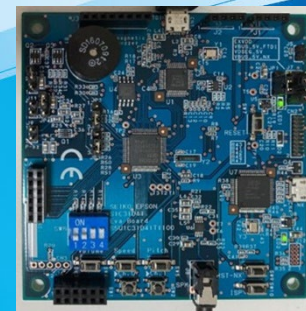


Sound MCU: S1C31D41

Demonstration Software Guide

"4. MEMCHECK (Memory Checking Example)"



Rev1.00

- Purpose : This document describes “MEMCHECK” software.

1. Evaluation Board
 - 1.1. Jumper Settings
 - 1.2. PC Connection
2. Demonstration Software
 - 2.1. Build and Download
 - 2.2. Operation Flow
3. HWP Library Specification(SOUNDPALY function)
 - 3.1. seMemCheckInit
 - 3.2. seMemCheckSetParameter
 - 3.3. seMemCheckRunCommand
 - 3.4. seMemCheckGetState
 - 3.5. seMemCheckFinish

1. Evaluation Board

1.1. Jumper Settings

Set the jumpers on the evaluation board according to each mode.

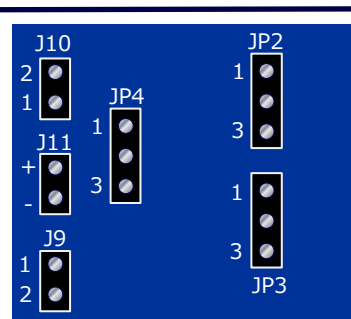


Table1.1 Output Mode Setting(*1)

Output Mode	JP2	JP3	JP4	J10	J8	J9	J11
Electromagnetic Buzzer (factory setting)	2-3 short	2-3 short	1-2 open	short	-	-	-

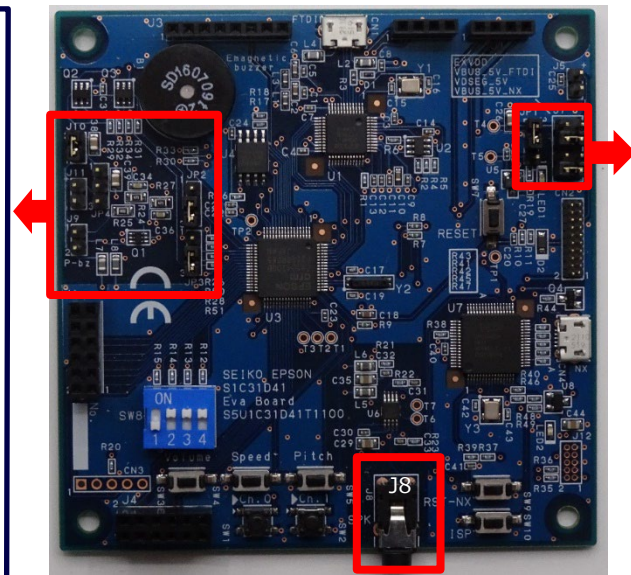


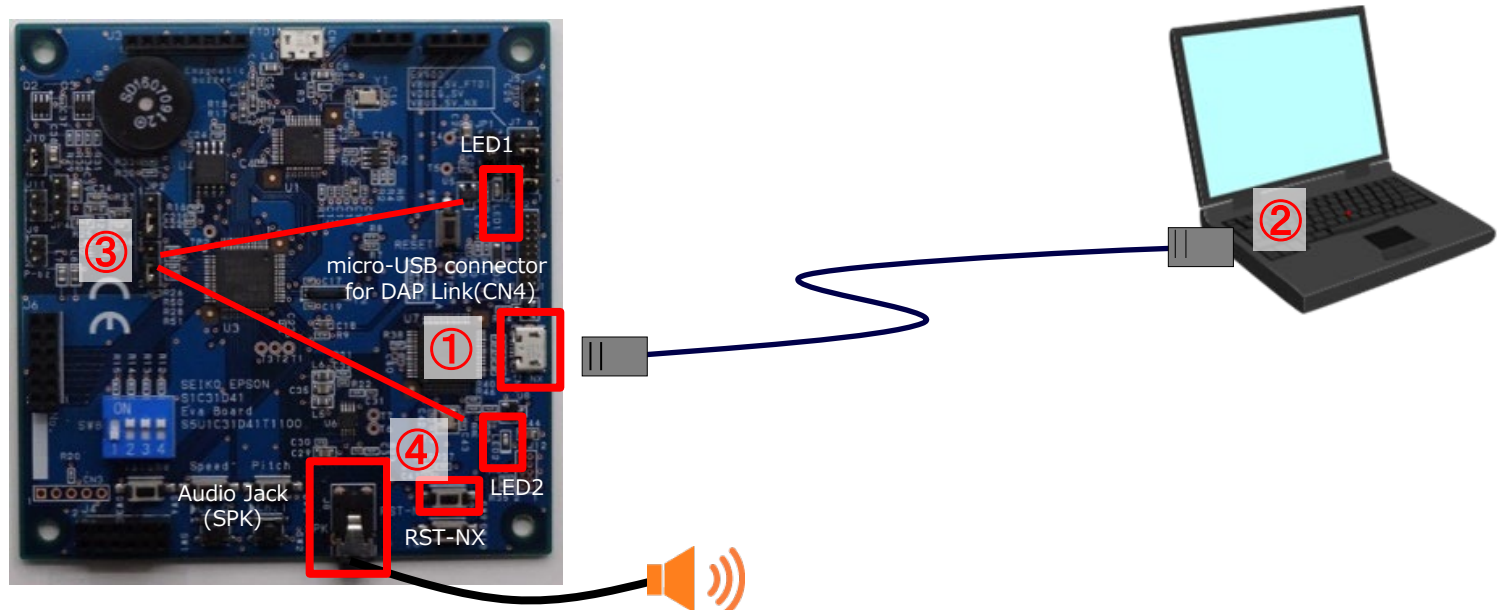
Table 1.2 Power Supply Setting			
Power	J7	JP1	
USB 5V supply	1-2 short 3-4 open 5-6 short 7-8 open	2-3 short	

*1: The output mode jumper settings do not affect the software, but just in case, use the factory settings from the evaluation board.

1.2. PC Connection

Follow the steps below to connect the PC to the evaluation board.

- ① Connect the micro USB cable to “CN4” connector on the evaluation board.
- ② Connect the micro USB cable to the USB port of the PC with IDE installed.
- ③ Make sure LED1 and LED2 on the evaluation board are lit.
- ④ Push “RST-NX” button.



2. Demonstration Software

2.1. Build and Download

Build **MEMCHECK** project in S1C31D41 peripheral sample software package and download the software to the evaluation board.

1. Double click the MEMCHECK workspace file to launch the IDE.
2. Build this project and download the built software to S1C31D41 on the evaluation board.

* For details on software build and download, refer to the following documents.

- S1C31 Family Peripheral Circuit Sample Software Manual

```
[s1c31d5xd41sp_verx_xx]
- [Licenses] : License group
- [Drivers] : Driver group
  - [board] : Drivers related to the evaluation board
    - [S5U1C31D41T1]
      - [ARM]
      - [IAR]
      - board.c/.h : Evaluation board setting program
      - settings.h : Definition file for setting the function of the evaluation board
      - ...
    - [CMSIS] : CMSIS Drivers
      - [Device]
        - [S1C31D41]
          - [Include]
            - S1C31D41.h : CMSIS peripheral circuit access layer header file
            - ...
          - [Source]
            - [ARM]
            - [IAR]
            - startup_S1C31D41.s : CMSIS startup program
            - system_S1C31D41.c : CMSIS peripheral circuit access layer program
          - [Driver]
          - [SVD]
        - [sePeripheralLibrary] : Peripheral circuit library
          - se_clg.c/.h
          - se_i2c.c/.h
          - ...
      - [Middleware] : Middleware group
        - [seHWProcLibrary] : HW Processor library
        - [seEepromLibrary] : EEPROM emulation library
        - [seFlashLibrary] : Self-programming library
      - [Projects] : Sample software group
        - [Demonstration] : Various demonstration software
          - [SOUNDPLAY] : Memory Checking Example Project
            - [ARM] : MDK-ARM Project
              - memcheck_D41.uvmpw : Workspace file
            - [IAR] : IAR EWARM Project
              - MEMCHECK_D41.eww : Workspace file
            - main.c
            - ...
          - [Applications] : Application software group
          - [Examples] : Peripheral example software group
        - README.txt
```

2.2. Operation Flow

MEMCHECK uses the memory check function provided by the HW processor to perform self-diagnosis of the internal RAM, internal Flash, and external QSPI-Flash.

1. Set the system clock for MEMCHECK. (See InitSCLK function in main.c.)
2. Run the RAM check(R/W, March-C) and the Flash check(checksum, CRC-16) for internal RAM. (See runTestMemCheckRAM and runTestMemCheckROM in test_memcheck.c.)
 - 2.1. Call the seMemCheckInit() function to initialize the MEMCHECK.
 - 2.2. Call seMemCheckSetParameter() function to set parameters for MEMCHECK.
 - 2.3. Call seMemCheckRunCommand() function to start the memory check.
 - 2.4. Wait for an interrupt indicating the memory check is complete.
3. Initialize the QSPI to access an external QSPI Flash.
4. Perform the Flash check(checksum, CRC-16) for external QSPI Flash. (See runTestMemCheckROM in test_memcheck.c.)
 - 4.1. Call the seMemCheckInit() function to initialize the MEMCHECK.
 - 4.2. Call seMemCheckSetParameter() function to set parameters for MEMCHECK.
 - 4.3. Call seMemCheckRunCommand() function to start the memory check.
 - 4.4. Wait for an interrupt indicating the memory check is complete.

出力例

```
Terminal I/O
Output: Log file: Off
CLG Initialization ok
B: 0x0001, 0x0001
M: 0x0002, 0x0201
E: 0x0001, 0x0201, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0001, 0x0301
E: 0x0001, 0x0301, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0003, 0x0101
E: 0x0001, 0x0201, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0001, 0x0301
E: 0x0001, 0x0301, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0004, 0x0201
E: 0x0001, 0x0201, RESULT=0x00000485
B: 0x0001, 0x0001
M: 0x0004, 0x0201
E: 0x0001, 0x0201, RESULT=0x0000090a
B: 0x0001, 0x0001
M: 0x0001, 0x0301
E: 0x0001, 0x0301, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0005, 0x0101
E: 0x0001, 0x0201, RESULT=0x0000d96f
B: 0x0001, 0x0001
M: 0x0005, 0x0101
E: 0x0001, 0x0201, RESULT=0x00003906
B: 0x0001, 0x0001
M: 0x0001, 0x0301
E: 0x0001, 0x0301, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0004, 0x0201
E: 0x0001, 0x0201, RESULT=0x000007d5
B: 0x0001, 0x0001
M: 0x0004, 0x0201
E: 0x0001, 0x0201, RESULT=0x00000faa
B: 0x0001, 0x0001
M: 0x0001, 0x0301
E: 0x0001, 0x0301, RESULT=0x00000000
B: 0x0001, 0x0001
M: 0x0005, 0x0101
E: 0x0001, 0x0201, RESULT=0x00006efc
B: 0x0001, 0x0001
M: 0x0005, 0x0101
E: 0x0001, 0x0201, RESULT=0x0000c687
B: 0x0001, 0x0001
M: 0x0001, 0x0301
E: 0x0001, 0x0301, RESULT=0x00000000
```


3. HWP Library Specification (MEMCHECK function)

3.1. seMemCheckInit

Syntax	<code>void seMemCheckInit (void)</code>
Arguments	-
Return value	-
Explanation	This function initializes MEMCHECK function. Call this function first when using MEMCHECK function.

3.2. seMemCheckSetParameter

Syntax	<pre>void seMemCheckSetParameter (unsigned long memory_start_address, unsigned long memory_size, unsigned short initial_value)</pre>
Arguments	<p>memory_start_address Start address of the memory area to be checked.</p> <ul style="list-style-type: none">- In case of RAM: 0x15 0000, ..., 0x15 1FFF 0x15 3000, ..., 0x15 67FF- In case of internal Flash: 0x00 0000, ..., 0x02 FFFF- In case of external QSPI-Flash: 0x00 0000 + OFFSET, ..., 0x0F FFFF + OFFSET * The "OFFSET" is 0x04 0000, the start address of the memory mapped access area for external QSPI-Flash. <p>memory_size Size of the memory area to be checked (in bytes).</p> <p>initial_value Initial value used for calculation of checksum or CRC16. 0x0000, ..., 0xFFFF *Normally set to 0x0000.</p>
Return value	-
Explanation	This function sets the parameters for MEMCHECK.

EPSON
EXCEED YOUR VISION

12

3.4. seMemCheckGetState

Syntax	unsigned short seMemCheckGetState (void)
Arguments	-
Return value	Current state of MEMCHECK. 0x0000 (mc_state_init): On initializing 0x0001 (mc_state_idle): Idle 0x0002 (mc_state_ram_rw): Running RAM Check by Read/Write 0x0003 (mc_state_ram_march_c): Running RAM Check by March-C 0x0004 (mc_state_checksum): Running Checksum 0x0005 (mc_state_crc): Running CRC
Explanation	This function gets the current status of MEMCHECK function.

3.5. seMemCheckFinish

Syntax	void seMemCheckFinish (void)
Input argument	-
Return value	-
Explanation	This function finishes MEMCHECK function.

EPSON
EXCEED YOUR VISION