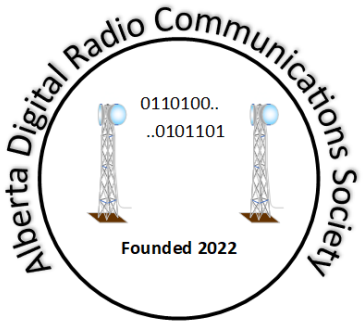

Digital Radio Projects

IP400 Node Software

Document Number: IP400-MCU
Revision: 1.0
Status: Released
Written By: Martin C. Alcock, M. Sc, MIEEE, VE6VH

Table of Contents

List of Tables

List of Figures

References

[1] gnu.org, "General Public Licence," [Online]. Available: https://www.gnu.org/licenses/gpl-3.0.en.html. [Accessed 25th February 2018].

[2] ST Microelectronics, "STM32WL33CC: Sub-GHz Wireless Microcontrollers.," ST Microelectronics, [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32wl33cc.html. [Accessed 28 1 2025].

[3] ST Microelectronics, "Integrated Development Environment for STM32," [Online]. Available: https://www.st.com/en/development-tools/stm32cubeide.html. [Accessed 26 1 2025].

Revision Status

Revision	Date	Description
0.1	January 26th[h], 2025	Initial draft
0.2	February 2nd , 2025	Added release notes for V0.3
0.3	February 6th, 2025	Added release notes for V0.4, added Pi HAT description
0.4a	February 7th, 2025	Added release notes for V0.4a, updated GPS section
0.4b	February 9th, 2025	Added BW table and fixed missing pictures, and GPS fix
1.0	March 24th, 2025	Updated for phase 2 release

Table 1 Revision status

Reference Documents

Author	Issue Date	Document Number	Description
M. Alcock	Jan 2025	IP400-PHY	Physical Layer Specification
M. Alcock	Mar 2025	IP400-SPI	Radio Node SPI Protocol specification
M. Alcock	Mar 2025	IP400-CC2	Nucleo-CC2 Experimenter Node

Table 2 Reference Documents

Intellectual Property Notice

The hardware components and all intellectual property described herein is the exclusive property of the Alberta Digital Radio Communications Society and others (“the owners”), all rights are reserved.

The owners grant licence to any Amateur for personal or club use, on an as is and where is basis under the condition that its use is for non-commercial activities only, all other usages are strictly prohibited. Terms and conditions are governed by the GNU public licence [1].

No warranty, either express or implied or transfer of rights is granted in this licence and the owner is not liable for any outcome whatsoever arising from such usage.

Disclaimer

This document is a preliminary release for a product still in development and may be subject to change in future revisions. The software described herein may be subject to unpredictable behaviour without notice. You are advised to keep a can of RAID™ Ant, Roach and Program Bug killer handy. Spray liberally on the affected area when needed.

If any page in this document is blank, it is completely unintentional.

Introduction

The IP400 project was launched to experiment with digital mesh networking on the 400 MHz band, using commercial devices designed to run in this band.

To get the project rolling, code has been developed for an STM32WL33 microcontroller [2], on several platforms that support the basic mode. Further hardware will be developed in the future for different modes.

Operation of the node is segmented into a physical layer that runs on the microcontroller, and other layers that run on different host processors. The lower layer code contains the bare minimum to send and receive frames, repeating frames and building a mesh table, and contains a simple application.

The document describes the internal application which includes a simple setup menu, and the ability to change and store station and radio parameters, as well as a simple chat application to demonstrate the capabilities. It periodically sends a ‘beacon’ frame to build the mesh tables, which contains information about the station, including latitude, longitude and grid square. Provision has also been made to connect to a GPS receiver to update the position information dynamically.

The current revision level of the firmware is the first of the phase 2 to be released.

Roadmap

Figure 1 illustrates the roadmap for the IP400 node software.

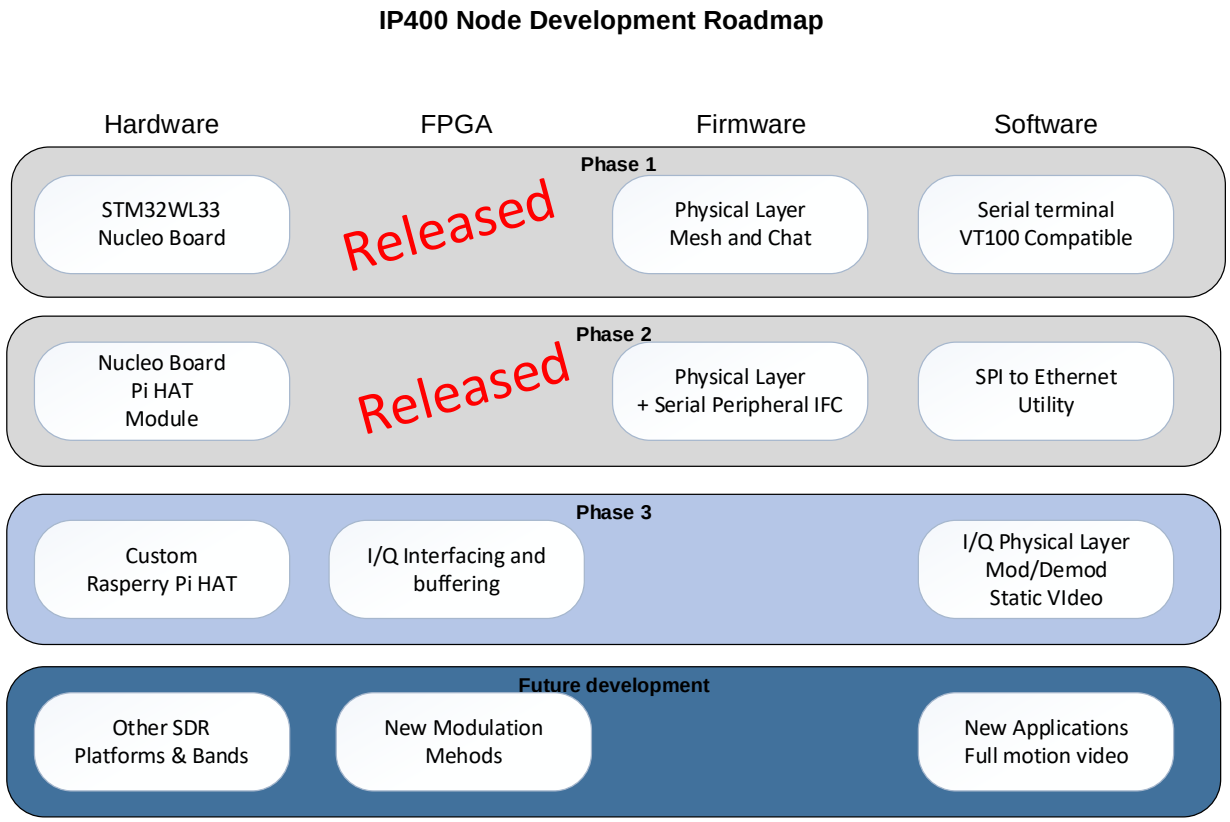

Figure 1 IP400 Node Development Roadmap

There are four phases of development for the IP400 node:

- Phase 1 is the initial evaluation phase using the Nucleo board and a simple chat application and is purely a firmware exercise. Using the PuTTY application, a connection can be made to the USART on the board. The software will implement a basic frame transmitter and receiver and be able to build a mesh table and repeat frames. The firmware can only be loaded using the ST integrated development environment (IDE) [3] and an integrated STLINK Debugger.
- Phase 2 utilizes either the Nucleo or an integrated HAT for the raspberry Pi platform. The firmware will be upgraded to include a high speed SPI, plus a downloader will be developed to alleviate the absolute requirement for the IDE, but support for an external STLINK debugger has been provided, so the IDE can also be utilized as an option. New applications on the Pi can be developed for routing, audio, file transfers, AX.25 and encapsulated IP.
- Phase 3 will introduce new custom hardware and FPGA-based signal processing for higher modulation methods and add new applications, to be determined.
- The final phase will migrate to other bands and platforms.

Operating the software

Main Menu

The main menu is enabled after a restart. It is shown Figure 2.

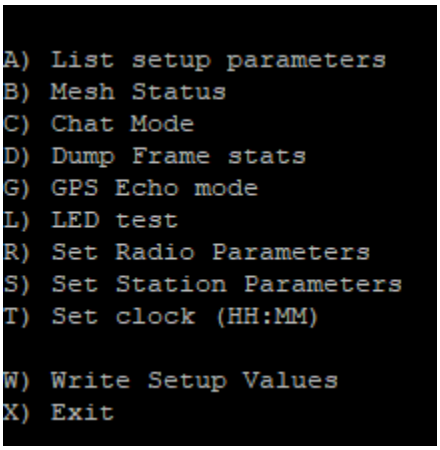

Figure 2 Main Menu

The menu items are described in Table 3.

Menu Item	Function
A	Lists the current setup parameters, including clock, station and radio
B	Displays the mesh table contents
C	Enters chat mode
D	Dumps the frame statistics
G	Dumps the NMEA sentences received from the GPS receiver[1]
L	Runs an LED test
R	Sets the radio parameters
S	Sets the station parameters
T	Sets the time of day clock
W	Writes the setup data to the flash
X	Exits the menu and puts the node in silent mode

Table 3 Main Menu Items

Most are self-explanatory, however further explanation is required.

Not

[1] Only shown when compiled with GPS code enabled

Setup Parameters

Figure 3 illustrates the setup parameters. Menu Items 'S' and 'T' set the station and radio parameters; menu item 'T' sets the time of day clock. Table 4 explains them.

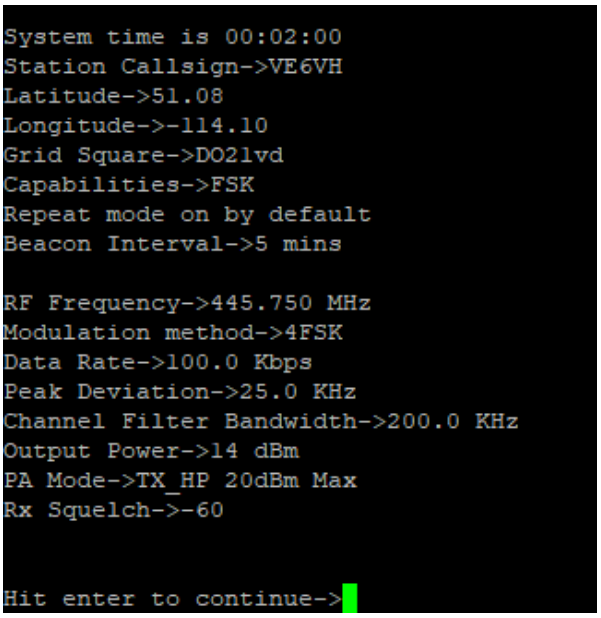

Figure 3 Setup Parameters

Item	Purpose	Format
Time	Time of day clock with 10 second resolution	HH:MM
Callsign	Station callsign	Up to 6 characters[2]
Latitude	Latitude, positive for N, negative for S	±xxx.xxx[3]
Longitude	Longitude, positive for E, negative for W	±xxx.xxx
Grid Square	Home grid square	XXNNxx
Capabilities	Lists station operating modes	FSK or OFDM
Repeat Mode	Set the repeat flag in sent frames	On or OFf
Beacon Interval	Interval between beacons, normally 5 mins	xx minutes
RF Frequency	RF operating frequency from 420 to 450 MHz	XXX.XXX or NNNNNNNNN
Modulation	Modulation method, usually 4FSK	2FSK or 4FSK
Data Rate	Data rate 1.2 to 600Kb/s	xx.xx KHz or NNNNNN
Peak Deviation	Peak FM deviation	xx.xx KHz or NNNNNN
Channel Filter BW	BW of the receiver	xx.xx KHz or NNNNNN
Output power	Sets the transmitter power	0 to +20 dBm
Rx Squelch	Sets the receiver threshold	-30 to -130 dBm

Table 4 Setup parameters

[2] An extended method has been designed but not yet implemented
[3] Used in beacon frames, unless a GPS receiver is connected

Channel Bandwidth Settings

The bandwidth values must be one of those listed in Table 5.

Permissible Bandwidth values (KHz)							
1600	666	266	100	41.6	16.6	6.1	2.6
1510	622	244	94.4	38.9	15.3	5.9	2.4
1422	577	222	88.9	36.1	13.9	5.5	2.2
1332	533	200	83.3	33.3	12.5	5.2	2.1
1244	488	188	77.8	30.5	11.8	4.8	1.9
1154	444	178	72.2	27.8	11.1	4.5	1.7
1066	400	166	66.7	25.0	10.4	4.1	2.6
976	377	155	61.1	23.6	9.7	3.8	
888	355	144	55.6	22.2	9.0	3.5	
800	333	133	50.0	20.8	8.3	3.1	
755	311	122	47.2	19.4	7.6	2.9	
711	288	111	44.4	18.1	6.9	2.7	

Table 5 Permissible Bandwidth Values (KHz)

Mesh Table

Figure 4 illustrates the mesh table contents, and Table 6 explains them.

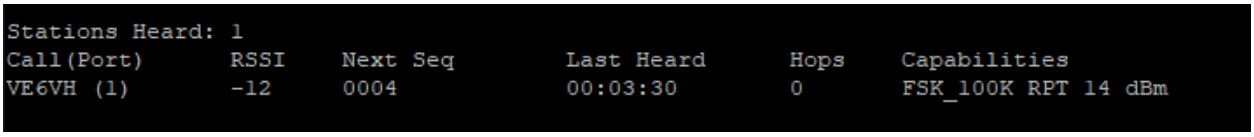

Figure 4 Mesh Status Table

Item	Explanation
Call(port)	Callsign of the transmitting station, and port number heard
RSSI	Receive signal strength when SYNC received (in dBm)
Next Seq	Next anticipated sequence number
Last Heard	TOD clock reading when last heard
Hops	Number of repeat hops, 0 indicates a direct signal
Capabilities	Data rate, repeat capabilities and transmitted signal strength

Table 6 Mesh Table explanation

Chat Mode

There are several control keys in the chat mode that are interpreted, as shown in Table 7.

Key	Purpose
ESC	Enables entering a new destination address, normally broadcast.
CTRL+'R'	Toggles repeat mode
CTRL+'D'	Toggles dump mode
CTRL+'Z'	Exits chat and returns to the main menu
ENTER	Sends the current frame
BACKSPACE	Deletes the last character entered, or more if repeated

Table 7 Chat mode contol keys

If the 'ESC' key is hit a second time in the destination address entry, it will be reset back to a broadcast address.

Chat frames are displayed as follows:

Originating callsign(source port)>Destination Callsign(dest port)[number of repeats]:text message...
NOCALL(17)>BROADCAST(1085)[0]:hello to you...

The chat port number is always 17.

LED Test

The LED test will cycle through the patterns shown in the following table:

Test	Nucleo Board	PI HAT
1	RED Led	Bi Color Red
2	GREEN Led	Bi Color Green
3	All off	All off
4	Blue On	Tx LED On
5	All off	Tx LED Off

Table 8 LED Test Cycling

GPS Echo Mode

The NMEA sentences from the GPS receiver are dumped to the screen as shown below:

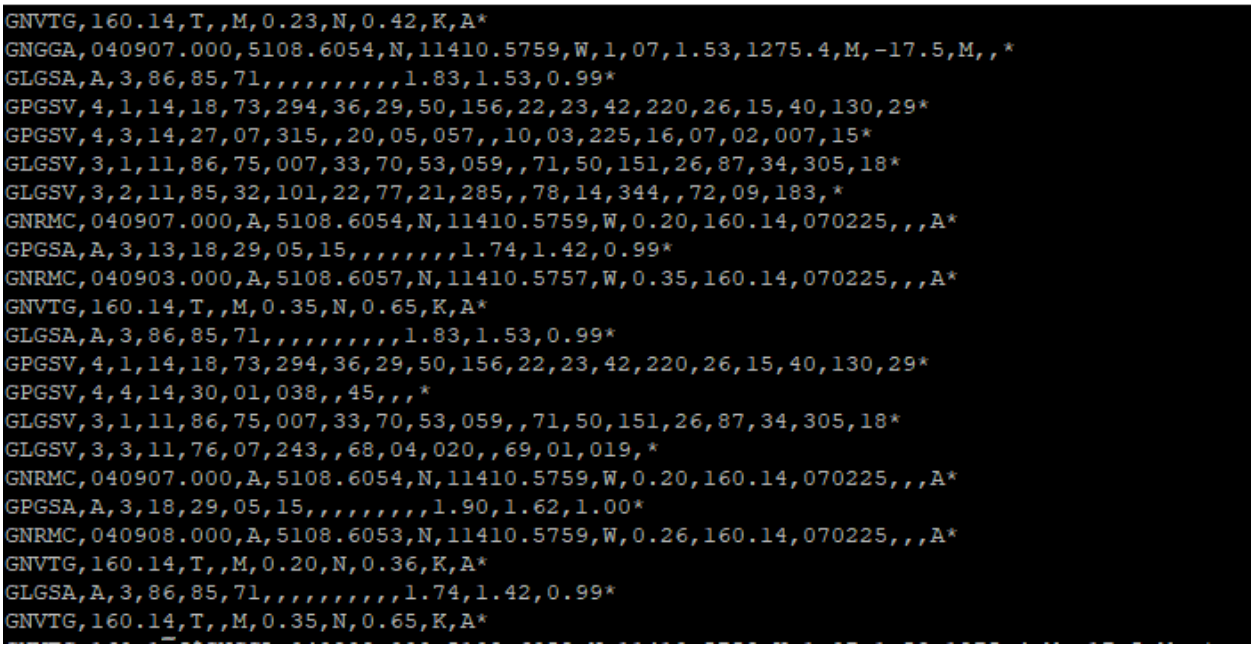

Figure 5 GPS NMEA sentences

While this mode is active, two keys are processed:

1. Pressing the 'ESC' key will pause and prompt for the enter key to be hit before returning to the main menu. The data remains on the screen.
2. Pressing the ENTER key will clear the screen and return to the main menu.

Frame Statistics

The statistics for each frame are shown in Table 9.

Item	Meaning
Transmitted Frames	The number of frames transmitted
Good Rx Frames	The number of frames with a good CRC
CRC Errors	The number of frames with CRC errors
Rx Timouts	The number of times the receiver timed out

Table 9 Frame statistics

Setting the clock

The time of day is entered as HH: MM. The clock has a granularity of 10 seconds and uses 24 hour format, and knows nothing about time zones, as it considers you are in your home zone.

Mesh Table

The mesh table is built by receiving a frame from any source. If it has not previously been heard, it is added to the mesh table. The mesh table can be view from menu item B. Each entry is the mesh table is displayed as shown in Figure 6.

```
Stations Heard: 1
Call(Port)      IP Addr         RSSI    Next Seq        Last Heard      Hops    Capabilities
VE6VH (0)       172.29.122.83   -6      0055            17:04:20        0       FSK_100K RPT 14 dBm
```

Figure 6 Mesh Table Display

The table contents are listed in Table 10.

Field	Explanation
Call (Port)	Callsign of sending station, and port number where frame was heard
IP Address	Suggested Ip address for the node
RSSI	Signal strength when sync was detected in dBm
Next Seq	Next expected sequence number from this station
Last Heard	Timestamp from the system clock when the last frame was heard
Hops	Number of times the frame was repeated, zero is direct
Capabilities	The capabilities of the station and its transmit power (into the antenna)
Tx Power	Transmit power in dBm.

Table 10 Mesh table contents

The station capabilities can only be populated upon receipt of a beacon frame. If a non-beacon is received from a new station, it is added to the table without this field. If a beacon is heard subsequently, all the fields are updated. See Table 11 for details of these.

Bit	Capability	Meaning
0	FSK	Station can send 2 or 4 FSK
1	OFDM	Station is capable of OFDM transmissions
2	AREDN	Station is an AREDN node or has a path to one
3	REPEAT	Default setting of the repeat flag
4	EXT	Callsign is greater than 6 characters and has an extension
5	RATE	Maximum speed capability of the station[4]

Table 11 Capabilities field

[4] An explanation of these can be found in the frame layer documentation

Phase 1 Firmware Release Notes

V0.3b

Minor change made to the radio parameters display and entry. The frequency is now displayed as xxx.yyy and can be entered the same way. The old method of specifying it with all the digits is still supported.

Added the ability to set the clock. Format is HH:MM, 24 hours, time zone agnostic.
Granularity is 10 seconds. Current time added to setup display.

Chat mode was not changed, a reminder on how it works:
In the chat mode, to send a line of text just key it in and hit enter. Backspace removes one key at time.
CTRL+R changes the repeat flag. This flag tells a receiving station to repeat the frame. Defaults to setup.
CTRL+D: sets dump mode. When enabled, the frame header will be dumped instead of interpreted.
ESC keu (only at the beginning of a line) enables entry of a destination callsign. Defaults to broadcast.
CTRL+Z. Exits chat mode and returns to the main menu.

The received frames are interepreted as:
Originating callsign(source port)>Destination Callsign(dest port)[number of repeats]:text message...
NOCALL(17)>BROADCAST(1085)[0]:hello to you...
The number of repeats is the number of times the frame has been repeated.

Beacon frames can be dumped to the console, by enabling __DUMP_BEACON in frame.c. If enabled, ensure that the receiving station is in receive mode to avoid running out of heap space.

Limited support has been added for a GPS receiver using the LPUART. It uses the DMA mode to receive a message and parses a GGA message for lat/long and a timestamp. These fields are used in the beacon frame. It is enabled by __ENABLE_GPS in beacon.c

BUGS FIXED
The beacon mode is now fully implemented. See the protocol spec for a description. Frequency is in the station setup, it can be overwritten by enabling __SPEED_DAEMON in beacon.c.
The mesh table is built using beacon frames. It lists the callsign, capabilities and last heard time.
The setup parameters can now be changed and stored in the internal flash.

NEW BUGS
The repeat mode has not yet been implemented.
A facility to set the clock needs to be added to the main menu.
The GPS code has yet to be fully tested.

V0.4

The receive frame processing was overhauled to make it consistent with the transmitter. Both now work in units of IP400_FRAMES.

The mesh table is now operational, and now contains an RSSI reading, timestamp, hop count, and capabilities of each station heard.

The repeat mode has been implemented. An inbound frame with the repeat flag set and hop count less than the maximum is repeated, providing it was not sourced by the receiving station.

The frame now contains a sequence number which is stored in the mesh table. If a frame is received with a previously known sequence, it is dropped.

The transmitted power is sent in the beacon frame to enable calculation of the path loss.

An LED manager has been added to control the LED's and provide better error indications. See the description in the documentation.

Menu changes

The current firmware and time of day clock have been moved to the list setup parameters menu item.
The mesh status now lists the receive RSSI, next expected sequence number, a timestamp, number of hops, and capabilities.
The frame status now reports how frames were processed, as beacons, repeated, duplicated and dropped. It also shows the current radio errors, if any, and the current radio FSM state.
An LED test has been added, as well as a set time of day clock.

BUGS FIXED
Repeat has been implemented.

NEW BUGS
GPS code is still pending.

V0.4a

GPS code is operational, an echo mode has been added.

BUGS FIXED
A bug was found in the callsign compare, it failed on callsigns less than 6 characters.

NEW BUGS
SPI mode has not yet to been implemented, pending arrival of a working Pi board.

V0.4b

The GPS issues were fixed, and a return message has been sent back to reduce the number of messages to only a fix message.

A second escape key hit in the destination callsign field will reset it to a broadcast address.

A fix was made in the packet length to ensure that the DMA worked correctly, by rounding up to a multiple of 4 bytes.

Phase 2 Firmware Release Notes

V1.0

New Features:

1. An experimental method of generating IP addresses has been added to mesh table.
2. Frames that are repeated now have a hop table that describes the routing methods.
3. SPI communication to host a processor has been added, details are available in a separate SPI specification document.
4. New code for the Pi to exchange data between the SPI and an ethernet port.
5. Wireshark dissector for packet examination using the generic dissector.
6. Allowances have been made for a local control packet to be utilized to change parameters such as frequency, bandwidth, etc, but not fully implemented.

Changes:

1. The previous scheme of using a port number in the callsign (MAC) address field was dropped, to be reconsidered in the future for multiple nodes. Callsigns are still compressed, the other bytes can be used for additional nodes under the same callsign, up to 4 bytes.
2. Packets are now processed based on their coding types. Further information is available in the SPI protocol specification document.
3. A message is sent to the GPS receiver, when enabled, to limit the type and frequency of message to reduce the overall traffic load.

Bugs Fixed:

1. A memory leak in the queuing code has been investigated and fixed.