

μSmartDigi™ : an APRS® Digipeater and D-Gate™ D-STAR Gateway

Rich Painter, ABØVO
Painter Engineering, Inc.
8470 Swan Rd
Black Forest, CO 80908
<http://usmartdigi.com/>
painter@usmartdigi.com

Abstract

The μSmartDigi is an expansion board for the TNC-X KISS-mode Terminal Node Controller that provides significant computing resources for various digital communications applications. The μSmartDigi eliminates the dedicated PC and laptop computers commonly used in amateur radio applications. The first two applications, the μSmartDigi APRS Digipeater and D-Gate D-STAR Gateway are described in this paper.

Keywords

μSmartDigi, uSmartDigi, microSmartDigi, D-Gate, D-STAR, APRS, Digipeater, Gateway, TNC-X, Icom, dsPIC, Microchip

Introduction

The requirement for what is now the μSmartDigi was proposed by Dave Davis, WB5WIA to solve a number of problems with the existing APRS RF repeater networks. The author expanded these requirements while advancing with several valuable features never before envisioned. The μSmartDigi was also designed to use the existing TNC-X KISS modem. This modem was designed to accept expansion printed circuit boards for applications just like this.

Existing APRS Digipeater Problems

The μSmartDigi had to solve the following problems:

- Some digipeaters did not detect duplicate packets correctly
- Existing digipeaters had no provisions to limit excessive and multiple WIDEn
- Some users were not configuring appropriate digipeater paths

Totally New Digipeater Features

Besides solving these existing problems the author proposed several new features never before envisioned:

- Eliminate the dedicated PC or laptop computer
- Operate on low power (8-16VDC, ~200mA), small and portable
- Filter packets based on user-configurable rules such as source (call) and destination fields and geographic location with a rich set of commands

- Provide a user-configurable filter for RELAY and ignore TRACE
- Provide No Source Routing (NSR) as an option to supplant digi paths or some alternate method to provide the Layer 2 support
- Use an optional GPS receiver for static or dynamic geographic position.

µSmartDigi Basic Design

The µSmartDigi was designed to use Microchip's dsPIC30F6012 DSP microcontroller. This is a 16-bit, 30 MIPS CPU with numerous devices, including a DSP, 2 serial ports, 144kB program, 8kB RAM, 4kB EEPROM, on a 14mm square 64-pin chip. The printed circuit board (PCB) was required to fit into the expansion board space for both versions 1 and 2 of the TNC-X. The PCB was designed to also accommodate the soon-be-released Microchip 33FJ family. These new chips sport faster clock speeds and more memory. This µSmartDigi system is designed in such a way to easily allow uses and applications to be developed not currently envisioned.

In fact, a new application has already been proposed and implemented- the µSmartDigi D-Gate D-STAR Gateway. This will be described later.

Figure 1 is a photograph of the µSmartDigi PCB version 1. Provisions for the Microchip 33FJ can be seen as the unpopulated pads, primarily the 5v to 3.3v regulator U3. The TNC-X provides via J2 5v, ground, serial transmit and receive to the TNC-X and the main and auxiliary RS232 ports. J3 is used to set the receive and transmit configuration for the serial ports. J4 is the 5-pin Microchip ICD2 programming connector.

The device is flash programmed using the ICD2 through J4. Firmware updates in the field are expected to use the µSmartDigi PC Utility that communicates using the RS232 serial port. However, users having a Microchip IDE and ICD2 can also use them with J4. Firmware is provided in Intel Hex format, required by Microchip's IDE. PC Utility, firmware and documentation updates are provided on the µSmartDigi.com web site.

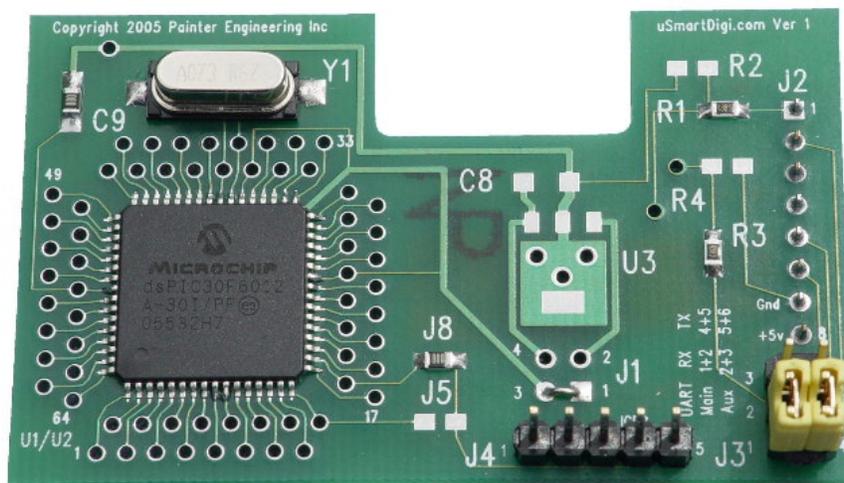


Figure 1. µSmartDigi PCB Version 1

Software for this system consists of imbedded code for the chip and a PC Windows Console Utility program to process offline Rules and Configuration files and download firmware updates to the device. Both the PC and imbedded software are written in C. Microchip's Integrated Development Environment

(IDE) and GNU-based C compiler are used. The software for the μ SmartDigi is native, functioning without a formal operating system.

This μ SmartDigi software provides all of the device drivers, interrupt processing and applications in a well-structured manner to allow easy application changes. Power is conserved by placing the CPU in a low-power idle state when no processing is required. Numerous subprogram modules support AX.25 and KISS packet decoding and encoding, GPS NMEA decoding, serial port management, CRC, Rule processing and other functions needed for the basic platform. Currently there are approximately 16,000 lines of C code (wc *.ch) for the chip and 8,000 for the PC Utility program.

Configurations and Rules

The Configuration Parameters and Rules are configured using the PC Utility program and optionally Configuration Parameters can be programmed directly by communicating with μ SmartDigi using a serial terminal or emulator. The Rules require a sizeable amount of processing to prepare them for saving in the μ SmartDigi EEPROM. Currently, this processing only exists in the PC Utility but may in the future be duplicated within the μ SmartDigi when larger chip memories are available. The Rules and Configuration files can be created and edited with any ASCII text editor, making them portable and easy to manage.

The Rules and Configuration Parameters are permanently stored in the μ SmartDigi EEPROM until altered using the PC Utility (Rules and Configuration) or the μ SmartDigi Monitor command for the Configuration Parameters.

The Configuration Parameters define numerous settings for baud rates, WIDEn path parameters, GPS existence, duplicate packet window time, RELAY denial, non-position packet handling, Call, SSID and location by latitude and longitude (if not using a GPS).

An example Digipeater Configuration file might contain:

```
# For use with the LA APRS Test Data from Scott Miller
call ab0vo // call
ssid      3           // my ssid
posit N34d W117.8d
havegps n
host      57600
gps,4800
tnc 19200
nsr 0
relay n           //california option
widemax 3
widetotal 6
dupewin 30
dropmang Y      #default
haltonerr 0     #default
nonaprs drop
```

See Appendix 1 for the details of the Configuration Parameters.

The Rules file consists of ASCII text containing comments and rule commands in a reasonably free format. The following rules are currently supported:

- Actions are *DROP* or *PASS*
- Implicit rule (*DROP* or *PASS*)

- Match *SOURCE* or *DESTINATION* fields with optional wild character *
- Match geographic locations (*CIRCLE*, *COMPASS*, *RECTANGLE* and *SECTOR*)
- Flexible Latitude, Longitude and Angle format specifications (*Colon*, *Dotted* and *Degrees-Minutes-Seconds*)

An example Rule file might contain:

```
pass impl
drop dst ab0vo
drop dst ID*
drop sector 180d, 270d, 0
drop compass E N34d, W117.8d
```

See Appendix 3 for the details for Rules and Angles.

µSmartDigi Monitor Commands

The µSmartDigi provides a rich set of Monitor commands available to the user through the ASCII RS232 serial port. These allow the user to set and alter Configuration Parameters and the Operating Mode of the device. The Operating Modes consist of HOST, GPS, TNC and MONITOR. Upon application of power, the µSmartDigi automatically enters an operating mode depending on a previously set parameter defining the existence of a GPS unit (attached or not) for the Digipeater (not for the D-Gate). The device has a very small window at startup to detect the special Firmware Download sequence provided by the PC Utility and a 20-second window to allow the user to interact with the Monitor. The user may alter Configuration Parameters and then turn the device over to its normal operating mode. In any case, without any interaction with the Host-Monitor port, the device automatically goes into normal operating mode 20 seconds after powering up. If a GPS unit is configured the device monitors the GPS serial port for NMEA messages during the 20-second startup. If none are received the GPS mode is disabled until the next power reset. Baud rates for the Host and GPS modes are user configurable with certain known values set when delivered.

See Appendix 2 for the details for the Monitor Commands.

µSmartDigi Digipeater

A complete µSmartDigi installed in a TNC-X is shown in Figure 2. The system reads KISS packets from the TNC-X and optionally from the GPS using interrupt-driven I-O. The APRS TNC-X operation will be described followed by the GPS operation.

The Digipeater receives APRS packets from a radio into the TNC-X. The TNC-X decodes the audio to a KISS packet that is sent to the µSmartDigi. Here the KISS packet is verified for consistency and the AX.25 packet within is decoded. Significant error checking is performed to confirm the packet's integrity. Bad packets are dropped.

Next, the digi path is examined and the parameters relating to the path are compared to it. Exhausted paths are dropped. If RELAY is not allowed yet found the packet is dropped. Besides the powerful position-based Rules the µSmartDigi has a method to limit excessive digi paths. Two Configuration Parameters, WIDEn and WIDE Total provide the user the ability to set limits for these within the path. WIDEn is the maximum WIDE value that can be found anywhere in the path, since there can be more than one. WIDE Total is the maximum sum of all WIDE and WIDEn in the path. If either of these is

exceeded then the packet is dropped. TRACE is ignored. At this stage the path is altered as if it was going to be PASSEd including the setting of the H bit within the HDLC frame.

Duplicate checking is performed next. A 16-bit CRC (CCITT) is computed from the SOURCE, SOURCE SSID, DESTINATION and INFORMATION fields. This is compared to the recent list stored in RAM for a match within the user-specified time window. If not there or outside the window this is added to the list and the packet moves on to the next check. Otherwise it is deemed a duplicate and the packet is dropped.

Next, the decoding of the APRS information is done, again with lots of error checking. Bad packets are dropped. If the user requires valid positions and one is not found the packet is dropped. The position information types currently supported are \$, !=, /@, ` \ and } (supporting D-Gate due out shortly).

Once the position information is decoded the Rules can be processed. Most of the rules currently rely on geographic position. Each Rule is compared to the packet until an Action is taken (short-circuit logic). The user can configure positive or negative logic for the Rules. The Implicit Rule determines what happens if all rules fail to match. The configured Action (PASS or DROP) for a matching rule determines its disposition.

If a packet PASSes the Rules a new AX.25 HDLC frame is constructed, then a KISS packets is constructed and lastly it is sent to the serial port of the TNC-X for transmission.



Figure 2. μ SmartDigi and TNC-X

Using interrupt processing, the μ SmartDigi reads both the TNC-X and optional GPS unit. The GPS unit must provide NMEA RMC or GGA messages to be processed. This allows the user to utilize the μ SmartDigi in a portable or mobile fashion, possibly for emergency operations, where the user will not have to program the μ SmartDigi with its base position. Future μ SmartDigi may use the time provided by the GPS and offer time-based Rules such as applicable for different times of the day. Most GPS units provide NMEA message output every second. The μ SmartDigi is programmed to only read the GPS every 30 seconds to conserve power.

Some elements of No Source Routing have been implemented but not made functional at this time. It is envisioned that in the near future when the community has settled on a Layer 2 routing solution it will be implemented in the μ SmartDigi with a software update.

μ SmartDigi D-Gate D-STAR Gateway

Ray Novak, N9JA of Icom America asked the author at DCC 2005 to consider a new application of the μ SmartDigi platform to create a gateway from Icom's D-STAR digital radios to the APRS RF network. With Ray's help the μ SmartDigi D-Gate quickly came into being. The D-Gate is in beta testing as of this writing. A demonstration system was shown at Dayton's Hamvention and Dallas' HamCom earlier this year. Production units will be available by the time of publication.

The D-Gate consists of the μ SmartDigi, TNC-X, a D-STAR radio and a conventional analog radio. D-Gate firmware is loaded into the μ SmartDigi, requiring no hardware change. Due to the nature of this configuration (two radios and no GPS connected to μ SmartDigi) the two external TNC-X serial ports are configured in a special split manner to allow a third device, a PC, laptop or other ASCII serial device, to act as a monitor. Figure 3 depicts this type of configuration.

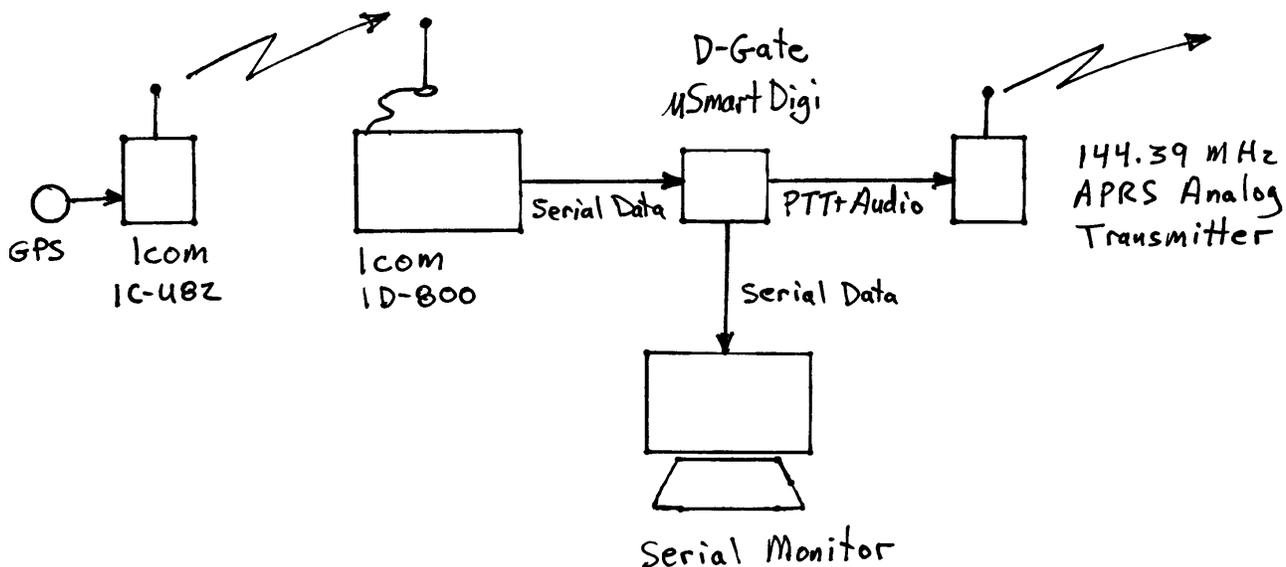


Figure 3. μ SmartDigi D-Gate D-STAR Gateway

The Icom D-STAR radios operate in a digital mode for both voice and data channel simultaneously. With a GPS unit connected to a transmitting radio with a very specific configuration, messages containing GPS data can be received at another D-STAR radio. At this receiving end, a μ SmartDigi configured as a D-Gate Gateway, receives the serial data from the D-STAR radio, decodes and verifies the message, reformats it and sends it to an APRS RF radio connected to the μ SmartDigi's TNC-X.

The D-Gate has several features different than the Digipeater yet most in common. When a D-STAR radio configured in this manner transmits, the GPS data and CALL message are sent too. This can happen many times per minute. Thus a special, fixed, 10-second window is implemented in the D-Gate to ignore any transmission from the same source within this window regardless if the message content (position) has changed. The D-Gate also has a Configuration Parameter to configure the digi path where the Digipeater does not.

Next, the normal Duplicate Packet Check and Rules processing is then applied just as in the Digipeater.

The D-STAR radio configuration is critical to make this gateway function correctly. The GGA and RMC NMEA sentences **must** be configured to provide the gateway the required GPS information. Pete Loveall's Message Calculator web page provides input boxes for the user to configure their desired parameters. It then displays the exact strings the user needs to configure into the radio, making it easy to cut and paste into the radio configuration program. Here is a sample:

D-PRS Message Calculator

[[Home](#)] [[Up](#)] [[D-PRS Message Calculator](#)] [[D-PRS Symbols](#)] [[DStarTNC2.zip](#)]

This page calculates the necessary GPS message for use with D-PRS. Your browser must have JavaScript enabled to use this page.

Note that the symbol code occupies the first four characters of the GPS message (space padded) and that there is a checksum appended to the text preceded by an asterisk.

The D-PRS CallSSID can be used to look up your station on www.jfindu.net or www.findu.com

MyCall 8 Char Max Right Space Padded	D-PRS Symbol	GPS Message
<input style="width: 90%;" type="text" value="AB0VO 9"/>	DIGI (white center) Symbol: <input type="checkbox"/> Overlay: <input style="width: 50px;" type="text"/>	<input style="width: 90%;" type="text" value="D-GATE TEST"/>
D-PRS CallSSID: <input style="width: 90%;" type="text" value="AB0VO-9"/>		Input into TX Message C1: <input style="width: 90%;" type="text" value="BD D-GATE TEST*71"/>

APRS is a registered trademark of APRS Software and Bob Bruninga, WB4APR.

*Copyright © 2006 - Peter Loveall AE5PL pete@ae5pl.net
 Hosted by [AME Corp.](#)*

The message format for the APRS RF packet is a third-party type signified with the } character and consists of an information field with a textual source, destination and ! type position format that includes the course, heading and altitude. An example conversion might look like this:

```
AB0VO-3>APRS,WIDE1-1,WIDE2-2 <UI Len=69>:
}AB0VO-9>APRS,DSTAR*:*!3901.69N/10440.15W#337/001 D-GATE TEST/A=007587
```

The following are example output log messages of various D-STAR messages:

Example 1- Two-message sequence where the first is gated (PASS) and the second is dropped because it arrived from the same source within the 10-second window:

```
D-STAR:
$GPGGA,163212,3901.6726,N,10440.1415,W,1,05,2.8,2319.4,M,-21.7,M,,*40
$GPRMC,163214,A,3901.6717,N,10440.1413,W,1.7,200.6,140806,9.7,E,A*07
AB0VO 9,BD D-GATE TEST*71
```

```
PASS:
AB0VO-3>APRS,WIDE1-1,WIDE2-2 <UI Len=69>:
}AB0VO-9>APRS,DSTAR*:*!3901.67N/10440.14W#200/001 D-GATE TEST/A=007607
```

```
D-STAR:
$GPGGA,163220,3901.6686,N,10440.1422,W,1,07,2.1,2319.6,M,-21.7,M,,*47
$GPRMC,163220,A,3901.6686,N,10440.1422,W,1.9,211.4,140806,9.7,E,A*07
AB0VO 9,BD D-GATE TEST*71
```

```
DROP: Call 10-sec. duplicate: AB0VO 9
```

Example 2- This message was dropped because it matched Rule 4:

```
drop compass E N34d, W117.8d
```

This rule states to drop any message originating East of a point that was West of the source's Latitude and Longitude:

```
D-STAR:  
$GPGGA,162904,3901.6895,N,10440.1521,W,1,05,5.9,2305.7,M,-21.7,M,,*44  
$GPRMC,162904,A,3901.6895,N,10440.1521,W,0.0,185.5,140806,9.7,E,A*02  
AB0VO 9,BD D-GATE TEST*71
```

```
DROP: Line 12, Rule 4:  
AB0VO-3>APRS,WIDE1-1,WIDE2-2 <UI Len=69>:  
}AB0VO-9>APRS,DSTAR*:*!3901.69N/10440.15W#185/000 D-GATE TEST/A=007562
```

Summary

The μ SmartDigi is a powerful and versatile complement to the TNC-X. It currently features an APRS Digipeater configuration and a D-Gate D-STAR Gateway configuration. At this writing the Digipeater is at revision 2.2 and the D-Gate 2.0 is expected to be out by publication time.

Continued development is planned and new applications are likely. Comments and suggestions for improvements, enhancements and new applications are welcomed at painter@usmartdigi.com.

References and Credits

1. μ SmartDigi, uSmartDigi and D-Gate are Trademarks of Painter Engineering, Inc.
APRS is a Registered Trademark of APRS Engineering LLC.
ICOM is a Registered Trademark of Icom Incorporated.
dsPIC is a Registered Trademark of Microchip Technology Inc.
2. μ SmartDigi web site <http://usmartdigi.com/>
3. TNC-X by John Hansen, Coastal ChipWorks, <http://tnc-x.com/>
4. *AX.25 Link Access Protocol for Amateur Radio*, Version 2.2 Revision 11 November 1997, http://www.tapr.org/pub_ax25.html
5. *Automatic Position Reporting System APRS Protocol Reference*, Protocol Version 1.0, APRS Working Group, 31 August 2000, http://www.tapr.org/aprs_working_group.html
6. *The KISS TNC: A simple Host-to-TNC communications protocol*, Mike Chepponis, K3MC and Phil Karn KA9Q, 1987, <http://people.qualcomm.com/karn/papers/kiss.html>
7. Icom's D-STAR web site <http://www.icomamerica.com/amateur/d-star/dstar2.asp>
8. Pete Loveall's D-STAR DV message calculator <http://www.aprs-is.net/dprscalc.htm>
9. *Eliminating Digipeater (Source) Routing from AX.25*, Pete Loveall, AE5PL, August 2005 http://www.aprs-is.net/nosource_routing.htm
10. *Intel Hexadecimal Object File Format Specification*, Revision A 1/6/88
11. Microchip Technology Inc., Development Tools and dsPIC[®] Digital Signal Controllers, <http://microchip.com>

12. Microchip dsPIC30F6012 and dsPIC30F6012A Digital Signal Controllers Data Sheets and User Guides, <http://microchip.com>

Acknowledgements

The author would like to thank and acknowledge the following people for their support in the development of these products. The μ SmartDigi would not exist without them.

Dave Davis, WB5WIA

John Hansen, W2FS, TNC-X Coastal ChipWorks

Ray Novak, N9JA, Icom America, Inc.

Pete Loveall, AE5PL

Scott Miller, N1VG, OpenTracker

© 2006 Painter Engineering, Inc. All Rights Reserved.

Permission is granted for publishing in the *Proceedings of the ARRL and TAPR Digital Communications Conference*.

Appendix 1- μ SmartDigi Configuration Parameters Description

The following table summarizes the Configuration Parameters:

Parameter	Arguments	Description
call	CALL	Sets the Call Sign
ssid	SSID	Sets the SSID with Decimal, Octal or Hex number
position	Lat Lon	Sets the Base Latitude and Longitude (alias base)
base	Lat Lon	same as position
havegps ¹	y n	Sets existence of a GPS
log	y n	Enables Disables Logging to the Monitor/GPS Port
nonaprs ¹	y n	Enables Disables repeating of packets without valid position info
dropmangled	y n	Enables Disables dropping mangled packets
haltonerr	y n	Enables Disables halting on error [for testing only]
host	[baud]	Sets optional Host baud
tnc	[baud]	Sets optional TNC baud
gps ¹	[baud]	Sets optional GPS baud
dstar ²	[baud]	Sets optional D-STAR baud
nsr ³	y n	Enables Disables No Source Routing
relay	y n	Enables Disables repeating anything with RELAY in path
widemax	N	Sets the maximum wideN in the remaining path
widetotal	N	Sets the maximum total wide for the remaining path
dupewin	seconds	Sets the Duplicate Check Window Time in seconds
digipath ²	string	Comma separated (no space or tab) Digi Path
fixvalid ²	y n	Enables Disables requiring GPS fix to be valid

Notes:

1. Digipeater parameter only.

2. D-Gate parameter only.

3. Not fully implemented in Digipeater Version 2.

Appendix 2- µSmartDigi Monitor Commands

The following table summarizes the Monitor Commands:

Command	Arguments	Description
help		Displays the Commands (alias ?)
?		same as help
monitor		Enables TNC and enters Monitor mode
log	y n	Enables Disables Logging to the Monitor/GPS Port
havegps ¹	y n	Sets existence of a GPS
tnc	[baud]	Sets optional baud, enables TNC and enters Monitor mode
gps ¹	[baud]	Sets optional baud and enters GPS mode
dstar ²	[baud]	Sets optional baud and enters D-STAR mode
host	[baud]	Sets optional Host baud
call	CALL	Sets the Call Sign
ssid	SSID	Sets the SSID with Decimal, Octal or Hex number
implicit	drop pass	Sets the Implicit Rule Action (drop or pass)
widemax	N	Sets the maximum wideN in the remaining path
widetotal	N	Sets the maximum total wide for the remaining path
dupewin	seconds	Sets the Duplicate Check Window Time in seconds
nsr ³	y n	Enables Disables No Source Routing
relay	y n	Enables Disables repeating anything with RELAY in path
nonaprs ¹	y n	Enables Disables repeating of packets without valid position info
dropmangled	y n	Enables Disables dropping mangled packets
haltonerr	y n	Enables Disables halting on error [for testing only]
fixvalid ²	y n	Enables Disables requiring GPS fix to be valid
position	Lat Lon	Sets the Fixed Base Latitude and Longitude (alias base)
base	Lat Lon	same as position
lat	Lat	Sets the Fixed Base Latitude
lon	Lon	Sets the Fixed Base Longitude
dconfigure		Displays the Configuration
drules		Displays all the Rules
clearrules		Temporarily clears the Rules. EEPROM Rules are not affected. Restored by RESET
angles		Displays the Angles format used for Lat and Lon and Rules
boot		Reboots the system (aliases reboot, reset)
reboot		same as boot
reset		same as boot

Notes:

1. Digipeater command only.
2. D-Gate command only.
3. Not fully implemented in Digipeater Version 2.

Appendix 3- μ SmartDigi Rules Description

uSmartDigi for Amateur Radio Packet Filtering Rules

Rich Painter, ab0vo
(c) Copyright 2004 - 2006 Painter Engineering, Inc.

File format

Action	Command	Args...
--------	---------	---------

Where Action and Command can be any case combination.
Abbreviations exist for both Actions and Commands.
Text following the args is ignored and can serve as comments.
Action, Command may be preceded or followed by any amount of white space.
All Latitudes and Longitudes are specified in any of 3 formats (see below).
White space is any combination of spaces and tabs.
White lines are ignored.
Lines may start with white space.
Comments start with the characters [#;/] and are ignored.

Basic operation

For every packet that is APRS the ordered list of rules are checked. If a packet matches a rule then the action (Drop or Pass) is taken. If a packet does not match any rule then it is implicitly Pass by default unless changed with the Implicit Command.

Actions:

drop	do not repeat the packets that match this rule
pass	repeat the packets that match this rule

Commands:

implicit	This has no arguments. This changes the implicit rule to that of Drop or Pass. Only one of these can be in the rule file.
----------	---

source	Specify a source call sign (complete or partial) with optional trailing * as a wild-card matching operator. The argument can be a single * to match everything.
--------	---

Format
{action} source call_sign[*]

destination	Specify a destination string with optional trailing * as a wild-card matching operator. The argument can be a single * to match everything. Valid APRS destination strings can be found in the APRS Spec v1.0.1 page 13.
-------------	--

Format
{action} destination dst_string[*]

circle Specify a circular zone of a specific radius in statute miles centered on a specified latitude and longitude.

Format
{action} circle radius Latitude_center, Longitude_center

Where the Latitude and Longitude arguments MUST be followed by a comma.

compass Specify a compass direction centered on a specified latitude and longitude.

Format
{action} compass direction Latitude_center,
Longitude_center

Where the Latitude and Longitude arguments MUST be followed by a comma and the Direction identifier is one of the following:

ID	Direction
N	everything N
S	everything S
E	everything E
W	everything W
NE	everything in NE quadrant
SE	everything in SE quadrant
SW	everything in SW quadrant
NW	everything in NW quadrant

rectangle Specify a rectangle using two (2) latitude-longitude pairs that form the Upper Left (NW) and Lower Right (SE) corners of a rectangle.

Format
{action} rectangle NW_Latitude, NW_Longitude,
SE_Latitude, SE_Longitude

Where the Latitude and Longitude arguments MUST be followed by a comma.

sector Specify a pie-slice shaped region using two (2) angles referenced from North, an Inner range and optional Outer range centered at the repeater's location. Ranges are in statute miles. The action applies to ranges greater than Inner Range to the Outer range or infinity if Outer range is not specified.

Format
{action} sector Angle1, Angle2, Inner_range

[Outer_range]

Where the Angle1 and Angle2 arguments MUST be followed by a comma and use the Colon, Dotted or DMC formsts. These Angles must be positive, measured clockwise from North resulting in a range of [0..360].

Command Examples

```
drop dest TLM*           ; drops all telemetry packets

pass source ab0vo*       ; passes all ab0vo packets, including ab0vo-1
                        ; and the like
```

This two-command sequence creates a donut where no repeats happen:

```
pass circle 2.5  n 39d 31m,  w 104.669d  // hole
drop cir    50.5  n 39d 31m,  w 104.669d  // torus

drop compass W    39:31:00, w 104.669d    # drops everything W of
                                           # this Lat-Lon

drop rect    40d 0m .5c, -104d 30m,  39d, -103d /* drop within
                                           /* this Lat-Lon box

drop sect    32d, 60d, 5 // drops all packets 5 miles or
                        // greater from the repeater
                        // from 32 deg from North to 60 deg
                        // from North
```

Colon, Dotted and DMC formats for Angles, Latitudes and Longitudes

Colon format

The Colon format is a fixed-position, colon-separated notation with an optional negative sign and optional fractional Seconds. Two (2) colons are always required for this format! No imbedded white space is allowed!

```
[s]DD:MM:SS[.F*]      for Latitudes
[s]DDD:MM:SS[.F*]     for Longitudes
```

Where:

```
s          is an optional negative sign (-) representing South
           Latitudes or West Longitudes
DD         is 2 decimal digits representing Latitude in Degrees
DDD        is 3 decimal digits representing Longitude Degrees
MM         is 2 decimal digits representing Latitude or Longitude
           Minutes
SS         is 2 decimal digits representing Latitude or Longitude
           whole Seconds
.F*        is optional decimal point and 0 or more decimal digits
           representing Fractional seconds.
```

Examples:

45:22:33 -103:00:00 -38:33:29.222 45:18:

Dotted format

The Dotted format is a fixed-position, period-separated notation with an optional negative sign and optional fractional Seconds. Two (2) periods are always required for this format! No imbedded white space is allowed!

[s]DD.MM.[F*] for Latitudes
[s]DDD.MM.[F*] for Longitudes

Where:

s is an optional negative sign (-) representing South Latitudes or West Longitudes
DD is 2 decimal digits representing Latitude in Degrees
DDD is 3 decimal digits representing Longitude Degrees
MM is 2 decimal digits representing Latitude or Longitude Minutes
F* is optional decimal point and 0 or more decimal digits representing Fractional Latitude or Longitude Minutes.

Examples:

45.22.64 -103.00. -38.33.379

DMC format

The DMC format allows a wide range of means to specify any Latitude or Longitude. Extensive error checks are made to ensure that the resulting specification does not exceed legal ranges nor conflicting directions are used.

DMC stands for Degrees, Minutes and seConds. S cannot be used for the seconds identifier because that is reserved for the S direction.

Angle identifiers MUST follow the numeric real number (integer and decimal) but may be preceded with white space. White space is any combination of spaces and tabs. The angle identifiers are:

D, d Degrees
M, m Minutes
C, c seConds

The direction identifiers may be used before or after any one of the angle specifiers and are:

N, n North
S, s South
E, e East
W, w West

Direction identifiers are optional. Without them, or a negative angle specifier (see below), Latitudes are North and Longitudes are East.

An alternative method for specifying direction identifiers is to use a negative sign (-) with ONE and only one of the degrees, minutes or seconds

parts. It is usually placed on the degree component but it could instead be on the minutes or seconds. Negative sign can only be found ONCE and cannot be used in conjunction with the direction identifiers!

Any legal combination of degrees, minutes and seconds can be used, including fractions for any or all parts. They can be used in ANY order but usually are ordered degrees, minutes and seconds.

Examples:

N 39d 23.7m 33.9c	# a North latitude
39d23.7m33.9c	# the same thing
39.404415D	# the same thing (approx.)
n39.395d 33.9c	# the same thing (approx.)
-42.5d	# South latitude or West longitude
N39d1m41.1c	# author's latitude
W104d40m8.4c	# author's longitude
W40m8.4c104d	# author's longitude