DSH-G300
Smart Hub

Manual

Version 1.0

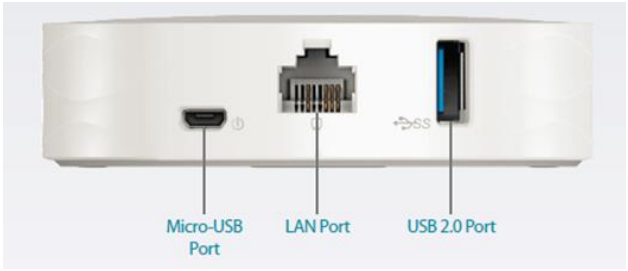
Dec 7th, 2017

Table of Contents

1. PRODUCT DESCRIPTION

The DSH-G300 is a ZigBee to Ethernet gateway device that uses a generic hardware design and can be deployed as different versions to fulfil different applications.

2. APPEARANCE



Top

- LED: Status indicator
- Wireless button: pairing/un-pairing

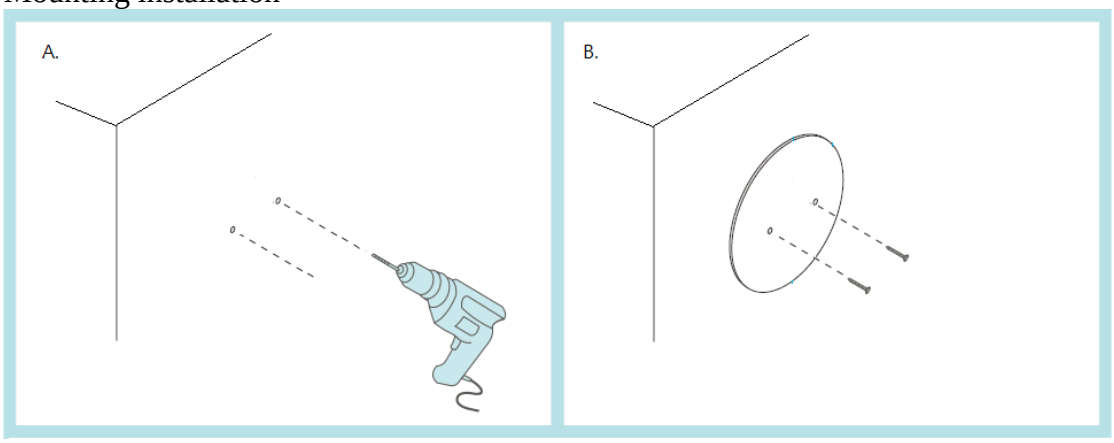
Rear

- LAN port
- Micro USB: Power jack, Vdc ⎓ : 5V/2.A
- USB port

Bottom

- Reset button

Mounting installation



3. INSTALLATIONS & CONFIGURATIONS

how to build firmware by GPL

Install tools

Install fedora linux 22, and make sure you can connect to internet.

Run command as below (ps : switch to root permission):

```
#yum -y update
#yum -y install gcc
#yum -y install zlib-devel openssl-devel
#yum -y install gcc-c++
#yum -y install bison
#yum -y install flex
#yum -y install ncurses-devel
#yum -y install fakeroot
#yum -y install patch
```

The workstation we used to build the image is:
OS Version: Fedora 22 (32-bit version on AMD or Intel system)
OS Kernel Version: 4.1.6-200.fc22.i686
GCC Version: 5.3.1 20160406 (Red Hat 5.3.1-6)
(You may use other Linux distribution as your workstation but

no guarantee of a successful build.)

Setup Build Enviornment($means command)

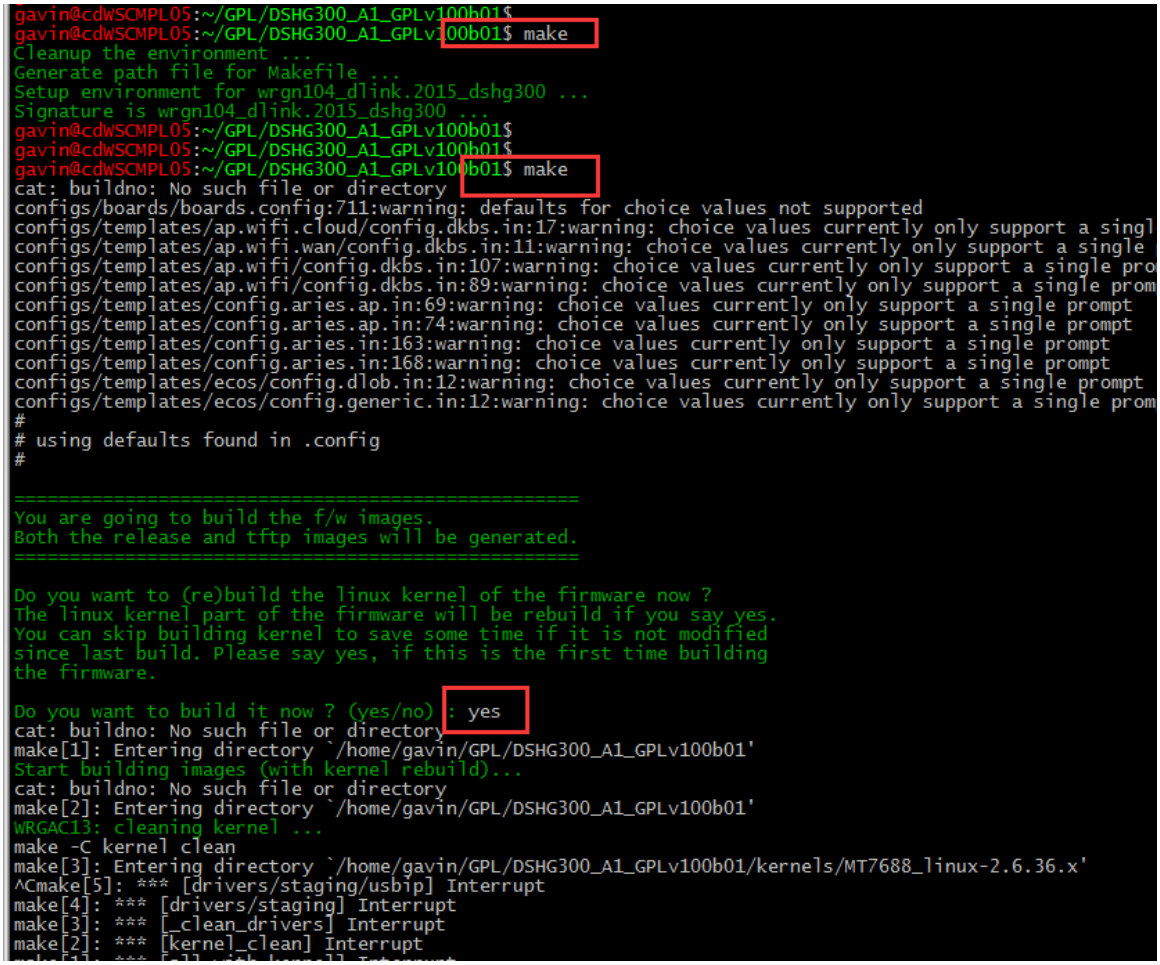
1) please login as a normal user such as john, and copy the gpl file to normal user folder, such as the folder /home/john
2) $cd /home/john
3) $tar zxvf DSHG300_A1_GPLv100b01.tar.gz
4) $cd DSHG300_A1_GPLv100b01
5) $su (ps : switch to root permission)
6) #mkdir /opt (if "/opt" is already exist, skip this step)
7) #cp -rf buildroot-gcc463 /opt
8) #exit (ps : switch back to normal user permission)
9) $source ./setupenv

Building the image

```
$make
```

```
gavin@cdWSCMPL05:~/GPL/DSHG300_A1_GPLv100b01$
gavin@cdWSCMPL05:~/GPL/DSHG300_A1_GPLv100b01$ make
Building config tool ...
make[1]: Entering directory `/home/gavin/GPL/DSHG300_A1_GPLv100b01/configs/config'
cp zconf.tab.h_shipped zconf.tab.h
gcc -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer -I. -c conf.c -o conf.o
conf.c: In function 'conf_askvalue':
conf.c:94:15: warning: variable 'dummy' set but not used [-Wunused-but-set-variable]
conf.c: In function 'conf_string':
conf.c:164:20: warning: variable 'help' set but not used [-Wunused-but-set-variable]
conf.c: In function 'conf_sym':
conf.c:198:6: warning: variable 'type' set but not used [-Wunused-but-set-variable]
conf.c: In function 'conf_choice':
```

```
$make
$make
==================================================
You are going to build the f/w images.
Both the release and tftp images will be generated.
==================================================
Do you want to build it now ? (yes/no) : yes
```



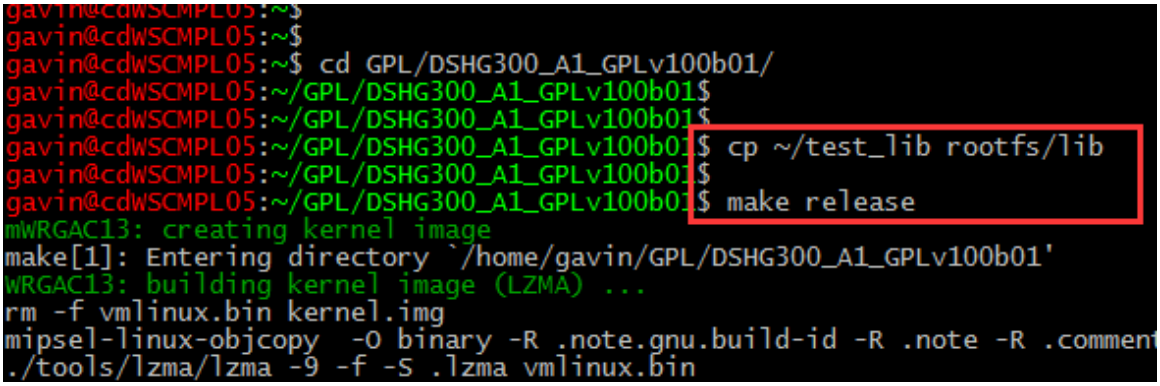
4) After make successfully, you will find the image file in ./images/.

- Tips: If there are some options need to be selected, please input "enter" key to execute the default action.

How to install file into firmware

Copy file into rootfs then rebuild gpl firmware by “make release”.

```
$cd DSHG300_A1_GPLv100b01
$cp ~/test_lib rootfs/lib
$make release
```



- Tips: If there is not rootfs, please using full steps to build firmware firstly.

How to add a mtd partition enable jffs2 support in kernel config

Before this step, please using full steps to build firmware

firstly.

Run kernel menuconfig and select jffs2 option, then save it.

```
$ cd DSHG300_A1_GPLv100b01/kernels/MT7688_linux-2.6.36.x
$ make menuconfig
```

```
File systems  --->
    Miscellaneous filesystems  --->
        <*> Journalling Flash File System v2 (JFFS2) support
        (0)   JFFS2 debugging verbosity (0 = quiet, 2 = noisy)
        [*]   JFFS2 write-buffering support
        [ ]   JFFS2 summary support (EXPERIMENTAL)
        [ ]   JFFS2 XATTR support (EXPERIMENTAL)
        [ ]   Advanced compression options for JFFS2
        <*> Compressed ROM file system support (cramfs)
```

```
--- Miscellaneous filesystems
< >   ADFS file system support (EXPERIMENTAL)
< >   Amiga FFS file system support (EXPERIMENTAL)
< >   Apple Macintosh file system support (EXPERIMENTAL)
< >   Apple Extended HFS file system support
< >   BeOS file system (BeFS) support (read only) (EXPERIMENTAL)
< >   BFS file system support (EXPERIMENTAL)
< >   EFS file system support (read only) (EXPERIMENTAL)
<*>   Journalling Flash File System v2 (JFFS2) support
(0)     JFFS2 debugging verbosity (0 = quiet, 2 = noisy)
[*]     JFFS2 write-buffering support
[ ]       Verify JFFS2 write-buffer reads
[ ]     JFFS2 summary support (EXPERIMENTAL)
[ ]     JFFS2 XATTR support (EXPERIMENTAL)
[*]     Advanced compression options for JFFS2
[*]       JFFS2 ZLIB compression support
[ ]       JFFS2 LZO compression support
[ ]       JFFS2 LZMA compression support
[*]       JFFS2 RTIME compression support
[ ]       JFFS2 RUBIN compression support
          JFFS2 default compression mode (priority)  --->
< >   LogFS file system (EXPERIMENTAL)
< >   Compressed ROM file system support (cramfs)
<*>   SquashFS 4.0 - Squashed file system support
[ ]     Squashfs XATTR support
[*]     Include support for ZLIB compressed file systems
[ ]     Include support for LZO compressed file systems
```

Copy kernel config into boards folder.

```
$ cp .config ../../boards/wrgn104/kernel.config
```

```
gavin@cdWSCMPL05:~/GPL/DSHG300_A1_GPLv100b01/kernels/MT7688_linux-2.6.36.x$
gavin@cdWSCMPL05:~/GPL/DSHG300_A1_GPLv100b01/kernels/MT7688_linux-2.6.36.x$ cp .config ../../boards/wrgn104/kernel.config
cp: overwrite `../../boards/wrgn104/kernel.config'? y
gavin@cdWSCMPL05:~/GPL/DSHG300_A1_GPLv100b01/kernels/MT7688_linux-2.6.36.x$
gavin@cdWSCMPL05:~/GPL/DSHG300_A1_GPLv100b01/kernels/MT7688_linux-2.6.36.x$
```

add mtd partition

Modify mtd partition map in file:
DSHG300_A1_GPLv100b01/kernels/MT7688_linux-2.6.36.x/drivers/mtd/ralink/ralink_bbu_spi.c

This is a sample which add a mtd partition named by “test”,

ralink_bbu_spi.c

and size of this partition is 320k:

Re-bulid gpl firmware and upgrade gpl firmware into device.

```
94
95 #define BOOTCODE_SIZE    0x30000
96 #define FACTORY_SIZE    0x10000
97 #define DEVCONF_SIZE    0x10000
98 #define DEVDATA_SIZE    0x10000
99 #define DATA_SIZE       0x500000
100 #define UPGRADE_SIZE    (FLASH_SIZE - BOOTCODE_SIZE -FACTORY_SIZE- DEVCONF_SIZE - DEVDATA_SIZE - LANGPACK_SIZE - MYDLINK_SIZE - DATA_SIZE)
101
102 #define DEVDATA_OFFSET  (BOOTCODE_SIZE)
103 #define FACTORY_OFFSET     (BOOTCODE_SIZE + DEVDATA_SIZE)
104 #define UPGRADE_OFFSET  (BOOTCODE_SIZE + DEVDATA_SIZE + FACTORY_SIZE)
105
106 #define DEVCONF_OFFSET  (BOOTCODE_SIZE + DEVDATA_SIZE+FACTORY_SIZE+  UPGRADE_SIZE)
107
108 #define MYDLINK_OFFSET  (BOOTCODE_SIZE + DEVDATA_SIZE+FACTORY_SIZE + UPGRADE_SIZE +DEVCONF_SIZE)
109
110 #define DATA_OFFSET     (BOOTCODE_SIZE + DEVDATA_SIZE+FACTORY_SIZE + UPGRADE_SIZE +DEVCONF_SIZE + MYDLINK_SIZE)
111
112 #define LANGPACK_OFFSET (BOOTCODE_SIZE + DEVDATA_SIZE+FACTORY_SIZE + UPGRADE_SIZE +DEVCONF_SIZE + MYDLINK_SIZE + DATA_SIZE)
113
114 static struct mtd_partition elbox_partitions[] =
115 {
116         /* The following partitions are the "MUST" in ELBOX. */
117         {name:"rootfs",         offset:0,                                       size:0,                         mask_flags:MTD_WRITEABLE, },
118         {name:"upgrade",        offset:UPGRADE_OFFSET,  size:UPGRADE_SIZE,      },
119         {name:"devconf",        offset:DEVCONF_OFFSET,  size:DEVCONF_SIZE,      },
120         {name:"devdata",        offset:DEVDATA_OFFSET,  size:DEVDATA_SIZE,      },
121         {name:"langpack",       offset:LANGPACK_OFFSET, size:LANGPACK_SIZE,     },
122         {name:"data",           offset:DATA_OFFSET,                  size:DATA_SIZE, },
123         {name:"flash",          offset:0,                                       size:FLASH_SIZE,        mask_flags:MTD_WRITEABLE, },
124
125         /* The following partitions are board dependent. */
126         {name:"u-boot",         offset:0,                                       size:BOOTCODE_SIZE,     mask_flags:MTD_WRITEABLE, },
127         {name:"boot env",       offset:DEVDATA_OFFSET,  size:0x8000,                  mask_flags:MTD_WRITEABLE, },
128         {name:"Factory",        offset:FACTORY_OFFSET,  size:FACTORY_SIZE},
129 #ifdef CONFIG_MTD_ELBOX_MYDLINK
130         {name:"mydlink",        offset:MYDLINK_OFFSET,  size:MYDLINK_SIZE,      }
131 #endif
132 };
```

mount mtd partition

After firmware upgrade, show all mtd partitions by below command:

```
$ cat proc/mtd
```

```
# cat /proc/mtd
dev:    size   erasesize  name
mtd0: 01000000 00010000 "raspi"
mtd1: 00822000 00010000 "rootfs"
mtd2: 00a00000 00010000 "upgrade"
mtd3: 00010000 00010000 "devconf"
mtd4: 00010000 00010000 "devdata"
mtd5: 00020000 00010000 "langpack"
mtd6: 00500000 00010000 "data"
mtd7: 01000000 00010000 "flash"
mtd8: 00030000 00010000 "u-boot"
mtd9: 00008000 00010000 "boot env"
mtd10: 00010000 00010000 "Factory"
mtd11: 00080000 00010000 "mydlink"
#
```

For this example, “data” mtd partition is mtd6 , it’s means that dev path of “data” mtd partition is /dev/mtdblock/6.

Use below command to mount mtd partition “test” with jffs2 type.

```
$ mount -t jffs2 /dev/mtdblock/6 /data
```

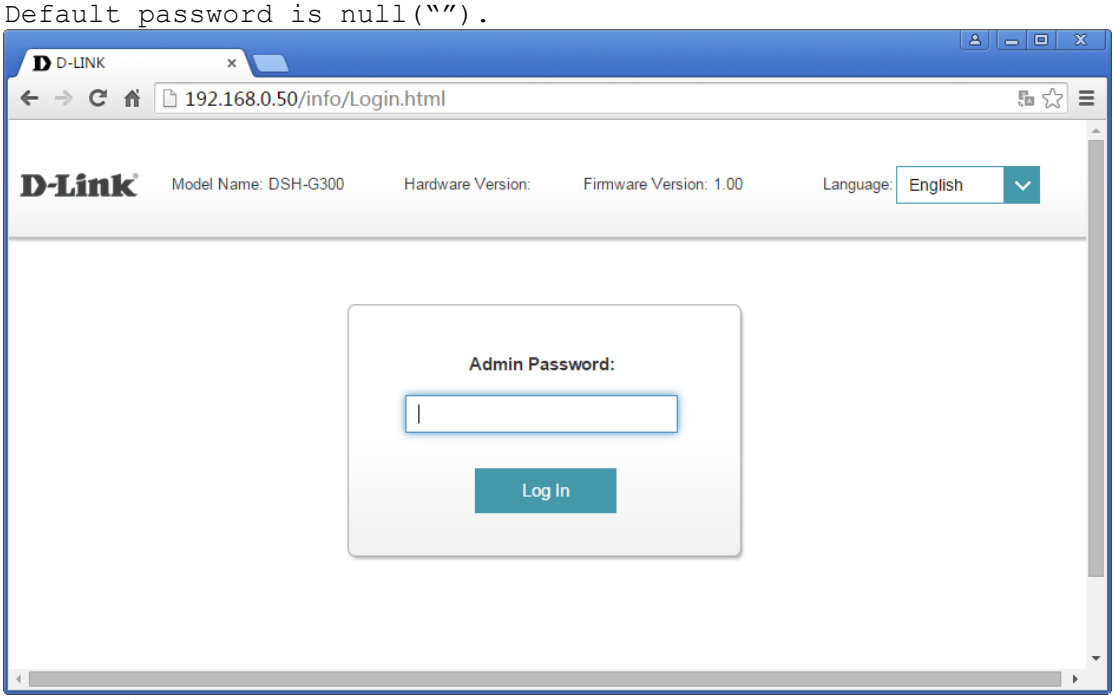
- Tips1: mtd partition must larger than 320k.
- Tips2: we can add mtd partition by reduce “upgrade” mtd partition size, but upgrade is used to upgrade firmware. So we must make sure size of “upgrade” mtd partition always larger than firmware size.
- Tips3: in gpl v100b04, we auto mount the ”data” mtd partition to /data while device boot up.

```
#
# mount
rootfs on / type rootfs (rw)
/dev/root on / type squashfs (ro,relatime)
devtmpfs on /dev type devtmpfs (rw,relatime,size=62464k,nr_inodes=15616,mode=755)
none on /proc type proc (rw,relatime)
ramfs on /var type ramfs (rw,relatime)
sysfs on /sys type sysfs (rw,relatime)
tmpfs on /dev type tmpfs (rw,relatime)
devpts on /dev/pts type devpts (rw,relatime,mode=600)
/dev/mtdblock/6 on /data type jffs2 (rw,relatime)
#
```

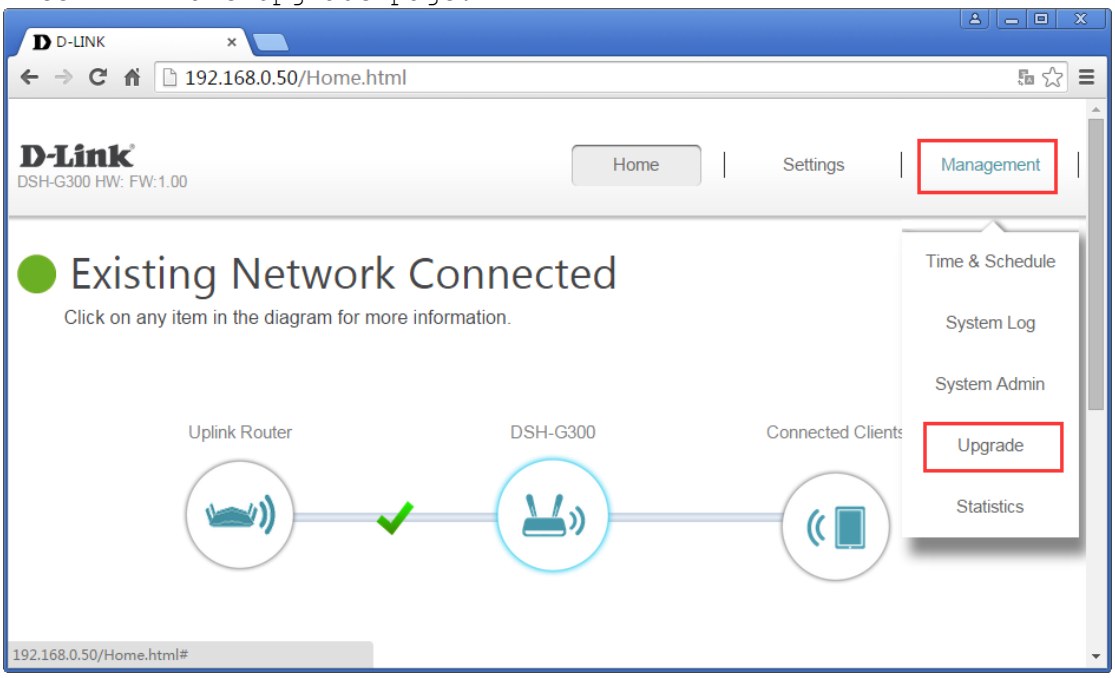
How to upgrade firmware by web

Enter URL http://192.168.0.50/ and login it.

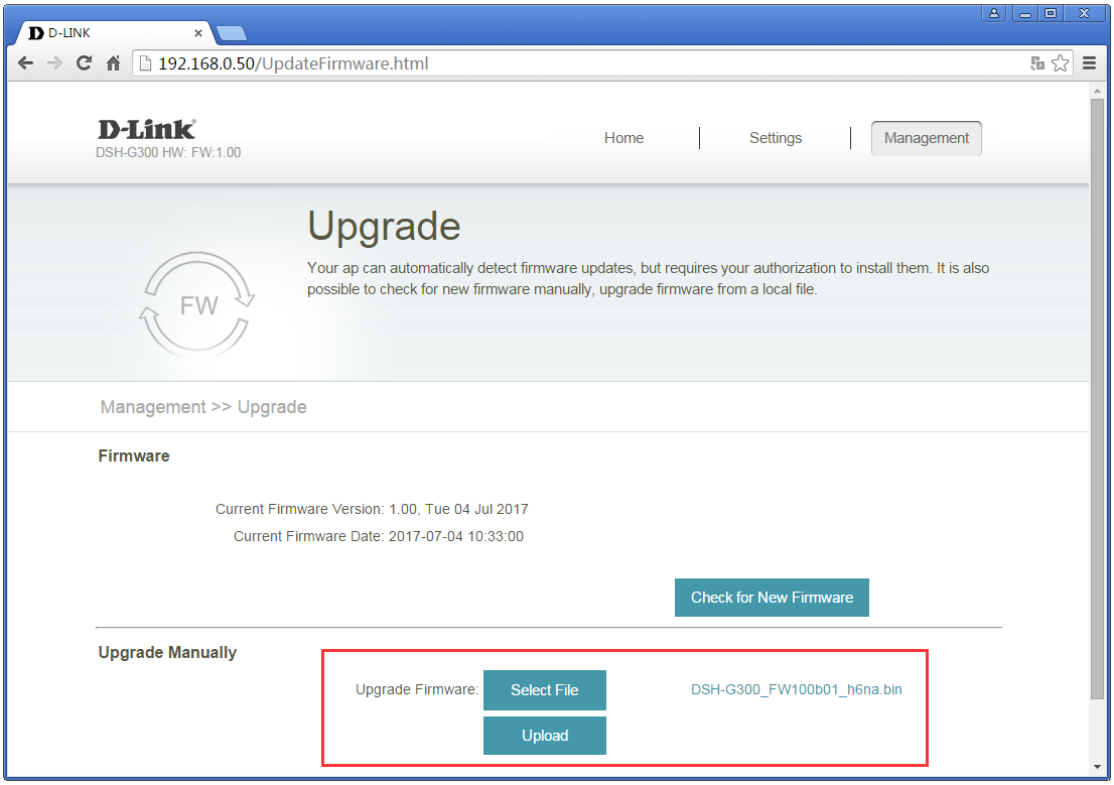
Default password is null(“”).



Enter firmware upgrade page.



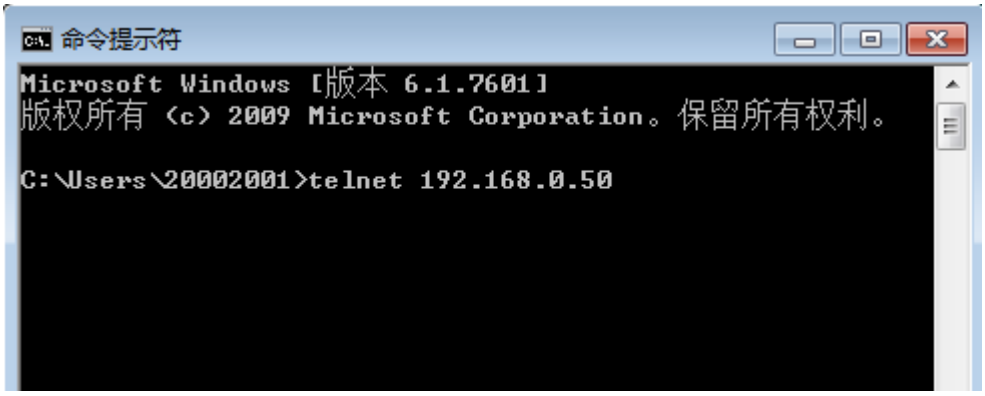
Select firmware file and press Upload button.



How to use telnet and tftp

How to use telnet

Telnet is enable as default,use tool to access it. Example is Microsoft cmd tool. Device default ip is 192.168.0.50.



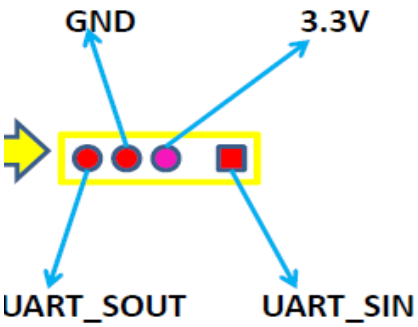
How to use tftp

Tftp can transfer a file from/to tftp server.
This is an exanple(tftp server ip is 192.168.0.38).

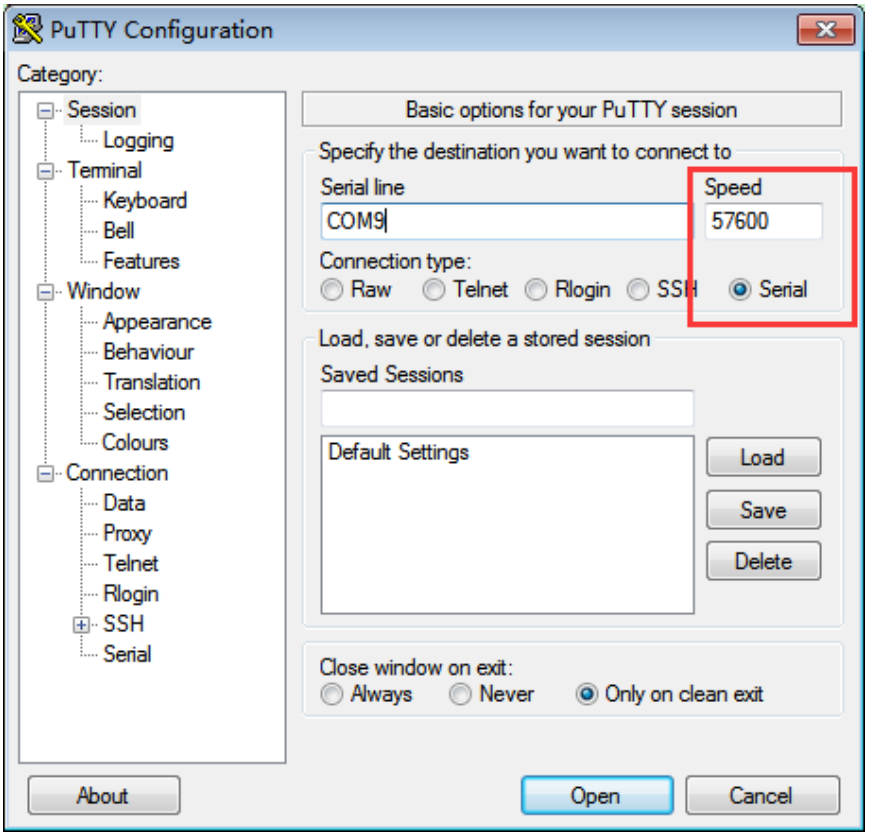
```
$ #get file from tftp server
$ tftp -r get_file.txt -l get_file.txt -g 192.168.0.38

$ #put file from tftp server
$ tftp -r put_file.txt -l put _file.txt -g 192.168.0.38
```

```
#
# rm -rf /var/test/
#
#
#
# mkdir /var/test
# cd /var/test/
# ls
#
# tftp -r get_file.txt -l get_file.txt -g 192.168.0.38
#
# ls
get_file.txt
#
# cat get_file.txt
aaaaaaaaaaaaa#
#
```

Console define



Console connection type is Serial and speed is 57600



FCC Statement:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communication. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

FCC Caution:

Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

(1) This device may not cause harmful interference, and

(2) this device must accept any interference received, including interference that may cause undesired operation.

If this device is going to be operated in 5.15 ~ 5.25GHz frequency range, then it is restricted in indoor environment only.

IMPORTANT NOTICE:

FCC Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

The availability of some specific channels and/or operational frequency bands are country dependent and are firmware programmed at the factory to match the intended destination. The firmware setting is not accessible by the end user.