# Arduino Naza Decoder Driver

Generated by Doxygen 1.8.9.1

Mon Dec 28 2015 21:00:47

## **Contents**

1	Class Index						
	1.1	Class List		1			
2	File Index						
	2.1	File List		1			
3	Class Documentation						
	3.1	NazaDecoder Class Reference		2			
		3.1.1 Detailed Description		3			
		3.1.2 Member Enumeration Documentation		3			
		3.1.3 Constructor & Destructor Documentation		3			
		3.1.4 Member Function Documentation		3			
		3.1.5 Member Data Documentation		5			
4	File Documentation						
	4.1	NazaDecoder.cpp File Reference		7			
	4.2	NazaDecoder.cpp		7			
	4.3	NazaDecoder.h File Reference		10			
		4.3.1 Macro Definition Documentation		10			
	4.4	NazaDecoder.h		11			
Ind	lex			13			
1	Cla	ass Index					
1.1	CI	ass List					
не	re are	the classes, structs, unions and interfaces with brief descriptions:					
	Naza	Decoder		2			
2	File	e Index					
_							
2.1	Fil	e List					
He	re is	a list of all files with brief descriptions:					
	Naza	Decoder.cpp		7			
	Naza	Decoder.h		10			
3	Cla	ass Documentation					

## 3.1 NazaDecoder Class Reference

```
#include <NazaDecoder.h>
```

## **Public Types**

• enum fixType { NO\_FIX = 0, FIX\_2D = 2, FIX\_3D = 3, FIX\_DGPS = 4 }

## **Public Member Functions**

- NazaDecoder ()
- uint8 t decode (int16 t input)
- int32\_t getLat ()
- int32\_t getLon ()
- int32\_t getGpsAlt ()
- int32\_t getSpeed ()
- fixType getFixType ()
- uint8 t getNumSat ()
- int32\_t getHeading ()
- int32\_t getCog ()
- int32\_t getGpsVsi ()
- int32\_t getHdop ()
- int32\_t getVdop ()
- uint8\_t getYear ()
- uint8\_t getMonth ()
- uint8\_t getDay ()
- uint8\_t getHour ()
- uint8\_t getMinute ()
- uint8\_t getSecond ()

#### **Private Member Functions**

- void pack (uint8\_t i, uint8\_t mask, uint8\_t \*result, uint8\_t len)
- void updateChecksum (int16\_t input)

#### **Private Attributes**

- int16\_t payload [58]
- int16 t seq
- int16\_t cnt
- int16\_t msgld
- int16\_t msgLen
- uint8\_t cs1
- uint8\_t cs2
- int16\_t magXMin
- int16\_t magXMax
- int16\_t magYMin
- int16\_t magYMax
- int32\_t lon
- int32\_t lat
- int32\_t gpsAlt
- int32\_t spd
- fixType fix

```
• uint8_t sat
     • int32_t heading

    int32_t cog

     • int32_t gpsVsi
     • int32_t hdop
     • int32_t vdop
     uint8_t year
     • uint8_t month
     uint8_t day
     • uint8_t hour
     • uint8_t minute

    uint8_t second

3.1.1 Detailed Description
Definition at line 16 of file NazaDecoder.h.
3.1.2 Member Enumeration Documentation
3.1.2.1 enum NazaDecoder::fixType
Enumerator
     NO_FIX
     FIX_2D
     FIX_3D
     FIX_DGPS
Definition at line 19 of file NazaDecoder.h.
3.1.3 Constructor & Destructor Documentation
3.1.3.1 NazaDecoder::NazaDecoder ( )
Definition at line 4 of file NazaDecoder.cpp.
3.1.4 Member Function Documentation
3.1.4.1 uint8_t NazaDecoder::decode ( int16_t input )
Definition at line 71 of file NazaDecoder.cpp.
3.1.4.2 int32_t NazaDecoder::getCog()
 Definition at line 40 of file NazaDecoder.cpp.
3.1.4.3 uint8_t NazaDecoder::getDay ( )
Definition at line 58 of file NazaDecoder.cpp.
3.1.4.4 NazaDecoder::fixType NazaDecoder::getFixType ( )
 Definition at line 31 of file NazaDecoder.cpp.
```

```
3.1.4.5 int32_t NazaDecoder::getGpsAlt ( )
Definition at line 25 of file NazaDecoder.cpp.
3.1.4.6 int32_t NazaDecoder::getGpsVsi()
Definition at line 43 of file NazaDecoder.cpp.
3.1.4.7 int32_t NazaDecoder::getHdop()
Definition at line 46 of file NazaDecoder.cpp.
3.1.4.8 int32_t NazaDecoder::getHeading ( )
Definition at line 37 of file NazaDecoder.cpp.
3.1.4.9 uint8_t NazaDecoder::getHour()
Definition at line 61 of file NazaDecoder.cpp.
3.1.4.10 int32_t NazaDecoder::getLat ( )
Definition at line 19 of file NazaDecoder.cpp.
3.1.4.11 int32_t NazaDecoder::getLon()
Definition at line 22 of file NazaDecoder.cpp.
3.1.4.12 uint8_t NazaDecoder::getMinute ( )
Definition at line 64 of file NazaDecoder.cpp.
3.1.4.13 uint8_t NazaDecoder::getMonth()
Definition at line 55 of file NazaDecoder.cpp.
3.1.4.14 uint8_t NazaDecoder::getNumSat ( )
Definition at line 34 of file NazaDecoder.cpp.
3.1.4.15 uint8_t NazaDecoder::getSecond ( )
Definition at line 67 of file NazaDecoder.cpp.
3.1.4.16 int32_t NazaDecoder::getSpeed ( )
Definition at line 28 of file NazaDecoder.cpp.
3.1.4.17 int32_t NazaDecoder::getVdop()
Definition at line 49 of file NazaDecoder.cpp.
3.1.4.18 uint8_t NazaDecoder::getYear ( )
Definition at line 52 of file NazaDecoder.cpp.
3.1.4.19 void NazaDecoder::pack ( uint8_t i, uint8_t mask, uint8_t * result, uint8_t len ) [private]
Definition at line 8 of file NazaDecoder.cpp.
```

```
3.1.4.20 void NazaDecoder::updateChecksum ( int16_t input ) [private]
Definition at line 14 of file NazaDecoder.cpp.
3.1.5 Member Data Documentation
3.1.5.1 int16_t NazaDecoder::cnt [private]
Definition at line 52 of file NazaDecoder.h.
3.1.5.2 int32_t NazaDecoder::cog [private]
Definition at line 88 of file NazaDecoder.h.
3.1.5.3 uint8_t NazaDecoder::cs1 [private]
Definition at line 57 of file NazaDecoder.h.
3.1.5.4 uint8_t NazaDecoder::cs2 [private]
Definition at line 60 of file NazaDecoder.h.
3.1.5.5 uint8_t NazaDecoder::day [private]
Definition at line 100 of file NazaDecoder.h.
3.1.5.6 fixType NazaDecoder::fix [private]
Definition at line 79 of file NazaDecoder.h.
3.1.5.7 int32_t NazaDecoder::gpsAlt [private]
Definition at line 73 of file NazaDecoder.h.
3.1.5.8 int32_t NazaDecoder::gpsVsi [private]
Definition at line 91 of file NazaDecoder.h.
3.1.5.9 int32_t NazaDecoder::hdop [private]
Definition at line 94 of file NazaDecoder.h.
3.1.5.10 int32_t NazaDecoder::heading [private]
Definition at line 85 of file NazaDecoder.h.
3.1.5.11 uint8_t NazaDecoder::hour [private]
Definition at line 101 of file NazaDecoder.h.
3.1.5.12 int32_t NazaDecoder::lat [private]
Definition at line 70 of file NazaDecoder.h.
3.1.5.13 int32_t NazaDecoder::lon [private]
Definition at line 67 of file NazaDecoder.h.
3.1.5.14 int16_t NazaDecoder::magXMax [private]
Definition at line 62 of file NazaDecoder.h.
```

```
3.1.5.15 int16_t NazaDecoder::magXMin [private]
Definition at line 61 of file NazaDecoder.h.
3.1.5.16 int16_t NazaDecoder::magYMax [private]
Definition at line 64 of file NazaDecoder.h.
3.1.5.17 int16_t NazaDecoder::magYMin [private]
Definition at line 63 of file NazaDecoder.h.
3.1.5.18 uint8_t NazaDecoder::minute [private]
Definition at line 102 of file NazaDecoder.h.
3.1.5.19 uint8_t NazaDecoder::month [private]
Definition at line 99 of file NazaDecoder.h.
3.1.5.20 int16_t NazaDecoder::msgld [private]
Definition at line 53 of file NazaDecoder.h.
3.1.5.21 int16_t NazaDecoder::msgLen [private]
Definition at line 54 of file NazaDecoder.h.
3.1.5.22 int16_t NazaDecoder::payload[58] [private]
Definition at line 50 of file NazaDecoder.h.
3.1.5.23 uint8_t NazaDecoder::sat [private]
Definition at line 82 of file NazaDecoder.h.
3.1.5.24 uint8_t NazaDecoder::second [private]
Definition at line 103 of file NazaDecoder.h.
3.1.5.25 int16_t NazaDecoder::seq [private]
Definition at line 51 of file NazaDecoder.h.
3.1.5.26 int32_t NazaDecoder::spd [private]
Definition at line 76 of file NazaDecoder.h.
3.1.5.27 int32_t NazaDecoder::vdop [private]
Definition at line 97 of file NazaDecoder.h.
3.1.5.28 uint8_t NazaDecoder::year [private]
Definition at line 98 of file NazaDecoder.h.
The documentation for this class was generated from the following files:
```

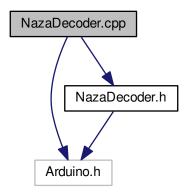
- · NazaDecoder.h
- NazaDecoder.cpp

4 File Documentation 7

## 4 File Documentation

## 4.1 NazaDecoder.cpp File Reference

```
#include <Arduino.h>
#include "NazaDecoder.h"
Include dependency graph for NazaDecoder.cpp:
```



## 4.2 NazaDecoder.cpp

```
00001 #include <Arduino.h>
00002 #include "NazaDecoder.h'
 00004 NazaDecoder::NazaDecoder()
00005
                                                   : \  \, \mathsf{seq}(0), \  \, \mathsf{cnt}(0), \  \, \mathsf{msgId}(0), \  \, \mathsf{msgLen}(0), \  \, \mathsf{cs1}(0), \  \, \mathsf{cs2}(0), \  \, \mathsf{magXMin}(0), \  \, \mathsf{magXMax}(0), \  \, \mathsf{magYMin}(0), \
0), lon(0), lat(0), gpsAlt(0), spd(0), fix(NO_FIX), sat(0), heading(0), cog(0), gpsVsi(0), hdop(0), vdop(0), year(0), month(0), day(0), hour(0), minute(0), second(0) {
00006 }
 00007
 00008 void NazaDecoder::pack(uint8_t i, uint8_t mask, uint8_t *result, uint8_t len) {
                            for (int16_t j = 0; j < len; j++) {
    *(result++) = payload[i + j] ^ mask;</pre>
 00009
 00010
 00011
 00012 }
 00013
 00014 void NazaDecoder::updateChecksum(int16_t input) {
 00015
                            cs1 += input;
00016
                                  cs2 += cs1;
 00017 }
00018
 00019 int32_t NazaDecoder::getLat() {
                             return lat;
 00021 }
 00022 int32_t NazaDecoder::getLon() {
                            return lon;
00023
00024 }
 00025 int32_t NazaDecoder::getGpsAlt() {
                              return gpsAlt;
 00027 }
 00028 int32_t NazaDecoder::getSpeed() {
 00029
                                  return spd;
00030 }
 00031 NazaDecoder::fixType NazaDecoder::getFixType() {
                                  return fix;
 00033 }
 00034 uint8_t NazaDecoder::getNumSat() {
00035
                              return sat;
 00036 }
 00037 int32_t NazaDecoder::getHeading() {
00038
                                  return heading;
00039 }
```

```
00040 int32_t NazaDecoder::getCog() {
00041
          return cog;
00042 }
00043 int32_t NazaDecoder::getGpsVsi() {
00044
         return gpsVsi;
00045 }
00046 int32_t NazaDecoder::getHdop() {
00047
        return hdop;
00048 }
00049 int32_t NazaDecoder::getVdop() {
        return vdop;
00050
00051 }
00052 uint8_t NazaDecoder::getYear() {
00053
        return year;
00054 }
00055 uint8_t NazaDecoder::getMonth() {
00056
          return month;
00057 }
00058 uint8_t NazaDecoder::getDay() {
00059
          return day;
00060 }
00061 uint8_t NazaDecoder::getHour() {
00062
          return hour;
00063 }
00064 uint8_t NazaDecoder::getMinute() {
         return minute;
00066 }
00067 uint8_t NazaDecoder::getSecond() {
00068
          return second;
00069 }
00070
00071 uint8_t NazaDecoder::decode(int16_t input) {
00072
00073
          // header (part 1 - 0x55)
00074
          if ((seq == 0) \&\& (input == 0x55)) {
00075
               seq++;
00076
00077
00078
          // header (part 2 - 0xAA)
00079
          else if ((seq == 1) && (input == 0xAA)) {
00080
              cs1 = 0;

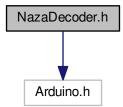
cs2 = 0;
00081
00082
          seq++;
} else if (seq == 2) {
  msgId = input;
00083
00084
00085
               updateChecksum(input);
00086
              seq++;
00087
          }
00088
00089
          // message id
          // message payload length (should match message id)
00090
00091
          // store payload in buffer
     else if ((seq == 3) && (((msgId == 0x10) && (input == 0x3A)) || ((
msgId == 0x20) && (input == 0x06)))) {
00092
00093
              msgLen = input;
00094
              cnt = 0;
00095
              updateChecksum(input);
00096
               seq++;
00097
          } else if (seq == 4) {
              payload[cnt++] = input;
00098
               updateChecksum(input);
00099
               if (cnt >= msgLen) {
00100
00101
                   seq++;
00102
00103
          }
00104
          // verify checksum #1
else if ((seq == 5) && (input == cs1)) {
00105
00106
00107
             sea++;
00108
00109
00110
          // verify checksum #2
00111
          else if ((seq == 6) && (input == cs2)) {
00112
              seq++;
          } else {
00113
00114
              seq = 0;
00115
          }
00116
          // all data in buffer
if (seq == 7) {
    seq = 0;
00117
00118
00119
00120
               // Decode GPS data
00121
               if (msgId == NAZA_MESSAGE_GPS) {
00122
                   uint8_t mask = payload[55];
00123
                   uint32_t time;
                   pack(0, mask, (uint8_t *) &time, 4);
second = time & 0x3f;
00124
00125
```

```
time >>= 6;
00127
                      minute = time & 0x3f;
00128
                      time >>= 6;
00129
                      hour = time & 0x0f;
00130
                      time >>= 4;
                      day = time & 0x1f;
00131
                      time >>= 5;
00132
00133
                      if (hour > 7) {
00134
                           day++;
00135
00136
                      month = time & 0x0f;
00137
                      time >>= 4;
00138
                      year = time & 0x7f;
00139
00140
                      uint32_t aux;
                      pack(4, mask, (uint8_t *) &aux, 4);
lon = (int32_t) aux / 10000000;
00141
00142
                      pack(8, mask, (uint8_t *) &aux, 4);
lat = (int32_t) aux / 10000000;
00143
                      pack(12, mask, (uint8_t *) &aux, 4);
gpsAlt = (int32_t) aux / 1000;
00145
00146
                      pack(28, mask, (uint8_t *) &aux, 4);
int32_t nVel = (int32_t) aux / 100;
00147
00148
                      pack(32, mask, (uint8_t *) &aux, 4);
int32_t eVel = (int32_t) aux / 100;
00149
00150
                      spd = sqrt(nVel * nVel + eVel * eVel);
00151
00152
                       cog = atan2(eVel, nVel) * 180.0 / M_PI;
00153
                      if (cog < 0) {</pre>
00154
                            cog += 360.0;
00155
                      pack(36, mask, (uint8_t *) &aux, 4);
gpsVsi = -(int32_t) aux / 100;
00156
00157
00158
                      pack(42, mask, (uint8_t *) &aux, 4);
00159
                       vdop = (int32_t) aux / 100;
                      pack(44, mask, (uint8_t *) &aux, 4);
int32_t ndop = (int32_t) aux / 100;
00160
00161
                      pack(46, mask, (uint8_t *) &aux, 4);
int32_t edop = (int32_t) aux / 100;
00162
00163
00164
                      hdop = sqrt (ndop * ndop + edop * edop);
00165
                       sat = payload[48];
                      uint8_t type = payload[50] ^ mask;
uint8_t flag = payload[52] ^ mask;
00166
00167
00168
                      switch (type) {
00169
                      case 2:
00170
                          fix = FIX_2D;
00171
                           break;
00172
                      case 3:
                           fix = FIX_3D;
00173
00174
                           break:
00175
                      default:
                           fix = NO_FIX;
00176
00177
00178
                      if ((fix != NO_FIX) && (flag & 0x02)) {
   fix = FIX_DGPS;
00179
00180
00181
                      }
                 }
00183
00184
                  // Decode compass data (not tilt compensated)
                 else if (msgId == NAZA_MESSAGE_COMPASS) {
    uint8_t mask = payload[4];
    mask = (((mask ^ (mask >> 4)) & 0x0f) | ((mask << 3) & 0xf0)) ^ (((mask & 0x01) << 3) | ((mask</pre>
00185
00186
00187
      & 0x01) << 7));
00188
                      int16_t x;
00189
                      pack(0, mask, (uint8_t *) &x, 2);
00190
                      int16_t y;
00191
                      pack(2, mask, (uint8_t *) &y, 2);
if (x > magXMax) {
00192
00193
                           magXMax = x;
00194
00195
                      if (x < magXMin) {</pre>
00196
                           magXMin = x;
00197
00198
                      if (y > magYMax) {
                           magYMax = y;
00199
00200
00201
                      if (y < magYMin) {</pre>
00202
                           magYMin = y;
00203
                      heading = -atan2(y - ((magYMax + magYMin) / 2), x - ((
00204
      magXMax + magXMin) / 2)) * 180.0 / M_PI;

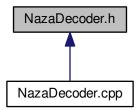
if (heading < 0) {
00205
00206
                           heading += 360.0;
00207
                      }
00208
00209
                 return msgId;
00210
            } else {
```

## 4.3 NazaDecoder.h File Reference

#include <Arduino.h>
Include dependency graph for NazaDecoder.h:



This graph shows which files directly or indirectly include this file:



## Classes

class NazaDecoder

## Macros

- #define NAZA\_MESSAGE\_NONE 0x00
- #define NAZA MESSAGE GPS 0x10
- #define NAZA\_MESSAGE\_COMPASS 0x20
- 4.3.1 Macro Definition Documentation
- 4.3.1.1 #define NAZA\_MESSAGE\_COMPASS 0x20

Definition at line 14 of file NazaDecoder.h.

4.4 NazaDecoder.h 11

## 4.3.1.2 #define NAZA\_MESSAGE\_GPS 0x10

Definition at line 13 of file NazaDecoder.h.

#### 4.3.1.3 #define NAZA\_MESSAGE\_NONE 0x00

Arduino Naza Decoder.

Inspired in the Pawelsky's work.

Definition at line 12 of file NazaDecoder.h.

#### 4.4 NazaDecoder.h

```
00001
00007 #ifndef __ARDUINO_NAZA_DECODER_H_
00008 #define __ARDUINO_NAZA_DECODER_H_
00010 #include <Arduino.h>
00011
00012 #define NAZA_MESSAGE_NONE
                                    0 \times 00
00013 #define NAZA_MESSAGE_GPS
                                    0x10
00014 #define NAZA_MESSAGE_COMPASS 0x20
00015
00016 class NazaDecoder {
00017
00018 public:
        typedef enum {
00019
            NO_FIX = 0,
00020
              FIX_2D = 2,
00021
00022
              FIX_3D = 3,
00023
              FIX_DGPS = 4
00024
         } fixType;
00025
00026
         NazaDecoder();
00027
00028
          uint8_t decode(int16_t input);
00029
          int32_t getLat();
00030
          int32_t getLon();
00031
          int32_t getGpsAlt();
00032
          int32_t getSpeed();
00033
          fixType getFixType();
          uint8_t getNumSat();
00034
00035
          int32_t getHeading();
00036
          int32_t getCog();
00037
          int32_t getGpsVsi();
00038
          int32_t getHdop();
int32_t getVdop();
00039
          uint8_t getYear();
00040
00041
          uint8_t getMonth();
00042
          uint8_t getDay();
00043
00044
          // Note that for time between 16:00 and 23:59 the hour returned from GPS module is actually 00:00 -
       7:59.
00045
         uint8_t getHour();
00046
          uint8_t getMinute();
00047
          uint8_t getSecond();
00048
00049 private:
        int16_t payload[58];
00050
          int16_t seq;
00051
00052
          int16_t cnt;
00053
          int16_t msgId;
00054
          int16_t msgLen;
00055
          // checksum #1
00056
          uint8_t cs1;
00057
00058
00059
          // checksum #2
00060
          uint8_t cs2;
          int16_t magXMin;
00061
00062
          int16_t magXMax;
00063
          int16 t magYMin;
00064
          int16_t magYMax;
00065
00066
           // longitude in degree decimal
00067
          int32_t lon;
00068
00069
          // latitude in degree decimal
00070
          int32_t lat;
00071
```

```
00072
           // altitude in m (from GPS)
00073
00074
           int32_t gpsAlt;
           // speed in m/s
int32_t spd;
00075
00076
00077
           // fix type
fixType fix;
00078
00079
08000
           // number of satellites
00081
           uint8_t sat;
00082
00083
00084
           // heading (not tilt compensated) in degrees
00085
           int32_t heading;
00086
00087
           // \ {\tt course \ over \ ground}
00088
           int32_t cog;
00089
00090
           // vertical speed indicator (from GPS) in m/s (a.k.a. climb speed)
00091
           int32_t gpsVsi;
00092
           \//\ horizontal dilution of precision
00093
00094
           int32_t hdop;
00095
00096
           // vertical dilution of precision
00097
           int32_t vdop;
00098
           uint8_t year;
00099
           uint8_t month;
00100
           uint8_t day;
           uint8_t hour;
uint8_t minute;
uint8_t second;
00101
00102
00103
00104
00105
           void pack(uint8_t i, uint8_t mask, uint8_t *result, uint8_t len);
00106
00107 };
           void updateChecksum(int16_t input);
00108
00109 #endif /* __ARDUINO_NAZA_DECODER_H_ */
```

# Index

and .	Nama Danadari 4
Cnt Naza Dagadar E	NazaDecoder, 4
NazaDecoder, 5	getYear
COG Noza Docador E	NazaDecoder, 4
NazaDecoder, 5	gpsAlt
CS1	NazaDecoder, 5
NazaDecoder, 5	gpsVsi
CS2	NazaDecoder, 5
NazaDecoder, 5	hdop
day	NazaDecoder, 5
NazaDecoder, 5	heading
decode	NazaDecoder, 5
NazaDecoder, 3	hour
Nazabeoodoi, o	NazaDecoder, 5
FIX 2D	Nazabecoder, o
NazaDecoder, 3	lat
FIX 3D	NazaDecoder, 5
NazaDecoder, 3	lon
FIX DGPS	NazaDecoder, 5
NazaDecoder, 3	ruzuboodor, o
fix	magXMax
NazaDecoder, 5	NazaDecoder, 5
fixType	magXMin
NazaDecoder, 3	NazaDecoder, 5
Nazabecodei, 3	magYMax
getCog	NazaDecoder, 6
NazaDecoder, 3	magYMin
getDay	
NazaDecoder, 3	NazaDecoder, 6
getFixType	minute
	NazaDecoder, 6
NazaDecoder, 3	month
getGpsAlt NazaDecoder, 3	NazaDecoder, 6
•	msgld
getGpsVsi	NazaDecoder, 6
NazaDecoder, 4	msgLen
getHdop	NazaDecoder, 6
NazaDecoder, 4	NAZA MECCACE COMPACC
getHeading	NAZA_MESSAGE_COMPASS
NazaDecoder, 4	NazaDecoder.h, 10
getHour	NAZA_MESSAGE_GPS
NazaDecoder, 4	NazaDecoder.h, 10
getLat	NAZA_MESSAGE_NONE
NazaDecoder, 4	NazaDecoder.h, 11
getLon	NO_FIX
NazaDecoder, 4	NazaDecoder, 3
getMinute	NazaDecoder, 2
NazaDecoder, 4	cnt, 5
getMonth	cog, 5
NazaDecoder, 4	cs1, 5
getNumSat	cs2, 5
NazaDecoder, 4	day, 5
getSecond	decode, 3
NazaDecoder, 4	FIX_2D, 3
getSpeed	FIX_3D, 3
NazaDecoder, 4	FIX DGPS, 3
	FIX_DGF3, 3
getVdop	fix, 5

14 INDEX

	fixType, 3	seq	
	getCog, 3	•	NazaDecoder, 6
	getDay, 3	spd	ŕ
	getFixType, 3	•	NazaDecoder, 6
	getGpsAlt, 3		ŕ
	getGpsVsi, 4	upda	ateChecksum
	getHdop, 4	-	NazaDecoder, 4
	getHeading, 4		
	getHour, 4	vdop	)
	getLat, 4		NazaDecoder, 6
	getLon, 4		
	getMinute, 4	year	
	getMonth, 4		NazaDecoder, 6
	getNumSat, 4		
	getSecond, 4		
	getSpeed, 4		
	getVdop, 4		
	getYear, 4		
	gpsAlt, 5		
	gpsVsi, 5		
	hdop, 5		
	heading, 5		
	hour, 5		
	lat, 5		
	lon, 5		
	magXMax, 5		
	magXMin, 5		
	magYMax, 6		
	magYMin, 6		
	minute, 6		
	month, 6		
	msgld, 6		
	msgLen, 6		
	NO_FIX, 3		
	NazaDecoder, 3		
	pack, 4		
	payload, 6		
	sat, 6		
	second, 6		
	seq, 6		
	spd, 6		
	updateChecksum, 4		
	vdop, 6		
	year, 6		
	aDecoder.cpp, 7		
Naza	aDecoder.h, 10		
	NAZA_MESSAGE_COMPASS, 10		
	NAZA_MESSAGE_GPS, 10		
	NAZA_MESSAGE_NONE, 11		
pack			
اد بحرم	NazaDecoder, 4		
paylo			
	NazaDecoder, 6		
sat			
Jui	NazaDecoder, 6		
seco			
5550	NazaDecoder, 6		