# RE01 1500KB, 256KB Group

## CMSIS Driver R_PMIP Specifications

### Summary

This document describes the specifications of the R_PMIP driver provided in the CMSIS software package for the RE01 1500KB and 256KB group (hereinafter referred to as the PMIP driver).

### Target Device

Device: RE01 1500KB group, 256KB group

Parallel MIP LCD panel (monochrome): TN0104ANVAANN-GN00 (Kyocera)

When applying this driver to a microcontroller or a parallel MIP LCD panel other than the above, modification should be made as appropriate to match the specification of the device and careful evaluation performed.

### Contents

# 1. Overview

The R-PMIP driver allows the RE01 1500KB, 256KB group products to output image data to a parallel MIP LCD panel.

The overview of this driver is shown in the table below.

**Table 1-1 Overview of PMIP Driver**

| Overview of Driver Operation | Peripheral Modules Used | Driver Modules Mainly Used |
|---|---|---|
| Outputs image data to a parallel MIP LCD panel using the MIP LCD controller (MLCD).<br>DMAC is used for data input to MLCD. | LPM, DMAC | R_LPM, R_DMAC |

# 2. Driver Configuration

## 2.1 File Configuration

The R_PMIP driver conforms to the CMSIS HAL driver package and consists of three files: r_pmip_api.c, r_pmip_api.h, and r_pmip_cfg.h in the vendor-specific file storage directory. The functions of the files are shown in Table 2-1 and the file configuration is shown in Figure 2-1.

**Table 2-1 Functions of Files of R_PMIP Driver**

| File Nae | Description |
|---|---|
| r_pmip_api.c | Driver source file<br>It provides the detail of the driver function.<br>To use the R_PMIP driver, it is necessary to build this file. |
| r_pmip_api.h | Driver header file<br>The macro, type, and prototype declarations that can be referenced by the user are defined.<br>To use the R_PMIP driver, it is necessary to include this file. |
| r_pmip_cfg.h | Configuration definition file<br>It provides configuration definitions that can be modified by the user. |

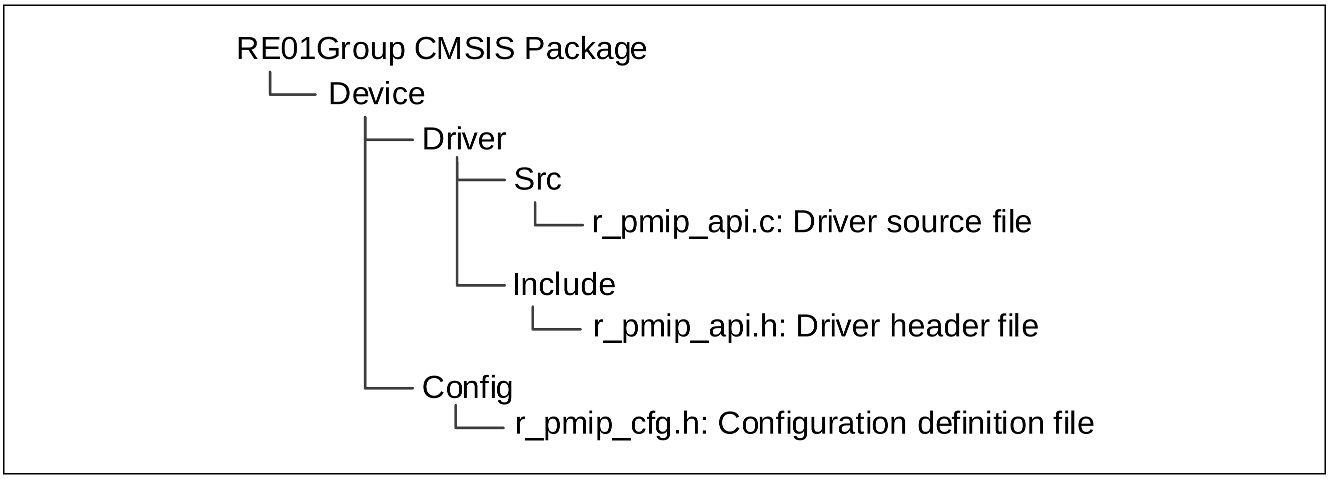

RE01Group CMSIS Package
Device
Driver
Src
r_pmip_api.c: Driver source file
Include
r_pmip_api.h: Driver header file
Config
r_pmip_cfg.h: Configuration definition file

Figure 2-1 File Configuration of PMIP Driver

## 2.2 Configurations

For the R_PMIP driver, the configuration definitions that can be modified by the user are provided in the r_pmip_cfg.h or r_pmip_api.h file. Table 2-2 and Table 2-3 show lists of configurations.

**Table 2-2 Settings of r_pmip_cfg.h**

| Name | Description | Setting Value | Initial Value |
|---|---|---|---|
| PMIP_ CFG_DISP_WIDTH | MIP LCD horizontal size (no. of pixels) | Set according to the specification of a parallel MIP LCD panel | 176 |
| PMIP_ CFG_DISP_HEIGHT | MIP LCD vertical size (no. of pixels) | Set according to the specification of a parallel MIP LCD panel | 176 |
| PMIP_CFG_SCLKH | High width of transmission clock (µs) | Set according to the specification of a parallel MIP LCD panel | 1 |
| PMIP_ CFG_VCOM_CLK | Input signal frequency for generating VCOM (Hz) | Selected from among MLCDVCOMCTL.VCOMW[1:0] bit setting values<br>0: 1 (VCOMW[1:0]=01b)<br>1: 0.5 (VCOMW[1:0]=00b)<br>2: 0.25 (VCOMW[1:0]=10b)<br>3: 0.1 (VCOMW[1:0]=11b) | 0 |
| PMIP_CFG_PARAM_CK_EN | Parameter check function | 0: Disabled<br>1: Enabled | 1 |
| PMIP_CFG_ENB_TBL | Enable signal control table select | 0: Kyocera (TN0104ANVAANN-GN00)(Note1)<br>1: Other | 0 |
| PMIP_CFG_INTERRUPT_LEVEL | MLCD interrupt priority level | 0 (high) to 3 (low) | 0 |
| PMIP_CFG_DISP_INI_DATA | MIP-LCD initialization data selection when calling PowerOn API | Set according to the specifications of the parallel MIP LCD panel<br>0: Output 0x00<br>1: Output 0xFF<br>Set "0" when your panel is Normally White and "1" when it is Normally Black | 1 |
| PMIP_CFG_FMASK_US (Note2) | VCOM mask time before data transfer | Set VCOM mask time before data transfer(Note4) | 4000 |
| PMIP_CFG_BMASK_US (Note3) | VCOM mask time after data transfer | Set VCOM mask time after data transfer(Note4) | 1000 |
| PMIP_CFG_SUPPORT_DMAC | Select whether to enable or disable DMAC | 0: DMAC disabled<br>1: DMAC enabled | 1 |
| PMIP_CFG_SUPPORT_DTC | Select whether to enable or disable DTC | 0: DTC disabled(Note5) | 0 |

Note1. When PMIP_CFG_ENB_TBL = 0 is selected, the shortest enable signal width that can be used by the Kyocera TN0104ANVAANN-GN00 is automatically set. When setting the enable signal width to an arbitrary value, or when using another parallel MIP LCD panel, PMIP_CFG_ENB_TBL should be set to 1.

Note2. Available only for 256KB groups.
The setting value is converted to the value set in the MLCDVCOMCTL.FMASK [7: 0] bits.

Note3. Available only for 256KB groups.
The setting value is converted to the value set in the MLCDVCOMCTL.BMASK [7: 0] bits.

Note4. In the actual mask time, an error of 0 to 488us (sampling cycle) occurs from the set value.
[Setting Example]
Setting value: 4000us
Mask time: 4392us (488 * 9)

Note5. DTC transfer is not supported. Be sure to set the value to 0.

**Table 2-3 Settings of r_pmip_api.h**

| Name | Description | Setting Value | Initial Value |
|---|---|---|---|
| OTHER_ENB<br>(e_pmip_enb_tbl_other_t) | $t_{oENB}$[us] and $t_{bENB}$[us](Note) | Set according to the specification of parallel MIP LCD used<br>* Set a larger value than that of PMIP_CFG_SCLKH. | 4 |
| OTHER_ENBH<br>(e_pmip_enb_tbl_other_t) | $t_{wENBH}$[us](Note) | Set according to the specification of parallel MIP LCD used | 20 |

Note: Valid only when the "Other" table is selected (PMIP_CFG_ENB_TBL = 1). For details including $t_{oENB}$, refer to section 58.3.12, MCLD Timing, in the User's Manual: Hardware.

## 2.3 Macro and Type Definitions

For the R_PMIP driver, the configuration definitions that can be modified by the user are provided in the r_pmip_api.h file.

### 2.3.1 Type Definitions

| Definition | Value | Description |
|---|---|---|
| e_trans_mode_t | DMAC_TR | DMAC transfer |
| | CPU_TR | CPU transfer |
| e_trans_cmd_t | CMD0 | Bit array for data transmission |
| | CMD1 | Horizontal address auto update method |
| | CMD2 | Vertical address auto update method |

### 2.3.2 PMIP Error Code Definitions

These define the PMIP error codes.

**Table 2-4 List of PMIP Error Code Definitions**

| Definition | Description | Solution |
|---|---|---|
| PMIP_OK | Normal completion | - |
| PMIP_ERROR | MLCD module clock start error | LPM driver error occurs. See the R_LPM_ModuleStart function of LPM driver. |
| | MLCD module clock stop error | LPM driver error occurs. See the R_LPM_ModuleStop function of LPM driver. |
| | MLCD system initial processing error | 1. Execute the R_PMIP_Open function before transmission.<br>2. LPM driver error occurs. Refer to the LPM driver specification document. |
| PMIP_ERROR_TRANS_MODE | Transfer method setting error | Set either DMAC_TR or CPU_TR for the argument tr_mode.mode of R_PMIP_Open. |
| PMIP_ERROR_DMAC_TRANS | Error occurrence on selected DMAC channel | DMAC driver error occurs. Refer to the DMAC driver specification document. |
| PMIP_ERROR_DMAC_CFG | DMAC channel setting error | Set one of 0 to 3 values for the argument tr_mode.sel of R_PMIP_Open. |
| | Selected DMAC channel in use | The selected DMAC channel is in use. Set an unused DMAC channel for the argument tr_mode.sel of R_PMIP_Open. |
| PMIP_ERROR_INPUT _CLK_OFFRANGE | Transmit clock high width setting error | Set so as to meet the condition:<br>PMIP_CFG_SCLKH(us)* System clock (Hz) / 1000000 <256 |
| PMIP_ERROR_ENBEG_CFG | ENBG signal width setting error | 1. Set so as to meet the condition:<br>KYOCERA_ENB > PMIP_CFG_SCLKH or OTHER_ENB > PMIP_CFG_SCLKH<br>2. The calculated ENBG width overflows. |
| PMIP_ERROR_CONTROL _CMD | Command setting error | Set one of CMD0, CMD1 and CMD2 for the argument cmd of R_PMIP_Control. |
| | Command setting value error | Set either 0 or 1 for the argument set of R_PMIP_Control |
| PMIP_ERROR_SEND_CFG | Transmission parameter setting error | Set so as to meet the conditions:<br>st_h: 0 to (PMIP_ CFG_DISP_WIDTH-1);<br>however, a multiple of 8<br>st_v: 0 to (PMIP_ CFG_DISP_HEIGHT-1)<br>size_h: 1 to (PMIP_ CFG_DISP_WIDTH/8)<br>size_v: 1 to PMIP_ CFG_DISP_HEIGHT<br>img_size_h:<br>2 to (PMIP_ CFG_DISP_WIDTH/8) |
| PMIP_ERROR_SYSTEM _SETTING | Interrupt setting error | Set an interrupt priority level or an MLCD interrupt. Interrupt setting means setting of r_system_cfg.h file in the Config folder. |

### 2.3.3 PMIP Event Definitions

These define the PMIP events.

| Definition | Value | Description |
|---|---|---|
| PMIP_EVENT_SEND_COMPLETE | 1 | Data output completed |
| PMIP_EVENT_ERROR_DMAC_TRANS | 2 | Data output failed |

## 2.4 Structure Definitions

For the PMIP driver, the structures that can be modified by the user are defined in the r_pmip_api.h file.

### 2.4.1 st_transmode_t Structure

| Element Name | Type | Description |
|---|---|---|
| mode | e_trans_mode_t | Transfer mode<br>* DTC transfer is not supported. |
| sel | uint8_t | DMAC channels<br>0: DMAC0<br>1: DMAC1<br>2: DMAC2<br>3: DMAC3<br>* Invalid if CPU transfer is selected (mode=CPU_TR) |

### 2.4.2 st_mlcd_mode_info_t Structure

| Element Name | Type | Description |
|---|---|---|
| cb_event | pmip_cb_event_t | Callback function |
| flags | uint8_t | Flag used for DMA transfer<br>* Should not be changed |

## 3. State Transitions

The state transition diagram of the PMIP driver is shown in Figure 3-1.

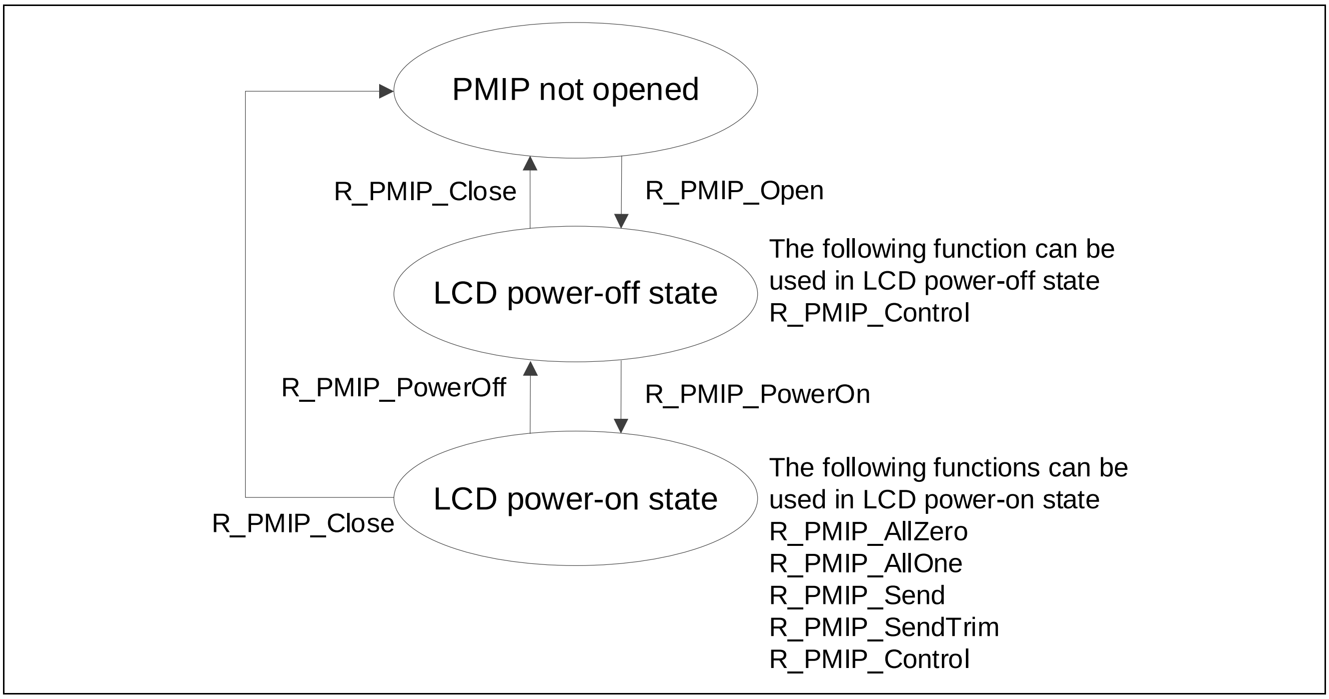


Figure 3-1 State Transitions (Note 1)

Note The R_PMIP_GetVersion function can be called from any state.

# 4. Driver Functions

## 4.1 Function Specifications

### 4.1.1 R_PMIP_Open

| | | |
|---|---|---|
| Overview | System initial processing | |
| Format | e_pmip_err_t R_PMIP_Open(pmip_cb_event_t cb, st_transmode_t* tr_mode) | |
| Description | The MLCD module stop bit is cancelled, a burn-in prevention signal (VCOM), memory rewrite activating signal (ENB), transmission clock high width (μs), transfer, and callback function are set.<br>In setting the transmission clock high width, the current system clock frequency is acquired within this function, and calculations and settings are made such that the clock frequency high width matches PMIP_CFG_SCLKH. | |
| Argument | pmip_cb_event_t cb | Callback function |
| | tr_mode.mode | Transmission method selection |
| | tr_mode.sel | DMAC channel selection |
| Return Value | PMIP_OK<br>PMIP_ERROR<br>PMIP_ERROR_TRANS_MODE<br>PMIP_ERROR_DMAC_CFG<br>PMIP_ERROR_INPUT_CLK_OFFRANGE<br>PMIP_ERROR_ENBEG_CFG<br>PMIP_ERROR_SYSTEM_SETTING | |
| Remarks | ･**Operate the clock correction circuit (CCC)** before calling the R_PMIP_Open function (see chapter 5 for how to operate it).<br>･$t_{oENB}$ and $t_{bENB}$ are calculated by the following equation.<br>$t_{oENB}$ , $t_{bENB}$ = (SCLK high width) + (ENBEG[7:0]*PCLKA 1 cycle) | |

### 4.1.2 R_PMIP_PowerOn

| | |
|---|---|
| Overview | MIP power-on sequence |
| Format | e_pmip_err_t R_PMIP_PowerOn(void) |
| Description | Performs power-on sequence of the parallel MIP-LCD.<br>When PMIP_CFG_DISP_INI_DATA is 1, all bits of the parallel MIP-LCD are initialized to “1”.<br>When PMIP_CFG_DISP_INI_DATA is 0, all bits of the parallel MIP-LCD are initialized to “0”. |
| Argument | None |
| Return Value | PMIP_OK<br>PMIP_ERROR |
| Remarks | |

### 4.1.3 R_PMIP_Control

Overview: Transmission setting

Format: e_pmip_err_t R_PMIP_Control(e_trans_cmd_t cmd, void* set)

Description: Set any one among the following settings

- Bit array for data transmission (MLCDCR.BITSW)
- Horizontal address auto-update method (MLCDCR.HADDRDEC)
- Vertical address auto-update method (MLCDCR.VADDRDEC)

By modifying these settings, horizontal and vertical inverted display of image data is possible.

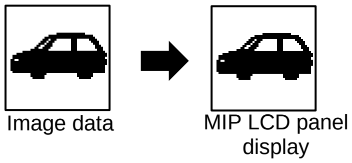
| Argument | | |
|---|---|---|
| | e_trans_cmd_t cmd | Selects a setting item and value. |
| | void* set | ◆ When cmd = CMD0 (uint8_t)set<br>0: MSB transfer<br>1: LSB transfer<br>◆ When cmd = CMD1 (uint8_t)set<br>0: Address incremented<br>1: Address decremented<br>◆ When cmd = CMD2 (uint8_t)set<br>0: Address incremented<br>1: Address decremented |

Return Value: PMIP_OK  
PMIP_ERROR  
PMIP_ERROR_CONTROL_CMD

Remarks: Examples of calling function

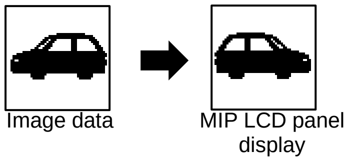
◆ Normal display

```
R_PMIP_Control(CMD0,0); /* default */
R_PMIP_Control(CMD1,0); /* default */
R_PMIP_Control(CMD2,0); /* default */
R_PMIP_Send(0,0,32,256,img);
```



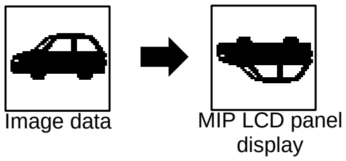
◆ Horizontally inverted display

```
R_PMIP_Control(CMD0,1);
R_PMIP_Control(CMD1,1);
R_PMIP_Send(248,0,32,256,img);
```



◆ Vertically inverted display

```
R_PMIP_Control(CMD2,1);
R_PMIP_Send(0,255,32,256,img);
```



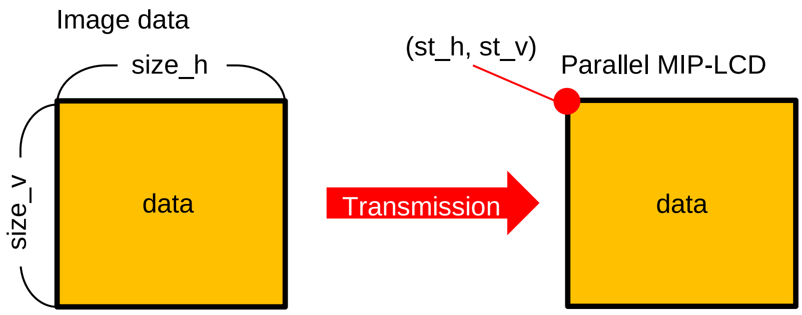
### 4.1.4 R_PMIP_Send

Overview Data transmission

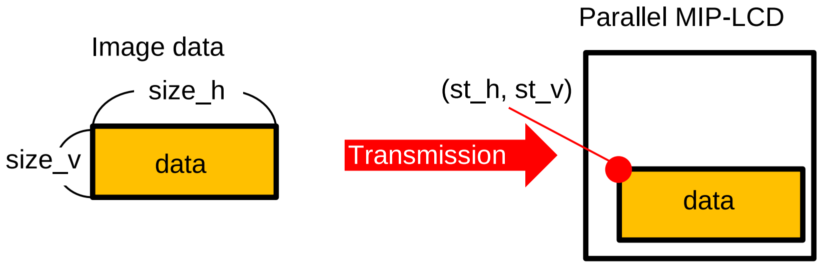
Format e_pmip_err_t R_PMIP_Send(uint8_t st_h, uint8_t st_v, uint8_t size_h, uint16_t size_v, uint8_t* &data)

Description Outputs the image data (size_h bits × size_v bits) to the coordinates of the parallel MIP-LCD (st_h, st_v).

1) When updating LCD full-screen data



2) When updating LCD partial screen data



Argument

| | |
|---|---|
| uint8_t st_h | Transmit start pixel (horizontal) (SL) |
| uint8_t st_v | Transmit start pixel (vertical) (GL) |
| uint8_t size_h | Transmit data size (no. of horizontal bytes) |
| uint16_t size_v | Transmit data size (no. of vertical rows) |
| uint8_t* &data | Memory address of transmit data |

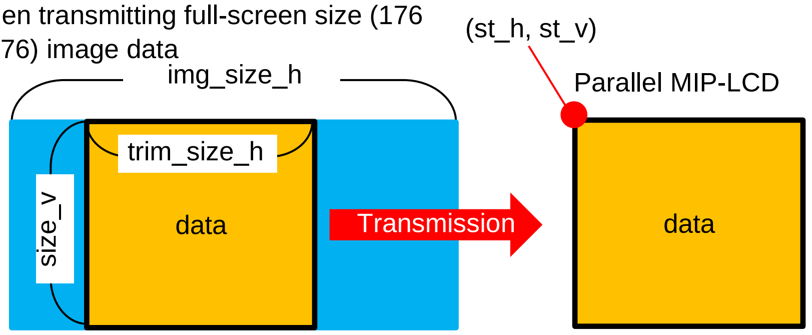
Return Value PMIP_OK
PMIP_ERROR
PMIP_ERROR_DMAC_TRANS
PMIP_ERROR_SEND_CFG
PMIP_ERROR_SYSTEM_SETTING

Remarks A multiple of 8 should be set for transmit start pixel (SL).

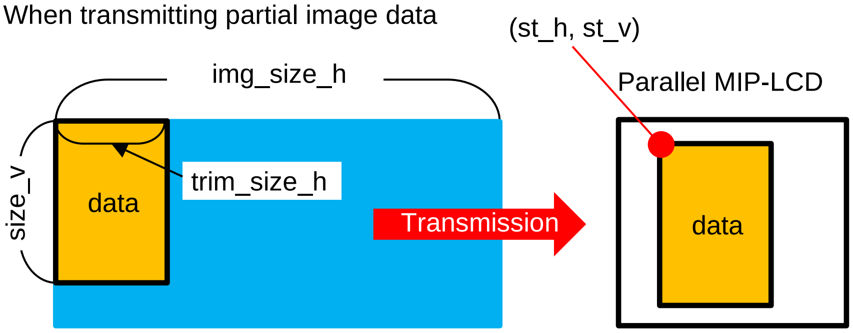
### 4.1.5 R_PMIP_SendTrim

|  |  |  |
|---|---|---|
| Overview | Data transmission | |
| Format | e_pmip_err_t R_PMIP_SendTrim(uint8_t st_h, uint8_t st_v, uint16_t img_size_h, uint8_t trim_size_h, uint16_t size_v, uint8_t* &data) | |
| Description | Outputs image data (trim_size_h bits ×size_v bits) to coordinates (st_h, st_v) of a parallel MIP-LCD. The horizontal size of the image data is img_size_h, and a part with a clipping size trim_size_h is clipped and output. | |

1) When transmitting full-screen size (176 × 176) image data



2) When transmitting partial image data



|  |  |  |
|---|---|---|
| Argument | uint8_t st_h | Transmit start pixel (horizontal) (SL) |
| | uint8_t st_v | Transmit start pixel (vertical) (GL) |
| | uint16_t img_size_h | Transmit data size (no. of horizontal bytes) |
| | uint8_t trim_size_h | Transmit data clipping size (no. of horizontal bytes) |
| | uint16_t size_v | Transmit data size (no. of vertical rows) |
| | uint8_t* &data | Memory address of transmit data |
| Return Value | PMIP_OK<br>PMIP_ERROR<br>PMIP_ERROR_DMAC_TRANS<br>PMIP_ERROR_SEND_CFG<br>PMIP_ERROR_SYSTEM_SETTING | |
| Remarks | A multiple of 8 should be set for transmit start pixel (SL). | |

### 4.1.6 R_PMIP_AllOne

|  |  |
|---|---|
| Overview | Transmission of data 1s for the entire screen |
| Format | e_pmip_err_t R_PMIP_AllOne(void) |
| Description | Outputs 1s for all bits of the parallel MIP-LCD |
| Argument | None |
| Return Value | PMIP_OK<br>PMIP_ERROR |
| Remarks | |

### 4.1.7 R_PMIP_AllZero

| | |
|---|---|
| Overview | Transmission of data 0s for the entire screen |
| Format | e_pmip_err_t R_PMIP_AllZero(void) |
| Description | Outputs 0s for all bits of the parallel MIP-LCD |
| Argument | None |
| Return Value | PMIP_OK<br>PMIP_ERROR |
| Remarks | |

### 4.1.8 R_PMIP_PowerOff

| | |
|---|---|
| Overview | MIP power-off sequence |
| Format | e_pmip_err_t R_PMIP_PowerOff(void) |
| Description | Performs power-off sequence of the parallel MIP-LCD. |
| Argument | None |
| Return Value | PMIP_OK<br>PMIP_ERROR |
| Remarks | |

### 4.1.9 R_PMIP_Close

| | |
|---|---|
| Overview | System termination processing |
| Format | e_pmip_err_t R_PMIP_Close(void) |
| Description | Initializes the value specified in R_SMIP_Open and sets the MLCD module stop bit to a stop state for system termination processing. |
| Argument | None |
| Return Value | PMIP_OK<br>PMIP_ERROR |
| Remarks | CCC should not be stopped. If CCC needs to be stopped, set with the user program. |

### 4.1.10 R_PMIP_GetVersion

| | |
|---|---|
| Overview | API version acquisition |
| Format | uint32_t R_PMIP_GetVersion(void) |
| Description | Acquires the API version |
| Argument | None |
| Return Value | PMIP driver version |
| Remarks | |

## 4.2 Setting Interrupts

**The interrupts that are used for communication control must be registered to NVIC using the r_system_cfg.h file.** For details on interrupts (NVIC), refer to the "Setting Interrupts (NVIC)" in the RE01 1500KB, 256KB Group Getting Started Guide to Development Using CMSIS Package (r01an4660). The interrupt definitions to beused by the PMIP driver are shown in Table 4-1, and an example of interrupt registration is shown in Figure 4-1.

**Table 4-1 NVIC Registration Definitions**

| NVIC Registration Definition |
|---|
| SYSTEM_CFG_EVENT_NUMBER_MLCD_TEI |
| SYSTEM_CFG_EVENT_NUMBER_MLCD_TEMI |

```
#define SYSTEM_CFG_EVENT_NUMBER_SCI5_TXI
                    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)   /*!< Numbers 1/9/17/25 only */
#define SYSTEM_CFG_EVENT_NUMBER_MLCD_TEI
                    (SYSTEM_IRQ_EVENT_NUMBER9)   /*!< Numbers 1/9/17/25 only */
#define SYSTEM_CFG_EVENT_NUMBER_PORT_IRQ5
                    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)   /*!< Numbers 5/13/21/29 only */
...
#define SYSTEM_CFG_EVENT_NUMBER_SPI1_SPII
                    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)   /*!< Numbers 2/10/18/26 only */
#define SYSTEM_CFG_EVENT_NUMBER_MLCD_TEMI
                    (SYSTEM_IRQ_EVENT_NUMBER10)   /*!< Numbers 2/10/18/26 only */
#define SYSTEM_CFG_EVENT_NUMBER_PORT_IRQ9
                    (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)   /*!< Numbers 2/10/18/26 only */
...
```

Figure 4-1 Example of Interrupt Registration to NVIC in r_system_cfg.h

## 5. Driver Use Examples

Figure 5-1 shows a setting example when outputting an image using the PMIP driver.

```
#include "r_pmip_api.h"
#include "r_system_api.h"
#include "r_lpm_api.h"

static void callback(uint32_t event);
void set_ccc(void);
extern const uint8_t image[3872]; /* 176 x 176 bit monochrome image data */
main()
{
    st_transmode_t tmd;
    tmd.mode = DMAC_TR;
    tmd.sel   = 0; // DMAC channel select

    set_ccc();

    (void)R_PMIP_Open((pmip_cb_event_t)&callback, &tmd); // Initialization of system
    (void)R_PMIP_PowerOn(); // MIP power ON sequence
    (void)R_PMIP_Control(CMD0, (void*)1);

    While(1)
    {
        R_SYS_SoftwareDelay(1000, SYSTEM_DELAY_UNITS_MILLISECONDS);
        (void)R_PMIP_Send(0,0,22,176,image);
        R_SYS_SoftwareDelay(1000, SYSTEM_DELAY_UNITS_MILLISECONDS);
        (void)R_PMIP_AllZero();
    }
}

static void callback(uint32_t event)
{
    /* Describe the process when all transfers are completed */
    switch(event)
    {
        case PMIP_EVENT_SEND_COMPLETE:
        {
            /* Describe the process when transmission is normally completed */
        }
        break;
        case PMIP_EVENT_ERROR_DMAC_TRANS:
        {
            /* Describe the process when transmission error occurs */
        }
        break;
    }
}

void set_ccc(void)
{
  /* CCC needs to be set for VCOM operation of PMIP(MLCD) driver */
  R_LPM_ModuleStart(LPM_MSTP_CCC); // release of CCC module stop
  R_SYS_RegisterProtectEnable(SYSTEM_REG_PROTECT_CGC);
  R_SYS_SubOscSpeedClockStart();
  R_SYS_SoftwareDelay(1000, SYSTEM_DELAY_UNITS_MILLISECONDS); // wait 1s
  CCC->R128CTRL_b.CADJUSCEN = 1; /* Start CCC */
}
```

Figure 5-1 Example of PMIP Driver Setting and Transmission

# 6. Usage Notes

## 6.1 SCLK High Width Setting

The setting of the MLCDCR.SCKCR bit is calculated from PMIP_CFG_SCLKH and PCLKA. When the MLCDCR.SCKCR bit is 1, the width of the high signal actually output from the MLCD_SCLK pin has a NOP time added, to become two cycles (PCLKA equivalent). Care must be exercised since this does not match the high width set by PMIP_CFG_SCLKH. For details, refer to section 58.3.12, MLCD Timing, in the User’s Manual: Hardware.

## 6.2 Restrictions on Function Execution

The functions used to output image data by using this driver (R_PMIP_Send, R_PMIP_SendTrim, R_PMIP_AllOne, and R_PMIP_AllZero) can be used only after the R_PMIP_PowerOn function is executed. The R_PMIP_Control function can be used only after the R_PMIP_Open function is executed.

# 7. Troubleshooting

## 7.1 No Output from MLCD_VCOM

A) Check to make sure the CCC clock has been set up correctly.

The CCC clock output should be set before executing the R_PMIP_Open function.

For details, see chapter 5, Driver Use Examples.

## 7.2 Occurrence of Hard Fault Error when API of CMSIS Driver Is Called

A) The API has not possibly been copied to RAM.

Before calling an API function that is mapped to RAM, make sure that it has been copied to RAM by the R_SYS_CodeCopy function. For details, refer to the related document No. R01AN4660.

## 7.3 Peripheral Function Fails to Operate when API Is Called

A) Check the API's return value to see if an error has occurred. In particular, errors are often caused by problems related to interrupts not being set in r_system_cfg.h. For details, refer to the related document No. R01AN4660.

## 7.4 Normal API Return Value But Peripheral Function Pin Fails to Work

A) Check to make sure the pins have been set up correctly by the functions in pin.c.

For details, refer to the related document No. R01AN4660.

# 8. Reference Documents

User's Manual: Hardware

RE01 1500KB Group User's Manual: Hardware R01UH0796
RE01 256KB Group User's Manual: Hardware R01UH0894
(The latest version can be downloaded from the Renesas Electronics website.)

RE01 Group CMSIS Package Startup Guide

RE01 1500KB, 256KB Group Startup Guide to Development Using CMSIS Package R01AN4660

(The latest version can be downloaded from the Renesas Electronics website.)

RE01 1500KB, 256KB Group CMSIS Driver R_LPM Specifications (R01AN4838)

(The latest version can be downloaded from the Renesas Electronics website.)

RE01 1500KB, 256KB Group CMSIS Driver R_DMAC Specifications (R01AN4730)

(The latest version can be downloaded from the Renesas Electronics website.)

Technical Update/Technical News

(The latest version can be downloaded from the Renesas Electronics website.)

User's Manual: Development Tools

(The latest version can be downloaded from the Renesas Electronics website.)

## Revision History

| Rev. | Date | Description | |
|---|---|---|---|
| | | Summary | Summary |
| 1.01 | Dec.2.2019 | —<br>Program | First edition issued<br>Fixed the RAM / ROM placement in r_pmip_cfg.h.<br>Fixed the problem that the following internal functions are not placed in RAM even if they are set in RAM.<br>- e_mlcd_cpu_normal_trans function |
| 1.03 | Feb.25.2020 | 4<br>8<br>Program | Added the following configuration definitions<br>• PMIP_CFG_DISP_INI_DATA<br>Added description of R_PMIP_PowerOn function<br>• Addition of definition (PMIP_CFG_DISP_INI_DATA) for selecting MIP-LCD initialization data when calling PowerOn API<br>• Fixed incorrect RAM / ROM allocation of R_PMIP_Send function.<br>• Fixed inadequate RAM / ROM allocation of internal variables (resources)<br>- Changed the VCOM wait time in the R_PMIP_PowerOff function to 2ms |
| 1.04 | Mar.5.2020 | —<br>4<br>Program (256KB) | Compatible with 256KB group<br>Added the following configuration definitions<br>• PMIP_CFG_FMASK_US<br>• PMIP_CFG_BMASK_US<br>• Modified to match 256KB IO definition<br>• VCOM mask control changed from software control to hardware control<br>• Added configuration definition for VCOM mask time setting<br>- PMIP_CFG_FMASK_US<br>- PMIP_CFG_BMASK_US |
| 1.05 | Apr.23.2020 | 4<br>Program | Added the following configuration definitions<br>• PMIP_CFG_SUPPORT_DMAC<br>• PMIP_CFG_SUPPORT_DTC<br>Changed the configuration so that it can be built without DMAC and DTC drivers. |

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.