DS3 Manual

Prepared for: Linino SW team

Prepared by: Federico Musto (federico@linino.org)

Thursday, August 1, 2013

Brand Name: dog hunter

Model Name:DHQAR-W03

Product Name:dogstick

Version: 0.4b

Revision History

Date / Version	Name-Function	Note			
7/31/13 / 0.1	FM	First Version			

Dog Hunter LLC
8 Faneuil Hall
Boston, MA 02109 USA
T (978) 631-0369

Dog Hunter AG
Baarerstrasse 14
6300 Zug Switzerland

Dog Hunter Inc. (多航有限公司)
14th Floor, No. 8, Sec. 5, Xinyi Rd
Taipei Taiwan 11049

HW specification

AR9331 Highly-Integrated and Cost Effective IEEE 802.11n - 1x1 2.4 GHz SoC for AP and Router Platforms

Features

- Complete IEEE 802.11n 1x1 AP or router in a single chip
- MIPS 24K processor operating at up to 400 MHz
- 64MB DDR2 memory
- 16MB SPI Flash memory
- 1 WAN port
- External RF connector
- Internal Antenna
- USB connector /High-speed UART for console support (4.75~ 5.25V Input)
- USB 2.0 host/device mode support
- GPIO/LED support –with Voltage Level Translator
- 5V Input (4.75~ 5.25V)
- POE Function support (48V Input ,5V/2A output)
- Micro SD card support
- Expansion IO compatible with the Arduino Leonardo

Summary Linino image

How to build environment and first image

The standard build environment for linino images is:

Debian squeeze amd64

Using a chroot´ed environment will also work.

This tutorial does not explain how to install Debian or perform sysadmin tasks.

Make sure updates and security-updates are enabled and installed.

As root (or via sudo):

```
# apt-get update && apt-get upgrade
```

```
# apt-get install git subversion build-essential python gawk unzip
libncurses5-dev libz-dev fastjar asciidoc flex libgtk2.0-dev intltool
perl-modules python2.6-dev rsync ruby unzip wget gettext xsltproc tex4ht
texlive-lang-french
```

On Ubuntu 12.10 LTS (quantal)

```
#sudo apt-get install git subversion build-essential python gawk
unzip libncurses5-dev zlib1g-dev fastjar asciidoc flex libgtk2.0-dev
intltool perl-modules python-dev rsync ruby unzip wget gettext
xsltproc tex4ht texlive-lang-french
```

As **normal use**r: (no root)

Get the source - see https://dev.openwrt.org/wiki/GetSource:
any other method is also fine, I prefer git, you might have to adapt the following commands.

Once linino branch is created, we will replace the checkout with the official repo.

```
$ git clone git://nbd.name/openwrt.git
$ git clone git://nbd.name/packages.git
$ cd openwrt
```

Old way

```
$ git clone git://nbd.name/openwrt.git
$ cd openwrt/trunk
```

Download the package feeds and install them:

```
$ ./scripts/feeds update -a
$ ./scripts/feeds install -a
```

Create the default config and perform a basic check for installed packages required to build:

```
$ make defconfig
$ make prereq
```

"make defconfig" will generate linino custom config by default.

Customize the build (optional for first time running):

```
$ make menuconfig
```

"make menuconfig" will be unnecesasry in linino branch.

Build (use of -j $number_of_cores is highly recommended):

```
$ make
```

build will download and compile everything with defconfig. It takes > 7GB of disk space and a lot of bw/cpu to perform this task.

Results will be stored here if build is completed:

```
$ cd bin/ar71xx/
```

```
$ ls *3020*
```

ignore jffs2 images for now.
squashfs images come in two forms:
factory.bin - to be used for the very first time when switching from factory firmware to openwrt firmware.
sysupgrade.bin - to be used to perform upgrades from openwrt to openwrt.

Build the documentation:

```
$ make -C docs/
```

OpenWRT documentation has an incredibly simple HowTO build packages and feeds. I verified that the contents it´s actually correct and those are the URLs:

http://wiki.openwrt.org/doc/devel/packages

http://wiki.openwrt.org/doc/devel/feeds#creating.your.own.feed

Tons of examples can be found in

```
$ ~/openwrt/root/trunk/feeds$ pwd/home/wrt/openwrt/root/trunk/feeds
```

(based on your checkout path/directory of course)

Simple package can be: ipv6calc

```
$ ~/openwrt/root/trunk/feeds$ find . -name "*6calc*"
./packages/ipv6/ipv6calc
```

Summary GPIO JP1

Pin definitions

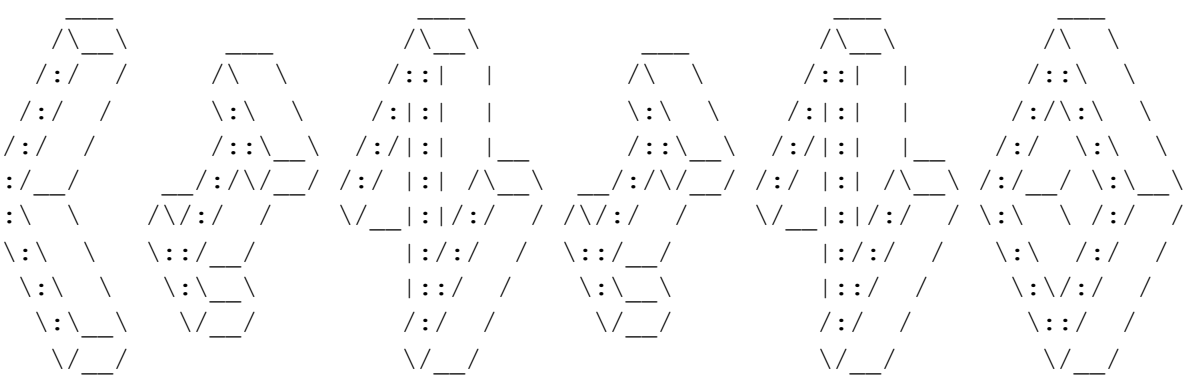
PIN #	Definition	GPIO over SPI using AVRDUDE
PIN 1	VDD33	
PIN 2	VDD33	
PIN 3	GND	
PIN 4	GND	
PIN 5	GPIO20	RESET
PIN 6	GPIO19	SCK
PIN 7	GPIO22	MOSI
PIN 8	GPIO18	MISO
PIN 9	GND	
PIN 10	LAN_LED	

How to activate a GPIO

To enable GPIO on linino.org board

```
homepsds-Mac-mini:~ federico$ ssh root@192.168.1.1
root@192.168.1.1's password: doghunter

BusyBox v1.19.4 (2013-01-04 11:35:48 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
```



```
 _______                     ________        __
|       |.-----.-----.-----.|  |  |  |.----.|  |_
|   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
|_______||   __|_____|__|__||________||__|  |____|
         |__| W I R E L E S S   F R E E D O M
-----------------------------------------------------
 BARRIER BREAKER (Bleeding Edge, r35001)
-----------------------------------------------------
  * 1/2 oz Galliano         Pour all ingredients into
  * 4 oz cold Coffee        an irish coffee mug filled
  * 1 1/2 oz Dark Rum       with crushed ice. Stir.
  * 2 tsp. Creme de Cacao
-----------------------------------------------------
root@linino:~#

root@linino:~# cd /sys/class/gpio/

root@linino:/sys/class/gpio# ls -l
--w-------    1 root     root          4096 Jan  1 00:00 export
lrwxrwxrwx    1 root     root             0 Jan  1 00:00 gpiochip0 -> (...)virtual/gpio/gpiochip0
--w-------    1 root     root          4096 Jan  1 00:00 unexport

root@linino:/sys/class/gpio# echo 20 > export

root@linino:/sys/class/gpio# ls -l
--w-------    1 root     root          4096 Jan  1 00:07 export
lrwxrwxrwx    1 root     root             0 Jan  1 00:07 gpio20 -> (...)virtual/gpio/gpio20
lrwxrwxrwx    1 root     root             0 Jan  1 00:00 gpiochip0 -> (...)virtual/gpio/gpiochip0
--w-------    1 root     root          4096 Jan  1 00:00 unexport

root@linino:/sys/class/gpio# cd gpio20/
```

```
root@linino:/sys/devices/virtual/gpio/gpio20# ls -l
-rw-r--r--    1 root     root    4096 Jan  1 00:16 active_low
-rw-r--r--    1 root     root    4096 Jan  1 00:16 direction
lrwxrwxrwx    1 root     root       0 Jan  1 00:16 subsystem -> ../../../../class/gpio
-rw-r--r--    1 root     root    4096 Jan  1 00:16 uevent
-rw-r--r--    1 root     root    4096 Jan  1 00:16 value

root@linino:/sys/devices/virtual/gpio/gpio20# cat direction
in

root@linino:/sys/devices/virtual/gpio/gpio20# cat value
0

root@linino:/sys/devices/virtual/gpio/gpio20# echo out > direction

root@linino:/sys/devices/virtual/gpio/gpio20# cat direction
out

root@linino:/sys/devices/virtual/gpio/gpio20# cat value
0

root@linino:/sys/devices/virtual/gpio/gpio20# echo 1 > value



root@linino:/sys/devices/virtual/gpio/gpio20# echo 0 > value
```

How to use AVRDUDE using GPIO

```
# avrdude -c linuxgpio -v -C ./avrdude.conf -p m328p -U flash:w:Blink.cpp.hex
```

```
avrdude: Version 5.11svn, compiled on Feb 26 2013 at 23:31:49
        Copyright (c) 2000-2005 Brian Dean, http://www.bdmicro.com/
        Copyright (c) 2007-2009 Joerg Wunsch

        System wide configuration file is "./avrdude.conf"
        User configuration file is "/root/.avrduderc"
        User configuration file does not exist or is not a regular file, skipping

        Using Port                    : unknown
        Using Programmer              : linuxgpio
        AVR Part                      : ATmega328P
        Chip Erase delay              : 9000 us
        PAGEL                         : PD7
        BS2                           : PC2
        RESET disposition             : dedicated
        RETRY pulse                   : SCK
        serial program mode           : yes
        parallel program mode         : yes
        Timeout                       : 200
        StabDelay                     : 100
        CmdexeDelay                   : 25
        SyncLoops                     : 32
        ByteDelay                     : 0
        PollIndex                     : 3
        PollValue                     : 0x53
        Memory Detail                 :
```

```
                                Block Poll Page                       Polled
          Memory Type Mode Delay Size  Indx Paged  Size   Size #Pages MinW  MaxW   ReadBack
          ----------- ---- ----- ----- ---- ------ ------ ---- ------ ----- ----- ---------
          eeprom        65    20     4    0 no       1024    4 0  3600  3600 0xff 0xff
          flash         65     6   128    0 yes     32768  128 256  4500  4500 0xff 0xff
          lfuse          0     0     0    0 no          1    0 0  4500  4500 0x00 0x00
          hfuse          0     0     0    0 no          1    0 0  4500  4500 0x00 0x00
          efuse          0     0     0    0 no          1    0 0  4500  4500 0x00 0x00
          lock           0     0     0    0 no          1    0 0  4500  4500 0x00 0x00
          calibration    0     0     0    0 no          1    0 0     0     0 0x00 0x00
          signature      0     0     0    0 no          3    0 0     0     0 0x00 0x00

        Programmer Type : linuxgpio
        Description     : Use the Linux sysfs interface to bitbang GPIO lines

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e950f
avrdude: safemode: lfuse reads as FF
avrdude: safemode: hfuse reads as D6
avrdude: safemode: efuse reads as 5
avrdude: NOTE: "flash" memory has been specified, an erase cycle will be performed
        To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "Blink.cpp.hex"
avrdude: input file Blink.cpp.hex auto detected as Intel Hex
avrdude: writing flash (1072 bytes):

Writing | ################################################## | 100% 0.58s

avrdude: 1072 bytes of flash written
avrdude: verifying flash memory against Blink.cpp.hex:
avrdude: load data flash data from input file Blink.cpp.hex:
avrdude: input file Blink.cpp.hex auto detected as Intel Hex
avrdude: input file Blink.cpp.hex contains 1072 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 1.08s

avrdude: verifying ...
avrdude: 1072 bytes of flash verified
```

```
avrdude: safemode: lfuse reads as FF
avrdude: safemode: hfuse reads as D6
avrdude: safemode: efuse reads as 5
avrdude: safemode: Fuses OK

avrdude done.  Thank you.
```

Appendix

Setting Up An SSH Certificate

If you want to SSH login without password or automate your task between two servers, you need to setup SSH login via certificate.

Step1: Generating RSA key

You login to your server and type the following command:

```
[root@linino~]#ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
a5:4c:29:3f:92:be:ee:41:03:8d:aa:59:c0:3e:f1:85 root@linino
```

The command ssh-keygen -t rsa initiated the creation of the key pair.
No passphrase your was entered.
After this is completed, two files generated. The private key was saved in .ssh/id_rsa and public key was saved in .ssh/id_rsa.pub

Copy the public key to .ssh/authorized_keys with command:

```
[root@linino~]#cat .ssh/id_rsa.pub >> .ssh/authorized_keys
```

Step2: Edit SSH config

```
[root@linino~]#vi /etc/ssh/sshd_config
```

Make sure that the following settings as shown 

```
[RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys
PasswordAuthentication no
```

Step3: Restart SSH service

```
[root@linino~]#/etc/init.d/sshd restart
```

Please note that for Putty you will need to import the generated private key into puttygen and export it into a new private key. This is because Putty does not support the SSH generated private key.

Federal Communication Commission Interference 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 communications. 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.

This device and its antenna(s) must not be co-located or operating in conjunction with any other antenna or transmitter.

Country Code selection feature to be disabled for products marketed to the US/CANADA

IMPORTANT NOTE:

This module is intended for OEM integrator. The OEM integrator is still responsible for the FCC compliance requirement of the end product, which integrates this module.

20cm minimum distance has to be able to be maintained between the antenna and the users for the host this module is integrated into. Under such configuration, the FCC radiation exposure limits set forth for an population/uncontrolled environment can be satisfied.

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

USERS MANUAL OF THE END PRODUCT:

In the user’s manual of the end product, the end user has to be informed to keep at least 20cm separation with the antenna while this end product is installed and operated. The end user has to be informed that the FCC radio-frequency exposure guidelines for an uncontrolled environment can be satisfied. The end user has to also be informed that any changes or modifications not

expressly approved by the manufacturer could void the user's authority to operate this equipment. If the size of the end product is smaller than 8x10cm, then additional FCC part 15.19 statement is required to be available in the users manual: This device complies with Part 15 of 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.

LABEL OF THE END PRODUCT:

The final end product must be labeled in a visible area with the following " Contains TX FCC ID: 2AAO2DOGSTICK". If the size of the end product is larger than 8x10cm, then the following FCC part 15.19 statement has to also be available on the label: This device complies with Part 15 of 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.