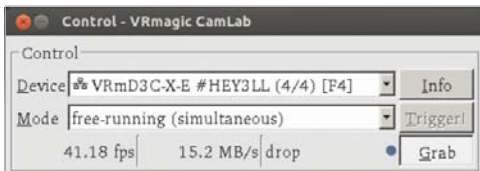


図14: CamLabで選択されたカメラ

3. カメラ画像を見るには**Grab**を選択します。
4. 以下のステップでは**CamLab**を開いたままにします。

## 6.2 ターミナルでのデバイスのアクセス

### 6.2.1 SSHを介したアクセス

CamLabでカメラを検出したら、SSH接続を使用してカメラにアクセスすることができます。これを行うためにLinuxターミナルアプリケーションを使用できます。けれどもPuTTYの使用を推奨します。これは、オープンソースSSH、Telnet、シリアルクライアントです。PuTTYは、VRmagic Custom Ubuntuに事前にインストールされて設定されています(→セクション4.2.1)。

#### カメラのIPアドレスの検出:

1. カメラのIPアドレスを見るにはCamLabの**info**ボタンをクリックします (→43ページの図14)。
  - ▶ デバイス情報ウィンドウにカメラのIPアドレスが表示されます。

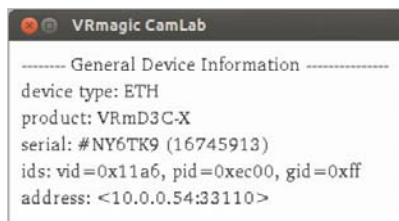


図15: Device Infoウィンドウ

#### PuTTYを使用したSSHを介したデバイスへのアクセス:

1. ターミナルでputtyを実行するかランチャーを使用してホストPCでPuTTYを開始します。
2. **Session**カテゴリーで**Connection type**に**SSH**を選択します。
3. カメラのIPアドレスをタイプして**Open**を選択します。
  - ▶ コンソールウィンドウが表示されてlogin as: と促されます。
4. **root**としてログインしてパスワードとして**vrmagic**を入力します。パスワードを変更した場合は、新しいパスワードを入力します。
  - ▶ これでSSHを介してカメラに接続されます。

```
root@VRmD3C-X-E-HEY3LL: ~  
login as: root  
root@10.0.0.185's password:  
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 2.6.37-VRm-1.06 armv7l)  
  
* Documentation:  https://help.ubuntu.com/  
Last login: Mon Jun 12 23:18:52 2000 from vrmcp27.vrm.local  
VRmD3C-X-E-HEY3LL:~# █
```

図16: PuTTYコンソールを使用したSSH接続

### Linuxターミナルを使用したSSHを介したデバイスへのアクセス:

1. ホストPCでターミナルウィンドウを開きます。
2. `ssh root@<IP address>`を実行します。
  - ▶ コンソールは`root@<IP address>`のパスワードを促します:
3. パスワードとして`vrmagic`を入力します。パスワードを変更した場合は、新しいパスワードを入力します。
  - ▶ これでSSHを介してカメラに接続されます。

## 6.2.2 シリアル通信(RS232)を介したアクセス

ほとんどのアプリケーションでは、SSHを介した接続で十分です。シリアルコンソールは、デバッグのためとVRmagicアップデートSDカードを使用したアップデートにのみ推奨します。

### 備考 要件

- カメラのシリアルコンソールは、付属のヌルモデムケーブルを使用してホストPCに接続されなければなりません。ホストPCとの接続にはUSBシリアルアダプターの使用を推奨します。USBシリアルアダプターはVRmEIO3アクセサリキットに含まれています。
- シリアルターミナルアプリケーションはホストPCにインストールしなければなりません。VRmagic Custom Ubuntuを使用している場合は、PuTTYはすでにインストールされています。

### *PuTTYを使用したRS232を介したデバイスへのアクセス:*

1. カメラのシリアルコンソールをホストPCに接続します (→第5章)。
2. ホストPCでシリアルターミナルソフトウェアを開始します。PuTTYの使用を推奨します。
3. VRmagic Custom Ubuntuをインストールした場合は、シリアルUSBアダプターを使用したシリアル接続の設定はすでにPuTTYに格納されています。**Session**カテゴリで**Serial@USB0**セッションを読み出すだけです。さもないければ、PuTTYの**Connection > Serial**カテゴリで以下の設定を入力します。
  - Serial line: /dev/ttyUSB0 (serial-to-USB adapter) または /dev/ttyS1 (serial interface)
  - Speed (baud): 115200
  - Data bits: 8
  - Stop bits: 1
  - Parity: none
  - Flow control: none
4. **Session**カテゴリで、**Connection type**として**Serial**を選択します。
5. **Open**をクリックしてセッションを開始してシリアルコンソールを開きます。
6. ログインプロンプトを取得するために<Enter>キーを押します。
7. rootとしてログインしてパスワードとして**vrmagic**を入力します。
  - ▶ これでカメラのシリアルコンソールに接続されます。

## 6.3 カメラの初期設定パスワードの変更

カメラにアクセスする初期設定パスワードは“vrmagic”です。カメラの初期設定パスワードの変更を強く推奨します。

### *カメラのパスワードの変更:*

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. パスワードを変更するにはコマンド**passwd**をタイプして<Enter>を押します。
  - ▶ 新規パスワードを指定するように促されます。
3. 新規パスワードを入力して<Enter>を押します。
4. 新規パスワードを反復して<Enter>を押します。
  - ▶ これでパスワードが変更されます。

## 6.4 カメラのIPアドレスの変更

DHCPを使用してダイナミックIPアドレスで初回にカメラを起動した後に、カメラのネットワークコンフィギュレーションを変更することによって静的IPアドレスに切り替えることができます。

### 備考

カメラに静的IPアドレスを割り当てない場合は、カメラの電源を入れるたびにDHCPサーバーは異なったIPアドレスをカメラに割り当てる場合があります。IPアドレスは常にCamlabで確認できます(→セクション6.2.1)。

#### ネットワークコンフィギュレーションの変更:

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. `cd /etc/network/`とタイプして/etc/network /ディレクトリをたどります。
3. `nano interfaces`とタイプしてNanoテキストエディタでinterfacesファイルを開きます。
4. ファイルを編集して、望むようにコンフィギュレーションを変更します。また、イーサネットコンフィギュレーションは"`auto eth0`"をコメントアウトして完全に無効にすることができます。詳しくは<https://wiki.debian.org/NetworkConfiguration>を参照してください。
5. `<Ctrl>+<S>`でファイルを保存します。
6. `<Ctrl>+<X>`でエディタを終了します。

## 6.5 ファイルとデータの交換

カメラとホストコンピュータ間でファイル交換をするには、ネットワークシェア、`scp`コマンド、様々なデータメディアなどの方法があります。

### 6.5.1 NFSまたはSamba/Windows Shares

**NFSまたはSamba/Windows sharesのカメラへのマウント:**

1. ホストコンピュータにネットワークシェアを作成して、特定のユーザーのアクセス権 (ユーザー名とパスワード) を取得します。ステップ3でユーザーデータが必要になります。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. ネットワークシェアがマウントされるカメラにマウントディレクトリを作成します。`mkdir`コマンドを使用して新規ディレクトリを作成します。たとえば`mkdir my_mount_dir`とタイプして現在のディレクトリの中に新規ディレクトリ`"my_mount_dir"`を作成します。
4. シェアをマウントするには以下のようにタイプします:

NFS:

```
mount <ip-address>:<name-of-share> /<local-mnt-dir>
```

Samba/Windows:

```
mount //<name-of-host または ip-address>/<name-of-share> /<local-mnt-dir> -o username=<name>
```

▶ ユーザーネームおよび/またはパスワードを入力するように促されます。

5. ログインデータをタイプします。
  - ▶ シェアがマウントされます。`ls /<local-mnt-dir>`とタイプして内容を見ることができします。

説明と例

|                                    |                              |
|------------------------------------|------------------------------|
| <code>&lt;ip-address&gt;</code>    | ネットワークシェアを含むホストコンピュータのIPアドレス |
| <code>&lt;name-of-host&gt;</code>  | ネットワークシェアを含むホストコンピュータの名称     |
| <code>&lt;name-of-share&gt;</code> | ネットワーク上で共有されるホストディレクトリの名称    |
| <code>&lt;local-mnt-dir&gt;</code> | シェアをマウントしたいカメラ上のディレクトリ       |
| <code>&lt;name&gt;</code>          | シェアにアクセスするのに必要なユーザー名         |



## 6.5.2 scpコマンド

scpコマンドによって、ファイルとディレクトリをホストPCとカメラの間でコピーすることができます。これはSSHを使用するので、SSHと同じ認証を必要とします。

### 備考

#### ホストPCとカメラでのscpの使用

ホストPCでscpコマンドを使用するには、カメラでSSHサーバーが動作していないければなりません。初期設定ではSSHサーバーはカメラですでに動作しています。

#### ホストPCからのscpコマンドの使用PC:

1. ホストPCでコマンドシェルを開始します。
2. ローカルホストPCからリモートカメラにファイルをコピーするには、以下を実行します。

```
scp <FILE_NAME>  
<D3_USERNAME>@<D3_IP_ADDRESS>:/some/remote/  
directory
```

3. リモートカメラからローカルホストPCにファイルをコピーするには、以下を実行します。

```
scp <D3_USERNAME>@<D3_IP_ADDRESS>:<FILE_NAME>  
/some/local/directory
```

4. ローカルホストPCからリモートカメラにディレクトリ（反復）をコピーするには、以下を実行します。

```
scp -r /some/local/directory  
<D3_USERNAME>@<D3_IP_ADDRESS>:/  
some/remote/directory
```

### 6.5.3 MicroSDカード

#### *microSD*カードのマウント:

1. カメラの*microSD*カードスロットにSDカードを挿入します (→16ページの図7)。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. `cd /dev`とタイプして/devディレクトリに変更します。
4. `ls -l`とタイプしてディレクトリの内容をリストします。
  - ▶ SDカードは*mmcblk1*とリストされるはずですが。
  - ▶ SDカードの各パーティションは*mmcblk1p<X>*とリストされるはずですが。(X = パーティションナンバー)。
5. `mkdir`でローカルマウントディレクトリを作成します。例:
 

```
mkdir /mnt/sd-card
```
6. 以下のようにタイプしてSDカードをローカルマウントディレクトリにマウントします。
 

```
mount /dev/mmcblk1p<X> /mnt/sd-card (X = パーティションナンバー)
```

  - ▶ これでローカルマウントディレクトリでSDカードにアクセスできます。

### 6.5.4 USBフラッシュドライブ

#### *USB*フラッシュドライブのマウント:

1. カメラのUSBポートの1つにUSBフラッシュドライブを差し込みます。ポートはUSBホストポートとして設定されなければなりません(初期設定)。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. `cd /dev`とタイプして/devディレクトリに変更します。
4. `ls -l`とタイプしてディレクトリの内容をリストします。
  - ▶ USBフラッシュドライブは*sda*とリストされる場合があります。
  - ▶ USBフラッシュドライブの各パーティションは*sda<X>*とリストされるはずですが。(X = パーティションナンバー)。
5. `mkdir`でローカルマウントディレクトリを作成します。例:
 

```
mkdir /mnt/usb
```
6. 以下のようにタイプしてUSBフラッシュドライブをローカルマウントディレクトリにマウントします。
 

```
mount /dev/sda<X> /mnt/usb (X = partition number)
```

  - ▶ これでローカルマウントディレクトリでUSBフラッシュドライブにアクセスできます。

## 6.6 CamserverとCamLab

Vrmcamserverと呼ばれるプロセスがカメラで動作しています。Vrmcamserverはイーサネット上のAPIトンネルで、ホストPC上でのカメラアプリケーションの開発とデバッグを可能にします。またこれはVRmagic CamLabアプリケーションが、画像の転送とイーサネット上のカメラのコントロールに使用します。

### i 備考

Camserverは開発ツールです。Camserverが動作しているかぎり、カメラは同じローカルエリアネットワーク内でCamLabアプリケーションが動作しているだけからもあるいはカスタムvrmusbcamアプリケーションからもアクセスできます。カメラが生産オペレーション内にある場合はCamserverを無効にすることを推奨します。

予想どおりに動作しない場合はログファイルを見るのが役立ちます (/tmp/vrmcamserver.log)。

以下のステップではカメラへのターミナル接続が必要です(→セクション6.2)。

#### カメラへのアクセスの制限:

1. 特定のホストだけにカメラへのアクセスを許可するには、カメラのファイルシステム上の/etc/vrmagicディレクトリへ移ります。
2. nano vrmcamserver.iniとタイプしてvrmcamserver.iniファイルをNanoテキストエディタで編集します。
3. その後の指示についてはファイル内のコメントを見てください。

#### camserverプロセスの一時的な無効化:

1. プロセスを一時的に無効にするにはstop vrmcamserverとタイプします。
  - ▶ 再開するか次の起動までプロセスは停止します。
2. camserverプロセスを再開するにはstart vrmcamserverとタイプします。

#### camserverプロセスの永続的な無効化:

1. カメラのファイルシステム上の/etc/initディレクトリへ移ります。
2. vrmcamserver.confとタイプしてvrmcamserver.confファイルを編集します。事前に元のvrmcamserver.confファイルのバックアップを作成することを推奨します。
3. 初期設定では、camserverプロセスは起動時に開始してシャットダウン時に以下のラインで停止します:  
start on runlevel [2345]  
stop on runlevel [016]

4. 起動時の自動開始を無効にするにはLinux runlevelsを編集します。最初のラインを削除かコメントアウトして次のラインを編集します:  

```
# start on runlevel [2345]
stop on runlevel [0123456]
```
5. ファイルを保存して、変更が適用されるようにカメラを再起動します。

## 6.7 カメラのGPIOへのアクセス



### 備考

以下の表は利用可能なGPIOの概要です。→すべてのGPIOの詳しいピン配列については第11章を参照してください。

#### 利用可能なGPIO

| インターフェースボード | GPIOナンバー  | コネクタ名      |
|-------------|---|------------|
| VRmCUEO3    | 25, 26, 27  | GPIO       |
| VRmEIO3     | 25, 26, 27 (muxed with McASP audio)<br>DIP switch 1 of OPT2: ON for GPIOs | muxed GPIO |
|             | 0, 1 (muxed with CAN bus)<br>DIP switch 4 of OPT2: ON for GPIOs           | muxed GPIO |
|             | 36, 37 (muxed with UART1)<br>DIP switch 5 of OPT2: ON for GPIOs           | muxed GPIO |
|             | 92 ... 118 (muxed with RGB888)<br>DIP switch 8 of OPT1: ON for GPIOs      | VIDEO I/O  |
|             | 136 ... 143 (via GPIO expander)   | GPIO_0     |
|             | 152 ... 159 (via GPIO expander)   | GPIO_1     |

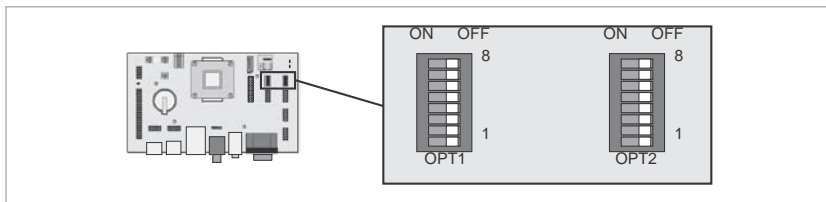


図17: VRmEIO3インターフェースボード上のディップスイッチOPT1とOPT2のレイアウト

### カメラのGPIOへのアクセス:

Linuxコマンドラインからgpio-sysfsインターフェースを使用してGPIOピンにアクセスすることができます。GPIOにアクセスする前に、GPIOをユーザースペースにエクスポートしなければなりません。

1. ターミナルを使用してカメラにアクセスします (→セクション 6.2)。
2. 以下のようにタイプしてアクセスしたいGPIOをuserspaceにエクスポートします。  
`echo <gpio_no> > /sys/class/gpio/export`

出力25の例: `echo 25 > /sys/class/gpio/export`

3. 出力にアクセスする場合、出力をhigh = 1またはlow = 0に設定することができます。  
0に設定: `echo 0 > /sys/class/gpio/gpio<GPIO_NO>/value`  
1に設定: `echo 1 > /sys/class/gpio/gpio<GPIO_NO>/value`

出力25の例: `echo 1 > /sys/class/gpio/gpio25/value`

4. 入力にアクセスする場合、まずdirectionを"in"に設定します:  
`echo in > /sys/class/gpio/gpio<GPIO_NO>/direction`

入力のステートを読み出すには以下のようにタイプします。

`cat /sys/class/gpio/gpio<GPIO_NO>/value`

入力25の読み出しの例: `cat /sys/class/gpio/gpio25/value`

5. VRmEIO3評価用インターフェースボード上 (GPIOs 148 ... 151) の4つのユーザーLED (ULED1 ... ULED4) をコントロールするには以下のようにタイプします。

`echo 1 > /sys/class/gpio/gpio<GPIO_NO>/value` (LED on)

`echo 0 > /sys/class/gpio/gpio<GPIO_NO>/value` (LED off)

例(LED on): `echo 1 > /sys/class/gpio/gpio151/value`

### 備考

以下の入力電圧レベルはローまたはハイになります。

GPIO 25 ... 118    0 ... 0.8 V: low (zero)            2.0 ... 3.3 V: high (one)

GPIO 136 ... 159    0 ... 1.5 V: low (zero)            3.5 ... 5.0 V: high (one)

## 6.8 D3への追加ソフトウェアのインストール

追加ソフトウェアをダウンロードするにはインターネット接続が必要です。カメラがインターネットアクセスのあるネットワークに接続されている場合はすでに用意ができていることとなります。

### 6.8.1 VRmagicリポジトリの有効化

たとえばHALCONサポートパッケージの.NETラッパーなどのさらなるVRmagicソフトウェアをD3カメラにインストールすることができます。ソフトウェアをインストールするにはVRmagicリポジトリをD3で有効にする必要がありますが、これについては以下で説明します。

#### ■ 備考

##### D3シリアルナンバーとパスワード

以下の手順ではカメラのシリアルナンバーと対応するパスワードが必要です。いずれもカメラに添付されています。

##### *D3カメラでのVRmagicリポジトリの有効化:*

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. 以下を実行してvrmagic.listファイルをNanoテキストエディタで編集します。  
`nano /etc/apt/sources.list.d/vrmagic.list`
3. 以下のラインをコメントアウトします:  
`deb http://<SERIAL>:<PASSWORD>@www.vrmagic.com/packages/  
imaging/repository1/linux/debian precise contrib`
4. <SERIAL>を使用するカメラの6桁のD3カメラシリアルナンバーと置き換えます。
5. <PASSWORD>を対応するパスワードに置き換えます。
6. <Ctrl>+<S>でファイルを保存します。
7. <Ctrl>+<X>でエディタを終了します。

## 6.8.2 サードパーティのソフトウェア

D3カメラは通常のUbuntu Linuxオペレーティングシステムを使用しているため、**aptitude**や**apt-get**のようなパッケージ管理システムを使用して、Ubuntu Linuxで動作するカメラにどんな種類のソフトウェアもインストールすることができます。



### **apt-get**によるD3カメラへの追加ソフトウェアのインストール:

1. ターミナルを使用してカメラにアクセスします (→セクション 6.2)。
2. **apt-get update**を実行してパッケージリストをアップデートします。
3. **apt-get install <package\_name>**で望みのパッケージをインストールします。

→さらに詳しくはパッケージ管理システムのドキュメントを参照してください。

## 6.9 トリガーとストロボの使用法

インターフェースボードに応じて、トリガーとストロボ用のピンのあるコネクタは異なります。以下の表はトリガーとストロボ用のコネクタの違いと対応するピン配列を示しています。コネクタの位置がよくわからない場合は→セクション3.3を参照してください。

|  <b>10ピンヘッダー</b><br>VRmEIO3インターフェースボードを備えたカメラ |                           |  <b>JST BM04B-NSHSS-TBT</b><br>VRmCUEO3インターフェースボードを備えたカメラ |                       |
|--|---------------------------|--|-----------------------|
| ピン   | シグナル                      | ピン   | シグナル                  |
| 1  | GND                       | 1  | + 3.3 V               |
| 2  | パッシブトリガー In - (3...24 V)  | 2  | アクティブトリガー In (3.3 V)  |
| 3  | アクティブストロボ Out (5 V)       | 3  | アクティブストロボ Out (3.3 V) |
| 4  | パッシブトリガー In + (3...24 V)  | 4  | GND                   |
| 5...6  | +5.0 V                    |  |                       |
| 7  | パッシブストロボ Out + (3...24 V) |  |                       |
| 8  | アクティブトリガー In (5 V)        |  |                       |
| 9  | パッシブストロボ Out - (3...24 V) |  |                       |
| 10   | GND                       |  |                       |

## 6.9.1 トリガー入力

### パッシブトリガー入力(無電位)

パッシブトリガー入力は正電圧(3...24 V)を受け入れます。電圧が3Vを超えるとシグナルは論理的にハイと解釈され、電圧が1V未満まで低下すると論理的にローと解釈されます。TriggerIn + とTrigge In - の間の電流は3.3Vで約2mA、24Vで約5mAです。アクティヴトリガー入力とパッシブトリガー入力は内部的に接続されて、別々にコントロールすることはできません。

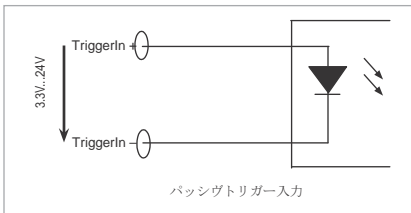


図18: パッシブトリガー入力

### アクティヴトリガー入力(TTL)

アクティヴトリガー入力の場合はGNDに対してLV TTL/ TTL互換シグナルを供給することができます。入力には内部プルダウン抵抗があり、入力電流は最大1mAです。



## 6.9.2 ストロボ出力

### パッシヴstroボ出力(無電位)

カメラは、無電位、光結合出力トランジスタを使用して外部電流をStrobeOut +とStrobeOut -の間に変換します。StrobeOut +とStrobeOut -の間の電圧は常に正です(3...24 V)。出力トランジスタは100mAまでのスイッチング電流が可能です。それに伴う電圧降下は1V以下に収まります。

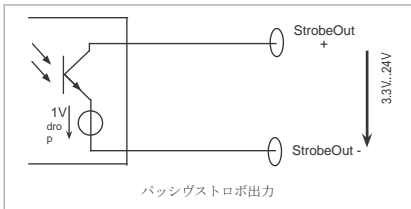


図19: パッシヴstroボ出力

### アクティヴstroボ出力(TTL)

アクティヴstroボ出力では、GNDに対してTTL互換stroボシグナルが生じます。アクティヴstroボ出力は、20mAまでのシンク電流あるいはソース電流が可能です。

## 6.10 カメラのOn/Offの切り替え

電源を切断する前にカメラの電源を切ることを推奨します。これで、内部のLinux OSは適切に停止するようになります。

### 備考

VRmCUEO3インターフェースボードを備えたカメラの場合は→セクション5.2.7に示したように外部の電源ボタンをつなぎます。代わりに、カメラのスイッチに電源を再接続します。

#### 電源ボタンを使用してカメラをonに切り替え:

1. カメラが電源に接続されていても電源が切られている場合は、電源ボタンを押してカメラに電源を入れます (→セクション3.3)。
  - ▶ カメラは自動的に起動します。ステータスLEDが点灯するとすぐにカメラは準備ができています。

#### 電源を接続してカメラをonに切り替え:

1. カメラが電源に接続されていない場合は、カメラを電源に再接続します。
  - ▶ カメラは自動的に起動します。ステータスLEDが点灯するとすぐにカメラは準備ができています。

#### カメラをoffに切り替え:

1. SSHまたはRS232を介してカメラにアクセスしている場合は、`halt -p`とタイプしてカメラをシャットダウンします。
  - ▶ 数秒後にカメラはシャットダウンします。ステータスLEDは消えます。
2. 別の選択肢はCamLabを使用することです。カメラを選択して**Misc**タブの**Shutdown**ボタンを押します。
  - ▶ 数秒後にカメラはシャットダウンします。ステータスLEDは消えます。
3. 代わりに、カメラのインターフェースボードで電源ボタンを押すとカメラは簡単にシャットダウンします。
  - ▶ 数秒後にカメラはシャットダウンします。ステータスLEDは消えます。

## 7 アプリケーションの開発

### 備考

プログラミング環境はLinux Ubuntu 12.04 LTSシステム上でセットアップすることを推奨します。

### ディレクトリ構造とデモアプリケーションのための最良の設定

問題を避けてより効率的に作業するために、パスネームに特殊文字（スペース文字、句読点、ウムラウトなど）を使用しないディレクトリ構造にコンポーネントをインストールすることを推奨します。下線文字 `_` の使用は問題ありません。

### 7.1 デモアプリケーションの解凍

#### 備考

Ubuntu LinuxホストPCの/opt/vrmagic/sdk-\*/D3/development\_kitディレクトリにはARMとDSP (\* = SDKバージョン)のデモコードのある2つのzipファイルが含まれています。

ARMデモ:                   vrmagic-linux-d3-armhf-demos-src-\*.zip

DSPデモ:                   vrmagic-linux-d3-armhf-dsp-demos-src-\*.zip

両方のファイルを同じディレクトリに解凍できます。ディレクトリ構造は合わせられます。

#### 備考

#### 事前設定のEclipseプロジェクト

ディレクトリarmとarm-dspのすべてのデモは事前設定のEclipseプロジェクトを含んでいます。それらは**File > Import > Existing Projects into Workspace**を介してeclipseワークスペースにインポートできます。ルートディレクトリをvrm-demos-\*ディレクトリに設定するには**Browse**ボタンを使用します。利用可能なすべてのeclipseプロジェクトがプロジェクトリストに表示されます。**finish**をクリックしてすべてをインポートするか、チェックボックスを使用して特定のプロジェクトをインポートします。プロジェクトはEclipseのプロジェクトエクスプローラーに表示されます。

## 7.1.1 ARMデモとC++ APIラッパー

### ARMデモの解凍:

1. ホストPCでコマンドシェルを開始します。
2. ホームディレクトリなどの選択したディレクトリにデモコードを解凍するには、そのディレクトリを指定します。
3. 以下を実行して現在のディレクトリにデモのあるzipファイルを解凍します。  
`sudo unzip /opt/vrmagic/sdk-<SDK_VERSION>/D3/development_kit/vrmagic-linux-d3-armhf-demos-src-<SDK_VERSION>.zip`  
 ▶ 以下のディレクトリ構造が作成されます。

| vrm-demos-d3-*/ (* = SDKバージョンナンバー)   |                  |  |            |        |  |     |  |  |            |       |     |  |  |  |
|--|------------------|--|------------|--------|--|-----|--|--|------------|-------|-----|--|--|--|
| <b>wrapper</b>   | C++ ラッパー         |  |            |        |  |     |  |  |            |       |     |  |  |  |
| <b>demos</b>   | API使用のデモアプリケーション |  |            |        |  |     |  |  |            |       |     |  |  |  |
| <table border="1"> <tr> <td rowspan="4"><b>all</b></td> <td></td> <td><b>vc#</b></td> <td>.NETデモ</td> </tr> <tr> <td></td> <td>...</td> <td></td> </tr> <tr> <td></td> <td><b>arm</b></td> <td>ARMデモ</td> </tr> <tr> <td>...</td> <td></td> <td></td> </tr> </table> | <b>all</b>       |  | <b>vc#</b> | .NETデモ |  | ... |  |  | <b>arm</b> | ARMデモ | ... |  |  |  |
| <b>all</b>   |                  |  | <b>vc#</b> | .NETデモ |  |     |  |  |            |       |     |  |  |  |
|  |                  |  | ...        |        |  |     |  |  |            |       |     |  |  |  |
|  |                  |  | <b>arm</b> | ARMデモ  |  |     |  |  |            |       |     |  |  |  |
|  | ...              |  |            |        |  |     |  |  |            |       |     |  |  |  |

## 7.1.2 DSPデモ

### DSPデモの解凍:

1. ホストPCでコマンドシェルを開始します。
2. ホームディレクトリなどの選択したディレクトリにデモコードを解凍するには、そのディレクトリを指定します。
3. 以下を実行して現在のディレクトリにデモのあるzipファイルを解凍します。  
`sudo unzip /opt/vrmagic/sdk-<SDK_VERSION>/D3/development_kit/vrmagic-linux-d3-armhf-dsp-demos-src-<SDK_VERSION>.zip`  
 ▶ 以下のディレクトリ構造が作成されます。

| vrm-demos-d3-*/ (* = SDKバージョンナンバ) |                  |
|-----------------------------------|------------------|
| <b>demos</b>                      | API使用のデモアプリケーション |
| <b>arm-dsp</b>                    | ARMおよびDSPデモ      |
| <b>sobel_viewer_d3</b>            |                  |
| <b>sobel_viewer_d3_imglib</b>     |                  |

## 7.2 ARMデモのコンパイルと実行

ARMデモはVRmagic APIの使用法を示しています。→セクション7.1.1に示したステップを実行した場合は、作成した新規デモディレクトリにARMデモがあります: /<path-to-your-demo-dir>/demos/arm。

VRmagic D3プラットフォーム用のアプリケーションをコンパイルする最も簡単な方法はUbuntu 12.04 LTSによって提供されているクロスコンパイラツールチェーンを使用することです。

以下のデモが含まれています:

- **deviceinfo**: 接続されたすべてのVRmagicデバイスのプロパティをリストします。
- **restorefactorydefaults**: 接続されたすべてのVRmagicカメラの設定をすべて初期設定にリセットします。
- **sdlviewer**: SDLライブラリを使用してカメラの現在設定されているビデオ出力(Sビデオ、HDMI、LCD)のカメラのビデオ画像を表示します。
- **sdlviewer\_cpp**: ラッパーディレクトリ内にあるC++ラッパーを使用する以外はsdlviewerと同じです。
- **vm\_lib\_demo**: オブジェクト認識、プロプセグメント化などのアルゴリズムを含んでいるVM\_Lib画像処理ライブラリのデモです。

### LinuxホストPC上でのARMデモのコンパイル:

デモをコンパイルする前に、ARM EABI toolchainをインストールしたことを確認してください(→セクション4.2)。

1. 初期設定ではデモはD3プラットフォーム用にビルドされます。Linux X86ホストで動作するデモをビルドするにはファイルRules.makeを編集します。
2. すべてのデモをビルドするにはARMデモディレクトリに移ってmakeとタイプします。
  - ▶ ビルドプロセスが終了すると、対応するサブディレクトリ内に実行ファイルができます。
  - ▶ ビルドプロセスが動作しない場合は、コンソール上に表示されるエラーメッセージを読み出してください。
3. カメラ上のバイナリをテストするには、→セクション6.5にあるように以下のどれかを使用します。
  - ネットワーク上でデモディレクトリを共有して、カメラにマウントします。
  - scpコマンドを使用してカメラにデモをコピーします。
  - USBフラッシュドライブかmicroSDカードを使用してファイルを交換します。

### 備考

#### 事前にインストールされたデモ

デモとバイナリはD3カメラの以下のディレクトリにすでにインストールされています。  
~/vrm-demos-\*/ directory (\* = SDKバージョン)。

### D3上でのARMデモの実行:

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. アプリケーションを実行するには以下のようにタイプします。  
./path/to/executable/name\_of\_executable

すでに実行可能ファイルと同じディレクトリになっている場合は、以下のようにタイプします。

./name\_of\_executable

例: deviceinfoアプリケーションを実行するには以下のようにタイプします。

./path/to/demo\_directory/demos/arm/deviceinfo/deviceinfo

## 7.3 DSPデモのコンパイルと実行

DSPデモは、画像処理にコプロセッサとしてDSPを使用することによってARMアプリケーションを拡張する方法を示しています。→セクション7.1.2にあるステップを実行した場合は、作成した新規デモディレクトリにDSPデモがあります:

`</path-to-your-demo-dir>/demos/arm-dsp`。以下のデモが含まれています:

- `sobel_viewer_d3`: このデモはVRmagicの`vrmagic`を使用してカラー変換するのにDSPを試用します。ソーベルフィルターの例にユーザーが書き込んだDSPコードを追加して実行する方法も示しています。デモはCodec Engineフレームワークを使用します。
- `sobel_viewer_d3_imglib`: `sobel_viewer_d3`と機能は同じですが外部libにリンクします。このデモにはTI IMGLIBが必要です。以下からダウンロードできます。  
[http://software-dl.ti.com/dsp/dsp\\_public\\_sw/c6000/web/c64p\\_imglib/latest/exports/c64plus\\_imglib\\_2\\_02\\_00\\_00\\_Linux-x86\\_Setup.bin](http://software-dl.ti.com/dsp/dsp_public_sw/c6000/web/c64p_imglib/latest/exports/c64plus_imglib_2_02_00_00_Linux-x86_Setup.bin)。

各デモについて詳しくはそれぞれのデモディレクトリにあるREADMEファイルを参照してください。

### 備考

#### VRmagic D3 EZSDKとTI EZSDK

DSPデモのどれかをコンパイルする前に、LinuxホストPCにVRmagic D3 EZSDKとTI EZSDKをインストールして設定してあることを確認してください(→セクション4.3.4)。

#### Linux host PCでのDSPデモのコンパイル:

1. ホストPCで、ビルドしたいデモのディレクトリに変更します。例:
 

```
cd vrm-demos-d3-4.2.0/demos/arm-dsp/sobel_viewer_d3
```
2. デモをビルドするには**make**とタイプします。
  - ビルドが完了すると、サブディレクトリbinに2つの実行ファイルができます:
    - ARM実行ファイルはたとえば**sobel\_viewer**
    - DSP実行ファイルは後に**.xe674**が付いてたとえば**server.xe674**
  - ▶ ビルドがうまくいかない場合は、コンソールに表示されるエラーメッセージを見てください。
3. カメラ上のバイナリをテストするには、→セクション6.5にあるように以下のどれかを使用します。
  - ネットワーク上でデモディレクトリを共有して、カメラにマウントします。
  - `scp`コマンドを使用してカメラにデモをコピーします。
  - USBフラッシュドライブかmicroSDカードを使用してファイルを交換します。

## 備考

### DSPデモを実行する前にDSPサポートを有効にします

D3でDSPサポートを有効にするには、D3で1度`vrm-dsp-support enable`を実行します。これはCMEMカーネルモジュールが永続的に756 MB (0xC0000000 – 0xEFFFFFFF)を使用するように設定して、メモリの最高の256 MBをDSP用にします。DSPサポートを無効にするには、`vrm-dsp-support disable`を実行します。これはCMEMが1GB (0xC0000000 – 0xFFFFFFFF)を使用するように設定します。

### D3でのDSPデモの実行:

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. 実行ファイルのある`<demo_dir>/bin`ディレクトリに変更します。
3. ARM実行ファイルを実行するには、`./name_of_executable`とタイプします。
  - ▶ ARM実行ファイルが開始するとすぐにDSP実行ファイルがDSPに読み込まれて自動的に実行されます。

## 備考

### DSP実行ファイルの場所

DSP実行ファイルはARM実行ファイルを呼び出すのと同じディレクトリになければなりません。さもなければDSP実行ファイルは検出されません (ランタイムエラー "DspInit(): can't open engine")。VRmagic demo makefilesはARMおよびDSP実行ファイルを常に同じディレクトリ`<demo_dir>/bin`にビルドします。



## 7.4 .NETデモのコンパイルと実行

### 7.4.1 .NETサポートパッケージのインストール

C#デモをビルドして実行するには、まず以下の.NETサポートパッケージをLinuxホストPCとD3にインストールしなければなりません。

#### *LinuxホストPCへの必要なパッケージのインストール:*

以下のパッケージがPCにインストールされてビルドと実行ができるようになります:

C# demos: MonoDevelop (統合開発環境)、mono-gmcs  
(Mono C# 2.0 and C# 3.0 compiler for CLI 2.0)、vrm-usbcamnet。

1. Ubuntu LinuxホストPCでコマンドシェルを開始します。
2. `sudo apt-get update`を実行してパッケージリストをアップデートします。
3. MonoDevelopとmono-gmcsをインストールするには以下を実行します。  
`sudo apt-get install monodevelop mono-gmcs`
4. vrm-usbcamnetをクロスビルド環境にインストールするには以下を実行します。  
`sudo vrm-multiarch-armhf download_install --assume-yes vrm-usbcamnet`

#### *D3への必要なパッケージのインストール:*

以下ではVRmagic .NETラッパーとMonoランタイムがD3にインストールされます。これでD3上でC#と.NETに対応するようになります。

1. まだの場合はD3カメラでVRmagicリポジトリを有効にします (→セクション6.8.1)。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. `apt-get update`を実行してパッケージリストをアップデートします。
4. vrm-usbcamnet .NETラッパーパッケージをインストールするには以下を実行します。  
`apt-get install vrm-usbcamnet`

### 7.4.2 C#デモのコンパイルと実行

→セクション7.1.1にあるステップを実行した場合は、すでに.NETデモがインストールされています。作成したデモディレクトリにC#デモがあります:

`/<path-to-your-demo-dir>/demos/all/vc#`

以下のデモが含まれています:

- **exposuretrigger**: 最初のVRmagic取り込みデバイスから一連の画像を取り込んで現在のディレクトリに保存します。
- **viewer**: VRmUsbCam .NET API v2と.Net Viewerを使用するデモアプリケーションです。
- **xmldeviceinfo**: 接続されたデバイスの情報があるXMLファイルを生成するVRmUsbCam .NET API v2を使用するC#デモアプリケーションです。

#### **LinuxホストPCでのC#デモのコンパイル:**

1. MonoDevelopでビルドしたいデモのソリューション (\*.sln) ファイルを開きます。C#デモは/`<path-to-your-demo-dir>/demos/all/vc#`にあります。
2. VRmagic camera. NETラッパー参照が正しく設定されていることを確認します。vrmsbcam .NETラッパーは以下にインストールされます。  
`/usr/arm-linux-gnueabi/hf/vrmagic/usr/lib/cli/ VRmUsbCamNET/`
3. ソリューションファイルをビルドします。
4. カメラ上の実行ファイルをテストするには、→セクション6.5にあるように以下のどれかを使用します。
  - ネットワーク上でデモディレクトリを共有して、カメラにマウントします。
  - `scp`コマンドを使用してカメラにデモをコピーします。
  - USBフラッシュドライブかmicroSDカードを使用してファイルを交換します。

#### **ホストPCでのC#デモの実行:**

1. VRmUsbCamNET.dllとVRmUsbCamNET.dll.configが実行ファイルと同じディレクトリにあることを確認します。Monodevelopは通常は外部ライブラリを初期設定で実行ファイルの隣にコピーします。
2. MonoDevelopで実行ファイルを実行します。コマンドラインから実行ファイルを実行するには実行ファイルがあるディレクトリに変更して以下をタイプします。  
`mono ./<executable>.exe`

#### **D3でのC#デモの実行:**

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. 実行ファイルがあるディレクトリに変更します。
3. VRmUsbCamNET.dllとVRmUsbCamNET.dll.configが実行ファイルと同じディレクトリにあることを確認します。
4. 実行ファイルを実行するには以下をタイプします。  
`mono ./<executable>.exe`

## 7.5 HALCON Embedded

本章はD3カメラでHALCON Embeddedを使用する方法を説明します。D3上でHALCON Embeddedを使用するには、以下の必要条件を満たされなければなりません。必要なすべてのステップは以下のセクションで説明します。

### 必要条件

- D3当たり1つのHALCONライセンスが必要です(各カメラに付けられます)。
- HALCON EmbeddedランタイムをD3にインストールしなければなりません。
- HALCON VRmUsbCam取り込みドライバーを以下にインストールしなければなりません。
  - ホストシステム(HALCONとともにインストールされます。ホストシステムで開発と試験をしたい場合に必要です)。
  - D3カメラ

### オプションコンポーネント

- HALCON開発ライセンス
- ホストシステム用HALCON HDevelop

### 7.5.1 HALCONライセンスの取得

開発と生産のためのHALCONライセンスはドイツのHALCON代理店で得られます。ドイツ外で注文する場合は現地の代理店に問い合わせてください。HALCONライセンスについて詳しく知りたい場合はVRmagicの営業部にも問い合わせ可能です。価格表を送ることができます([sales.imaging@vrmagic.com](mailto:sales.imaging@vrmagic.com))。

#### ドイツのHALCON代理店

Dr. Karin Engelhardt  
CGI Systems GmbH  
Pettenkoferallee 39  
82402 Seeshaupt

Tel: (08801) 912 322  
Fax: (08801) 912 338  
<http://www.cgisystems.de>

## 7.5.2 HALCON HDevelopの取得

HDevelopはHALCON用の統合開発環境(IDE)で、登録したHALCON顧客が利用可能です。HDevelopはHALCONディストリビューションの一部で以下からダウンロードできます：<http://www.halcon.com/halcon/download/>。

HALCONは以下のプラットフォームで利用可能です：

- Windows (SSE2)
- Windows (x64)
- Linux (gcc- 4.x、SSE2)
- Linux (x86\_64、gcc- 4.x)
- Mac OS X (VRmagicは対応していません)

HALCONに付属の説明書に従ってホストPCにHALCONをインストールしてセットアップします。

## 7.5.3 D3上でのHALCONサポートのセットアップ

D3でHALCONアプリケーションを動作させるには、以下の説明に従ってD3カメラにHALCONサポートパッケージをインストールします。サポートパッケージにはHALCON EmbeddedライントタイムとHALCON VRmUsbCam取り込みドライバーが含まれています。

### *D3へのHALCONサポートパッケージのインストール:*

1. まだの場合は、D3カメラでVRmagicリポジトリを有効にします (→セクション 6.8.1)。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. `apt-get update`を実行してパッケージリストをアップデートします。
4. 以下を実行してD3にHALCONサポートパッケージをインストールします：  
`apt-get install halcon11-rt halcon11-rrm-imgacq-interface halcon11-rrm-support`  
▶ 関連ファイルはすべて/opt/halcon11にインストールされます。
5. HALCON ライセンスをたとえばSCP コマンドを使用してカメラの /opt/halcon11/licensesディレクトリにコピーします。カメラのライセンスファイルはlicense.datという名称でなければなりません(ファイルの名称変更が必要になる場合があります)。

## 7.5.4 D3でのHALCONインストールのテスト

### *HBenchの実行:*

HBenchスクリプトは、HALCONとHALCONライセンスがD3に正しくインストールされていることを確認します。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. HBenchを開始するには以下を実行します。  
/opt/halcon11/bin/arm-vmagic\_d3-linux/hbench

### *テスト画像の取り込み:*

このデモスクリプトは、カメラを開いて、1フレーム取り込んで、現在のディレクトリにJPEGファイルとして保存します。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. デモスクリプトを開始するには以下を実行します。  
/opt/halcon11/bin/arm-vmagic\_d3-linux/hrun -p /opt/halcon11/  
procedures /opt/halcon11/demos/VrmAcqSimpleDemo.hdev  
▶ JPEGファイルはデモディレクトリ/opt/halcon11/demos/に保存されます。
3. この画像をカメラからホストPCに転送できます。ホストPCでscpコマンドを使用してこれを行うことができます (→セクション6.5.2):  
scp <D3\_USERNAME>@<D3\_IP\_ADDRESS>:<IMAGE\_NAME>  
/some/local/directory

## 7.5.5 HALCON出力の視覚化

カメラのHDMIかRGB888インターフェースに外部モニターを接続できます。モニターは、画像、領域、テキストの表示などにHALCONが使用できます。HALCON出力を視覚化する前に、XServerをカメラにインストールする必要があります。

### *カメラへのXServerのインストール:*

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. apt-get updateを実行してパッケージリストをアップデートします。
3. 以下を実行してXServer をインストールします:  
apt-get install --no-install-recommends xserver-xorg-core xserver-xorg-input-all xserver-xorg-video-fbdev xfonts-scalable

### *X電源管理とスクリーンセーバーの無効化:*

外部のモニターが自動的に電源を切られてスクリーンセーバーが表示されるのを防ぐために、XServerコンフィギュレーションファイルを編集します。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. NanoテキストエディタでXServerコンフィギュレーションファイルを編集します:  
`nano /etc/X11/xorg.conf`
3. 以下を入力します:  
Section "ServerFlags"  
    Option "IgnoreABI" "True"  
    Option "BlankTime" "0"  
    Option "StandbyTime" "0"  
    Option "SuspendTime" "0"  
    Option "OffTime" "0"  
EndSection
4. ファイルを保存します。

### *起動時のXServer自動開始の有効化:*

カメラの電源投入時にXServerを自動的に開始するにはrc.localファイルを編集します。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. Nanoテキストエディタでrc.localファイルを編集します:  
`nano /etc/rc.local .`
3. 以下のラインを追加します:  
`/usr/bin/Xorg -nocursor &`
4. ファイルを保存します。
  - ▶ これで、D3のHDMIやRGB888インターフェースに接続されたモニターをHALCONで使用することができます。モニターの使用方法について詳しくはVRmagic HALCONパッケージに付属のデモを参照してください。

## 7.5.6 VRmagicによるHALCONサポートプロシーダの使用

VRmagicはHALCONの扱いを改良する特別なプロシーダを開発しました。これでHALCON内の扱いがユーザーにはかなり簡単になります。以下のプロシーダが利用可能です。

- VRmInitCamera
- VRmInitDisplayWindow
- VRmSetGPIO
- VRmOpenSerial

### 備考

#### VRmagic HALCONデモ

デモにはVRmagic HALCONプロシーダの使用法に関する詳細な情報を提供するVRmagic HALCONパッケージが付属しています。すべてのデモはD3の/opt/halcon11/demosにあります。

#### VRmInitCamera

このプロシーダはカメラを開きます。成功すると、画像取り込みのハンドルとセンサーの幅と高さを受け取ります。

VRmInitCamera ('DeviceName', 'ExtTrigger', ImageWidth, ImageHeight, AcqHandle, Result)

このプロシーダの入力パラメーター:

|            |                                |
|------------|--------------------------------|
| DeviceName | for example 'VRmD3MFC #XXXXXX' |
| ExtTrigger | 'true' // 'false'              |

#### VRmInitDisplayWindow

システムに応じて、このプロシーダは開発しているシステムまたはD3上の出力のためのウィンドウを開きます。1つ以上のディスプレイウィンドウを開いて組織化する方法の情報についてはデモを見てください。

VRmInitDisplayWindow (Window\_width, Window\_height, border, Window\_posx, Window\_posy, WindowHandle)

このプロシーダの入力パラメーター:

|               |       |
|---------------|-------|
| Window_width  | in px |
| Window_height | in px |
| border        | in px |
| Window_posx   | in px |
| Window_posy   | in px |

## VRmSetGPIO

このプロシージャで、D3プラットフォーム上でGPIOを楽に切り換えることができます。

VRmSetGPIO (gpio, value, ret)

このプロシージャの入力パラメーター:

|       |                         |
|-------|-------------------------|
| gpio  | 切り換えるGPIOの数             |
| value | GPIOの値 (0 = off、1 = on) |



### 備考

#### D3 GPIO

利用可能なすべてのGPIOの概要については→[セクション6.7](#)を参照してください。

## VRmOpenSerial

このプロシージャはD3上でシリアルコンソールを開きます。返されるSerialHandleでHALCON read\_serialとwrite\_serialファンクションを使用できます。

VRmOpenSerial ('/dev/ttyO1', 115200, 8, 'none', 'none', 1, 1, 1000, SerialHandle, Exception, ret)

このプロシージャの入力パラメーター:

|                  |              |
|------------------|--------------|
| Port             | '/dev/ttyO1' |
| Baud rate        | 115200       |
| Data bits        | 8            |
| FlowControl      | 'none'       |
| Parity           | 'none'       |
| StopBits         | 1            |
| Timeout          | 1            |
| InterCharTimeout | 1000         |



### 備考

#### D3シリアルポート

D3のシリアルポートの情報については→[5章](#)と→[11章](#)を参照してください。



## 8 カメラコンフィギュレーション

### 8.1 ビデオ出力の設定

D3のHD Video Processing Subsystemは3つのグラフィックスパイプラインと3つのビデオ出力を備えています。(→73ページの図20)。グラフィックスパイプラインは独立してビデオ出力に切り換えることができます。以下のビデオ出力があります。

- **HDMI (display0)**: このデジタルビデオ出力はカメラのRGB888とHDMIコネクタに接続されます。グラフィックスパイプラインをHDMIグラフィックス出力に切り換えると、RGB888とHDMIコネクタに同じ信号が出力されます。
- **DVO2 (display1)**: このビデオ出力は接続されません。
- **SD (display2)**: アナログビデオ出力はカメラのSビデオコネクタに接続されます。

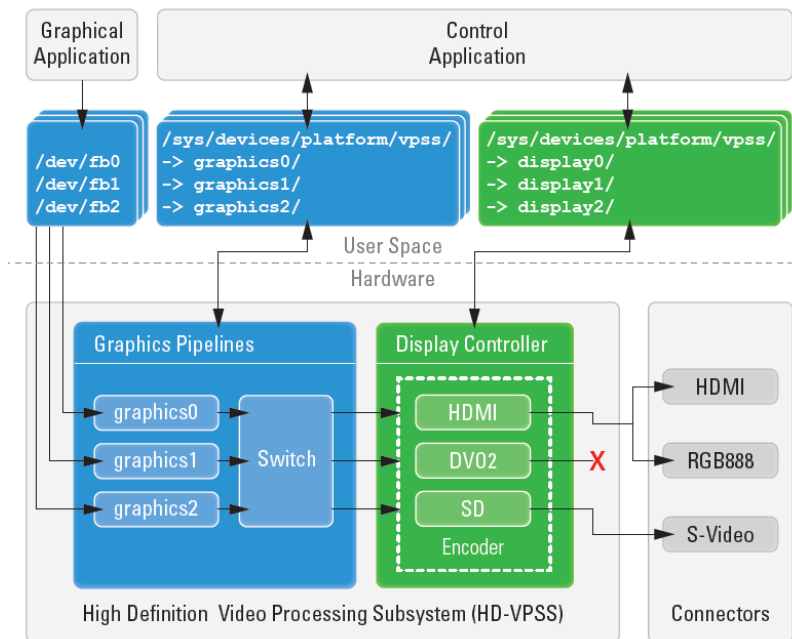


図20: D3のビデオ処理システム(簡略図)

## 備考

HD-VPSSの設定の詳細な情報は[Texas Instrumentsのホームページ](#)(TI81X X PSP VPSS Video Driver User Guide)にあります。

### *グラフィックスパイプラインのビデオ出力への切り換え:*

グラフィックスパイプラインのON/OFFを切り換えることもパイプラインを特定のビデオ出力に切り換えることもできます。便利なコンフィギュレーションに以下のスクリプトを使用することを推奨します。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. コマンド`vpss_switch_graphics_to_output`を実行します。
  - ▶ 現在の設定とコマンド使用の情報が表示されます。
3. グラフィックスパイプラインを設定するには、スクリーン上の指示に従います。

## 備考

グラフィックスパイプラインをHDMIビデオ出力に切り換えると、このグラフィックスパイプラインのビデオシグナルはカメラのHDMIインターフェースとRGB888インターフェースに同時に出力されます。

初期設定では、`graphics0`パイプラインはHDMIビデオ出力に切り換えられ、`graphics2`パイプラインはSDビデオ出力に切り換えられます。

### *ビデオ出力のビデオモードの設定:*

ビデオ出力用のビデオモードを設定するには、便利なコンフィギュレーションに以下のスクリプトを使用することを推奨します。HDMI(`display0`)ビデオ出力はカメラのHDMI/RGB888コネクタに接続され、SD(`display2`)ビデオ出力はSビデオコネクタに接続されます。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. コマンド`vpss_set_graphic_mode`を実行します。
  - ▶ 現在の設定とコマンド使用の情報が表示されます。
3. ビデオ出力のどれかにビデオモードを設定するには、スクリーン上の指示に従います。

## 8.2 USBコントローラー(USBホストまたはUSBデバイス)の設定

### 可能なUSBホスト/デバイスの設定

D3には2つのUSBインターフェース、USB0およびUSB1があります。両方のUSBインターフェースは初期設定ではUSBホストとして設定されます。D3がUSBデバイスとして機能するようにインターフェースのUSBコントローラーの機能を変更できます。以下の設定が可能です。

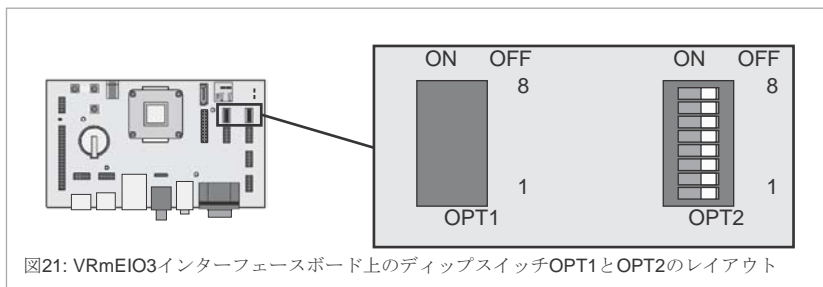
| USB0インターフェース | USB1インターフェース | USBコンフィギュレーションの名称 |
|--------------|--------------|-------------------|
| ホスト          | ホスト          | ホスト/ホスト           |
| ホスト          | デバイス         | ホスト/デバイス          |
| デバイス         | ホスト          |                   |
| デバイス         | 未使用          | デバイスのみ            |

### LinuxカーネルでのUSBドライバー

カメラのUSBコンフィギュレーションはLinuxカーネルのUSBドライバーに書き込まれます。USBコンフィギュレーションを変更するには、以下のセクションに示すようにD3カメラのLinuxカーネルを再構築して再コンパイルしなければなりません。

### VRmEIO3インターフェースボードのディップスイッチ

D3にVRmEIO3評価用インターフェースボードが備わっている場合は、カーネルコンフィギュレーションに加えてOPT1のディップスイッチ5と6の設定を変更しなければなりません。



| OPT1   ディップスイッチ5と6 |    |            |     |             |  |
|--------------------|----|------------|-----|-------------|--|
| 5                  | ON | USB0 = ホスト | OFF | USB0 = デバイス |  |
| 6                  | ON | USB1 = ホスト | OFF | USB1 = デバイス |  |

## 備考

USBインターフェースの設定の詳細情報は[Texas Instrumentsのホームページ](#) (DM81xx AM38X X USB User Guide)にあります。

### 8.2.1 D3用のカスタムカーネルの作成

## 備考

USBコンフィギュレーションを変更するには、D3のLinuxカーネルを再構築して再コンパイルしなければなりません。LinuxカーネルソースはVRmagic D3 EZSDKに含まれています、これはホストPCにインストールされていなければなりません(→セクション4.3.6)。

このマニュアルやREADMEファイルではカスタムLinuxカーネルの全体を扱うことはできません。カスタムカーネルを作成するにはLinuxを深く理解しなければなりません。

以下が基本的手順です:

- カメラ用のカスタムカーネルを作成するには、VRmagic D3 EZSDKに付属のD3 Linuxカーネルソースにカーネルを基礎づける必要があります。詳しくは→セクション4.3.6を参照してください。
- 以下の3つの→セクションで指定されている望みの設定でD3カーネルを再構築して再コンパイルします。これはホストPCで行われます。
- カメラに必要と思われるドライバーモジュールを含む新規カーネルをコピーします。

## 8.2.2 ホスト/ホストとしてのコンフィギュレーション

このコンフィギュレーションでは、両方のUSBインターフェースUSB0とUSB1はホストとして動作します。

### **i** 備考

D3は初期設定ではホスト/ホストコンフィギュレーションになっています。

#### *USBコンフィギュレーションをホスト/ホストに設定:*

1. ホストPCの/opt/vrmagic/vrm-d3-ezsdk/board-support/linux-DM8148-\*にあるREADMEファイルを開いて詳しい説明を見ます。
2. READMEファイルの指示に従います。make ARCH=arm menuconfigコマンドを入力した後に、この順序で以下の設定をします:  
**Device Drivers > USB support > USB Gadget Support:** Deactivate (N).  
**Device Drivers > USB support > Support for host-side USB:** Activate (Y).  
**Device Drivers > USB support > Driver Mode:** "USB Host"に設定。
3. 修正を保存してmenuconfigを終了します。
4. make ARCH=arm ulmageを実行してカーネルをコンパイルします。
5. ステップ1のREADMEファイルにあるようにD3に新規カーネルをコピーします。
6. カメラにVRmEIO3インターフェースボードが備わっている場合は、OPT1の両方のディップスイッチ5と6をONの位置に設定します (→75ページの図21)。
7. カメラを再起動して新規カーネルを読み込みます。

## 8.2.3 ホスト/デバイスとしてのコンフィギュレーション

このコンフィギュレーションでは、1つのUSBポートはホストとして、もう片方のポートはデバイスとして動作します。インターフェースのどちらがホストでどちらがデバイスになるかは次によって決まります:

- ディップスイッチの設定 (VRmEIO3インターフェースボードのみ)。
- 使用するケーブルのタイプ (VRmCUEO3インターフェースボードのみ)。

### USBコンフィギュレーションをホスト/デバイスに設定:

1. ホストPCの/opt/vrmagic/vrm-d3-ezsdk /board-support/linux- DM8148-\*にあるREADMEファイルを開いて詳しい説明を見ます。
2. READMEファイルの指示に従います。make ARCH=arm menuconfigコマンドを入力した後に、この順序で以下の設定をします:  
**Device Drivers > USB support > USB Gadget Support: Activate (Y).**  
**Device Drivers > USB support > Driver Mode: "Both host and peripheral: USB OTG (On The Go) Device"**に設定。  
**Device Drivers > USB support > USB Gadget Support > USB Peripheral Controller: "Inventra HRC USB Peripheral (TI, ADI,...)"**に設定。
3. **Device Drivers > USB support > USB Gadget Support > USB Gadget Drivers**に進みます。
4. <M>とタイプして必要なガジェットドライバーを選択します。これはガジェットドライバー (たとえば"Mass Storage Gadget")をカーネルモジュールとしてビルドします。
5. 修正を保存してmenuconfigを終了します。
6. make ARCH=arm ulmageを実行してカーネルをコンパイルします。
7. make ARCH=arm modulesを実行してカーネルモジュールをコンパイルします。
8. ステップ1のREADMEファイルにあるようにD3に新規カーネルをコピーします。カメラにカーネルモジュールをコピーする必要もあります。カーネルモジュールは、通常はカーネルファイルディレクトリのサブディレクトリにあって、ファイル拡張子は\*.koです。カーネルモジュールをコピーする場所について詳しくは外部のドキュメントを参照してください。
9. カメラにVRmEIO3インターフェースボードが備わっている場合は、OPT1のディップスイッチ5と6(→75ページの図21)の片方をONの位置(=ホスト)、もう片方をOFFの位置(=デバイス)に設定します。
10. カメラを再起動して新規カーネルを読み込みます。

#### ■備考

##### USBインターフェースの接続

インターフェースのどちらがホストでどちらがデバイスになるかは次によって決まります:

- ディップスイッチの設定 (VRmEIO3インターフェースボードのみ)。
- 使用するケーブルのタイプ (VRmCUEO3インターフェースボードのみ)。

USBインターフェースの接続については→第5章を参照してください。

## 8.2.4 デバイスのみとしてのコンフィギュレーション

このコンフィギュレーションでは、USB0ポートはデバイスとして動作して、USB1ポートは動作しません。

### USBコンフィギュレーションをデバイスのみを設定:

1. ホストPCの/opt/vrmagic/vrm-d3-ezsdk/board-support/linux-DM8148-\*にあるREADMEファイルを開いて詳しい説明を見ます。
2. READMEファイルの指示に従います。make ARCH=arm menuconfigコマンドを入力した後に、この順序で以下の設定をします:  
**Device Drivers > USB support > USB Gadget Support: Activate (Y)。**  
**Device Drivers > USB support > Support for host-side USB: Deactivate (N)。**  
**Device Drivers > USB support > Driver Mode: "USB Peripheral (gadget stack)"**に設定。  
**Device Drivers > USB support > USB Gadget Support > USB Peripheral Controller: "Inventra HDRC USB Peripheral (TI, ADI, ...)"**に設定。
3. **Device Drivers > USB support > USB Gadget Support > USB Gadget Drivers**に進みます。
4. <M>とタイプして必要なガジェットドライバーを選択します。これはガジェットドライバー (たとえば"Mass Storage Gadget")をカーネルモジュールとしてビルドします。
5. 修正を保存してmenuconfigを終了します。
6. make ARCH=arm ulmageを実行してカーネルをコンパイルします。
7. make ARCH=arm modulesを実行してカーネルモジュールをコンパイルします。
8. ステップ1のREADMEファイルにあるようにD3に新規カーネルをコピーします。カメラにカーネルモジュールをコピーするの必要もあります。カーネルモジュールは、通常はカーネルファイルディレクトリのサブディレクトリにあって、ファイル拡張子は\*.koです。カーネルモジュールをコピーする場所について詳しくは外部のドキュメントを参照してください。
9. カメラにVRmEIO3インターフェースボードが備わっている場合は、OPT1のディップスイッチ5 (→75ページの図21) をOFFの位置 (USB0=デバイス) に設定します。

### 備考

#### USBインターフェースの接続

カメラが「デバイスのみ」に設定されている場合は、カメラのUSB0インターフェースをUSBホストに接続します。「デバイスのみ」のコンフィギュレーションではUSB1インターフェースは使用されません。

USBインターフェースの接続については→第5章を参照してください。

## 8.3 WiFi接続のセットアップ

### 8.3.1 WiFiサポート

ワイヤレスネットワーク接続はD3ファームウェアのバージョン4.2から対応しています。D3付属品キットに適切な無線アダプターが含まれているかまたはVRmagicから別に入手することができます。

もちろん他の無線USBアダプターを使用することもできます。以下のチップセットに基づくすべてのアダプターが対応しています: Atheros、Ralink RT2800/RT2x00、Realtek 8192C/8188C。

### 8.3.2 WiFiインターフェースの開始と停止

#### *手動でのWiFiインターフェースの開始と停止:*

1. WiFi USBアダプターをカメラのUSBホストポートに接続します (→第5章)。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. WiFiインターフェースを開始するには `ifup wlan0` を実行します。
4. WiFiインターフェースを停止するには `ifdown wlan0` を実行します。

#### *自動でのWiFiインターフェースの開始と停止:*

PCプラットフォーム用のほとんどのLinuxディストリビューションとは対照的に、WiFiインターフェースのホットプラグによる自動開始やWiFiローミングモードのような高度な機能には対応していません。

1. WiFi WiFi USBアダプターをカメラのUSBホストポートに接続します (→第5章)。  
起動時にWiFiインターフェースを開始するには、WiFi USBアダプターが永続的に差し込まれていることを確認してください。
2. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
3. `/etc`ディレクトリに変更します。
4. `nano rc.local`とタイプして`rc.local`ファイルを編集します。
5. 起動時にWiFiインターフェースを開始するには、`ifup wlan0`を`rc.local`ファイルの`exit 0`の前に追加します。



### 8.3.3 ネットワークコンフィギュレーションの変更

#### **DHCPネットワークコンフィギュレーションを使用したWiFiインターフェースの設定:**

ネットワークがDHCPサーバーを備えている場合は、以下のコンフィギュレーションを使用します。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. `/etc/network`ディレクトリに変更します。
3. インターフェースコンフィギュレーションファイルで以下のラインをコメントアウトします:  

```
iface wlan0 inet dhcp
wpa-driver wext
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

#### **静的ネットワークコンフィギュレーションを使用したWiFiインターフェースの設定:**

1. 前の指示のステップ1と2に従います。
2. インターフェースコンフィギュレーションファイルで、以下のラインを含むように、`wlan0`コンフィギュレーションを変更します。括弧内の値を実際の静的ネットワークコンフィギュレーションに置き換えます:  

```
iface wlan0 inet static
address
<IP_ADDRESS>
netmask <NETMASK>
gateway
<GATEWAY_IP>
dns-nameservers <PRIMARY_DNS_IP> [<SECONDARY_DNS_IP> ...]
wpa-driver wext
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

### 8.3.4 ワイヤレスネットワークへの接続

WiFi認証と暗号化はwpa\_supplicantユーティリティを介して管理されますが、これはD3カメラファームウェア内にすでにインストールされています。WPA2暗号化で確保されたWiFiネットワークに接続するには、SSID(「Name」)とネットワークのパスワードの両方を必要とします。

*ワイヤレスネットワークへの接続:*

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. 事前に共有されたキー (PSK) を生成するには、以下を実行します。  
wpa\_passphrase <SSID> <PASSWORD> .
3. /etc/wpa\_supplicantディレクトリに変更します。
4. wpa\_supplicant.confファイルで、ssidとpskの値をそれに従って変更します。

### 8.3.5 WiFiアクセスポイントのスキャン

*WiFiアクセスポイントのスキャン:*

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. WiFiアクセスポイントをスキャンするには、以下を実行します。  
ifconfig wlan0 up  
iwlist wlan0 scan

### 8.3.6 エキスパートWiFiコンフィギュレーション

公式のサポートはWPA2の安全なアクセスポイントに接続したWiFi(最も一般的なセットアップ)の使用にのみ制限されますが、WiFi機能を利用するにはさらに可能性があります。

- ▶ ネットワークインターフェースコンフィギュレーションとwpa\_supplicantに関するさらに詳しい明細についてはUbuntuのドキュメントを参照してください。

## 8.4 電源管理

### 8.4.1 CPU周波数とコア電圧の設定

カーネルバージョン1.08(VRmagic SDKバージョン4.2.0)から、カーネルはCPU周波数を600メガヘルツか1ギガヘルツに切り換えることができます。カーネルはCPU周波数に応じてコア電圧を自動的に合わせます。

#### オペレーティングモード

- 電力節約モード: CPU周波数600 MHz、コア電圧1.1 V
- 性能優先モード: CPU周波数1 GHz、コア電圧1.35 V

電力節約モードではカメラは約250mW未満の電力を消費します。

#### Governors

ガバナーはCPU周波数をコントロールするドライバーです。以下のガバナーがカーネルによってサポートされています。

- **ondemand** (初期設定):  
このガバナーは現在のCPU負荷に応じて2つのオペレーティングモードを自動的に切り換えます。ほとんどの場合はこの設定が最適です。
- **powersave**  
このガバナーは電力節約モードに永続的に切り換えます。
- **performance**  
このガバナーは性能優先モードに永続的に切り換えます。

#### CPUオペレーティングモードの設定:

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. 利用可能なガバナーを表示するには、以下をタイプします。  

```
cat /sys/devices/system/cpu/cpu0/cpufreq/  
scaling_available_governors
```
3. 現在のCPU周波数を表示するには、以下をタイプします。  

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
```
4. 現在有効なガバナーを表示するには、以下をタイプします。  

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

5. 上に述べたガバナーのどれかを有効にするには、以下をタイプします。

```
echo ondemand > /sys/devices/system/cpu/cpu0/cpufreq/
scaling_governor
```

```
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/
scaling_governor
```

```
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/
scaling_governor
```

#### 8.4.2 デバイスの電源切断

電力を節約するために、以下のD3サブシステムの電源を切断することができます。

| デバイス           | 初期設定 | 消費電力 (アイドル) |
|----------------|------|-------------|
| Ethernet       | 有効   | 350 mW      |
| Etherney Phy   | 有効   | 100 mW      |
| HDVPSS (Video) | 有効   | 575 mW      |
| USB            | 有効   | 100 mW      |
| SATA           | 無効   | 170 mW      |
| CAN            | 無効   | 15 mW       |
| Sound          | 無効   | 0 mW        |

イーサネットインターフェースのon/offの切り換え:

Ethernet PHYをoffに切り換えるとイーサネットは自動的に無効になります(以下の指示を参照)。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. イーサネットをoffに切り換えるにはifdown eth0とタイプします。
3. イーサネットをonに切り換えるにはifup eth0とタイプします。
4. イーサネットの自動開始を無効にするには以下をタイプします。  
rm /etc/rc2.d/S20ifplugd
5. イーサネットの自動開始を有効にするには以下をタイプします。  
ln -s /etc/init.d/ifplugd /etc/rc2.d/S20ifplugd

**イーサネット物理的レイヤー (PHY)のon/offの切り換え:**

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. イーサネットをoffに切り換えるにはvrm-ethernet-phy disableとタイプします。
3. イーサネットをonに切り換えるにはvrm-ethernet-phy enableとタイプします。

**HDビデオ処理サブシステム(HDVPSS)のon/offの切り換え:**

HDVPSSプロパティは、動作中は無効にすることができません。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. HDVPSSの自動開始を無効にするには以下をタイプします。  
`rm /etc/rc2.d/S20load-hd-firmware.sh`
3. イーサネットの自動開始を再び有効にするには以下をタイプします。  
`ln -s /etc/init.d/load-hd-firmware.sh /etc/rc2.d/S20load-hd-firmware.sh`

**USB/SATA/CAN/Soundのon/offの切り換え:**

対応するカーネルドライバーモジュールは、`/etc/modules`ファイルを介して自動的に読み込むことができます。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. すべてのカーネルモジュールがリストされている`/etc/modules`ファイルを編集します。
3. 無効または有効にするモジュールをコメントまたはコメントアウトします。

## 8.5 リアルタイムクロックの設定

D3インテリジェントカメラは内部にバッテリー駆動のリアルタイムクロックを備えています。クロックに現在の日付と時間を設定するには以下の手順で行います。

### ■ 備考

VRmEIO3インターフェース評価用ボードはリチウムコインセルを備えていて、カメラの電源が切られてもリアルタイムクロックは動作します。

カメラにVRmCUEO3インターフェースボードが備わっている場合は、バッテリーはありません。リアルタイムクロックを設定する前に3Vバッテリーが接続されていることを確認してください。

#### *リアルタイムクロックの設定:*

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. 日付と時間を設定するには `hwclock --set --date="MM/DD/YY HH:MM:SS"` とタイプします。
3. ハードウェアクロックからシステムタイムを設定するには `hwclock -s` とタイプします。

## 9 さらになるデバイスの接続

### 9.1 SPIデバイス

D3カメラに外部SPIデバイスを接続することができます。SPIピンは「SPI」というラベルのあるコネクタにあります (VRmCUEO3およびVRmEIO3インターフェースボード)。

#### 備考

インターフェースボードの図とSPIインターフェースの詳しいピン配列については以下のセクションを参照してください。

- ▶ VRmCUEO3インターフェースボード: →セクション11.2
- ▶ VRmEIO3インターフェースボード: →セクション11.4

対応するキャラクタデバイスファイルは/dev/spidev3.0です。デバイスファイルが存在しない場合は、コマンド`modprobe spidev`で読み出すことができます。

- ▶ spidevデバイスについて詳しくは以下を参照してください  
<https://www.kernel.org/doc/Documentation/spi/spidev>

#### SPIインターフェースのテスト:

spidev\_testとspidev\_fdxツールを使用してSPIインターフェースのテストができます。両方のツールは、ファームウェアバージョン4.2以降はD3カメラにすでにインストールされています。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. テストメッセージを作成するには`spidev_test -D /dev/spidev3.0`を実行します。
3. 全二重オペレーションをテストするには`spidev_fdx /dev/spidev3.0`を実行します。

#### 備考

両方のツールのソースコードはkernel.orgで入手できます。ソースコードはspidevドライバーの使用例として使用できます:

[https://www.kernel.org/doc/Documentation/spi/spidev\\_test.c](https://www.kernel.org/doc/Documentation/spi/spidev_test.c)  
[https://www.kernel.org/doc/Documentation/spi/spidev\\_fdx.c](https://www.kernel.org/doc/Documentation/spi/spidev_fdx.c)

## 9.2 I2Cデバイス

D3カメラに外部I2Cデバイスを接続することができます。VRmCUEO3インターフェースボードでは、I2Cピンは「RGB888」というラベルのあるコネクタにあります。VRmEIO3インターフェースボードでは、I2Cピンは「I2C」というラベルのあるコネクタにあります。

対応するキャラクタデバイスファイルは/dev/i2c-3です。

### 備考

インターフェースボードの図とI2Cインターフェースの詳しいピン配列については以下のセクションを参照してください。

- ▶ VRmCUEO3インターフェースボード: →セクション11.2
- ▶ VRmEIO3インターフェースボード: →セクション11.4

### I2Cツールのインストール:

### 備考

I2Cツールは、ファームウェアバージョン4.2以降はD3カメラにすでにインストールされています。カメラがより古いファームウェアを使用している場合にのみ、以下に示したようにUbuntuパッケージ管理機能を使用してI2Cツールをインストールしてください。

I2Cツールの詳しいドキュメントは以下にあります:  
<http://www.lm-sensors.org/wiki/i2cToolsDocumentation>

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. パッケージリストを最新のものにするには`sudo apt-get update`を実行します。
3. I2Cツールをインストールするには`sudo apt-get install i2c-tools`を実行します。
  - ▶ インストール後に、コマンド*i2cset*、*i2cget*、*i2cdump*、*i2cdetect*が利用可能になります。

### I2Cバスを介してテストメッセージを送る:

1. I2Cツールを使用してI2Cバスを介してテストメッセージを送ることができます。このためには以下を実行します。  
`i2cset 3 <CHIP-ADDRESS> <DATA-ADDRESS> [<VALUE>]`

### I2Cデバイスの読み込み:

1. I2Cデバイスを読み込むには以下を実行します。  
`i2cget 3 <CHIP-ADDRESS> <DATA-ADDRESS>`



## 9.3 CANデバイス

D3のCANバスインターフェースに外部デバイスを接続することができます。VRmCUEO3インターフェースボードでは、CANピンは「SER SVID」というラベルのあるコネクタにあります。VRmEIO3インターフェースボードでは、CANピンは「CAN」というラベルのあるコネクタにあります。

### 備考

インターフェースボードの図とI2Cインターフェースの詳しいピン配列については以下のセクションを参照してください。

- ▶ VRmCUEO3インターフェースボード: →セクション11.2
- ▶ VRmEIO3インターフェースボード: →セクション11.4

### CANドライバーモジュールの読み込み:

CANドライバーモジュールは初期設定では読み込まれません。読み込みは以下の手順で行います。

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. CANモジュールを読み込むには、`/etc/modules file`の以下のラインをアンマウントします:

```

can
can-raw
can-bcm
can-dev
vcan
d_can
d_can_platform
    
```

### CANインターフェースの開始:

1. ターミナルを使用してカメラにアクセスします (→セクション6.2)。
2. CANインターフェースを開始するには以下を実行します:
 

```

ip link set can0 type can bitrate 50000 triple-sampling on
ip link set can0 up
            
```
3. CANインターフェースをテストするには、事前にインストールされている`canutils`: `canconfig`, `candump`, `canecho`, `cansend`, `cansequence`を使用します。短いヘルプを見るには以下を実行します。
 

```

<toolname> --help .
            
```

### 備考

DCAN Linuxドライバーについて詳しくは以下を参照してください:

[http://processors.wiki.ti.com/index.php/TI81X\\_X\\_PSP\\_DCAN\\_Driver\\_User\\_Guide](http://processors.wiki.ti.com/index.php/TI81X_X_PSP_DCAN_Driver_User_Guide)

## 10 ファームウェアの更新と回復

### 10.1 一般的説明

カメラのファームウェアを更新したり、現在インストールされているファームウェアを元の状態に戻したりすることができます。手順は基本的に同じです。現在、VRmagicカメラを更新するには2つの選択肢があります。

#### イーサネットによる更新(回復には不適切です)

イーサネットを介した更新プロセスはホストPCで動作しているホストアプリケーションによってコントロールされます。イーサネットによる更新には完全なファームウェアのあるカメラと動作するSSHサーバー(出荷時初期設定)を必要とします。したがって、回復の目的には不適切です。

#### SDカードによる更新/回復

このメソッドは最も安全で、回復にも適切です。SDカードがカメラブーツに接続されると、カメラは内部メモリからではなくSDカードから起動されます。起動後に、ファームウェアのすべての部分を交換することができます。



#### 注意

##### 更新や回復中にデータが損失する場合があります

更新プロセス中にカメラに格納されたすべてのデータが失われるので注意してください。データの損失を避けるために以下のようにしてください:

- データを別のコンピュータか記憶装置にバックアップしてください。
- 自動的に保存されて更新後に復旧されるファイルとディレクトリのリストを指定してください。このオプションについては以下で説明します。

## 10.2 カメラファームウェアのダウンロード

カメラのファームウェアを更新するには、まずVRmagic Linux D3 Camera SDKに含まれる最新のファームウェアをダウンロードする必要があります。最新のVRmagic Linux D3 Camera SDKをホストPCにダウンロードするにはVRmagic Easy Installerを使用します。

**VRmagic Easy Installerを使用したVRmagic Linux D3 Camera SDKの更新:**

1. →セクション4.2.2にあるようにVRmagic Easy Installerで最新のVRmagicソフトウェアをインストールします。このプロセスで、少なくともファームウェアを含んでいるVRmagic Linux D3 Camera SDKを選択します。
  - ▶ 新規SDKはホストPCの/opt/vrmagic/sdk-\*ディレクトリにインストールされます (\* = SDKバージョンナンバー)。
  - ▶ vrmagic-linux-d3-firmware-\*.zipという名称の新規ファームウェアのzipファイルがホストPCの/opt/vrmagic/firmwareディレクトリにコピーされます (\* = ファームウェアバージョンナンバー)。

## 10.3 イーサネットによるファームウェアの更新

**要件**

- イーサネットの更新機能はSSHプロトコルを使用します。したがって、更新が開始する前にSSHサーバーがカメラで動作している必要があります。初期設定ではDropbear SSHサーバーがVRmagicのすべてのインテリジェントカメラで動作します。
- カメラはネットワークに接続されていてホストPCからアクセス可能でなければなりません (→セクション6.1.2)。
- MonoがLinuxホストPCにインストールされていなければなりません。VRmagic Custom Ubuntuを使用している場合はすでにインストールされています。

**イーサネットによる更新の実行:**

1. →10.2章にあるようにファームウェアをダウンロードします。
2. Ubuntu LinuxホストPCでコマンドシェルを開始します。
3. コマンドcdを使用してホームディレクトリに移ります。
4. ホームディレクトリに新規ファームウェアを解凍します:  
`unzip /opt/vrmagic/firmware/vrmagic-linux-d3-firmware-<VERSION_NUMBER>.zip`
  - ▶ ホームディレクトリに新規ディレクトリvrmagic-linux-d3-firmware-\*が作成されます (\* = ファームウェアバージョンナンバー)。



## 備考

### カスタムD3 Linuxカーネル

カスタムカーネルまたはルートファイルシステムを使用するには、カーネルをD3 Linuxカーネルソースに基づかせる必要があります。D3カーネルソースは、ホストPCにインストールしなければならないVRmagic D3 EZSDKに含まれています (→セクション4.3.6)。

- 更新プロセス中に失われてはならないファイルとディレクトリを指定するには、`/vrmagic-linux-d3-firmware-*/flash-data/backup.txt` ファイルを編集します。`backup.txt` ファイルの各ラインは、更新前にバックアップされて更新後に回復されるD3カメラ上のファイルやディレクトリを表します。

`backup.txt` の入力例:

```
/etc/vrmagic (カメラに格納される設定の保存)
/your/path/to/directory (ディレクトリ全体の保存)
/your/path/to/file.xyz (特定のファイルの保存)
```



## 注意

### `backup.txt` の入力の間違いで更新プロセスが失敗する可能性があります

`backup.txt` で指定するファイルとディレクトリに注意してください。たとえば新規システムで古いVRmagicライブラリを回復するとカメラが適切に動作しなくなることがあります。`backup.txt` で指定したファイルおよび/またはディレクトリが本当に存在することを確認してください。さもなければ、更新プロセスは失敗します。

- 更新するカメラのIPアドレスを見つけます。これには3つの選択肢があります:
  - `ifconfig` コマンドを使用してIPアドレスを見つけます (カメラへのSSH/シリアル接続が必要です)。
  - CamIabで**info** ボタンをクリックしてIPアドレスを見つけます (→セクション6.1)。
  - ホストシステムのターミナルで `vrm-eth-updater - l` を実行します。

後者の2つの選択肢は、カメラで `vrmcamserver` プロセスを動作させる必要があります (→セクション6.6)。



## 注意

### 電源切断によってデバイスが損傷する可能性があります

更新プロセス中にPCまたはカメラの電源が切れると、カメラが故障して修理のためにVRmagicに返送しなければならなくなることがあります。

7. イーサネット更新機能を実行します:  
`vrm-eth-updater [options] <IP_OF_CAMERA>  
 <PATH_TO_UNZIPPED_FIRMWARE_DIRECTORY>`

例:

`vrm-eth-updater -y 10.0.0.167 ~/vrmagic-linux-d3-firmware-1.76`

## **i** 備考

### イーサネット更新オプション

- l `vrmcamserver`の動作するネットワークのVRmagicデバイスをリストします。
- u `<USER_NAME>` カメラのSSHログイン (初期設定: root)。
- p `<PASSWORD>` カメラのSSHパスワード (初期設定: vrmagic)。
- y 更新をするかどうかユーザーに尋ねない。
- h コマンドオプション(このヘルプメッセージ)を表示して終了。
- r `<DIRECTORY>` リモート更新機能のディレクトリ (初期設定: `./remote-updater`); このオプションは通常は不要です。

- ▶ PCでファイアウォールが動作している場合は、更新ツールがネットワークにアクセスしようとするときWindowsは警告を表示します。アクセスを許可しないと更新プロセスは終了しません。
  - ▶ ステップ7で-yオプションを使用しない場合は、更新をするかどうか尋ねられます。
8. `<y>`と答えて更新プロセスを継続します。
- ▶ カメラは再起動されて、一時的にホストPCからLinuxシステムを読み出します。それからすべてのファームウェアファイルがホストPCから読み出されます。
  - ▶ これで更新は実際に開始されます。これが更新プロセスの重要な分岐点です: この時点でうまくいかない場合は、カメラを古いファームウェアバージョンで問題なく再起動するすることができます。この時点の後で問題が起これば(たとえば停電)、カメラが破損して修理のためにVRmagicに返送しなければならないことがあります。けれどもカメラをVRmagicに返送する前にカメラの回復を試すことができます (→セクション10.5)。
  - ▶ 更新が終了するとすぐに、カメラは再起動して、短時間後に新規ファームウェアバージョンで使用の準備ができます。

## **i** 備考

何か問題がある場合や、単に更新プロセスを確認したい場合は、ホストPCの `/tmp/vrm-flash-fw-<serial_no>.log` ファイルを見てください。

さらに、`vrmcdethupdater` ツール自体のすべてのメッセージはファイル `/tmp/vrmcdethupdater.log` に記録されます。

## 10.4 SDカードによるファームウェアの更新

オプションのVRmagic付属品キットに付属のSDカードを使用してファームウェアを更新することができます。自前のSD更新カードを作成することも可能です (→セクション10.4.1)。

### 備考

#### VRmagic microSDカードまたはその他のmicroSDカードによる更新

カメラのファームウェアを新規バージョンに更新したい場合は、以下のセクションで説明するように新規ファームウェアバージョンを使用するmicroSDカードを作成してください。VRmagic microSDカードまたは他のどんなmicroSDカードでも使用できます。

### 10.4.1 更新用SDカードの作成

#### 要件

- 最低2GBの容量のmicroSDカードか元のVRmagic microSDカード。
- システムに以下のユーティリティをインストールしなければなりません。
  - fdisk
  - mkfs.vfat
  - mkfs.ext2
- スクリプトはsfdiskやmountのようなコマンドを使用するので、スクリプトを実行するためにシステム上でルートユーザー権限を必要とします。

#### 注意

SDカード上のすべてのデータは以下の手順で消去されます。

#### 更新用SDカードの作成:

1. →章10.2にあるようにファームウェアをダウンロードします。
2. microSDカードをLinuxホストPCに接続します。
3. microSDカードをマウントしてはいけません。自動的にマウントされた場合は、アンマウントしますが、差し込まれたままにします。

## 備考

記憶デバイスの自動マウントを無効にするには以下を実行します。  
`gsettings set org.gnome.desktop.media-handling automount false`  
`gsettings set org.gnome.desktop.media-handling automount-open false`

再び自動マウントを有効にするには、同じコマンドを**true**で使用します。

4. コマンドシェルを開始します。
5. コマンド**cd**を使用してホームディレクトリに移ります。
6. ホームディレクトリに新規ファームウェアを解凍します:  
`unzip /opt/vrmagic/firmware/vrmagic-linux-d3-firmware-<VERSION_NUMBER>.zip`
  - ▶ ホームディレクトリに新規ディレクトリ**vrmagic-linux-d3-firmware-\***が作成されます (\* = ファームウェアバージョンナンバー)。
7. SDカードのLinuxデバイスファイルの名称を確認します。SDカードがカードリーダーに直接接続される場合は、通常は**/dev/mmcblk1**とします。SDカードがUSBアダプターを使用して接続されている場合は、通常**/dev/sdb**になります。
8. 更新用SDカードの作成を開始するには以下を実行します。  
`sudo vrm-generate-update-sd --usb-device=<SD_CARD_DEVICE>`  
`--firmware-dir=<PATH_TO_UNZIPPED_FIRMWARE_DIRECTORY>`
  - ▶ スクリプトが終了するとSDカードは使用の準備ができます。使用法は以下のセクションで説明します。

### 更新用SDカードのパーティション

SDカードには、2つのパーティション、FATパーティションとext2パーティションがあります。FATパーティションは、ファームウェアファイルとアップデートスクリプトのためのコンフィギュレーションファイルを含んでいます。このパーティションはWindowsとLinuxでアクセス可能です。ext2パーティションはLinuxカーネルとルートファイルシステムを含んでいます。このパーティションはLinuxでのみアクセス可能です。

## 10.4.2 更新プロセスのカスタマイズ

カメラのファームウェアを更新する前に、実行される更新のタイプを指定しなければなりません。2つの更新モードが可能です:

- **手動更新:** 手動更新は更新プロセスのさらなるコントロールが可能です。カメラのRS232インターフェースを介したターミナル接続が必要です。
- **自動更新:** シリアル接続は不要です。これは最速の方法です。

さらに更新プロセス中に失われてはならないファイルやディレクトリを指定することができます。

**update.cfgでの更新モード(手動または自動)の設定:**

1. 更新用SDカードをホストPCに接続します。
2. 更新用SDカードの第1パーティション (FAT、たとえば/dev/mmcblk1) をホストPCのローカルディレクトリにマウントします。
3. SDカードのFATパーティションにある/flash-data/update.cfgファイルを編集して更新モードを設定します:  
AUTOMATIC\_UPDATE=0 (手動更新の場合)  
AUTOMATIC\_UPDATE=1 (自動更新の場合)
4. update.cfgファイルを保存します。

**更新プロセス中に失われてはならないファイルとディレクトリの指定:**

1. 更新用SDカードをホストPCに接続します。
2. 更新用SDカードの第1パーティション (FAT、たとえば/dev/mmcblk1) をホストPCのローカルディレクトリにマウントします。
3. SDカードのFATパーティションにある/flash-data/backup.txtファイルを編集します。  
backup.txtファイルの各ラインは、更新前にバックアップされて更新後に回復されるD3カメラ上のファイルやディレクトリを表します。

backup.txtの入力例:

/etc/vrmagic (カメラに格納される設定の保存)  
/your/path/to/directory (ディレクトリ全体の保存)  
/your/path/to/file.xyz (特定のファイルの保存)



**注意**

**backup.txtの入力の間違いで更新プロセスが失敗する可能性があります**

backup.txtで指定するファイルとディレクトリに注意してください。たとえば新規システムで古いVRmagicライブラリを回復するとカメラが適切に動作しなくなることがあります。backup.txtで指定したファイルおよび/またはディレクトリが本当に存在することを確認してください。さもなければ、更新プロセスは失敗します。

4. backup.txtファイルを保存します。



### 10.4.3 手動更新

#### 更新用SDカードを使用した手動更新の実行:

1. →セクション10.4.2にあるように更新モードを手動に設定します。
2. →セクション10.4.2にあるように更新プロセス中に失われてはならないD3カメラ上のファイルとディレクトリを指定します。
3. カメラに更新用SDカードを接続して、カメラを再起動します。
  - ▶ カメラはSDカードから起動されます。これはログインプロンプトが“VRmagic-#<Serial Number>-SD”を読み出すことをチェックして確認できます。
4. シリアルコンソールを介してカメラにアクセスします (→セクション6.2.2)。
  - ▶ カメラへのターミナル接続がなされるはずですが。
5. 更新手順を開始するには`vrmm-update-fw.sh`を実行します。
  - ▶ 更新手順が開始します。
  - ▶ セキュリティの問い合わせが表示されて、カメラのすべてのデータを失ってもいいかどうか尋ねられます。もちろん、ステップ1で指定されたファイルとディレクトリは更新後に復旧されます。
6. “yes”と答えて更新プロセスを継続します。
  - ▶ 更新が終了すると以下のメッセージが表示されます:  
`Update process successfully completed.`
  - ▶ エラーがある場合は、再び`vrmm-update-fw.sh`を実行します。これが失敗したら次を入力します。

```
cat /tmp/vrmm-update-fw.log cat  
/tmp/vrmm-update-fw.klog
```

そして出力を`service@vrmmagic.com`に送ってください。

7. 更新がうまくいったら、`halt -p`コマンドでカメラをシャットダウンします。
  - ▶ カメラはシャットダウンします。ステータスLEDが消えるのを待ちます(→セクション3.3)。
8. SDカードを切断します。
9. 電源ボタンを押して再びカメラの電源を入れます。
  - ▶ これでカメラは新規ファームウェアで起動します。ステータスLEDが点灯するのを待ちます(→セクション3.3)。カメラは用意ができます。

#### 10.4.4 自動更新

複数のVRmagicカメラを更新する場合にのみこのメソッドを使用することを推奨します。また、前のセクションで示した手動メソッドによって少なくとも1台のカメラを更新して、更新プロセスが成功することを確認してください。

##### *更新用SDカードを使用した自動更新の実行:*

1. →セクション10.4.2にあるように更新モードを自動に設定します。
2. →セクション10.4.2にあるように更新プロセス中に失われてはならないD3カメラ上のファイルとディレクトリを指定します。
3. カメラに更新用SDカードを接続して、カメラを再起動します。
  - ▶ カメラはSDカードから起動されます。
  - ▶ 起動後にvrm-update-fw.shスクリプトが自動的に実行されます。
4. スクリプトがうまく実行されて、カメラのステータスLEDが点灯するのを待ちます (→セクション3.3)。
5. 電源ボタンを押して一時的にカメラをシャットダウンします。
  - ▶ カメラはシャットダウンします。ステータスLEDが消えるのを待ちます。
6. SDカードを切断します。
7. 電源ボタンを押して再びカメラの電源を入れます。
  - ▶ これでカメラは新規ファームウェアで起動します。ステータスLEDが点灯するのを待ちます(→セクション3.3)。カメラは用意ができます。
8. めったにないことですが5分以上経過してもステータスLEDが点灯しない場合は、カメラのブートローダーの設定が間違っていることが考えられます。可能な場合は、手動更新を実行してください(→セクション10.4.3)。

##### **備考**

カメラのファームウェアが思いがけなく上書きされるのを防ぐために、update.cfg内でAUTOMATIC\_UPDATEを0に設定することによってSDカードを再び"disarm"にします。(→セクション10.4.2)。

## 10.5 SDカードによるデバイスの回復

D3インテリジェントカメラが起動しないなど問題がある場合は、更新用SDカードを使用してカメラを回復することができます。更新用SDカードを挿入した状態でカメラを起動すると、カメラは更新用SDカードから直接起動します。このやり方でカメラのルートファイルシステムの誤設定を回避することができます。

## 備考

### VRmagic MicroSDカードによる更新

VRmagic付属品キットに付属するMicroSDカードには事前にインストールされたファームウェアが含まれています。このファームウェアバージョンを回復するのにMicroSDカードを使用することができます。

### MicroSDカードによる更新

回復に別のMicroSDカードを使用したい場合は、→セクション10.4.1にあるようにまずカードを作成します。回復するには、現在カメラにインストールされているのと同じファームウェアバージョンを選択します。

### 更新用SDカードによるデバイスの回復:

回復手順は以下に説明する3つのステップからなっています。

1. **SDカードからの起動:** 更新用SDカードをカメラのマイクロSDカードスロットに挿入してSDカードから起動します。これは→セクション10.4.3 (手動更新)のステップ1~4にあります。カメラへのターミナル接続がなされるはずですが。
2. **カメラからのデータの回復:** カメラのファイルシステムからデータを回復したい場合は、圧縮されたターボールのカメラからのルートファイルシステム全体をSDカードのFATパーティションにコピーすることができます。以下のコマンドを使用します:

```
mkdir /mnt/sd_fat  
mount /dev/mmcblk1p1 /mnt/sd_fat
```

```
mkdir /mnt/cam_rootfs  
mount /dev/mmcblk0p1 /mnt/cam_rootfs
```

```
tar czf /mnt/sd_fat/custom-rootfs.tgz /mnt/cam_rootfs
```

3. **動作するルートファイルシステムの回復:** カメラを再び起動可能にするには、→セクション10.4.3のステップ5~9にある手動ファームウェア更新手順を開始します。これは誤設定されたカメラルートファイルシステムを更新用SDカードに含まれるファームウェアに取り替えます。

## 11 補足

### 11.1 電気特性



**警告!**

機器が破損する可能性があります

D3上の3.3 V I/Oピンは5 Vに耐性がありません。最大定格はV<sub>CC</sub> +0.3 VとGND - 0.3 Vです。最大定格を守らないと器機の部品が破損する場合があります。

#### 一般電気特性

|                              |  |
|------------------------------|--|
| 電源入力                         | 5 V DC (+/-5%)   |
| コネクタの電源ピン                    | 3.3 V (+/- 3%)、最大20 A  |
| 3.3 V IOピンの最大定格              | V <sub>CC</sub> +0.3 V、GND - 0.3 V<br>5 V 非耐性  |
| IOコネクタのすべてのアクティヴIOピンと電圧ピンの保護 | TVSダイオードによる保護:<br>IEC 61000 - 4-2 (ESD) ±15 kV (air)、±8 kV (contact)<br>IEC 61000 - 4- 4 (EFT) (5/50 ns) 20 A (I/O)、40 A (VDD) IEC 61000 - 4-5 (Lightning) 6 A (8/20 μs) |

#### GPIO 25 ... 118

|              |                                      |
|--------------|--------------------------------------|
| 駆動性能         | 6 mA                                 |
| 立ち上がり/立ち下り時間 | 3.5 ns<br>入力最小時間は31 μs (内部グリッチフィルター) |
| 高レベル入力電圧     | 2 ... 3.3 V                          |
| 低レベル入力電圧     | 0 ... 0.8 V                          |

#### GPIO 136 ... 159

|              |  |
|--------------|--|
| 駆動性能         | 8 mA   |
| 立ち上がり/立ち下り時間 | これらのGPIOはI <sup>2</sup> Cによって設定と読み出しがなされます。したがって高速GPIOファンクションやリアルタイムアプリケーションには使用できません。 |
| 高レベル入力電圧     | 3.5 ... 5.0 V  |
| 低レベル入力電圧     | 0 ... 1.5 V  |

## 11.2 VRmCUEO3インターフェースボードのピン配列

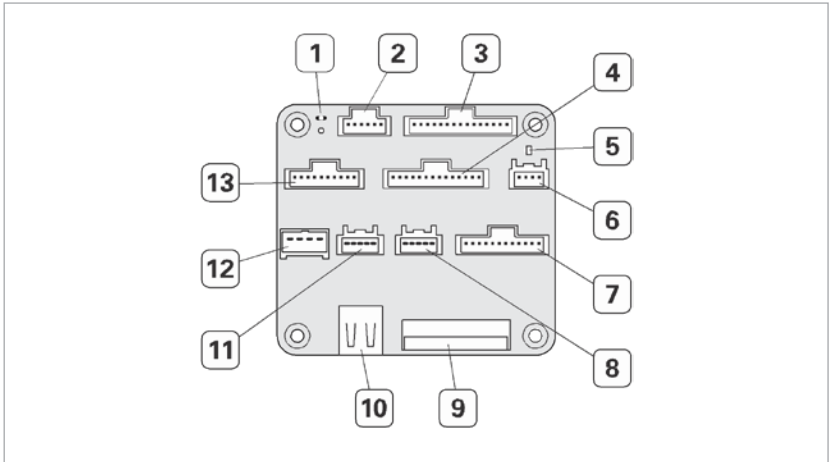


図22: VRmCUEO3インターフェースボード (インターフェース名については下表を参照)

| ボード上のラベル | 説明          | 内部名                                    |       |
|----------|-------------|--|-------|
| 1        | VBAT        | リアルタイムクロック用3Vバッテリーコネクタ                 | -     |
| 2        | GPIO        | オーディオコーデック用GPIOまたはMcASPバス(SPDIF出力)     | CON12 |
| 3        | ETHERNET    | ギガビットイーサネット                            | CON7  |
| 4        | SER SVID    | シリアルコンソール、RS232、S-Video、UART1、CAN      | CON5  |
| 5        |             | ステータスLED                               | -     |
| 6        | TRIG + STRB | トリガーとストロボ                              | CON8  |
| 7        | SPI         | コントロールLED、ウォッチドッグ、電源ボタン、リセット、SPI       | CON9  |
| 8        | USB0        | USBポート0                                | CON11 |
| 9        | RGB888      | RGB888とI <sup>2</sup> C 2線シリアルインターフェース | CON3  |
| 10       | HDMI        | Micro HDMI                             | CON13 |
| 11       | USB1        | USBポート1                                | CON10 |
| 12       | 5VDC        | 電源入力 (5V DC)                           | CON4  |
| 13       | uSD         | SDIO (たとえばMicroSDカード用)                 | CON6  |

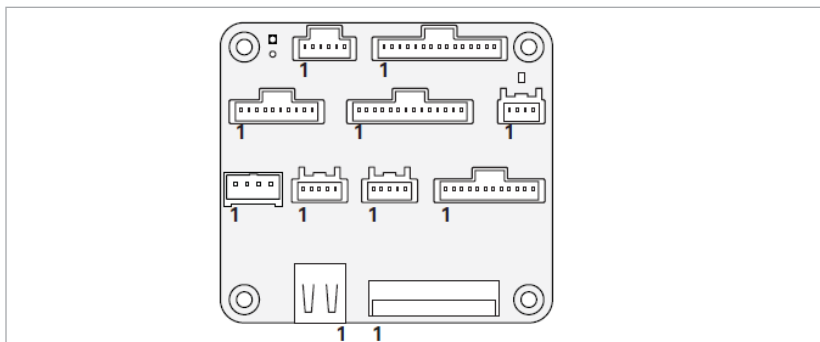


図23: VRmCUEO3インターフェースボード上のコネクタのピン1の位置

**i**

**備考**

以下の表の見出しは、ボード上のコネクタのラベル、短い説明、コネクタのタイプを(この順で)示しています。

**VBAT** | バッテリーコネクタ

| ピン | シグナル | ピン | シグナル |
|----|------|----|------|
| □  | +3 V | ○  | GND  |

**GPIO** | 汎用I/O | JST BM06B-NSHSS-TBT

| ピン | シグナル                              | ピン | シグナル                        |
|----|-----------------------------------|----|-----------------------------|
| 1  | +3.3 V<br>供給ピン、3.3 V ± 3%、最大20 mA | 2  | GPIO25 / 3.3 V / IO<br>汎用ピン |
| 3  | GPIO26 / 3.3 V / IO<br>汎用ピン       | 4  | GND                         |
| 5  | GPIO27 / 3.3 V / IO<br>汎用ピン       | 6  | 保留                          |

**ETHERNET | ギガビットイーサネット | JST BM15B-NSHSS-TBT**

| ピン | シグナル   | ピン | シグナル                                    |
|----|--|----|---|
| 1  | LINK_LED_C / GND & 220 Ω<br>LINK LEDのカソードに接続         | 2  | LINK_LED_A / 3.3 V<br>LINK LEDのアノードに接続  |
| 3  | MDI0_P / LVDS / IO<br>差動ペア0正シグナル                     | 4  | MDI0_N / LVDS / IO<br>差動ペア0負シグナル        |
| 5  | GND  | 6  | MDI1_P / LVDS / IO<br>差動ペア1正シグナル        |
| 7  | MDI1_N / LVDS / IO<br>差動ペア1負シグナル                     | 8  | GND                                     |
| 9  | MDI2_P / LVDS / IO<br>差動ペア2正シグナル                     | 10 | MDI2_N / LVDS / IO<br>差動ペア2負シグナル        |
| 11 | GND  | 12 | MDI3_P / LVDS / IO<br>差動ペア3正シグナル        |
| 13 | MDI3_N / LVDS / IO<br>差動ペア3負シグナル                     | 14 | ACT_LED_C / GND<br>ACTIVITY LEDのカソードに接続 |
| 15 | ACT_LED_A / +3.3 V & 220 Ω /<br>ACTIVITY LEDのアノードに接続 |    |   |

ボード上のパルス変成器。パルス変成器のガルバーニ絶縁 1500 V、1分。

CPU側はTVSダイオードで保護。

IEC 61000 - 4-2 (ESD) ±15 kV (air)、±8 kV (contact)

IEC 61000 - 4-4 (EFT) (5/50 ns) 20 A (I/O)、40 A

(VDD) IEC 61000 - 4-5 (Lightning) 6 A (8/20 μs)

**SER SVID | シリアル接続、RS232、S-Video、UART1、CAN | JST BM14B-NSHSS-**

| ピン | シグナル  | ピン | シグナル   |
|----|---|----|--|
| 1  | +3.3 V CAN  | 2  | CAN_TxD/GPIO32 / 3.3 V<br>CAN送信 (UARTのみ) / 汎用ピン        |
| 3  | CAN_RxD/GPIO33 / 3.3 V<br>CAN受信 (UARTのみ) / 汎用ピン         | 4  | GND CAN  |
| 5  | +3.3 V UART1  | 6  | UART1_TxD/GPIO36 / 3.3 V<br>UART1送信 (UARTのみ) / 汎用ピン    |
| 7  | UART1_RxD/GPIO37 / 3.3 V<br>UART1受信 (UARTのみ) / 汎用ピン     | 8  | GND UART1  |
| 9  | RS232_TxD / +/-5.0 V / OUT<br>RS232送信 (UART0、シリアルコンソール) | 10 | RS232_RxD / +/-5.0 V / IN<br>RS232受信 (UART0、シリアルコンソール) |
| 11 | GND RS232   | 12 | SVIDEO_Y   |
| 13 | SVIDEO_C  | 14 | GND  |

**TRIG+STRB | トリガーとストロボ | JST BM04B-NSHSS-TBT**

| ピン | シグナル                               | ピン | シグナル                               |
|----|------------------------------------|----|------------------------------------|
| 1  | + 3.3 V                            | 2  | Trigger / 3.3V / IN<br>アクティブトリガー入力 |
| 3  | Strobe / 3.3V / OUT<br>アクティヴストロボ出力 | 4  | GND                                |

Strobe OUT: 駆動性能 8 mA; 立ち上がり時間 3.5 ns; 立ち下がり時間 3.5 ns; デューティサイクルはソフトウェアによって調整可能; 極性はソフトウェアによって調整可能

Trigger IN: 立ち上がり時間 > 1 μs; 立ち下がり時間 > 1 μs; デューティサイクルは ExposureTime と ReadoutTime に依存; アクティヴハイまたはローはソフトウェアによって調整可能; 内部プルアップ抵抗

FPGAなしのカメラ: 最小シグナルタイム 31 μs

FPGAを備えたカメラ: 最小シグナルタイプ 300 ns



**SPI | SPI、LED、電源ボタン、リセットボタン、ウォッチドッグ | JST BM12B-NSHSS-**

| ピン | シグナル   | ピン | シグナル  |
|----|--|----|---|
| 1  | +3.3 V_MISC  | 2  | LED_CTRL# / max 3.3 V / OD<br>ユーザー設定可能LED、外部PUが必要 |
| 3  | WD_OUT# / 3.3 V / OUT<br>ウォッチドッグローアクティブ出力(エッジトリガー) | 4  | GND_MISC  |
| 5  | PWRBTN# / 3.3 V / IN<br>外部電源ボタン用シグナル               | 6  | RESET# / 3.3 V / IN<br>外部リセットボタン用シグナル             |
| 7  | SPI2_CS# / 3.3 V / OUT<br>SPIバスチップ選択出力             | 8  | SPI2_MOSI / 3.3 V / OUT<br>SPIバスマスター出力スレーブイン      |
| 9  | +3.3 V_SPI   | 10 | SPI2_MISO / 3.3 V / IN<br>SPIバスマスタースレーブアウト        |
| 11 | SPI2_CLK / 3.3 V / OUT<br>SPIバスクロック出力              | 12 | GND_SPI   |

ピン2: LEDに必要なのが10 mA未満の場合は、LEDをGNDと適切なR<sub>v</sub>で直接駆動できます。もっと大きな電流が必要な場合は、ピン2のシグナルをトランジスタのコントロールに使用してLEDを切り換えることができます。この場合は20 mAまで供給できる電源ピンのどれかを使用できます。

ピン5: GNDに対して直接電源ボタンを使用します。ボード上にプルアップ抵抗とデバウンスがあります。ローシグナルの最小時間は100 μs。

ピン6: GNDに対して直接リセットボタンを使用します。ボード上にプルアップ抵抗とデバウンスがあります。

**USB0 | USBポート0 | JST BM05B-NSHSS-TBT**

| ピン | シグナル                                | ピン | シグナル                                |
|----|-------------------------------------|----|-------------------------------------|
| 1  | +VCC_USB1                           | 2  | USB1_N / 3.3 V / IO<br>USBポート1負シグナル |
| 3  | USB1_P / 3.3 V / IO<br>USBポート1正シグナル | 4  | USB1_ID / 3.3 V / IN<br>USB IDシグナル  |
| 5  | GND_USB1                            |    |                                     |

### USB1 | USBポート1 | JST BM05B-NSHSS-TBT

| ピン | シグナル                                | ピン | シグナル                                |
|----|-------------------------------------|----|-------------------------------------|
| 1  | +VCC_USB0                           | 2  | USB0_N / 3.3 V / IO<br>USBポート1負シグナル |
| 3  | USB0_P / 3.3 V / IO<br>USBポート1正シグナル | 4  | USB0_ID / 3.3 V / IN<br>USB IDシグナル  |
| 5  | GND_USB1                            |    |                                     |

### RGB888 | デジタルビデオI/O | Hirose FH23-45S-0.3SHAW-05

| ピン    | シグナル   | ピン    | シグナル   |
|-------|--|-------|--|
| 1     | RGB24_IN_OUT# / IO / 3.3 V<br>ビデオポート方向シグナル; RGB888 OUTの場合はGNDに接続; INの場合はオープン | 2...4 | GND  |
| 5...7 | +5.0 V   | 8     | I2C(1)_SDA / IO / 3.3 V<br>I2Cデータ入力/出力           |
| 9     | RGB24_PWR_EN / OUT / 3.3 V<br>外部回路用電源有効                                      | 10    | I2C(1)_SCL / OUT / 3.3 V<br>I2Cクロック出力            |
| 11    | RGB24_INT / IN / 3.3 V<br>割込入力、たとえばHDMI HPDET用                               | 12    | GND  |
| 13    | VOUT_CLK / VIN_HSync / 3.3 V<br>ビデオクロック出力/HSync入力                            | 14    | GND  |
| 15    | +3.3 V   | 16    | VOUT_AVID / VIN_CLK / 3.3 V<br>アクティブビデオ出力/クロック入力 |
| 17    | VOUT_VS / VIN_DE / 3.3 V<br>Vsync出力/データ有効入力                                  | 18    | VOUT_HS / VIN_VS / 3.3 V<br>HSync出力/ VSync入力     |
| 19    | VOUT_R9 / VIN_D20 / 3.3 V<br>赤ビット9/入力データビット20                                | 20    | VOUT_R8 / VIN_D19 / 3.3 V<br>赤ビット8/入力データビット19    |
| 21    | VOUT_R7 / VIN_D18 / 3.3 V<br>赤ビット7/入力データビット18                                | 22    | VOUT_R6 / VIN_D17 / 3.3 V<br>赤ビット6/入力データビット17    |
| 23    | VOUT_R5 / VIN_D16 / 3.3 V<br>赤ビット5/入力データビット16                                | 24    | VOUT_R4 / VIN_D15 / 3.3 V<br>赤ビット4/入力データビット15    |
| 25    | VOUT_R3 / VIN_D22 / 3.3 V<br>赤ビット3/入力データビット22                                | 26    | VOUT_R2 / VIN_D23 / 3.3 V<br>赤ビット2/入力データビット23    |
| 27    | GND  | 28    | VOUT_G8 / VIN_D13 / 3.3 V<br>緑ビット8/入力データビット13    |
| 29    | VOUT_G9 / VIN_D14 / 3.3 V<br>緑ビット9/入力データビット14                                | 30    | VOUT_G6 / VIN_D11 / 3.3 V<br>緑ビット6/入力データビット11    |

**RGB888** | デジタルビデオI/O | Hirose FH23- 45S- 0.3SHAW- 05

|    |   |    |   |
|----|---|----|---|
| 31 | VOOUT_G7/ VIN_D12 / 3.3 V<br>緑ビット7/入力データビット12 | 32 | VOOUT_G4/ VIN_D9 / 3.3 V<br>緑ビット4/入力データビット9   |
| 33 | VOOUT_G5/ VIN_D10 / 3.3 V<br>緑ビット5/入力データビット10 | 34 | VOOUT_G2/ VIN_D21 / 3.3 V<br>緑ビット2/入力データビット21 |
| 35 | VOOUT_G3/ VIN_D8 / 3.3 V<br>緑ビット3/入力データビット8   | 36 | GND   |
| 37 | VOOUT_B9/ VIN_D6 / 3.3 V<br>青ビット9/入力データビット6   | 38 | VOOUT_B8/ VIN_D5 / 3.3 V<br>青ビット8/入力データビット5   |
| 39 | VOOUT_B7/ VIN_D4 / 3.3 V<br>青ビット7/入力データビット4   | 40 | VOOUT_B6/ VIN_D3 / 3.3 V<br>青ビット6/入力データビット3   |
| 41 | VOOUT_B5/ VIN_D2 / 3.3 V<br>青ビット5/入力データビット2   | 42 | VOOUT_B4/ VIN_D1 / 3.3 V<br>青ビット4/入力データビット1   |
| 43 | VOOUT_B3/ VIN_D0 / 3.3 V<br>青ビット3/入力データビット0   | 44 | VOOUT_B2/ VIN_D7 / 3.3 V<br>青ビット2/入力データビット7   |
| 45 | GND   | 46 | —   |

**5V DC** | 電源入力 | Molex 87437- 0442

|    |                      |    |                      |
|----|----------------------|----|----------------------|
| ピン | シグナル                 | ピン | シグナル                 |
| 1  | +5.0 V DC IN (+/-5%) | 2  | +5.0 V DC IN (+/-5%) |
| 3  | GND                  | 4  | GND                  |

**μSD** | SDカードとデバイス用のコネクタ | JST BM10B-NSHSS-TBT

|    |   |    |                                     |
|----|---|----|-------------------------------------|
| ピン | シグナル                                      | ピン | シグナル                                |
| 1  | SDIO_D2 / 3.3 V / IO<br>SDIOポートビット2       | 2  | SDIO_D3 / 3.3 V / IO<br>SDIOポートビット3 |
| 3  | SDIO_CMD / 3.3 V / IO<br>SDIOポートコマンドシグナル  | 4  | +3.3 V                              |
| 5  | SDIO_CLK / 3.3 V / OUT<br>SDIOポートクロックシグナル | 6  | GND                                 |
| 7  | SDIO_D0 / 3.3 V / IO<br>SDIOポートビット0       | 8  | SDIO_D1 / 3.3 V / IO<br>SDIOポートビット1 |
| 9  | SDIO_CD / 3.3 V / IO<br>SDIOポートカード検出シグナル  | 10 | GND                                 |

## 11.3 VRmCUEO3インターフェースボードのケーブルプラン



### 備考

以下のケーブルプランは、VRmCUEO3付属品キットにあるケーブルを指します (→セクション3.4.1)。

#### 電源ケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_PWR  
 コネクタ#1: Molex 87439 - 0401  
 コネクタ#2: +5 Vレセプタクル

| #1ピン | #2ピン   | シグナル | ケーブルカラー |
|------|--------|------|---------|
| 1    | pin    | +5 V | オレンジ    |
| 2    | pin    | +5 V | 赤       |
| 3    | sleeve | GND  | 黒       |
| 4    | sleeve | GND  | 茶       |

#### トリガー/ストロボケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_TRG  
 コネクタ#1: JST NSHR- 04V-S  
 コネクタ#2 ... #5: 0.1"コネクタ用レセプタクル

| #1ピン | レセプタクル | シグナル       | ケーブルカラー |
|------|--------|------------|---------|
| 1    | #2     | +3.3 V DC  | 黄       |
| 2    | #3     | Trigger IN | 赤       |
| 3    | #4     | Strobe OUT | 黒       |
| 4    | #5     | GND        | 紫       |

### USB-Aアダプターケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_USB\_A  
 コネクタ#1: JST NSHR- 05V-S  
 コネクタ#2: USBタイプAレセプタクル

| #1ピン | #2ピン | シグナル | ケーブルカラー |
|------|------|------|---------|
| 1    | 1    | VBUS | 赤       |
| 2    | 2    | D-   | 白       |
| 3    | 3    | D+   | 緑       |
| 4    | 4    | ID   | 黄       |
| 5    | 4    | GND  | 黒       |

### USB OTGアダプターケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_USB\_OTG  
 コネクタ#1: JST NSHR- 05V-S  
 コネクタ#2: USBタイプMini ABレセプタクル

| #1ピン | #2ピン | シグナル | ケーブルカラー |
|------|------|------|---------|
| 1    | 1    | VBUS | 赤       |
| 2    | 2    | D-   | 白       |
| 3    | 3    | D+   | 緑       |
| 4    | 4    | ID   | 黄 (黒)*  |
| 5    | 5    | GND  | 黒 (黄)*  |

\*カラーは使用するUSBケーブルに依存します。ピンナンバーに対応するワイヤは常に正しいものです。

### GPIOケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_GPIO  
 コネクタ#1: JST NSHR- 06V-S  
 コネクタ#2 ... #7: 0.1"コネクタ用レセプタクル

| #1ピン | レセプタクル | シグナル      | ケーブルカラー |
|------|--------|-----------|---------|
| 1    | #2     | +3.3 V DC | 緑       |
| 2    | #3     | GPIO25    | 黄       |
| 3    | #4     | GPIO26    | 青       |
| 4    | #5     | GND       | 赤       |
| 5    | #6     | GPIO27    | 黒       |
| 6    | #7     | 保留        | 紫       |

### SPIケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_SPI  
 コネクタ#1: JST NSHR-12V-S  
 コネクタ#2 ... #13: 0.1"コネクタ用レセプタクル

| #1ピン | レセプタクル | シグナル            | ケーブルカラー |
|------|--------|-----------------|---------|
| 1    | #2     | +3.3 V DC       | オレンジ/赤  |
| 2    | #3     | LED_CTRL        | オレンジ/黒  |
| 3    | #4     | WD_OUT#         | 灰/赤     |
| 4    | #5     | GND             | 灰/黒     |
| 5    | #6     | PWRBTN#         | 白/赤     |
| 6    | #7     | RESET#          | 白/黒     |
| 7    | #8     | SPI_CS#         | 黄/赤     |
| 8    | #9     | SPI_MOSI        | 黄/黒     |
| 9    | #10    | +3.3 V DC (SPI) | ピンク/赤   |
| 10   | #11    | SPI_MISO        | ピンク/黒   |
| 11   | #12    | SPI_CLK         | オレンジ/2赤 |
| 12   | #13    | GND (SPI)       | オレンジ/2黒 |

### microSDアダプターケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_USD  
 コネクタ#1: JST NSHR-10V-S  
 コネクタ#2: JST NSHR-10V-S

| #1ピン | #2ピン | シグナル   | ケーブルカラー |
|------|------|--------|---------|
| 1    | 1    | D2     | 赤       |
| 2    | 2    | D3     | 黒       |
| 3    | 3    | CMD    | 黒       |
| 4    | 4    | +3.3 V | 黒       |
| 5    | 5    | CLK    | 黒       |
| 6    | 6    | GND    | 黒       |
| 7    | 7    | D0     | 黒       |
| 8    | 8    | D1     | 黒       |
| 9    | 9    | CD     | 黒       |
| 10   | 10   | GND    | 黒       |

## Serial/S-Videoケーブル

|                |                  |
|----------------|------------------|
| VRmagic注文ナンバー: | CAB_I_CUEO3_SVID |
| コネクタ#1:        | JST NSHR-14V-S   |
| コネクタ#2 ... #9: | 0.1"コネクタ用レセプタクル  |
| コネクタ#10:       | D-Sub9、オス        |
| コネクタ#11:       | S-Video、メス       |

| #1ピン | レセプタクル/ピン | シグナル              | ケーブルカラー |
|------|-----------|-------------------|---------|
| 1    | #2        | +3.3 V DC (CAN)   | 黒       |
| 2    | #3        | CAN_TXD/GPIO32    | 白       |
| 3    | #4        | CAN_RXD/GPIO33    | 茶       |
| 4    | #5        | GND (CAN)         | 緑       |
| 5    | #6        | +3.3 V DC (UART1) | 黄       |
| 6    | #7        | UART1_TXD/GPIO36  | 灰       |
| 7    | #8        | UART1_RXD/GPIO37  | ピンク     |
| 8    | #9        | GND (UART1)       | 青       |
| 9    | #10/3     | RS232_TXD         | 黄       |
| 10   | #10/2     | RS232_RXD         | 黒       |
| 11   | #10/5     | GND (RS232)       | 赤       |
| 12   | #11/3     | S-Video_Y         | 黄       |
| 13   | #11/4     | S-Video_C         | 紫       |
| 14   | #11/1+2   | GND               | 赤       |

### GBit Ethernetアダプターケーブル

VRmagic注文ナンバー: CAB\_I\_CUEO3\_ETH  
 コネクタ#1: JST NSHR-15V-S  
 コネクタ#2: DF13-10S

| #1ピン | #2ピン | シグナル   | ケーブルカラー |
|------|------|--------|---------|
| 1    |      | NC     |         |
| 2    |      | NC     |         |
| 3    | 2    | MDI0_P | オレンジ/赤  |
| 4    | 3    | MDI0_N | オレンジ/黒  |
| 5    | 1    | SHLD   | 黒       |
| 6    | 4    | MDI1_P | 灰/赤     |
| 7    | 7    | MDI1_N | 灰/黒     |
| 8    |      | NC     |         |
| 9    | 5    | MDI2_P | 白/赤     |
| 10   | 6    | MDI2_N | 白/黒     |
| 11   | 10   | SHLD   | 黒       |
| 12   | 8    | MDI3_P | 黄/赤     |
| 13   | 9    | MDI3_N | 黄/黒     |
| 14   |      | NC     |         |
| 15   |      | NC     |         |



## 11.4 VRmEIO3インターフェースボードのピン配列

RJ45、HDMI、USBなどの標準的なコネクタのピン配列はインターネットで簡単にわかるので、このマニュアルには記載しません。

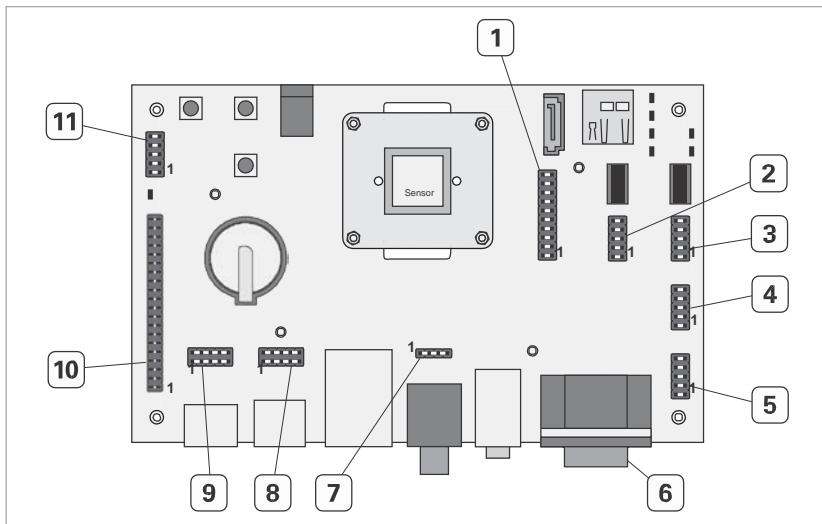


図24: VRmEIO3ボードのインターフェース (上面図)とピン1の位置

| ボード上のラベル | 説明         | 内部名  |       |
|----------|------------|--|-------|
| 1        | JTAG debug | TI JTAGコネクタ用JTAGインターフェース   | CON10 |
| 2        | GPIO_1     | I <sup>2</sup> Cを介した8 GPIO   | CON22 |
| 3        | GPIO_0     | I <sup>2</sup> Cを介した8 GPIO   | CON21 |
| 4        | CAN        | CANバス  | CON7  |
| 5        | SYS panel  | システムパネル  | CON19 |
| 6        |            | 上部: UART0 (RS232を介した)シリアルコンソール<br>下部: UART1 (RS232 1またはRS422/RS485)  | CON6  |
| 7        | I2C        | I <sup>2</sup> C 2線シリアルインターフェース                                      | CON9  |
| 8        | SPI        | SPIバス  | CON8  |
| 9        | muxed GPIO | 8 GPIO   | CON11 |
| 10       | VIDEO I/O  | OPT1のディップスイッチ8に応じてRGB888またはGPIO: ON = GPIO、OFF = RGB888 ( -セクション3.2) | CON20 |
| 11       | TRIG & STR | トリガーとストロボ  | CON16 |



### 警告!

装置が破損する可能性があります

カメラが動作している間に、ピンヘッダーまたはSATAインターフェースの接続や切断をしないでください! どのピンもショートさせないでください!

ピンヘッダーまたはSATAコネクタでコンポーネントの接続または切断を行う前に、カメラに電源が入っていないことを確認してください。



### 備考

以下の表の見出しは、ボード上のコネクタのラベル、短い説明、コネクタのタイプを(この順で)示しています。

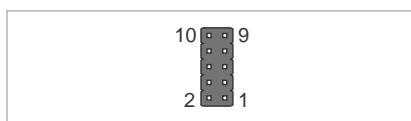


図25: ピンヘッダーのピンの順序

#### JTAG debug | JTAGインターフェース | 2x 10ピンヘッダー、TI JTAGコネクタを使用

| ピン    | シグナル                                  | ピン | シグナル                                  |
|-------|---------------------------------------|----|---------------------------------------|
| 1     | TMS / IN / 3.3 V<br>JTAGテストポートモード選択入力 | 2  | TRST# / IN / 3.3V<br>JTAGテストポートリセット入力 |
| 3     | TDI / IN / 3.3 V<br>JTAGテストポートデータ入力   | 4  | GND                                   |
| 5     | +3.3 V                                | 6  | KEY                                   |
| 7     | TDO / OUT / 3.3 V<br>JTAGテストポートデータ出力  | 8  | GND                                   |
| 9     | RTCK / OUT / 3.3 V<br>JTAGリターンクロック出力  | 10 | GND                                   |
| 11    | TCK / IN / 3.3 V<br>JTAGテストクロック入力     | 12 | GND                                   |
| 13    | EMU0 / IO / 3.3 V<br>エミュレータピン0        | 14 | EMU1 / IO / 3.3 V<br>エミュレータピン1        |
| 15    | SYS_RST# / IN / 3.3 V<br>システムリセット入力   | 16 | GND                                   |
| 17-19 | NC                                    | 20 | GND                                   |

## 備考

コネクタGPIO\_0とGPIO\_1の以下のGPIO (136...143と152...159)はI<sup>2</sup>C GPIOエキスパンダーを使用して提供されます。したがってそれらは使用を中断されません。

### GPIO\_0 | 汎用I/O | 2 x 5ピンヘッダー

| ピン | シグナル                         | ピン | シグナル                         |
|----|------------------------------|----|------------------------------|
| 1  | +5.0 V                       | 2  | GND                          |
| 3  | GPIO136 / IO / 5.0 V         | 4  | GPIO137 / IO / 5.0 V         |
| 5  | 汎用ピン<br>GPIO138 / IO / 5.0 V | 6  | 汎用ピン<br>GPIO139 / IO / 5.0 V |
| 7  | 汎用ピン<br>GPIO140 / IO / 5.0 V | 8  | 汎用ピン<br>GPIO141 / IO / 5.0 V |
| 9  | 汎用ピン<br>GPIO142 / IO / 5.0 V | 10 | 汎用ピン<br>GPIO143 / IO / 5.0 V |
|    | 汎用ピン                         |    | 汎用ピン                         |

### GPIO\_1 | 汎用I/O | 2 x 5ピンヘッダー

| ピン | シグナル                         | ピン | シグナル                         |
|----|------------------------------|----|------------------------------|
| 1  | +5.0 V                       | 2  | GND                          |
| 3  | GPIO152 / IO / 5.0 V         | 4  | GPIO153 / IO / 5.0 V         |
| 5  | 汎用ピン<br>GPIO154 / IO / 5.0 V | 6  | 汎用ピン<br>GPIO155 / IO / 5.0 V |
| 7  | 汎用ピン<br>GPIO156 / IO / 5.0 V | 8  | 汎用ピン<br>GPIO157 / IO / 5.0 V |
| 9  | 汎用ピン<br>GPIO158 / IO / 5.0 V | 10 | 汎用ピン<br>GPIO159 / IO / 5.0 V |
|    | 汎用ピン                         |    | 汎用ピン                         |

### CAN | CANバス | 2 x 5ピンヘッダー

| ピン | シグナル   | ピン   | シグナル  |
|----|--|------|---|
| 1  | NC   | 2    | NC  |
| 3  | CANL / IO / 5.0 V<br>ローレベルCAN電圧入力/出力<br>(絶縁) | 4    | CANH / IO / 5.0V<br>ハイレベルCAN電圧入力/出力<br>(絶縁) |
| 5  | GND_CAN                                      | 6-10 | NC  |

### SYS Panel | 様々なシグナルのあるシステムパネル | 2 x 5ピンヘッダー

| ピン | シグナル                                     | ピン | シグナル  |
|----|--|----|---|
| 1  | +5.0 V (always ON)                       | 2  | +3.3 V  |
| 3  | SLP_IN / IN / 3.3 V<br>アクティヴハイSLEEP入力    | 4  | GND   |
| 5  | PWRBTN# / IN / 3.3 V<br>外部電源ボタン用シグナル     | 6  | ACT_LED / OUT / 3.3 V<br>アクティヴハイイーサネット動作状態LED     |
| 7  | RESET# / IN / 3.3 V<br>外部リセットボタン用シグナル    | 8  | LINK_LED# / OUT / 3.3 V<br>アクティヴローイーサネットリンクLED    |
| 9  | WATCHDOG / OUT / 3.3 V<br>ウォッチドッグアクティヴ出力 | 10 | LED_CTRL# / OD / max 5.0 V<br>ユーザー設定可能LED、外部PUが必要 |

### UART0 | シリアルポート0 (RS232を介したシリアルコンソール) | D-Sub DE- 09P

| ピン  | シグナル   | ピン | シグナル  |
|-----|--|----|---|
| 1   | NC   | 2  | RS232_RxD# / +/-5.0 V / IN<br>シリアルポート0 RS232受信器 |
| 3   | RS232_TxD# / +/-5.0 V / OUT<br>シリアルポート0 RS232送信器 | 4  | NC  |
| 5   | GND  | 6  | NC  |
| 7-9 | NC   |    |   |

**UART1<sub>1</sub> | シリアルポート1 (RS232<sub>2</sub> またはRS422/RS485<sub>2</sub>) | D-Sub DE- 09P**

| ピン | シグナル   | ピン | シグナル  |
|----|--|----|---|
| 1  | NC   | 2  | RS232_RxD#/RS485_RxD# /<br>+/-5.0 V / IN <sub>1</sub><br>シリアルポート1 RS232受信器/<br>シリアルポート1 RS485負受信器 |
| 3  | RS232_TxD#/RS485_TxD# /<br>+/-5.0 V / OUT <sub>1</sub><br>シリアルポート1 RS232送信器/<br>シリアルポート1 RS485負送信器 | 4  | NC  |
| 5  | GND  | 6  | NC  |
| 7  | RS485_TxD / +5.0V / OUT <sub>1</sub><br>シリアルポート1 RS485正送信器   | 8  | RS485_RxD / +5.0V / IN <sub>1</sub><br>シリアルポート1 RS485正受信器   |
| 9  | NC   |    |   |

- 1 UART1を有効にするには、OPT2 / ディップスイッチ5をOFFに設定しなければなりません。
- 2 OPT2 / ディップスイッチ7を使用してRS232かRS485を選択します。ON = RS232、OFF = RS485。

**I2C | I2C 2線インターフェース(ポート2) | 1 x 4ピンヘッダー**

| ピン | シグナル                                   | ピン | シグナル   |
|----|--|----|--------|
| 1  | I2C(2)_SCL / OUT / 5.0 V<br>I2Cクロック出力  | 2  | +5.0 V |
| 3  | I2C(2)_SDA / IO / 5.0 V<br>I2Cデータ入力/出力 | 4  | GND    |

**SPI | シリアル周辺インターフェース(ポート2) | 2 x 5ピンヘッダー**

| ピン | シグナル                                   | ピン | シグナル   |
|----|--|----|--|
| 1  | +5.0 V                                 | 2  | NC   |
| 3  | SPI2_CS# / OUT / 3.3 V<br>SPIバスチップ選択出力 | 4  | SPI2_MOSI / OUT / 3.3 V<br>SPIバスマスター出力スレーヴイン |
| 5  | +3.3 V                                 | 6  | NC   |
| 7  | SPI2_CLK / OUT / 3.3 V<br>SPIバスクロック出力  | 8  | SPI2_MISO / IN / 3.3 V<br>SPIバスマスターインスレーヴアウト |
| 9  | GND                                    | 10 | NC   |

**mixed GPIO | 汎用I/O | 2 x 5ピンヘッダー**

| ピン | シグナル                             | ピン | シグナル                             |
|----|----------------------------------|----|----------------------------------|
| 1  | +5.0 V                           | 2  | GND                              |
| 3  | GPIO25 / IO / 5.0 V<br>汎用ピン 25 3 | 4  | GPIO26 / IO / 5.0 V<br>汎用ピン 26 3 |
| 5  | GPIO27 / IO / 5.0 V<br>汎用ピン 27 3 | 6  | 保留                               |
| 7  | GPIO32 / IO / 5.0 V<br>汎用ピン 32 4 | 8  | GPIO33 / IO / 5.0 V<br>汎用ピン 33 4 |
| 9  | GPIO36 / IO / 5.0 V<br>汎用ピン 36 5 | 10 | GPIO37 / IO / 5.0 V<br>汎用ピン 37 5 |

- 3 OPT2 / ディップスイッチ1 = ONでオーディオを無効にした場合
- 4 OPT2 / ディップスイッチ4 = ONでCANバスを無効にした場合
- 5 OPT2 / ディップスイッチ5 = ONでUART1を無効にした場合

| Video I/O   デジタルビデオI/O <sub>6</sub> (ディップスイッチ8がOPT1 = OFFの場合)   2x20ピンヘッダー |  |    |  |
|--|--|----|--|
| ピン   | シグナル   | ピン | シグナル   |
| 1  | VOUT_CLK / VIN_HSync / 3.3 V<br>ビデオCLK出力/HSync入力 | 2  | VOUT_VS/ VIN_DE / 3.3 V<br>Vsync出力/データ有効入力   |
| 3  | VOUT_AVID/ VIN_CLK / 3.3 V<br>アクティヴビデオ出力/クロック入力  | 4  | VOUT_HS/ VIN_VS / 3.3 V<br>HSync出力/ VSync入力  |
| 5  | GND  | 6  | GND  |
| 7  | VOUT_R2/ VIN_D23 / 3.3 V<br>赤ビット2/入力データビット23     | 8  | VOUT_R3/ VIN_D22 / 3.3 V<br>赤ビット3/入力データビット22 |
| 9  | VOUT_R4/ VIN_D15 / 3.3 V<br>赤ビット4/入力データビット15     | 10 | VOUT_R5/ VIN_D16 / 3.3 V<br>赤ビット5/入力データビット16 |
| 11   | GND  | 12 | GND  |
| 13   | VOUT_R6/ VIN_D17 / 3.3 V<br>赤ビット6/入力データビット17     | 14 | VOUT_R7/ VIN_D18 / 3.3 V<br>赤ビット7/入力データビット18 |
| 15   | VOUT_R8/ VIN_D19 / 3.3 V<br>赤ビット8/入力データビット19     | 16 | VOUT_R9/ VIN_D20 / 3.3 V<br>赤ビット9/入力データビット20 |
| 17   | GND  | 18 | GND  |
| 19   | VOUT_G2/ VIN_D21 / 3.3 V<br>緑ビット2/入力データビット21     | 20 | VOUT_G3/ VIN_D8 / 3.3 V<br>緑ビット3/入力データビット8   |
| 21   | VOUT_G4/ VIN_D9 / 3.3 V<br>緑ビット4/入力データビット9       | 22 | VOUT_G5/ VIN_D10 / 3.3 V<br>緑ビット5/入力データビット10 |
| 23   | +3.3V  | 24 | +3.3V  |
| 25   | VOUT_G6/ VIN_D11 / 3.3 V<br>緑ビット6/入力データビット11     | 26 | VOUT_G7/ VIN_D12 / 3.3 V<br>緑ビット7/入力データビット12 |
| 27   | VOUT_G8/ VIN_D13 / 3.3 V<br>緑ビット8/入力データビット13     | 28 | VOUT_G9/ VIN_D14 / 3.3 V<br>緑ビット9/入力データビット14 |
| 29   | GND  | 30 | GND  |
| 31   | VOUT_B2/ VIN_D7 / 3.3 V<br>青ビット2/入力データビット7       | 32 | VOUT_B3/ VIN_D0 / 3.3 V<br>青ビット3/入力データビット0   |
| 33   | VOUT_B4/ VIN_D1 / 3.3 V<br>青ビット4/入力データビット1       | 34 | VOUT_B5/ VIN_D2 / 3.3 V<br>青ビット5/入力データビット2   |
| 35   | GND  | 36 | GND  |
| 37   | VOUT_B6/ VIN_D3 / 3.3 V<br>青ビット6/入力データビット3       | 38 | VOUT_B7/ VIN_D4 / 3.3 V<br>青ビット7/入力データビット4   |
| 39   | VOUT_B8/ VIN_D5 / 3.3 V<br>青ビット8/入力データビット5       | 40 | VOUT_B9/ VIN_D6 / 3.3 V<br>青ビット9/入力データビット6   |

6 OPT1のディップスイッチ7でビデオの入力または出力を設定します; ON = ビデオ入力、OFF = ビデオ出力

**Video I/O | GPIO (ディップスイッチ8がOPT1 = ONの場合) | 2 x 20ピンヘッダー**

| ピン | シグナル                             | ピン | シグナル                             |
|----|----------------------------------|----|----------------------------------|
| 1  | GPIO92 / IO / 3.3 V<br>汎用ピン 92   | 2  | GPIO94 / IO / 3.3 V<br>汎用ピン 94   |
| 3  | GPIO95 / IO / 3.3 V<br>汎用ピン 95   | 4  | GPIO93 / IO / 3.3 V<br>汎用ピン 93   |
| 5  | GND                              | 6  | GND                              |
| 7  | GPIO22 / IO / 3.3 V<br>汎用ピン 22   | 8  | GPIO117 / IO / 3.3 V<br>汎用ピン 117 |
| 9  | GPIO110 / IO / 3.3 V<br>汎用ピン 110 | 10 | GPIO111 / IO / 3.3 V<br>汎用ピン 111 |
| 11 | GND                              | 12 | GND                              |
| 13 | GPIO112 / IO / 3.3 V<br>汎用ピン 112 | 14 | GPIO113 / IO / 3.3 V<br>汎用ピン 113 |
| 15 | GPIO114 / IO / 3.3 V<br>汎用ピン 114 | 16 | GPIO115 / IO / 3.3 V<br>汎用ピン 115 |
| 17 | GND                              | 18 | GND                              |
| 19 | GPIO116 / IO / 3.3 V<br>汎用ピン 116 | 20 | GPIO103 / IO / 3.3 V<br>汎用ピン 103 |
| 21 | GPIO104 / IO / 3.3 V<br>汎用ピン 104 | 22 | GPIO105 / IO / 3.3 V<br>汎用ピン 105 |
| 23 | GND                              | 24 | GND                              |
| 25 | GPIO106 / IO / 3.3 V<br>汎用ピン 106 | 26 | GPIO107 / IO / 3.3 V<br>汎用ピン 107 |
| 27 | GPIO108 / IO / 3.3 V<br>汎用ピン 108 | 28 | GPIO109 / IO / 3.3 V<br>汎用ピン 109 |
| 29 | GND                              | 30 | GND                              |
| 31 | GPIO126 / IO / 3.3 V<br>汎用ピン 126 | 32 | GPIO96 / IO / 3.3 V<br>汎用ピン 96   |
| 33 | GPIO97 / IO / 3.3 V<br>汎用ピン 97   | 34 | GPIO98 / IO / 3.3 V<br>汎用ピン 98   |
| 35 | GND                              | 36 | GND                              |
| 37 | GPIO99 / IO / 3.3 V<br>汎用ピン 99   | 38 | GPIO100 / IO / 3.3 V<br>汎用ピン 100 |
| 39 | GPIO101 / IO / 3.3 V<br>汎用ピン 101 | 40 | GPIO102 / IO / 3.3 V<br>汎用ピン 102 |



### TRIG & STR | トリガーとストロボ | 2 x 5ピンヘッダー

| ピン | シグナル  | ピン | シグナル  |
|----|---|----|---|
| 1  | GND   | 2  | PAS_TRIGIN- / 3...24 V / IN<br>パッシヴトリガー入力(負)    |
| 3  | ACT_STROBE / 5.0 V / OUT<br>アクティヴストロボ出力       | 4  | PAS_TRIGIN+ / 3...24 V / IN<br>パッシヴカメラトリガー入力(正) |
| 5  | +5.0 V  | 6  | +5.0 V  |
| 7  | PAS_STROBE+ / 3...24 V / OUT<br>パッシヴストロボ出力(正) | 8  | ACT_TRIGIN / 5.0 V / IN<br>アクティヴカメラトリガー入力       |
| 9  | PAS_STROBE- / 3...24 V / OUT<br>パッシヴストロボ出力(負) | 10 | GND   |

### ULED1 – ULED4 | GPIO 148～151によってコントロールされるユーザー定義可能

| ピン | シグナル                                 | ピン | シグナル                                 |
|----|--------------------------------------|----|--------------------------------------|
|    | GPIO148 / IO / 5.0 V<br>ULED1のコントロール |    | GPIO149 / IO / 5.0 V<br>ULED2のコントロール |
|    | GPIO150 / IO / 5.0 V<br>ULED3のコントロール |    | GPIO151 / IO / 5.0 V<br>ULED4のコントロール |

## 11.5 Linux簡易リファレンス

詳細なLinuxコマンドのリファレンスについては、インターネットで得られるリファレンス表のどれかをダウンロードすることを推奨します。Linuxコマンドのマニュアルページを見るには、`man <COMMAND>`とタイプします。

### ホットキー

`<Arrow up>` 最後のコマンドを反復

`<ctrl> + <r>` 最近のコマンドを検索

### ファイルコマンド

`cd example` サブディレクトリ“example”に変更

`cd example/dir` サブディレクトリ“example/dir”に変更

`cd` ホームディレクトリに変更

`cd /` ルートディレクトリに変更

`cd ..` 親ディレクトリに変更

`ls` または `ls -l` ディレクトリの内容を表示(簡易表示または詳細表示)

`mkdir example` 現在のディレクトリ内に新規“example”ディレクトリを作成

### ファイルの閲覧と編集

`nano file.xyz` Nanoエディタでファイル“file.xyz”を閲覧および/または編集

### ネットワーク

`ifconfig` ローカルマシン上のすべてのデバイスのIPアドレスを表示

`ping host` “host”に返答要求をして結果を出力

### 停止と開始

`halt -p` すべてのプロセスを停止して電源切断で終了

`reboot` すべてのプロセスを停止して再起動

### ユーザーとパスワード

`passwd` ユーザーパスワードを変更

`sudo <command>` ルートユーザー権限でシングルコマンドを実行

## 11.6 Linuxデバイス

### 備考

内部のeMMCメモリは、カメラのルートファイルシステムとLinuxカーネルを含んでいます。ルートファイルシステムは自動的に/にマウントされ、カーネルパーティションは/bootになるので、eMMCは/dev/mmcblk0によってアドレス指定されません。

| デバイス                         | Linuxデバイスファイル |
|------------------------------|---------------|
| CAN                          | can0          |
| I2C[2]                       | /dev/i2c-3    |
| Internal eMMC memory (備考を参照) | /dev/mmcblk0  |
| SDIO (SDcard)                | /dev/mmcblk1  |
| UART0                        | /dev/ttyO0    |
| UART1                        | /dev/ttyO1    |
| USB0                         | usb0          |
| USB1                         | usb1          |

## 12 索引

### A

Accessories 19  
Application development 59  
armhf packages 26  
ARM Toolchain 25

### B

Battery 8

### C

Camera architecture 9  
Camera models  
    Multi-sensor 12  
    Remote-sensor 11  
    Single-sensor 10  
CamLab 21, 43, 51  
Camserver 51  
CAN devices 89  
Core voltage 83  
CPU frequency 83  
Custom kernel 76

### D

Demo applications 59  
    ARM 60, 61  
    C# 65  
    DSP 61, 63  
    .NET 65  
DIP switches 18, 52, 75  
DSP support 64

### E

Eclipse 59  
Electrical characteristics 100  
ESD prevention 8  
Ethernet setup 42  
External sensor boards 31

### F

File exchange 48  
    MicroSD card 50  
    Network shares 48  
    scp command 49  
    USB flash drive 50  
Firmware 22, 91  
Firmware recovery 98  
Firmware update 90  
    Ethernet 91  
    SD Card 94

### G

Governors 83  
GPIOs 52, 53  
Graphics pipelines 74

### H

HALCON Embedded 67  
HALCON support 68

### I

I2C devices 88  
Interface boards 13  
IP address 44, 47

## K

Kernel sources: See Linux kernel sources

## L

Linux commands 122  
 Linux devices 123  
 Linux kernel sources 22, 29

## M

Memory  
     CMEM 9  
     DSP 9  
     Linux 9  
 multiarch-armhf 22

## P

Password 23  
     Camera Linux password 46  
     Change Camera Linux password 46  
 Pinout 101, 113  
 Power management 83  
 PuTTY 44

## R

Real-time clock 86  
 Recovery 90, 98

## S

Safety 8  
 scp command 49  
 Serial connection 45  
 Serial number 23  
 Software installation 21  
 SPI devices 87  
 SSH 44  
 sudo 122  
 Switch on and off 58

## T

Third-party software 55  
 TI EZSDK 24, 27, 28, 63  
 Trigger and strobe 55

## U

USB  
     Device Only 79  
     Host/Device 77  
     Host/Host 77  
 USB controller 18, 75  
 USB driver 75

## V

Video mode 74  
 Video output 73  
 Virtual machine 22, 23  
 VRmagic Custom Ubuntu 21, 22  
 VRmagic D3 EZSDK 22, 24, 27, 29, 63  
 VRmagic Easy Installer 21, 23  
 VRmagic Linux D3 Camera SDK 22  
 VRmagic Linux PC Camera Runtime 21  
 VRmagic Linux PC Camera SDK 22  
 VRmagic repository 54  
 VRmCUEO3 13, 14, 19, 33, 101  
     Cable Plan 108

|                    |                        |
|--------------------|------------------------|
| CAN Bus            | 34                     |
| Ethernet           | 33                     |
| External Battery   | 36                     |
| GPIOs              | 34                     |
| HDMI               | 36                     |
| MicroSD            | 34                     |
| Power Supply       | 37                     |
| RS232              | 34                     |
| Serial Console     | 34                     |
| SPI                | 36                     |
| S-Video            | 34                     |
| Trigger and Strobe | 35                     |
| UART1              | 34                     |
| USB                | 35                     |
| Watchdog           | 36                     |
| VRmEIO3            | 8, 13, 15, 20, 37, 113 |
| Audio              | 38                     |
| Ethernet           | 37                     |
| HDMI               | 38                     |
| JTAG               | 40                     |
| Pin headers        | 39                     |
| Power supply       | 41                     |
| SATA               | 39                     |
| Serial ports       | 39                     |
| S-Video            | 38                     |
| USB                | 38                     |

## W

|      |    |
|------|----|
| WiFi | 80 |
|------|----|



VRmagic GmbH  
Augustaanlage 32  
68165 Mannheim  
Germany  
Phone +49 621 400 416-20  
Fax +49 621 400 416-99

[info.imaging@vrmagic.com](mailto:info.imaging@vrmagic.com)  
[www.vrmagic-imaging.com](http://www.vrmagic-imaging.com)

© 2014 VRmagic GmbH, Mannheim