WiBotic Inc.

4545 Roosevelt Way NE – Suite 400

Seattle, WA 98105

650-722-0679

http://www.wibotic.com

info@wibotic.com

Prepared by: WiBotic Inc

Date Updated: May 2018

Notice: This document is provided for informational purposes only. It represents WiBotic's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessments of the information in this document and any use of WiBotic's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from WiBotic.

Table of Contents

1 Getting Started

The WiBotic Network API is accessible over a WebSocket at the address *ws://192.168.2.20/ws* (where 192.168.2.20 is the IP address of the WiBotic Transmitter) using the "*wibotic"* WebSocket subprotocol. You can use any language or environment that can communicate over WebSockets to send and receive API data to the WiBotic system.

A helper library and demo application are available in Python to assist in development.

NOTE: The WebSocket server does not incorporate any authentication at this time. Do not connect the WiBotic system to a public network without considering the security implications.

NOTE: The WiBotic WebSocket server could be susceptible to an attack where a malicious webpage opens a socket and sends data to the WiBotic system in another isolated network segment. This will be addressed in release versions of the system.

2 General Packet Format

The WiBotic Network API uses binary WebSocket frames. The frames begin with a byte that indicates the frame type (see Response Packet Types and Request Packet Types below). This byte determines how the rest of the frame should be interpreted.

2.1 Response Packet Types

Name	Code	Description
Parameter Update	0x80	A parameter was updated
Parameter Response	0x81	Response to a request for data from a parameter
ADC Update	0x82	New real-time ADC packet

2.2 Request Packet Types

Name	Code	Description
Read Parameter	0x01	Request a parameter to be read
Write Parameter	0x03	Request a parameter to be written

3 API Requests

The following diagrams illustrate the position of the bytes that should make up the binary WebSocket frames that contain the API requests and responses. Multiple successive bytes with the same name indicate that the value is to be split over those bytes. The bytes illustrated continue on to the next line if there are too many bytes to show on one line.

3.1 Building API Requests

3.1.1 Read Parameter

0x01	Device ID	Parameter ID	Parameter ID	Parameter ID	Parameter ID

3.1.2 Write Parameter

0x03	Device ID	Parameter ID	Parameter ID	Parameter ID	Parameter ID
New Data	New Data	New Data	New Data		

3.2 Parsing API Responses

3.2.1 Parameter Update

0x80	Device ID	Parameter ID	Parameter ID	Parameter ID	Parameter ID
Status					

3.2.2 Parameter Response

0x81	Device ID	Parameter ID	Parameter ID	Parameter ID	Parameter ID
Param Data	Param Data	Param Data	Param Data		

3.2.3 ADC Update

0x82	Device ID	ADC ID 1	ADC ID 1	ADC Data 1	ADC Data 1
ADC ID 2	ADC ID 2	ADC Data 2	ADC Data 2	…	ADC ID *n*
ADC ID *n*	ADC Data *n*	ADC Data *n*			

The number of ADC values in a packet can be determined by the following equation:

$$\frac{(Number\ of\ Bytes) - 2}{4}$$

4 Parameters

Name	ID	Description	Read	Write	Availability
Address	3	Address of the device on the internal point to point wireless link	Yes	No	Both
RadioChannel	4	Current device radio channel	Yes	No	Both
DigitalBoardVersion	26	Version of the digital board that is running the system	Yes	No	Both
BatteryCurrentMax	34	Maximum current that should be delivered to the battery in milliamps	Yes	Yes	Both
ChargerCurrentLimit	35	Current limit that the system has decided can be delivered to the battery	Yes	No	RX
MobileRxVoltageLimit	36	Calculated maximum battery voltage based on chemistry, number of cells, and voltage per cell	Yes	No	Both
RxBatteryVoltageMin	37	Calculated minimum battery voltage based on chemistry, number of cells, and voltage per cell	Yes	No	RX
BuildHash	38	Version hash of the firmware that is loaded on the device	Yes	No	Both
TargetFirmwareId	39	Type of firmware image that is running on the device	Yes	No	Both
RxBatteryVoltage	42	Last read ADC value of the battery's voltage	Yes	No	RX
RxBatteryCurrent	43	Last read ADC value of the battery's current	Yes	No	RX
RxTemperature	44	Last read ADC value of the power board's temperature	Yes	No	RX
EthIPAddr	45	Network Static IP Address if no DHCP	Yes	Yes	TX
EthNetMask	46	Network Subnet Mask if no DHCP	Yes	Yes	TX
EthGateway	47	Network Gateway if no DHCP	Yes	Yes	TX
EthDNS	48	Network DNS Server	Yes	Yes	TX
EthUseDHCP	49	Device should use DHCP on the network	Yes	Yes	TX
EthUseLLA	50	Device should fallback to Link-Local Addressing if DHCP fails	Yes	Yes	TX

DevMACOUI	51	MAC Address Organizationally Unique Identifier	Yes	No	Both
DevMACSpecific	52	MAC Address Specific Identifier	Yes	No	Both
EthMTU	54	Ethernet MTU	Yes	Yes	TX
EthICMPReply	55	Reply to Pings	Yes	Yes	TX
EthTCPTTL	56	TCP Time to Live	Yes	Yes	TX
EthUDPTTL	57	UDP Time to Live	Yes	Yes	TX
EthUseDNS	58	Use DNS	Yes	Yes	TX
EthTCPKeepAlive	59	Keep TCP Connections Alive	Yes	Yes	TX
ChargeEnable	60	Enable or disable the ability for this device to transfer power or charge a battery	Yes	Yes	Both
I2cAddress	61	Address of this device on a I2C bus	Yes	Yes	RX
RxBatteryNumCells	62	Number of cells in the battery this charger is configured to charge	Yes	Yes	RX
RxBatterymVPerCell	63	Voltage of a battery cell (in millivolts) that this charger is configured to charge	Yes	Yes	RX
LogEnable	67	Enable logging battery charge data	Yes	Yes	TX
RxBatteryChemistry	68	Chemistry of the attached battery	Yes	Yes	RX
IgnoreBatteryCondition	70	Ignore battery condition when charging. Potentially Dangerous.	Yes	Yes	RX
PowerBoardVersion	71	Version of the power board that is running the system	Yes	No	Both
AccessLevel	78	Current level of access that external entities have to the system.	Yes	Yes	Both

5 Parameter Status Codes

Name	Code	Description
Failure	0	The parameter was not set due to a general failure
Hardware Failure	1	Some hardware did not respond as expected. The parameter was not set.
Invalid Input	2	The data that was to be written to the parameter was not valid for the parameter.
Non-critical Fail	3	The data was not written to the parameter, but this should be anticipated for the given parameter
Read only	4	The parameter is currently in a read only state and should only be read
Success	5	The data to be written to the parameter was written successfully
Not Authorized	6	The current session was not authorized to change the selected parameter

6 Real-time ADC Packets

Name	ID	Data Type	Description
PacketCount	0	uint32_t	Tick timebase in milliseconds
ChargeState	1	uint8_t	Current state of system
Flags	2	uint16_t	Flags
PowerLevel	3	uint16_t	Output power level
VMon3v3	4	float	3.3v voltage
VMon5v	5	float	5v voltage
IMon5v	6	float	5v current
VMon12v	7	float	12v voltage
IMon12v	8	float	12v current
VMonGateDriver	9	float	Gate Driver voltage
IMonGateDriver	10	float	Gate Driver current
VMonPA	11	float	Power Amplifier voltage
IMonPa	12	float	Power Amplifier current
TMonPa	13	float	Power Amplifier temperature
VMonBatt	14	float	Battery Voltage
VMonBattProg	15	float	Charger Voltage
VRect	16	float	Rectified Voltage
TBoard	17	float	Board Temperature
ICharger	18	float	Charger current
IBattery	19	float	Battery current
TargetIBatt	20	float	Target battery current
IMaster	21	float	Charger 1 current
ISlave1	22	float	Charger 2 current
ISlave2	23	float	Charger 3 current

7 Device ID

Device	Address
Transmitter	1
Charger	2