MXIC vEE Middleware

Hardware and Software Requirement

1. Renesas FSP v4.0.0 or later
2. Renesas EK-RA6M3/EK-RA6M4 Board
3. MXIC Nor Flash and SPI Nand Flash

Introduction

1. MXIC vEE(virtual EEPROM Emulation) propose an algorithm and system architecture for implementing the EEPROM emulation on a Read-While-Write flash memory. It support to run read and write/erase operations simultaneously in multi-threaded and real-time environment. It gains about 20% increase on overall read/write performance.
2. MXIC vEE can be built upon both QSPI and OSPI driver. And if you choose QSPI, you can use SPI Nand Flash.

Sample Code

```
#define NUM 30
void blinky_thread_entry (void * pvParameters)
{
    FSP_PARAMETER_NOT_USED(pvParameters);
    int ret = 0;
    module_init();
    ret = RM_VEE_Open(g_mxic_qspi_vee0.p_ctrl, g_mxic_qspi_vee0.p_cfg);
    if(ret)
        SEGGER_RTT_printf(0,"open error\n");

    ret = RM_VEE_Format(g_mxic_qspi_vee0.p_ctrl);
    if(ret)
        SEGGER_RTT_printf(0,"format error\n");

    ret = RM_VEE_Init(g_mxic_qspi_vee0.p_ctrl);
    if(ret)
        SEGGER_RTT_printf(0,"init error\n");

    uint8_t randdata[NUM] = {0};
    uint32_t randaddr[NUM] = {0};
    uint8_t cmpdata[NUM] = {0};

    for(int i = 0; i < NUM; i++)
    {
        randdata[i] = (uint8_t)(rand()%255);
        randaddr[i] = (uint32_t)(rand()%(MX_EEPROM_TOTAL_SIZE-1));
        ret = RM_VEE_Write(g_mxic_qspi_vee0.p_ctrl, randaddr[i], 1, &
(randdata[i]));
        if(ret)
            SEGGER_RTT_printf(0,"Write error\n");
    }
    ret = RM_VEE_Flush(g_mxic_qspi_vee0.p_ctrl);

    for(int i = 0; i < NUM; i++)
```

```
    {
        ret = RM_VEE_Read(g_mxic_qspi_vee0.p_ctrl, randaddr[i], 1, &
(cmpdata[i]));
        if(ret)
            SEGGER_RTT_printf(0,"Read error\n");
    }

    if(!(memcmp(randdata, cmpdata, NUM)))
        SEGGER_RTT_printf(0,"Data OK\n");
    else
        SEGGER_RTT_printf(0,"Data Error\n");
    while (1)
    {
        vTaskDelay(configTICK_RATE_HZ);
    }
}
```