

INSTRUCTION MANUAL

Sensoray Model 2411

Still Image Capturing Server

Revision 1.2

Aug 26, 2005

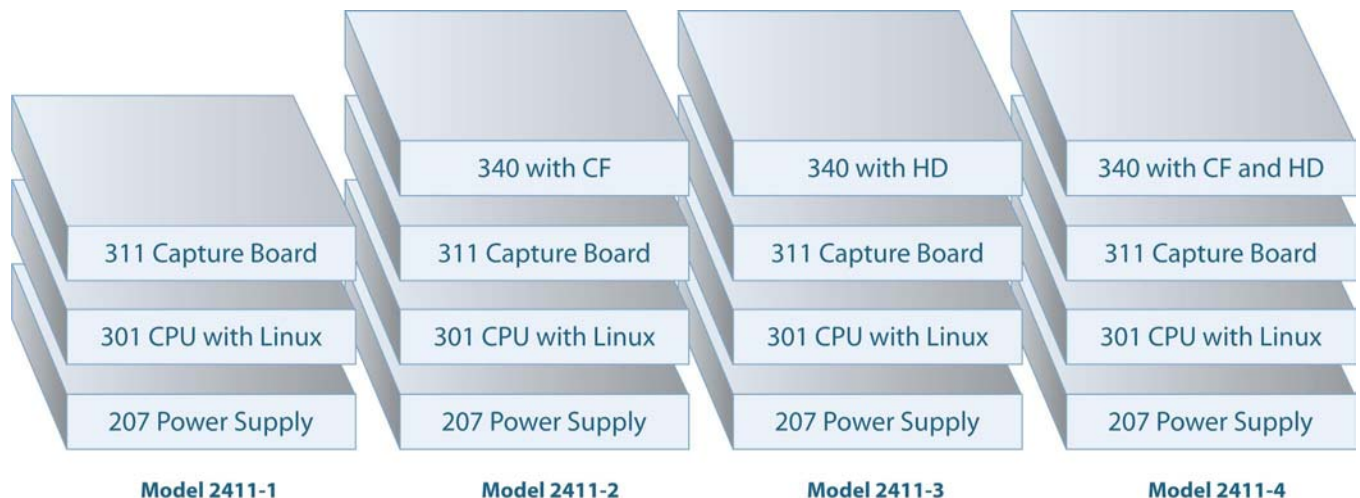


I. Introduction:

The Sensoray model 2411 stack consists of a Sensoray Model 301 CPU board, one/two Sensoray Model 311 frame grabber board(s), a Sensoray Model 207 power supply board, an optional Sensoray Model 340 PCMCIA adaptor or Model 300TA IDE adaptor board, and an embedded Linux operating system installed on a solid state disk.

As a summary, following 2411 configurations can be chosen:

- 2411-1: 301+311(s)+207
basic configuration, without big storage; the captured frames can be stored in SDRAM on 301, and/or transferred over Ethernet via FTP or UDP apps.
- 2411-2: 301+311(s)+340+207
with PCMCIA type of storage, like CF (Compact Flash) or MD (MicroDrive) plus PCMCIA adapter, or removable PCMCIA HD (Hard Drive); the captured frames can be stored on such PC card type of storage, and/or transferred over Ethernet via FTP or UDP apps.
As an alternative and unreleased configuration, the captured frames can be stored on a USB HD (Hard Drive) through one of USB1.1 ports on 340.
- 2411-3: 301+311(s)+300TA+207
with IDE HD (Hard drive) storage; the captured frames can be stored on an IDE type of HD, and/or transferred over Ethernet via FTP or UDP apps.
- 2411-4: 301+311(s)+ 300TA+340+207
with IDE HD storage and PC-card type of storage or USB HD; the captured frames can be stored on a CF/MD and HD, and/or transferred over Ethernet via FTP or UDP apps.



The v4l (Video4Linux) bttv is used for frame capturing driver in the 2411. All related modules are pre-built and distributed with 2411 embedded Linux OS kernel. The version of the bttv used in the 2411 is bttv-0.7.85.

The frame capturing can be initiated/controlled by software or triggered by a digital signal (for example, from a push button) with hardware connection to GPIO port on 311 in the stack. A separate device driver is used for GPIO operation (refer to Appendix-III for details).

After applying power, this system will automatically boot from the embedded Linux and be fully functional after about 20 seconds.

The 2411 is set up to user's desired mode of operation by modifying configuration files. Some minimal knowledge of Linux would be helpful. There are two ways to accomplish this. The simplest is to open a telnet session on a workstation connected to the network with the IP address (10.135.1.99, default setting on factory 2411, which can be changed once you get accessed to the target 2411 unit). If this IP address conflicts with another node on your network, you will need to use an alternative way connecting a dumb terminal (or terminal emulator (VT100) on a PC) to the serial port (com1) on the 2411 (9600 baud, 8 bits, no parity). A step-by-step procedure will be given.

Procedures for setting up and getting started with 2411 are given below.

II. Setting up the 2411 via network connection (recommended):

1. Connect network cable to 2411. If the factory IP address (10.135.1.99) conflicts with your network you need to follow the "Setting up the 2411 via serial terminal" procedure (in next section).
2. Connect +12V to +24V DC to the Power Supply board, Model 207 in the stack. Turn on the power.
3. Wait 20 seconds, or till system boot up.
4. Open a telnet session on a workstation connected to the network and telnet to 10.135.1.99.

HINT: From a Windows based workstation a telnet session can be opened from a DOS window (Command prompt window), typing 'telnet 10.135.1.99' and then pressing <enter>.

5. At '2411 login' prompt, type 'root' and then press <enter> key.
6. At 'Password' prompt, type factory password and press <enter> key
7. You should now see "2411SYS:~ #". If you mistyped something and the login failed, just press <enter> until you get the '2411 login' prompt again.
8. Type 'cd /usr/etc' to take you to the directory containing the user modifiable configuration files.
9. Type 'ls' and a listing of the files will be shown.
10. Now the IP address can be changed. At the shell prompt, type 'netconfig eth0 -i your_ip -m your_netmask' and then press <enter>.

for example, 'netconfig eth0 -i 192.168.1.23 -m 255.255.255.0'

Once done, reboot the 2411 (at the shell prompt type 'reboot' and then press <enter>).

11. Another way to modify the IP address, is to edit the 'system.config' file by typing "vi /usr/etc/system.config".

12. vi is a small editor, loosely based on old WordStar. If the display is incorrect, type 'ctrl – R'. If vi refuses to start, your telnet client is either not setup properly or is a bad telnet client (Windows HyperTerminal does work, if setup correctly). Type 'ctrl – KX' to exit the editor, and 'ctrl – KH' for a help screen.
13. When finished changing parameters. You can either wait 10 seconds and then turn the power off or type 'reboot' to restart the 2411.
14. If you have changed the IP address, you will need to use the new IP address with subsequent telnet sessions.

III. Setting up the 2411 via serial terminal: (should be used if there is a network conflict, or you have forgotten the target IP address, or any other network problem):

1. Connect one end of a serial (crossover / null modem) cable to the 2411 com1 port (a DB-9 to 10 pin header cable is supplied) and connect the other end of the serial cable to either a dumb terminal or a workstation with terminal emulation.
Connect the DB-9-to-10 pin header cable mentioned above to the 301 CPU board in the stack via a header connector located next to the RJ-45 network jack. When installed, the red wire on the cable must point away from the RJ-45 jack.
The terminal settings must be 9600 baud, no parity, 8 bits, 1 stop, software flow control (xon/xoff). Ideally the terminal type should be VT100 but ANSI will work.

HINT: Most Windows based workstations have HyperTerminal installed (under Programs/Accessories/Communications). To setup, start HyperTerminal, go to File -> Properties -> Configure and set the bits per second to 9600, bits to 8, stop bits to 1, parity to none, and flow control to Xon/Xoff. Press OK and set the com port.

2. Start the terminal.
3. Connect +12V to +24V DC to the Power Supply board, Model 207 in the stack. Turn on the power.
4. Within 5 seconds or so you should be seeing the console output on the terminal. The boot process will be finished after about 20 seconds.
5. At '2411 login' prompt, type 'root' and then press <enter> key.
6. At 'Password' prompt, type factory password and press <enter> key
7. You should now see "2411SYS:~ #". If you mistyped something and the login failed, just press <enter> until you get the '2411 login' prompt again.
8. Type 'cd /usr/etc' to take you to the directory containing the user modifiable configuration files.
9. Type 'ls' and a listing of the files will be shown.
10. Now the IP address can be changed. At the shell prompt type 'netconfig eth0 -i your_ip -m your_netmask' and then press <enter>.

for example, 'netconfig eth0 -i 192.168.1.23 -m 255.255.255.0'

11. Another way to modify the IP address, is to edit the 'system.config' file by typing "vi /usr/etc/system.config".
12. vi is a small editor, loosely based on old WordStar. If the display is incorrect, type 'ctrl – R'. If vi refuses to start or doesn't behave properly, your terminal is not setup properly (Windows HyperTerminal does work, if setup correctly). Type 'ctrl – KX' to exit the editor, and 'ctrl – KH' for a help screen.

IV. Accessing 2411 via FTP:

The 2411 can be accessed via FTP (either command line based FTP or Windows based FTP). This is an easy way for the 2411 user(s) or application developer(s) to transfer files between 2411 target system and any computer connected to the network. For example, a developer/user may use FTP to load the cross-compiled application(s) from a development computer to the 2411 system, or upload the captured image files from the 2411 system to a host computer.

A. From a DOS command prompt or a Linux shell terminal prompt:

Simply type 'ftp 2411_ip_addr', <enter>, to login from 'root'.
(for example, 'ftp 10.135.1.99', if you still use factory IP address)
Then type any other FTP commands to do your transferring.

B. From a Windows explorer or a Linux file manager/browser like Konqueror:

In address/location, type 'ftp://root@2411_ip_addr', <enter>, and then login from 'root' within a pop-up login window. (For example, 'ftp://root@10.135.1.99', if you still use factory IP address for the 2411). Then go to any folder/directory to manipulate the files you are interested in.

V. Demo Applications:

In order to demonstrate frame grabbing with the 2411, nine sample applications are pre-built and loaded into the 2411 system. In the distributed 2411 system (stack), you will find the demo programs, under directory /usr/user/demo2411:

- pgm2411ntsc: is for NTSC black/white frame capturing; image size been set to: 320x240
- ppm2411ntsc: is for NTSC color frame capturing; image size been set to: 640x480
- ppm2411pal: is for PAL color frame capturing; saved image size been set to: 768x576
- ppm2411p: is for PAL color frame capturing; saved image size been set to: 640x480
- grab10_CF: grabbing 10 frames, NTSC color, about 1 frame/sec, size: 640x480
save images in .ppm files (bitmap under Unix/Linux) on PC-card storage like CF, MD, or HD
- grab100_CF: grabbing 100 frames, NTSC color, about 2 frame/sec, size: 320x240
save images in .ppm files (bitmap under Unix/Linux) on PC-card storage like CF, MD, or HD
- grbjp10: grabbing 10 frames, NTSC color, JPEG compression, size: 640x480
save images in .jpg files on PC-card storage (CF, MD, or HD)
- lpcapbmp_cf: grabbing triggered with a push-button, NTSC color, size: 640x480
save images in .bmp files (bitmap under Windows) on PC-card storage like CF, MD, or HD
- btgjpg: grabbing 100 frames triggered with a push-button, NTSC, size: 640x480
convert and save images into .jpg files on PC-card storage (CF, MD, or HD)

All sample application source code also been copied into the same directory with distribution. User(s) could develop their own applications by referring to or directly port from the sample code.

To experience and/or test the frame capturing with 2411 and run individual sample apps:

1. Connect a video source to J2 (BNC connector) on Model 311.
2. Make sure 311 jumper settings are one of the following combinations for configuration 2411-1 and 2411-3:

```
J3AB, J8 closed      ( J3-BA: || -- slot 0,  with J8: | -- INT A )
J3B, J7 closed       ( J3-BA: |: -- slot 1,  with J7: | -- INT B )
J3A, J6 closed       ( J3-BA: :| -- slot 2,  with J6: | -- INT C )
J3AB open, J5 closed ( J3-BA: :: -- slot 3,  with J5: | -- INT D )
```

3. For configuration 2411-2 and 2411-4, make sure 311 jumper settings are one of the following combinations:

```
J3AB, J8 closed      ( J3-BA: || -- slot 0,  with J8: | -- INT A, default )
J3AB open, J5 closed ( J3-BA: :: -- slot 3,  with J5: | -- INT D )
```

The 340 jumper settings should be:

```
J1-pin1 and J1-pin2 closed ( disable Ethernet port on 340 )
J2-pin2 and J2-pin3 closed ( enable USB ports on 340 )
J3-pin2 and J3-pin3 closed ( enable PCMCIA on 340 )
J4 open                    ( free INTA# from 340 )
J5 closed                  ( USB routed to INTB# )
J6 closed                  ( PCMCIA routed to INTC# )
```

4. Connect the 2411 to the network. Or 5 below.
5. Connect a serial cable described in III from a computer that will be used as the console terminal to J2 on 301, if you are going to operate the 2411 via a serial terminal.
6. Connect +12 to +24V DC to the 2411 Power Supply Board, Model 207. Turn on the power supply to power the 2411 system.
7. After about 20 seconds, the 2411 will be ready for login.
8. Open the terminal for 2411 console login, either via telnet through the network connection as described in II or via serial terminal through the serial port as described in III. (See section II or III above for the details).
9. Enter 'root' and password you set up to login.
10. Type 'cd /usr/user/demo2411' to the directory under which the demo apps are resided.
11. Type 'modprobe -r bttv' and 'modprobe bttv card=73 pll=1' to load the bttv driver. Note that 'modprobe -r bttv' is optional and just for the case you have loaded bttv without using pll=1 option. For PAL video source user(s), the pll=1 option is needed to make proper capture with the bttv for PAL video source.
12. Run any one of the './pgm2411ntsc', './ppm2411ntsc', './ppm2411pal', './ppm2411p', './grab10_CF', './grab100_CF', and './grbjp10' sample applications for capturing with the 2411. The demos will show you the directory and file names into which the captured frames/images have been saved. When testing 'pgm2411ntsc', 'ppm2411ntsc', 'ppm2411pal', and/or 'ppm2411p', the output directory /var/. is mounted to SDRAM on 301. When running 'grab10_CF', 'grab100_CF', and 'grbjp10', the output directory /mnt/cflash/. is mounted to PC-card storage media (CF, MD, or HD) on 340 (for 2411-2 or 2411-4).
13. If you choose 2411-2 or 2411-4 configuration, you can point to the mount point '/mnt/cflash/.' as output directory for all the sample (demo) applications, like 'grab10_CF', 'grab100_CF', and 'grbjp10' pointed. This will let you save the captured frames/images onto a PC-card type of storage (CF, MD, or HD) through 340. Revise the sample source code and build the new application with gcc on a Linux workstation that is used as a development host computer. Load

the new executable app code from host to the 2411 target system via FTP. Then run the app for test. Refer to Appendix-I for the instructions on operating a PC-card type of storage media in the 2411-2 or 2411-4 systems.

14. For 2411-3 and 2411-4, you can point to the mount point '/media/hda1/.' for saving the captured frames/images onto HD (Hard Drive). Revise the sample source code and build the new application with gcc on a Linux workstation that is used as a development host computer. Load the new executable app code from host to the 2411 target system via FTP. Then run the app for test. Refer to Appendix-II for the instructions on setting up HD in the 2411 system.
15. For running the sample apps capturing with push-button triggering, you need to load the driver for 311 GPIO and Push-Button with "modprobe swbt311drv", before running "lpcapbmp" or "btgjpg". As mentioned at the beginning of this section (V), the output directory for this two demo applications is pointed to '/mnt/cflash' that is mounted on PC-card storage media (CF, MD, or HD) on 340.
16. Use FTP described in section IV to upload the captured image file(s) from 2411 target system to your host/storage computer. Then, all .ppm, .pgm and .jpg image files could be viewed with Linux utilities like "xv", "Kuickshow", "imageMagick", "GIMP", Those utility applications give you the ability to convert the images from a format to another/other different formats. Or, if you upload the captured .jpg image files or bring-over the captured .jpg image files on a PC-card type of storage to a Windows platform, the .jpg files could be viewed with any Windows based image viewers or processing programs.
17. When all done, shutdown the system with 'shutdown -t 0' and then turn-off the power supply. Or, simply turn-off the power supply.

VI. Application Development:

1. Use a Linux workstation as a cross development computer. Make sure that it has editor(s) and gcc installed.
2. Use any stand-alone editors like 'vi', 'Emacs', 'Kwrite', 'Kate', ..., or any built-in editor with any IDEs like 'Kdevelop' to develop your app code in C or C++.
3. Use gcc to compile and build your executable app program for the target system.
4. Connect the development Linux workstation and the 2411 system to the network via Ethernet.
5. Via FTP (refer to section IV), load the executable app from the development Linux workstation to a desired place/directory in the 2411 system.
6. Through telnet or serial terminal described in II or III, get launched into 2411 console terminal.
7. Login to 2411.
8. cd to the directory you have put your app code to. For example, 'cd /usr/user/my_app'.
9. Run app and check your results.
10. Repeat 2, 3, 5, and 9, if you need debugging.

VII. Other Documentations:

Refer to following documents, for the detailed hardware information on each individual model/board in the 2411 stack,

- 301 Instruction Manual
- 311 Instruction Manual
- 340 Instruction Manual
- 300TA Instruction Manual
- 207 Instruction Manual

Appendix-I:

Instruction to Using PC-Card type of Storage Media (CompactFlash, MicroDrive, and HardDrive) (for Sensoray 24xx only)

1) check the PCMCIA setting in file /usr/etc/system.config, and make sure that the setting is still with default setting (PCMCIA = "yes"). The PCMCIA section should be like:

```
#Enable PCMCIA - "no" or "yes"
PCMCIA="yes"
```

Note: this step is optional. If you never touch the distributed 24xx kernel and make any change different from default setting, skip this step.

2) check and make sure the following files under the certain directories:

```
ide_cs.o    under:    /lib/modules/2.4.10-S301/pcmcia
ide         /etc/pcmcia
ide.conf    /etc/pcmcia
ide.opts    /etc/pcmcia
```

run "ls -la /etc/pcmcia" to check file "ide" with permission "-r-xr-xr-x". If not, run "cd /etc/pcmcia", "chmod 555 ide".

Note: step 2) is optional. If you never touch the distributed 24xx kernel and make any change different from distribution, skip this step.

3) with latest 24xx kernel distribution, following PC-Card type of storage media are supported:

SanDisk	CompactFlash (CF)
Lexar	CompactFlash (CF)
Hitachi	MicroDrive (MD)
Toshiba	PC-Card ATA/PCMCIA HD (Hard Drive)

If user(s) need add different brand(s) of PC-Card storage media, contact Sensoray for support of adding. Once you received new "ide.conf" and "ide.opts" files from Sensoray, you need load them (via ftp) into 24xx target /etc/pcmcia to replace old ones for the new support. Note: you may need to run "remount -w /" first before loading the files down to 24xx target. It will enable write access to kernel directories for loading the new files.

4) run "lsmod", see if you get the related driver modules installed and loaded. Note: this step needs you have a PC-Card storage media (CF, MD, or HD) inserted in one of PC-Card slots on Model 340 in 24xx stack:

```
msdos
fat
ide-disk
ide_cs
ide-probe-mod
ide-mod
```

Note: this step is optional and just for checking. If you never touch the distributed 24xx kernel and make any change to the kernel, skip this step.

5) run "mount" and make sure that "/dev/hda1 on /mnt/cflash type msdos (rw)" is in the displayed message. If you see it, it shows that the mount point /mnt/cflash has been mounted correctly, and all files that you will access or will save into under /mnt/cflash should be on the PCMCIA PC-Card storage (that you are using).

Note: this step is optional and just for checking. If you never touch the distributed 24xx kernel and make any change to the kernel, skip this step.

6) if you made any setting or kernel file change, "depmod -a" you may need. Then, run "shutdown -h 0" to shutdown the system.

7) insert the PC-Card storage media (CF, MD, or HD) into one of the PC Card slots on Model 340 in the 24xx stack.

8) "restart" (re-power up) the system.

9) login. ("root", return; password, return)

10) go/point to mount point /mnt/cflash, to access (read/write/cp/ftp) files on PC-Card storage media (CF, MD, or HD).

11) for safely hot-ejecting any storage PC card on 340 in the stack, run "cardctl eject" before real physically ejecting any PC cards.

12) for inserting a new media PC card or inserting the old one back into a PC-card slot, run "cardctl insert" before inserting any PC cards physically.

13) following commands are available for un-mount and re-mount a PC-card storage media. For safety reason, you may need to do so some times (for example, you need temporarily prevent the files on the media from accessing for protection).

```
"umount /mnt/cflash"          un-mount PC-card storage media from /mnt/cflash
"mount /dev/hda1 /mnt/cflash" re-mount PC-card storage media back onto /mnt/cflash
```

14) with hot-ejecting and hot-inserting, a PC-card storage media (CF, MD, or HD) could be moved out from 24xx stack and be accessed with/from other Windows/Linux machine(s). As a main application, for example, a PC-card storage media is normally used to save the image files captured with 2411 or save the video data/clips recorded with 2412/2416, then brought to Windows platform for viewing or processing.

Appendix-II:

Instruction to Enable Hard Drive (for Sensoray 24xx only)

1. Install Sensoray's model 300TA, connect IDE flat cable and power cable to the hard drive.
2. The IDE driver is not loaded automatically by default, because normally hard drive is not installed on the system.
3. Hard drive should be mounted BEFORE using it.
4. Under /usr/etc, open "services" file using any editor. For example, "vi /usr/etc/services".
5. Add "mount /dev/hda1 /media/hda1" in the file.

Example:

```
mount /dev/hda1 /media/hda1
modprobe bttv card=73 pll=1
sleep 5
```

6. Restart 24xx

Appendix-III:

Instruction to GPIO and Push Button Driver (for Sensoray 2411/311/611 only)

1) with any ftp utility like FileZilla, load the distributed driver "swbt311drv.o" (for 311 GPIO and Push-Button) down to the following directory in the 2411 target system: /lib/modules/2.4.10-S301/kernel/drivers/sensoray/.

2) run following to make node for 311 GPIO and Push-Button:

```
"cd /dev", "mknod swbt311drv0 c 126 0", "mknod swbt311drv1 c 126 1",  
"ln -s /dev/swbt311drv0 /dev/swbt311drv"
```

3) run "depmod -a" to update dependency of the modules.

4) "reboot" system.

5) login. ("root", return; password, return)

6) run "modprobe bttv card=73 pll=1" to load the "bttv" driver; and
run "modprobe swbt311drv" to load the "swbt311drv" driver.

7) now, you are ready to run your own frame capturing application or demo application distributed with 2411. For example, "cd /usr/user/demo2411" and "./lpcabbmp_cf" to run "lpcabbmp_cf" demo program described in V; "cd /usr/user/demo2411" and "./btgjjpg" to run "btgjjpg" demo program described in V.

8) as a summary, the pin definition of 311 JP1 for 311 GPIO is given blow:

pin-1	GPIO0	pin-2	LVDD
pin-3	GPINT	pin-4	GPO3
pin-5	GND	pin-6	GND
pin-7	+12V	pin-8	+5V
pin-9	GPO2	pin-10	GPI3
pin-11	GPI2	pin-12	GPI1
pin-13	GPO0	pin-14	GPO1

There are 4 general inputs (GPIO0, GPI1, GPI2 and GPI3), and 4 general outputs (GPO0, GPO1, GPO2 and GPO3). Except GPIO0 is optocoupled, all other GPIOs are as digital Inputs/Outputs with TTL signal level. Refer to "311 Instruction Manual" for more detailed information on GPIO signals of 311 and circuitry connection.

In the demo application "lpcabbmp_cf" and "btgjjpg", the GPIO0 is used to connect a push-button that will be used to take trigger for starting frame capturing. As the GPIO0 is optocoupled, the JP1-pin-1 and JP1-pin-5 of 311 are connected to the two ends (terminals) of a switch push-button, respectively; and JP1-pin-2 and JP1-pin-8 are connected together to provide +5V power for optocoupler.

9) In the "swbt311drv" driver, two API functions are provided:

ioctl(fd_bt, ReadRegDrv, &request) is used for GPIO read. The parameter "request.param1" contains the data read-in from the GPIO of 311. The GPI3&GPI2&GPI1&GPIO0 are aligned up on bit3&bit2&bit1&bit0 of "request.param1". And, the status of GPO3&GPO2&GPO1&GPO0 are aligned up on bit7&bit6&bit5&bit4 of "request.param1".

ioctl(fd_bt, WriteRegDrv, &request) is used for GPIO write. The parameter "request.param1" contains the data that will be written out to GPIO. The GPO3&GPO2&GPO1&GPO0 are aligned up on bit7&bit6&bit5&bit4 of "request.param1".