Related Products	All products that support the JRC software
Subject	Examples of JRC usage
Document Name	JRCUsage_E114.doc
Usage	Common

1. REVISION HISTORY

Date	Document Name	Subjects added, changed, deleted	Changed by
22-Feb-00	JAP0027.DOC	Initial release	C. Hoch
11-Oct-00	JAP0027.DOC	Added MOPSlcd6 and MOPS/686+	C. Hoch
23-Jan-02	JAP0027.DOC	Added all ETX modules, DIMM-PC/520-I and DIMM-PC/586-IE	C. Hoch
23-Apr-02	JAP0027.DOC	English proofreading	D. Gunter
04-Jun-02	JAP0027.DOC	Added chapter 3, JRC commands and MOPSlcdGX1	C. Hoch
04-Jul-02	JAP0027.DOC	Added explanations for memread/write and diskread/write	C. Hoch
23-Aug-02	JAP0027.DOC	Changed to Kontron style	H. Bruhn
17-Dec-02	JRCUsage_E111.DOC	Reformated, added several products	H. Bruhn
03-Mar-03	JRCUsage_E112.DOC	Updated and added new ETX	C. Hoch
04-Mar-03	JRCUsage_E113.DOC	Updated revision history	H. Bruhn
03-Nov-03	JRCUsage_E114.DOC	Added more PC/104 and JRex boards	H. Bruhn

2. TABLE OF CONTENTS

3. BASIC INFORMATION

Kontron Embedded Modules GmbH offers with the JRC-tool a possibility to read or write the memory contents (CMOS RAM, EEPROM, NVRAM, Flash, Hard or Floppy Disks) of an Embedded Computer, which is accessible only by its serial port (COM1 to COM4).
This application note shall give some examples where JRC can be used.
The short form **Kontron** will be used for **Kontron Embedded Modules GmbH** in this document.

3.1. Requirements for JRC usage

The requirements for using **Kontron´s** Remote Control are:

- A **Kontron** module with a JRC-BIOS extension. The following table lists those modules.

Kontron module	JRC client implemented since BIOS
ETX-VEx	**MOD9R110**
ETX-P3m	**MOD8R110**
ETX-P3E/C3E	**MOD7R110**
ETX-P3/C3	**MOD6R120**
ETX-P1	**MOD5R120**
ETX-mgx	**MOD1R130**
DIMM-PC/520-I/-IU	**D501R110**
DIMM-PC/486-I	**D401R110**
DIMM-PC/386-I/-IE	**D201R110**
DIMM-PC/386-B	**D101R112**
MOPSlite	**P386R214**
MOPSplus	**P386R117**
MOPSlcd3	**P388R112**
MOPS/386A	**P389R110**
MOPS/520	**P489R110**
MOPS/586	**P488R110**
MOPSlcd4	**P488R110**
MOPS/MZ	**PMZ1R111**
MOPSlcdMZ	**PMZ1R111**
MOPS/686+	**P588R114**
MOPSlcd6	**P588R114**
MOPSlcdGX1	**PGX1R110**
MOPSlcd7	**PVC3R110**
JRex-GX1	**BQG1R110**
JRex-P3 / JRex-C3	**BQP3R110**
JRex-VE / JRex-VC	**BQC3R110**
JRex-PM	**BQBAR110**

- A regular PC (host) running either Win9x or WinNT with JRC1I1xx.exe installed or MS-DOS with JRC1I2xx.exe installed.
  HINT: rename JRC1I1xx.exe or JRC1I2xx.exe to **JRC.EXE** for your convenience (Note that the DOS version of the JRC tool is usually referred to as **JRCD.EXE**).
- A cable to connect both systems (pls. refer to the JRC manual).

3.2. JRC commands

Image Mode Commands:

connect[<port>[<maxbaud>]]
re**s**ume
re**b**oot
mem**r**ead <mem#> <file>[/b]
mem**w**rite <mem#> <file>[/b]
disk**r**ead <drive> <file>[<sectorcount>]
disk**w**rite <drive> <file>[<sectorcount>][/vv]
flash**r**ead <file>[<blockcount>[<blockstart>]]
flash**w**rite <file>[<blockcount>[<blockstart>]][/v] [/vv]
flash**e**rase[<blockcount>[<blockstart>]]
ipwrite <ipaddr> <ipmask>[<iprouter>][/r]
board**i**nfo[/f]

Server Mode Commands:

ser**v**er <file>|. [<port>[<maxbaud>]][/u]
img**c**reate <drive> <file>

Each command can be abbreviated by usually two letters. These are underlined.

3.3. Memread/write

With this command following areas can be read or written:

- BIOS CMOS Settings in the volatile RAM of the RTC (in the JRC manual referred as CMOS).

  Use command **jrc memread/write 0**.
  During the next boot, this data will be overwritten with the BIOS CMOS Settings stored in the CMOS Backup Flash. Also this data will get lost after power off, if no backup battery is connected to maintain the RTC data.

- BIOS CMOS Settings in the CMOS Backup Flash (in the JRC manual referred as CMOS EEPROM).

  Use command **jrc memread/write 1**.
  The data is written to the Flash when the changes in the BIOS Setup Utility are saved. This data remains stored after the power off, even if no battery is connected to the system. During the next boot the CMOS Settings are copied to the volatile RAM of the RTC.

- Data in the JIDA NVRAM (BIOS CMOS Settings **and** user information).

  Use command **jrc memread/write 2**.
  The NVRAM is the same device used to store the BIOS CMOS Settings (CMOS Backup Flash). Beside this data the NVRAM is used to store user information (alphanumeric-LCD settings, darkboot settings, user bites from EEPROM, etc.) and **Kontron** device specific data (serial number, fabrication date, last repair date, etc.). This device specific data is stored in a secured area and is not affected by the **meread/write 2** command!

3.4. Diskread/write and Flashread/write

- The command **diskread/write** can be used for all storage device accessed by interrupt INT13h.
  This storage device can be a floppy drive, a Solid State Disk (SSD, on the DIMM-PC/386-B), a Flash Disk (for example a chipDISK) or a Hard Disk Drive.
  Floppy drives known in DOS as A: respective B: can be accessed with the command **diskread/write 0** respective **1** and the storage device known as C: respective D: can be accessed by using the command **diskread/write 128** respective **129**.

- When using the onboard SSD (BIOS compatible Flash Disk) the command **diskread/write** can be replaced with **flashread/write**.
  The difference between the two commands is how the data is handled. While **diskread** stores the data according to the track and sector organization of the SSD, **flashread** stores the data according to the flash page and block organization.

The commands **diskwrite** respective **flashwrite** must be used only with images made with **diskread** respective **flashread**.

4. CHANGING THE BIOS SETTINGS

Starting with the JRC Client version 2.5, and JRC Host version 2.7, a very useful feature has been implemented. The clients BIOS settings can be changed using the JRC server mode. In order to do this, the host and the slave have to be connected and the JRC server mode has to be started:
jrc server <port> <maxbaud>

After the connection to the client has been made, press the **"F2"** button on the host keyboard to enter the CMOS setting screen. Now the necessary changes can be made. After the changes have been stored and the BIOS Setup Utility has been closed the client will reboot and enter the server mode again.

Boards that support JRC Client version 2.5 or higher:

Kontron module	JRC client implemented since BIOS
ETX-VEx	**MOD9R110**
ETX-P3m	**MOD8R110**
ETX-P3E/C3E	**MOD7R111**
ETX-P3/C3	**MOD6R123**
ETX-P1	**MOD5R121**
ETX-mgx	**MOD1R311**
MOPS/386A	**P389R113**
MOPS/520	**P489R113**
MOPSlcdGX1	**PGX1R110**
MOPS/686+ redesign –17-x/-27-x	**P588R140**
MOPSlcd6 redesign –17-x/-27-x	**P588R140**
MOPSlcd7	**PVC3R110**
JRex-GX1	**BQG1R110**
JRex-P3 / JRex-C3	**BQP3R110**
JRex-VE	**BQC3R110**
JRex-PM	**BQBAR110**

5. BOOT UP POSSIBILITIES WITH JRC

If there is no floppy disk drive, HDD, Flash Disk or SSD connected to the **Kontron** unit, it can be booted from a image file on the hard disk of the host or by mapping the host's floppy drive to the clients drive specifier A:.

- The image of a bootable floppy disk can be created with the command **jrc imgcreate a: bootdisk.img**. After this, a client can boot from this image in the JRC server mode by using the created image file: **jrc server bootdisk.img**.

- The **Kontron** device can also boot from an image made from the bootable HDD, Flash Disk or SSD. This image can be made with the command **jrc diskread 128 <file>**.

- With the command **jrc server a:** the client boots from the floppy inserted in the host's floppy drive A: (be sure that the boot sequence in the BIOS is set to **A:, C:** and the floppy disk drive contains a bootable floppy).

6. HOW TO "CLONE" A KONTRON MODULE

With JRC the customer has the possibility to clone a **Kontron** module. That means the contents of the HDD, Floppy Disk or SSD, the NVRAM and the CMOS settings of a sample device are transferred into a new device, the "**clone**".

NOTE:

Do not use the command **jrc memread/memwrite 2 <file>** with **BIOS revisions up to XxxxR121** (MOPSlite, MOPSplus, MOPSlcd3, DIMM-PC/386-B and DIMM-PC/386-I) and **BIOS revisions up to D401R110** (DIMM-PC/486-I). This command causes troubles when used with the mentioned BIOSes.
All other Kontron boards are not affected.

It's important that the sample and the clone use the same BIOS revisions and the size of the storage device (SSD, Flash Disk or HDD) being used are the same.

6.1. Steps for preparing the data needed for cloning

	command	Remark
1.	**jrc connect [<port> [<maxbaud>]]**	- <port> = used port on the host PC, <maxbaud> can be up to 115200 baud. This command reboots the client if a connection already exists. If there is no connection between host and client, the client must be reset manually.
2.	**jrc memread 1 eeprom.img**	- writes the contents of the CMOS Backup Flash into the mentioned file.
(3.	*jrc memread 2 nvram.img*)	- this command reads all data stored in the EEPROM (CMOS setup changes and user data) and writes the contents into the mentioned file. When this command is used, there is no need for the command **jrc memread 1 <file>**.
3.	**jrc diskread 128 ssd.img**	- writes the contents of the HDD, Flash Disk or SSD (enabled in the CMOS setup as drive C:) into the mentioned file. NOTE: The time for reading a 1MB SSD can be up to 2 min (!).
4.	**jrc reboot**	- reboots the client. After this command, the connection between the host and client is terminated.

6.2. Steps for writing the sample data to the clone

	command	Remark
1.	**jrc connect [<port> [<maxbaud>]]**	- This command reboots the client if a connection already exists. If there is no connection between the host and client, the client must be reset manually.
2.	**jrc memwrite 1 eeprom.img**	- writes the contents (BIOS CMOS Settings) of the mentioned file to the CMOS Backup Flash of the client. Use files created only with **'memread 1'** command.
(3.	*jrc memwrite 2 nvram.img*)	- this command reads the contents of the mentioned file and writes the data into the EEPROM (CMOS setup changes and user data). When this command is used, there is no need for the command **jrc memwrite 1 <file>**. Use files created only with **'memread 2'** command.
4.	**jrc connect [<port> [<maxbaud>]]**	- the client must be rebooted so the BIOS CMOS Settings are taken over.
5.	**jrc diskwrite 128 ssd.img**	- writes the contents of the mentioned file to the HDD, Flash Disk or SSD (enabled in the CMOS setup as drive C:) of the client. NOTE: The time for writing a 1MB SSD can be up to 7 min (!). Use files created only with **'diskread'** command.
6.	**jrc reboot**	- reboots the client. After this command, the connection between the host and client is terminated.

6.3. JRC commands in a batch file

The JRC commands could be inserted into a batch file so that the necessary steps will be carried out automatically.

Example: getdata.bat

```
jrc connect com3 57600
    if errorlevel 1 goto error
jrc memread 1 eeprom.img
    if errorlevel 1 goto error
.....
.....
echo SSD ready
goto end
:error
        echo An error occurred.
:end
```

7. JRC IMAGES AFTER A BIOS UPDATE

Just affecting products with Solide State Disks (SSD's)

When updating to a new BIOS revision, the structure of how the SSD BIOS organizes the flash disk space may change. In this case the new SSD BIOS is not able to handle the actual data stored in the flash device anymore. In a worst case scenario the SSD BIOS considers the flash disk data totally corrupted and erases it automatically. In order to save the flash disk data before updating to a new BIOS revision you must copy the files to some other device using the DOS command **"copy"** or any other command or tool suitable to copy files. The target device may be a local floppy disk drive, a local hard disk drive or a remote floppy disk drive (via **jrc server a:**).

NOTE! You cannot use the **jrc diskread** or **jrc flashread** commands to save the data because the images created with these commands would also store the flash data using the structure of the old SSD BIOS.

After a BIOS update the structure of the used onboard SSD must be reorganized. The SSD must be erased, which can be done by setting the option **"Erase SSD On Next Boot"** in the BIOS-setup or by using the JRC command **flasherase**.

On some **Kontron** devices the BIOS is stored in the same flash device used for the SSD. None of the JRC commands **flashread/write** or **erase** can affect the BIOS flash blocks!

7.1. Example: install a new operating system after a BIOS update.

	Command	remark
1.	**jrc connect**	
2.	**jrc flasherase**	- Erases the contents of the flash.
3.	**jrc server a:**	- The system must now be rebooted. The command **jrc server** reboots the system if a connection already exists. The bootable floppy in the host's floppy drive A: should contain the DOS utility programs **fdisk** and **format**.
4.	The host and the client are now in server mode. With **fdisk** (JRC command window), partitions can be created on the clients SSD.	
5.	Exit the server mode by pressing, simultaneously, both control keys on the keyboard and turn the system power off.	
6.	**jrc server a:**	- Turn the system power on and enter into the server mode again.
7.	The client's SSD can be formatted. **format** (JRC command window).	
8.	Now the operating system can be copied from the host's floppy drive A: to the clients SSD.	
9.	Exit the server mode by pressing simultaneously, both control keys on the keyboard. After this, the connection between the host and client is terminated	