



M7 Modem Application Note

I7 Interface Card

Q7-IP Vyatta/Datum Quick Start Guide

Revision History

Rev 0.2	2-7-2012	Initial Release.
Rev 0.3	4-29-2012	Added M7 series modem information.
Rev 0.4	5-9-2012	Expanded information.
Rev 0.5	11-3-2012	Expanded information.
Rev 0.6	2-15-2013	Changes in Section 3.6. for re-installing from scratch.

1.0 Q7 IP Interface Overview

The Datum Systems' M7 series satellite modems can include optional IP – Ethernet based data interface cards. The initial M7 I7 IP interfaces use a sophisticated processor consisting of an x86 based embedded computer on a module which is plugged into the interface card. The module is referred to here as a “Q7” because it follows a computer interface standard known as Q-seven. The Q7 is a complete x86 based embedded Linux computer including multiple interfaces. The interface board with Q7 module is housed entirely within the M7 (I7) series of Datum Systems satellite modems. The smart IP interfaces based on the Q7 currently in production are:

1. I-7 - M7 Modem High Performance IP interface.

The purpose of the I7 with Q7 processor is to act as an interface between IP Ethernet networks and satellite modems as part of point-to-point or point-to-multipoint data transfer. The I7 is not a fixed dumb interface, it is a full sophisticated router class device and therefore requires configuration.

The purpose of this Quick Start Guide is to simply get you started with IP interface cards that use the Q7 processor based platform for routing and bridging. Detailing the full capabilities of these IP cards is beyond the scope of this short guide. Our goal here instead is to:

1. Introduce you to this card,
2. Show initial set up the or I7 IP addresses allowing access for full configuration,
3. Show a few initial Vyatta configuration methods.
4. Show command line methods of modem control from these cards.

1.0.1 More about the Q7 Processor Cards

The Q7 Linux software starts life as a standard Vyatta Router which normally contains virtually all of a standard Debian OS overlaid with the Vyatta Router system. We modify both Debian and Vyatta for this application and also add modifications to support the M7 series of modems. The Vyatta Router provides an industry standard open source and very consistent router facility. The modem modifications consist of changes to the Linux/Debian drivers to support our custom embedded board and modem interface plus some changes to the Vyatta system and finally added programs specifically intended for monitor and control of the specific host modem. Everything added except some tweaks to the underlying kernel to aid in reboot is accomplished by standard Debian packages and the installed “apt” package management system. Users of Linux and Vyatta should feel comfortable in this environment while users standard routers should also adapt very quickly.

The Q7-IP provides the following features:

- Powerful x86 Atom processor based. Up to 1.6 GHz and dual virtual processors.

- Robust Debian Linux Operating System – currently modified Debian “Squeeze”.
- Industry Standard Vyatta Router overlay on Linux for IOS/JunOS like control. One major advantage of Vyatta is that it provides a clean single point configuration interface and saved configuration file.
- Datum System's modifications and additions to both Linux and Vyatta providing modem command line and web control.
- Custom applications become relatively easy with multiple built-in user available programming languages including Bash, Perl, Python, C and C++ compilers.

Although more complex and expensive than the previous generation SnIP this system offers significant advantages. Not only does it use an industry standard router subsystem, but the full power of Debian Linux is available including many languages (like Python), standard compilers and the Debian packaging system which allow a user to install almost any program made for Linux. An installed system is even capable of compiling its own programs.

This Tech Note describes the basic resources and procedures necessary to initially install, configure, modify, and update the Q7 Vyatta system and Linux Kernel as necessary.

1.1 Naming Conventions Terms and Definitions

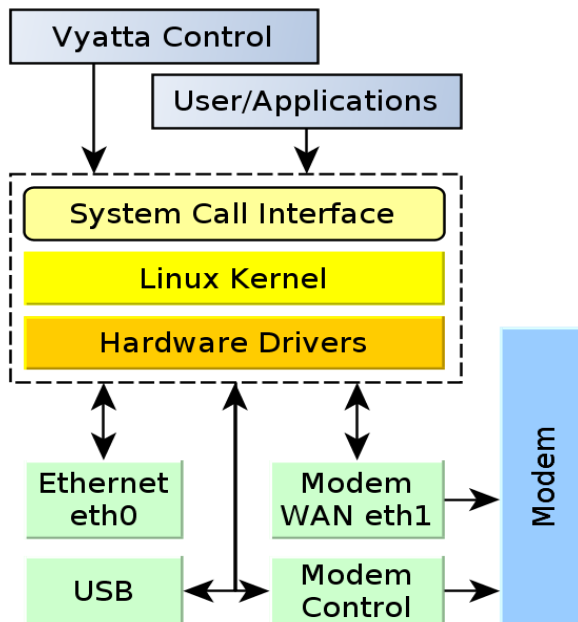
For the purposes of the remainder of this document the conventions for naming the different portions of the Linux operating system and technical terms are given below in short form. Almost everything here is standard terminology and more information is available on the web:

- **“Kernel and Kernel Space”**. These refer to the Linux kernel itself and the modules that are compiled with the kernel to be loaded as part of the kernel. The current kernel is Version 2.6.35 minimum and some 3.x systems are in testing.
- **“User and User Space”** Multiple types of programs can exist in the user space. The user space is where all the files that a user can directly interact with reside. This includes the control programs and configuration files for the routing and bridging functions as created by Vyatta.
- **“Root Filesystem”** The Root Filesystem consists of the kernel modules not built into the kernel, common and user libraries and all of the user space programs and configuration. The Datum Systems' implementation is based on the Debian “Squeeze” distribution of Linux with Vyatta router overlay and Datum Systems additions specific to satellite modem use.
- **“Packages”** are the Linux term for software programs and drivers. The entire Linux system consists of a small set of kernel and base programs with many packages added to provide the particular capabilities needed for its intended use. Most of the drivers and virtually all of the user space programs are packages. Packages can be installed removed and upgraded via the package manager. In the Q7 X86 card the package manager is named “APT” and common package functions are performed using the apt-get command line directives.
- **“Package Repository”** is a web location containing groups of packages which can be installed in the Linux system via the package manager. Two are necessary, the Vyatta repository and the Datum Systems repository although there are many more available including the standard Debian repository.
- **“I7”** These are the base IP interface cards for the M7 modems. The Q7 processor is plugged into these base “carrier” cards as the processing element.
- **“LAN and WAN”** are terms for the Local Area Network and Wide Area Network interfaces standard on the I7. The LAN is always the standard Ethernet TCP/IP Gigabit Ethernet port (GbE) on the rear panel of the interface card. It's interface name is **“eth0”**. The WAN is the internal custom interface port which connects to the modem. Because it appears to Linux and Vyatta as a standard Ethernet port its default name is **“eth1”**, although that can be changed.

- **“Static and DHCP”** Are two possible ways of configuring the LAN side IP addresses and masks. By default the eth0 LAN is configured with a static IP address.
- **“Console and Terminal Sessions”** These refer to a command line connection to the I7. The console is a standard serial RS-232 connection on the rear of the interface card which can be connected to a computer running a standard terminal emulation program such as Hyperterminal or Putty or Linux minicom and other programs. A terminal session is a more generic term and can refer to virtual terminal sessions connected via TCP/IP in either Telnet or SSH mode. Both present the standard Linux/Vyatta command line interface.
- **“Bridge”** is the standard network term for an OSI model Layer 2 process that is intended to connect two segments of the same LAN together having common IP addresses within the same LAN mask. The bridge function only looks at MAC addresses. A bridge can be compared to a smart switch.
- **“Router”** is the standard network term for Layer 3 processes intended to direct and control traffic in a transmission system between two or more networks with different addresses. It operates on the OSI Model Layer 3, working with IP Addresses. Commonly routing also includes the concepts of multiple processes including NAT, traffic shaping, QOS, etc.
- **“daemon”** is a term for a program that runs separately in the background and produces little or no output to any of the terminal sessions running, instead sending output to log files. There are a significant number of daemons always running in Linux, but in a sleep state waiting for some input that requires their attention.

1.2 The Debian Linux and Vyatta Operating System

A complete Debian-Vyatta-Datum Systems OS consists of a lot of individual software and program elements installed on top of the base Linux kernel. For the purposes of this quick start guide we will define and show a few of those software, application and hardware elements for reference in the following diagram:



The Vyatta Control and User/Applications are part of what is commonly called the Application Layer and exists in the “user space”, while the kernel and drivers plus the system call interface are part of the “kernel space”. User space programs plus some of the driver modules, those not built into the kernel, are managed by a package manager, in this implementation it is “apt-get”.

Packages of software applications can be installed over the web from several available package repositories such as those maintained by Debian, Vyatta and Datum Systems.

User programs are either compiled by a language like C or are simple text scripts that are executed by interpreters like Bash, Perl or Python. Python is an especially powerful modern language available in this system that is almost as fast in execution as C.

1.2.1 Linux Resources

The Linux Vyatta system in this interface has almost all of the facilities of a full normal PC type computer except that it does not provide a video based graphical user interface or "GUI". Control is either by command line text in a console session (CLI) or a web based browser interface. The initial setup of the web implementation is for control of the Vyatta system only.

You have a lot of resources available however, but you may not be familiar with their use. The Vyatta CLI purposely looks very much like that in a Cisco router and is modeled on the Juniper router CLI. The Linux CLI includes enormous power due to the vast number of programs available plus several relatively easy to use programming languages that are text based. They include the Bash shell language plus Perl and Python, all of which are interactive and usable directly from the command line. The classic vi editor plus a more graphical one named "nano" are also available.

You can also load hundreds of additional programs via the apt-get package manager. For this and other resources in Debian Linux a search of the web will bring up multiple sites that can help.

When you connect and log into the console command line of this system you are initially talking to the Linux system. As we will illustrate later the Vyatta configuration is entered into using the keyword command "configure" and you exit back to Linux using the command "exit". Most Vyatta commands to display information are available directly in Linux without entering the configure mode.

2.0 Initial Installation and Configuration

A complete Debian-Vyatta-Datum Systems OS is installed at the factory, so software installation is normally not necessary. Instructions for re-installing the entire system is given at the end of this document if necessary.

It is highly advised that you also read the documentation at www.vyatta.org to get a feel for configuring and controlling the Vyatta router. Vyatta is a modern up to date router overlay on Linux comparable to Cisco or Juniper IOS and using similar language and procedures.

It is also recommended that you accomplish all IP and router interface functions using the Vyatta control interface. Note that it is possible to do most of the Vyatta configuration directly using Linux, but this may bypass the wonderful single configuration interface and storage available through Vyatta.

The initial sequence shown assumes a standard factory setup and normally should not require a connection to the I7 rear panel RS-232 console port, just setting an IP address on the modem front panel and then using SSH to log into the I7 system over Ethernet. If you experience any problems setting the IP address or logging in via SSH, then the backup is to connect a PC serial connection to the I7 console port and run a terminal emulation program like Putty at 9600 bps.

The following is a short list of initial configuration items:

1. Connect the two modems into your network as normal and turn on the modems. Modem parameters are configured as normal either via the front panel or remote methods. Make sure that you connect an Ethernet cable to the I7, preferably with access to the Internet if you intend to download packages.

2. Wait approximately 1 to 2 minutes for the Q7 to boot completely. In the meantime determine and set your modem parameters on both ends. If you want to wait until later, the I7 runs a full M7 series modem command set via the “mcl” command line program.
3. **Setting the IP Address** - The I7 interface card is factory configured with a default static IP address, for example 192.168.2.130 or similar, which can be read on the modem front panel display in the interface section. If this is not within your network either set a new IP address on the modem front panel or log into the console port of the Interface board at: 9600,N,8,1 and use Vyatta to set an IP address.

Setting the LAN IP address on the modem front panel is the easiest method to get started with a new modem.- Simply navigate to the interface and scroll right to the IP column and then down until you see the listed IP address. Enter or edit to input the new desired IP address and press enter.

On the modem front panel you must select either the mod or demod tab below the LCD display first, and the “Data Interface” is selected as “I7”. Then you are presented with the “Intf” tab. Selecting the Intf tab will allow you to set the I7 parameters.

For later setting or resetting of IP addresses and all of the Vyatta configuration it is most easily done via the console or an SSH or telnet session **To log in the default user is vyatta and the default password is “vyatta” (or “datum” all lower case.** You can change these and add other users easily.

- The M7 modem will replace the current Vyatta configured first IP address and save the configuration.
- Alternately to enter a new address from the console in Vyatta type: [Note see Section 2.0.1 for more information about IP addresses and masks.]

```
$ configure
# set interfaces ethernet eth0 address aaa.bbb.ccc.ddd/mask-prefix
# commit
# save
# exit
```

4. Once you have a working IP address you can then SSH into the I7 at this new set address. From any computer's command line you `ssh vyatta@<ipaddress>`.
The default user is vyatta and the default password is vyatta or datum.
5. Some sample Vyatta commands: Note the leading “\$” prompt is shown in normal mode while the “#” prompt is shown when in “**configure**” mode.

```
$ show configuration – shows the full configuration.
$ configure – enter the configuration mode.
# show – shows the full configuration from configure mode.
# set – Set a parameter or variable.
# delete – Delete a parameter or variable.
# commit – commits or executes currently entered commands.
# save – Saves the currently committed configuration in the configuration file.
# exit – Leaves the configuration mode.
```

6. You will need to set up Vyatta for your network needs. The Vyatta command line interface is similar to both JunOS and Cisco IOS, but is unique. All the same functions however should be possible. Reading the Vyatta documentation is very helpful.

2.0.1 About LAN IP Addresses and Network Masks

The modem only displays a single IP address and mask for the LAN interface. Vyatta however is capable of having multiple IP addresses for each interface. Vyatta also enters and displays

network masks in the CIDR mask-prefix format – that is the class C mask of 255.255.255.0 is shown and entered as “/24”. This mask-prefix refers to the number of leading 1's in the mask.

The standard operation of the M7 modem front panel entries is to replace the first address in the Vyatta configuration. If there is no address currently then the front panel entry will add that as the first address. The polling software does this by detecting the front panel change and first deleting the existing first address and then adding the new one entered. That is also the normal Vyatta method for replacing an IP address, delete the current and set (add) the new one.

In Vyatta if an interface is to get its IP address and mask via DHCP then its address is actually set to “dhcp”. On the M7 modem when the front panel “dhcp” parameter is set to “1” or enabled (M7 modem only) then Vyatta sets its IP address to “dhcp” and forwards the obtained leased address to the modem's IP address parameter. If DHCP is then disabled it will retain the leased address. To set a new static IP address you enter the new desired address either on the front panel or in Vyatta's configure mode.

The comparable IP entries on the Modem front panel vs the Vyatta configuration are shown in the table below. In Vyatta a single configuration entry encompasses the 3 controls on the modem from panel of the M7.

Vyatta Configuration	M7 Modem FP Display
Address: 192.168.2.131/24 [Optional additional addresses]	DHCP: Disabled IP Address: 192.168.2.131 Mask: 255.255.255.0
Address: dhcp Note: DHCP lease address sent to modem.	DHCP: Enabled IP Address: 192.168.4.109 Mask 255.255.255.0

Although the IP address coupled to the modem's front panel display is typically shown as “eth0” in this document, when the interface is set to bridge mode the eth0 and eth1 interfaces are “enslaved” to the bridge-group which normally has a name “br0”. Therefore in bridge mode setting an address on the front panel sets br0, not eth0.

2.1 Essential Initial Configuration Tasks

The Debian/Vyatta/Datum system is initially shipped with a default set of parameters including the standard user “vyatta” with password “vyatta” or “datum”, SSH enabled, a standard router complement including RIP BGP, several repository pointers and others. Additional configuration will probably be required to operate within your network topology.

The configuration is initially set up as a minimal router architecture. That is port forwarding is enabled and only static routing is enabled, but only set up for routing between the two modem internal WAN ports.

If your configuration is relatively simple, such as connecting a remote office via a satellite link then bridging may be the easiest configuration option. Bridges require that the network on both ends of the link share the same network mask. For most configurations more complex routing is probably the best connection type. Routing is Vyatta's specialty and much of their documentation centers around setting up sophisticated routing schema.

Currently one service is added to the Vyatta configuration suite – that is modem polling. To see all of the configuration settings you use Vyatta's “show configuration” command. Vyatta includes very good command completion much like Cisco IOS, so to use this command you could just type “sh” and the tab key which will complete the “show” with a space after it. If you hit the tab key again it will show all of the possible commands appropriate for show. If you type “conf” and tab again it will complete the command and then hitting Enter will show the current configuration. In

Linux a display like this is long and includes methods of moving up and down via the arrows or page up and page down. The space bar works like page down and the “q” key quits.

Note in the “service” category there should be an entry for “modem poll”. If it was not there then the method for enabling this, or other services would be similar to: Note that some short explanation is shown after the actual commands which are in bold.

```

$ configure – enter the configuration mode.
# set service modem-poll
# set other parameters desired.
# commit – commits or executes currently entered commands.
# save – Saves the currently committed configuration in the configuration file.
# exit – Leaves the configuration mode.

```

2.2 Further Vyatta Configuration

The default configuration does not include telnet, but SSH instead. To add telnet as an additional service you would follow the above configure process using the set command:

```
# set service telnet
```

If for some reason ssh was not already enabled the command to enable it is:

```
# set service ssh
```

Most users will want to use a custom user name and password plus other values in their systems. Customization is relatively easy in Vyatta and the following is a sample of several parameters you may wish to set. Of course use your own values for all the parameters, for example replace <myname> with the short user name – maybe “lucy” and the password with your desired one.

```

$ configure
# set system gateway-address 192.168.15.1 //[use your gateway]
# set system name-server 205.171.3.25      //[use your own name-server]
# set system host-name test1//[use your own desired host name]
# set system time-zone US/Pacific        //[use your own time zone]
# set service https
# set system login user <myname> authentication plaintext-password mydoghasfleas
# set system login user <myname> full-name "Your Full Name"
# set system login user <myname> level admin
# commit
# save
# exit
$

```

Note here that we gave this new user “admin” level privileges. This allows this user to use “sudo” to execute commands normally reserved for the root. It also allows them to become the root by typing “sudo su” where the \$ prompt will be replaced by a “#”. When finished use “exit” to go back to the role of normal admin user.

In Vyatta all configuration involves going into the configure mode, setting or deleting parameters, committing those parameters and finally saving the changes to the configuration file. Vyatta offers

a very complete set of commands allowing configuring the Q7 as a router or bridge with features such as filtering, fire-walling, QOS, NAT, etc. See the Vyatta documentation for more information.

2.3 Modem Configuration

Two modem control packages are installed by default and able to be updated. In the I7 they are the “m7-control” package and a future “m7-web” package. The control packages contain a group of Linux user space programs necessary to communicate with the modem including a polling daemon that functions as a keep-alive to let the modem know it is present and ready.

Web control of the M7 is normally done via the modem's controller card but is also planned in a future release to be available via a browser from the web server in the I7-Q7. It is used by pointing to the configured LAN IP address of the Q7 from a web browser. Command line control is provided by the modem control language “mcl” command which is part of the m7-control package.

If the “m7-control” package is re-installed to update it with a newer version then the polling service will automatically be restarted to use the newer program version. If for some reason you should need to restart the polling service, it is done directly in Linux rather than in Vyatta using the following command

```
$ sudo service m7-polld restart
```

Also see the note in software re-installation about setting Vyatta to run the polling daemon automatically at boot time. This is normally set up at the factory but in case polling has been turned off for any reason it can result in the modem performing an automatic reset periodically. The Vyatta command sequence to persistently enable polling is:

```
$ configure
# set service modem-poll
# commit
# save
# exit
$
```

2.3.1 M7 Modem Configuration Using “mcl”

The I7 interface uses a modem control language named “mcl”. Although mcl uses a similar command line form to the m500ctl language used in M500 modems, its command set is significantly different than that used in m500ctl. The M7 modem protocol allows complex binary commands which are custom built to get exactly the information needed rather than the larger groups of commands used in the M500 protocol.

To use “mcl” the general form of the command is:

```
$ sudo mcl [option switches] [unit-address] card: mnemonic-command [parameter]...
```

The simplest form uses none of the optional elements in brackets [] which defaults to the address 0 and a request. An example would be:

```
$ sudo mcl -v unit: id1 id2 tx: frq lvl bit rx: frq lst bit
```

Which gets the id for the cards in slots 1 and 2 plus frequency, level and data bit rate for the modulator and demodulator. The -v option prints the output in verbose mode. Although this may look complex initially it is very straightforward after a little use.

Mcl uses keywords to define what card and parameters are desired. In the example above the cards are unit, tx, and rx. Although not necessary it makes the command sequence clearer if those are followed by colons. The parameters themselves are normally 3 letter codes like frq for frequency, bit for data bit rate and lvl for level. The response to the above command might be:


```
mike@vyatta:~$ sudo mcl -v unit: id1 id2 tx: frq lvl bit rx: frq lst bit

Modem Unit Card in Slot: 0
  Slot A ID = 'Std IF Modulator'
  Slot B ID = 'Std IF Demodulator'
IF Modulator Card in Slot: 1
  Mod CXR Frequency = 60000000
  Transmit CXR Level = -3000
  Mod Send Data Rate = 8000000
IF Demodulator Card in Slot: 2
  Demod CXR Frequency = 80000000
  Demod Rcv Cxr Level = -2891
  Demod Bit Rate = 2000000
```

The mcl can be used to set parameters as well as viewing them. For example the transmit frequency and data rate could be set with the command:

```
$ sudo mcl tx: frq 71250000 lvl -2600 bit 2048000
```

which would set the frequency to 71.25 MHz, the level to -26.00 dBm and the data rate to 2.048 Mbps. Note that in the mcl parameters are viewed and set in their least significant increment, so frequencies are in Hz, level are in 0.01 dB increments, data rates are in bits per second, etc.

To see more help information on using mcl use the Q7 console or SSH session and type “mcl” without any parameters. There is also an mcl “man” page available and a separate I7 mcl User Guide.

2.3.2 Displaying a List of Mnemonic Codes

The list of available codes for each of the cards can be shown by using the option “--codes” or -n, for example:

```
sudo mcl --codes tx
```

will display all of the codes for the transmit modulator cards with the short description. If you want all of the code values you would use the “-t” option, and by adding the verbose modifier a human readable format is shown. The full command for this last option would be

```
sudo mcl -vtn tx
```

where the “vtn” is short for verbose, test codes. If you change this to “-qtn” then the 3 letter codes and description plus the current value of each transmit parameter is show. There is a single option code combining the “-q”, “-t” and “-n” options, which is “-a” or “--all”. To display all of the available transmit modulator parameters you would use:

```
sudo mcl -a tx
```

You can see a basic set of help for the mcl program by typing “mcl” on the command line and hitting enter with no arguments or parameters. This follows standard Linux conventions. The use of option switches that have a single letter following a dash and can be combined with other single letter options using only a single dash is also a Linux convention as is more verbose option switches that use two dashes (“--”).

2.4 SNMP

Standard full Linux router Net-SNMP is included by default in the I7-Q7 interface. This includes all of the standard Linux MIBS, which are locally available in the /usr/share/snmp/mibs folder. These standard MIBs are usually pre-installed in most SNMP browsers. SNMP for the modem parameters is available via the modem's own IP interface.

To enable the SNMP server you would use the Vyatta configuration system similar to the above examples. Here is a sample configuration which sets the community name “public” to read only and the community name “private” to read/write authorization:

```
$ configure
# set service snmp
# set service snmp community "public" authorization ro
# set service snmp community "private" authorization rw
# set service snmp description "Sub-Hub 1"
# set service snmp location "On the Beach"
# commit
# save
# exit
$
```

This configuration should allow access to snmp from any interface at the interfaces IP Address. You can also set another snmp listen address with the set service snmp listen-address command.

2.5 Common Linux Tips

Since we are running a standard Linux distribution there are many methods in use for various situations that can be found by searching on the web. Some of those are listed here.

2.5.1 Using the Command Line History

The Linux shell keeps a fairly long history of previously typed commands and allows showing and editing them, so you don't need to type many commands more than once.

- Accessing command history – Press the up and down arrow keys to scroll back and forth through the command history. You can edit commands too and hit enter to execute them.
- Searching command history – Hold the Ctrl key and hit “r”. This allows you to now start typing any part of a previously issued command which you can then use or even edit.

For more tips do a Google search for “Bash Command History”.

2.5.2 Getting Program Help and Information

Most Linux programs include a manual for help. You get this by typing:

```
$ man <program name>
```

You can find a programs location by typing:

```
$ sudo which <program name>
```

Many commands may produce output too long to easily view on one screen. It is possible to show less than the full output at one time and allow scrolling up and down in the produced results using a pipe to the “less” command as shown here:

```
$ sudo <command> | less
```

2.5.3 Auto Connection Logout

If you want to have a connection to any console port automatically be logged out with inaction for a specified period of time here is a general procedure. First log into either the physical console or a telnet/ssh session and issue the following commands:

```
$ sudo su
# echo "TMOUT=300" >> /etc/bash.bashrc
# echo "readonly TMOUT" >> /etc/bash.bashrc
# echo "export TMOUT" >> /etc/bash.bashrc
```

```
# exit
$
```

Note that in most Linuxes there are `etc/profile` files and/or `etc/bashrc` files. In Debian `bash.bashrc` is sourced by `etc/profile`. These are system wide profiles for anyone using the system, while to make similar changes to only one user then similar changes would be made to the `.bashrc` file in the user's home folder.

This set of changes modifies the `.bashrc` file for all users from anywhere including the physical RS-232 console port or any virtual port. The timeout is 300 seconds or 5 minutes. It can be set to anything else if desired.

3.0 Updating the Software – the Package Manager

Standard Linux packages that can either be included in the initial load and/or installed and updated in the factory or field via “apt-get”. Adding or updating software depends on access to the Internet where the package repositories are located. Therefore to perform any of the following actions you must have a connection to the Internet. The standard form for showing if a package is currently installed might be:

```
$ aptitude show <package-name>
```

Once those packages are installed on the interface card, one can upgrade the packages by the commands:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Note that the update command does not update software, just the list of available packages.

To upgrade just one of our packages, or any single package you use the install method even though it is already installed. For example to update the `m7-control` package you simply say:

```
$ sudo apt-get install m7-control
```

There is an enormous amount of information available on the web on using the apt package manager as it is common to Debian, Ubuntu, Mint and many other Linux distributions.

To see a full list of all installed packages use the `dpkg` command as show here:

```
$ dpkg --get-selections
```

Note that is two dashes before `get-selections`

This will produce a fairly long listing.

3.6 Re-Installing the Software

If for some reason the kernel or filesystem is completely destroyed or a full system upgrade is required it is possible to relatively easily re-install the entire system from scratch. A minimum 4 GB USB flash drive is required for system installation in addition to the normal 4 or more GB flash drive supplied with the IP card.

3.6.1 Creating a “Live-USB” for Installation

The Datum Systems software is provided as a standard “ISO” file which could be loaded onto a CD ROM as a “live-CD” for installation, but the standard method is to load it onto a secondary USB flash drive that acts like a live-USB. First download the current Datum Systems ISO file from www.datumsystems.com

To create a live-USB drive from the Datum ISO file there are very good instructions on the Ubuntu main web site at

<http://www.ubuntu.com/download/help/create-a-usb-stick-on-windows>
or for Linux computers at
<http://www.ubuntu.com/download/help/create-a-usb-stick-on-ubuntu>

You will be substituting the downloaded Datum ISO name instead of the Ubuntu one however. It appears that Ubuntu versions later than 10.04 may have changed the live-USB program so that it no longer works with ISO files other than Ubuntu itself. A new procedure based on the open source program "Unetbootin" is shown here.

1. Download and install the proper version of Unetbootin for the computer you will be operating from. It is available for Linux, Windows and Mac. Currently the web address is: <http://unetbootin.sourceforge.net/>
2. Download the latest customized version of Vyatta for the I7-Q7 card from Datum Systems web site at <http://datumsystems.com/downloads>. The current version is named "**vyatta-6.2-ts7_custom-130214.iso**". The number after custom shows the date created, yymmdd, in this case Feb 14, 2013.
3. Start Unetbootin and select the "from iso" button.
4. Plug in a flash drive to hold the Vyatta Live-USB. This USB flash drive must be at least 2 GB in size. Unetbootin should recognize the drive and display it as a choice in the pull down selection. If it does not find the USB drive then it may not be formatted as FAT 32 and must be reformatted. In Linux it is easiest to use Gparted.
5. Select the Datum Vyatta iso file previously downloaded.
6. Start the copy. When finished you can unplug the new Live-USB.

Caution: Before beginning this process you should turn **off** the M7 modem's automatic "Watchdog" based reset of the I7 interface if it does not see polling activity within approximately 2 minutes of power-up. To do this you should turn on the modem and select the I7 card then go to the "**Interface**" selection on the modem front panel. Moving to the "**IP**" column and down to select the IP "**Q7 Reset Control**" and set it to "**Manual**". Without doing this the modem will repeatedly attempt to reset the I7 controller.

Next connect your computer's serial port, or USB to Serial adapter, to the I7's physical console port, dB9, on the I7 card. The connection is at 9600,N,8,1.

With the modem turned off the live-USB created above is plugged into one of the rear panel USB ports on the I7 and all other USB drives are removed, then the modem is turned on and the Q7 is booted from that live-USB drive. This runs a system image that is fairly complete but not usable for anything but testing and installation because it does not store information. Next the final destination USB drive is plugged into the other USB port on the rear and the live session is used to install the system onto the destination drive. First we determine which is the drive just plugged in. Type:

```
$ ls /dev/sd*
```

Which should show two drives – one will be /dev/sda and /dev/sda1 which is the drive that you booted from. The second will probably be /dev/sdb or /dev/sdc which should be the drive you just inserted. Notice that Linux identifies this type of hard drive by a designation of "sdx" where the "x" is a unique letter for each drive. There are also partitions on the drives which would appear as sdx1, sdx2, etc. You can verify that we are going to install on the correct drive by now typing

```
$ dmesg | tail
```

which should show the last 10 log messages and should display some lines about a USB drive just inserted. They should also refer to sdc but it could be another letter depending on exactly how your system is set up. Just remember this drive designation – e.g. "sdb" or "sdc". You will use it when the installation asks which drive to use.

With that knowledge you type

```
$ install-system
```

- Note on some later Vyatta setups it might not have the hyphen and be “install system” instead. Follow the instructions to put the system onto the sdb or sdc drive as the destination.

If a previously used USB drive is used for the destination then the install-system process will offer to recover and use the encryption keys and configuration files from the previous version. This is highly recommended to avoid having to re-enter everything.

If you feel that any of that may have been corrupted or compromised then it is still easy to just start from scratch.

During installation several questions will be asked. Make sure that you answer “sdb” or “sdc” in the appropriate places. No data is written to the sda drive containing the live-USB. Answer No to the RAID configuration question. We normally use a user name of “vyatta” and password of “datum” as the default. Also select the Auto Partition when asked. Most questions will show a default answer in brackets after the question, and pressing enter will select that default.

Note: *If you accidentally answer some question incorrectly it is usually possible to type “Ctrl C” at any point before the “copy to disk” operation to abort the system install and you can start over.*

When the actual install begins it should take about 8 to 10 minutes to complete. It is normal for the copy to drive display to stop for awhile at several points for example 13%, 32%, 43%, etc.

The last operation will be to set up the GRUB bootloader – be sure to set it up on the new installation drive sdb or sdc.

3.6.2 Post Installation Setup

After installation to the final destination flash drive is complete you can turn off the modem power and remove the live-USB. Then power on with only the newly created flash drive, log in and you’re almost ready to go. Even if using a previously configured system you will need to load the modem specific packages and enable them, in this case for the M7 modem.

Have a look at the configuration and see if this version has the appropriate repository settings for the M7 modem series already configured. If not do the following

In Vyatta to be persistent the new Datum Systems repository is entered like this.

```
$ configure
# set service ssh
# set system package repository modem
# set system package repository modem distribution m7
# set system package repository modem components main
# set system package repository modem url http://dsi-eng.com/packages/m7/
# commit
# save
# exit
$ sudo apt-get update
$ sudo apt-get install python-serial
$ sudo apt-get install m7-control
```

Note that the python-serial package was added to this repository for convenience. After installation of the modem control software package, Vyatta must be configured to run the modem polling daemon beginning at start-up.

```
$ configure
```

```
# set service modem-poll
# commit
# save
# exit
$
```

You would now go to the normal configuration as shown in Section 2 to set up your IP addresses, gateway-address, DNS, etc.

In some more rare cases you may also need to install the information for the standard Debian repository as follows. This repository contains packages needed for compiling and the python-serial package used by the mcl program.

```
$ configure
# set system package repository debian
# set system package repository modem distribution squeeze
# set system package repository modem components main
# set system package repository modem url http://mirrors.kernel.org/debian
# commit
# save
# exit
$ sudo apt-get update
$ sudo apt-get install python-serial
```

End of Document.