MapleDSP

PUTSI™ WiFi Telemetry Module

Written by:
Martin Alcock, M. Sc, VE6VH

For Software Revision: 0.2

MapleDSP is a trademark of
Praebius Communications Inc,
43 Citadel Bay NW,
Calgary, AB T3G 3Y4
www.praebius.com

Table of Contents

List of Figures

List of Tables

Document Revision History

Date	Rev	Description
June 2023	0.1	Draft of dual mode operation
Sept, 2023	0.2	Updated to current firmware

Table 1 Revision History

Distribution

Rev	Distribution
All	For amateur use only.

Table 2 Distribution

Reference Documents

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

[2] Ettus Research, "USRP Hardware Driver and USRP Manual," [Online]. Available: https://files.ettus.com/manual/page_transport.html. [Accessed 21 October 2020].

Glossary of Terms

PBX	Private Branch Exchange. A node in a telephone network that provides connectivity for a series of local extensions to a set of trunks.
VOIP	Voice over Internet Protocol. A system where telephone calls are placed, and audio is exchanged using the Internet Protocol.
PUTSI	PIC USB Telemetry System Interface
Wi-Fi	Wireless Fidelity implementing the IEEE 802.11x standards.

Intellectual Property Notice

The hardware components and all intellectual property described herein is the exclusive property of Praebius Communications Inc (“the owner”), all rights are reserved.

The owner grants 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.

MaplePi, TopHAT are registered trademarks of Praebius Communications Inc.

Copyright © Praebius Communications Inc, all rights reserved.

Disclaimer

This document is a preliminary release for a product still in development and may be subject to change in future revisions. The software 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.

Release notes

Initial Release version 0.1:

1. **Dual operating modes**
   The current firmware supports operation in both the Wi-Fi and USB modes, which can be used simultaneously.

2. **Bidirectional Digital I/O**
   Digital I/O is now supported in both directions in both communication modes.

3. **Bi-Colour Communications LED**
   The COM led can now be off, red, green or alternating to reflect the current state of the firmware.

4. **Wi-Fi Configuration Mode**
   The Wi-Fi parameters can be entered using a host mode and single web page.

Update Rev 0.2:

1. **Bugs Fixed**
   A bug was found in the configuration mode that disabled both Wi-Fi and USB modes. The configuration mode is entered automatically if this condition occurs.

2. **Wi-Fi setup shortened**
   The port numbers and auto configuration were removed from the Wi-Fi setup and implemented as compilation options. Port numbers are unlikely to change, and the auto configuration is only applicable to the Wi-Fi mode, so now it is always enabled.

3. **USB timeout added**
   A timeout has been added to the USB mode. If no response is heard for 2 seconds, USB mode is disabled until a new handshake can be carried out.

Overview

The digital telemetry system is designed to gather data at a remote site by a reporting device which sends it over an IP network to a remote site, where it is written to a database to be displayed at will. The data content can be from an analog source using and off the shelf converter, or from a digital input such as an open door sensor. A mechanism is also in place to return data for digital outputs, which can be used to activate remote devices such as lights, switches, etc.

Packets are sent over an IP network using the USRP (Universal Software Radio Peripheral) protocol, as user datagram packets (UDP), which does not require a connection to be maintained. Packets are acknowledged by the receiver to verify that they were received.

Two types of reporting devices for gathering data have been developed, the MaplePi FPGA controller and a standalone device which employs a PIC processor called PUTSI (PIC USB Telemetry System Interface) which can be used in conjunction with another device such as a Raspberry Pi or other computer over a USB connection, or it can transmit packets directly over a Wi-Fi connection.

The MaplePi FPGA controller has a driver for Allstar that has the telemetry functionality built in, which makes use of the on-chip 8 channel A/D converter, and with an additional module can implement up to 8 outputs and 3 inputs. The PUTSI is equipped with 8 analog inputs, and 5 digital inputs and outputs, and can implement the Chameleon functionality required by Allstar, or be used in a standalone mode.

The receiving site has a web-based interface and runs on a standalone server. Data from the reporting device is written to a database, which dynamically updates a web page when needed. Analog data is interpreted by a pre- and post-scalar formula into more ‘meaningful’ data, such as temperature in degrees Centigrade, Kelvin or Fahrenheit, and digital I/O can be represented as ‘real world’ devices.

Alarms can be generated that can be sent to key personnel using an external e-mail facility, or, if present, sent as a SIP message to an IP phone, or as an SMS message to a cell phone. The frequency of updates is completely programmable.

Hardware

Figure 1 illustrates the PUTSI (PIC USB Telemetry System Interface) standalone hardware module for telemetry gathering. It has a standalone processor and can be connected to an Allstar system with USB or directly to a hub using a Wi-Fi module. In the USB mode, it emulates the Chameleon device supported natively by Allstar.

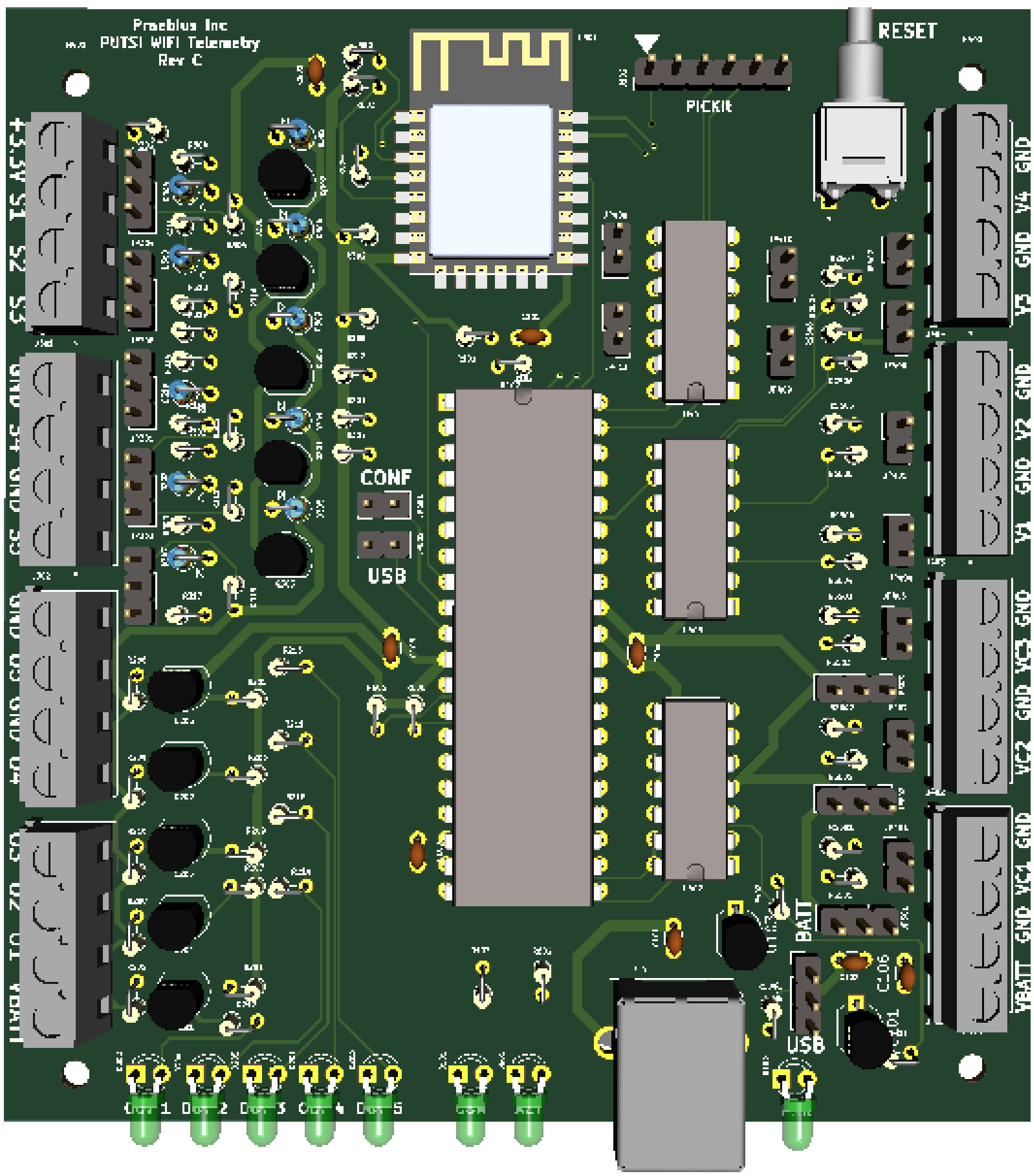

Figure 1 Pic based USB/Wi-Fi Telemetry Board (Rev C)

Specifications

The module can be powered either by the USB connection or a separate battery connection. The specifications are illustrated in Table 3.

Parameter	Min	Typ	Max	Units
Number of Analog Voltage inputs		4		
Number of Configurable Analog Channels		3		
Number of battery monitoring channels		1		
Number of uncommitted op amps		4		
Analog input voltage range	0		16[1]	V
Number of digital inputs		5		
Digital Input voltage range	0		vBatt	V
Number of digital outputs		5		
Output voltage range	0		vBatt	V
Output sink current			20	mA
Battery input voltage (vBatt)	5	12	16	V

Table 3 PUTSI module specifications

Power Options

The board can be powered either from the +5V supply on the USB cable, or from the VBATT terminals. Jumper JP101 determines which source is to be used, as illustrated in Table 4.

JP101 Jumper Settings	Power Source
1-2	+5V from the USB cable (recommended)
2-3	From the VBATT terminals (J401 1-2)

Table 4 Power Options

LED Indicators

There are 8 LED indictors as shown in Table 5.

LED Name	Purpose
CONN	Indicates that a Bluetooth connection has been established
PWR	Indicates power has been applied, through either source
COM	Indicates that the board is communicating with the host computer
OUT 1 to 5	One LED to show the status of each digital output

Table 5 LED indicators

[1] Higher voltages can be accommodated by adjusting Rg and Rs for the specific input.

Reset

There is a reset toggle switch beside the COM LED. Depressing it will reset the PIC processor, to re-establish communications the host software may need to be restarted.

Connectors

There are 8 connectors for inputs and outputs:

Connector	Pin	Desig	Purpose
J401	1	VBATT	Battery connection for power or monitoring
	2	GND	
	3	VC1	Configurable voltage input #1
	4	GND	
J402	1	VC2	Configurable voltage input #2
	2	GND	
	3	VC3	Configurable voltage input #3
	4	GND	
J403	1	V1	Analog input #1
	2	GND	
	3	V2	Analog input #2
	4	GND	
J404	1	V3	Analog input #3
	2	GND	
	3	V4	Analog input #4
	4	GND	
J201	1	VBATT	Battery voltage output
	2	O1	Digital Output #1
	3	O2	Digital Output #2
	4	O3	Digital Output #3
J202	1	O4	Digital Output #4
	2	GND	
	3	O5	Digital Output #5
	4	GND	
J302	1	S5	Sense Input #5
	2	GND	
	3	S4	Sense Input #4
	4	GND	
J301	1	S3	Sense Input #3
	2	S2	Sense Input #2
	3	S1	Sense Input #1
	4	VCC	Regulated +3.3V output

Table 6 Connector Pinouts

Converter Range

The A/D converter has a resolution of 3.3V/2047 per count. Using a resistor divider of 820/3300, the battery voltage can be scaled to fit into its linear range as shown in Figure 2.

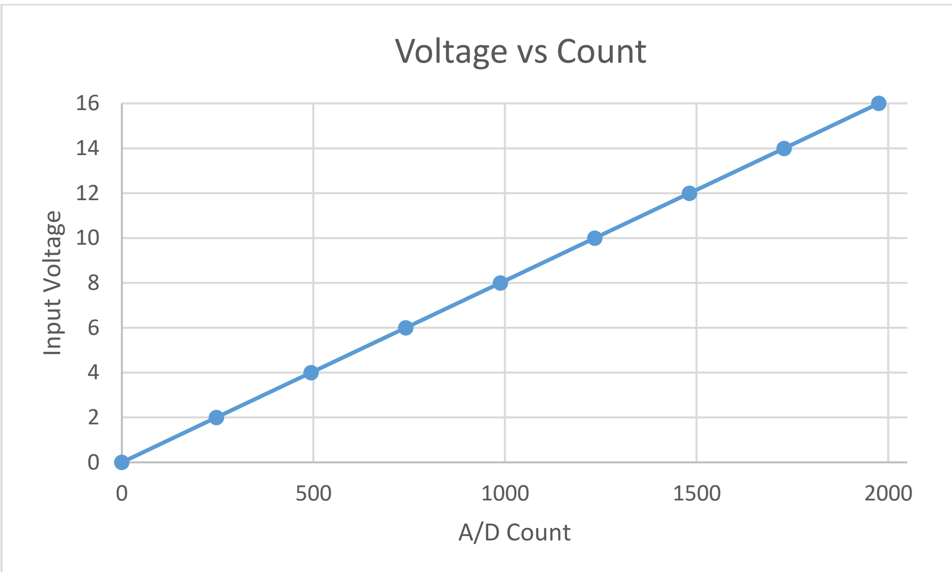

Figure 2 Scaled battery input voltage.

Analog Inputs

At each input a dedicated op amp follower provides buffering before the converter. Figure 3 illustrates one of the analog voltage inputst:

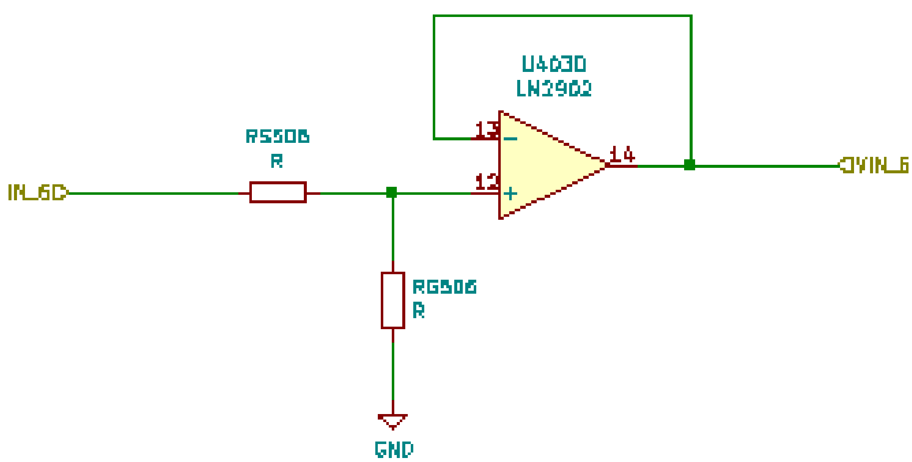

Figure 3 Analog input

Choosing the Resistor Divider Values

Resistors RS and RG (Series and Ground) implement a voltage divider to bring the analog voltage into operating range of the A/D converter, which can be from 0 to 3.3V.

The resistor values will vary depending on the maximum input value. It is recommended that the input voltage to the A/D not exceed 3.3V, so the divider value determines how the input value is scaled. In the monitoring software, a scalar value can be applied to the received value to restore the original voltage.

*Table 7*Table 7 lists recommended values based on a maximum input voltage. The scalar values shown can be used to restore the original voltage in the monitoring software.

Max Input Voltage	RS (KΩ)	RG (Ω)	Max A/D Input	Scalar	Scaled
24	3.3	470	2.99	8.0213	24
16	3.3	820	3.18	5.0244	16
12	3.3	1200	3.2	3.7500	12
8	3.3	2200	3.2	2.5000	8
6	3.3	2900	3.25	1.8462	6

Table 7 Recommended Resistor Values

The formula for calculating the resistor values is shown in the following equation:

$$3.3V = Vin \times \frac{Rs}{Rs + Rg}$$

For example, if a battery voltage was to be measured that had a maximum of 16V, the Rs can be calculated by first fixing Rg at a known value, and then solving for it. The recommended values of 820/3.3K will yield an approximate divide by five, which will work for most voltages used in repeater systems.

Each input has a jumper before the resistive divider. There are four uncommitted op amps configured as unity followers, and the jumper terminals can be connected to one of these to provide high impedance isolation of any input, before the divider.

Configurable Analog Inputs

Configurable inputs have a different circuit as illustrated in Figure 4. There is an additional jumper that enables the RG resistor to either act as a divider, when connected to ground, or as a voltage source when connected to Vcc.

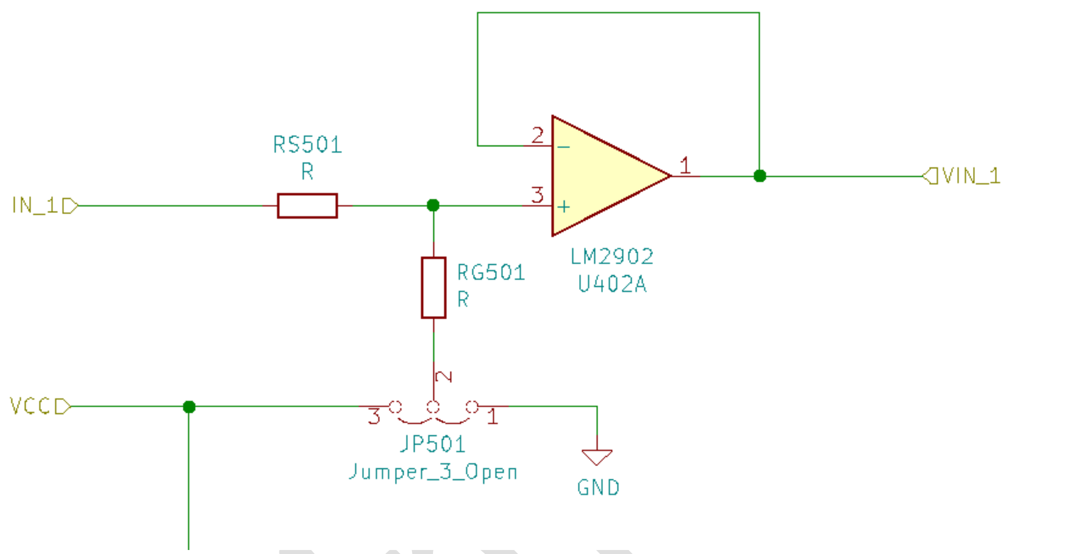

Figure 4 Configurable analog Input

A thermistor type temperature probe or active device can be configured for measuring temperatures, such as the shack, power amplifiers, etc. The recommended devices for temperature measurement is the TMP36 from Analog Devices, which retails for about $3.

To use this device, connect the left-most pin to the regulated supply voltage of 3.3v (available at J301-4), the centre pin to the positive input and right most to the negative. For temperatures in °C, Rs can be jumpered with a wire, and Rg can be omitted, and scalar of 100 and offset of -50 will restore the original temperature value, as illustrated in Table 8

Temperature	Voltage	Scalar	Offset	Scaled
-50	0	100	-50	-50
-25	0.25			-25
0	0.5			0
25	0.75			25
50	1			50
75	1.25			75
100	1.5			100
125	1.75			125

Table 8 Measuring Temperature in deg C using TMP36

For degrees Fahrenheit, the voltage has to be scaled by 0.9 at the input, by setting Rs to 43K, and Rg to 4.7K. A scalar of 200 and offset of -58 will convert to the approximate temperature values (±1 deg).

Temp °C	Temp °F	TMP36 Output	A/D input	Scaled
-50	-58	0	0.000	-58
-25	-13	0.25	0.225	-13
0	32	0.5	0.451	32
25	77	0.75	0.676	77
50	122	1	0.901	122
75	167	1.25	1.127	167
100	212	1.5	1.352	212

Table 9 Measuring Temperature in deg F using TMP36

Digital Inputs

Digital inputs can sense the presence of an analog voltage up to 12v, which can be hooked up to sense environmental changes such as a door being opened. A typical input circuit for each digital input is shown in Figure 5.

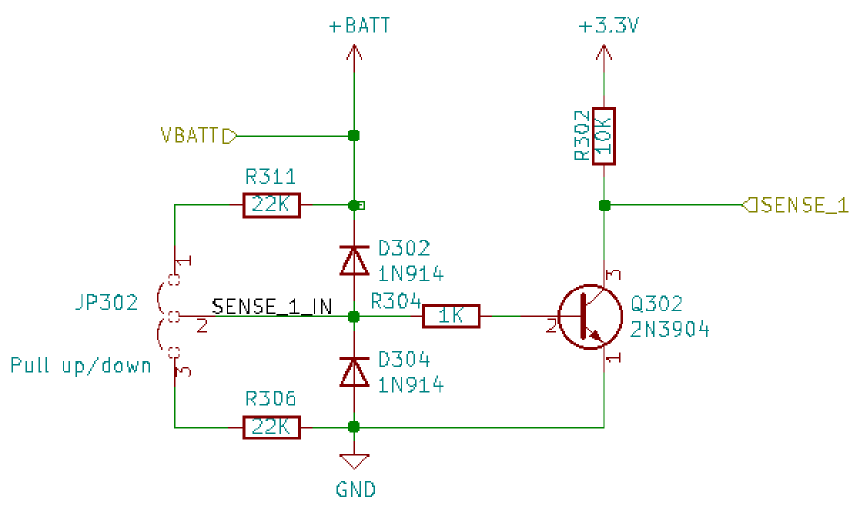

Figure 5 Digital input conditioning

Each input can be pulled up or down depending on the jumper setting. Following that a pair of protection diodes ensure that the voltage does not go below ground and above the battery voltage. It this were to occur it is clamped. The transistor buffers and inverts the input, the software removes this inversion. The quiescent voltage when pulled up is about 1V, it can be changed by modifying the pullup resistor value.

Digital Outputs

The circuit of a digital output is shown in Figure 6. Each has a light emitting diode (LED) to show its status and is buffered with a field effect transistor (FET). When the output from the microprocessor is at ground, the LED is turned off and no current is sourced to the gate of the FET, and the output rises to the battery voltage. When current is applied by the microprocessor, the LED is lit up and the FET is turned on.

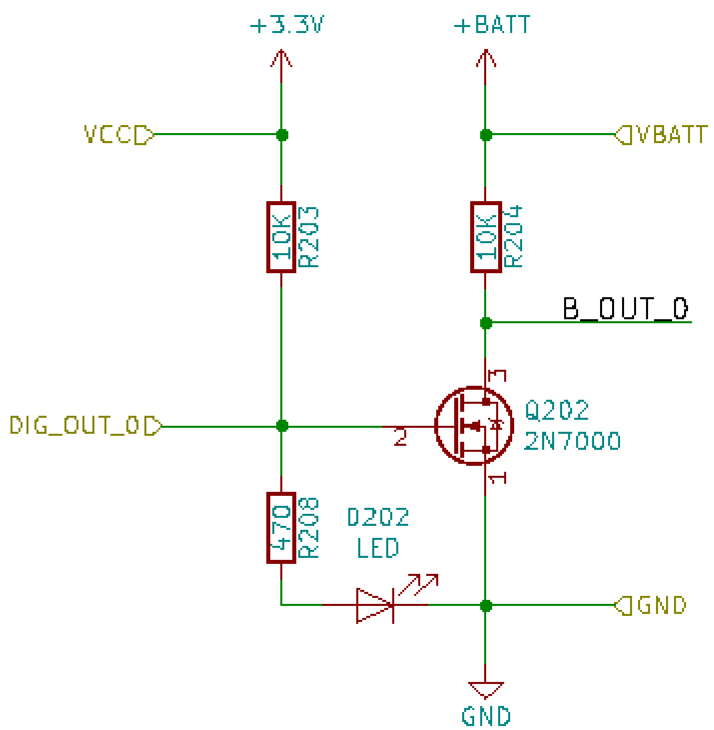

Figure 6 Typical Digital Output

Configuration

Configuring for USB mode only

The device is preconfigured to work in the USB mode with the Wi-Fi disabled. No further configuration is required if this is the only mode of operation required. All setup parameters are ignored as the USB requires no further configuration.

Configuring for WiFi mode

If the WiFi mode is to be used, either in lieu of the USB or in tandem with it, it has to be configured before it can be used. To enter the configuration mode, insert a jumper into the 'CONF' pins close to the processor, and toggle the reset switch. It will come up with the COM led flashing Red.

The Wi-Fi setup page can be launched by connecting to the network 'PUTSI Wi-Fi', using a PC, smart phone or tablet; there is no password needed. Launch a web browser and go to address 192.168.36.1, the device will respond with the configuration page as shown in Figure 7.

All parameters for the server IP, Radio ID, Network name and password must be entered with the correct information.

Dual mode

The device can operate in both modes simultaneously, but this has to be configured using the Wi-Fi setup page.

WIFI Setup Page

Figure 7 Wi-Fi Configuration Page

The settings are illustrated in Table 10.

Setting	Contents
Server IP Address	IP address of the telemetry server on the network for WiFi mode
Radio ID	Unique radio ID, up to 7 characters
Network Name	Name of the network where to connect (Access point)
Password	Network password
Reporting Interval	Interval between reports in seconds; recommended value is 30.
Enable WiFi Mode	Enables the transmission of packets over WiFi
Enable USB Mode	Enables the USB mode

Table 10 Wi-Fi Configuration settings

Configuring Allstar

To monitor using the USB interface, the configuration must be entered into rpt.conf. First the physical device and the pin definitions must be defined. The following must be added to the [telemetry] section of the file as illustrated.

```
; Data Acquisition configuration

[daq-list]
device = daq-cham-pic

[daq-cham-pic]              ; Defined in [daq-list]
hwtype = uchameleon         ; DAQ hardware type
devnode = /dev/ttyACM0      ; DAQ device node (if required)
1 = inadc                   ; Pin definition for an ADC channel
2 = inadc
3 = inadc
4 = inadc
5 = inadc
6 = inadc
7 = inadc
8 = inadc
9 = inp                     ; Pin definition for an input
10 = inp
11 = inp
12 = inp
13 = inp
14 = out                    ; Pin definition for an output
15 = out
16 = out
17 = out
18 = out
```

To generate alarms, the alarms section must be configured as shown below. The parameters are listed in *Table 11*.

```
[alarms]
;
;tag = device,pin,node,ignorefirst,func-low,func-hi
;
doorpc = daq-cham-pc,9,1,1000,*51,-
pwrfailpc = daq-cham-pc,10,0,1000,*52,-
```

Parameter	Purpose
tag	A unique tag identifying the alarm
device	Device to poll as defined in the [daq-list] stanza
pin	Pin number to monitor
ignorefirstalarm	set to 1 to throwaway first alarm event, or 0 to report it
node	the node number to execute the function on
func-lo	the DTMF function to execute on a high to low transition
func-high	the DTMF function to execute on a low to high transition

Table 11 Allstar alarm parameters

Operation

USB Mode

The USB interface provides a popular connection to most devices. It follows the protocol of the USB chameleon device that was initially supported by AllStarLink. All data is sent as ASCII characters and is terminated with a line feed ($0A_{16}$) character.

There are three phases of communication:

1. Handshake
2. Configuration
3. Data Exchange

The handshake mode is entered after powering up and when the reset button is depressed. The COM LED is off until the handshake has been completed. The device waits for a message from the host to send an identification string, and, if recognized, the host then sends a command to activate the COM led, which be on a solid RED when the connection has been established. A response is sent to the host at least once every 50ms to maintain the connection. If no data has been solicited or is ready to be sent, an empty line is sent to maintain the connection.

There are two types of data exchanges, host originated and asynchronous. The host originated mode is used to solicit data from the analog channels as well as changing the digital outputs. When the hosts wants to get an update from the analog inputs, it sends the channel number and type information, and the device responds with the current reading, in the range of 00_{16}-FF_{16}.

For digital outputs, the channel number and state are sent in the message, no response is sent. A change in a digital input results in a message being sent asynchronously to the host with no solicitation.

Asterisk (Allstar) only performs the handshake and configuration once after it has been restarted. If the communication fails, the USB mode will be disabled after 2 seconds of inactivity and the COM Led turned off. To restart it requires that asterisk also be restarted. From the console, as a super-user, it can be accomplished with the command:

```
# service asterisk restart
```

WiFi Mode

Once the client data has been configured and, if inserted, the CONF jumper is removed, the reset key is depressed to restart the device. If a connection to the network is successful, the COM led is set to a steady GREEN. The ACT led shows when data traffic is being over the wireless connection.

Periodic updates are sent to the monitoring server based on the interval specified on the configuration page, the inputs and outputs are only scanned once during this interval, responses to a digital output command can be delayed by this time as well.

If both modes are enabled, then the COM Led will alternately flash red and green when BOTH connections are established.

COM Led

The COM Led is a bicolour LED that can be red or green, depending on the operational mode. The LED can either be on or flashing, as illustrated in Table 12.

Color	Mode	Meaning
NONE	Off	No communication mode is currently active.
RED	Flashing	Configuration mode has been entered; AP is active.
RED	Steady	A connection has been made to a WiFi access point; the USB mode has not completed a handshake.
GREEN	Steady	The USB is connected to a host computer and the handshake is compete. No Wifi Connection has been made.
RED/GREEN	Flashing	Both USB and WiFi modes are operational.

Table 12 COM Led modes

WiFi Telemetry Packet Specification

Telemetry packets use the USRP protocol format and are sent using the User Datagram Protocol (UDP) for its transport, which is connectionless. Each packet consists of several fields as illustrated below:

USRP Header 32 Bytes	TLV Header 3 Bytes	Payload Data Up to 255 bytes

Table 13 Telemetry packet format

USRP header

The packet header fields are shown in Table 14.

Field Name	Size (Bytes)	Type	Value	Usage for Telemetry
Eye	4	byte	'USRP' in ASCII	Required
Sequence	4	unsigned integer	Sequencer counter	Incremented by one
Memory	4		Memory ID or zero	Ignored
Keyup	4		1 to indicate transmitter keyed, 0 otherwise	
Talkgroup ID	4		ID of the current talkgroup	
Type	4		Packet type, see Table 15	only TLV used for telemetry
MPXID	4		Multiplex ID	ignored
Reserved	4		Reserved for future use	

Table 14 USRP Header fields

There are 7 defined packet payload types, all telemetry packets are TLV packets. The others are listed purely for documentation purposes.

Packet Type	Definition	Contents
0	Voice	Voice coded as 16-bit linear PCM in little endian format
1	DTMF	DTMF coded as per RFC4733
2	TEXT	Text message
3	PING	Ping message
4	TLV	Type-Length-Value packet
5	ADPCM	Voice coded as adaptive delta pulse code modulation
6	ULAW	Voice coded using G711 µLaw

Table 15 USRP packet types

TLV Header

The TLV header is defined in Table 16.

Byte	Size	Content
1	8-bit unsigned integer	Type field
2		Length of entire payload (including TLV header)
3		Value field, up to 255 bytes

Table 16 TLV packet header

There are 13 different types of TLV payload types as illustrated in Table 17, all telemetry packets use a type 12, the others are listed for documentation purposes.

Type	Payload
0	Begin transmit tag
1	AMBE coded voice packet
2	End transmit tag
3	Tune
4	Play AMBE packet
5	Remote Command
6	AMBE_49
7	AMBE_72
8	Information packet
9	IMBE
10	DSAMBE
11	File transfer
12	Telemetry

Table 17 TLV packet types

The telemetry system uses a type 12 packet.

Payload Data

The payload data contains a radio ID, and a series of data fields, up to the maximum data field size, as shown in Table 18.

Radio ID	Data Field 1	Data Field 2	Data Field n

Table 18 Payload Data fields

Data fields contain three different entries as illustrated in Table 19

Entry	Size (bytes)	Contents
Entry type	1	Identifies the entry type
Number of Entries	1	Number of entries in the field
Field Data	1-2	Field data in big endian format

Table 19 Information packet fields

There are three entry types as shown in Table 20:

Type	Size (bytes)	Data type
0	2	Analog input reading
1	1	Digital inputs, zero=off, non-zero=on
2	1	Digital outputs

Table 20 Entry types

Packet Exchange

The frequency of packets is up to the telemetry device to determine. At the receiver the database is updated immediately, the display is dynamically updated each time a new record is written to the database.

The uploaded packet must contain all field types, all fields must reflect the current state of all inputs and outputs. A response packet is sent back immediately, and if the device supports digital outputs this field will be included, otherwise the payload field will be empty. The response packet will contain the same sequence number in the header as the uploaded packet, so this field can be used by the telemetry device as an implicit acknowledgement.

If no response is heard, then it can be assumed that the link is down, and the packet was not received. The recovery procedure is up to the specific device. It is recommended that local statistics be kept for this purpose.