BCM4210

Installation Guide For iLine10^{TM} USB Drivers

Part Number 00-0846-01 • 6/20/00

Broadcom Corporation reserves the right to make changes to the products or information contained in this document without notice. No liability is assumed as a result of their use or application. No rights under any patent accompany the sale of any such products or information.

Epigram, InsideLine, and iLine10 are trademarks of Broadcom Corporation.

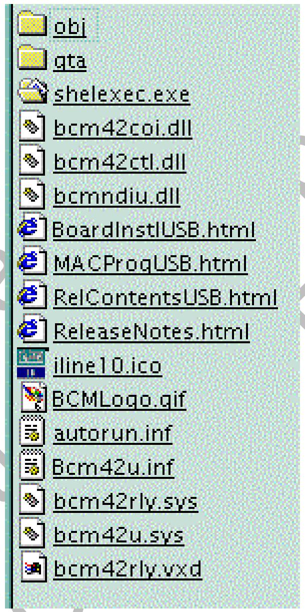
Broadcom Corporation
870 West Maude Avenue
Sunnyvale CA 94086
www.broadcom.com

1. Introduction

This document deals with the installation of USB

1.1 Release Contents

This binary release of the software for Broadcom's USB iLine10™ Network Driver consists of the Windows 98/ME/Win2000 driver and 2 directories.



obj
qta
shelexec.exe
bcm42coi.dll
bcm42ctl.dll
bcmndiu.dll
BoardInstlUSB.html
MACProgUSB.html
RelContentsUSB.html
ReleaseNotes.html
iline10.ico
BCMLogo.gif
autorun.inf
Bcm42u.inf
bcm42rly.sys
bcm42u.sys
bcm42rly.vxd

- The toplevel directory contains the Windows driver and ***inf*** files.
- **qta** - contains the executables for the tools to download, dump, and program the KLSI code and MAC address into the hardware.
- **obj** - contains the object files for the target processor: the basic initialization and descriptors to put in the i2c; and a few small standalone diagnostics.

1.2 Win98/ME Compatible Tools

- *qtsarena* - Displays the current memory usage. No input parameters.
- *qtsdump* - Displays memory. Input parameters are address and length.
- *qtsload* - Loads a program into memory and executes it. Input parameter is file name without an extension.
- *qtsi2c, qtsli2c* - Program the small (qtsi2c) or large (qtsli2c) i2c eeprom. Input parameters are the file names of programs without extensions or a file name with the ".bin" extension for an image file, with plus signs between them and no spaces.

macparse- This is the main tool, that runs under all Windows environments which reads and updates the MAC address, vendor and device identifiers and string descriptions from the small i2c.

1.3 Target Firmware

- *i2cimg.bin* - This is the USB firmware image that we want to program into the E2PROM.
- *blink.(fix,obj)* - A simple iLine10 chip communication test that blinks the LED1.
- *memtest.(fix,obj)* - A Memory test program.
- *blast.(fix,obj)* - A program that blasts packets out continuously at 2 bpb.
- *blast4.(fix,obj)* - A program that blasts packets out continuously at 4 bpb.
- *blast8.(fix,obj)* - A program that blasts packets out continuously at 8 bpb.
- *sink.(fix,obj)* - A program that receives packets and discards them.
- *loopbk.(fix,obj)* - A program that puts the iLine10 chip in loopback mode and sends and receives packets, checking them for correct contents.
- *fttest.(fix,obj)* - A program that tests the filter table by filling it with various patterns and sending packets in loopback mode, checking whether the are received or not according with the filter rules

Important Note

It is recommended that user dedicates a machine to E2PROM programming. *This machine should not have the NDIS driver installed into it* so that even boards that have already been programmed with the Vendor ID and Product ID claimed by the NDIS driver can be re-programmed. To be able to do this, it is necessary to edit the file qta\qtprog.inf and adding the proper vid/pid pair under the *QTProgInstall* section *before* installing the *qtprog* driver.

2. System Requirements

Software (supports the following operating systems):

- Windows 98
- Windows 98, Second Edition
- Windows latest version of ME i.e. 4.90.2499
- Windows 2000 Build 2195 or later

Hardware:

- Minimum configuration: Pentium 200 MHz, 32 MB RAM
- Broadcom USB iLine10 Reference Design, revision 1.0

2.1 New Board Installation

To begin using a new USB board with a blank EEPROM only, it is necessary to program its serial E2PROM (i2c compatible) with the proper descriptors and download code.

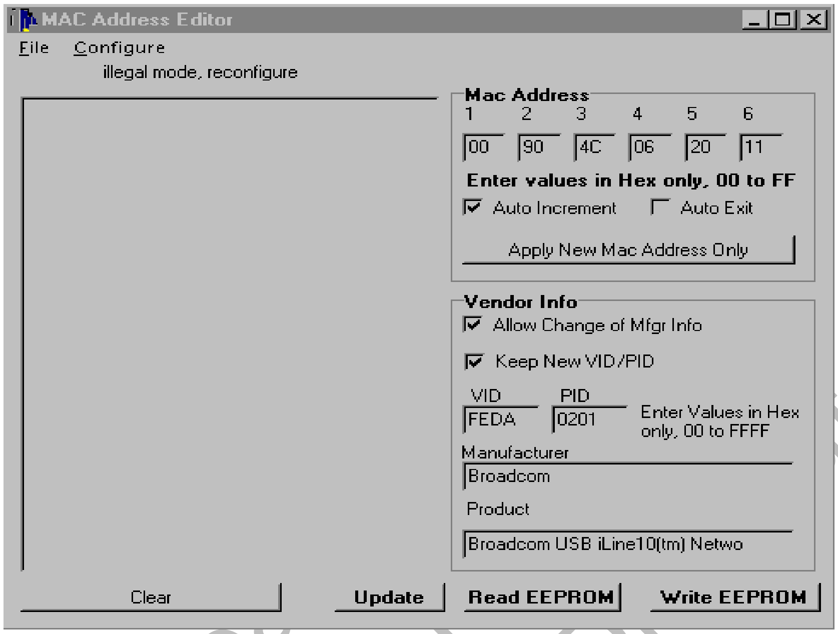
2.1.1 Setting up the Environment

2.1.1.1 Windows 98/98SE/ME

1. The first time user plugs in a USB dongle into a PC, it should bring up the “Add New Hardware Wizard” for “Unprogrammed Broadcom USB iLine10 Network Adapter”. Click on the “Next” button.
2. It will then offer to “Search for the best driver for your device.” Click on “Next” again.
3. In the next panel you should select the option to “Specify a location.” Point it to “C:\bcm42xu\qta” or the location of the “*qta”* directory created by this distribution and click on “Next.”
4. It will then say the it found a driver and that its location is “C:\bcm42xu\qta\qtprog.inf,” click on “Next” one more time.
5. At that point, it will say “Windows has finished installing the software that the new hardware device requires.” Click on “Finish” and you are done. After one has gone through this process an a machine, the qtprog driver installed by it is now in place and it claims ownership of the unprogrammed devices, so user will not see the “Add New Hardware Wizard” again.

2.1.1.2 MAC, VID, PID Programming

Start the program *macparse.exe in the “qta” directory*. This is a Windows GUI application that allows one to program the EEPROM from scratch as well as change the MAC address, Vendor ID (VID), ProductID (PID), and the strings descriptions of a previously programmed device (see Figure 1 below)



illegal mode, reconfigure

Mac Address: 00 90 4C 06 20 11

Enter values in Hex only, 00 to FF

☑ Auto Increment ☐ Auto Exit

☑ Allow Change of Mfgr Info

☑ Keep New VID/PID

VID: FEDA PID: 0201

Enter Values in Hex only, 00 to FFFF

Manufacturer: Broadcom

Product: Broadcom USB iLine10(tm) Netwo

Figure 1. MACPARSE GUI

2.1.2 Programming a board for the first time (Blank E2PROM)

1. Click on File->Read BIN File and type i2cimg.bin or browse to find it.
2. Fill in the new MAC address, vid, pid, and strings inside the GUI.
3. Finally, click on "Write EEPROM" (**not Update**) and wait for the "Done" message.
4. You can now unplug that board and program another one by simply plugging it in, updating the MAC address and clicking on "Write EEPROM." The MAC address can be made to increment automatically by checking the "Auto Increment" box, but ***BEWARE*** that the program has a bug - it increments the numbers as if they were decimal when in fact they are hex, so 19 becomes 20, not 1A, even though 1A will become 1B. You can program as many boards as required following this procedure.

2.1.3 Updating a previously programmed board

1. Fill in the values you want for all of the MAC, PID and VID fields and click on "**Update**". It will read the current values from the *small i2c* and then it will write the new ones back. When it is done it displays the message "Done."
2. Disconnect/reconnect to reset it. Reconnecting should bring up the "Add New Hardware Wizard" for "Broadcom USB iLine10 Network Adapter" after a couple of seconds.
3. Follow the Wizard screens through, searching for the best driver, and specifying a location to search as the c:\bcm42xx directory. This is the directory containing the installer file, *bcm42u.inf.*
4. Adjust the network settings for the "TCP/IP = Broadcom USB iLine10 Network Adapter" as appropriate for your network.

2.1.4 Debugging the Installation

When a new board is being brought up in Windows 98/ME environment only (*QTA tools from KLSI are not supported in NT, Win2000 environment*), one can debug the installation using the serial tools in the ***qta*** directory of the distribution.

To run the tools, execute qtenv.bat in the qta directory, it will add C:\bcm42xu\qta to the path. If you used a different install directory, you must edit this file first. The tools communicate with the target over a serial port, by default COM1 is used; if you need to use a different serial port, you must edit the file “qtserial.ini” and replace com1 with the proper port.

2.1.5 Board Bringup

1. First, make sure that you start with a blank E2. The following command will clear them:

```
C:\bcm42xu\> qtsi2c zero.bin
com1: baud=14400 parity=N data=8 stop=1
```

2. Disconnect/reconnect. The arena should look like:

```
C:\bcm42xu\> qtsarena
com1: baud=14400 parity=N data=8 stop=1
start 02fa size 0902 Empty
start 0c00 size f400 Used
```

3. Now program the i2c. Use the following command:

```
C:\bcm42xu\> qtsi2c i2cimg.bin
com1: baud=14400 parity=N data=8 stop=1
```

4. Wait 45 seconds after qtsi2c command returns to allow KLSI to finish programing i2c (measuring it takes about 30 seconds; 45 seconds should allow for any variability), then disconnect/reconnect. The arena should look like (last line depends on strin sizes):

```
C:\bcm42xu\> qtsarena
com1: baud=14400 parity=N data=8 stop=1
start 02fa size 05c8 Used
start 08c6 size b670 Empty
start bf3a size 40c6 Used
```

5. This is already an indication that the memory is mostly working. Run blink:

```
C:\bcm42xu\> qtsload blink
com1: baud=14400 parity=N data=8 stop=1
```

   You should have LED1 blinking with a period of 800 milliseconds. This indicates basic communication with the D-chip is working. If the software detects an error, you’ll have a fast blink (100 millisecond period) instead.

6. Disconnect/reconnect to reset it. Run memtest:

```
C:\bcm42xu\> qtsload memtest
com1: baud=14400 parity=N data=8 stop=1
```

You should now have a slow blinking LED1 if everything is OK or a fast blinking one if a test fails. The program tests by filling and then checking for zeroes, all-ones, 0x5555, 0xaaaa, and with increasing and then decreasing values starting with 1, 2, 4, 8, etc. up to 0x8000. It then repeats all over.

7. Disconnect/reconnect to reset it. Run blast:

```
C:\bcm42xu\> qtsload blast
com1: baud=14400 parity=N data=8 stop=1
```

You should now have the Tx LED (LED2), on almost continuously, and packets coming out on the phone line at a very high rate. The LI LED (LED1) toggles every 256 packets transmitted. Packets are a few hundred millivolts peek-to-peek, and about 2.2 ms long.

You can use a second dongle to receive the packets by running sink on it:

```
C:\bcm42xu\> qtsload sink
com1: baud=14400 parity=N data=8 stop=1
```

Sink receives packets and discards them so this dongle should have the Rx LED (LED3) on continuously. The LI LED toggles every 256 packets received.

This process can be repeated with blink4 and blink8, which are just like blink, but they send the packets at 4 and 8 bits per baud respectively (instead of the 2 bits per baud that blast does). The sink program can be used to receive packets from any of them.

8. Disconnect/reconnect to reset it. Run loopbk:

```
C:\bcm42xu\> qtsload loopbk
com1: baud=14400 parity=N data=8 stop=1
```

Although this program behaves just like blink, slowly flashing LED1, it actually exercises a lot more of the iLine10 chip, sending and receiving packets in loopback mode.

9. Disconnect/reconnect to reset it. Run fttest:

```
C:\bcm42xu\> qtsload fttest
com1: baud=14400 parity=N data=8 stop=1
```

This program also seems to just blink, but it is sending and receiving packets in loopback mode, while at the same time changing the filter contents to make sure that the filter memory and function are working correctly.

10. Disconnect/reconnect to reset it. Now program the correct MAC address on the device.