B&R SRAM Driver (Windows)

User's Manual

Date:	November 4, 2015
Project Number:	AT-30-095076

We reserve the right to change the content of this manual without prior notice. The information contained herein is believed to be accurate as of the date of publication, however, B&R makes no warranty, expressed or implied, with regards to the products or the documentation contained within this document. B&R shall not be liable in the event if incidental or consequential damages in connection with or arising from the furnishing, performance or use of these products. The software names, hardware names and trademarks used in this document are registered by the respective companies.

I Versions

Version	Date	Comment	Edited by
1.00	Mar 05, 2014	First Edition	PAM
1.10	Nov 04, 2015	Windows 8.1 and Windows 10 were added as supported operating systems.	PAM

Table 1: Versions

II Safety Notices

Safety notices in this document are organized as follows:

Safety notice	Description
Danger!	Disregarding the safety regulations and guidelines can be life-threatening.
Warning!	Disregarding the safety regulations and guidelines can result in severe injury or heavy damage to material.
Caution!	Disregarding the safety regulations and guidelines can result in injury or damage to material.
Information:	Important information used to prevent errors.

Table 2: Safety notices

III Table of Contents

1 Introduction

This document provides technical information about the B&R SRAM driver.
The descriptions and figures refer to B&R SRAM driver version 1.1.

The B&R SRAM driver is a Windows device driver that enables access to a B&R SRAM module.

More information about the SRAM modules can be found in the corresponding user manuals. These manuals are available in the download area of the B&R website (www.br-automation.com).

2 Supported hardware

The B&R SRAM driver can be used with the following IF modules:

- 5PP5IF.FCAN-00
- 5PP5IF.FETH-00
- 5PP5IF.FPLM-00
- 5PP5IF.FX2X-00
- 5PP5IF.FXCM-00
- 5AC901.ISRM-00

3 Supported operating systems

The B&R SRAM driver can be used on the following operating systems:

- Windows XP SP3
- Windows Embedded Standard 2009
- Windows 7 SP1 (32 and 64 Bit)
- Windows Embedded Standard 7 SP1 (32 and 64 bit)
- Windows 8.1 (32 and 64 Bit)
- Windows 10 (32 and 64 Bit)

4 Installation

The following components are installed on the PC:

- SRAM driver
- SRAM Test (optional)
- IF Update (optional)

The SRAM driver is installed in the Device Manager under the category *System devices* with the label *SRAM*. For all other functions of the module are “dummy driver” installed, also under the category *System devices*, with the label *Interface resources*.

> **Information:**
>
> **The SRAM module has to be mounted before installing the driver.**

5 Features

The B&R SRAM driver provides the following features:

- Access to the SRAM with the standard Windows functions *GetFileSize(), ReadFile(), WriteFile()* and *SetFilePointer()*.
- Firmware update

6 Device interface

The following Win32 functions are supported by the B&R SRAM driver device interface.

6.1 Device open

The device can be opened with the Windows function *CreateFile()* via the GUID.

```
DEFINE_GUID (GUID_DEVINTERFACE_BrSramDrv,
0x2951eeb6,0xb43b,0x4112,0xb2,0xb3,0x27,0x46,0xe9,0xf0,0x51,0xcc);

SP_INTERFACE_DEVICE_DATA IfDevData;
DWORD DevInfoLen;
DWORD DevInstance = 0;
HANDLE hSramDev = NULL;


HDEVINFO hDevInfo = SetupDiGetClassDevs(
                                     pGuid,
                                     NULL,
                                     NULL,
                                     DIGCF_PRESENT | DIGCF_DEVICEINTERFACE );

if(hDevInfo  != INVALID_HANDLE_VALUE)
{
    IfDevData.cbSize = sizeof(IfDevData);

    if(SetupDiEnumDeviceInterfaces(
                                  DevInfo,
                                  NULL,
                                  &GUID_DEVINTERFACE_BrSramDrv,
                                  DevInstance,
                                  &IfDevData))
    {
            SetupDiGetDeviceInterfaceDetail(
                                         hDevInfo, &IfDevData,
                                         NULL, 0,
                                         &DevInfoLen, NULL);

            PSP_INTERFACE_DEVICE_DETAIL_DATA pIfDevDetail =
            (PSP_INTERFACE_DEVICE_DETAIL_DATA)malloc(DevInfoLen);

            if(pIfDevDetail != NULL)
            {
                    pIfDevDetail->cbSize = sizeof(SP_INTERFACE_DEVICE_DETAIL_DATA);
                    if(SetupDiGetDeviceInterfaceDetail(
                                                 hDevInfo,
                                                 &IfDevData,
                                                 pIfDevDetail,
                                                 DevInfoLen,
                                                 NULL,
                                                 NULL))
                    {
                            hSramDev = CreateFile(
                                         pIfDevDetail->DevicePath,
                                         GENERIC_READ | GENERIC_WRITE,
                                         FILE_SHARE_READ | FILE_SHARE_WRITE,
                                         NULL, OPEN_EXISTING,
                                         FILE_ATTRIBUTE_NORMAL, NULL);
                    }
                    free(pIfDevDetail);
            }
    }
    SetupDiDestroyDeviceInfoList(hDevInfo);
}
```

Listing 1: Device open

6.2 Device close

The device can be closed with the Windows function *CloseHandle()*.

```
BOOL ret = CloseHandle(hSramDev);
```

Listing 2: Device close

6.3 Get device size

The size of the SRAM can be determined with the Windows function *GetFileSize()*.

```
DWORD dwSize = GetFileSize(hSramDev, NULL);
```

Listing 3: Get device size

6.4 Device read

The device can be read with the Windows function *ReadFile()*. If the end of the SRAM is reached and the byte count is greater than the remaining size, the function will return the number of bytes that are really read.

```
DWORD BytesRead;
BYTE bValue;
BOOL ret = ReadFile(hSramDev, &bValue, sizeof(bValue), &BytesRead, NULL);
```

Listing 4: Device read

6.5 Device write

The device can be written with the Windows function *WriteFile().* If the end of the SRAM is reached and the byte count is greater than the remaining size, the function will return the number of bytes that are really written.

```
DWORD BytesWritten;
BYTE bValue = 0x55;
BOOL ret = WriteFile(hSramDev, (LPCVOID)&bValue, sizeof(bValue), &BytesWritten, NULL);
```

Listing 5: Device write

6.6 Device seek

The Windows function *SetFilePointer()* allows moving the file pointer with respect to the beginning of the SRAM, the current position in the memory or the end-of-memory position. If the new position is before the beginning of the SRAM, the file pointer is not moved and the WIN32 error code *ERROR_NEGATIVE_SEEK* will be returned.

```
DWORD ret = SetFilePointer(hSramDev, 0, NULL, FILE_BEGIN);
```

Listing 6: Device seek

7 IF Update

The IF Update tool allows showing the firmware version of the IF modules and updating the firmware. The firmware updates can be requested via the B&R support.

An update of the firmware is possible on the following IF modules:

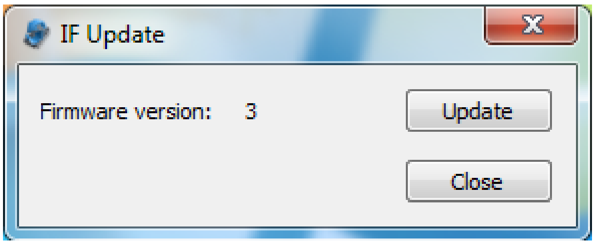
- 5PP5IF.FETH-00
- 5AC901.ISRM-00

Information:

After the firmware update, it is necessary to shut down the device, switch off/on the power supply and reinstall the SRAM driver. Only after that, the new firmware is applied and displayed in the IF Update tool.

Information:

Please note that the IF Update tool can only update the first IF module with SRAM that has been found in the system.



Firmware version: 3

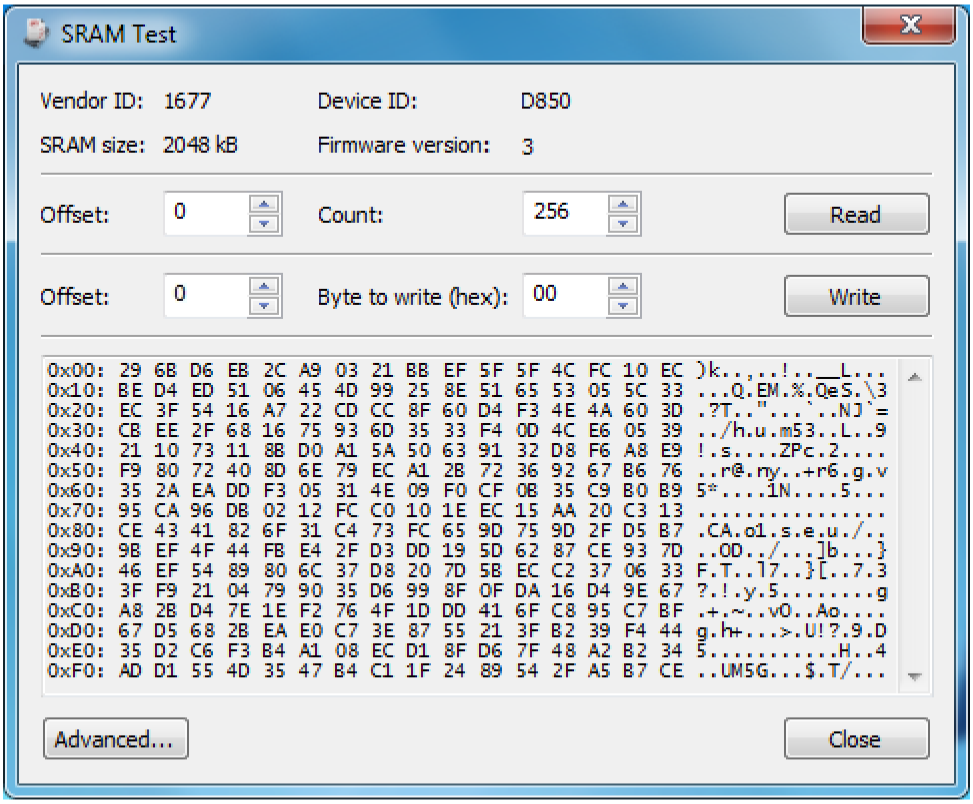
Figure 1: IF Update

8 SRAM Test

The SRAM Test tool allows showing SRAM information (size, firmware version, vendor ID- and device ID), reading and writing the SRAM and executing tests (memory tests and performance tests).

Information:

Please note that the SRAM Test tool can only access the first SRAM module that has been found in the system.



Vendor ID: 1677 Device ID: D850

SRAM size: 2048 kB Firmware version: 3

Offset: 0 Count: 256

Offset: 0 Byte to write (hex): 00

```
0x00: 29 6B D6 EB 2C A9 03 21 BB EF 5F 5F 4C FC 10 EC )k..,..!..__L...
0x10: BE D4 ED 51 06 45 4D 99 25 8E 51 65 53 05 5C 33 ...Q.EM.%.QeS.\3
0x20: EC 3F 54 16 A7 22 CD CC 8F 60 D4 F3 4E 4A 60 3D .?T.."...`..NJ`=
0x30: CB EE 2F 68 16 75 93 6D 35 33 F4 0D 4C E6 05 39 ../h.u.m53..L..9
0x40: 21 10 73 11 8B D0 A1 5A 50 63 91 32 D8 F6 A8 E9 !.s....ZPc.2....
0x50: F9 80 72 40 8D 6E 79 EC A1 2B 72 36 92 67 B6 76 ..r@.ny..+r6.g.v
0x60: 35 2A EA DD F3 05 31 4E 09 F0 CF 0B 35 C9 B0 B9 5*....1N....5...
0x70: 95 CA 96 DB 02 12 FC C0 10 1E EC 15 AA 20 C3 13 ................
0x80: CE 43 41 82 6F 31 C4 73 FC 65 9D 75 9D 2F D5 B7 .CA.o1.s.e.u./..
0x90: 9B EF 4F 44 FB E4 2F D3 DD 19 5D 62 87 CE 93 7D ..OD../...]b...}
0xA0: 46 EF 54 89 80 6C 37 D8 20 7D 5B EC C2 37 06 33 F.T..l7..}[..7.3
0xB0: 3F F9 21 04 79 90 35 D6 99 8F 0F DA 16 D4 9E 67 ?.!.y.5........g
0xC0: A8 2B D4 7E 1E F2 76 4F 1D DD 41 6F C8 95 C7 BF .+.~..vO..Ao....
0xD0: 67 D5 68 2B EA E0 C7 3E 87 55 21 3F B2 39 F4 44 g.h+...>.U!?.9.D
0xE0: 35 D2 C6 F3 B4 A1 08 EC D1 8F D6 7F 48 A2 B2 34 5...........H..4
0xF0: AD D1 55 4D 35 47 B4 C1 1F 24 89 54 2F A5 B7 CE ..UM5G...$.T/...
```

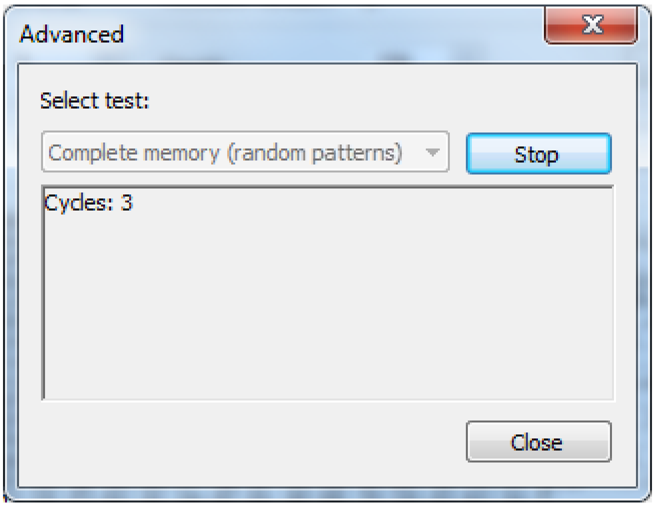
Figure 2: SRAM Test

8.1 Tests

The following tests are provided in the SRAM test tool.

8.1.1 Memory test

The test *Complete Memory (random patterns)* writes random values to the whole SRAM, reads the values back and compares the read values with the previously written values. After each test cycle, the cycles are incremented.



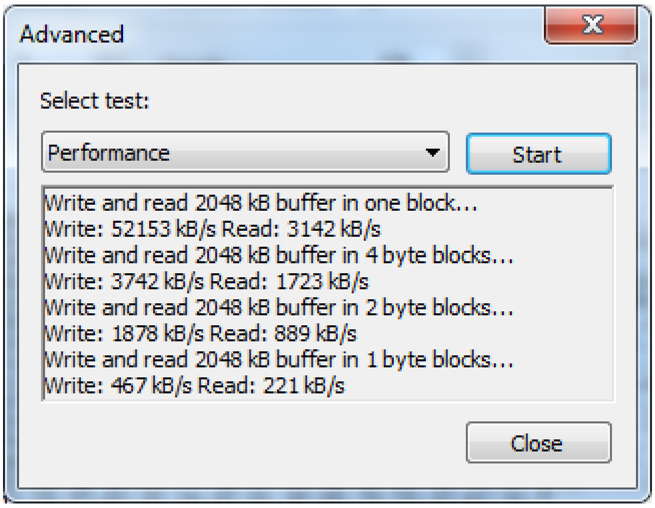
Select test: Complete memory (random patterns)

Cycles: 3

Figure 3: Memory test

8.1.2 Performance test

The test *Performance* writes and reads the whole SRAM with different block lengths and measures the write and read speed.
In the first test step, a buffer with the size of the SRAM is passed in one block to the driver, written to the SRAM und read back.
In the next steps, a buffer with the size of the SRAM is passed in blocks of 1, 2, 4 and 8 bytes (only 64-bit operating systems) to the driver, written to the SRAM and read back.



```
Write and read 2048 kB buffer in one block...
Write: 52153 kB/s Read: 3142 kB/s
Write and read 2048 kB buffer in 4 byte blocks...
Write: 3742 kB/s Read: 1723 kB/s
Write and read 2048 kB buffer in 2 byte blocks...
Write: 1878 kB/s Read: 889 kB/s
Write and read 2048 kB buffer in 1 byte blocks...
Write: 467 kB/s Read: 221 kB/s
```

Figure 4: Performance test

9 Figure Index

10 Table Index

11 Listing Index

12 Index