

i.MX 6UL スタータボード
GOHSSC-6UL

ソフトウェアマニュアル

本資料に記載の全ての情報は本資料発行時点のものであり、予告なしに、本資料に記載した製品または仕様を変更することがあります。

安全に関する注意事項

- 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
- 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
- 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
- 当社製品のマニュアルに記載以外の改造、改変、複製等しないでください。改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
- 当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。
- 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
- 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、ソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
- ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
- 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
- お客様の転売等により、本安全に関する注意事項記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
- 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

取り扱い上の注意事項

- 水・湿気・ほこり・油煙等の多い場所に設置しないでください。火災、故障、感電などの原因になる場合があります。
- 本製品に搭載されている部品の一部は、発熱により高温になる場合があります。周囲温度や取扱いによってはやけどの原因となる恐れがあります。本体の電源が入っている間、または電源切断後本体の温度が下がるまでの間は、基板上の電子部品、及びその周辺部分には触れないでください。
- 本製品には CMOS デバイスを使用しており、静電気により破壊されるおそれがあります。本製品を開封するときは、低湿度状態にならないよう注意し、静電防止用マットの使用、導電靴や人体アースなどによる作業者の帯電防止対策、備品の放電対策、静電気対策を施された環境下で行ってください。また、本製品を保管する際は、静電気を帯びやすいビニール袋やプラスチック容器などは避け、導電袋や導電性の容器・ラックなどに収納してください。

商標および登録商標について

- すべての商標および登録商標は、それぞれの所有者に帰属します。

目次

目次.....	4
1. 概要.....	6
1.1 概要.....	6
1.2 構成.....	7
1.3 外部仕様.....	8
1.4 外観.....	10
1.5 ブロック図.....	11
1.6 部品配置図.....	12
1.7 メモリ配置図.....	16
1.8 絶対最大定格.....	17
1.9 動作条件.....	17
2. 機能仕様.....	18
2.1 機能概要.....	18
2.2 SoC.....	19
2.2.1 i.MX6UL概要.....	19
2.2.2 GOHSSC-6UL端子機能一覧.....	19
2.2.3 GOHSSC-6UL搭載デバイス使用関係一覧.....	37
2.3 ソフトウェア構成.....	38
3. 開発環境構築.....	39
3.1 準備するもの.....	39
3.2 環境構築手順.....	39
3.3 フォルダ構成.....	39
3.4 Yocto Project ビルド環境インストール.....	40
3.5 Git管理ツールrepoインストール.....	40
3.6 ツールチェイン入手.....	41
3.7 ソフトウェアビルド.....	42
3.7.1 ソフトウェア解凍.....	42
3.7.2 u-bootビルド.....	42
3.7.3 カーネルビルド.....	43
3.8 ルートファイルシステム入手.....	44
4. ソフトウェア起動手順.....	45
4.1 準備するもの.....	45
4.2 起動までの手順.....	45
4.3 接続方法.....	46
4.4 SDカード作成.....	47
4.4.1 パーティション作成.....	48
4.4.2 ブートローダ格納.....	49

4.4.3	カーネル格納.....	49
4.4.4	ルートファイルシステム格納.....	50
4.5	ブートモード設定.....	51
4.6	シリアル通信ソフトウェアの設定.....	51
4.7	起動.....	52
4.8	MACアドレス設定.....	53
5.	u-boot操作手順.....	54
5.1	DDR.....	54
5.2	QSPI NOR FLASH.....	55
5.3	NAND FLASH.....	56
5.4	SD/MMC.....	58
5.5	I ² C.....	60
5.6	Ethernet.....	61
5.7	USB.....	62
6.	Linux操作手順.....	64
6.1	QSPI NOR FLASH.....	64
6.2	NAND FLASH.....	65
6.3	SDカード.....	66
6.4	e-MMC.....	66
6.5	Ethernet.....	68
6.6	USB.....	71
6.7	オーディオ.....	71
6.8	CAN.....	71
7.	拡張機能.....	72
7.1	HDMI.....	72
7.2	NAND FLASH.....	73
7.3	USB2.0 Micro AB.....	74
7.4	センサ.....	75
7.5	e-MMCブート.....	75

1. 概要

1.1 概要

i.MX 6UL スタータボード GOHSSC-6UL（以下 GOHSSC-6UL と称す）は、NXP セミコンダクターズ製アプリケーションプロセッサ i.MX 6UltraLite-3（以下 i.MX 6UL と称す）を搭載したシステム開発向けスタータボードです。

i.MX 6UL は、ARM Cortex®-A7 シングルコアを搭載した高性能および低消費電力を実現した SoC (System on Chip) であり、POS (Point Of Sales system) 端末等の携帯機器向けの用途に最適なアプリケーションプロセッサです。

以下に GOHSSC-6UL の特長を示します。

- 機能としては、4 種類のメモリ (DDR3L SDRAM, e-MMC, QSPI NOR FLASH, EEPROM) ※、複数の通信系インタフェース (USB, Ethernet, CAN, UART)、映像/オーディオのマルチメディアを搭載しています。また各種センサ (磁気センサ、加速度センサ、ジャイロ스코ープ) を搭載しており、様々な応用分野にあわせたアプリケーション開発が可能です。
- 標準 OS に Linux (Kernel Ver.3.14.52) を採用しており、豊富なソフトウェア資産を利用して、多様な機能を効率的に開発することができます。
- i.MX シリーズ向けの NXP セミコンダクターズ製パワーマネジメント IC である PF3000 を搭載しているため、i.MX 6UL 単体の機能・性能評価だけでなく、システムとしてアプリケーションソフトウェアの先行開発・評価を行うことができます。
- ユーザデバッグインタフェースとの接続用に、ARM JTAG 20 コネクタを実装しています。

※4 種類のメモリ (DDR3L SDRAM, e-MMC, QSPI NOR FLASH, EEPROM) を搭載した GOHSSC-6UL と 3 種類のメモリ (DDR3L SDRAM, NAND FLASH, EEPROM) を搭載した GOHSSC-6UL-B (オプション扱い) があります。メモリ以外の機能については、GOHSSC-6UL と GOHSSC-6UL-B の違いはありません。

1.2 構成

図1.1に GOHSSC-6UL を用いたシステム構成例を示します。

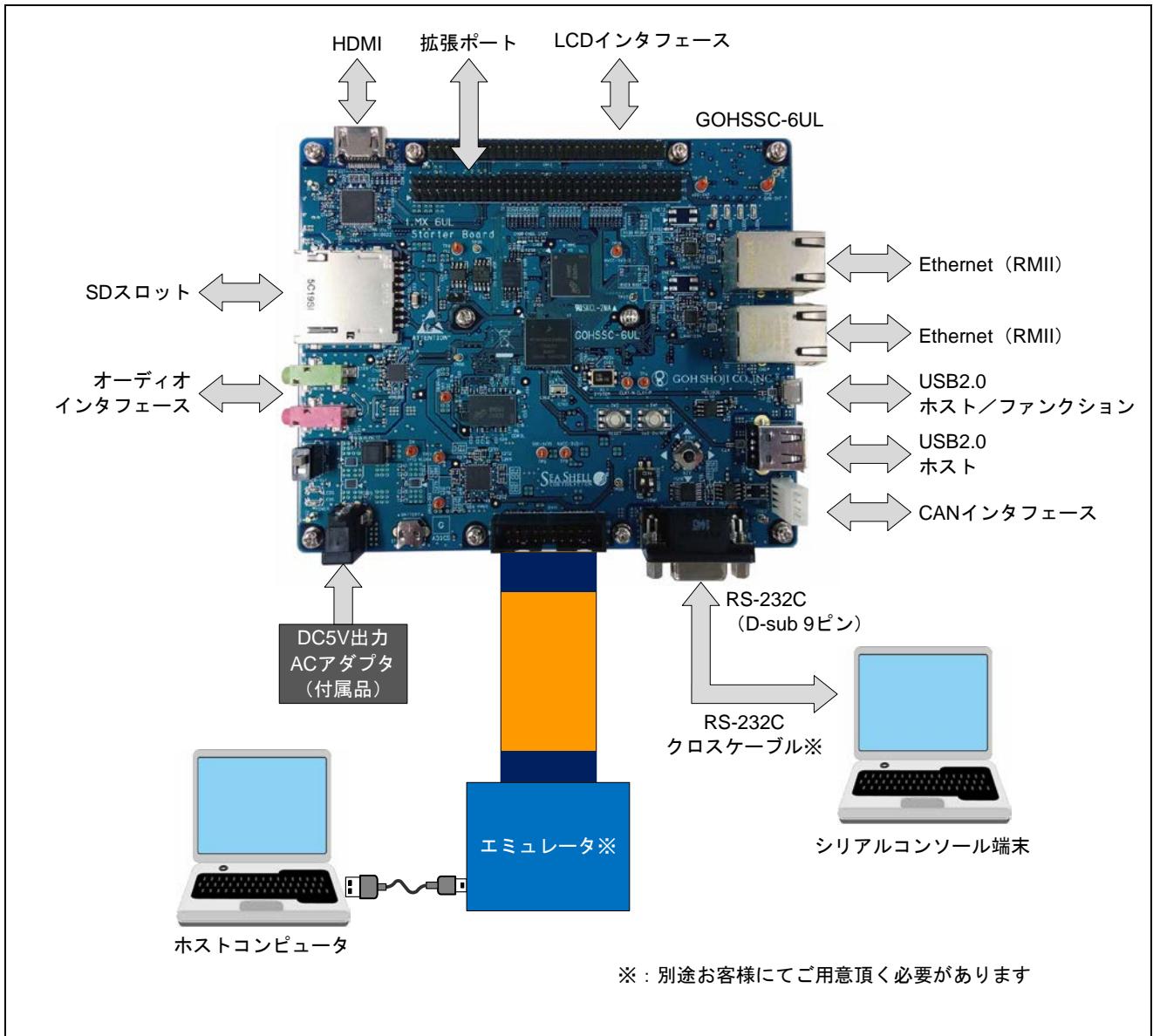


図1.1 GOHSSC-6ULを用いたシステム構成例

1.3 外部仕様

表1.1、表1.2に GOHSSC-6UL の外部仕様一覧を示します。

表1.1 GOHSSC-6UL仕様一覧 (1)

項目	内容
SoC	<ul style="list-style-type: none"> • i.MX 6UL-3 ARM Cortex®A7 シングルコア 入力 (XTAL) クロック : 24MHz システムクロック : 最大 528MHz USB クロック : 480MHz DDR クロック : 最大 396MHz SD クロック : 最大 192MHz RTC クロック : 32.768kHz 289 ピン BGA 14mm×14mm 0.8mm ピッチ
メモリ	<ul style="list-style-type: none"> • DDR3L SDRAM : 512M バイト×2 (合計 1G バイト) バス幅 : 16 ビット • QSPI NOR FLASH : 32M バイト×1 • e-MMC : 8G バイト×1 バス幅 : 8 ビット • EEPROM : 4K バイト×1 • NAND FLASH : 4G バイト×1 (オプション扱い) ※
コネクタ	<ul style="list-style-type: none"> • SD スロット ×1 • USB2.0 TypeA コネクタ ×1 • USB2.0 MicroAB コネクタ ×1 • CAN コネクタ ×1 • RS-232C コネクタ ×1 • Ethernet コネクタ (RMII) ×2 • ヘッドフォンコネクタ (φ3.5) ×1 • マイクコネクタ (φ3.5) ×1 • HDMI コネクタ ×1 • ARM JTAG 20 コネクタ ×1 • LCD コネクタ ×1 • 電源コネクタ ×1 • 拡張コネクタ ×1 • ボタン電池ソケット ×1
センサ	<ul style="list-style-type: none"> • 磁気センサ ×1 (デフォルト無効) • 加速度センサ ×1 (デフォルト無効) • ジャイロスコープ ×1 (デフォルト無効)
LED	<ul style="list-style-type: none"> • ユーザ LED (黄) ×3 • 電源 LED (緑) ×1 • 過電圧検出用 LED (赤) ×1 • LAN 通信用 LED (橙) ×2

※NAND FLASH を搭載している GOHSSC-6UL-B では、QSPI NOR FLASH および e-MMC が非搭載になります。

表1.2 GOHSSC-6UL仕様一覧 (2)

項目	内容
スイッチ	<ul style="list-style-type: none">電源 ON/OFF スイッチ ×1リセットスイッチ ×1電源スイッチ ×1ブートモード切り替え 2 ピンディップスイッチ ×1ブートモード切り替え 8 ピンディップスイッチ ×3
その他	<ul style="list-style-type: none">寸法 : 150mm × 125mm実装形態 : 8 層 両面実装基板構成 : 1 枚付属品 : AC アダプタ ×1

1.4 外観

図1.2に GOHSSC-6UL の外観を示します。

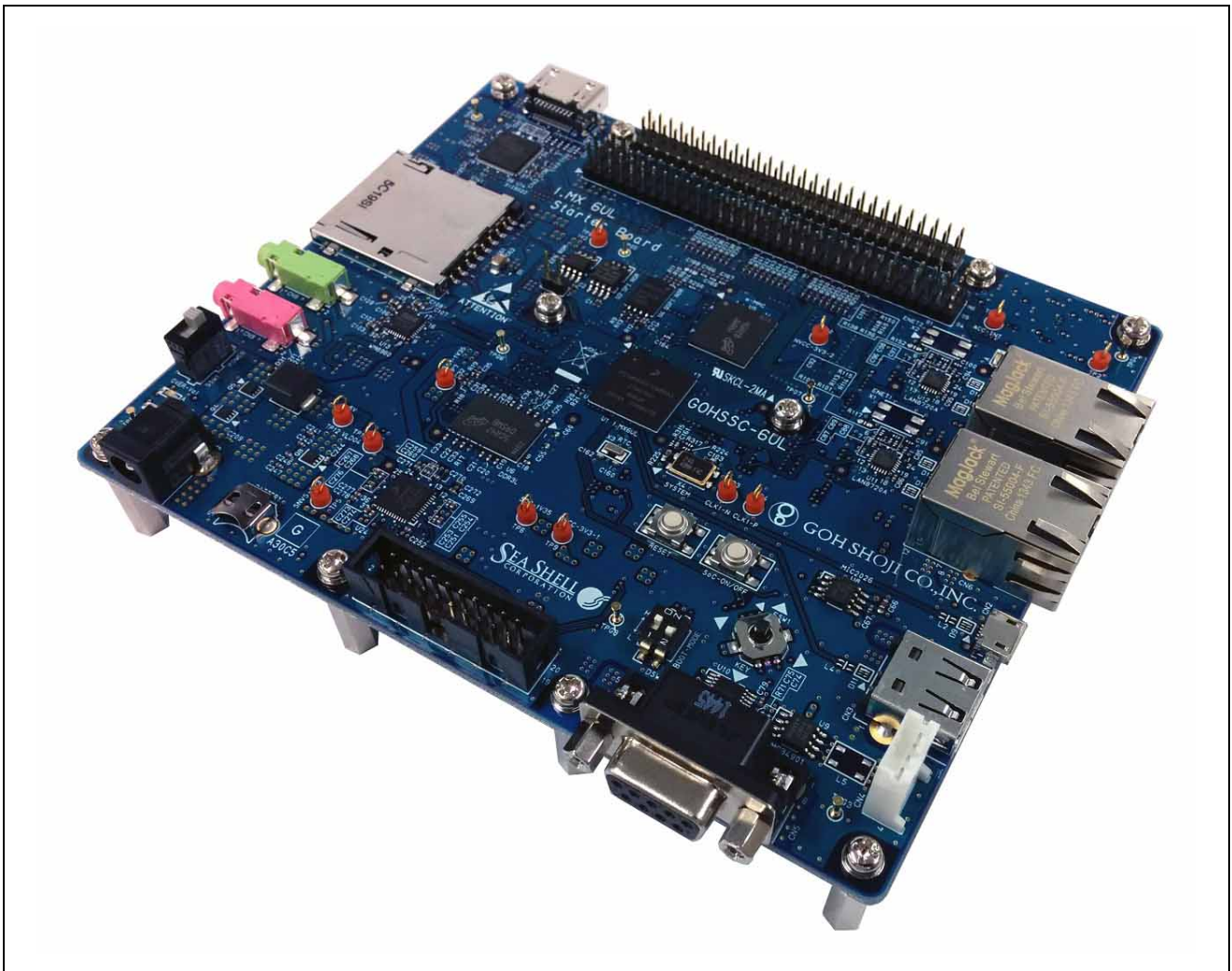


図1.2 GOHSSC-6UL外観図

1.5 ブロック図

図1.3に GOHSSC-6UL のブロック図を示します。

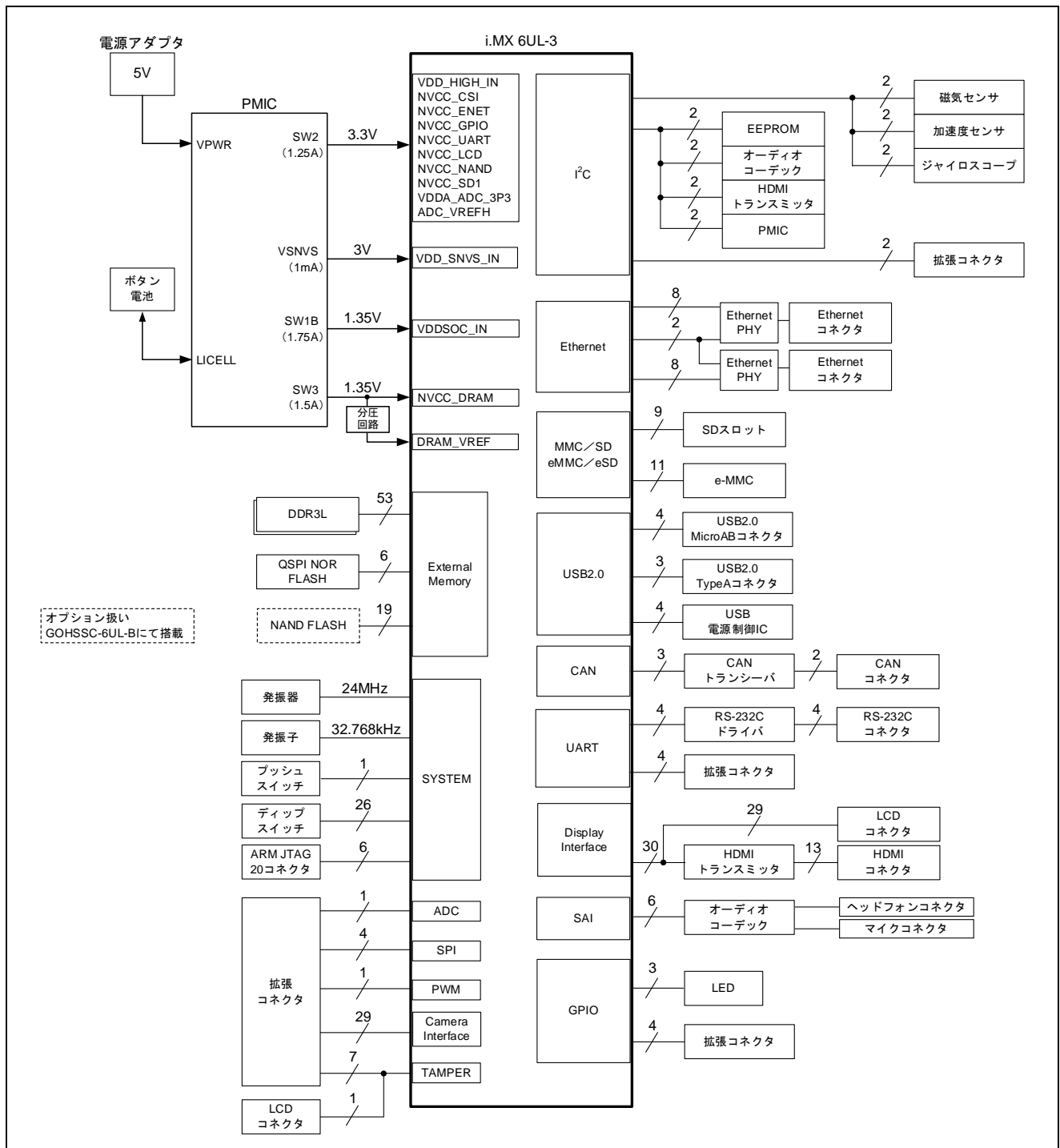


図1.3 ブロック図

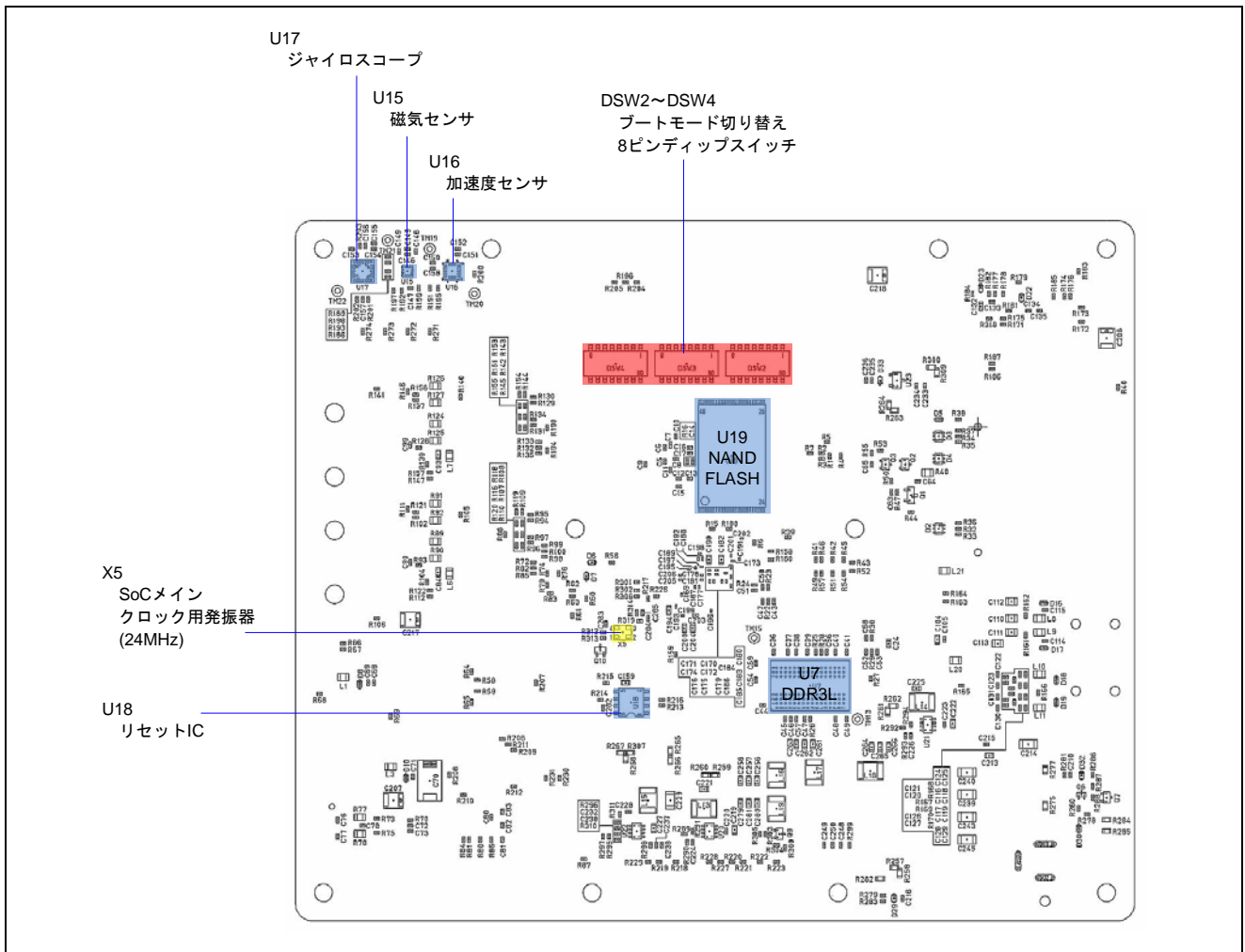


図1.5 GOHSSC-6UL部品配置図 (S面上面図)

表1.3～表1.5に GOHSSC-6UL の主な実装部品一覧を示します。

表1.3 GOHSSC-6ULの主な実装部品一覧 (IC)

部品番号	部品名称	型名、メーカー	備考
U1	SoC	i.MX 6UL-3「MCIMX6G3DVM05AA」 (NXP)	
U3	EEPROM	M24C32-RMN6TP (ST Micro)	
U4	QSPI NOR FLASH	N25Q256A13EF840 (Micron)	
U5	e-MMC	MTFC8GACAAAM (Micron)	
U6, U7	DDR3L SDRAM	MT41K256M16HA-125IT:E (Micron)	
U8	USB 電源制御 IC	MIC2026-1YM (Micrel)	
U9	CAN トランシーバ	MC34901WEF (NXP)	
U10	RS-232C ドライバ	SP3232EEY-L (Exar)	
U11, U12	Ethernet トランシーバ	LAN8720Ai-CP-TR (Microchip)	
U13	オーディオコーデック	WM8960 (Wolfson)	
U14	HDMI トランスミッタ	SiI9022ACNU (Lattice)	
U15	磁気センサ	MAG3110FCR1 (NXP)	
U16	加速度センサ	FXLS8471QR1 (NXP)	
U17	ジャイロスコープ	FXAS21002CQR1 (NXP)	
U18	リセット IC	RNA51957BFP (Renesas)	
U19	NAND FLASH	MT29F32G08CBADBWP-12IT:D (Micron)	オプション扱い※
U24	PMIC	PC34PF3000A3EP (NXP)	
X1, X2	Ethernet 用発振子	ABMM-25.000MHz-B2 (Abracon)	
X3	RTC 用発振子	ABS07-32.768kHz-T (Abracon)	
X5	SoC メインクロック用発振器	AU-24.000MBE-T (Txc)	

※NAND FLASH を搭載している GOHSSC-6UL-B では、QSPI NOR FLASH および e-MMC が非搭載になります。

表1.4 GOHSSC-6ULの主な実装部品一覧（コネクタ）

部品番号	部品名称	型名、メーカー	備考
CN1	SD スロット	DM1AA-SF-PEJ（ヒロセ電機）	
CN2	USB2.0 MicroAB コネクタ	10104111-0001LF（JST）	
CN3	USB2.0 TypeA コネクタ	UBA-4R-D14T-4D(LF)(SN)（JST）	
CN4	CAN コネクタ	BH4B-PH（JST）	
CN5	RS-232C コネクタ	1734354-1（TE）	
CN6, CN7	Ethernet コネクタ	SI-55004-F（Stewart Connector）	
CN8	ヘッドフォンコネクタ	SJ-3524-SMT-TR-GR（CUI Inc）	
CN9	マイクコネクタ	SJ-3524-SMT-TR-PI（CUI Inc）	
CN10	HDMI コネクタ	10029449-001RLF（FCI）	
CN11	ARM JTAG 20 コネクタ	1761681-7（Molex）	
CN12	LCD コネクタ	FFC-56BMEP1B（本多通信工業）	
CN13	電源コネクタ	PJ-050AH（CUI Inc）	
JP2	拡張コネクタ	PSS-430256-32（廣杉計器）	
B1	ボタン電池ソケット	2999（KEystone）	

表1.5 GOHSSC-6ULの主な実装部品一覧（その他機構部品）

部品番号	部品名称	型名、メーカー	備考
JP1	EEPROM ライト プロテクト用ジャンパピン	FFC-2ASM1B（本多通信工業）	
SW1	電源 ON/OFF スイッチ	B3SN-3112P（オムロン）	
SW2	リセットスイッチ	B3SN-3112P（オムロン）	
DSW1	ブートモード切り替え 2 ピンディップスイッチ	A6S-2104H（オムロン）	
DSW2~DSW4	ブートモード切り替え 8 ピンディップスイッチ	A6H-8101（オムロン）	
SSW1	電源スイッチ	AS-12AP（NKK スイッチズ）	
LED1~LED3	ユーザ LED	SML-310YTT86（ローム）	黄色
LED5	過電圧検出用 LED	SML-310VTT86（ローム）	赤色
LED6	電源 LED	SML-310MTT86（ローム）	緑色

1.7 メモリ配置図

表1.6に GOHSSC-6UL での i.MX 6UL-3 のメモリマップを示します。

表1.6 メモリマップ

Start Address	End Address	i.MX 6UL ※	GOHSSC-6UL
0x0000 0000	0x0001 7FFF	Boot ROM	
0x0001 8000	0x000F FFFF	Reserved	
0x0010 0000	0x0010 FFFF	CAAM	
0x0011 0000	0x008F FFFF	Reserved	
0x0090 0000	0x0097 FFFF	OCRAM	
0x0098 0000	0x009F FFFF	Reserved	
0x00A0 0000	0x00A0 7FFF	ARM Peripherals	
0x00A0 8000	0x00AF FFFF	Reserved	
0x00B0 0000	0x00EF FFFF	configuration port	
0x00F0 0000	0x0180 3FFF	Reserved	
0x0180 4000	0x0180 BFFF	APBH DMA	
0x0180 C000	0x01FF FFFF	Reserved	
0x0200 0000	0x021F FFFF	i.MX6UL Internal Register	
0x0220 0000	0x0BFF FFFF	Reserved	
0x0C00 0000	0x0DFF FFFF	QSPI1 Rx Buffer	
0x0E00 0000	0x0FFF FFFF	Reserved	
0x1000 0000	0x4FFF FFFF	AES Encryption	
0x5000 0000	0x57FF FFFF	EIM CS (NOR/SRAM)	
0x5800 0000	0x5FFF FFFF	EIM Aliased	
0x6000 0000	0x6FFF FFFF	QSPI1 Memory	QSPI NOR FLASH
0x7000 0000	0x7FFF FFFF	Reserved	
0x8000 0000	0xFFFF FFFF	MMDC	DDR3L SDRAM

※i.MX 6UL 仕様の詳細は、「i.MX 6UltraLite Applications Processor Reference Manual」を参照してください。

1.8 絶対最大定格

表1.7に GOHSSC-6UL の絶対最大定格を示します。

表1.7 GOHSSC-6ULの絶対最大定格

記号	項目	定格値	備考
5V	5V系電源電圧	-0.3V~7.5V	Vss基準
Tstg	保存周囲温度	-10°C~80°C	湿度90%以下（結露無きこと） 腐食性ガス環境は不可

1.9 動作条件

表1.8に GOHSSC-6UL の動作条件を示します。

表1.8 GOHSSC-6ULの動作条件

記号	項目	定格値	備考
5V	5V系電源電圧	4.75V~5.25V	Vss基準
—	最大消費電流	3A以下	5V系、3.3V系、1.35V系、1.2V系電源の合計値
Topr	動作周囲温度	0°C~70°C	湿度90%以下（結露無きこと） 腐食性ガス環境は不可

周囲温度とは、ボードに限りなく近い部分の空気温度のことです。

2. 機能仕様

2.1 機能概要

表2.1に GOHSSC-6UL の機能モジュール一覧を示します。

表2.1 GOHSSC-6UL機能モジュール一覧

機能	内容
SoC	<ul style="list-style-type: none"> i.MX 6UL-3 ARM Cortex®A7 シングルコア 入力 (XTAL) クロック : 24MHz システムクロック : 最大 528MHz DDR クロック : 最大 396MHz
メモリ	<ul style="list-style-type: none"> DDR3L SDRAM : 512M バイト×2 バス幅 : 16 ビット QSPI NOR FLASH : 32M バイト×1 e-MMC : 8G バイト×1 バス幅 : 8 ビット EEPROM : 4K バイト×1 NAND FLASH : 4G バイト×1 (オプション扱い) ※
Ethernet (RMII) インタフェース	<ul style="list-style-type: none"> i.MX 6UL Ethernet コントローラ (ENET) と Ethernet PHY の接続
USB インタフェース	<ul style="list-style-type: none"> i.MX 6UL USB2.0 ホスト/ファンクションモジュールと USB コネクタの接続
SD ホストインタフェース	<ul style="list-style-type: none"> i.MX 6UL Ultra Secure Digital Host Controller (uSDHC) と SD スロットの接続
LCD インタフェース	<ul style="list-style-type: none"> i.MX 6UL LCD インタフェース (LCDIF) と HDMI トランスミッタおよび LCD コネクタの接続
拡張インタフェース	<ul style="list-style-type: none"> i.MX 6UL 入出力ポート (GPIO) と拡張コネクタの接続
オーディオインタフェース	<ul style="list-style-type: none"> i.MX 6UL Synchronous Audio Interface (SAI) とオーディオコーデックの接続
CAN インタフェース	<ul style="list-style-type: none"> i.MX 6UL Flexible Controller Area Network (FLEXCAN) と CAN コネクタの接続
シリアルインタフェース	<ul style="list-style-type: none"> i.MX 6UL UART インタフェース (UART) と RS-232 トランシーバの接続
JTAG インタフェース	<ul style="list-style-type: none"> i.MX 6UL システム JTAG コントローラ (SJC) と ARM JTAG 20 コネクタの接続
センサ	<ul style="list-style-type: none"> 磁気センサ ×1 (デフォルト無効) 加速度センサ ×1 (デフォルト無効) ジャイロ스코ープ ×1 (デフォルト無効)
電源構成	<ul style="list-style-type: none"> GOHSSC-6UL のシステム電源構成
リセット制御	<ul style="list-style-type: none"> GOHSSC-6UL に実装されているデバイスのリセット制御
クロック構成	<ul style="list-style-type: none"> システムクロック構成
操作仕様	<ul style="list-style-type: none"> コネクタ、スイッチ、ジャンパ、LED 第3章にて詳細説明

※NAND FLASH を搭載している GOHSSC-6UL-B では、QSPI NOR FLASH および e-MMC が非搭載になります。

2.2 SoC

2.2.1 i.MX6UL 概要

GOHSSC-6UL は、ARM Cortex®-A7 シングルコア内蔵のアプリケーションプロセッサ i.MX 6UL を搭載しています。

2.2.2 GOHSSC-6UL 端子機能一覧

表2.2～表2.19に GOHSSC-6UL で使用する i.MX 6UL 端子の割付と用途を示します。

表2.2 i.MX 6UL-3端子割付 (1)

端子	i.MX 6UL 端子機能	方向	接続先	用途
A1	VSS	—	GND	GND
A2	USDHC1_DATA3 ／GPT2_CAPTURE2／ SAI2_TX_DATA／FLEXCAN2_RX／EIM_ADDR24／ GPIO2_IO21／CCM_CLKO2／ANATOP_OTG2_ID	In／ Out	SD スロット	SD スロット用 データ信号
A3	RAWNAND_READY_B ／USDHC1_DATA4／ QSPI_A_DATA00 ／ECSPI3_SS0／EIM_CS1_B／ GPIO4_IO12 ／UART3_TX	In／ Out	QSPI NOR FLASH／ 加速度センサ／ 拡張コネクタ	QSPI NOR FLASH 用 データ信号／ センサ割り込み信号
A4	RAWNAND_CLE ／USDHC1_DATA7／ QSPI_A_DATA03 ／ECSPI3_MISO／EIM_ADDR16／ GPIO4_IO15 ／UART3_RTS_B	In／ Out	QSPI NOR FLASH／ 拡張コネクタ	QSPI NOR FLASH 用 データ信号／ 汎用ポート
A5	RAWNAND_DATA07 ／ USDHC2_DATA7 ／ QSPI_A_SS1_B／ECSPI4_SS0／EIM_AD15／ GPIO4_IO09／UART2_RTS_B	In／ Out	e-MMC	e-MMC 用データ信号
A6	RAWNAND_DATA06 ／ USDHC2_DATA6 ／ SAI2_RX_BCLK／ECSPI4_MISO／EIM_AD14／ GPIO4_IO08／UART2_CTS_B	In／ Out	e-MMC	e-MMC 用データ信号
A7	RAWNAND_DATA02 ／ USDHC2_DATA2 ／ QSPI_B_DATA00／KPP_ROW02／EIM_AD10／ GPIO4_IO04／ECSPI4_SS2	In／ Out	e-MMC	e-MMC 用データ信号
A8	LCDIF_CLK ／LCDIF_WR_RWN／UART4_TX／ SAI3_MCLK／EIM_CS2_B／GPIO3_IO00／ WDOG1_WDOG_RST_B_DEB	Out	HDMI／LCD コネクタ	LCDIF 用クロック信号
A9	LCDIF_DATA01 ／PWM2_OUT／ ENET1_1588_EVENT2_OUT／I2C3_SCL／ GPIO3_IO06／SRC_BT_CFG01／SAI1_TX_SYNC	Out	HDMI／LCD コネクタ	LCDIF 用データ信号

赤字：GOHSSC-6UL デフォルト使用機能

青字：GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字：GOHSSC-6UL-B 使用機能（**青字**機能は使用できません）

表2.3 i.MX 6UL-3端子割付 (2)

端子	i.MX 6UL 端子機能	方向	接続先	用途
A10	LCDIF_DATA06 ／UART7_CTS_B／ ENET2_1588_EVENT3_IN／SPDIF_LOCK／ GPIO3_IO11／SRC_BT_CFG06／ECSPI1_SS2	Out	HDMI／LCD コネクタ	LCDIF 用データ信号
A11	LCDIF_DATA09 ／SAI3_MCLK／CSI_DATA17／ EIM_DATA01／GPIO3_IO14／SRC_BT_CFG09／ FLEXCAN1_RX	Out	HDMI／LCD コネクタ	LCDIF 用データ信号
A12	LCDIF_DATA14 ／SAI3_RX_DATA／CSI_DATA22／ EIM_DATA06／GPIO3_IO19／SRC_BT_CFG14／ USDHC2_DATA4	Out	HDMI／LCD コネクタ	LCDIF 用データ信号
A13	LCDIF_DATA18 ／PWM5_OUT／ CA7_MX6UL_EVENT0／CSI_DATA10／ EIM_DATA10／GPIO3_IO23／SRC_BT_CFG26／ USDHC2_CMD	Out	HDMI／LCD コネクタ	LCDIF 用データ信号
A14	LCDIF_DATA22 ／MQS_RIGHT／ECSPI1_MOSI／ CSI_DATA14／EIM_DATA14／GPIO3_IO27／ SRC_BT_CFG30／USDHC2_DATA2	Out	HDMI／LCD コネクタ	LCDIF 用データ信号
A15	ENET2_TDATA00 ／UART7_RX／ SIM1_PORT0_SVEN／ I2C4_SDA ／EIM_EB_B02／ GPIO2_IO11／KPP_COL05	In／ Out	Ethernet PHY2／ センサ	Ethernet 用データ信号 ／センサ制御
A16	ENET2_TDATA01 ／UART8_TX／ SIM2_PORT0_TRXD／ ECSPI4_SCLK ／ EIM_EB_B03／GPIO2_IO12／KPP_ROW06／ USB_OTG2_PWR	Out	Ethernet PHY2／ LCD コネクタ／ 拡張コネクタ	Ethernet 用データ信号 ／拡張シリアルポート
A17	VSS	—	GND	GND
B1	USDHC1_DATA2 ／GPT2_CAPTURE1／ SAI2_RX_DATA／FLEXCAN2_TX／EIM_ADDR23／ GPIO2_IO20／CCM_CLKO1／USB_OTG2_OC	In／ Out	SD スロット	SD スロット用 データ信号
B2	USDHC1_DATA1 ／GPT2_CLK／SAI2_TX_BCLK／ FLEXCAN1_RX／EIM_ADDR22／GPIO2_IO19／ USB_OTG2_PWR	In／ Out	SD スロット	SD スロット用 データ信号
B3	USDHC1_DATA0 ／GPT2_COMPARE3／ SAI2_TX_SYNC／FLEXCAN1_TX／EIM_ADDR21／ GPIO2_IO18／ANATOP_OTG1_ID	In／ Out	SD スロット	SD スロット用 データ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えて使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.4 i.MX 6UL-3端子割付 (3)

端子	i.MX 6UL 端子機能	方向	接続先	用途
B4	RAWNAND_ALE / USDHC2_RESET_B / QSPI_A_DQS / PWM3_OUT / EIM_ADDR17 / GPIO4_IO10 / ECSPi3_SS1	Out	e-MMC	e-MMC 用 リセット信号
B5	RAWNAND_CE1_B / USDHC1_DATA6 / QSPI_A_DATA02 / ECSPi3_MOSI / EIM_ADDR18 / GPIO4_IO14 / UART3_CTS_B	In / Out	QSPI NOR FLASH / ジャイロスコープ / 拡張コネクタ	QSPI NOR FLASH 用 データ信号 / センサ割り込み信号
B6	RAWNAND_DATA05 / USDHC2_DATA5 / QSPI_B_DATA03 / ECSPi4_MOSI / EIM_AD13 / GPIO4_IO07 / UART2_RX	In / Out	e-MMC	e-MMC 用データ信号
B7	RAWNAND_DATA01 / USDHC2_DATA1 / QSPI_B_DQS / KPP_COL01 / EIM_AD09 / GPIO4_IO03 / ECSPi4_SS1	In / Out	e-MMC	e-MMC 用データ信号
B8	LCDIF_ENABLE / LCDIF_RD_E / UART4_RX / SAI3_TX_SYNC / EIM_CS3_B / GPIO3_IO01 / ECSPi2_RDY	Out	HDMI / LCD コネクタ	LCDIF 用 イネーブル信号
B9	LCDIF_DATA00 / PWM1_OUT / ENET1_1588_EVENT2_IN / I2C3_SDA / GPIO3_IO05 / SRC_BT_CFG00 / SAI1_MCLK	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
B10	LCDIF_DATA05 / UART8_RTS_B / ENET2_1588_EVENT2_OUT / SPDIF_OUT / GPIO3_IO10 / SRC_BT_CFG05 / ECSPi1_SS1	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
B11	LCDIF_DATA08 / SPDIF_IN / CSI_DATA16 / EIM_DATA00 / GPIO3_IO13 / SRC_BT_CFG08 / FLEXCAN1_TX	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
B12	LCDIF_DATA13 / SAI3_TX_BCLK / CSI_DATA21 / EIM_DATA05 / GPIO3_IO18 / SRC_BT_CFG13 / USDHC2_RESET_B	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
B13	LCDIF_DATA17 / UART7_RX / CSI_DATA00 / EIM_DATA09 / GPIO3_IO22 / SRC_BT_CFG25 / USDHC2_DATA7	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
B14	LCDIF_DATA21 / UART8_RX / ECSPi1_SS0 / CSI_DATA13 / EIM_DATA13 / GPIO3_IO26 / SRC_BT_CFG29 / USDHC2_DATA1	Out	HDMI / LCD コネクタ	LCDIF 用データ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.5 i.MX 6UL-3端子割付 (4)

端子	i.MX 6UL 端子機能	方向	接続先	用途
B15	ENET2_TX_EN / UART8_RX / SIM2_PORT0_CLK / ECSPI4_MOSI / EIM_ACLK_FREERUN / GPIO2_IO13 / KPP_COL06 / USB_OTG2_OC	Out	Ethernet PHY2 / LCD コネクタ / 拡張コネクタ	Ethernet 用送信イネー ブル信号 / 拡張シリアルポート
B16	LCDIF_DATA23 / MQS_LEFT / ECSPI1_MISO / CSI_DATA15 / EIM_DATA15 / GPIO3_IO28 / SRC_BT_CFG31 / USDHC2_DATA3	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
B17	ENET2_RX_EN / UART7_TX / SIM1_PORT0_RST_B / I2C4_SCL / EIM_ADDR26 / GPIO2_IO10 / KPP_ROW05 / ENET1_REF_CLK_25M	In / Out	Ethernet PHY2 / センサ	Ethernet 用受信イネー ブル信号 / センサ制御
C1	USDHC1_CLK / GPT2_COMPARE2 / SAI2_MCLK / SPDIF_IN / EIM_ADDR20 / GPIO2_IO17 / USB_OTG1_OC	Out	SD スロット	SD スロット用 クロック信号
C2	USDHC1_CMD / GPT2_COMPARE1 / SAI2_RX_SYNC / SPDIF_OUT / EIM_ADDR19 / GPIO2_IO16 / SDMA_EXT_EVENT00 / USB_OTG1_PWR	Out	SD スロット	SD スロット用 コマンド信号
C3	VSS	—	GND	GND
C4	NVCC_SD1	—	3.3V	電源
C5	RAWNAND_CE0_B / USDHC1_DATA5 / QSPI_A_DATA01 / ECSPI3_SCLK / EIM_DTACK_B / GPIO4_IO13 / UART3_RX	In / Out	QSPI NOR FLASH / 磁気センサ / 拡張コネクタ	QSPI NOR FLASH 用 データ信号 / センサ割 り込み信号
C6	RAWNAND_DATA04 / USDHC2_DATA4 / QSPI_B_DATA02 / ECSPI4_SCLK / EIM_AD12 / GPIO4_IO06 / UART2_TX	In / Out	e-MMC	e-MMC 用データ信号
C7	VSS	—	GND	GND
C8	RAWNAND_WE_B / USDHC2_CMD / QSPI_B_SS0_B / KPP_COL00 / EIM_EB_B01 / GPIO4_IO01 / ECSPI3_SS3	In / Out	e-MMC	e-MMC 用 コマンド信号
C9	LCDIF_VSYNC / LCDIF_BUSY / UART4_RTS_B / SAI3_RX_DATA / WDOG2_WDOG_B / GPIO3_IO03 / ECSPI2_SS2	Out	HDMI / LCD コネクタ	LCDIF 用垂直同期信号
C10	LCDIF_DATA04 / UART8_CTS_B / ENET2_1588_EVENT2_IN / SPDIF_SR_CLK / GPIO3_IO09 / SRC_BT_CFG04 / SAI1_TX_DATA	Out	HDMI / LCD コネクタ	LCDIF 用データ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字** 機能は使用できません)

表2.6 i.MX 6UL-3端子割付 (5)

端子	i.MX 6UL 端子機能	方向	接続先	用途
C11	VSS	—	GND	GND
C12	LCDIF_DATA12 / SAI3_TX_SYNC / CSI_DATA20 / EIM_DATA04 / GPIO3_IO17 / SRC_BT_CFG12 / ECSPI1_RDY	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
C13	LCDIF_DATA16 / UART7_TX / CSI_DATA01 / EIM_DATA08 / GPIO3_IO21 / SRC_BT_CFG24 / USDHC2_DATA6	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
C14	LCDIF_DATA20 / UART8_TX / ECSPI1_SCLK / CSI_DATA12 / EIM_DATA12 / GPIO3_IO25 / SRC_BT_CFG28 / USDHC2_DATA0	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
C15	VSS	—	GND	GND
C16	ENET2_RDATA01 / UART6_RX / SIM1_PORT0_CLK / I2C3_SDA / ENET1_MDC / GPIO2_IO09 / KPP_COL04 / USB_OTG1_OC	In / Out	Ethernet PHY2 / LCD コネクタ / 拡張コネクタ	Ethernet 用データ信号 / 拡張 I2C 制御信号
C17	ENET2_RDATA00 / UART6_TX / SIM1_PORT0_TRXD / I2C3_SCL / ENET1_MDIO / GPIO2_IO08 / KPP_ROW04 / USB_OTG1_PWR	In / Out	Ethernet PHY2 / LCD コネクタ / 拡張コネクタ	Ethernet 用データ信号 / 拡張 I2C 制御信号
D1	CSI_DATA09 / USDHC2_DATA7 / SIM2_PORT1_TRXD / ECSPI1_MISO / EIM_AD07 / GPIO4_IO28 / SAI1_TX_DATA / USDHC1_VSELECT	Out	オーディオコーデック	オーディオ用 データ信号
D2	CSI_DATA08 / USDHC2_DATA6 / SIM2_PORT1_SVEN / ECSPI1_MOSI / EIM_AD06 / GPIO4_IO27 / SAI1_RX_DATA / USDHC1_RESET_B	In	オーディオコーデック	オーディオ用 データ信号
D3	CSI_DATA07 / USDHC2_DATA5 / SIM2_PORT1_RST_B / ECSPI1_SS0 / EIM_AD05 / GPIO4_IO26 / SAI1_TX_BCLK / USDHC1_CD_B	Out	オーディオコーデック	オーディオ用 クロック信号
D4	CSI_DATA06 / USDHC2_DATA4 / SIM2_PORT1_CLK / ECSPI1_SCLK / EIM_AD04 / GPIO4_IO25 / SAI1_TX_SYNC / USDHC1_WP	Out	オーディオコーデック	オーディオ用 シンク信号
D5	RAWNAND_WP_B / USDHC1_RESET_B / QSPI_A_SCLK / PWM4_OUT / EIM_BCLK / GPIO4_IO11 / ECSPI3_RDY	Out	QSPI NOR FLASH / 拡張コネクタ	QSPI NOR FLASH 用 クロック信号 / PWM 制御信号
D6	RAWNAND_DATA03 / USDHC2_DATA3 / QSPI_B_DATA01 / KPP_COL02 / EIM_AD11 / GPIO4_IO05 / ECSPI4_SS3	In / Out	e-MMC	e-MMC 用データ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (青字機能は使用できません)

表2.7 i.MX 6UL-3端子割付 (6)

端子	i.MX 6UL 端子機能	方向	接続先	用途
D7	RAWNAND_DATA00 / USDHC2_DATA0 / QSPI_B_SS1_B/KPP_ROW01/EIM_AD08 / GPIO4_IO02/ECSPI4_RDY	In / Out	e-MMC	e-MMC 用データ信号
D8	RAWNAND_RE_B / USDHC2_CLK / QSPI_B_SCLK/KPP_ROW00/EIM_EB_B00 / GPIO4_IO00/ECSPI3_SS2	Out	e-MMC	e-MMC 用 クロック信号
D9	LCDIF_HSYNC /LCDIF_RS/UART4_CTS_B / SAI3_TX_BCLK/WDOG3_WDOG_RST_B_DEB / GPIO3_IO02/ECSPI2_SS1	Out	HDMI/LCD コネクタ	LCDIF 用 水平同期信号
D10	LCDIF_DATA03 /PWM4_OUT / ENET1_1588_EVENT3_OUT/I2C4_SCL / GPIO3_IO08/SRC_BT_CFG03/SAI1_RX_DATA	Out	HDMI/LCD コネクタ	LCDIF 用データ信号
D11	LCDIF_DATA07 /UART7_RTS_B / ENET2_1588_EVENT3_OUT/SPDIF_EXT_CLK / GPIO3_IO12/SRC_BT_CFG07/ECSPI1_SS3	Out	HDMI/LCD コネクタ	LCDIF 用データ信号
D12	LCDIF_DATA11 /SAI3_RX_BCLK/CSI_DATA19 / EIM_DATA03/GPIO3_IO16/SRC_BT_CFG11 / FLEXCAN2_RX	Out	HDMI/LCD コネクタ	LCDIF 用データ信号
D13	LCDIF_DATA15 /SAI3_TX_DATA/CSI_DATA23 / EIM_DATA07/GPIO3_IO20/SRC_BT_CFG15 / USDHC2_DATA5	Out	HDMI/LCD コネクタ	LCDIF 用データ信号
D14	LCDIF_DATA19 /PWM6_OUT / WDOG1_WDOG_ANY/CSI_DATA11 / EIM_DATA11/GPIO3_IO24/SRC_BT_CFG27 / USDHC2_CLK	Out	HDMI/LCD コネクタ	LCDIF 用データ信号
D15	ENET1_RX_ER /UART7_RTS_B/PWM8_OUT / CSI_DATA23 /EIM_CRE/GPIO2_IO07 / KPP_COL03/GPT1_CAPTURE2	In	Ethernet PHY1 / 拡張コネクタ	Ethernet 用受信データ エラー信号 / CSI 用データ信号
D16	ENET2_RX_ER /UART8_RTS_B / SIM2_PORT0_SVEN/ ECSPI4_SS0 / EIM_ADDR25/GPIO2_IO15/KPP_COL07 / WDOG1_WDOG_ANY	In / Out	Ethernet PHY2 / LCD コネクタ / 拡張コネクタ	Ethernet 用受信データ エラー信号 / 拡張シリアルポート

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字** 機能は使用できません)

表2.8 i.MX 6UL-3端子割付 (7)

端子	i.MX 6UL 端子機能	方向	接続先	用途
D17	ENET2_TX_CLK / UART8_CTS_B / SIM2_PORT0_RST_B / ECSPi4_MISO / ENET2_REF_CLK2 / GPIO2_IO14 / KPP_ROW07 / ANATOP_OTG2_ID	In / Out	Ethernet PHY2 / LCD コネクタ / 拡張コネクタ	Ethernet 用クロック信号 / 拡張シリアルポート
E1	CSI_DATA05 / USDHC2_DATA3 / SIM2_PORT1_PD / ECSPi2_MISO / EIM_AD03 / GPIO4_IO24 / SAI1_RX_BCLK / UART5_CTS_B	In / Out	拡張コネクタ	汎用ポート / CSI 用データ信号
E2	CSI_DATA04 / USDHC2_DATA2 / SIM1_PORT1_TRXD / ECSPi2_MOSI / EIM_AD02 / GPIO4_IO23 / SAI1_RX_SYNC / UART5_RTS_B	In / Out	拡張コネクタ	汎用ポート / CSI 用データ信号
E3	CSI_DATA03 / USDHC2_DATA1 / SIM1_PORT1_SVEN / ECSPi2_SS0 / EIM_AD01 / GPIO4_IO22 / SAI1_MCLK / UART5_RX	Out	オーディオコーデック	オーディオ用 クロック信号
E4	CSI_DATA02 / USDHC2_DATA0 / SIM1_PORT1_RST_B / ECSPi2_SCLK / EIM_AD00 / GPIO4_IO21 / SRC_INT_BOOT / UART5_TX	In / Out	拡張コネクタ	汎用ポート / CSI 用データ信号
E5	CSI_PIXCLK / USDHC2_WP / RAWNAND_CE3_B / I2C1_SCL / EIM_OE / GPIO4_IO18 / SNVS_HP_VIO_5 / UART6_RX	In / Out	拡張コネクタ	汎用ポート / CSI 用データ信号
E6	RAWNAND_DQS / CSI_FIELD / QSPI_A_SS0_B / PWM5_OUT / EIM_WAIT / GPIO4_IO16 / SDMA_EXT_EVENT01 / SPDIF_EXT_CLK	Out	QSPI NOR FLASH / 拡張コネクタ	QSPI NOR FLASH 用 制御信号 / CSI 用制御信号
E7	NVCC_NAND	—	3.3V	電源
E8	VSS	—	GND	GND
E9	LCDIF_RESET / LCDIF_CS / CA7_MX6UL_EVENT1 / SAI3_TX_DATA / WDOG1_WDOG_ANY / GPIO3_IO04 / ECSPi2_SS3	Out	HDMI / LCD コネクタ	LCDIF 用リセット信号
E10	LCDIF_DATA02 / PWM3_OUT / ENET1_1588_EVENT3_IN / I2C4_SDA / GPIO3_IO07 / SRC_BT_CFG02 / SAI1_TX_BCLK	Out	HDMI / LCD コネクタ	LCDIF 用データ信号
E11	VSS	—	GND	GND
E12	LCDIF_DATA10 / SAI3_RX_SYNC / CSI_DATA18 / EIM_DATA02 / GPIO3_IO15 / SRC_BT_CFG10 / FLEXCAN2_TX	Out	HDMI / LCD コネクタ	LCDIF 用データ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (青字機能は使用できません)

表2.9 i.MX 6UL-3端子割付 (8)

端子	i.MX 6UL 端子機能	方向	接続先	用途
E13	NVCC_LCD	—	3.3V	電源
E14	ENET1_TDATA01 / UART6_CTS_B / PWM5_OUT / CSI_DATA20 / ENET2_MDIO / GPIO2_IO04 / KPP_ROW02 / WDOG1_WDOG_RST_B_DEB	In / Out	Ethernet PHY1 / 拡張コネクタ	Ethernet 用データ信号 / CSI 用データ信号
E15	ENET1_TDATA00 / UART5_CTS_B / CSI_DATA19 / FLEXCAN2_RX / GPIO2_IO03 / KPP_COL01 / USDHC2_VSELECT	In / Out	Ethernet PHY1 / 拡張コネクタ	Ethernet 用データ信号 / CSI 用データ信号
E16	ENET1_RX_EN / UART5_RTS_B / CSI_DATA18 / FLEXCAN2_TX / GPIO2_IO02 / KPP_ROW01 / USDHC1_VSELECT	In	Ethernet PHY1 / 拡張コネクタ	Ethernet 用受信イネー ブル信号 / CSI 用データ信号
E17	ENET1_RDATA01 / UART4_CTS_B / PWM2_OUT / CSI_DATA17 / FLEXCAN1_RX / GPIO2_IO01 / KPP_COL00 / USDHC2_LCTL	In	Ethernet PHY1 / 拡張コネクタ	Ethernet 用データ信号 / CSI 用データ信号
F1	DRAM_ODT1	Out	DDR3L	DDR3L 用オンダイタ ーミネーション信号
F2	CSI_VSYNC / USDHC2_CLK / SIM1_PORT1_CLK / I2C2_SDA / EIM_RW / GPIO4_IO19 / PWM7_OUT / UART6_RTS_B	In / Out	EEPROM / オーディ オコーデック / HDMI / PMIC	I2C 信号
F3	CSI_HSYNC / USDHC2_CMD / SIM1_PORT1_PD / I2C2_SCL / EIM_LBA_B / GPIO4_IO20 / PWM8_OUT / UART6_CTS_B	In / Out	EEPROM / オーディ オコーデック / HDMI / PMIC	I2C 信号
F4	NVCC_CSI	—	3.3V	電源
F5	CSI_MCLK / USDHC2_CD_B / RAWNAND_CE2_B / I2C1_SDA / EIM_CS0_B / GPIO4_IO17 / SNVS_HP_VIO_5_CTL / UART6_TX	In / Out	拡張コネクタ	汎用ポート / CSI 用制御信号
F6	VSS	—	GND	GND
F7	VSS	—	GND	GND
F8	VSS	—	GND	GND
F9	VSS	—	GND	GND
F10	VSS	—	GND	GND
F11	VSS	—	GND	GND
F12	VSS	—	GND	GND
F13	NVCC_ENET	—	3.3V	電源

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字** 機能は使用できません)

表2.10 i.MX 6UL-3端子割付 (9)

端子	i.MX 6UL 端子機能	方向	接続先	用途
F14	ENET1_TX_CLK / UART7_CTS_B / PWM7_OUT / CSI_DATA22 / ENET1_REF_CLK1 / GPIO2_IO06 / KPP_ROW03 / GPT1_CLK	In / Out	Ethernet PHY1 / 拡張コネクタ	Ethernet 用クロック信号 / CSI 用データ信号
F15	ENET1_TX_EN / UART6_RTS_B / PWM6_OUT / CSI_DATA21 / ENET2_MDC / GPIO2_IO05 / KPP_COL02 / WDOG2_WDOG_RST_B_DEB	In / Out	Ethernet PHY1 / 拡張コネクタ	Ethernet 用送信イーネ ブル信号 / CSI 用データ信号
F16	ENET1_RDATA00 / UART4_RTS_B / PWM1_OUT / CSI_DATA16 / FLEXCAN1_TX / GPIO2_IO00 / KPP_ROW00 / USDHC1_LCTL	In	Ethernet PHY1 / 拡張コネクタ	Ethernet 用データ信号 / CSI 用データ信号
F17	UART5_TX / ENET2_CRS / I2C2_SCL / CSI_DATA14 / CSU_CSU_ALARM_AUT00 / GPIO1_IO30 / ECSP12_MOSI	In / Out	LED / 拡張コネクタ	LED 制御 / CSI 用データ信号
G1	DRAM_ADDR14	Out	DDR3L	DDR3L 用 アドレス信号
G2	DRAM_ADDR06	Out	DDR3L	DDR3L 用 アドレス信号
G3	VSS	—	GND	GND
G4	DRAM_RESET	Out	DDR3L	DDR3L 用 リセット信号
G5	VSS	—	GND	GND
G6	NVCC_DRAM	—	DRAM_1V35	電源
G7	VSS	—	GND	GND
G8	VDD_SOC_CAP	—	コンデンサ	—
G9	VDD_ARM_CAP	—	コンデンサ	—
G10	VDD_ARM_CAP	—	コンデンサ	—
G11	VDD_ARM_CAP	—	コンデンサ	—
G12	VSS	—	GND	GND
G13	UART5_RX / ENET2_COL / I2C2_SDA / CSI_DATA15 / CSU_CSU_INT_DEB / GPIO1_IO31 / ECSP12_MISO	In / Out	LED / 拡張コネクタ	LED 制御 / CSI 用データ信号
G14	UART3_RTS_B / ENET2_TX_ER / FLEXCAN1_RX / CSI_DATA11 / ENET1_1588_EVENT1_OUT / GPIO1_IO27 / WDOG1_WDOG_B	In	CAN トランシーバ / 拡張コネクタ	CAN 用データ信号 / CSI 用データ信号
G15	VSS	—	GND	GND

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.11 i.MX 6UL-3端子割付 (10)

端子	i.MX 6UL 端子機能	方向	接続先	用途
G16	UART4_RX/ENET2_TDATA03/I2C1_SDA/ CSI_DATA13 /CSU_CSU_ALARM_AUT01/ GPIO1_IO29 /ECSPI2_SS0	In/ Out	拡張コネクタ	汎用ポート/ CSI用データ信号
G17	UART4_TX/ENET2_TDATA02/I2C1_SCL/ CSI_DATA12 /CSU_CSU_ALARM_AUT02/ GPIO1_IO28 /ECSPI2_SCLK	In/ Out	LED/拡張コネクタ	LED制御/ CSI用データ信号
H1	DRAM_SDBA1	Out	DDR3L	DDR3L用 バンクアドレス信号
H2	DRAM_ADDR01	Out	DDR3L	DDR3L用 アドレス信号
H3	DRAM_ADDR13	Out	DDR3L	DDR3L用 アドレス信号
H4	DRAM_ADDR07	Out	DDR3L	DDR3L用 アドレス信号
H5	DRAM_CS1_B	Out	DDR3L	DDR3L用 チップセレクト信号
H6	NVCC_DRAM	—	DRAM_1V35	電源
H7	VSS	—	GND	GND
H8	VDD_SOC_CAP	—	コンデンサ	—
H9	VDD_SOC_IN	—	SOC_1V35	電源
H10	VDD_SOC_IN	—	SOC_1V35	電源
H11	VDD_ARM_CAP	—	コンデンサ	—
H12	VSS	—	GND	GND
H13	NVCC_UART	—	3.3V	電源
H14	UART2_RTS_B /ENET1_COL/FLEXCAN2_RX/ CSI_DATA09 /GPT1_COMPARE3/GPIO1_IO23/ SJC_FAIL/ECSPI3_MISO	In/ Out	RS-232Cドライバ/ 拡張コネクタ	UART送信リクエスト 信号/ CSI用データ信号
H15	UART3_CTS_B/ENET2_RX_CLK/ FLEXCAN1_TX / CSI_DATA10 / ENET1_1588_EVENT1_IN/GPIO1_IO26/ EPIT2_OUT	In/ Out	CANトランシーバ/ 拡張コネクタ	CAN用データ信号/ CSI用データ信号
H16	UART3_RX/ENET2_RDATA03/SIM2_PORT0_PD/ CSI_DATA00 /UART2_RTS_B/ GPIO1_IO25 / EPIT1_OUT	In/ Out	拡張コネクタ	汎用ポート/ CSI用データ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (青字機能は使用できません)

表2.12 i.MX 6UL-3端子割付 (11)

端子	i.MX 6UL 端子機能	方向	接続先	用途
H17	UART3_TX/ENET2_RDATA02/SIM1_PORT0_PD/ CSI_DATA01 /UART2_CTS_B/ GPIO1_IO24 / SJC_JTAG_ACT/ANATOP_OTG1_ID	In/ Out	拡張コネクタ	汎用ポート/ CSI 用データ信号
J1	DRAM_SDWE_B	Out	DDR3L	DDR3L 用ライトイネーブル信号
J2	DRAM_CAS_B	Out	DDR3L	DDR3L 用カラムアドレスストロブ信号
J3	DRAM_SDCKE1	Out	DDR3L	DDR3L 用クロックイネーブル信号
J4	DRAM_ADDR08	Out	DDR3L	DDR3L 用アドレス信号
J5	VSS	—	GND	GND
J6	NVCC_DRAM	—	DRAM_1V35	電源
J7	VSS	—	GND	GND
J8	VDD_SOC_CAP	—	コンデンサ	—
J9	VDD_SOC_IN	—	SOC_1V35	電源
J10	VDD_SOC_IN	—	SOC_1V35	電源
J11	VDD_SOC_CAP	—	コンデンサ	—
J12	VSS	—	GND	GND
J13	NVCC_GPIO	—	3.3V	電源
J14	UART1_RTS_B/ENET1_TX_ER/ USDHC1_CD_B / CSI_DATA05/ENET2_1588_EVENT1_OUT/ GPIO1_IO19/USDHC2_CD_B	In	SD スロット	SD スロット用 カード検出信号
J15	UART2_CTS_B /ENET1_CRS/FLEXCAN2_TX/ CSI_DATA08 /GPT1_COMPARE2/GPIO1_IO22/ SJC_DE_B/ECSPI3_MOSI	In	RS-232C ドライバ/ 拡張コネクタ	UART 用送信イネーブル信号/ CSI 用データ信号
J16	UART2_RX /ENET1_TDATA03/I2C4_SDA/ CSI_DATA07 /GPT1_CAPTURE2/GPIO1_IO21/ SJC_DONE/ECSPI3_SCLK	In	RS-232C ドライバ/ 拡張コネクタ	UART 用データ信号/ CSI 用データ信号
J17	UART2_TX /ENET1_TDATA02/I2C4_SCL/ CSI_DATA06 /GPT1_CAPTURE1/GPIO1_IO20/ ECSPI3_SS0	In/ Out	RS-232C ドライバ/ 拡張コネクタ	UART 用データ信号/ CSI 用データ信号
K1	DRAM_ADDR02	Out	DDR3L	DDR3L 用 アドレス信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.13 i.MX 6UL-3端子割付 (12)

端子	i.MX 6UL 端子機能	方向	接続先	用途
K2	DRAM_SDBA2	Out	DDR3L	DDR3L 用バンクアドレス信号
K3	DRAM_ADDR11	Out	DDR3L	DDR3L 用アドレス信号
K4	DRAM_ADDR04	Out	DDR3L	DDR3L 用アドレス信号
K5	DRAM_ADDR15	Out	DDR3L	DDR3L 用アドレス信号
K6	NVCC_DRAM	—	DRAM_1V35	電源
K7	VSS	—	GND	GND
K8	VDD_SOC_CAP	—	コンデンサ	—
K9	VDD_SOC_IN	—	SOC_1V35	電源
K10	VDD_SOC_IN	—	SOC_1V35	電源
K11	VDD_SOC_CAP	—	コンデンサ	—
K12	VSS	—	GND	GND
K13	I2C2_SCL/GPT1_CAPTURE1/ ANATOP_OTG1_ID /ENET1_REF_CLK1/ MQS_RIGHT/GPIO1_IO00/ ENET1_1588_EVENT0_IN/SRC_SYSTEM_RESET/ WDOG3_WDOG_B	In	USB	USB 用
K14	UART1_TX/ENET1_RDATA02/I2C3_SCL/ CSI_DATA02/GPT1_COMPARE1/GPIO1_IO16/ SPDIF_OUT	Out	HDMI/LCD コネクタ	HDMI 出力制御信号
K15	UART1_CTS_B/ENET1_RX_CLK/ USDHC1_WP / CSI_DATA04/ENET2_1588_EVENT1_IN/ GPIO1_IO18/USDHC2_WP	In	SD スロット	SD スロット用 WP 信号
K16	UART1_RX/ENET1_RDATA03/I2C3_SDA/ CSI_DATA03 /GPT1_CLK/ GPIO1_IO17 /SPDIF_IN	In/ Out	拡張コネクタ	汎用ポート/ CSI 用データ信号
K17	ENET1_MDIO /ENET2_MDIO/ USB_OTG_PWR_WAKE/ CSI_MCLK/USDHC2_WP/GPIO1_IO06/ CCM_WAIT/CCM_REF_EN_B/UART1_CTS_B	In/ Out	Ethernet PHY1/ Ethernet PHY2	Ethernet PHY 用 データ信号
L1	DRAM_ADDR05	Out	DDR3L	DDR3L 用 アドレス信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.14 i.MX 6UL-3端子割付 (13)

端子	i.MX 6UL 端子機能	方向	接続先	用途
L2	DRAM_ADDR09	Out	DDR3L	DDR3L 用 アドレス信号
L3	VSS	—	GND	GND
L4	DRAM_ADDR12	Out	DDR3L	DDR3L 用 アドレス信号
L5	DRAM_ADDR00	Out	DDR3L	DDR3L 用 アドレス信号
L6	NVCC_DRAM	—	DRAM_1V35	電源
L7	VSS	—	GND	GND
L8	VDD_SOC_CAP	—	コンデンサ	—
L9	VDD_SOC_CAP	—	コンデンサ	—
L10	VDD_SOC_CAP	—	コンデンサ	—
L11	VDD_SOC_CAP	—	コンデンサ	—
L12	VSS	—	GND	GND
L13	VDDA_ADC_3P3	—	3.3V	電源
L14	I2C1_SCL/GPT1_COMPARE2/ USB_OTG2_PWR / ENET1_REF_CLK_25M/USDHC1_WP/ GPIO1_IO02/SDMA_EXT_EVENT00/ SRC_ANY_PU_RESET/UART1_TX	Out	USB 用電源 IC	USB 用電源制御信号
L15	I2C2_SDA/GPT1_COMPARE1/ USB_OTG1_OC / ENET2_REF_CLK2/MQS_LEFT/GPIO1_IO01/ ENET1_1588_EVENT0_OUT/ SRC_EARLY_RESET/WDOG1_WDOG_B	In	USB 用電源 IC	USB 用電源制御信号
L16	ENET1_MDC /ENET2_MDC/ USB_OTG_HOST_MODE/CSI_PIXCLK/ USDHC2_CD_B/GPIO1_IO07/CCM_STOP/ UART1_RTS_B	Out	Ethernet PHY1/ Ethernet PHY2	Ethernet PHY 用 クロック信号
L17	I2C1_SDA/GPT1_COMPARE3/ USB_OTG2_OC / USDHC1_CD_B/GPIO1_IO03/ CCM_DI0_EXT_CLK/SRC_TESTER_ACK/ UART1_RX	In	USB 用電源 IC	USB 用電源制御信号
M1	DRAM_SDBA0	Out	DDR3L	DDR3L 用バンクアド レス信号
M2	DRAM_ADDR03	Out	DDR3L	DDR3L 用 アドレス信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.15 i.MX 6UL-3端子割付 (14)

端子	i.MX 6UL 端子機能	方向	接続先	用途
M3	DRAM_SDCKE0	Out	DDR3L	DDR3L 用クロックイネーブル信号
M4	DRAM_ADDR10	Out	DDR3L	DDR3L 用アドレス信号
M5	DRAM_RAS_B	Out	DDR3L	DDR3L 用ロウアドレスストロブ信号
M6	NVCC_DRAM	—	DRAM_1V35	電源
M7	VSS	—	GND	GND
M8	VSS	—	GND	GND
M9	VSS	—	GND	GND
M10	VSS	—	GND	GND
M11	VSS	—	GND	GND
M12	NGND_KEL0	—	GND	GND
M13	ADC_VREFH	—	3.3V	電源
M14	SJC_TCK / GPT2_COMPARE2 / SAI2_RX_DATA / PWM7_OUT / GPIO1_IO14 / SIM2_POWER_FAIL	In	ARM JTAG 20 コネクタ	JTAG 用
M15	PWM2_OUT / WDOG1_WDOG_ANY / SPDIF_IN / CSI_HSYNC / USDHC2_RESET_B / GPIO1_IO09 / USDHC1_RESET_B / UART5_CTS_B	In / Out	SD スロット用電源制御回路 / 拡張コネクタ	SD スロット用リセット信号
M16	ENET1_REF_CLK1 / PWM3_OUT / USB_OTG1_PWR / USDHC1_RESET_B / GPIO1_IO04 / ENET2_1588_EVENT0_IN / UART5_TX	Out	USB 用電源 IC	USB 用電源制御信号
M17	ENET2_REF_CLK2 / PWM4_OUT / ANATOP_OTG2_ID / CSI_FIELD / USDHC1_VSELECT / GPIO1_IO05 / ENET2_1588_EVENT0_OUT / UART5_RX / ADC1_IN5	In / Out	PMIC / 拡張コネクタ	汎用ポート / AD 入力
N1	DRAM_ODT0	Out	DDR3L	DDR3L 用オンダイターミネーション信号
N2	DRAM_CS0_B	Out	DDR3L	DDR3L 用チップセレクト信号
N3	VSS	—	GND	GND
N4	DRAM_ZQPAD	—	DDR3L	DDR3L 用 ZQ キャリブレーション用参照電圧
N5	VSS	—	GND	GND

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.16 i.MX 6UL-3端子割付 (15)

端子	i.MX 6UL 端子機能	方向	接続先	用途
N6	NVCC_DRAM_2P5	—	コンデンサ	—
N7	TEST_MODE	In	GND	TEST_MODE
N8	SNVS_TAMPER5	In	拡張コネクタ	タンパ用
N9	SNVS_TAMPER8	In	拡張コネクタ	タンパ用
N10	SNVS_TAMPER7	In	拡張コネクタ	タンパ用
N11	SNVS_TAMPER6	In	拡張コネクタ	タンパ用
N12	VDD_SNVS_CAP	—	コンデンサ	—
N13	VDD_HIGH_IN	—	3.3V	電源
N14	SJC_TRSTB /GPT2_COMPARE3/ SAI2_TX_DATA/PWM8_OUT/GPIO1_IO15/ CAAM_RNG_OSC_OBS	In	ARM JTAG 20 コネクタ	JTAG 用
N15	SJC_TDO /GPT2_CAPTURE2/SAI2_TX_SYNC/ CCM_CLKO2/CCM_STOP/GPIO1_IO12/ MQS_RIGHT/EPIT2_OUT	Out	ARM JTAG 20 コネクタ	JTAG 用
N16	SJC_TDI /GPT2_COMPARE1/SAI2_TX_BCLK/ PWM6_OUT/GPIO1_IO13/MQS_LEFT/ SIM1_POWER_FAIL	In	ARM JTAG 20 コネクタ	JTAG 用
N17	PWM1_OUT/WDOG1_WDOG_B/SPDIF_OUT/ CSI_VSYNC /USDHC2_VSELECT/ GPIO1_IO08 / CCM_PMIC_RDY/UART5_RTS_B	In/ Out	拡張コネクタ	汎用ポート/ CSI 用制御信号
P1	DRAM_SDCLK0_P	Out	DDR3L	DDR3L 用 クロック信号
P2	DRAM_SDCLK0_N	Out	DDR3L	DDR3L 用 クロック信号
P3	DRAM_DATA13	In/ Out	DDR3L	DDR3L 用データ信号
P4	DRAM_VREF	—	DDR3L	DDR3L 用 参照電圧
P5	DRAM_DATA12	In/ Out	DDR3L	DDR3L 用データ信号
P6	DRAM_SDQS0_P	Out	DDR3L	DDR3L 用データスト ロープ信号
P7	DRAM_SDQS0_N	Out	DDR3L	DDR3L 用データスト ロープ信号

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.17 i.MX 6UL-3端子割付 (16)

端子	i.MX 6UL 端子機能	方向	接続先	用途
P8	POR_B	In	リセット IC	SYSTEM リセット信号
P9	SNVS_TAMPER4	In	拡張コネクタ	タンパ用
P10	SNVS_TAMPER3	In	GND	GND
P11	SNVS_TAMPER2	In	拡張コネクタ	タンパ用
P12	VDD_SNVS_IN	—	SNVS_3V	電源
P13	NVCC_PLL	—	コンデンサ	—
P14	SJC_TMS / GPT2_CAPTURE1 / SAI2_MCLK / CCM_CLKO1 / CCM_WAIT / GPIO1_IO11 / SDMA_EXT_EVENT01 / EPIT1_OUT	In	ARM JTAG 20 コネクタ	JTAG 用
P15	SJC_MOD / GPT2_CLK / SPDIF_OUT / ENET1_REF_CLK_25M / CCM_PMIC_RDY / GPIO1_IO10 / SDMA_EXT_EVENT00	In	ARM JTAG 20 コネクタ	JTAG 用
P16	CCM_CLK1_N	Out	TP3	モニタ用
P17	CCM_CLK1_P	Out	TP4	モニタ用
R1	DRAM_DATA15	In / Out	DDR3L	DDR3L 用データ信号
R2	DRAM_DATA14	In / Out	DDR3L	DDR3L 用データ信号
R3	VSS	—	GND	GND
R4	DRAM_DATA11	In / Out	DDR3L	DDR3L 用データ信号
R5	VSS	—	GND	GND
R6	SNVS_TAMPER9	In	LCD コネクタ	タンパ用
R7	VSS	—	GND	GND
R8	ONOFF	In	電源 ON/OFF スイッチ	メイン電源 ON/OFF 信号
R9	SNVS_TAMPER1	In	NC	—
R10	SNVS_TAMPER0	In	拡張コネクタ	タンパ用
R11	VSS	—	GND	GND
R12	VDD_USB_CAP	—	コンデンサ	—
R13	GPANAIO	—	NC	—
R14	VDD_HIGH_CAP	—	コンデンサ	—
R15	VDD_HIGH_CAP2	—	コンデンサ	—
R16	VSS	—	GND	GND

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (**青字**機能は使用できません)

表2.18 i.MX 6UL-3端子割付 (17)

端子	i.MX 6UL 端子機能	方向	接続先	用途
R17	VSS	—	GND	GND
T1	DRAM_SDQS1_P	Out	DDR3L	DDR3L 用データスト ロープ信号
T2	DRAM_SDQS1_N	Out	DDR3L	DDR3L 用データスト ロープ信号
T3	DRAM_DQM1	Out	DDR3L	DDR3L 用データ信号
T4	DRAM_DATA00	In/ Out	DDR3L	DDR3L 用データ信号
T5	DRAM_DATA06	In/ Out	DDR3L	DDR3L 用データ信号
T6	DRAM_DATA02	In/ Out	DDR3L	DDR3L 用データ信号
T7	DRAM_DQM0	Out	DDR3L	DDR3L 用データ信号
T8	DRAM_DATA05	In/ Out	DDR3L	DDR3L 用データ信号
T9	SNVS_PMIC_ON_REQ	Out	PMIC	PMIC 電源 ON リクエスト信号
T10	BOOT_MODE0	In	2 ピンディップ スイッチ	BOOTMODE 選択信号
T11	RTC_XTALI	In	RTC 用発振子	RTC クロック入力
T12	USB_OTG1_VBUS	In/ Out	USB	USB 用
T13	USB_OTG2_DN	In/ Out	USB	USB 用
T14	VSS	—	GND	GND
T15	USB_OTG1_DN	In/ Out	USB	USB 用
T16	XTALI	Out	NC	—
T17	XTALO	In	SoC メインクロック用 発振器	SYSTEM 用 クロック入力
U1	VSS	—	GND	GND
U2	DRAM_DATA08	In/ Out	DDR3L	DDR3L 用データ信号
U3	DRAM_DATA09	In/ Out	DDR3L	DDR3L 用データ信号

赤字：GOHSSC-6UL デフォルト使用機能

青字：GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字：GOHSSC-6UL-B 使用機能（青字機能は使用できません）

表2.19 i.MX 6UL-3端子割付 (18)

端子	i.MX 6UL 端子機能	方向	接続先	用途
U4	DRAM_DATA07	In/ Out	DDR3L	DDR3L 用データ信号
U5	DRAM_DATA10	In/ Out	DDR3L	DDR3L 用データ信号
U6	DRAM_DATA01	In/ Out	DDR3L	DDR3L 用データ信号
U7	DRAM_DATA03	In/ Out	DDR3L	DDR3L 用データ信号
U8	DRAM_DATA04	In/ Out	DDR3L	DDR3L 用データ信号
U9	CCM_PMIC_STBY_REQ	Out	PMIC	PMIC スタンバイリクエスト信号
U10	BOOT_MODE1	In	2 ピンディップ スイッチ	BOOTMODE 選択信号
U11	RTC_XTALO	Out	RTC	RTC クロック出力
U12	USB_OTG2_VBUS	Out	USB	USB 用
U13	USB_OTG2_DP	In/ Out	USB	USB 用
U14	VSS	—	GND	GND
U15	USB_OTG1_DP	In/ Out	USB	USB 用
U16	USB_OTG1_CHD_B	—	NC	—
U17	VSS	—	GND	GND

赤字 : GOHSSC-6UL デフォルト使用機能

青字 : GOHSSC-6UL にて 0Ω 抵抗の付け替えで使用可能な機能

緑字 : GOHSSC-6UL-B 使用機能 (青字機能は使用できません)

2.2.3 GOHSSC-6UL 搭載デバイス使用関係一覧

表2.20に GOHSSC-6UL に搭載しているデバイスの使用可否一覧表を示します。表中の「○」印は両モジュールが併用可能なことを示し、「×」印は併用不可能なことを示します。

表2.20 GOHSSC-6UL搭載デバイス使用可否一覧表

i.MX 6UL 周辺機能	部品 No.	デバイス名	EEPROM	QSPI NOR FLASH	e-MMC	DDR3L SDRAM1	DDR3L SDRAM2	USB 電源制御 IC	CAN トランシーバ	RS-232C ドライバ	Ethernet PHY1	Ethernet PHY2	オーディオコーデック	HDMI	磁気センサ	加速度センサ	ジャイロスコープ	NAND FLASH	LCD コネクタ	拡張コネクタ	
I ² C	U3	EEPROM	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
QSPI	U4	QSPI NOR FLASH	○	○	○	○	○	○	○	○	○	○	○	○	×	×	×	×	○	×	○
uSDHC	U5	e-MMC	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×	○	○
MMDC	U6	DDR3L SDRAM1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
MMDC	U7	DDR3L SDRAM2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
USB	U8	USB 電源制御 IC	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
FLEXCAN	U9	CAN トランシーバ	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×
UART	U10	RS-232C ドライバ	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×
ENET	U11	Ethernet PHY1	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×
ENET	U12	Ethernet PHY2	○	○	○	○	○	○	○	○	○	○	○	○	×	×	×	○	×	×	×
SAI	U13	オーディオコーデック	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
LCDIF, SPDIF	U14	HDMI	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	×	○
I ² C	U15	磁気センサ	○	×	○	○	○	○	○	○	○	×	○	○	○	○	○	○	×	○	○
I ² C	U16	加速度センサ	○	×	○	○	○	○	○	○	○	×	○	○	○	○	○	○	×	○	○
I ² C	U17	ジャイロスコープ	○	×	○	○	○	○	○	○	○	×	○	○	○	○	○	○	×	○	○
GPMI	U19	NAND FLASH	○	×	×	○	○	○	○	○	○	○	○	○	×	×	×	○	○	○	×
LCDIF	CN12	LCD コネクタ	○	○	○	○	○	○	○	○	○	×	○	×	○	○	○	○	○	○	○
CSI, ADC, I ² C, ECSPI, GPIO	JP2	拡張コネクタ	○	×	○	○	○	○	×	×	×	×	○	○	○	○	○	○	×	○	○

※i.MX 6UL 周辺機能に対するデバイスの割り当てを示しており、機能が重なる場合は同時に使用できません。各機能の切り替えはチップ抵抗の実装／未実装で変更可能です。詳細は2. 機能仕様の各インタフェース仕様をご参照ください。

2.3 ソフトウェア構成

GOHSSC-6UL 用ソフトウェア（以降、本ソフトウェアと呼ぶ）には、ブートローダとして動作する u-boot と、メインとなる Linux の 2 つが存在します。図2.1に u-boot ソフトウェア構成図、図2.2に Linux ソフトウェア構成図をそれぞれ示します。

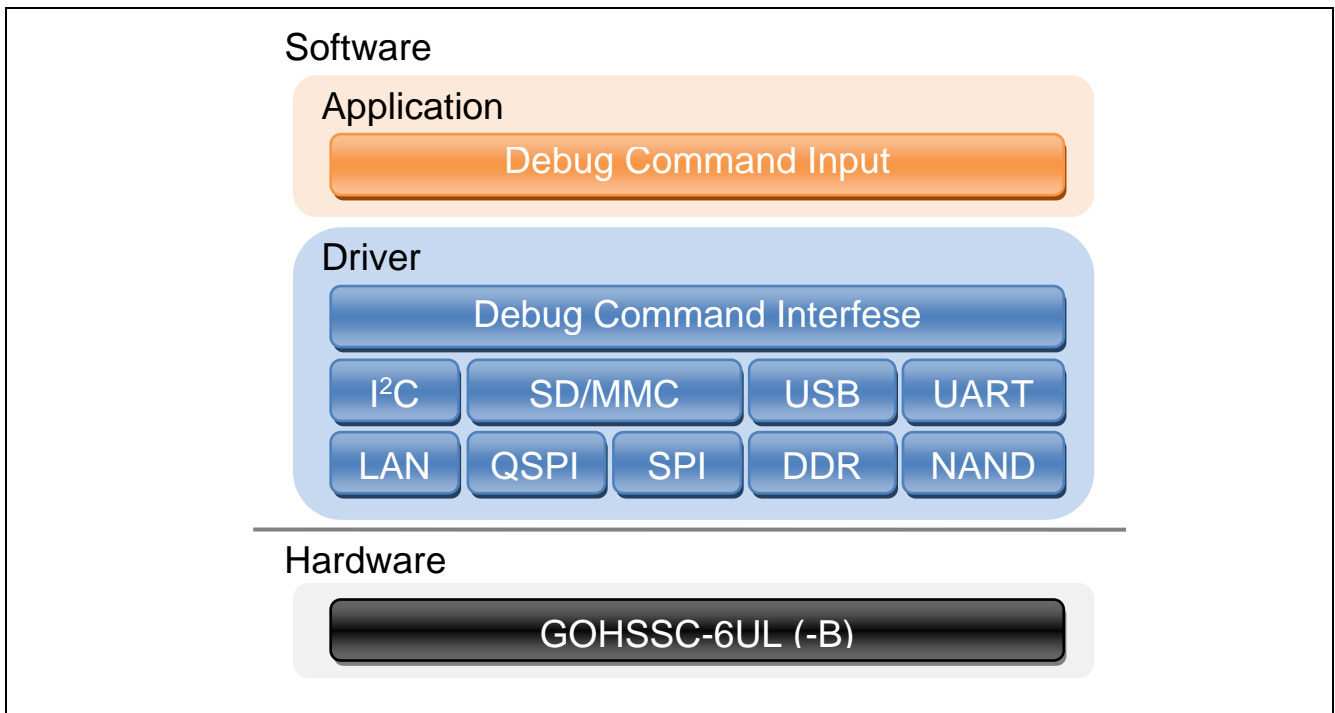


図2.1 u-boot ソフトウェア構成図

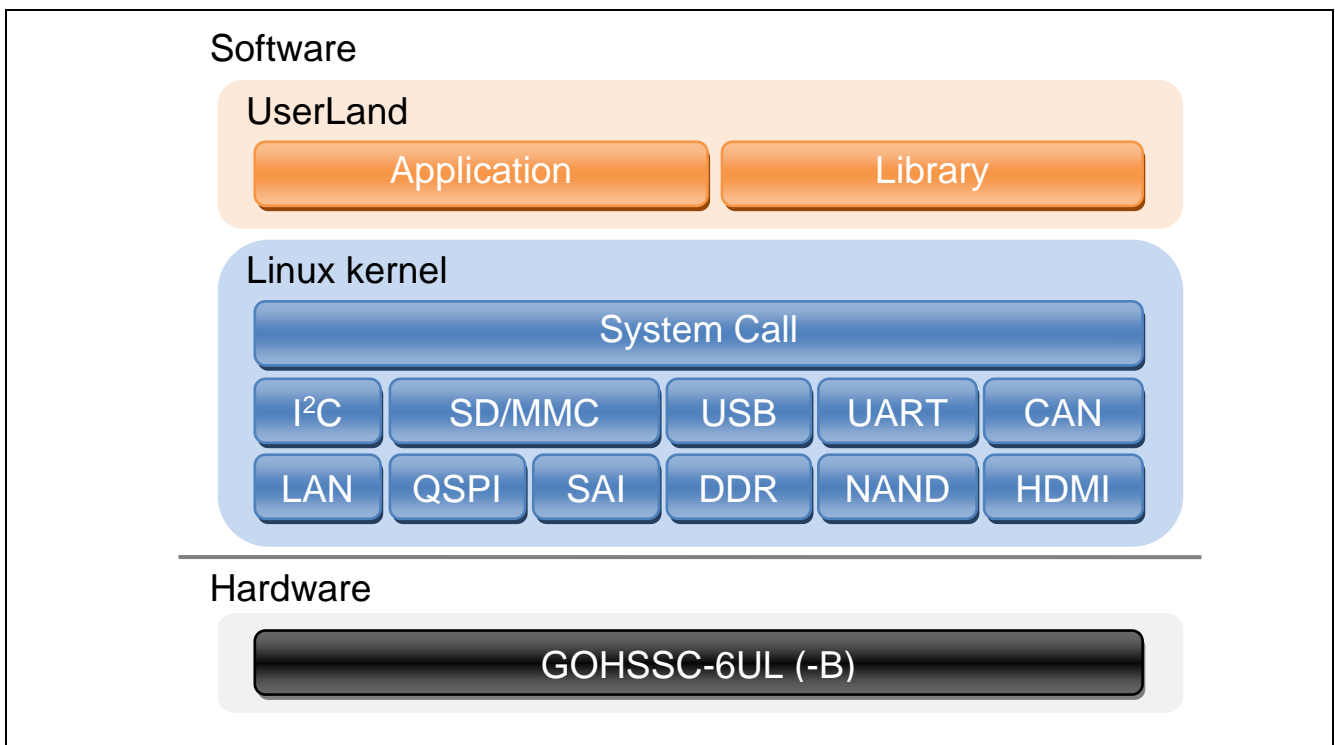


図2.2 Linux ソフトウェア構成

3. 開発環境構築

本章では、本ソフトウェアに必要なブートローダ、カーネル、ルートファイルシステムの入手手順および、ブートローダ、カーネルのビルド手順を示します。ブートローダは、電源投入時に最初に動作するソフトウェアであり、u-boot を使用します。カーネルは Linux3.14.52 を使用します。ルートファイルシステムは NXP 製ボードサポートパッケージ（以下、BSP と呼ぶ）を使用します。本資料では、グラフィックスバックエンドに x11 を選択した場合のルートファイルシステムを使用します。

3.1 準備するもの

表3.1に本ソフトウェアの開発環境構築に必要な機材を示します。

表3.1 開発環境機材一覧

項目	内容
開発 PC	<ul style="list-style-type: none"> ・ソフトウェアのビルドに使用する PC ・OS は Ubuntu14.04LTS/Linux を使用 ・ハードディスクは 120G バイト以上の空き容量が必要 ・インターネットに接続できる環境

3.2 環境構築手順

本ソフトウェアの開発環境は以下手順にて構築します。

- ① Yocto Project ビルド環境インストール（3.4章）
- ② Git管理ツールrepoインストール（3.5章）
- ③ ツールチェイン入手（3.6章）
- ④ ソフトウェアビルド（3.7章）
- ⑤ ルートファイルシステム入手（3.8章）

3.3 フォルダ構成

図3.1に本ソフトウェア開発環境のフォルダ構成を示します。

/opt/fsl-imx-x11/3.14.52-1.1.0	←ツールチェイン
~/bin	←repo 格納先
~/mountpoint	←各マウントポイント
~/fsl-release-bsp	←NXP 製 BSP
~/gohssc-6ul	←GOHSSC-6UL ソースコード

図3.1 開発環境フォルダ構成図

3.4 Yocto Project ビルド環境インストール

本ソフトウェアは Yocto Project にて構築された NXP 製開発環境を用います。図3.2に Yocto Project ビルド環境構築に使用するコマンドを示します。

```
> sudo apt-get install gawk wget git-core diffstat unzip texinfo gcc-multilib ¥
> build-essential chrpath socat libstd1.2-dev
> sudo apt-get install libstd1.2-dev xterm sed cvs subversion coreutils texi2html ¥
> docbook-utils python-pysqlite2 help2man make gcc g++ desktop-file-utils ¥
> libgl1-mesa-dev libglu1-mesa-dev mercurial autoconf automake groff curl lzop asciidoc
> sudo apt-get install u-boot-tools
```

図3.2 Yocto Project ビルド環境インストールコマンド

3.5 Git 管理ツール repo インストール

NXP 製 BSP は Git にて管理されています。開発環境入手のため、Git 管理ツールである repo をインストールします。図3.3に repo インストールに使用するコマンドを示します。

```
> mkdir ~/bin
> curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
> chmod a+x ~/bin/repo
```

図3.3 repo インストールコマンド

repo インストール後は、環境 PATH を設定するために".bashrc"ファイルを編集し、開発 PC を再起動します。図3.4に".bashrc"ファイルの編集内容を示します。

```
export PATH=~/.bin:$PATH
```

図3.4 ".bashrc"ファイル編集内容

再起動後は Git 環境の設定をします。図3.5に Git 環境の設定に使用するコマンドを示します。ここでは仮に、user.name を umikai.taro、user.email を umikai.taro@goh.co.jp に設定します。実際にコマンド入力する際は変更してください。

```
> git config --global user.name umikai.taro
> git config --global user.email umikai.taro@goh.co.jp

以下コマンドにて user.name、user.email 設定内容を確認します。
> git config --list
user.name=umikai.taro
user.email=umikai.taro@goh.co.jp
color.ui=auto
```

図3.5 Git 環境設定コマンド

3.6 ツールチェイン入手

NXP 製ツールチェインを入手します。図3.6にツールチェインの入手に使用するコマンドを示します。なお、ツールチェインの入手には時間がかかります。ファイルのダウンロードに 1 日程度の時間を見込んだ上で実行ください。

```
> mkdir fsl-release-bsp
> cd fsl-release-bsp
> repo init -u git://git.freescale.com/imx/fsl-arm-yocto-bsp.git -b imx-3.14.52-1.1.0_ga
> repo sync
> MACHINE=imx6ulevk source fsl-setup-release.sh -b build-x11 -e x11
> bitbake meta-toolchain
> cd tmp/deploy/sdk/
> sudo ./fsl-imx-x11-glibc-x86_64-meta-toolchain-cortexa7hf-vfp-neon-toolchain-3.14.52-1.1.0.sh
Enter target directory for SDK (default: /opt/fsl-imx-x11/3.14.52-1.1.0):
> <ENTER>
```

図3.6 ツールチェイン入手コマンド

ツールチェイン入手後は環境 PATH を設定します。図3.7に環境 PATH の設定に使用するコマンドを示します。環境 PATH の設定はターミナル起動毎に必要になります。なお、本コマンドを実行すると環境 PATH を書き換えます。変更した環境 PATH はコマンドを実行したターミナル内でのみ有効ですので、元に戻す場合はターミナルを閉じてください。

```
> ./opt/fsl-imx-x11/3.14.52-1.1.0/environment-setup-cortexa7hf-vfp-neon-poky-linux-gnueabi
```

図3.7 ツールチェイン環境 PATH 設定コマンド

3.7 ソフトウェアビルド

本ソフトウェアのビルド手順を示します。

3.7.1 ソフトウェア解凍

本ソフトウェアを格納した圧縮ファイルを解凍します。図3.8にソフトウェア解凍に使用するコマンドを示します。

```
> cd ~/
> tar xzf gohssc-6ul.tar.gz
> cd gohssc-6ul/source
> ls ← 各ソースコード格納フォルダの存在を確認する
    uboot-imx ← u-boot 用ソースコード格納フォルダ
    kernel-source ← Linux カーネル用ソースコード格納フォルダ
```

図3.8 ソフトウェア解凍コマンド

3.7.2 u-boot ビルド

本ソフトウェアのブートローダとなる u-boot をビルドします。u-boot 用ソースコード格納フォルダに移動し、make コマンドを実行します。図3.9に u-boot ビルドに使用するコマンドを示します。

```
> cd ~/gohssc-6ul/source/uboot-imx
> make distclean mx6ul_gohssc_defconfig ARCH=arm ¥
> CROSS_COMPILE=/opt/fsl-imx-x11/3.14.52-1.1.0/sysroots/x86_64-pokysdk-linux/usr/¥
> bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-
> make ARCH=arm CROSS_COMPILE=/opt/fsl-imx-x11/3.14.52-1.1.0/sysroots/x86_64-pokysdk-linux/usr/bin/¥
> arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-
:
MKIMAGE u-boot.imx ← ビルド成功時に出力される
```

図3.9 u-boot ビルドコマンド

ビルドにて出力されたファイル”~/gohssc-6ul/source/uboot-imx/u-boot.imx”が u-boot の実行ファイルになります。もし、端末に”MKIMAGE u-boot.imx”が出力されずビルドに失敗した場合は、エラーの内容を確認し再度各ツールのインストールおよびビルドをしてください。

3.7.3 カーネルビルド

本ソフトウェアのメインとなる Linux カーネルをビルドします。Linux カーネル用ソースコード格納フォルダに移動し、make コマンドを実行します。図3.10にカーネルビルドに使用するコマンドを示します。

```
> cd ~/gohssc-6ul/source/kernel-source
> make imx_v7_defconfig zImage imx6ul-gohssc.dtb ARCH=arm ¥
> CROSS_COMPILE=/opt/fsl-imx-x11/3.14.52-1.1.0/sysroots/x86_64-pokysdk-linux/usr/¥
> bin/arm-poky-linux-gnueabi/arm-poky-linux-gnueabi-
:
Kernel: arch/arm/boot/zImage is ready ← ビルド成功時に出力される
```

図3.10 カーネルビルドコマンド

ビルドにて出力されたファイル”~/gohssc-6ul/source/kernel-source/arch/arm/boot/zImage”が Linux カーネルの実行ファイルになります。また、ファイル”~/gohssc-6ul/source/kernel-source/arch/arm/boot/dts/imx6ul-gohssc.dtb”がデバイスツリーファイルになります。

3.8 ルートファイルシステム入手

ルートファイルシステムを入手します。図3.11にルートファイルシステム入手に使用するコマンドを示します。図3.11ではグラフィックスバックエンドに x11 を選択した場合のルートファイルシステムを入手しておりますが、DirectFB 等、他のグラフィックスバックエンドのルートファイルシステムが必要な場合は、NXP 製のドキュメント”Freescale_Yocto_Project_User's_Guide.pdf”をご確認頂き、適宜コマンドを変更してください。なお、ルートファイルシステムの入手には時間がかかります。ファイルのダウンロードに 1 日程度の時間を見込んだ上で実行ください。

```
> cd ~/fsl-release-bsp
> MACHINE=imx6ulevk source fsl-setup-release.sh -b build-x11 -e x11
> bitbake fsl-image-gui
```

図3.11 ルートファイルシステム入手コマンド

出力されたファイル”~/fsl-release-bsp/build-x11/tmp/deploy/images/imx6ulevk/fsl-image-gui-imx6ulevk.ext3”がルートファイルシステムのバイナリデータになります。シリアルコンソールからの Linux 操作を有効にするため、ルートファイルシステムのバイナリデータをマウントし、”/etc/inittab”ファイルを変更します。

図3.12にルートファイルシステムマウント時に使用するコマンドを、図3.13に”/etc/inittab”ファイルの編集内容を示します。

```
> mkdir ~/mountpoint
> mkdir ~/mountpoint/guiimage
> sudo mount -o loop -t ext3 ~/fsl-release-bsp/build-x11/tmp/deploy/images/imx6ulevk/fsl-image-gui-imx6ulevk.ext3 ¥
~/mountpoint/guiimage

<”/etc/inittab”ファイル編集>

> sudo umount ~/mountpoint/guiimage
```

図3.12 ルートファイルシステムマウントコマンド

```
変更前) mxc0:12345:respawn:/sbin/getty -L 115200 ttymx0
変更後) mxc0:12345:respawn:/sbin/getty -L 115200 ttymx1
```

図3.13 ”/etc/inittab”ファイル編集内容

4. ソフトウェア起動手順

本章では、本ソフトウェアの起動手順を示します。

4.1 準備するもの

表4.1に本ソフトウェアの動作に必要な機材を示します。

表4.1 準備機材一覧

項目	内容
開発 PC	<ul style="list-style-type: none">・ SD カードにソフトウェアを書き込む際に使用する PC・ OS は Ubuntu14.04LTS/Linux を搭載・ SD スロットが必要
SD カード	<ul style="list-style-type: none">・ GOHSSC-6UL 用ソフトウェアを格納する SD カード・ 最低 4G バイトの容量が必要
シリアルコンソール端末	<ul style="list-style-type: none">・ GOHSSC-6UL と通信するための PC・ シリアルインタフェースが必要・ D-Sub9 ピンのクロスケーブルが必要・ シリアル通信ソフトウェアが必要 Linux では「minicom」、Windows では「TeraTermPro」など

4.2 起動までの手順

本ソフトウェアの起動を以下手順にて行います。

- ① SDカード作成 (4.4章)
- ② ブートモード設定 (4.5章)
- ③ シリアル通信ソフトウェアの設定 (4.6章)
- ④ 起動 (4.7章)
- ⑤ MACアドレス設定 (4.8章)

4.3 接続方法

図4.1に GOHSSC-6UL と周辺機器の接続図を示します。

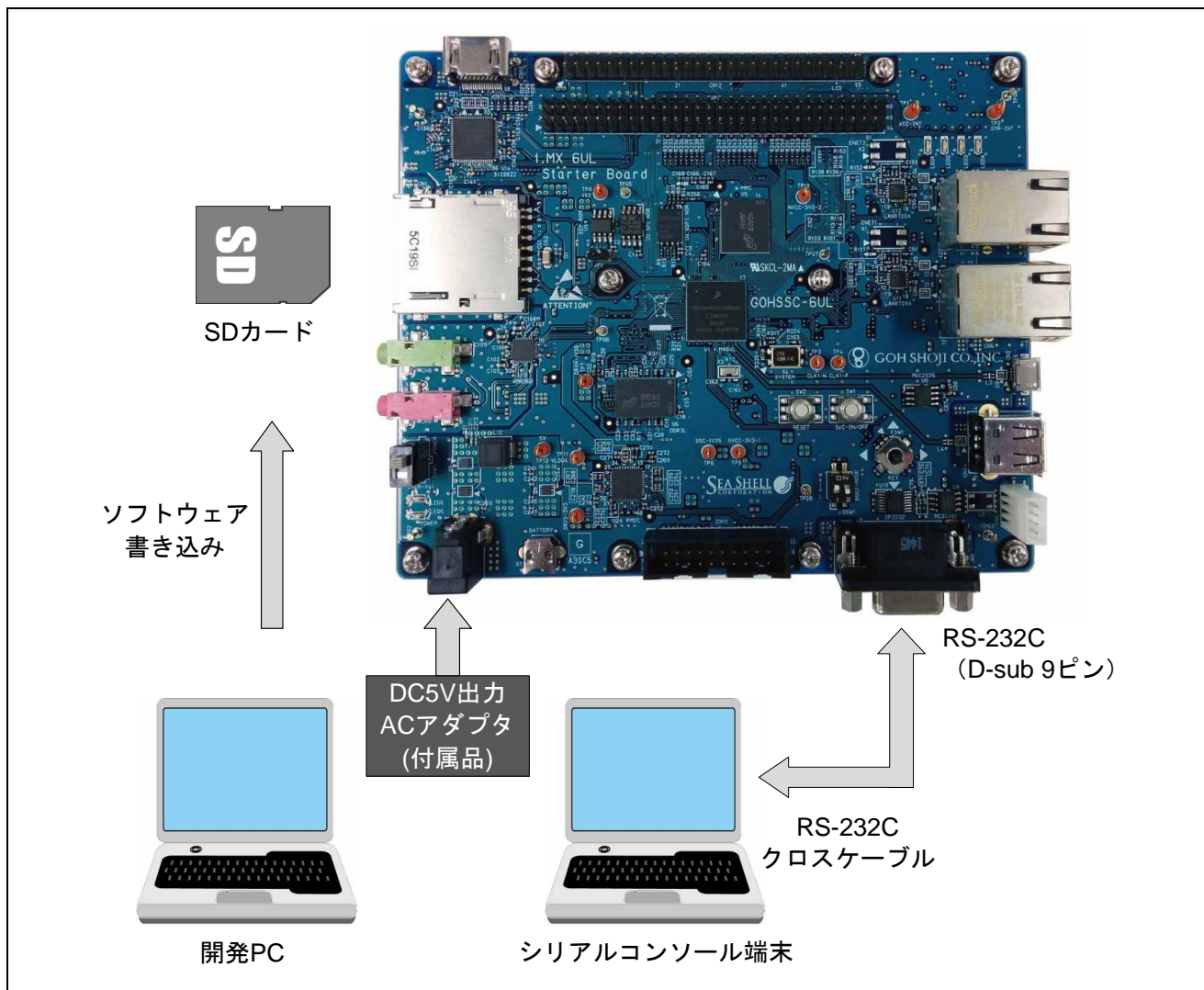


図4.1 GOHSSC-6UL接続図

4.4 SD カード作成

本項では、本ソフトウェアのブートローダ、カーネル、ルートファイルシステムをSDカードに格納する手順を示します。本作業は開発PCを使用します。開発PCにSDカードをマウントし、4.4.1章～4.4.4章に示すコマンドを入力してください。また、SDカード内のデータは全て削除されるため注意してください。なお、本コマンドはSDカードが/dev/sddにマウントされた場合の例です。実際にコマンド入力する際は、それぞれの環境に合わせたマウント先を設定してください。図4.2にSDカードのメモリマップを示します。

0x0000 0000	Reserved
0x0000 0400	ブートローダ領域(u-boot) フォーマット : RAW
0x00A0 0000	カーネル領域(Linux) フォーマット : FAT16
0x1FE0 0000	Reserved
0x2580 0000	ルートファイルシステム領域 フォーマット : EXT3

図4.2 SD カードメモリマップ

4.4.1 パーティション作成

図4.3に SD カードのパーティションを作成するコマンドを示します。

```
> sudo fdisk /dev/sdd
> p
> d
> u
> u
> n
> p
> 1
> 20480
> 1024000
> p
> n
> p
> 2
> 1228800
> <ENTER>
> p
> w
```

図4.3 パーティション作成コマンド

パーティション作成後に再度”fdisk”コマンドを使用し、パーティションが正常に作成されているか確認します。図4.4は 4G バイトの SD カードを使用した場合のパーティションです。/dev/sdd1 の始点が”20480”、終点が”1024000”、/dev/sdd2 の始点が”1228800”となっていることを確認します。

```
> sudo fdisk /dev/sdd
> p
```

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sdd1		20480	1024000	501760+	83	Linux
/dev/sdd2		1228800	7741439	3256320	83	Linux

図4.4 パーティション確認 (4G バイト SD カード)

4.4.2 ブートローダ格納

図4.5にブートローダを SD カードに格納するコマンドを示します。

```
> sudo dd if=~/gohssc-6ul/source/uboot-imx/u-boot.imx of=/dev/sdd bs=512 seek=2 conv=fsync
566+0 レコード入力
566+0 レコード出力
289792 バイト (290kB) コピーされました、0.266997 秒、1.1MB/秒
```

図4.5 ブートローダ格納コマンド

4.4.3 カーネル格納

図4.6にカーネルを SD カードに格納するコマンドを示します。

```
> mkdir ~/mountpoint/linux
> sudo umount /dev/sdd1
> sudo mkfs.vfat /dev/sdd1
> sudo mount /dev/sdd1 ~/mountpoint/linux
> sudo cp ~/gohssc-6ul/source/kernel-source/arch/arm/boot/zImage ~/mountpoint/linux/
> sync
> sudo cp ~/gohssc-6ul/source/kernel-source/arch/arm/boot/dts/imx6ul-gohssc.dtb ~/mountpoint/linux/
> sync
> sudo umount ~/mountpoint/linux
```

図4.6 カーネル格納コマンド

カーネル格納後は”/dev/sdd1”を再度マウントし、フォルダ内に”zImage”ファイルと”imx6ul-gohssc.dtb”ファイルが存在することを確認してください。

4.4.4 ルートファイルシステム格納

図4.7にルートファイルシステムをSDカードに格納するコマンドを示します。

```
> mkdir ~/mountpoint/rootfs
> sudo umount /dev/sdd2
> sudo mkfs.ext3 /dev/sdd2
> sudo mount /dev/sdd2 ~/mountpoint/rootfs
> sudo mount -o loop -t ext3 ~/fsl-release-bsp/build-x11/tmp/deploy/images/imx6ulevk/fsl-image-gui-imx6ulevk.ext3 ¥
~/mountpoint/guiimage
> cd ~/mountpoint/guiimage
> sudo cp -a * ~/mountpoint/rootfs
> sync
> cd ..
> sudo umount ~/mountpoint/rootfs
> sudo umount ~/mountpoint/guiimage
```

図4.7 ルートファイルシステム格納コマンド

ルートファイルシステム格納後は”/dev/sdd2”を再度マウントし、フォルダ内に”fsl-image-gui-imx6ul-gohssc.ext3”と同じファイルやフォルダが存在することを確認してください。

4.5 ブートモード設定

GOHSSC-6UL は SD カードブートモードを使用します。ブートモードは DSW1～DSW4 の設定にて切り替え可能です。表4.2に SD カードブートモードの端子設定を示します。

表4.2 SD カードブートモード設定

項目	端子設定							
	8	7	6	5	4	3	2	1
DSW1	-	-	-	-	-	-	H	L
DSW2	L	H	L	L	L	L	L	L
DSW3	L	L	L	L	L	L	L	L
DSW4	L	L	L	L	L	L	L	L

4.6 シリアル通信ソフトウェアの設定

表4.3にシリアル通信ソフトウェアの設定値を示します。

表4.3 シリアル通信ソフトウェア設定

項目	設定
転送レート	115,200bps
データ長	8bit
ストップビット	1bit
パリティ	なし
フロー制御	なし

4.7 起動

GOHSSC-6UL の SSW1 が OFF の状態にて DSW1~DSW4 を4.5章の内容に設定し、4.4章にて作成した SD カードを SD スロットへ挿入後、SSW1 を ON にするとシリアルコンソール端末に起動ログが表示され、Linux が起動します。図4.8に起動ログを示します。

```
U-Boot 2015.04-imx_v2015.04_3.14.52_1.1.0_ga+g6cf684a (Mar 10 2016 - 10:56:24)
```

```
CPU: Freescale i.MX6UL rev1.0 at 396 MHz
CPU: Temperature 41 C
Reset cause: POR
Board: GOHSSC-6UL
I2C: ready
DRAM: 1 GiB
MMC: FSL_SDHC: 0, FSL_SDHC: 1
In: serial
Out: serial
Err: serial
switch to partitions #0, OK
mmc0 is current device
Net: FEC0
Normal Boot
Hit any key to stop autoboot: 0
switch to partitions #0, OK
mmc0 is current device
switch to partitions #0, OK
mmc0 is current device
** No boot file defined **
reading zImage
6008864 bytes read in 304 ms (18.9 MiB/s)
Booting from mmc ...
reading imx6ul-gohssc.dtb
30997 bytes read in 19 ms (1.6 MiB/s)
Kernel image @ 0x80800000 [ 0x000000 - 0x5bb020 ]
## Flattened Device Tree blob at 83000000
   Booting using the fdt blob at 0x83000000
   Using Device Tree in place at 83000000, end 8300a914

Starting kernel ...
:
Freescale i.MX Release Distro 3.14.52-1.1.0 imx6ulevk /dev/ttymx1
imx6ulevk login:
```

図4.8 起動ログ

シリアルコンソール端末に”imx6ulevk login:”の文字が表示されれば、Linux の起動完了です。シリアルコンソール端末に”root”と入力するとログイン可能です。

4.8 MAC アドレス設定

Linux へのログイン確認後は、GOHSSC-6UL の MAC アドレスを設定します。MAC アドレスは u-boot にて設定するため GOHSSC-6UL を再起動し、Linux の autoboot 起動前に ENTER キー押下にて u-boot コマンドモードを立ち上げます。図4.9に MAC アドレスの設定手順を示します。設定する MAC アドレスは GOHSSC-6UL 表面に貼付しております。

```
U-Boot 2015.04-imx_v2015.04_3.14.52_1.1.0_ga+g6cf684a (Mar 10 2016 - 10:56:24)

CPU:   Freescale i.MX6UL rev1.0 at 396 MHz
CPU:   Temperature 41 C
Reset cause: POR
Board: GOHSSC-6UL
I2C:   ready
DRAM:  1 GiB
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
In:    serial
Out:   serial
Err:   serial
switch to partitions #0, OK
mmc0 is current device
Net:   FEC0
Normal Boot
Hit any key to stop autoboot:  3
> <ENTER> ← Hit any key to stop autoboot 表示後 3 秒以内に ENTER キーを押下
> setenv ethaddr 70:B3:D5:67:1x:xx
> setenv eth1addr 70:B3:D5:67:1x:xx
> printenv
:
ethaddr 70:B3:D5:67:1x:xx ← 設定した MAC アドレスが表示されることを確認
eth1addr 70:B3:D5:67:1x:xx ← 設定した MAC アドレスが表示されることを確認
:
> saveenv
```

図4.9 MAC アドレス設定手順

5. u-boot 操作手順

本章では、本ソフトウェアの u-boot を用いた各機能の操作手順を示します。GOHSSC-6UL 起動時、Linux の autoboot 起動前である”Hit any key to stop autoboot:”カウントダウン中に、ENTER キーを押下して u-boot コマンドモードを立ち上げます。u-boot コマンドモードでは各種メモリのリード/ライトや、I²C、Ethernet、USB などの通信確認が行えます。

5.1 DDR

DDR は u-boot 起動時から使用可能であり、領域はアドレス 0x80000000~0xBFFFFFFF です。”md”コマンドにてメモリのリード、”mw”コマンドにてメモリのライトが可能です。図5.1にメモリリードコマンドを、図5.2にメモリライトコマンドをそれぞれ示します。

```

=> md
    md - memory display

    Usage:
    md [.b, .w, .l] address [# of objects]
=> md.b 0x80000000 0x10
    80000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....

```

図5.1 メモリリードコマンド

```

=> mw
    mw - memory write (fill)

    Usage:
    mw [.b, .w, .l] address value [count]
=> mw.b 0x80000000 0x12 0x10
=> md.b 0x80000000 0x10
    80000000: 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12  .....

```

図5.2 メモリライトコマンド

表5.1に本ソフトウェアの DDR 設定値を示します。

表5.1 DDR設定値

仕様	内容
クロック	396MHz
構成	512M バイト (32 M ワード×16 ビット×8 バンク) ×2
CAS レイテンシ	5
リフレッシュ間隔	64ms 毎の 8192 リフレッシュサイクル
バンク数	BA[2:0]で制御する 8 バンク動作

5.2 QSPI NOR FLASH

QSPI NOR FLASH は u-boot コマンド”sf”にてアクセス可能です。”sf probe”コマンドにて QSPI NOR FLASH デバイスを認識し、”sf read”コマンドにてメモリのリード、”sf write”コマンドにてメモリのライトが可能です。図5.3に QSPI NOR FLASH アクセスコマンドを示します。

```

=> sf
    sf - SPI flash sub-system

    Usage:
    sf probe [[bus:]cs] [hz] [mode] - init flash device on given SPI bus
                                and chip select
    sf read addr offset len - read `len` bytes starting at
                                `offset` to memory at `addr`
    sf write addr offset len      - write `len` bytes from memory
                                at `addr` to flash at `offset`
    sf erase offset [+|len]      - erase `len` bytes from `offset`
                                `+|len` round up `len` to block size
    sf update addr offset len    - erase and write `len` bytes from memory
                                at `addr` to flash at `offset`

=> sf probe
    SF: Detected N25Q256 with page size 256 Bytes, erase size 4 KiB, total 32 MiB
=> md.b 0x80000000 0x20
    80000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff    .."3DUfw.....
    80000010: ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff    .....
=> sf write 0x80000000 0 0x10
    SF: 16 bytes @ 0x0 Written: OK
=> sf read 0x80000010 0 0x10
    SF: 16 bytes @ 0x0 Read: OK
=> md.b 0x80000000 0x20
    80000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff    .."3DUfw.....
    80000010: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff    .."3DUfw.....

```

図5.3 QSPI NOR FLASH アクセスコマンド

5.3 NAND FLASH

NAND FLASH は u-boot コマンド”nand”にてアクセス可能です。”nand info”コマンドにて NAND FLASH デバイスを認識し、”nand read”コマンドにてメモリのリード、”nand write”コマンドにてメモリのライトが可能です。図5.4、図5.5に NAND FLASH アクセスコマンドを示します。

なお、NAND FLASH を使用するにはソースコードの修正が必要になります。詳細は7.2 NAND FLASHをご参照ください。

```
=> nand
nand - NAND sub-system

Usage:
nand info - show available NAND devices
nand device [dev] - show or set current device
nand read - addr off|partition size
nand write - addr off|partition size
    read/write 'size' bytes starting at offset 'off'
    to/from memory address 'addr', skipping bad blocks.
nand read.raw - addr off|partition [count]
nand write.raw - addr off|partition [count]
    Use read.raw/write.raw to avoid ECC and access the flash as-is.
nand write.trimffs - addr off|partition size
    write 'size' bytes starting at offset 'off' from memory address
    'addr', skipping bad blocks and dropping any pages at the end
    of eraseblocks that contain only 0xFF
nand erase[.spread] [clean] off size - erase 'size' bytes from offset 'off'
    With '.spread', erase enough for given file size, otherwise,
    'size' includes skipped bad blocks.
nand erase.part [clean] partition - erase entire mtd partition'
nand erase.chip [clean] - erase entire chip'
nand bad - show bad blocks
nand dump[.oob] off - dump page
nand scrub [-y] off size | scrub.part partition | scrub.chip
    really clean NAND erasing bad blocks (UNSAFE)
nand markbad off [...] - mark bad block(s) at offset (UNSAFE)
nand biterr off - make a bit error at offset (UNSAFE)
```

図5.4 NAND FLASH アクセスコマンド(1/2)


```
=> nand info

Device 0: nand0, sector size 2048 KiB
  Page size      8192 b
  OOB size       744 b
  Erase size     2097152 b
=> md.b 0x80000000 0x20
80000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
80000010: c5 0f cd 0f 25 8b cd 3f c1 ef c5 0f 45 0e cf af  ....%..?....E...
=> nand write 0x80000000 0 0x10

NAND write: device 0 offset 0x0, size 0x10
  16 bytes written: OK
=> nand read 0x80000010 0 0x10

NAND read: device 0 offset 0x0, size 0x10
  16 bytes read: OK
=> md.b 0x80000000 0x20
80000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
80000010: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
```

図5.5 NAND FLASH アクセスコマンド(2/2)

5.4 SD/MMC

SD/MMC は u-boot コマンド”mmc”にてアクセス可能です。”mmc dev”コマンドにて SD/MMC デバイスを認識し、”mmc read”コマンドにてメモリのリード、”mmc write”コマンドにてメモリのライトが可能です。図5.6、図5.7に SD/MMC アクセスコマンドを示します。

```
=> mmc
mmc - MMC sub system

Usage:
mmc info - display info of the current MMC device
mmc read addr blk# cnt
mmc write addr blk# cnt
mmc erase blk# cnt
mmc rescan
mmc part - lists available partition on current mmc device
mmc dev [dev] [part] - show or set current mmc device [partition]
mmc list - lists available devices
mmc hwpartition [args...] - does hardware partitioning
arguments (sizes in 512-byte blocks):
[user [enh start cnt] [wrrel {on|off}]] - sets user data area attributes
[gp1|gp2|gp3|gp4 cnt [enh] [wrrel {on|off}]] - general purpose partition
[check|set|complete] - mode, complete set partitioning completed
WARNING: Partitioning is a write-once setting once it is set to complete.
Power cycling is required to initialize partitions after set to complete.
mmc bootbus dev boot_bus_width reset_boot_bus_width boot_mode
- Set the BOOT_BUS_WIDTH field of the specified device
mmc bootpart-resize <dev> <boot part size MB> <RPMB part size MB>
- Change sizes of boot and RPMB partitions of specified device
mmc partconf dev boot_ack boot_partition partition_access
- Change the bits of the PARTITION_CONFIG field of the specified device
mmc rst-function dev value
- Change the RST_n_FUNCTION field of the specified device
WARNING: This is a write-once field and 0 / 1 / 2 are the only valid values.
mmc setdsr <value> - set DSR register value
```

図5.6 SD/MMC アクセスコマンド(1/2)

```
=> mmc dev 0
    switch to partitions #0, OK
    mmc0 is current device
=> mmc info
    Device: FSL_SDHC
    Manufacturer ID: 2
    OEM: 544d
    Name: SD04G
    Tran Speed: 50000000
    Rd Block Len: 512
    SD version 3.0
    High Capacity: Yes
    Capacity: 3.7 GiB
    Bus Width: 4-bit
    Erase Group Size: 512 Bytes
=> md.b 0x80001000 0x10
    80001000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
=> md.b 0x80000000 0x10
    80000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
=> mmc write 0x80001000 0 1

    MMC write: dev # 0, block # 0, count 1 ... 1 blocks written: OK
=> mmc read 0x80000000 0 1

    MMC read: dev # 0, block # 0, count 1 ... 1 blocks read: OK
=> md.b 0x80000000 0x10
    80000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
```

図5.7 SD/MMC アクセスコマンド(2/2)

5.5 I²C

I²C は u-boot コマンド "i2c" にて制御可能です。"i2c dev" コマンドにて I²C のチャンネルを選択し、"i2c probe" コマンドにて接続先のデバイスアドレスを取得します。その後、"i2c md" コマンドにてリードコマンドの送信、"i2c mm" コマンドにてライトコマンドの送信が可能です。図5.8に I²C 制御コマンドを示します。

```
=> i2c
    i2c - I2C sub-system

Usage:
i2c bus [muxtype:muxaddr:muxchannel] - show I2C bus info
crc32 chip address[.0, .1, .2] count - compute CRC32 checksum
i2c dev [dev] - show or set current I2C bus
i2c loop chip address[.0, .1, .2] [# of objects] - looping read of device
i2c md chip address[.0, .1, .2] [# of objects] - read from I2C device
i2c mm chip address[.0, .1, .2] - write to I2C device (auto-incrementing)
i2c mw chip address[.0, .1, .2] value [count] - write to I2C device (fill)
i2c nm chip address[.0, .1, .2] - write to I2C device (constant address)
i2c probe [address] - test for and show device(s) on the I2C bus
i2c read chip address[.0, .1, .2] length memaddress - read to memory
i2c write memaddress chip address[.0, .1, .2] length [-s] - write memory
        to I2C; the -s option selects bulk write in a single transaction
i2c reset - re-init the I2C Controller
i2c speed [speed] - show or set I2C bus speed
=> i2c dev 1
    Setting bus to 1
=> i2c probe
    Valid chip addresses: 08 1A 50
=> i2c mw 0x50 0x0000.2 0x12 1
=> i2c md 0x50 0x0000.2 1
    0000: 12    .
=> i2c mw 0x50 0x0000.2 0x34 1
=> i2c md 0x50 0x0000.2 1
    0000: 34    4
```

図5.8 I²C 制御コマンド

5.6 Ethernet

Ethernet はネットワーク内に DHCP サーバが存在する場合、u-boot コマンド”dhcp”にて IP アドレスの自動割り当てが可能です。ネットワーク内に DHCP サーバが存在しない場合は、”setenv”コマンドにて IP アドレス、サブネットマスク、デフォルトゲートウェイが設定可能です。IP アドレス設定後は”ping”コマンドにて通信確認が可能です。図5.9に IP アドレス自動割り当てのコマンド、図5.10に IP アドレス手動割り当てのコマンドを示します。なお、u-boot では Ethernet コネクタ CN6 のみ有効です。

```
=> dhcp
    BOOTP broadcast 1
    DHCP client bound to address 192.168.1.72 (129 ms)
=> print
gatewayip=192.168.1.1 ← 自動的に割り当てられていることを確認
ipaddr=192.168.1.72 ←自動的に割り当てられていることを確認
netmask=255.255.255.0 ←自動的に割り当てられていることを確認
=> ping 192.168.1.1
    Using FEC0 device
    host 192.168.1.1 is alive
```

図5.9 IP アドレス自動割り当てコマンド

```
=> setenv ipaddr 192.168.1.72
=> setenv gatewayip 192.168.1.1
=> setenv netmask 255.255.255.0
=> print
gatewayip=192.168.1.1 ← 設定内容を確認
ipaddr=192.168.1.72 ←設定内容を確認
netmask=255.255.255.0 ←設定内容を確認
=> ping 192.168.1.1
    Using FEC0 device
    host 192.168.1.1 is alive
```

図5.10 IP アドレス手動割り当てコマンド

5.7 USB

USB は u-boot コマンド”usb”にて制御可能です。”usb start”コマンドにて USB 接続デバイスを認識し、接続デバイスが USB FLASH メモリの場合、”usb read”コマンドにてメモリのリード、”usb write”コマンドにてメモリのライトが可能です。図5.11、図5.12に USB 制御コマンドを示します。

```
=> usb
usb - USB sub-system

Usage:
usb start - start (scan) USB controller
usb reset - reset (rescan) USB controller
usb stop [f] - stop USB [f]=force stop
usb tree - show USB device tree
usb info [dev] - show available USB devices
usb test [dev] [port] [mode] - set USB 2.0 test mode
    (specify port 0 to indicate the device's upstream port)
    Available modes: J, K, S[E0_NAK], P[acket], F[orce_Enable]
usb storage - show details of USB storage devices
usb dev [dev] - show or set current USB storage device
usb part [dev] - print partition table of one or all USB storage devices
usb read addr blk# cnt - read `cnt` blocks starting at block `blk#`
    to memory address `addr`
usb write addr blk# cnt - write `cnt` blocks starting at block `blk#`
    from memory address `addr`
```

図5.11 USB 制御コマンド(1/2)

```
=> usb start
starting USB...
USB0: Port not available.
USB1: USB EHCI 1.00
scanning bus 1 for devices... 2 USB Device(s) found
      scanning usb for storage devices... Device NOT ready
      Request Sense returned 06 28 00
      1 Storage Device(s) found
      scanning usb for ethernet devices... 0 Ethernet Device(s) found
=> md.b 0x80000000 0x10
80000000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
=> md.b 0x80001000 0x10
80001000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
=> usb write 0x80000000 0x100 1

USB write: device 0 block # 256, count 1 ... 1 blocks write: OK
=> usb read 0x80001000 0x100 1

USB read: device 0 block # 256, count 1 ... 1 blocks read: OK
=> md.b 0x80001000 0x10
80001000: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff  .."3DUfw.....
```

図5.12 USB 制御コマンド(2/2)

6. Linux 操作手順

本章では、本ソフトウェアの Linux を用いた各機能の操作手順を示します。GOHSSC-6UL を起動し、Linux の autoboot 起動前である” Hit any key to stop autoboot:”のカウン트가 0 になると、Linux が起動し始めます。起動完了するとシリアルコンソール端末に”imx6ulevk login ”と表示されますので、”root”と入力しログインします。以降、Linux コマンドによる端末操作が可能になります。

6.1 QSPI NOR FLASH

QSPI NOR FLASH はマウントすることでメモリ領域として使用可能です。図6.1に QSPI NOR FLASH のマウントコマンドを示します。マウント以降、qspi フォルダ内に作成したファイルは QSPI NOR FLASH に保存されます。

```
> mkfs.ext4 /dev/mtdblock0 ← ext4 形式にてフォーマット
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
8192 inodes, 32768 blocks
1638 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=33554432
4 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

> mkdir ~/qspi
> mount /dev/mtdblock0 ~/qspi
EXT4-fs (mtdblock0): mounted filesystem with ordered data mode
```

図6.1 QSPI NOR FLASH マウントコマンド

6.2 NAND FLASH

NAND FLASH はマウントすることでメモリ領域として使用可能です。図6.1に NAND FLASH のマウントコマンドを示します。マウント以降、nand フォルダ内に作成したファイルは NAND FLASH に保存されます。

なお、NAND FLASH を使用するにはソースコードの修正が必要になります。詳細は7.2 NAND FLASHをご参照ください。

```
> flash_eraseall /dev/mtd0 ← 全領域フォーマット
Erasing 2048 Kibyte @ ffe00000 -- 100 % complete
> ubiattach /dev/ubi_ctrl -m 0
UBI: attaching mtd0 to ubi0
UBI: scanning is finished
UBI: empty MTD device detected
UBI: attached mtd0 (name "gpmi-nand", size 4096 MiB) to ubi0
UBI: PEB size: 2097152 bytes (2048 KiB), LEB size: 2080768 bytes
UBI: min./max. I/O unit sizes: 8192/8192, sub-page size 8192
UBI: VID header offset: 8192 (aligned 8192), data offset: 16384
UBI: good PEBs: 2042, bad PEBs: 6, corrupted PEBs: 0
UBI: user volume: 0, internal volumes: 1, max. volumes count: 128
UBI: max/mean erase counter: 0/0, WL threshold: 4096, image sequence number: 1206303894
UBI: available PEBs: 2004, total reserved PEBs: 38, PEBs reserved for bad PEB handling: 34
UBI: background thread "ubi_bgt0d" started, PID 823
UBI device number 0, total 2042 LEBs (4248928256 bytes, 4.0 GiB), available 2004 LEBs (4169859072 bytes, 3.9
GiB), LEB size 2080768 bytes (2.0 MiB)
> ubimkvol /dev/ubi0 -N rootfs -s 128MiB
Volume ID 0, size 65 LEBs (135249920 bytes, 129.0 MiB), LEB size 2080768 bytes (2.0 MiB), dynamic, name
"rootfs", alignment 1
> mount -t ubifs ubi0:rootfs ~/nand
UBIFS: default file-system created
UBIFS: background thread "ubifs_bgt0_0" started, PID 830
UBIFS: mounted UBI device 0, volume 0, name "rootfs"
UBIFS: LEB size: 2080768 bytes (2032 KiB), min./max. I/O unit sizes: 8192 bytes/8192 bytes
UBIFS: FS size: 114442240 bytes (109 MiB, 55 LEBs), journal size 16646145 bytes (15 MiB, 6 LEBs)
UBIFS: reserved for root: 4952683 bytes (4836 KiB)
UBIFS: media format: w4/r0 (latest is w4/r0), UUID 577D4B8E-4ED9-4850-9ACD-E6BA58679C43, small LPT model
```

図6.2 NAND FLASH マウントコマンド

6.3 SD カード

SD カードはマウントすることでメモリ領域として使用可能です。図6.3に SD カードのマウントコマンドを示します。マウント以降、sd フォルダ内に作成したファイルは SD カードに保存されます。

```
> mkdir ~/sd
> mount /dev/mmcblk0p1 ~/sd
FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be corrupt. Please run fsck.
> cd ~/sd
> ls
imx6ul-gohssc.dtb  zImage
```

図6.3 SD カードマウントコマンド

6.4 e-MMC

e-MMC はマウントすることでメモリ領域として使用可能です。図6.4に e-MMC のマウントコマンドを示します。マウント以降、emmc フォルダ内に作成したファイルは e-MMC に保存されます。

```
> fdisk /dev/mmcblk1
Device contains neither a valid DOS partition table, nor Sun, SGI, OSF or GPT disklabel
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that the previous content
won't be recoverable.

The number of cylinders for this disk is set to 236032.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

> Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
> p
> Partition number (1-4): 1
> First cylinder (1-236032, default 1): 1
> Last cylinder or +size or +sizeM or +sizeK (1-236032, default 236032): +128M
```

図6.4 e-MMC マウントコマンド (1/2)

```
> Command (m for help): p

Disk /dev/mmcbk1: 7734 MB, 7734296576 bytes
4 heads, 16 sectors/track, 236032 cylinders
Units = cylinders of 64 * 512 = 32768 bytes

    Device Boot      Start         End      Blocks   Id  System
/dev/mmcbk1p1          1         3907     125016   83  Linux

> Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table
mmcbk1: p1
> mkfs.ext4 /dev/mmcbk1p1 ← ext4 形式にてフォーマット
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
31360 inodes, 125016 blocks
6250 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
16 block groups
8192 blocks per group, 8192 fragments per group
1960 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

> mkdir ~/emmc
> mount /dev/mmcbk1p1 ~/emmc
EXT4-fs (mmcbk1p1): mounted filesystem with ordered data mode. Opts: (null)
```

図6.5 e-MMC マウントコマンド (2/2)

6.5 Ethernet

Ethernet はネットワーク内に DHCP サーバが存在する場合、自動的に IP アドレスが割り当てられます。IP アドレス設定後は”ping”コマンドにて通信確認が可能です。図6.6に Ethernet 通信コマンド示します。また、図6.7、図6.8に IP アドレスを固定にする場合のコマンドを示します。図中の MAC アドレスや設定する IP アドレスは適宜変更してください。

```
> ifconfig
eth0    Link encap:Ethernet  HWaddr 70:B3:D5:67:1x:xx
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1    Link encap:Ethernet  HWaddr 70:B3:D5:67:1x:xx
        inet addr:192.168.1.72  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::201:2ff:fe03:408/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:41 errors:0 dropped:9 overruns:0 frame:0
        TX packets:42 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:4555 (4.4 KiB)  TX bytes:7479 (7.3 KiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:10 errors:0 dropped:0 overruns:0 frame:0
        TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:700 (700.0 B)  TX bytes:700 (700.0 B)

> ping 192.168.1.13
PING 192.168.1.13 (192.168.1.13): 56 data bytes
64 bytes from 192.168.1.13: seq=0 ttl=64 time=0.721 ms
64 bytes from 192.168.1.13: seq=1 ttl=64 time=0.552 ms
64 bytes from 192.168.1.13: seq=2 ttl=64 time=0.607 ms
64 bytes from 192.168.1.13: seq=3 ttl=64 time=0.651 ms
64 bytes from 192.168.1.13: seq=4 ttl=64 time=0.556 ms
64 bytes from 192.168.1.13: seq=5 ttl=64 time=0.556 ms
> <Ctrl+C>
^C
--- 192.168.1.13 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.552/0.607/0.721 ms
```

図6.6 Ethernet 通信コマンド

```
> /usr/lib/connman/test/get-services
[ /net/connman/service/ethernet_70B3D5671xxx_cable ]
  IPv6.Configuration = { Method=auto Privacy=disabled }
  AutoConnect = true
  Name = Wired
  Nameservers = [ 192.168.1.1 125.170.93.233 210.145.254.169 ]
  Provider = { }
  Favorite = true
  Domains.Configuration = [ ]
  Timeservers.Configuration = dbus.Array([], signature=dbus.Signature('s'), variant_level=1)
  State = online
  Proxy = { Method=direct }
  Nameservers.Configuration = [ ]
  IPv4 = { Netmask=255.255.255.0 Gateway=192.168.1.1 Method=dhcp Address=192.168.1.72 }
  Timeservers = dbus.Array([dbus.String(u'192.168.1.1'), signature=dbus.Signature('s'), variant_level=1)
  IPv6 = { }
  Domains = [ ]
  Ethernet = { Interface=eth1 MTU=1500 Method=auto Address=70:B3:D5:67:1x:xx }
  Security = [ ]
  Proxy.Configuration = { }
  Type = ethernet
  Immutable = false
  IPv4.Configuration = { Method=dhcp }

> /usr/lib/connman/test/set-nameservers
Usage: /usr/lib/connman/test/set-nameservers <service> [nameserver*]
> /usr/lib/connman/test/set-ipv4-method ethernet_70B3D5671xxx_cable manual 192.168.1.100 255.255.255.0 ¥
> 192.168.1.1                               ↑ 70B3D5671xxx は MAC アドレスのため適宜変更する
Setting method manual for ethernet_70B3D5671xxx_cable
New IPv4.Configuration:  {'Netmask': dbus.String(u'255.255.255.0', variant_level=1), 'Gateway':
dbus.String(u'192.168.1.1', variant_level=1), 'Method': dbus.String(u'manual', variant_level=1), 'Address':
dbus.String(u'192.168.1.100', variant_level=1)}
```

図6.7 固定 IP アドレス設定コマンド(1/2)

```
> ifconfig
eth0      Link encap:Ethernet  HWaddr 70:B3:D5:67:1x:xx
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 70:B3:D5:67:1x:xx
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::201:2ff:fe03:408/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:739 errors:0 dropped:173 overruns:0 frame:0
          TX packets:64 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:62128 (60.6 KiB)  TX bytes:12982 (12.6 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:700 (700.0 B)  TX bytes:700 (700.0 B)

> ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: seq=0 ttl=255 time=16.310 ms
64 bytes from 192.168.1.1: seq=1 ttl=255 time=1.021 ms
64 bytes from 192.168.1.1: seq=2 ttl=255 time=0.990 ms
64 bytes from 192.168.1.1: seq=3 ttl=255 time=3.661 ms
64 bytes from 192.168.1.1: seq=4 ttl=255 time=2.681 ms
64 bytes from 192.168.1.1: seq=5 ttl=255 time=3.091 ms
64 bytes from 192.168.1.1: seq=6 ttl=255 time=0.982 ms
> <Ctrl+C>
^C
--- 192.168.1.1 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.982/4.105/16.310 ms
```

図6.8 固定 IP アドレス設定コマンド(2/2)

6.6 USB

USB2.0 Micro TypeAB コネクタ (CN2) や、USB2.0 TypeA コネクタ (CN3) に、USB マウスや USB キーボード、USB フラッシュメモリを接続すると、自動的に認識し、使用可能になります。USB FLASH メモリを使用する場合は、"mount"コマンドにてマウントする必要があります。図6.9に USB マウントコマンドを示します。マウント以降、usb フォルダ内に作成したファイルは USB FLASH メモリに保存されます。

```
> mkdir ~/usb
> mount /dev/sda ~/usb
```

図6.9 USB マウントコマンド

6.7 オーディオ

ヘッドフォンコネクタ (CN8) にスピーカを接続すると、WAV ファイルの再生が可能になります。WAV ファイルの再生は"aplay"コマンドを使用します。また、マイクコネクタ (CN9) にマイクを接続すると、WAV ファイルの録音が可能になります。WAV ファイルの録音は"arecord"コマンドを使用します。図6.10にオーディオ再生/録音コマンドを示します。

```
> arecord -Dhw:0,0 -r32000 -fS16_LE -c2 -d5 ~/record.wav
Recording WAVE '/home/root/record.wav' : Signed 16 bit Little Endian, Rate 32000 Hz, Stereo
> aplay ~/record.wav
Playing WAVE '/home/root/record.wav' : Signed 16 bit Little Endian, Rate 32000 Hz, Stereo
```

図6.10 オーディオ再生/録音コマンド

6.8 CAN

CAN コネクタ (CN4) に CAN 通信の対向機を接続すると、CAN 通信が可能になります。"cansend"コマンドにてデータを送信、"candump"コマンドにてデータを受信します。図6.11に CAN 通信コマンドを示します。

```
> /sbin/ip link set can0 up type can bitrate 500000
> cansend can0 123#12345678
> candump can0
```

図6.11 CAN 通信コマンド

7. 拡張機能

本章では、本ソフトウェアの設定変更により使用可能になる機能について示します。使用可能になる機能は以下の5つです。

- ・ HDMI
- ・ NAND FLASH
- ・ USB2.0 MicroAB (Host)
- ・ センサ（磁気／ジャイロ／加速度）
- ・ e-MMC ブート

7.1 HDMI

HDMI ドライバを有効にします。有効にすると、HDMI による 720p の映像出力が可能になります。図7.1に HDMI 設定変更箇所を示します。なお、GOHSSC-6UL 起動時から HDMI ケーブルにてモニタと接続しておく必要がありますのでご注意ください。

```
<ファイル名>
~/gohssc-6ul/source/kernel-source/arch/arm/boot/dts/imx6ul-gohssc.dts

<変更内容>
:
    sii902x_reset: sii902x-reset {
        compatible = "gpio-reset";
        reset-gpios = <&gpio3 4 GPIO_ACTIVE_LOW>;
        reset-delay-us = <100000>;
        #reset-cells = <0>;
        status = "okay"; ← status を"okay"に変更
    };
:
    sii902x@39 {
        compatible = "SiI,sii902x";
        interrupt-parent = <&gpio5>;
        interrupts = <2 IRQ_TYPE_EDGE_FALLING>;
        mode_str = "1280x720M@60";
        bits-per-pixel = <16>;
        resets = <&sii902x_reset>;
        reg = <0x39>;
        status = "okay"; ← status を"okay"に変更
    };
```

図7.1 HDMI 設定変更箇所

7.2 NAND FLASH

NAND FLASH ドライバを有効にします。有効にすると、NAND FLASH へのメモリアクセスが可能になります。図7.2、図7.3に NAND FLASH 設定変更箇所を示します。なお、NAND FLASH ドライバは GOHSSC-6UL-B にてご利用ください。

<ファイル名>

```
~/gohssc-6ul/source/kernel-source/arch/arm/boot/dts/imx6ul-gohssc.dts
```

<変更内容>

```
#define GOHSSC_CONFIG_NAND ← NAND FLASH 用定義を有効にする  
/* #define GOHSSC_CONFIG_QSPI */ ← QSPI FLASH 用定義を無効にする
```

図7.2 NAND FLASH 設定変更箇所(1/2)

<ファイル名>

```
~/gohssc-6ul/source/uboot-imx/include/configs/mx6ul_gohssc.h
```

<変更内容>

```
#define CONFIG_SYS_USE_NAND ← NAND FLASH 用定義を有効にする
```

図7.3 NAND FLASH 設定変更箇所(2/2)

7.3 USB2.0 Micro AB

USB2.0 Micro AB ドライバを有効にします。有効にすると、USB2.0 Micro AB の HOST 機能が有効になります。図7.4、図7.5に USB2.0 Micro AB 設定変更箇所を示します。ファンクション機能として使う場合は、変更箇所を元に戻してください。

```
<ファイル名>
~/gohssc-6ul/source/uboot-imx/board/gohshoji/mx6ul_gohssc/mx6ul_gohssc.c

<変更内容>
static iomux_v3_cfg_t const usb_otg_pads[] = {
    MX6_PAD_GPIO1_IO04__USB_OTG1_PWR | MUX_PAD_CTRL(NO_PAD_CTRL), ←USB 用定義を有効にする
    MX6_PAD_GPIO1_IO02__USB_OTG2_PWR | MUX_PAD_CTRL(NO_PAD_CTRL),
    MX6_PAD_GPIO1_IO00__ANATOP_OTG1_ID | MUX_PAD_CTRL(OTG_ID_PAD_CTRL),
};
```

図7.4 USB2.0 Micro AB 設定変更箇所(1/2)

```
<ファイル名>
~/gohssc-6ul/source/kernel-source/arch/arm/boot/dts/imx6ul-gohssc.dts

<変更内容>
#define GOHSSC_CONFIG_USB_HOST ←USB 用定義を有効にする
```

図7.5 USB2.0 Micro AB 設定変更箇所(2/2)

7.4 センサ

センサドライバを有効にします。有効にすると、I²C チャンネル 4 にてセンサ制御が可能になります。図7.6、図7.7にセンサ設定変更箇所を示します。

```
<ファイル名>
~/gohssc-6ul/source/kernel-source/arch/arm/boot/dts/imx6ul-gohssc.dts

<変更内容>
#define GOHSSC_CONFIG_SENSOR ← SENSOR 用定義を有効にする
/* #define GOHSSC_CONFIG_QSPI */ ← QSPI NOR FLASH 用定義を無効にする
```

図7.6 センサ設定変更箇所(1/2)

```
<ファイル名>
~/gohssc-6ul/source/uboot-imx/include/configs/mx6ul_gohssc.h

<変更内容>
#define CONFIG_SENSOR ← SENSOR 用定義を有効にする
/* #define CONFIG_FSL_QSPI */ ← QSPI NOR FLASH 用定義を無効にする
```

図7.7 センサ設定変更箇所(2/2)

7.5 e-MMC ブート

DSW1~DSW4 を切り替えると、e-MMC ブートモードに変更可能です。e-MMC ブート用のメモリマップは図4.2に示す SD カードブートと同じです。表7.1に e-MMC ブートモードの端子設定を示します。

表7.1 e-MMC ブートモード設定

項目	端子設定							
	8	7	6	5	4	3	2	1
DSW1	-	-	-	-	-	-	H	L
DSW2	L	H	H	L	L	L	L	L
DSW3	L	L	L	L	H	L	L	L
DSW4	L	L	L	L	L	L	L	L

赤色 : SD ブートモードからの変更箇所

改訂記録

Ver.	発行日	ページ	修正箇所
Ver.1.00	2015.12.22	—	初版発行
Ver.1.01	2016.03.11	全体	Linux カーネルバージョンを 3.14.52 に変更
		全体	フリースケールの名称を NXP に変更
		39	図 3.1 開発環境フォルダ構成図のツールチェーン格納フォルダを変更
		41	図 3.6 ツールチェーン入手コマンドのファイルパス変更 図 3.7 ツールチェーン環境 PATH 設定コマンドおよび説明文を変更
		42	図 3.9 u-boot ビルドコマンドのファイルパス変更
		44	図 3.12 ルートファイルシステムマウントコマンドのファイルパス変更
		50	図 4.7 ルートファイルシステム格納コマンドのファイルパス変更
		52	図 4.8 起動ログ変更
		53	図 4.9 MAC アドレス設定手順のログおよびコマンド変更
		56	u-boot の NAND FLASH アクセス説明文変更
		64	図 6.1 QSPI NOR FLASH マウントコマンド変更
		65	Linux の NAND FLASH アクセス説明文変更
		68	図 6.6 Ethernet 通信コマンドの補足説明追加
		69	図 6.7 固定 IP アドレス設定コマンド(1/2)に MAC アドレスの補足説明追加
70	図 6.8 固定 IP アドレス設定コマンド(2/2)にコマンドの補足説明追加		
Ver.1.02	2018.01.24	11	図 1.3 ブロック図に TAMPER モジュールを追加
		33,34	SNVS_TAMPER 信号の接続先を NC からコネクタへ変更

i.MX 6UL スタータボード
GOHSSC-6UL ソフトウェアマニュアル

発行年月日 2018年01月24日 Ver.1.02
発行 郷 商事 株式会社

i.MX 6UL スタータボード
GOHSSC-6UL
ソフトウェアマニュアル

