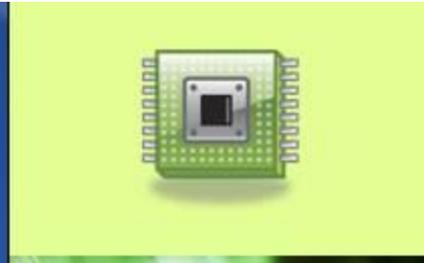
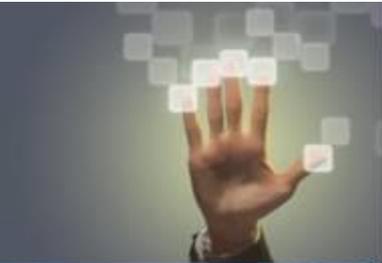


# SPI Burner Overview

Driving Innovations™

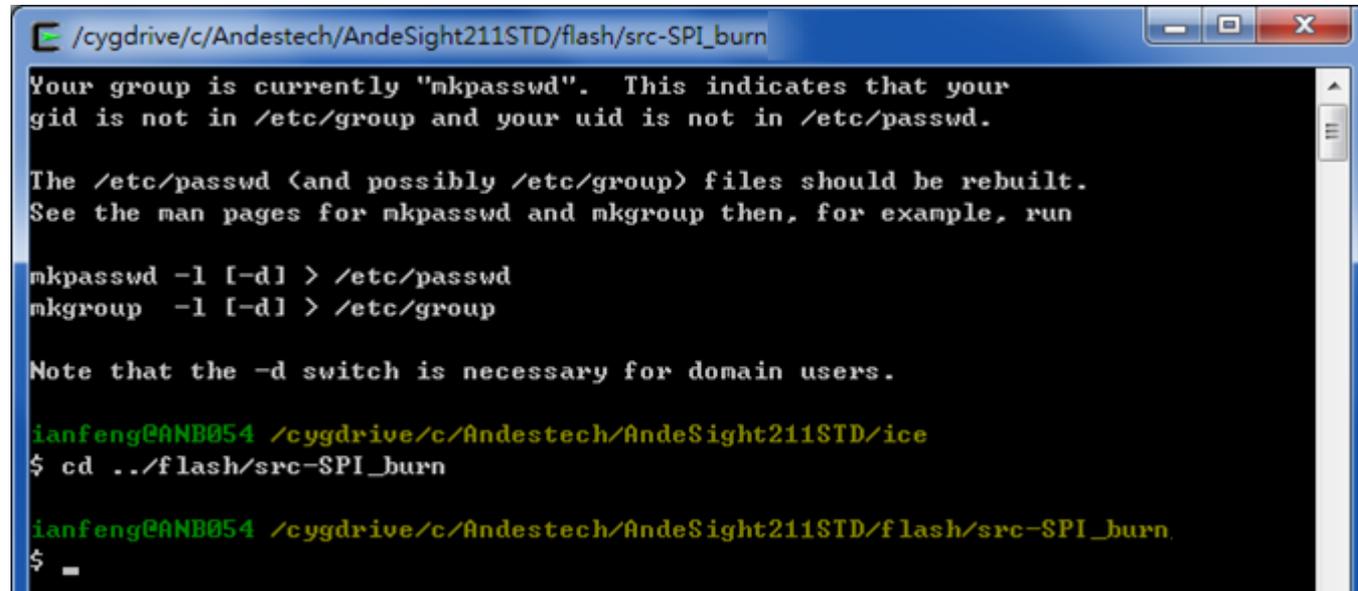
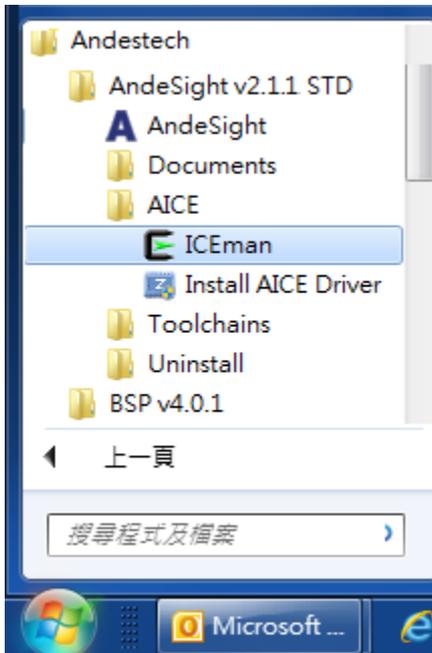


# Outline

- ❖ Build Burner Flow
- ❖ Socket connect to ICEman
- ❖ SPI Burner Flow
- ❖ Parse Parameters
- ❖ Target and Flash related Functions
- ❖ Burn Image
- ❖ Burn NOT Block-Aligned Data (opt.)
- ❖ Burn Boot Code

# Build Burner Flow (1/2)

- ❖ Open "Cygwin" environment.
- ❖ Get in the SPI burner source code in "C:\Andestech\



```
/cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
Your group is currently "mkpasswd". This indicates that your
gid is not in /etc/group and your uid is not in /etc/passwd.

The /etc/passwd (and possibly /etc/group) files should be rebuilt.
See the man pages for mkpasswd and mkgroup then, for example, run

mkpasswd -l [-d] > /etc/passwd
mkgroup -l [-d] > /etc/group

Note that the -d switch is necessary for domain users.

ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/ice
$ cd ../flash/src-SPI_burn

ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
$
```

# Build Burner Flow (2/2)

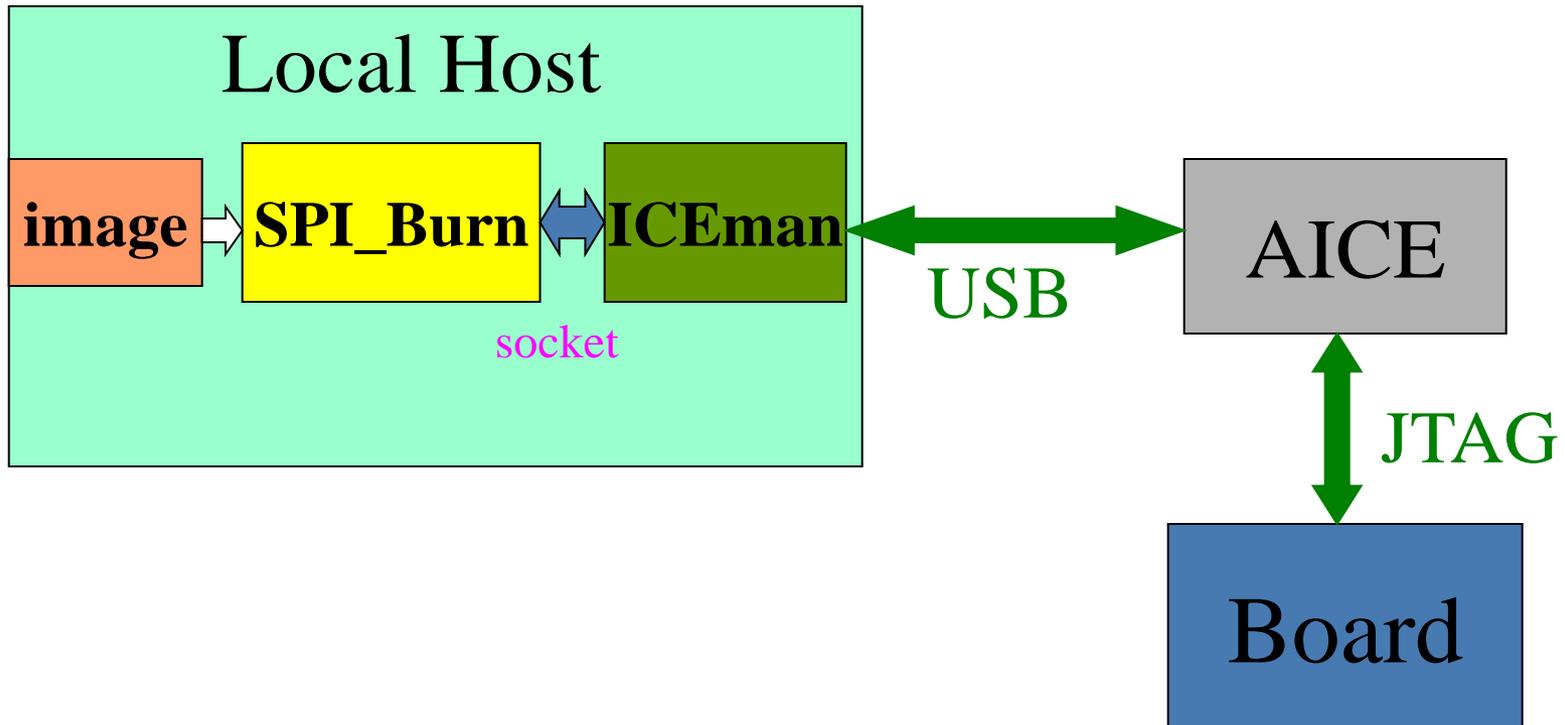
- ❖ Edit **"Makefile\_SPIburn\_win"** for toolchain command based on your environment.
- ❖ Build – **\$ make -f Makefile\_SPIburn\_win**
  - **"SPI\_burn.exe"** is generated.
- ❖ Clean – **\$ make -f Makefile\_SPIburn\_win clean**

```
Makefile_SPIburn_win x
1 #
2 # Makefile for Burner
3 #
4 BUILD_ID = ``date +%Y%m%d%H``
5
6 # For Windows platform with MinGW
7 #C++ = i386-mingw32-g++
8 #CC = i386-mingw32-gcc
9 C++ = g++
10 CC = gcc
11 RM = rm -f
12 #AR = i386-mingw32-ar
13 AR = ar
14 CFLAGS += -g3 -O2 -Wall -DSPI_BURN -DE
15 LIBS = -lwsock32 -lm -lgdi32 -luser32
```

```
/cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
$ make -f Makefile_SPIburn_win
g++ -g3 -O2 -Wall -DSPI_BURN -DBUILD_ID="`date +%Y%m%d%H`" -c -o /cygdrive/c/A
ndestech/AndeSight210STD/flash/src-SPI_burn_test/util.o /cygdrive/c/Andestech/An
deSight210STD/flash/src-SPI_burn_test/util.cpp
g++ -g3 -O2 -Wall -DSPI_BURN -DBUILD_ID="`date +%Y%m%d%H`" -c -o /cygdrive/c/A
ndestech/AndeSight210STD/flash/src-SPI_burn_test/main.o /cygdrive/c/Andestech/An
deSight210STD/flash/src-SPI_burn_test/main.cpp
g++ -g3 -O2 -Wall -DSPI_BURN -DBUILD_ID="`date +%Y%m%d%H`" -c -o /cygdrive/c/A
ndestech/AndeSight210STD/flash/src-SPI_burn_test/platform-ae210p.o /cygdrive/c/A
ndestech/AndeSight210STD/flash/src-SPI_burn_test/platform-ae210p.cpp
g++ -g3 -O2 -Wall -DSPI_BURN -DBUILD_ID="`date +%Y%m%d%H`" -c -o /cygdrive/c/A
ndestech/AndeSight210STD/flash/src-SPI_burn_test/spiflash-MXIC.o /cygdrive/c/And
estech/AndeSight210STD/flash/src-SPI_burn_test/spiflash-MXIC.cpp
g++ -static -o SPI_burn.exe -g3 -O2 -Wall -DSPI_BURN -DBUILD_ID="`date +%Y%m%d%H
`" /cygdrive/c/Andestech/AndeSight210STD/flash/src-SPI_burn_test/util.o /cygdr
ive/c/Andestech/AndeSight210STD/flash/src-SPI_burn_test/main.o /cygdrive/c/Andes
tech/AndeSight210STD/flash/src-SPI_burn_test/platform-ae210p.o /cygdrive/c/Andes
tech/AndeSight210STD/flash/src-SPI_burn_test/spiflash-MXIC.o -lwsock32 -lm -lgdi
32 -luser32 -lcfmgr32 -lsetupapi -lcomctl32
ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
$
```

# Socket connect to ICEman (1/3)

- ❖ Initialize socket and connect to host
  - Default connection host: 127.0.0.1(Local Host)
  - ICEman default open port 2354 for burner



# Socket connect to ICEman (2/3)

## ❖ Communication API

- `outw/outh/outb(address, data)`
  - ◆ Write word/half word/byte data to bus
- `inw/inh/inb(address)`
  - ◆ Read word/half word/byte data from bus
- `out_io(address, size, buffer)`
  - ◆ Write multi-word to same IO port address
- `fastin(address, size, buffer)`
  - ◆ Read multi-word data from bus by ICEman
- `fastout/fastin(address, size, buffer)`
  - ◆ Write/Read multi-word data from bus
- `multiout_w/ multiout_h/ multiout_b(address, data, num_of_pairs)`
  - ◆ write multi-words/multi-half word/multi-byte to different addresses by ICEman
- `multiin_w/ multiin_h/ multiin_b(address, data, num_of_pairs)`
  - ◆ read multi-words/multi-half word/multi-byte from different addresses by ICEman

# Socket connect to ICEman (3/3)

## ❖ Command code

- Andes-defined commands can be specified what action ICEman should do with the target or AICE.

- API

  - ◆ send\_cmd(char cmd): Send command to iceman

- These commands include:

  - ◆ RESET\_TARGET: Reset the target board

  - ◆ RESET\_HOLD: Reset the target board and make it stop at \$IVB, then **hold** the CPU before boot code executing.

  - ◆ RESET\_AICE: Reset AICE

# SPI Burner Flow

## ❖ Main (with macro : SPI\_BURN in Makefile)

- Parse parameters
- Initialize socket and connect to ICEman
- Initialize AICE-MCU
- Initialize target, get flash base address
- Check flash type(MXIC)
- Get burning image
- Verify Flash content with image if verify\_only is on.
- Unlock flash (optional)
- Preserve the first and last erase blocks (optional)
- Block erase
- Burn image
- Verify burning result (optional)
- Lock flash (optional)

# Required Parameters

❖ `--image=FILENAME`

■ Specify the image file to burn

# Optional Parameters (1/2)

- ❖ --host
  - Host name/ip to connect with ICEman (default: 127.0.0.1)
- ❖ --port
  - Specify the port number to connect with ICEman
- ❖ --preserve
  - Preserve the content in the first and last erase blocks
- ❖ --log
  - Specify the log file to store output message (default to stdout)
- ❖ --reset-target
  - Target reset
- ❖ --reset-hold
  - Target reset and stop at \$IVB
- ❖ --base
  - Specify the flash base address (default SPI base addr 0x00F0B000)
- ❖ --addr
  - Specify the flash target address to write, default write to flash base

# Optional Parameters (2/2)

- ❖ --verify
  - Verify after flash burning
- ❖ --verify-only
  - Only verify the content of Flash
- ❖ --version
  - Show flash burning version
- ❖ --unlock
  - Unlock flash before burning
- ❖ --lock
  - Lock flash after burning
- ❖ --erase-all
  - Erase entire flash before burning
- ❖ --measure-time
  - Estimate total time to burn image to ROM
- ❖ --core-num
  - Select CoreID
- ❖ --help
  - List the usages of the flash burner

# Target and Flash related Functions

- ❖ platform\_init()
  - Return SPI flash base address
- ❖ mxic\_chk()
  - Check MXIC rom ID
  - Unlock chip if necessary
- ❖ mxic\_erase()
- ❖ mxic\_program()
  - Burn data to flash
- ❖ mxic\_read()
- ❖ mxic\_lock()/mxic\_unlock()

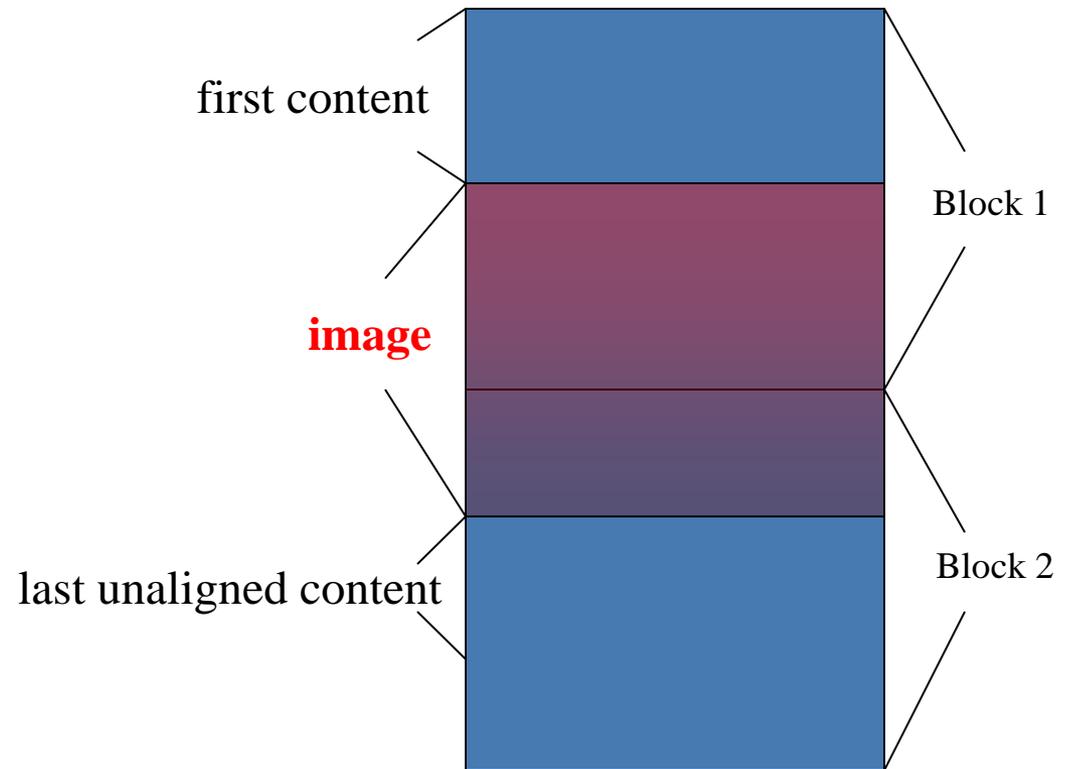
# Burn Image

## ❖ nds32\_burn()

- call gpFlash->flash\_program and finally call mxic\_program to do really flash burn
- burn one page at one time

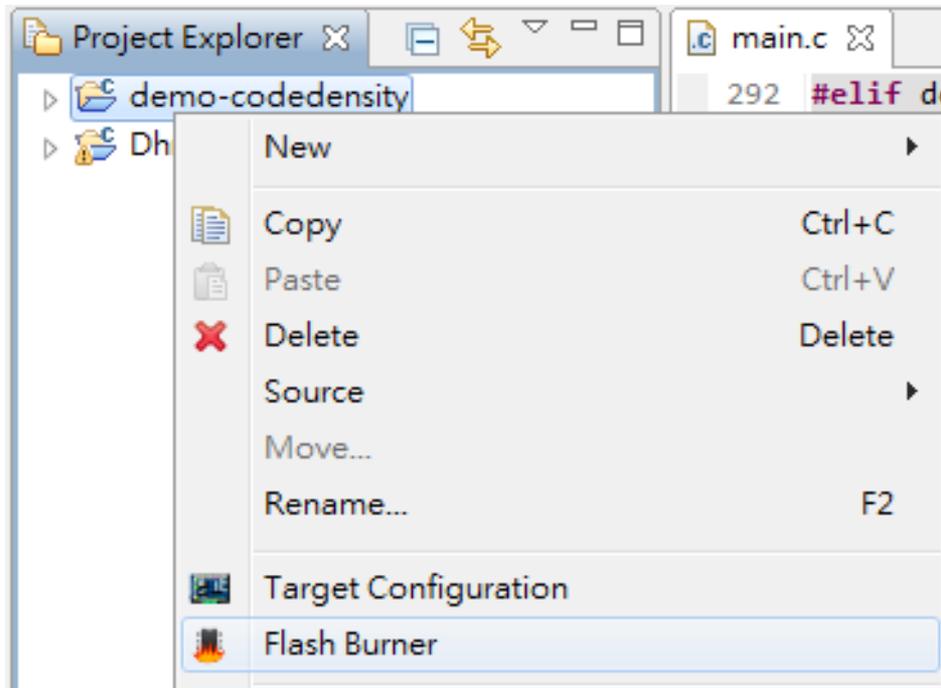
# Burn NOT Block-Aligned Data (opt.)

- ❖ In case that users want to burn the address which isn't block-aligned
- ❖ Preserve the first content and last unaligned content before erase block
- ❖ Burn back content after erase block



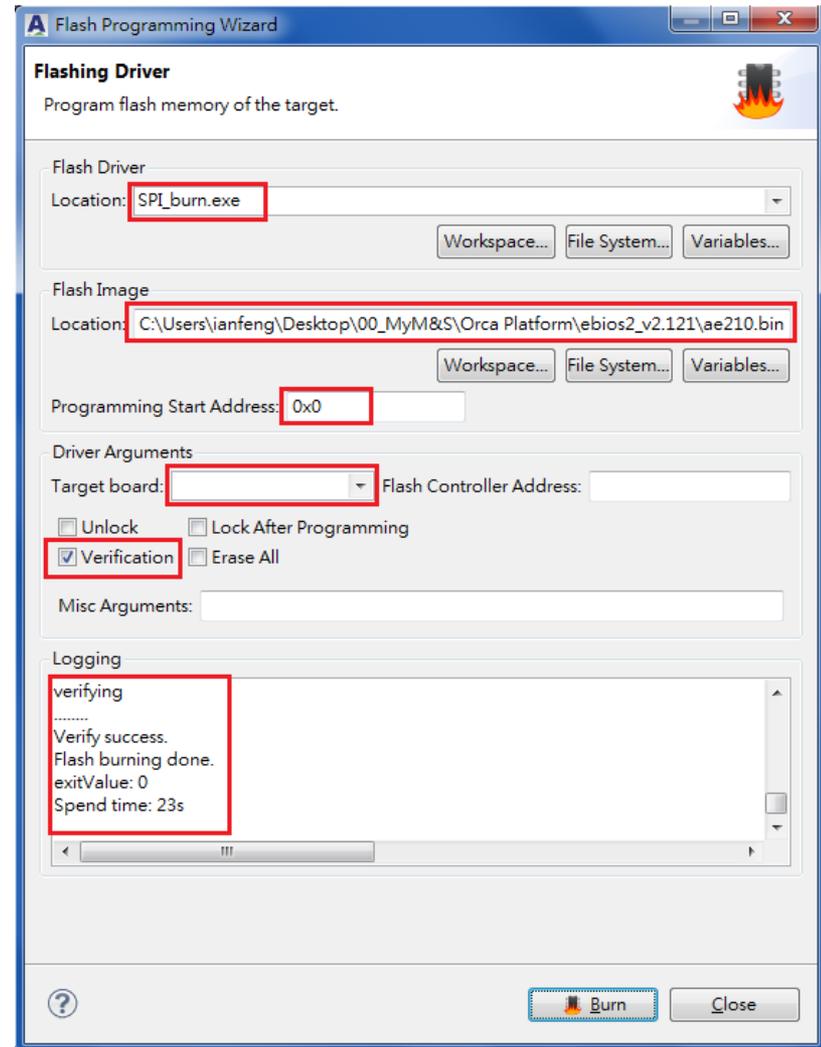
# Burn Boot Code in AndeSight (1/2)

1. Run AndeSight, and power-on ADP-XC7KFF676.
2. Connect AICE-MCU to PC and ADP-XC7KFF676.
3. Right-click a project, and select "Flash Burner".



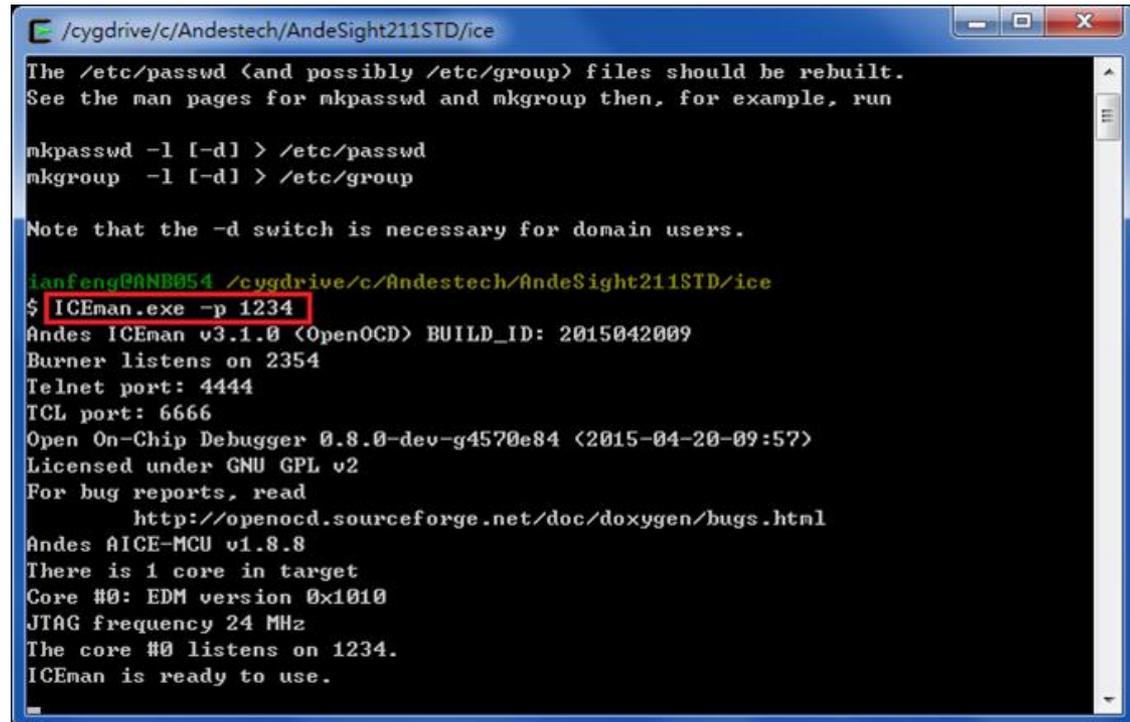
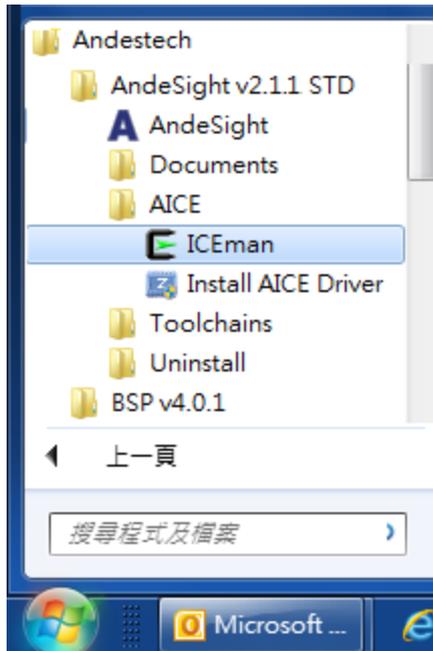
# Burn Boot Code in AndeSight (2/2)

4. Select "**SPI\_burn.exe**" in "Flashing Driver".
5. Select "Flashing Image".
6. Set "Start Address:" to 0x0.
7. Keep "Target board:" empty.
8. Select "Verification".
9. Press "Burn" button.



# Burn Boot Code in Cygwin (1/2)

- ❖ Open "Cygwin" environment.
- ❖ Run ICEman.



```

/cygdrive/c/Andestech/AndeSight211STD/ice
The /etc/passwd (and possibly /etc/group) files should be rebuilt.
See the man pages for mkpasswd and mkgroup then, for example, run

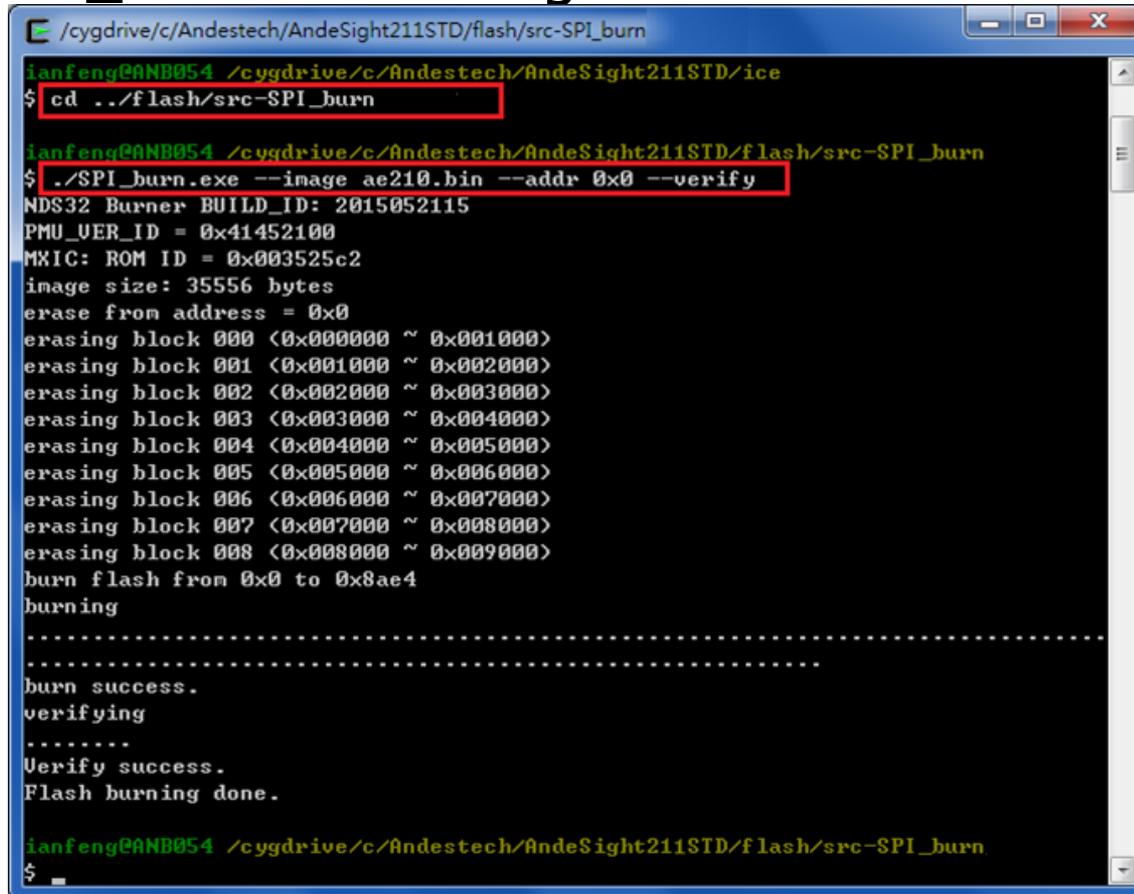
mkpasswd -l [-d] > /etc/passwd
mkgroup -l [-d] > /etc/group

Note that the -d switch is necessary for domain users.

ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/ice
$ ICEman.exe -p 1234
Andes ICEman v3.1.0 <OpenOCD> BUILD_ID: 2015042009
Burner listens on 2354
Telnet port: 4444
TCL port: 6666
Open On-Chip Debugger 0.8.0-dev-g4570e84 <2015-04-20-09:57>
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.sourceforge.net/doc/doxygen/bugs.html
Andes AICE-MCU v1.8.8
There is 1 core in target
Core #0: EDM version 0x1010
JTAG frequency 24 MHz
The core #0 listens on 1234.
ICEman is ready to use.
```

# Burn Boot Code in Cygwin (2/2)

- ❖ Open another “Cygwin” environment.
- ❖ Change to your SPI burner folder, and run it.
  - \$ `./SPI_burn.exe --image ae210.bin --addr 0x0 --verify`



```
ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
$ cd ../flash/src-SPI_burn
ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
$ ./SPI_burn.exe --image ae210.bin --addr 0x0 --verify
NDS32 Burner BUILD_ID: 2015052115
PMU_VER_ID = 0x41452100
MXIC: ROM ID = 0x003525c2
image size: 35556 bytes
erase from address = 0x0
erasing block 000 (0x000000 ~ 0x001000)
erasing block 001 (0x001000 ~ 0x002000)
erasing block 002 (0x002000 ~ 0x003000)
erasing block 003 (0x003000 ~ 0x004000)
erasing block 004 (0x004000 ~ 0x005000)
erasing block 005 (0x005000 ~ 0x006000)
erasing block 006 (0x006000 ~ 0x007000)
erasing block 007 (0x007000 ~ 0x008000)
erasing block 008 (0x008000 ~ 0x009000)
burn flash from 0x0 to 0x8ae4
burning
.....
burn success.
verifying
.....
Verify success.
Flash burning done.
ianfeng@ANB054 /cygdrive/c/Andestech/AndeSight211STD/flash/src-SPI_burn
$
```



- ❖ Andes **e-service** is a web-based supporting ticket system. It is a convenient way to get quick response for your questions.
- ❖ How to apply for a new account?
  - Please send your information to [es.admin@andestech.com](mailto:es.admin@andestech.com)
  - ◆ Include your **name, e-mail, company name and telephone number**

# How to Use E-service?

- E-service website: <http://es.andestech.com/ilogin.php>

The screenshot shows the Andes Embedded World website. The navigation menu includes 'About Andes', 'Products Showroom', 'Partnership', 'Technical Support', 'Download', 'Licensees', 'News/Events', and 'Contact Us'. The 'e-service' option is highlighted in the 'Technical Support' dropdown menu. A red box with the text 'Click e-service to log in' points to this option. Below the main banner, there are sections for 'Andes-Embedded World™' and 'Product'. The bottom part of the screenshot shows the 'SUPPORT TICKET SYSTEM' interface, which includes a navigation bar with 'New Ticket', 'My Tickets', 'Group Tickets', 'Change Password', and 'Log Out'. A search bar is present with the text 'Query: [ ] Search'. Below the search bar, there are buttons for 'View Open', 'View Closed', and 'Refresh'. The main content area displays a table of tickets with columns for Ticket #, Created on, Status, Subject, IPS staff, and Customer Full Name.

Ticket #	Created on	Status	Subject	IPS staff	Customer Full Name
1005	2011/12/26	Assigned	test ticket to check the content	Anthony Liao	pryn liao
997	2011/12/22	Assigned	測試 TICKET	Anthony Liao	pryn liao
996	2011/12/22	Assigned	abc test	Anthony Liao	pryn liao
995	2011/12/22	Assigned	test ticket	Anthony Liao	pryn liao

You can track your ticket  
in e-service system