

R32C/111 グループ

LCD ディレクトドライブ

要旨

この資料では、R32C を用いた LCD 制御例を掲載しています。
各機能の詳細については、ハードウェアマニュアルを参照してください。

動作確認デバイス

R32C/111 RSK 基板 R0K564112S001BE(P/N)と独 Segger 社開発 LCD アドオンボードを使用しています。

<http://www.segger.com/cms/renesas-rsk2r32c111-eval-board---segger-visualization-add-on.html>

目次

1. システム構成.....	2
2. 使用する周辺機能.....	3
3. クロックの設定.....	4
4. UARTの設定.....	4
5. LCD制御タイミング.....	5
6. 参考プログラム.....	7
7. 参考ドキュメント.....	9

1. システム構成

LCD ダイレクトドライブは、R32C/111 内蔵の DMAC と UART を使用して LCD を制御するため、別途 LCD コントローラ IC を必要としません。

図 1 に R32C/111 を用いた LCD ダイレクトドライブ (カラー 8 色) のブロック図を示します。

LCD の制御には DMAC を 3 チャンネル、UART を 3 チャンネルとプログラマブル入出力ポートを 2 本または 1 本を使用します。また画像バッファとして内部 RAM を約 29KB 使用します。

映像信号 (Rx, Gx, Bx, DCLK) はクロック同期シリアル、同期信号 (Vsync, Hsync または DEN) はプログラマブル入出力ポートで接続しています。接続方法によりカラー 8 色、8 階調グレースケール、UART を 1 チャンネルに減らして白黒画像にすることも可能です。

輝度調整や電源制御はドライバに含まれませんので、別途ご用意ください。

LCD 制御ドライバの CPU 負荷は約 45% です。使用する LCD モジュールの制御タイミングにより前後します。

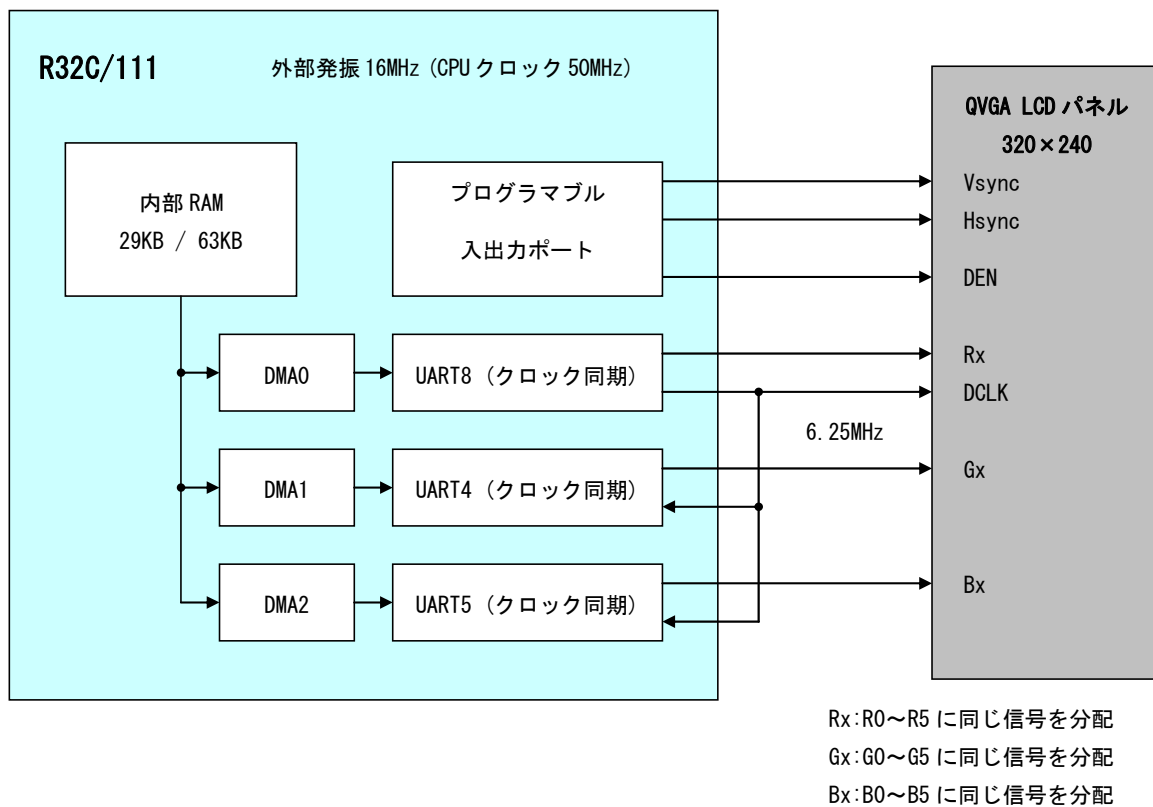


図 1 ブロック図

2. 使用する周辺機能

各周辺機能を以下の用途で使します。

機能名	用途
UART8	R 信号と DCLK を出力します。
UART4	G 信号を出力します。(外部クロック)
UART5	B 信号を出力します。(外部クロック)
DMA0	画像バッファから SIO8 への転送に使します。
DMA1	画像バッファから SIO4 への転送に使します。
DMA2	画像バッファから SIO5 への転送に使します。
P9_7	Vsync を出力します。
P7_5	Hsync を出力します。
P8_0	DEN を出力します。
内部 RAM	画像バッファとして約 29KB 使します。

画素数 320×240 のカラー8色(R,G,B 各 1bit)の場合、約 29KB の内部 RAM を使します。

$$320 \times 240 \times 3 = 230400[\text{bit}] = 28800[\text{byte}]$$

R,G,B の成分毎に連続したデータを必要とします。

一つの成分の画素データの並びは、LCD の画素(x,y)を(0,0)~(319,239)とすると以下のようになります。MSB 側が画面の左側になります。

	+0	+1	...	+39
+0	(0,0)~(7,0)	(8,0)~(15,0)	...	(312,0)~(319,0)
+40	(0,1)~(7,1)	(8,1)~(15,1)	...	(312,1)~(319,1)
	:	:		:
+9520	(0,238)~(7,238)	(8,238)~(15,238)	...	(312,238)~(319,238)
+9560	(0,239)~(7,239)	(8,239)~(15,239)	...	(312,239)~(319,239)

サンプルコードでは以下のように宣言しています。

```
#define LCD_WIDTH 320
#define LCD_HIGHT 240
#define LCD_WIDTH_byte (LCD_WIDTH/8)

uint8_t buffer_r[LCD_WIDTH_byte * LCD_HIGHT]; // red
uint8_t buffer_g[LCD_WIDTH_byte * LCD_HIGHT]; // green
uint8_t buffer_b[LCD_WIDTH_byte * LCD_HIGHT]; // blue
```

3. クロックの設定

各種クロックは以下の設定で使用します。

RSK 用のプロジェクト(RSKR32C111)を作成した場合は以下の設定になっています。

クロック	周波数
メインクロック	16MHz (外部発振)
PLL クロック	100MHz (=メインクロック*6.25)
ベースクロック	50MHz (=PLL クロック/2)
CPU クロック	50MHz (=ベースクロック)
周辺バスクロック	25MHz (=バスクロック/2)
周辺機能クロック	25MHz (=PLL クロック/4)

4. UARTの設定

クロック同期型シリアルインターフェースモードを使用します。

UART8 は内部クロックに設定し、UART4, UART5 は外部クロックに設定します。UART8 が出力するクロックを UART4, UART5 の外部クロックとして入力し 3 つの UART を同期させます。

クロックの速度は 6.25[MHz]に設定します。(U8BRG のカウントソース=f1、U8BRG=1)

$$f1 / 2 * (U8BRG + 1) = 25[MHz] / 4 = 6.25[MHz]$$

極性は CKPOL=0 (送受信クロックの立ち下がりに同期して送信データ出力、立ち上がりに同期して受信データ入力) に設定します。

UART の設定は、使用する LCD モジュールの仕様にあわせて設定する必要があります。

サンプルコードの関数 ConfigureUARTs を参照ください。

5. LCD制御タイミング

この資料で使用している LCD モジュールでは 2 種類の制御方法(V-SYNC と H-SYNC を用いる"SYNC Mode"と、DEN を用いる"DE Mode")があります。サンプルコードでは DE Mode を使用しています。

各 UART の連続送信には DMAC を使用します。DMA0~DMA2 を 8bit の単転送モードで使用し、ソースアドレスに画像バッファ、デスティネーションアドレスに各 UART の送信レジスタを設定します。

割り込みは DMA0 のみ使用し、割り込み処理内で同期信号の生成と水平 1 ラインの DMA 転送(3 チャンネル分)の初期設定を行います。

水平方向は 320 ドットあり、データ量としては 40 バイトになります。最初の 1 バイトは CPU が書き込みを行い、残りの 39 バイトを DMAC で転送します。R,G,B の 3 つの UART の同期を取るため、クロックを出力する UART8 を最後に書き込む必要があります。

SYNC Modeの制御タイミングを図 2、DE Modeの制御タイミングを図 3に示します。

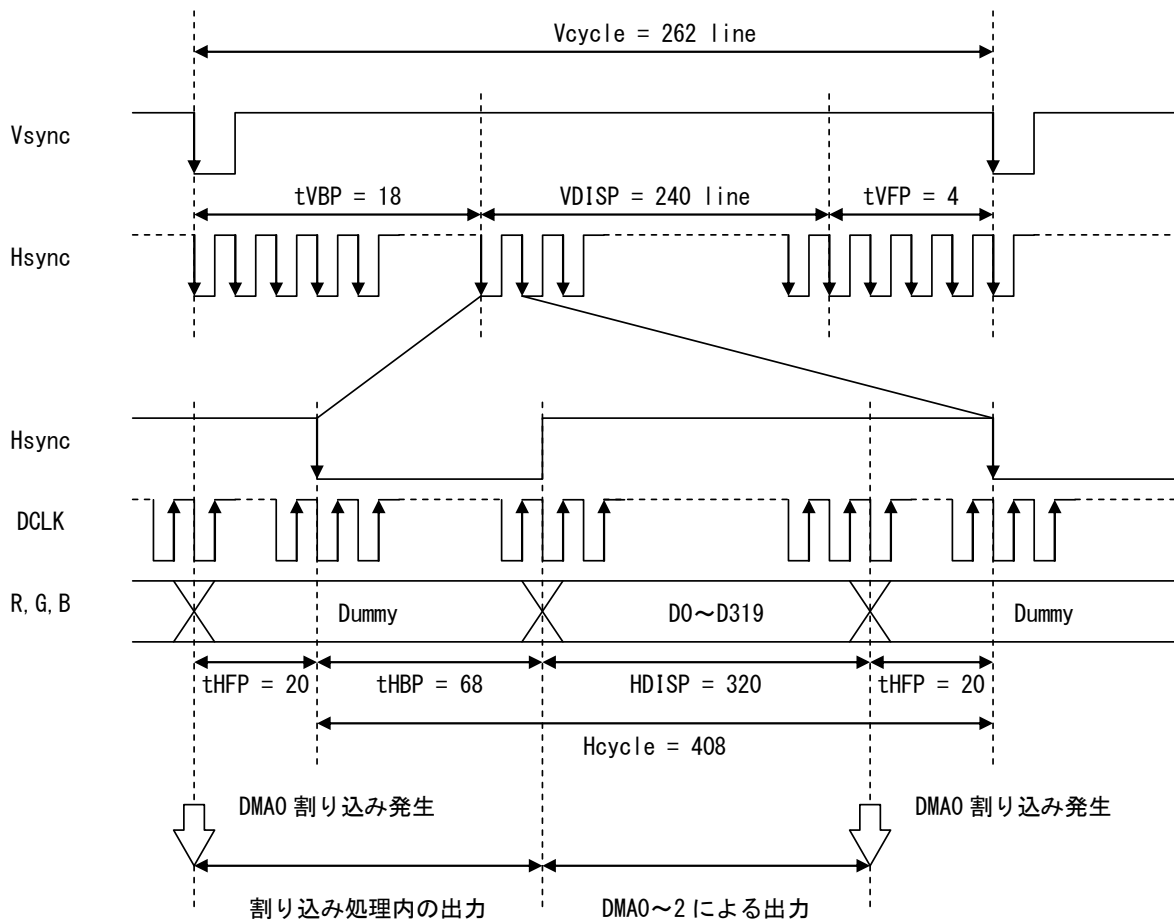


図 2 制御タイミング (SYNC Mode)

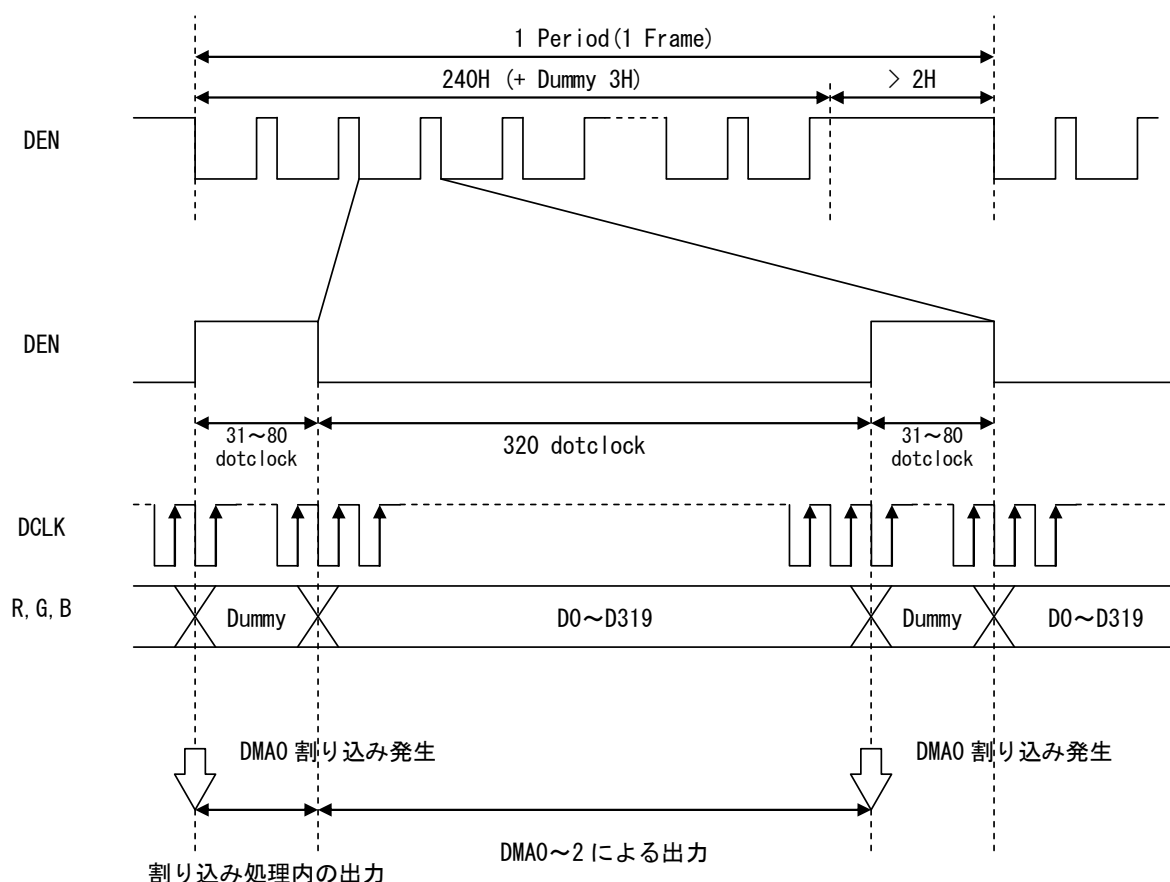


図 3 制御タイミング (DE Mode)

Vsync、Hsync、DEN のポート制御と Dummy データの送信は割り込み処理内で出力し、表示するデータ(D0 ~D319)は DMA 転送により出力します。垂直方向の 1~240 ライン以外の帰線期間中は 1 ライン目のデータをダミー送信しています。

割り込み処理内の詳細は、サンプルコードの関数 dma0 を参照ください。

SYNC Mode について

UART を使用しているため Dummy データを 8 の倍数(バイト単位)でしか出力できません。そのため今回の LCD モジュールのような tHFP=20 は簡単に出力できません。Dummy データを 16 クロックしか出力しないと、D0 ~D3 が Dummy データとして取り扱われ D4 以降が画面に表示されます。逆に 24 クロック出力すると、Dummy データの 21~24 クロックが画面の左側に表示されます。

DE Mode の場合は、DEN 端子で表示開始位置を制御し、Dummy データのクロックを 31~80 の範囲で選べるため問題ありません。(このような機能を持つ LCD モジュールに限られます。)

サンプルコード" direct_drive.c"にある定義" #define USE_DEN"をコメントアウトすると、SYNC Mode で動作します。

6. 参考プログラム

参考プログラムは、ルネサス テクノロジホームページから入手してください。

参考プログラムは、プログラマブル入出力ポート、UART、DMA の設定例と同期信号の生成部分を示します。ハードウェアの構成にあわせて変更してください。

初期化後、背景を青で塗りつぶし、白い枠を描画します。

6.1 初期化手順

初期化手順は以下のようになります。

- 初期化関数 `initial_lcd_driver()` を呼び出す。
- DMA0 の割り込みレベルを設定する(0 以外)
- 割り込み許可にする。
- 関数 `start_dma()` を呼び出す。(映像信号の出力を開始します)

映像信号の出力を停止する場合は関数 `stop_dma()` を呼び出してください。映像信号の出力を再開する場合は関数 `start_dma()` を呼び出してください。

初期化例

```
void main(void)
{
    initial_lcd_driver();

    dm0ic = 1;
    asm("fset i");           // enable interrupts
    start_dma();             // start DMA

    //(以下省略)
}
```

6.2 ドライバの構成

ドライバは以下の関数から構成されています。

関数名	処理概要
<code>initial_lcd_driver</code>	ドライバの初期化
<code>ConfigureUARTs</code>	UART の設定
<code>dma0</code>	DMA0 割り込み処理 : ドライバのメイン処理です
<code>wait_uart8</code>	UART8 の送信完了待ち
<code>uart4_tx</code>	UART4 の 1 バイト送信
<code>uart5_tx</code>	UART5 の 1 バイト送信
<code>uart8_tx</code>	UART8 の 1 バイト送信
<code>set_dma</code>	水平 1 ライン分の DMA 転送
<code>start_dma</code>	DMA 転送の開始
<code>stop_dma</code>	DMA 転送の停止

ドライバのメイン処理は関数 `dma0` で、水平 1 ラインごとに DMA 割り込み処理を行います。

DMA 割り込み処理内では、最初に DE や Vsync, Hsync のタイミングを制御し、最後に DMA 転送設定を行います。DMA 割り込み $240 + \alpha$ 回で 1 画面分のデータ転送が完了します。 $(\alpha = \text{PRE_GAP} + \text{POST_GAP})$ で、サンプルでは 24 回に設定)

DE Mode の場合のソースコード

```
void dma0(void)
{
    uint32_t buf_offset = 0;

    wait_uart8();                // UART の出力完了待ち
    LCD_EN = DEN_NAC;           // disable LCD (DEN L→H)
    uart8_tx(0x00);             // 32 クロックのダミーを出力
    uart8_tx(0x00);
    uart8_tx(0x00);
    uart8_tx(0x00);
    wait_uart8();                // UART の出力完了待ち

    switch(LCD_state){
        case 0:                  // pre gap
            if(++gap_line >= PRE_GAP){
                gap_line = 1;
                LCD_state = 1;
                vs_flag = DISPLAY_DATA;
            }
            break;
        case 1:                  // picture
            buf_offset = (uint32_t)pic_line * LCD_WIDTH_byte;
            LCD_EN = DEN_ACT;     // enable LCD (DEN H→L)
            if(++pic_line >= LCD_HIGHT){
                pic_line = 0;
                LCD_state = 2;
            }
            break;
        case 2:                  // post gap
            vs_flag = DISPLAY_DUMMY;
            if(++gap_line > POST_GAP){
                gap_line = 0;
                LCD_state = 0;
            }
            break;
        default:
            LCD_state = 0;
            gap_line = 0;
            pic_line = 0;
            break;
    }
    set_dma(buf_offset);        // 320 クロックの DMA 転送の設定
}
```


7. 参考ドキュメント

ハードウェアマニュアル

R32C/111 グループハードウェアマニュアル Rev.1.00

(最新版をルネサステクノロジホームページから入手してください。)

テクニカルニュース/テクニカルアップデート

(最新の情報をルネサステクノロジホームページから入手してください。)

ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問い合わせ先

<http://japan.renesas.com/inquiry>csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2010.1.8	—	初版発行

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。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事事務の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊社営業窓口で最新の情報をご確認いただけますとともに、弊社ホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された製品のご使用につき、特に最大定格、動作電源電圧範囲、放熱特性、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、社会的損害など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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特にマイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなきよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 本資料に関する詳細についてのお問い合わせ、その他お気付きの点等がございましたら弊社営業窓口までご照会ください。

D039444