

Introduction

Ubicom's Programming Dongle (shown in Figure 1, connected to a target) provides programming and debugging functions for the Ubicom32 family of processors such as IP3xxx, IP5xxx and IP7xxx through an Ethernet interface.

The Ubicom Programming Dongle is developed by Ubicom, and it provides fast in-system debugging (ISD) and In-System Programming capability using the ISD/ISP interface of Ubicom32 based processors.

The Ubicom Programming Dongle is functionally equivalent to a JTAG debugger. However, the debug capabilities and usability features are superior to those provided by a JTAG debugger. A typical JTAG uses serial interface to communicate with the host while Ubicom Dongle uses an Ethernet interface to communicate with the host system. This presents huge advantages over JTAG debuggers – advantages such as debugging over high speed interface, location independent debugging, and easy maintenance.



Figure 1. Ubicom Programming Dongle.

Contents

1. Features.....	3
2. How to Use Ubicom Programming Dongle	3
2.1. Connecting the Ubicom Dongle to the Target.....	3
2.2. IP Address of Dongle	4
2.2.1. Determining Current IP Address of the Dongle	4
2.2.2. Changing IP Address of the Dongle.....	5
2.3. Accessing Target Console with Dongle.....	6
2.4. Programming the Target using Dongle	7
2.5. Debugging Target using Dongle.....	8
2.6. Dongle firmware upgrade	9
2.7. Typical Error Notifications.....	10
2.7.1. Unable to find dongle.	10
2.7.2. Unable to establish ISD connection.....	10

1. Features

The Ubicom Programming Dongle has many features and uses such as:

- In-Circuit (IC) level debugging.
- IP connectivity through Ethernet interface
- Debugging capabilities for UBICOM32 family of processors.
- Remote debugging over IP through Ethernet interface.
- Application level debugging via telnet connection.
- Stand-alone / Programming only mode of operation.
- Controllable from a remote PC via HTTP UI.

For more information about dongle, debugging or others, please visit <http://developer.ubicom.com/>

2. How to Use Ubicom Programming Dongle

This section has several sections explaining typical usages of Ubicom Dongle from setting up the connections to debugging the running code on target.

2.1. Connecting the Ubicom Dongle to the Target

The picture **Error! Reference source not found.** shows Ubicom Programming Dongle connected to the target board. The different parts in the picture are listed and explained below.

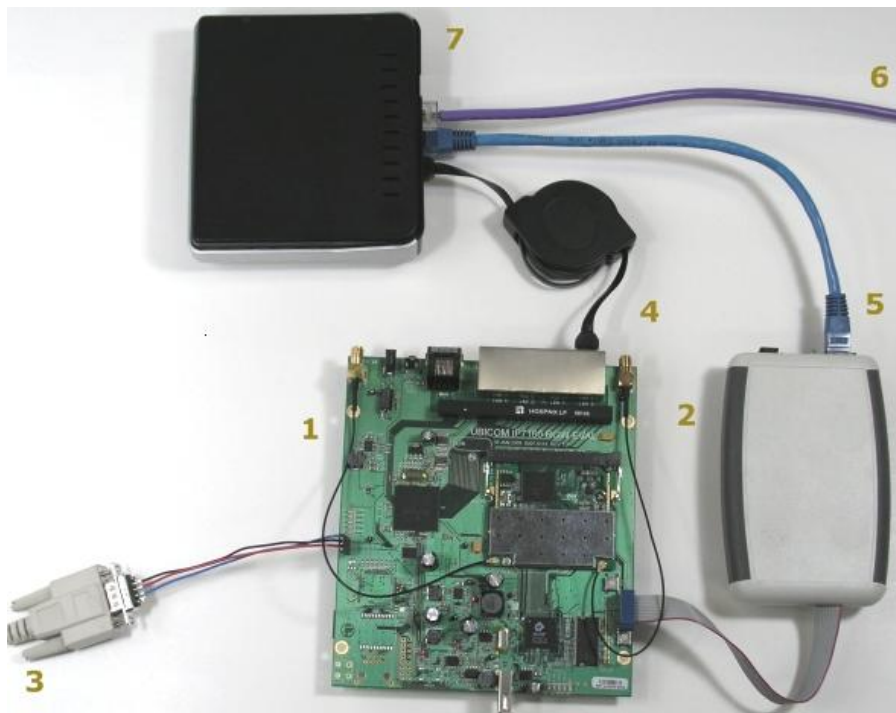


Figure 2. Connecting Ubicom Programming Dongle

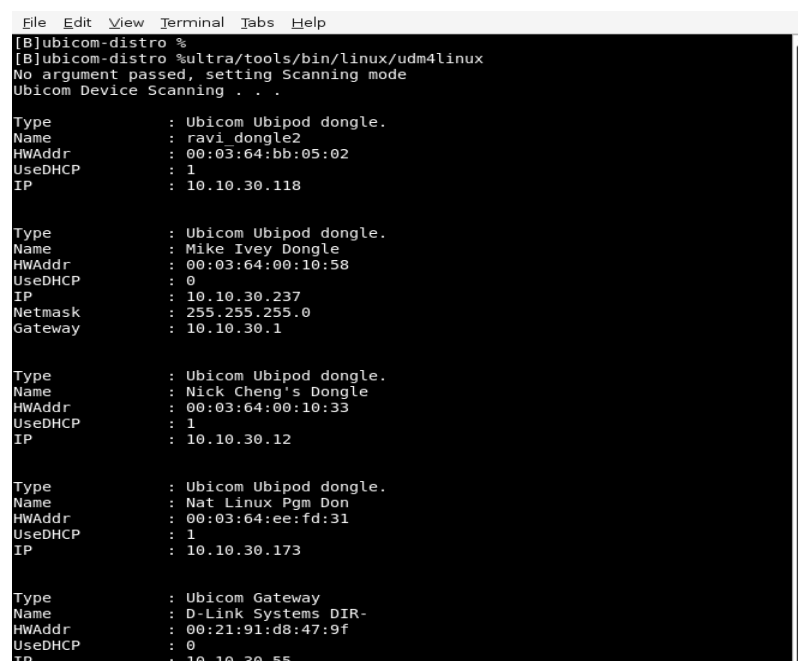
1. Ubicom reference board
2. Ubicom dongle.
3. UART connection from target to PC, for console (optional).
4. Ethernet LAN connection from board to hub/switch.
5. Ethernet connection from dongle to hub/switch.
6. Connection to PC Ethernet interface.
7. Hub/switch

2.2. IP Address of Dongle

The Ubicom programming dongle is IP addressable and it can get IP address either dynamically from a DHCP server or statically from local configuration. By default, Ubicom Programming dongle is configured to get IP addresses from DHCP server present in its LAN.

2.2.1. Determining Current IP Address of the Dongle

There are a couple of ways to determine IP address of the dongle. If the dongle has received an IP address from a DHCP server, the DHCP lease list on the DHCP server would show the IP address of the dongle. However, if the Dongle was previously assigned a static IP address, it can be determined by running **udm4linux** utility. **udm4linux** stands for “Ubicom Device Manager for Linux”. This utility is part of the Ubicom distribution and can be found in the directory `tools/bin/linux/`. Figure 3 shows output of **udm4linux** utility.



```

File Edit View Terminal Tabs Help
[BI]ubicom-distro %
[BI]ubicom-distro %ultra/tools/bin/linux/udm4linux
No argument passed, setting Scanning mode
Ubicom Device Scanning . . .

Type      : Ubicom Ubipod dongle.
Name      : ravi_dongle2
HWAddr    : 00:03:64:bb:05:02
UseDHCP   : 1
IP        : 10.10.30.118

Type      : Ubicom Ubipod dongle.
Name      : Mike Ivey Dongle
HWAddr    : 00:03:64:00:10:58
UseDHCP   : 0
IP        : 10.10.30.237
Netmask   : 255.255.255.0
Gateway   : 10.10.30.1

Type      : Ubicom Ubipod dongle.
Name      : Nick Cheng's Dongle
HWAddr    : 00:03:64:00:10:33
UseDHCP   : 1
IP        : 10.10.30.12

Type      : Ubicom Ubipod dongle.
Name      : Nat Linux Pgm Don
HWAddr    : 00:03:64:ee:fd:31
UseDHCP   : 1
IP        : 10.10.30.173

Type      : Ubicom Gateway
Name      : D-Link Systems DIR-
HWAddr    : 00:21:91:d8:47:9f
UseDHCP   : 0
IP        : 10.10.30.55
    
```

Figure 3. Output from udm4linux Tool

2.2.2. Changing IP Address of the Dongle

The IP address settings of the dongle can be changed either through **udm4linux** tool or through the web UI available at http://<current_dongle_ip_address>.

The **udm4linux** utility can be used to change the IP address or the IP mode (static to DHCP or vice versa). The help and options can be obtained by running **udm4linux -help** command, and the output is as shown in Figure 4.

```

File Edit View Terminal Tabs Help
Terminal
[B]rtr %
[B]rtr %./ultra/tools/bin/linux/udm4linux --help

    Ubicom Device Manager for Linux
    Usage: udm4linux [OPTION]

    -v,--verbose           = Increase verbosity
    -s,--scan              = Ubicom Device scanning
    -x,--xml               = XML output of scan function (will be used for
integration with Eclipse)
    -t,--target            = Hardware Address of Ubicom Device to be set-up
.Takes parameter as "AA:BB:CC:DD:EE:FF"
    -d,--dhcp              = Set target to DHCP mode (target parameter need
ed)
    -e,--set               = Set targets ip address.Takes parameter as "IpA
ddress:NetmaskAddress:GatewayAddress"(target parameter needed)
    -r,--reset             = Resets Ubicom Device. (target parameter neede
d.
    -f,--tftp              = Send image to Ubicom Device (not implemented y
et)
    -i,--interface         = Bind an interface to send packets from.(needs
superuser privileges)
    -m,--timeout           = Set lifetime value if there is no incoming pac
kets (default is 2 secs)
    -l,--ttl               = Set ttl number of Ip packet.(must be between 0
and 255)
    -h,--help              = Print this help message

    Examples:

    Scanning Devices connected to ethernet card 1 (eth1):
    #./udm4linux -s -i eth1

    Setting up Device to dhcp:
    $./udm4linux -t 00:03:64:00:05:33 -d

    Setting Up Device to: Ip Address : 172.15.22.12, Subnet Mask : 255.255.255.0,
Gateway : 172.15.22.1
    $./udm4linux -t 00:03:64:00:05:33 -e "175.15.22.12:255.255.255.0:172.15.22.1"
    Or you can leave gateway address blank
    $./udm4linux -t 00:03:64:00:05:33 -e "175.15.22.12:255.255.255.0:"

[B]rtr %

```

Figure 4. udm4linux Help and Options

The Dongle's IP address and settings can also be changed from the web UI as shown in Figure 5.

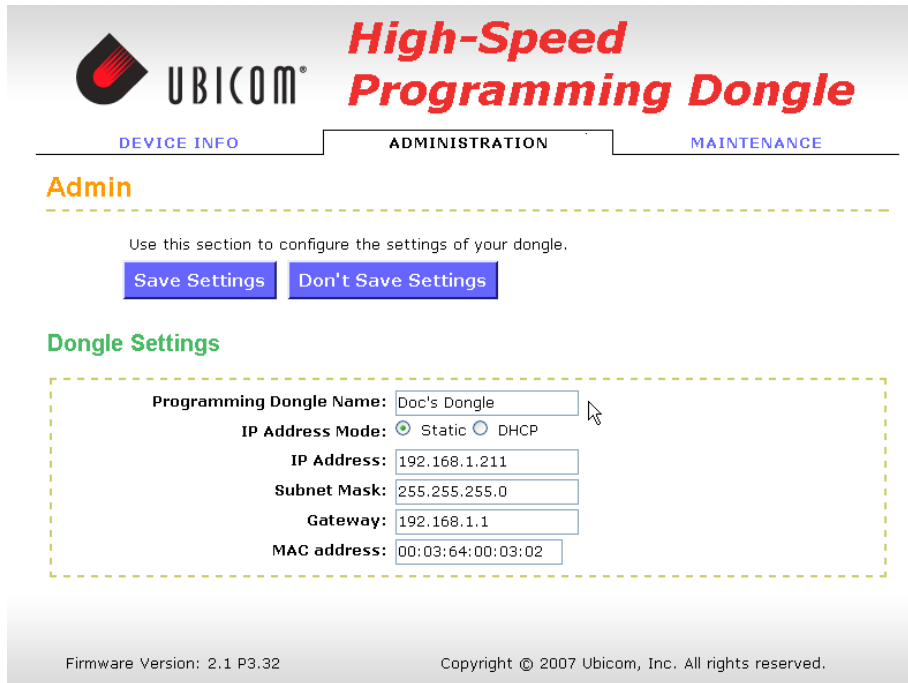


Figure 5. Web UI of Dongle.

2.3. Accessing Target Console with Dongle

The console on the target can be accessed through the Dongle if the console option has been enabled in the firmware running on the target. Refer <http://developer.ubicom.com> for information on building the firmware and debugging through console.

The target console is accessed by using the **telnet** utility on the host PC and connecting to dongle IP address and port 50. **Note that there is no “:” between IP address and port number.**

```
$ telnet dongle_ip_address 50
```

Typical busybox commands, such as **vi**, **find**, **ls**, **cd**, are available in the console. On bootup, the console would look as shown in Figure 6.

```
File Edit View Terminal Tabs Help
[B]l0work %
[B]l0work %
[B]l0work %telnet 10.10.30.118 50
Trying 10.10.30.118...
Connected to 10.10.30.118.
Escape character is '^'.

Ubicom High-Speed Programming Dongle. Firmware Version: 3.2.R1.0
Copyright (c) 2007 Ubicom, Inc. All rights reserved.

Telnet Console Target Board Connection Established.

/ #
/ # ls -l
drwxr-xr-x 2 root root 0 Aug 10 2009 bin
drwxr-xr-x 2 root root 0 Aug 10 2009 boot
drwxr-xr-x 4 root root 0 Jan 1 00:00 dev
drwxr-xr-x 1 root root 0 Aug 10 2009 etc
drwxr-xr-x 6 root root 0 Jan 1 00:00 flash
drwxr-xr-x 2 root root 0 Aug 10 2009 home
-rwxr-xr-x 1 root root 2175 Jul 28 2009 init
-rwxr-xr-x 1 root root 2398 Jul 28 2009 init.orig
-rwxr-xr-x 1 root root 475 Jul 28 2009 init.rgw
drwxr-xr-x 2 root root 0 Aug 10 2009 jffs
drwxr-xr-x 8 root root 0 Aug 10 2009 lib
drwxr-xr-x 2 root root 0 Aug 10 2009 mnt
dr-xr-xr-x 38 root root 0 Jan 1 00:00 proc
drwxr-xr-x 2 root root 0 Aug 10 2009 rom
drwx----- 4 root root 0 Jan 1 00:00 root
drwxr-xr-x 2 root root 0 Aug 10 2009/sbin
drwxr-xr-x 11 root root 0 Jan 1 00:00 sys
drwxr-xr-x 2 root root 0 Aug 10 2009/tetc
drwxr-xr-x 4 root root 0 Jan 1 00:00 tmp
drwxr-xr-x 7 root root 0 Aug 10 2009/usr
drwxr-xr-x 8 root root 0 Jan 1 00:00 var
drwxr-xr-x 3 root root 0 Aug 10 2009/www
/ #
/ #
/ #
```

Figure 6. Target console accessed by Dongle.

2.4. Programming the Target using Dongle

The Ubicom Dongle can be used to program the target with firmware through **ubicom32-elf-gdb** utility that is part of the standard GNU tool chain in Ubicom's Linux distribution. Before starting the GDB utility, connect the dongle to the target via the ISP/ISD interface and power-on both the target and the dongle. Assuming that the dongle IP address is 192.168.0.221, set up the following environment variables for easy usage:

```
$ export UBICOM_DONGLE=192.168.0.221:5010
$ export PATH=$PATH:~/ubicom-disrto/toolchain/bin
$ export LD_LIBRARY_PATH=~/ubicom-disrto/toolchain/lib
```

Invoke GDB at a shell command prompt:

```
$ ubicom32-elf-gdb vmlinux.elf
```

GDB will show some version information and a new prompt. Enter the following command to attach gdb to the target. Please note that specifying IP address and port is needed only if UBICOM_DONGLE environment variable is not set as mentioned in previous step.

```
(gdb) target ubicom32
or
(gdb) target ubicom32 192.168.0.221:5010
```

At this stage, GDB should connect to the target (i.e. the target) and reply: *Connected to target*. Please type:

```
(gdb) load
```

GDB will show the status of programming through a progressive series of dots and return to its prompt. Until gdb is instructed to disconnect from the target the target will remain in a halt/reset state. To disconnect from the target and quit GDB, execute the following 2 commands.

```
(gdb) detach
(gdb) quit
```

The target is now released from halt/reset state and would start/resume running.

2.5. Debugging Target using Dongle

The Ubicom Dongle can be used for In-System Debug (ISD) of target using **ubicom32-elf-gdb** or **ubicom32-elf-insight** utilities available as part of the tool chain supplied in Ubicom SDK. Please note that more information about debugging is available at <http://developer.ubicom.com>.

Using the dongle to debug is very simple operation as shown in the example below.

Launch the debugger using following command assuming that dongle's IP address is 192.168.0.221 and that the dongle is connected to a live target.

```
$ ubicom32-elf-insight vmlinux.elf
```

After debugger utility loads the target image, use the following command in the debugger console to attach the debugger utility to the target for In-System debugging.

```
$ (gdb) target ubicom32 192.168.0.221:5010
```

Since **ubicom32-elf-gdb** is based on standard GNU **gdb**, debugging with **ubicom2-elf-gdb** is very much like standard GNU **gdb** tools as shown below.

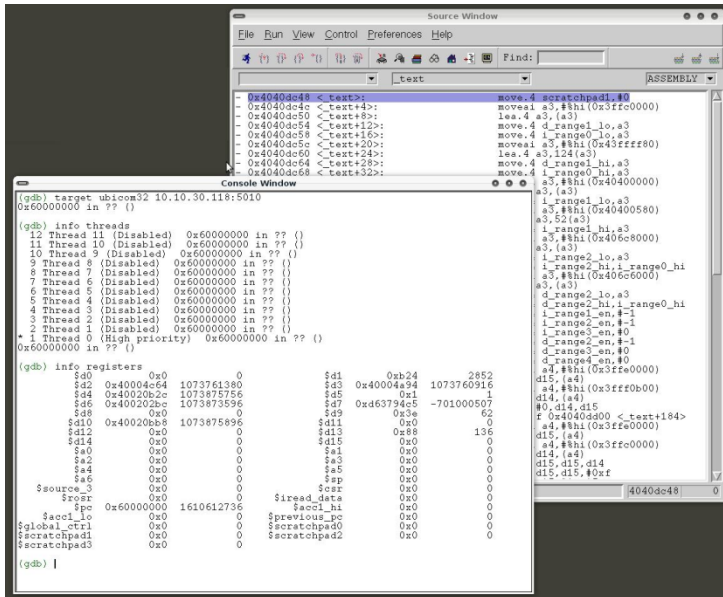


Figure 7. Target Debugging.

2.6. Dongle firmware upgrade

The Ubicom dongle can be updated to latest firmware available from the Ubicom distribution using the web UI. The latest firmware for the dongle is available in the distribution at the location:

```
../Firmware/Dongle/ubicom_dongle_with_console.bin
```

Note that you might want to save your configuration before updating the dongle with the new firmware.

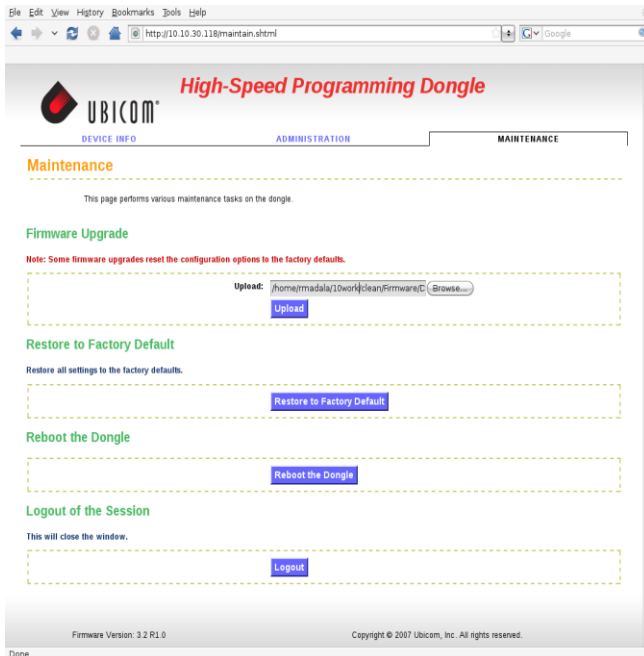


Figure 8. Web UI of Ubicom Programming Dongle

2.7. Typical Error Notifications

2.7.1. Unable to find dongle.

This error means that the **ubicom32-elf-gdb** utility was unable to find IP connectivity to the dongle. This error can occur if

- Dongle is not powered ON
- \$UBICOM_DONGLE environment variable is not set or is different from the current IP address of the Dongle.
- Dongle IP address is in a different subnet and there is no IP route to the dongle IP address.

Solution:

Check the IP address of the dongle using **udm4linux** utility as explained above.

2.7.2. Unable to establish ISD connection.

This error can occur if communication between the dongle and target is lost. Possible reasons for this issue are

- The physical connection between dongle and target is lost because target board is not powered ON.
- The Dongle connection header attached to the ISD pins on the target is not aligned to PIN1 or not connected properly.

Solution:

- Check the connection between dongle and target.
- Ensure that pin 1 of ISD connector on the development board matches with the pin 1 on the dongle. If the problem persists, try connecting the header other way.
- Power cycling the dongle or the target could help too.



195 Baypointe Parkway
San Jose, CA 94134

Tel 408.433.3300
Fax 408.433.3339

Email sales@ubicom.com
Web www.ubicom.com

Ubicom[®], Inc.

Ubicom develops networking and media processor solutions that address the unique demands of real-time interactive applications and multimedia content delivery in the digital home. The company provides optimized system-level solutions for a wide range of products including wireless routers, access points, bridges, VoIP gateways, networked digital photo frames, and streaming media players.

Copyright 2009 Ubicom, Inc. All rights reserved.

Ubicom, StreamEngine, IP7000, and UBICOM32 are trademarks of Ubicom, Inc. All other trademarks are the property of their respective holders.