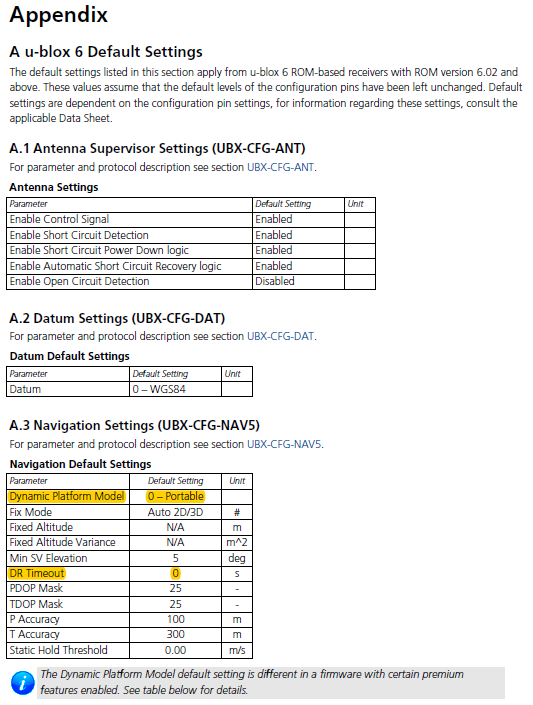
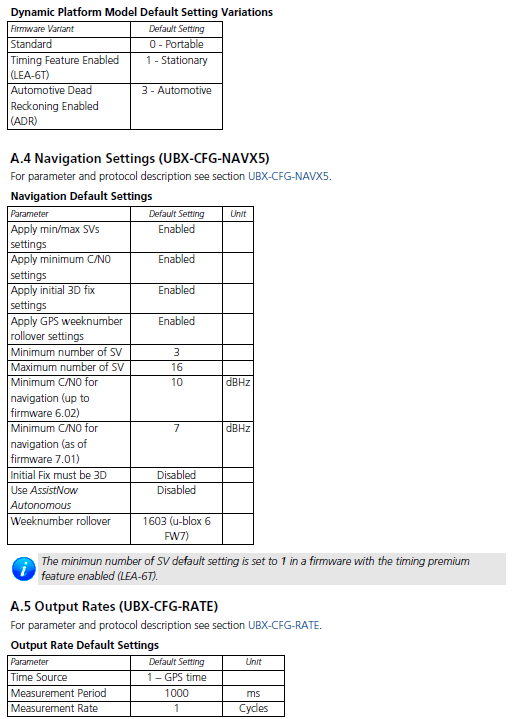
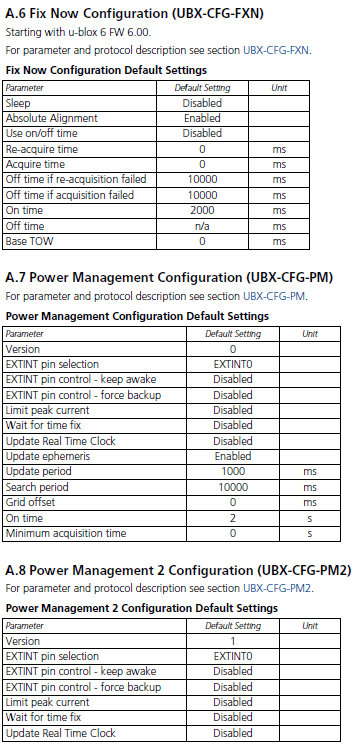
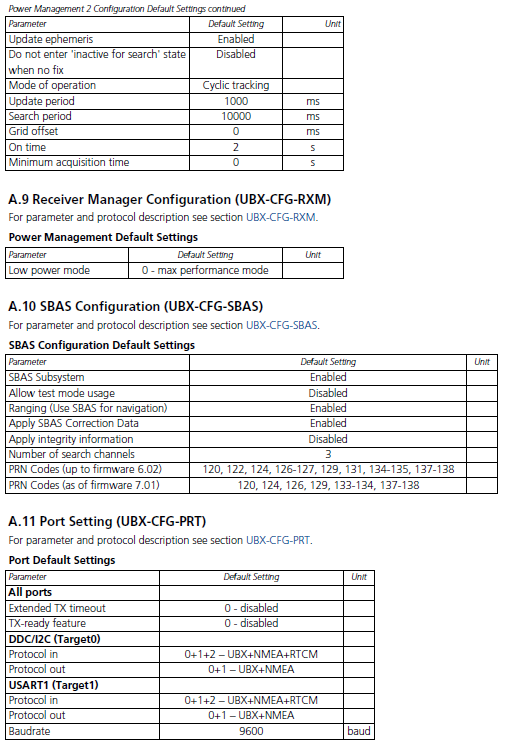
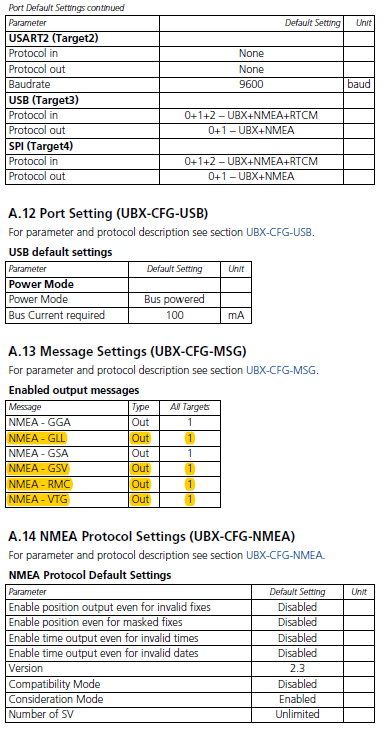
The following is from Appendix A of the u-Blox 6 Receiver Description including Protocol Specification. Default configuration parameters highlighted in yellow should be changed for the hike logging application. Changes are explained at the end of this document.

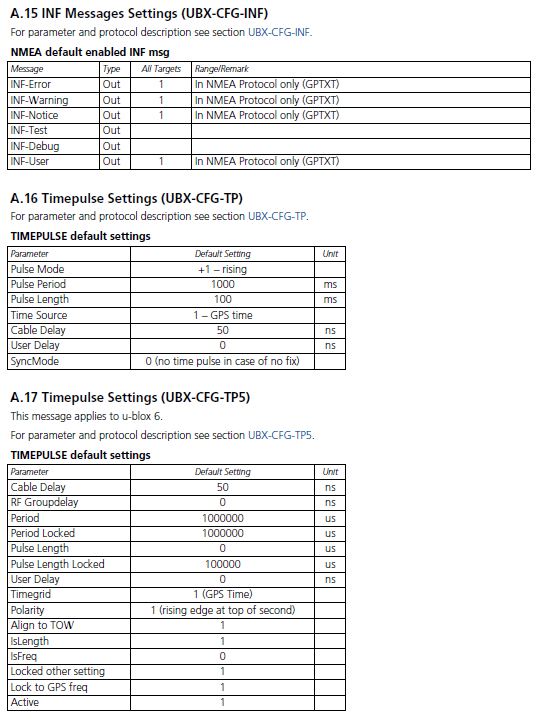


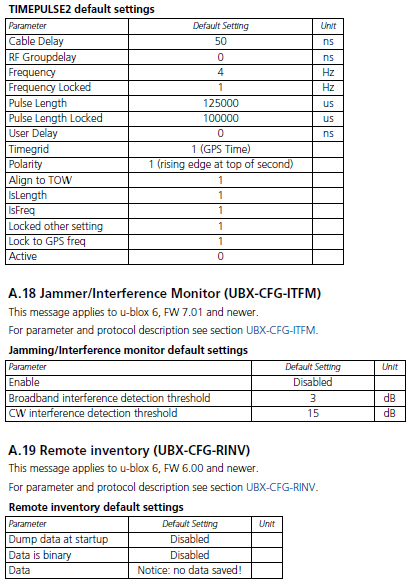












|  |  |  |  |
| --- | --- | --- | --- |
| **Default Navigation Settings (UBX-CFG-NAV5)** | | | |
| For parameter and protocol description see section UBX-CFG-NAV5. | | | |
|  | | | |
| **Parameter** | **Default Setting** | **Desired Setting** | **Unit** |
| Dynamic Platform Model | 0 – Portable | 3 -- Pedestrian |  |
| Fix Mode | Auto 2D/3D | Auto 2D/3D | # |
| Fixed Altitude | N/A | N/A | m |
| Fixed Altitude Variance | N/A | N/A | m^2 |
| Min SV Elevation | 5 | 5 | deg |
| DR Timeout | 0 | 30 | s |
| PDOP Mask | 25 | 25 | - |
| TDOP Mask | 25 | 25 | - |
| P Accuracy | 100 | 100 | m |
| T Accuracy | 300 | 300 | m |
| Static Hold Threshold | 0 | 0 | m/s |

Code to change NAV5 settings using NeoGPS library by SlashDevin:

Brute force method:

ublox::cfg\_nav5\_t setNav;

setNav.apply\_word = 0x0009; // only change dynModel and drLimit

setNav.dyn\_model = ublox::UBX\_DYN\_MODEL\_PEDESTRIAN;

setNav.fix\_mode = ublox::UBX\_POS\_FIX\_AUTO;

setNav.fixed\_alt = 0; // m

setNav.fixed\_alt\_variance = 10000; // 10000 \* m^2

setNav.min\_elev = 5; // degrees

setNav.dr\_limit = 20; // s

setNav.pos\_dop\_mask = 250; // 10 \* DOP

setNav.time\_dop\_mask = 250; // 10 \* DOP

setNav.pos\_acc\_mask = 100; // m

setNav.time\_acc\_mask = 300; // s (although spec says "m")

setNav.static\_hold\_thr = 0; // cm/s

setNav.dgps\_timeout = 0; // s

More readable and precise method:

// THIS COMMAND SETS THE NAV5 MODES using the more readable and precise method

ublox::cfg\_nav5\_t navCfg;

// set the mask bit indicating we're changing the dynamic model...

navCfg.apply.dyn\_model = true

// and change the model to Pedestrian, for hikers

// note that UBX\_DYN\_MODEL\_AIR\_4G was tested on an airline flight (PHX-BWI) and works

navCfg.dyn\_model = ublox::UBX\_DYN\_MODEL\_PEDESTRIAN;

// set the mask bit indicating we're changing the ded(uced) reckoning limit...

// and change the limit to 20 s, for hikers whose signals are sometimes blocked

navCfg.apply.dr\_limit = true;

navCfg.dr\_limit = 20;

while (!gps.send( navCfg )) {

DEBUG\_PORT.print( '.' );

delay( 500 );

}

|  |  |  |  |
| --- | --- | --- | --- |
| **Message Settings (UBX-CFG-MSG)** | | | |
| For parameter and protocol description see section UBX-CFG-MSG. | | | |
|  | | | |
| **Message** | **Default Targets & Rate** | **Desired Targets & Rate** |  |
| NMEA - GGA | All, 1 | UART 1, 1 |  |
| NMEA - GLL | All, 1 | None |  |
| NMEA - GSA | All, 1 | UART 1, 1 |  |
| NMEA - GSV | All, 1 | None |  |
| NMEA - RMC | All, 1 | None |  |
| NMEA - VTG | All, 1 | None |  |

Code to change message settings using NeoGPS library by SlashDevin:

NMEAGPS::nmea\_msg\_t msgType[6] =

{NMEAGPS::NMEA\_GGA, NMEAGPS::NMEA\_GLL, NMEAGPS::NMEA\_GSA, NMEAGPS::NMEA\_GSV, NMEAGPS::NMEA\_RMC, NMEAGPS::NMEA\_VTG};

uint8\_t rate[6] = {1, 0, 1, 0, 0, 0};

int maxI = (sizeof(msgType)/sizeof(NMEAGPS::nmea\_msg\_t));

for (int i = 0; i < maxI; i++) {

Serial.print(F("Setting message type "));

Serial.print(msgType[i]);

Serial.print(F(" to rate "));

Serial.println(rate[i]);

while (!ublox::configNMEA(gps, msgType[i], rate[i])) {

Serial.print( '.' );

delay( 500 );

}

Serial.println();

}

***Also, in NMEAGPS\_cfg.h of SlashDevin’s NeoGPS library***, make sure NMEAGPS\_PARSE\_SATELLITES, NMEAGPS\_PARSE\_GGA, NMEAGPS\_PARSE\_GSA, NMEAGPS\_COMMA\_NEEDED, and LAST\_SENTENCE\_IN\_INTERVAL NMEAGPS::NMEA\_GSA are #defined, consistent with the message types enabled by the code above.

***In GPSfix\_cfg.h of SlashDevin’s NeoGPS library***, comment/uncomment the #defines as shown below:

//#define GPS\_FIX\_DATE

#define GPS\_FIX\_TIME

#define GPS\_FIX\_LOCATION

//#define GPS\_FIX\_LOCATION\_DMS

#define GPS\_FIX\_ALTITUDE

//define GPS\_FIX\_SPEED

//#define GPS\_FIX\_VELNED

//define GPS\_FIX\_HEADING

#define GPS\_FIX\_SATELLITES

#define GPS\_FIX\_HDOP

#define GPS\_FIX\_VDOP

//#define GPS\_FIX\_PDOP

//#define GPS\_FIX\_LAT\_ERR

//#define GPS\_FIX\_LON\_ERR

//#define GPS\_FIX\_ALT\_ERR

//#define GPS\_FIX\_GEOID\_HEIGHT