



Using LAPB support with FarSync Adapters



Document Revision 1.0.1a
For use with FarSync Device Software 4.0.0 b408

FarSite Communications Ltd
info@farsite.com
www.farsite.com



© Copyright FarSite Communications Ltd. 2010..2024

Copyright

The copyright to this manual and the software described herein is owned by FarSite Communications Limited; it may not be translated or modified without prior written permission from FarSite Communications Limited.

Disclaimer

Whilst every effort is made to ensure accuracy within this manual, FarSite Communications Limited cannot be held responsible for errors or omissions, and reserves the right to revise this document without notice.

Trademarks

All trademarks and registered trademarks are acknowledged.



1	Introduction.....	3
1.1	Prerequisites	3
2	Preparing and Installing the WAN driver.....	4
2.1	Assemble the required files into the kernel-hdlc directory	4
2.1.1	Change to the kernel-hdlc directory.....	4
2.1.2	Copy the kernel's HDLC/LAPB source files.....	4
2.1.3	Update hdlc_x25.c.....	4
2.2	Edit Makefile	6
2.3	Install WAN driver	6
3	Post WAN driver install	7
3.1	Start driver	7
3.2	Load the LAPB and HDLC x25 modules	7
3.3	Edit port configuration files	7
3.3.1	ifcfg-hdlc0 (configured as a DTE)	7
3.3.2	ifcfg-hdlc1 (configured as a DCE).....	7
3.4	Open both ports	8
3.5	Managing interfaces without the use of ifcfg configuration files	8
4	Exercising the LAPB Interfaces	10
4.1	Build the LAPB sample application.....	10
4.2	Connect two ports together.....	10
4.3	Run the sample application in two terminals	10
4.3.1	Start the Server.....	10
4.3.2	Start the Client.....	10
4.4	Test Results	11
5	Stopping the driver and unloading the modules	12
5.1	Take down any active interfaces.....	12
5.2	Stop the farsync driver	12
5.3	Unload the hdlc_x25, lapb and hdlc modules.....	12
5.4	Example stop script	12



1 Introduction

This document describes how to setup, configure and run LAPB over FarSync cards/devices on a Linux-based installation.

RedHat Enterprise Linux 8 was used to produce this document, but the same process applies to other distributions.

Either a FarSync PCI(e) card or two FarSync Flex devices are required in order to exercise the LAPB support using the FarSync sample application, testLapb.

This guide is meant only to allow the user to setup, configure and exercise the LAPB support - more advanced, detailed information, such as for the LAPB API itself, is contained within the FarSync SDK V2.3.6.

It is assumed that the FarSync Device Software (farsync-4.0.0-b408.tar.gz) and FarSync SDK (farsync_sdk_2.3.6.tar.gz) files have been expanded to the user's home directory.

1.1 Prerequisites

Before continuing, please make sure the following are in place:

- The Kernel source has been installed.
- A symbolic link has been added in /usr/src/ pointing to the kernel source

```
[build@rhel8-4vm1 farsync-4.0.0-b408]$ ls -la /usr/src | grep linux
lrwxrwxrwx. 1 root root 46 Aug 12 2021 linux -> /usr/src/kernels/4.18.0-305.12.1.el8_4.x86_64/
[build@rhel8-4vm1 farsync-4.0.0-b408]$
```

- Kernel development tools should be installed including: C compiler, Make system and Patch utility.
- farsync_sdk_2.3.6 should have been copied to the system and expanded.
- The farsync-4.0.0-b408 folder should have been extracted from farsync-4.0.0-b408.tar.gz onto the build system.



2 Preparing and Installing the WAN driver

2.1 Assemble the required files into the kernel-hdlc directory

2.1.1 Change to the kernel-hdlc directory

```
cd ~/farsync-4.0.0-b408/kernel-hdlc
```

2.1.2 Copy the kernel's HDLC/LAPB source files

Note: If you do not have these files installed locally you will need to source them from a corresponding kernel repos.

```
cp Makefile-with-hdlc-3.17.0 Makefile
cp /usr/src/linux/drivers/net/wan/hdlc* .
cp /usr/src/linux/include/linux/hdlc.h .
cp /usr/src/linux/net/lapb/lapb_* .
```

2.1.3 Update hdlc_x25.c

Typically, to enable the LAPB support, the hdlc_x25.c source file included in the precise distribution/kernel version that you are using, needs to be updated.

The following example is from RHEL 8.6 with kernel 4.18.0-372.32.1. Your version may not match it exactly so the following information needs to be used as a guide rather than modifications to be applied verbatim.

Check hdlc_x25.c and ensure that the following line is present in the x25_connect_disconnect() and x25_data_indication() routines, immediately after the call to x25_type_trans().

```
.
.
    skb->protocol = x25_type_trans(skb, dev);
    skb->protocol = htons(ETH_P_CUST); ←
.
.
```

If it is not present then **add** it now.

The line containing the call to x25_type_trans() should remain as is.

If required, to support the module parameters referenced in Section 3.2 i.e. hdlc_dce_count and hdlc_dce_list, the following further mods can be made to hdlc_x25.c (adding the lines marked with a + accordingly):

```
.
.
#include <net/x25device.h>

static int x25_ioctl(struct net_device *dev, struct ifreq *ifr);

+/*
+ *      Module info
+ */
+/*
+ * Modules parameters and associated variables
+ */
+int  hdlc_dce_count=0;
+int  hdlc_dce_list[16];
+module_param (hdlc_dce_count,int,S_IRUGO);
+module_param_array (hdlc_dce_list,int, NULL, S_IRUGO);

/* These functions are callbacks called by LAPB layer */

static void x25_connect_disconnect(struct net_device *dev, int reason, int code)
.
.
static int x25_open(struct net_device *dev)
```



```
{
  int result;
  static const struct lapb_register_struct cb = {
    .connect_confirmation = x25_connected,
    .connect_indication = x25_connected,
    .disconnect_confirmation = x25_disconnected,
    .disconnect_indication = x25_disconnected,
    .data_indication = x25_data_indication,
    .data_transmit = x25_data_transmit,
  };
  +struct lapb_parms_struct    params;
  +int    port_no;
  +int    port_found;
  +int    i;

  result = lapb_register(dev, &cb);
  if (result != LAPB_OK)
    return result;
+  /*
+   * Convert the numeric part of the device name to an
+   * integer
+   */
+  port_no = dev->name[4] - 0x30;
+  port_found = 0 ;
+  for (i=0; i < hdlc_dce_count; i++)
+  {
+    if (port_no == hdlc_dce_list[i])
+    {
+      /*
+       * This port is a dce
+       */
+      printk("Opening X25 dce device %s %d\n", dev->name, i);
+      lapb_getparms(dev, &params);
+      params.mode = params.mode | 0x4;
+      lapb_setparms(dev, &params);
+      port_found++;
+      break;
+    }
+  }
+  if (!port_found)
+  {
+    printk("Opening X25 dte device %s\n", dev->name);
+  }
  return 0;
}
.
```



2.2 Edit Makefile

Below is an extract from the Makefile, all the parameters in **bold** require to be added or uncommented.

```
#
# Makefile for the Linux farsync & fsflex
#
# $Id$
#
COMPNY_DIR = /etc/farsite
MODULE_DIR = $(COMPNY_DIR)/modules
KMOD_DIR = /lib/modules/`uname -r`/kernel/drivers/net/wan
EXTRA_CFLAGS := -I/etc/farsite/include -DUSE_INTERRUPTS
EXTRA_CFLAGS += -DCONFIG_HDLC_RAW -DCONFIG_HDLC_RAW_ETH -DCONFIG_HDLC_PPP -
DCONFIG_HDLC_CISCO -DCONFIG_HDLC_FR -DCONFIG_HDLC_X25
lapb-objs := lapb_in.o lapb_out.o lapb_subr.o lapb_timer.o lapb_iface.o

ifneq ($(KERNELRELEASE),)
obj-m += lapb.o farsync.o fsflex.o hdlc.o hdlc_raw.o hdlc_raw_eth.o hdlc_cisco.o hdlc_fr.o hdlc_ppp.o hdlc_x25.o
else
KDIR := /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)
```

2.3 Install WAN driver

The install script is located in the farsync-4.0.0-b408 directory. So, run

```
cd ../
sudo ./install wan
```

The following error may well be reported:

```
hdlc_fr.c:1282:2: error: implicit declaration of function 'mark_driver_unmaintained'; did you mean
'mark_driver_unsupported'? [-Werror=implicit-function-declaration]
   mark_driver_unmaintained(THIS_MODULE->name);
   ^~~~~~
   mark_driver_unsupportfs
```

In which case edit the hdlc_fr.c file and change line 1282. In the extract below, line 1282 has been commented out and a new line added with the required change:

```
1282 /*      mark_driver_unmaintained(THIS_MODULE->name); */
1283      mark_driver_unsupported(THIS_MODULE->name);
```

After making the change, run:

```
sudo ./install wan
```



3 Post WAN driver install

3.1 Start driver

```
sudo /etc/init.d/farsync start
```

3.2 Load the LAPB and HDLC x25 modules

The modules are currently located in the kernel-hdlc directory

```
sudo insmod kernel-hdlc/lapb.ko
sudo insmod kernel-hdlc/hdlc_x25.ko hdlc_dce_count=1 hdlc_dce_list=1
```

```
[build@rhel8-4vm1 farsync-4.0.0-b408]$ sudo insmod kernel-hdlc/lapb.ko
[build@rhel8-4vm1 farsync-4.0.0-b408]$ sudo insmod kernel-hdlc/hdlc_x25.ko hdlc_dce_count=1 hdlc_dce_list=1
[build@rhel8-4vm1 farsync-4.0.0-b408]$
```

hdlc_dce_count the number of dce interfaces required
hdlc_dce_list a list of the interface indexes of the DCE's

3.3 Edit port configuration files

Two ports are required to be configured for the test. Edit both `/etc/farsite/farsync/ifcfg-hdlc0` and `/etc/farsite/farsync/ifcfg-hdlc1` and configure accordingly. `IPADDR`, `IPV6ADDR`, `POINTOPOINT`, `NETMASK`, `NETWORK` and `MTU`, can be commented out, as they are not required.

The use of **ifcfg** network configuration files has been deprecated for some time now and more recent Linux distributions have removed support for them entirely. For systems without support for **ifcfg** network files, please refer next to Section 3.5.

3.3.1 ifcfg-hdlc0 (configured as a DTE)

```
# Other media options are shdsl, E1, T1, x21d, v24, v35, rs530 and rs449
MEDIA=x21
# Network protocols are PPP or CISCO
PROTO=x25
# We are a DTE and use the external clock from the line/DCE
CLOCK=ext
# Make sure hdlc linemode is selected
LINEMODE=hdlc
```

3.3.2 ifcfg-hdlc1 (configured as a DCE)

```
# Other media options are shdsl E1, T1, x21d, v24, v35, rs530 and rs449
MEDIA=x21
# Network protocols are PPP or CISCO
PROTO=x25
# We are a DCE and provide clock at the specified rate
CLOCK=64000
# Make sure hdlc linemode is selected
LINEMODE=hdlc
```



3.4 Open both ports

For RedHat type systems:

```
sudo ifup hdlc0
sudo ifup hdlc1
```

For Debian type systems:

```
sudo farifup hdlc0
sudo farifup hdlc1
```

3.5 Managing interfaces without the use of ifcfg configuration files

If the system does not support **ifcfg** configuration files and/or **ifup** then the use of a user-managed script, such as the following example (to be run as root), is an alternative:

```
/etc/init.d/farsync start
ifconfig hdlc0 down
ifconfig hdlc1 down
insmod kernel-hdlc/lapb.ko
insmod kernel-hdlc/hdlc_x25.ko
farutil hdlc0 set media rs530
farutil hdlc0 set clock ext
farutil hdlc0 set proto x25
farutil hdlc1 set media rs530
farutil hdlc1 set clock 64000
farutil hdlc1 set proto x25
ifconfig hdlc0 up
ifconfig hdlc1 up
```

Such systems/versions may still report warnings such as e.g.

```
WARN: [ifup] You are using 'ifup' script provided by 'network-scripts', which are now deprecated.
WARN: [ifup] 'network-scripts' will be removed in one of the next major releases of RHEL.
WARN: [ifup] It is advised to switch to 'NetworkManager' instead - it provides 'ifup/ifdown' scripts as well.
```

To address this you can change `/etc/init.d/farsync` from

```
.
.
# Now call the interface up script for each if we have one
if [ -x ${IFUP} ]
then
    i=0
    while [ $i -lt 64 ]
    do
.
.
to
.
.
# Now call the interface up script for each if we have one
if [ -x ${IFUP} ]
then
    i=0
    while false #[ $i -lt 64 ]
    do
.
.
```



and from

```
.  
.  
    if [ -x ${IFDOWN} ]  
    then  
        i=0  
        while [ $i -lt 64 ]  
        do
```

```
.  
.  
to
```

```
.  
.  
    if [ -x ${IFDOWN} ]  
    then  
        i=0  
        while false #[ $i -lt 64 ]  
        do
```

and simply use the user-managed script(s) as above, and in Section 5.4, to manage the interface states.



4 Exercising the LAPB Interfaces

4.1 Build the LAPB sample application

```
$cd ~/farsync_sdk-2.3.6/examples/lapbApi/example/  
$ make  
cc -o testLapb -Wall -Wstrict-prototypes testLapb.c  
$
```

```
[build@rhel8-4vm1 example]$ ls  
Makefile testLapb.c  
[build@rhel8-4vm1 example]$ make  
cc -o testLapb -Wall -Wstrict-prototypes testLapb.c  
[build@rhel8-4vm1 example]$ ls  
Makefile testLapb testLapb.c  
[build@rhel8-4vm1 example]$
```

4.2 Connect two ports together

In this case the card being used is a T4Ue with an MTU4 cable. Connected to both A & B ports are UCX1 cables which are then connected together using a NULL-MX cable.

4.3 Run the sample application in two terminals

4.3.1 Start the Server

```
[build@rhel8-4vm1 example]$ sudo ./testLapb server hdlc1  
LapB test program V1.0  
Connecting to LapB  
Connected
```

4.3.2 Start the Client

```
[build@rhel8-4vm1 example]$ sudo ./testLapb client hdlc0  
LapB test program V1.0  
Connecting to LapB  
Connected
```



4.4 Test Results

After around 20 – 30 seconds the test should finish and the results displayed:

```
[build@rhel8-4vm1 example]$ sudo ./testLapb server hdlc1
LapB test program V1.0
Connecting to LapB
Connected
Transfer stats for hdlc0
Received 102401 bytes in 101 frames with 0 errors.
Transmitted 102401 bytes in 101 frames with 0 errors & 0 pauses.
[build@rhel8-4vm1 example]$
```

```
[build@rhel8-4vm1 example]$ sudo ./testLapb client hdlc0
LapB test program V1.0
Connecting to LapB
Connected
Transfer stats for hdlc0
Received 102401 bytes in 101 frames with 0 errors.
Transmitted 102401 bytes in 101 frames with 0 errors & 0 pauses.
[build@rhel8-4vm1 example]$
```



5 Stopping the driver and unloading the modules

To stop the farsync driver and unload the modules, please following the following sequence:

1. Take down any active interfaces
2. Stop the farsync driver
3. Unload hdlc_x25 module
4. Unload lapb module
5. Unload hdlc_ppp module
6. Unload hdlc module

5.1 Take down any active interfaces

```
sudo ifconfig hdlc1 down
sudo ifconfig hdlc0 down
```

5.2 Stop the farsync driver

```
sudo /etc/init.d/farsync stop
```

5.3 Unload the hdlc_x25, lapb and hdlc modules

```
sudo rmmod hdlc_x25
sudo rmmod lapb
sudo rmmod hdlc_ppp
sudo rmmod hdlc
```

5.4 Example stop script

```
ifconfig hdlc1 down
ifconfig hdlc0 down
/etc/init.d/farsync stop
rmmod hdlc_x25
rmmod lapb
rmmod hdlc_ppp
rmmod hdlc
```



+44 (0)1256 330 461

info@farsite.com

www.farsite.com