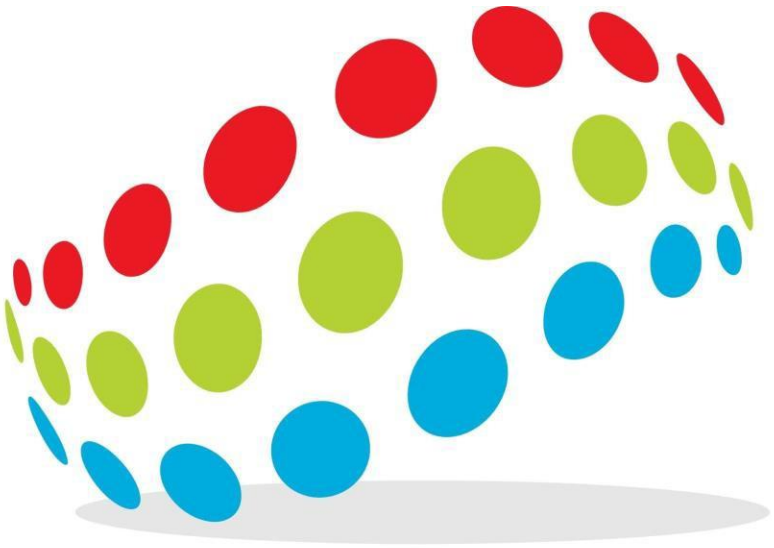
FJ2100 User Manual



Positioning Universal Inc.

Version 1.2

1 Programming and Configuration

FJ2100 can be programmed and configured either via the serial UART connection or via an Over the Air (OTA) process on the cellular data network.

Operation Temperature: -40°C to +85°C ； The max. operating temperature is Charging: 45°C, Discharging: 85°C.

1.1 Firmware, Application, and Settings Files

There are multiple levels of software and settings files contained on the device:

1. Modem software is flashed onto the cellular modem at the factory. This firmware controls communications between the modem and the cellular carrier. In the very rare event that this firmware needs to be upgraded, it will be performed by Positioning Universal.
2. Device operating system firmware manages the control of modules on the device and provides the operating environment for Applications. In the very rare event that this operating system firmware needs to be upgraded, it will be performed by Positioning Universal.
3. Applications are written in JavaScript and can be loaded to the device via the serial UART or OTA. Applications can be updated to add functionality or improve performance when opportunities for improvement are found.
4. Settings files are loaded to the device via the serial UART or OTA. Settings files contain the parameters used by the application to control tracking and other behaviors.

1.2 DMAN Server

A Device Manager Server on the Positioning Universal infrastructure automatically updates Applications and Settings files for groups of devices. When an update of the application or new settings are released, they are loaded into the DMAN server and assigned to the Groups of devices that are to receive the update. The DMAN server automatically updates devices to the latest assigned versions. In normal operation, devices “check in” to the DMAN server regularly to report their health, and to check whether they are due to get updates.

Device will take the updates either immediately if commanded via the “update” command, or during the next quiescent period, before entering Sleeping state.

1.3 UART Programming, Trace, and Commands

In the factory and on a test bench, the serial UART connection is used to flash an image containing the root firmware, application, and default settings files using special factory

software. Specific queries, commands, and debug tracing messages are also available over the UART. In normal operation, the UART is only used for debug, command, and tracing purposes. Custom applications may use this UART to communicate with accessories.

The serial UART consists of RX, TX, and Ground signals on the 4 pin UART connector. Communications uses a TTL level of 3.3V. A supply of 3.3V is available on this connector to power a low power accessory (like a logger). NOTE: supplying 3.3V to the FJ2100 via this interface from a USB-TTL serial adapter will power the processor, but will not provide enough power for the cellular modem to work.

To communicate between a computer and the device using the UART, it is necessary to have a serial terminal program installed on the computer. Several examples are Hyperterm and Realterm on Windows, and CoolTerm on either Windows or Mac computers. Communications is at 115200 baud, 8N1.

The most common way to physically connect a computer to the device is to attach a USB-TTL serial adaptor to the 4 pin serial harness. Note that the optional Vdd wire (red) is NOT usually connected for basic communications. Examples of suitable USB-TTL serial adapters include those that incorporate the MAX3232 integrated circuit, or the CH340 chipset. Note that you may need to download and install drivers for these devices on a computer in order to use them. There are many different versions of these adapters available. Typically, the device that incorporate a crystal perform better. One version is available on Amazon from this link: Serial Converter adapter

Once connected, trace information will be sent as ASCII data over the UART interface. This allows the user to monitor operation of the device and see how it responds to various occurrences, as well as see a report of GPS status every 10 seconds. Queries and Commands shown in the next chapter can be sent to the device from the computer and responses seen in the terminal window.

1.4 SMS Queries and Commands

The queries and commands shown in the following chapter of this guide can be sent by SMS to the phone number of the device, and responses will be sent to the phone number of the device used to send the query or command.

1.5 UDP Queries and Commands

The queries and commands shown in the following chapter of this guide can be sent by UDP to the IP and port of the device. Due to cellular carrier security features, these queries and commands are usually queued on the sending server, and sent immediately after the server receives a message from the device. In order to indicate that the UDP

message is a command, it is necessary to prepend the equals sign = to the front of the command. For example, to send the **agps** command to a device via UDP, it is necessary to send it as **=agps** so that the device sees the = first and recognizes the following data as a command.

Queries and Commands

Queries and commands can be sent to a device via the serial UART, an SMS message, or via a UDP message over cellular. All queries and most commands will generate a response if successful that begins with the word PASS. If a command is not recognized by the device, it will respond with a message beginning with the word FAIL.

1.6 Queries

Queries request status information from the device, and the device responds accordingly.

Version

The device responds with the version numbers of the application and settings file and other information. An example response to the **version** query is:

```
PASS::version;app=3023 sett=3023 firm=3124 cell=4.3.1.2 [29492]
imei=354196070019771 imsi=311480142180885
iccid=89148000001404515503 msidn=15889776934
```

Results in device reporting its:

- current application version
- settings version
- Firmware version
- Cellular modem firmware version
- IMEI for device
- IMSI for device
- ICCID
- MSISDN (phone number)

Getgps

The device responds with current GPS status and location information. An example response to the **getgps** query is:

```
PASS::getgps;17-8-17 21:30:47 49.26447 -123.125595 s11/0.9 0.0g
0.0k 309.8* 41.1V 4.2B 11101110 -115dbm  good=0.1,49.26447,-
123.125595 reset=0
```

Where the message can be broken into:

- Fix time
- Latitude
- Longitude
- # satellites, sX where X is number
- Horizontal Accuracy, /X where X is number
- Absolute G force, Xg where X is number
- Speed in kph, Xk
- Heading in degrees, X*
- VIN in Volts, XV
- Battery Voltage in Volts, XB
- Outputs/Inputs 0 = off, 1 = on, Output 3,2,1,0 then Input 3,2,1,0
- Rssi, Xdbm
- Time in seconds since Last good GPS fix, good=X
- Last good GPS lat, ,X
- Last good GPS lng, ,X
- Number of GPS resets since boot, reset=X

Getcomm

The device responds with current status of the cellular connection and details of the connection. An example response to the **getcomm** query is:

```
 PASS::getcomm;avail=true change=1136.6 wake=51.1 rssi=-51
lastAT=1.9 lastSent=17.0 lastRcvd=14.6 woAck=65536.4
cell=13911830 lac=13826 reset=0
```

Which consists of

- com available true or false
- time since last comm state change in seconds
- time from last comm module wake in seconds
- current rssi value
- time in seconds since last AT command was received
- time in seconds since last UDP message was sent
- time in seconds since last UDP message was received
- age in seconds of device when last comm was sent without an ack
- cell tower id
- local area code

- number of reset of comm module since device boot.

Clock

The device responds with the time as reported by the cellular modem, GPS, and CPU. An example response to the **clock** query is:

```
PASS::clock;age=66161.10435009765 mcu=17-8-15 22:54:52 gps=17-8-
15 22:54:32 cell=17-8-15 22:54:27
```

Which gives:

- the age in seconds device since boot,
- the current mcu time
- the last reported good gps time
- the last good cell time
- Note that GPS and Cellular Timestamps are not updated while in low power or deep sleep mode.

State

The device responds with the states the device is currently in. An example response to the **state** query is:

```
PASS::state;Moving heart=1762.9 ignOn=none vibr=131.3
ColdBoot=135.2 lastVolt=42.1 pwrOn=true mem=1408 sid=false
```

Which consists of:

- current state (Moving, Stopped, Sleeping)
- the time in seconds to next heartbeat
- whether ignition is on
- time since last vibration event in seconds
- the time since last Cold Boot in seconds
- the last reported vehicle voltage (Vin)
- whether power is connected to the device
- SID (starter interrupt device) is active or not

where

Returns a google map url using the last good lat lng reported by device. Used through SMS to quickly map the the device's current location.

Example:

```
PASS::where;http://maps.google.com/?49.26445910000,-
```

123.12561510000

settings
The **settings** command will respond with all values of the settings file in json format.

1.7 Commands

Commands force the device to take specified actions. They can be sent via SMS, a serial UART, or via the cellular data connection.

report,<event code>
The **report** command forces the device to send a regular event message with the event code specified in the command.
Eg. **report,9** would force a report with event code 9.

reboot
The **reboot** command forces a warm boot of the main processor and sub modules of the device. The reboot will occur in 5 seconds to give the device a chance to send a result in command ack based upon sms or udp delivery. If comm fails then no ack may be received even though the device does actually reboot. The device reports a WARM_BOOT message when it powers back up.

sms,<to phone number>,<payload>
Send an SMS message with specified payload to specified phone number.

update
Checks latest versions of application and settings for a device on DMAN and if different than existing application and/or setting then start update.

bootin
Sends a boot checkin message to DMAN, which updates status and statistics for the device.

1.8 AT Commands to Modem Module

AT commands can be sent directly to the cellular modem module on the device using the command **sendAT,<ATcmd>**

For example, **sendAT,CEREG?** will send the command "AT+CEREG?" to the cellular modem.

NOTE: Sending AT commands to the cellular modem can change critical settings on the cellular modem and may interfere with its operation and block communications. This command is only to be used by developers familiar with the use of AT commands on cellular modems.

1.9 SAFETY INFORMATION

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2)this device must accept any interference received, including interference that may cause undesired operation.
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
—Reorient or relocate the receiving antenna.
—Increase the separation between the equipment and receiver.
—Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
—Consult the dealer or an experienced radio/ TV technician for help.
Changes or modifications not expressly approved by the manufacturer could void the user' s authority to operate the equipment.

FCC RF Radiation Exposure Statement
This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. To comply with FCC RF Exposure compliance requirements, this grant is applicable to only Mobile Configurations. The antennas used for the transmitter must be installed to provide a separation distance of at least 20cm from all persons and must not be co‐located or operating in conjunction with any other antenna or transmitter.

IC STATEMENT
This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

In order to avoid the possibility of exceeding the IC radio frequency exposure limits, human proximity to the antenna shall not be less than 20cm (8 inches) during normal operation.

Afin d'éviter la possibilité de dépasser les limites d'exposition aux fréquences radio de la IC CNR102, la proximité humaine à l'antenne ne doit pas être inférieure à 20 cm (8 pouces) pendant le fonctionnement normal.