

PX4_JAVAD_GREIS

Technical Design and Integration Report - Tested Revision

Field	Source-defined value
Subsystem	PX4 generic GPS driver
JAVAD driver/helper	GPSDriverJavadGreis
Main GPS protocol selection	GPS_1_PROTOCOL = 7 (JAVAD GREIS)
Receiver protocol	JAVAD GREIS/JPS
Helper default baud	115200 when configure() receives baudrate = 0

Source scope: This report is derived only from the five supplied source files: CMakeLists.txt, gps.cpp, params.yaml, javad_greis.h, and javad_greis.cpp. No board-specific port assignments, external documentation, runtime logs, or previous test results are used as technical evidence.

1. Executive Summary

GPSDriverJavadGreis integrates JAVAD GREIS/JPS receiver support into the PX4 generic GPS driver. The generic GPS module owns the device path, opens the serial interface, selects the protocol, creates the helper, and publishes sensor_gps. GPSDriverJavadGreis configures the receiver output stream, parses recognized GREIS frames, validates checksums, decodes navigation and quality data, assembles complete epochs, and fills sensor_gps.

For the main GPS, the supplied parameter metadata defines JAVAD GREIS as GPS_1_PROTOCOL value 7. The parameter requires a reboot after change. In gps.cpp, value 7 selects the JAVAD mode and constructs GPSDriverJavadGreis. JAVAD is not present in the passive auto-detection list, so explicit protocol selection is the source-defined path.

2. Customer Configuration Summary

Item	What the supplied source defines
Build inclusion	javad_greis.cpp is compiled when CONFIG_GPS_JAVAD is enabled.
Main GPS protocol	Set GPS_1_PROTOCOL to 7 (JAVAD GREIS). The parameter is marked reboot_required.
Protocol detection	JAVAD is excluded from kAutoDetectModes; use explicit protocol selection.
Serial device	Not hard-coded in the JAVAD helper. The generic GPS module receives and opens the configured device path.
Baudrate	If the helper receives baudrate = 0, it sets 115200. If a nonzero baudrate is supplied, the helper keeps it.
Receiver stream	The helper configures RT, GT, ST, PG, VG, SG, DP, PS, and ET at a 0.1 s period.

3. Integration Files

The supplied source set contains the following JAVAD-related integration points. This table describes the role visible in the supplied files; it does not classify files as new or modified relative to another repository revision.

Source file	JAVAD-related role
CMakeLists.txt	Adds devices/src/javad_greis.cpp to the GPS driver sources when CONFIG_GPS_JAVAD is enabled.
gps.cpp	Adds JAVAD protocol mode, maps protocol value 7, constructs GPSTDriverJavadGreis, assigns the JAVAD device-type symbol, reports the protocol name, and publishes sensor_gps.
params.yaml	Defines JAVAD GREIS as value 7 for the GPS protocol parameter metadata and marks the protocol parameter as reboot-required.
javad_greis.h	Declares GPSTDriverJavadGreis, parser/configuration state, message set, epoch data, payload limits, timing constants, and helper functions.
javad_greis.cpp	Implements receiver configuration, frame parsing and checksum validation, message decoding, epoch assembly, navigation conversion, and sensor_gps population.

4. Overall Architecture

```

JAVAD GNSS receiver
| GREIS/JPS serial data
v
PX4 generic GPS module (gps.cpp)
| device path / baud / protocol selection
v
GPSTDriverJavadGreis
| configure -> parse -> checksum -> decode -> epoch assembly
v
sensor_gps
|
v
uORB publication by gps.cpp

```

Layer	Responsibility visible in the supplied source
Generic GPS module	Stores the device path, opens the serial interface, applies a configured baudrate when present, selects the driver mode, creates the helper, and publishes sensor_gps.
GPSTDriverJavadGreis	Uses GPSHelper callbacks for device I/O, configures the receiver stream, parses and validates GREIS frames, decodes messages, assembles epochs, and fills sensor_gps.
sensor_gps	Receives the standardized position, velocity, fix, satellite, DOP, and accuracy fields populated by the JAVAD helper.

5. Protocol Selection and Driver Creation

params.yaml exposes JAVAD GREIS as value 7 for GPS_1_PROTOCOL. The parameter default is 1, so JAVAD is not the default selection. gps.cpp reads the protocol parameter and maps value 7 to gps_driver_mode_t::JAVAD when CONFIG_GPS_JAVAD is compiled in.

```

GPS_1_PROTOCOL = 7
-> gps_driver_mode_t::JAVAD
-> GPSTDriverJavadGreis(&GPS::callback, this, &sensor_gps)
-> JAVAD device-type symbol assigned by gps.cpp

```

The supplied `gps.cpp` explicitly excludes JAVAD from `kAutoDetectModes`. The status path reports the selected mode as "protocol: JAVAD GREIS". A command-line protocol name "javad" is also recognized by the module, but the parameter selection above is the customer-facing configuration defined in `params.yaml`.

6. Serial Interface and Baudrate Behavior

`GPSTDriverJavadGreis` does not contain a board-specific serial-device name. The generic GPS module receives the device path, stores it, configures the UART port, optionally applies a configured baudrate, and opens the device before the helper runs. This keeps board-specific device mapping outside the JAVAD helper.

Condition	Source-defined behavior
Configured baudrate is nonzero	<code>gps.cpp</code> applies that baudrate before opening the UART. The JAVAD helper does not replace it with the default.
Helper receives baudrate = 0	<code>GPSTDriverJavadGreis</code> calls <code>setBaudrate(115200)</code> and returns 115200 through the baudrate reference.
Serial device path	Owned by the generic GPS module; no fixed device path is defined in <code>javad_greis.cpp</code> or <code>javad_greis.h</code> .

7. Receiver Auto-Configuration

The JAVAD helper uses a three-state configuration sequence: `DisableOutput`, `EnableStream`, and `Complete`. `configure()` resets parser and epoch state, initializes the state machine, applies the 115200 fallback only when needed, and immediately starts configuration.

State	Behavior
<code>DisableOutput</code>	Send <code>dm</code> followed by LF. On successful write, move to <code>EnableStream</code> and wait 200 ms.
<code>EnableStream</code>	Send the configured GREIS/JPS stream command. If the required message types have not all been observed, retry after 3 seconds.
<code>Complete</code>	Stop sending configuration commands after GT, RT, ST, ET, PG, and VG have all been observed.

Configuration-complete condition: SG, DP, and PS are requested by the stream command but are not required for the configuration state to become Complete. The required observation set is GT + RT + ST + ET + PG + VG.

8. ASCII Commands Sent to the Receiver

The supplied helper defines two receiver commands. The first disables periodic output. The second requests the GREIS/JPS messages used by the navigation implementation at a 0.1 s period.

```
dm\n
dm,;em,,jps/{RT,GT,ST,PG,VG,SG,DP,PS,ET}:0.1\n
```

The configured periodic stream requests RT, GT, ST, PG, VG, SG, DP, PS, and ET. PV is recognized and decoded by the helper as a fallback navigation source, but PV is not included in this configured stream command.

9. GREIS Message Support

Recognized message	Role in the implementation
GT	GPS week and time-of-week used by the epoch data. GT may be kept pending until an RT-started epoch exists.

Recognized message	Role in the implementation
RT / ~~	Starts a new epoch. The previous epoch state is reset and the RT time key is stored.
ST	Adds solution type to the active epoch. ST is accepted for the epoch only when its time key matches the active RT key.
ET / ::	Marks the active epoch as closed.
PG	Primary geodetic position source: latitude, longitude, ellipsoid height, position sigma, and solution type.
VG	Primary velocity source: north, east, up, velocity sigma, and solution type. Up is negated for NED Down.
PV	Fallback ECEF position/velocity source. Position is converted to geodetic coordinates and velocity to NED before publication.
SG	Optional horizontal, vertical, and speed accuracy data.
DP	Optional HDOP and VDOP data.
PS	Optional satellite-used count and solution-status data.

10. Parser and Frame Validation

The parser is a byte-by-byte state machine with the sequence ID1 -> ID2 -> LEN1 -> LEN2 -> LEN3 -> BODY. It accepts two printable ASCII message-ID bytes and reads the frame length from three uppercase hexadecimal ASCII characters.

- CR and LF are ignored while the parser is waiting for the first message-ID byte.
- The advertised body length is limited to 0xFFFF. The local payload buffer stores up to 128 bytes while the parser still consumes the complete advertised body.
- Recognized messages are checked against a message-specific minimum length before decoding.
- The recognized GREIS messages in this implementation use checksum validation based on the rotate/XOR checksum functions in javad_greis.cpp.
- Unknown messages are consumed using their advertised length and are not dispatched to a decoder.

After a recognized message passes length and checksum checks, dispatchMessage() calls the corresponding decode function and then attempts epoch publication.

11. Epoch Synchronization

Message	Epoch behavior
RT / ~~	Only epoch starter. resetEpoch() is called, the epoch becomes valid/open, and a pending GT is attached if available.
GT	Stores GPS week/time-of-week. It is attached to an active RT epoch when possible or held pending for the next RT.
ST	Accepted only for an open RT epoch and only when ST time equals the stored RT time.
Navigation / quality messages	Accepted only while the epoch is valid, has RT, has not received ET, and has not already been published.
ET / ::	Marks the active epoch closed.

epochComplete() requires a valid, unpublished epoch with RT, ST, ET, GT, and navigation data. Navigation is complete when PG + VG is available, or when the PV fallback has been successfully converted to geodetic position and NED velocity. Each epoch is published at most once.

12. Navigation Logic

Primary navigation:

PG + VG -> geodetic position + NED velocity

Fallback navigation:

PV -> ECEF position -> geodetic latitude / longitude / ellipsoid height
 -> ECEF velocity -> NED velocity

PG + VG has priority. PV is used only when the primary position/velocity pair is not complete and the PV conversion succeeds; a valid PV fallback does not override a complete PG + VG solution.

PG latitude and longitude are converted from radians to degrees. VG north/east values are used directly and its Up component is negated to produce NED Down. The PV path checks ECEF values for finiteness and a plausible Earth-radius range before conversion.

13. sensor_gps Population

sensor_gps field group	Source-defined behavior
Timestamp	timestamp and timestamp_sample use the current GPS-helper time. time_utc_usec is set to 0.
Latitude / longitude	Published in degrees from PG, or from converted PV fallback.
Altitude	altitude_ellipsoid_m receives ellipsoid height. altitude_msl_m is assigned the same value because the configured stream does not provide geoid separation.
Velocity / course	NED velocity is published; horizontal speed is derived from N/E velocity and course-over-ground from atan2(E, N).
Fix type	Derived from JAVAD solution-type fields. The implementation selects the lowest mapped fix value among the available solution-type inputs for the epoch.
Satellites used	Taken from PS when PS data is available; otherwise 0.
DOP	HDOP and VDOP are taken from DP when available; otherwise both use the internal unknown-DOP value 99.9.
Accuracy	SG supplies horizontal, vertical, and speed accuracy. If SG is absent, position/velocity sigma values from the navigation path are used when valid; otherwise the corresponding fields are set to 0.
Heading	The JAVAD helper does not populate heading. gps.cpp initializes and clears sensor_gps.heading to NAN when no heading update is available.

Fix-type mapping: The source maps solution type 1 to 3D, 2 to differential, 3 to RTK Float, 4 to RTK Fixed, 5 to 3D as a conservative mapping, and 0/other values to no fix. This statement documents the mapping implemented in solutionTypeToFix().

14. Runtime Sequence

```

PX4 generic GPS module starts
-> receives device path and configured baudrate
-> GPS_1_PROTOCOL = 7 selects JAVAD mode
-> opens the serial device
-> creates GPSTDriverJavadGreis
-> configure()
    -> reset parser / epoch state
    -> use 115200 only when baudrate == 0
    -> send receiver stream configuration
  
```

```
-> receive()
    -> read bytes through GPSHelper callback
    -> parse -> checksum -> decode
    -> assemble epoch -> publishEpoch()
-> gps.cpp publishes sensor_gps
```

15. Known Behaviors and Limitations

Item	What the supplied source shows
Explicit JAVAD selection	JAVAD is excluded from passive auto-detection. For the main GPS, protocol value 7 is the defined selection.
UTC output	GT week/time-of-week is decoded into epoch state, but publishEpoch() sets sensor_gps.time_utc_usec to 0.
MSL altitude	No geoid separation is available in the configured stream, so altitude_msl_m is assigned the ellipsoid height.
Heading	The JAVAD helper does not provide a GNSS heading value.
PV fallback availability	PV decoding exists, but PV is not requested by the helper's default periodic stream command.

16. Conclusion

The supplied source implements JAVAD GREIS/JPS support as a dedicated GPSTDriverJavadGreis helper integrated into the PX4 generic GPS driver. The customer-facing protocol selection is GPS_1_PROTOCOL = 7, and the source marks that parameter as reboot-required. The helper does not hard-code a board-specific port and uses 115200 only as a fallback when no baudrate is supplied to configure().

The receiver is configured to output RT, GT, ST, PG, VG, SG, DP, PS, and ET at 10 Hz. The parser validates recognized GREIS frames, assembles RT-based epochs, uses PG + VG as the primary navigation source, retains PV as a fallback decoder, and fills sensor_gps with position, velocity, fix, satellite count, DOP, and accuracy data according to the available messages.

Source-set boundary: The supplied files reference CONFIG_GPS_JAVAD and the JAVAD device-type symbol, but the definitions of those symbols and board-specific serial-port assignments are outside the supplied source set. This report therefore does not assign numeric values or board-specific device names to them.

Source Reference Summary

Topic	Supplied source reference
Build and parameter integration	CMakeLists.txt - CONFIG_GPS_JAVAD source inclusion; params.yaml - GPS_1_PROTOCOL value 7 = JAVAD GREIS and reboot_required.
Generic GPS integration	gps.cpp - JAVAD mode, protocol value 7 mapping, helper construction, status name, serial-device ownership, and sensor_gps publication.
Driver structure	javad_greis.h - parser/configuration states, recognized messages, epoch structure, payload limits, and timing constants.
JAVAD driver implementation	javad_greis.cpp - receiver configuration, parser/checksum, message decoding, epoch completion/publication, fix mapping, and ECEF-to-NED conversion.