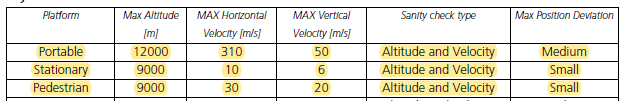
# UBX-7 – Important Stuff:

* **2.1 Platform settings**
  + Portable
    - Applications with low acceleration, e.g. portable devices. Suitable for most situations.
  + Stationary
    - Used in timing applications (antenna must be stationary) or other stationary applications. Velocity restricted to 0 m/s. Zero dynamics assumed.
  + Pedestrian
    - Applications with low acceleration and speed, e.g. how a pedestrian would move. Low acceleration assumed.



* **Navigation Output Filters**
  + Where a fix has been achieved, a check is made to determine whether the fix should be classified as valid or not. A fix is only valid if it passes the navigation output filters as defined in UBX-CFG-NAV5. In particular, both PDOP and accuracy values must lie below the respective limits.
    - Valid fixes are marked using the valid flag in certain NMEA messages (see Position Fix Flags in NMEA)
* Static Hold
  + This reduces the position wander caused by environmental factors such as multi-path and improves position accuracy especially in stationary applications
  + If the speed drops below the defined ‘Static Hold Threshold’, the Static Hold Mode will be activated.
* For maximum GPS coldstart sensitivity, ensure that the SBAS subsystem is enabled.
* **5.1 SBAS (Satellite Based Augmentation Systems)**
  + WAAS (Wide Area Augmentation System) for North America has been in operation since 2003.
  + With SBAS enabled the user benefits from additional satellites for ranging (navigation). u-blox GPS technology uses the available SBAS Satellites for navigation just like GPS satellites, if the SBAS satellites offer this service.
  + u-blox receivers are capable of receiving multiple SBAS signals in parallel, even from different SBAS systems (WAAS, EGNOS, MSAS, etc.). They can be tracked and used for navigation simultaneously. Every SBAS satellite tracked utilizes one vacant receiver tracking channel. Only the number of receiver channels limits the total number of satellites used. Each SBAS satellite, which broadcasts ephemeris or almanac information, can be used for navigation, just like a normal GPS satellite.
  + For receiving correction data, the u-blox GPS receiver automatically chooses the best SBAS satellite as its primary source. It will select only one since the information received from other SBAS satellites is redundant and/or could be inconsistent. The selection strategy is determined by the proximity of the satellites, the services offered by the satellite, the configuration of the receiver (Testmode allowed/disallowed, Integrity enabled/disabled) and the signal link quality to the satellite.
  + Planning is crucial to determine the best possible configuration, especially in areas where signals from different SBAS systems can be received:
  + **Example 1: SBAS Receiver in North America**
    - In the eastern parts of North America, be careful that EGNOS satellites do not take preference over WAAS satellites, the satellites from the EGNOS system should be disallowed using the PRN Mask.
  + **Although u-blox receivers try to select the best available SBAS correction data, it is recommended to configure them to disallow using unwanted SBAS satellites.**
  + **5.3 SBAS Configuration**
    - To configure the SBAS functionalities use the UBX proprietary message UBX-CFG-SBAS (SBAS Configuration).
    - By default, SBAS is enabled with three prioritized SBAS channels and it will use any received SBAS satellites (except for those in test mode) for navigation, ionosphere parameters and corrections.
* The relevant GPS to UTC conversion parameters are transmitted periodically (every 12.5 minutes) by GPS satellites, but can also be supplied to the receiver via the aiding message UBX-AID-HUI
* **8.1 Configuration Concept**
  + The Current Configuration can be changed during normal operation by sending any UBX-CFG-XXX message to the receiver over an I/O port.
  + Unless the Current Configuration is made permanent by using UBX-CFG-CFG as described below, the Current Configuration will be lost in case of:
    - a power cycle
    - a hardware reset
    - a (complete) controlled software reset
  + See the section on resetting a receiver for details. The Current Configuration can be made permanent (stored in a non-volatile memory) by saving it to the "Permanent Configuration".
* **Permanent Configuration Storage Media**
  + The Current Configuration is stored in the receiver's volatile RAM. Hence, any changes made to the Current Configuration without saving will be lost if any of the reset events listed in the section above occur. By using UBX-CFG-CFG/save, the selected configuration sub-sections are saved to all non-volatile memories available:
    - On-chip BBR (battery backed RAM). In order for the BBR to work, a backup battery must be applied to the receiver.
    - External flash memory, where available.
* **9 Forcing a Receiver Reset**
  + Cold start
    - In this mode, the receiver has **no information** from the last position (e.g. time, velocity, frequency etc.) at startup. Therefore, the receiver must **search the full time and frequency space, and all possible satellite numbers**. If a satellite signal is found, it is tracked to decode the *ephemeris* (18-36 seconds under strong signal conditions), whereas the **other channels continue to search satellites**. Once there is a sufficient number of satellites with valid ephemeris, the receiver can calculate position and velocity data. Please note that some competitors call this startup mode Factory Startup.
  + Warm start
    - In Warm start mode, the receiver has **approximate information** for time, position, and coarse satellite position data (***Almanac***). In this mode, after power-up, the receiver normally needs to **download ephemeris** before it can calculate position and velocity data. As the **ephemeris data usually is outdated after 4 hours**, the receiver will typically start with a **Warm start if it has been powered down for more than 4 hours**. In this scenario, several augmentations exist. See the section on Aiding and Acquisition.
  + Hot start
    - In Hot start, the receiver was **powered down only for a short time (4 hours or less)**, so that its **ephemeris is still valid**. Since the receiver doesn't need to download ephemeris again, this is the fastest startup method.
  + In the UBX-CFG-RST message, one can force the receiver to reset and clear data, in order to see the effects of maintaining/losing such data between restarts. For this, the CFG-RST message offers the navBbrMask field, where Hot, Warm and Cold starts can be initiated, and also other combinations thereof.
  + **Data stored in flash memory is not cleared by any of the options provided by UBX-CFG-RST. So, for example, if valid AlmanacPlus data stored in the flash it is likely to have an impact on a "Cold start".**
* **10 Remote Inventory**
  + The Remote Inventory enables storing user-defined data in the non-volatile memory of the receiver. The data can be either binary or a string of ASCII characters.
* **11 Power Management (Better off just reading from the pd for this section)**
  + The receiver is not able to download satellite data (e.g. the ephemeris) while it is working in ON/OFF or cyclic tracking operation
  + **When enabling Power Save Mode, SBAS support can be disabled since the receiver will be unable to download any SBAS data in this mode.**
  + Enabling the **AssistNow Autonomous** feature will lead to increased power consumption while prediction is calculated. The main goal of PSM is to reduce the overall power consumption. Therefore for each application special care must be taken to judge whether AssistNowAutonomous is beneficial to the overall power consumption or not.
* **11.2.2.6 Update RTC and Ephemeris**
  + To maintain the ability of a fast start-up, the receiver needs to calibrate its RTC and update its ephemeris data on a regular basis. This can be ensured by activating the update RTC and update Ephemeris option. The RTC is calibrated every 5 minutes and the ephemeris data is updated approximately every 30 minutes. See chapter Satellite data download for more information.
    - **Could use the RTC for the Arduino also?**
* **11.3 Peak current settings**
  + The peak current during acquisition can be reduced by activating the corresponding option. A peak current reduction will result in longer start-up times of the receiver.
  + **This setting is independent of the activated mode (Continuous or Power Save Mode).**
* **13.2 Jamming/Interference Indicator**
  + The field jamInd of the UBX-MON-HW message can be used as an indicator for continuous wave (narrowband) jammers/interference only. The interpretation of the value depends on the application. It is necessary to run the receiver in the application and then calibrate the 'not jammed' case. If the value rises significantly above this threshold, this indicates that a continuous wave jammer is present.
  + This indicator is always enabled.
* **13.3 Jamming/Interference Monitor (ITFM)**
  + The field *jammingState* of the MON-HW message can be used as an indicator for both broadband and continuous wave (CW) jammers/interference**. It is independent of the (CW only) jamming indicator described in Jamming/Interference Indicator above.**
  + This monitor reports whether jamming has been detected or suspected by the receiver. The receiver monitors the background noise and looks for significant changes.
    - Normally, with no interference detected, it will report 'OK'.
    - If the receiver detects that the noise has risen above a preset threshold, the receiver reports 'Warning'.
    - If in addition, there is no current valid fix, the receiver reports 'Critical'.
  + **The monitor is disabled by default**. The monitor is enabled by sending an appropriate UBX-CFG-ITFM message with the enable bit set. In this message it is also possible to specify the thresholds at which broadband and CW jamming are reported. These thresholds should be interpreted as the dB level above 'normal'. It is also possible to specify whether the receiver expects an active or passive antenna.
  + The monitor algorithm relies on comparing the currently measured spectrum with a reference from when a good fix was obtained. Thus the monitor will only function when the receiver has had at least one (good) first fix, and will report 'Unknown' before this time.
  + **Jamming/Interference monitor is not supported in Power Save Mode (PSM) ON/OFF mode.**
* **15.2 Startup Strategies**
  + **Cold start:** 
    - In this startup mode, the receiver has no information about last position, time, velocity, frequency etc. Therefore, the receiver has to search the full time- and frequency space, and also all possible satellite numbers. **If a satellite signal is found, it is being tracked to decode ephemeris (18-36 seconds under strong signal conditions), whereas the other channels continue to search satellites**. Once there are sufficient number of satellites with valid ephemeris, the receiver can calculate position- and velocity data. Note that some competitors call this startup mode Factory Startup.
  + **Warm start:** 
    - In Warm start mode, the receiver has approximate information of time, position, and coarse data on Satellite positions (**Almanac**). In this mode, after power-up, the receiver basically needs to download ephemeris until it can calculate position- and velocity data. As the ephemeris data usually is outdated after 4 hours, the receiver will typically start with a warmstart if it was powered down for more than that amount of time. For this scenario, several augmentations exist. See the sections on AssistNOW online and offline below.
  + **Hot start:** 
    - In Hot start, the receiver was powered down only for a short time (4 hours or less), so that its ephemeris is still valid. Since the receiver doesn't need to download ephemeris again, this is the fastest startup method. In the UBX-CFG-RST message, one can force the receiver to reset and clear data, in order to see the effects of maintaining/losing such data between restarts. For that, the UBX-CFG-RST message offers the navBbrMaskfield, where Hot, Warm and Cold starts can be initiated, and also other combinations thereof.
* **15.3 Aiding / Assisted GPS (A-GPS)**
  + **Users expect instant position information. With standard GPS this is not always possible because at least four satellites must transmit their precise orbital position data, called ephemeris, to the GPS receiver. Under adverse signal conditions, data downloads from the satellites to the receiver can take minutes, hours or even fail altogether**.
  + Assisted GPS (A-GPS) boosts acquisition performance by providing data such as ephemeris, almanac, accurate time and satellite status to the GPS receiver via mobile networks or the Internet. The aiding data enables the receiver to compute a position within seconds, even under poor signal conditions.
* **15.7.3.3 Sample Code**
  + **u-blox makes available sample code, written in C language, showing a server implementation, serving ALP data from its file system to a client. Please contact your nearest u-blox Field Application Engineer to receive a copy.**
* **15.8 AssistNow Autonomous**
  + The AssistNow Autonomous feature provides a functionality similar to AssistNow Offline without the need for a host and a connection
  + Based on a broadcast ephemeris downloaded from the satellite (or obtained by AssistNow Online) the receiver can autonomously (i.e. without any host interaction or online connection) generate an accurate satellite orbit representation («AssistNow Autonomous data») that is usable for navigation much longer than the underlying broadcast ephemeris was intended for. This makes downloading new ephemeris or aiding data for the first fix unnecessary for subsequent start-ups of the receiver.
* **15.8.4 Benefits and Drawbacks**
  + AssistNow Autonomous can provide quicker start-up times (lower the TTFF) provided that data is available for enough visible satellites.
  + It is, however, required that the receiver roughly know the absolute time, either from an RTC or from time-aiding, and that it knows which satellites are visible, either from the almanac or from tracking the respective signals
  + AssistNow Autonomous will typically extend a broadcast ephemeris for up to three days.
  + Position fixes that include AssistNow Autonomous orbit information may be significantly worse than fixes using only broadcast ephemerides. It might be necessary to adjust the limits of the Navigation Output Filters.
  + A fundamental deficiency of any system to predict satellite orbits precisely is unknown future events
  + The calculations required for AssistNow Autonomous are carried out on the receiver. This requires energy and users may therefore occasionally see increased power consumption during short periods (several seconds, rarely more than 60 seconds) when such calculations are running. Ongoing calculations will automatically prevent the power save mode from entering the power-off state. The power-down will be delayed until all calculations are done.
  + **The AssistNow Offline and AssistNow Autonomous features are exclusive and must not be used at the same time.**
* **Conflicts between config options**
  + Enabling the **AssistNow Autonomous** feature will lead to increased power consumption while prediction is calculated. The main goal of PSM is to reduce the overall power consumption. Therefore for each application special care must be taken to judge whether
  + When enabling Power Save Mode, SBAS support can be disabled since the receiver will be unable to download any SBAS data in this mode.
  + AssistNowAutonomous is beneficial to the overall power consumption or not.
  + Jamming/Interference monitor is not supported in Power Save Mode (PSM) ON/OFF mode.
  + The AssistNow Offline and AssistNow Autonomous features are exclusive and must not be used at the same time.
* **Set the configuration of the GPS modules using U-center so that it doesn’t have to be done via the Arduino**
  + **Will require setting the configuration as ‘permanent’**
* **GPS does have a cell battery, but do not believe it has memory**
  + **Does not allow me to write data & did not save Assist Autonomous data**
  + **Not sure what the cell battery is for though**
    - **Keep the oscillator warm?**
* **The receiver does work**
  + **Proof of this from changing it to Power Save Mode and from changing other configuration options**
* **NMEA and UBX (UBLOX) are different protocols relating to how messages are sent & received**
  + **UBX has all the cool stuff for configuring the GPS and getting neat info such as Jamming**
  + **NMEA seems to handle actual GPS data**
* **Download the Assist data for the entire time the Trackers are deployed OR just use the Assist Autonomous feature**
  + **Can only download data up to 14 days in advance**
* **GPS can penetrate objects as long as they are not to thick (and the material allows it)**
* In Hot start, the receiver was **powered down only for a short time (4 hours or less)**, so that its **ephemeris is still valid**
* **So basically, all I need to do is save the ephemeris and access it during Startup**
  + **Except ephemeris is not constantly broadcasted by the Satellites.**

**Key points of Discussion:**

* Go through Markdown notes
  + Some parts require additional info and fact checking. Got distracted with the UBX stuff
* Show GPS in action
  + Discuss soldering on wire to send data
  + Show U-center & the Guide PDF
* Discuss overall thoughts
  + NEO-6 chip with the Breakout Board would be easier to use because it has an on-board EEPROM. This has a few implications
    - U-Center can be used to set up the configuration and save it to the EEPROM. Otherwise, Arduino code would have to be made up do basically what U-Center already does
    - Ephemeris data can easily be saved and access from an EEPROM dedicated to the GPS Module
* Will need to design an algorithm that decides what to do based on time underwater
  + Short time, ephemeris data is still good, get GPS quick
  + Medium time, AssistNowAutonomous extrapolates ephemeris, still get GPS fairly quick
  + Long-time, need to get ephemeris data through satellite fix. Will need to utilize Almanac data (if good) and power saving features
    - Keep in mind that it takes time to update ephemeris data

Color Legend:

Yellow – general info

Green – reference to Specific commands

Blue – Accuracy related

Purple – Power related

Pink – Very Important

Red – Configuration Stuff (Default settings, memory stuff)