

## Arduino Naza Decoder Driver

Generated by Doxygen 1.8.9.1

Mon Dec 28 2015 21:00:47

## Contents

<b>1</b>	<b><a href="#">Class Index</a></b>	<b>1</b>
1.1	<a href="#">Class List</a>	1
<b>2</b>	<b><a href="#">File Index</a></b>	<b>1</b>
2.1	<a href="#">File List</a>	1
<b>3</b>	<b><a href="#">Class Documentation</a></b>	<b>1</b>
3.1	<a href="#">NazaDecoder Class Reference</a>	2
3.1.1	<a href="#">Detailed Description</a>	3
3.1.2	<a href="#">Member Enumeration Documentation</a>	3
3.1.3	<a href="#">Constructor &amp; Destructor Documentation</a>	3
3.1.4	<a href="#">Member Function Documentation</a>	3
3.1.5	<a href="#">Member Data Documentation</a>	5
<b>4</b>	<b><a href="#">File Documentation</a></b>	<b>7</b>
4.1	<a href="#">NazaDecoder.cpp File Reference</a>	7
4.2	<a href="#">NazaDecoder.cpp</a>	7
4.3	<a href="#">NazaDecoder.h File Reference</a>	10
4.3.1	<a href="#">Macro Definition Documentation</a>	10
4.4	<a href="#">NazaDecoder.h</a>	11
	<b><a href="#">Index</a></b>	<b>13</b>

## 1 Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">NazaDecoder</a>	<b>2</b>
-----------------------------	----------

## 2 File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">NazaDecoder.cpp</a>	<b>7</b>
<a href="#">NazaDecoder.h</a>	<b>10</b>

## 3 Class 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 msgId`
- `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::msgId` [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](#)

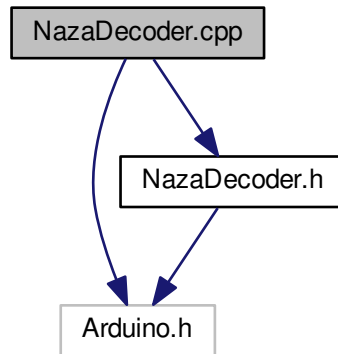
## 4 File Documentation

### 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"
00003
00004 NazaDecoder::NazaDecoder()
00005     : seq(0), cnt(0), msgId(0), msgLen(0), cs1(0), cs2(0), magXMin(0), magXMax(0), magYMin(0), magYMax(
00006         0), lon(0), lat(0), gpsAlt(0), spd(0), fix(NO_FIX), sat(0), heading(0), cog(0), gpsVsi(0), hdop(0), vdop(0),
00007         year(0), month(0), day(0), hour(0), minute(0), second(0) {}
00008
00009 void NazaDecoder::pack(uint8_t i, uint8_t mask, uint8_t *result, uint8_t len) {
00010     for (int16_t j = 0; j < len; j++) {
00011         *(result++) = payload[i + j] ^ mask;
00012     }
00013 }
00014
00015 void NazaDecoder::updateChecksum(int16_t input) {
00016     cs1 += input;
00017     cs2 += cs1;
00018 }
00019
00020 int32_t NazaDecoder::getLat() {
00021     return lat;
00022 }
00023
00024 int32_t NazaDecoder::getLon() {
00025     return lon;
00026 }
00027
00028 int32_t NazaDecoder::getGpsAlt() {
00029     return gpsAlt;
00030 }
00031
00032 int32_t NazaDecoder::getSpeed() {
00033     return spd;
00034 }
00035
00036 fixType NazaDecoder::getFixType() {
00037     return fix;
00038 }
00039
00040 uint8_t NazaDecoder::getNumSat() {
00041     return sat;
00042 }
00043
00044 int32_t NazaDecoder::getHeading() {
00045     return heading;
00046 }
  
```



```

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() {
00050     return vdop;
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() {
00065     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;
00081         cs2 = 0;
00082         seq++;
00083     } else if (seq == 2) {
00084         msgId = input;
00085         updateChecksum(input);
00086         seq++;
00087     }
00088
00089     // message id
00090     // message payload length (should match message id)
00091     // store payload in buffer
00092     else if ((seq == 3) && ((msgId == 0x10) && (input == 0x3A)) || ((
msgId == 0x20) && (input == 0x06))) {
00093         msgLen = input;
00094         cnt = 0;
00095         updateChecksum(input);
00096         seq++;
00097     } else if (seq == 4) {
00098         payload[cnt++] = input;
00099         updateChecksum(input);
00100         if (cnt >= msgLen) {
00101             seq++;
00102         }
00103     }
00104
00105     // verify checksum #1
00106     else if ((seq == 5) && (input == cs1)) {
00107         seq++;
00108     }
00109
00110     // verify checksum #2
00111     else if ((seq == 6) && (input == cs2)) {
00112         seq++;
00113     } else {
00114         seq = 0;
00115     }
00116
00117     // all data in buffer
00118     if (seq == 7) {
00119         seq = 0;
00120         // Decode GPS data
00121         if (msgId == NAZA_MESSAGE_GPS) {
00122             uint8_t mask = payload[55];
00123             uint32_t time;
00124             pack(0, mask, (uint8_t *) &time, 4);
00125             second = time & 0x3f;

```

```

00126         time >>= 6;
00127         minute = time & 0x3f;
00128         time >>= 6;
00129         hour = time & 0x0f;
00130         time >>= 4;
00131         day = time & 0x1f;
00132         time >>= 5;
00133         if (hour > 7) {
00134             day++;
00135         }
00136         month = time & 0x0f;
00137         time >>= 4;
00138         year = time & 0x7f;
00139
00140         uint32_t aux;
00141         pack(4, mask, (uint8_t *) &aux, 4);
00142         lon = (int32_t) aux / 10000000;
00143         pack(8, mask, (uint8_t *) &aux, 4);
00144         lat = (int32_t) aux / 10000000;
00145         pack(12, mask, (uint8_t *) &aux, 4);
00146         gpsAlt = (int32_t) aux / 1000;
00147         pack(28, mask, (uint8_t *) &aux, 4);
00148         int32_t nVel = (int32_t) aux / 100;
00149         pack(32, mask, (uint8_t *) &aux, 4);
00150         int32_t eVel = (int32_t) aux / 100;
00151         spd = sqrt(nVel * nVel + eVel * eVel);
00152         cog = atan2(eVel, nVel) * 180.0 / M_PI;
00153         if (cog < 0) {
00154             cog += 360.0;
00155         }
00156         pack(36, mask, (uint8_t *) &aux, 4);
00157         gpsVsi = -(int32_t) aux / 100;
00158         pack(42, mask, (uint8_t *) &aux, 4);
00159         vdop = (int32_t) aux / 100;
00160         pack(44, mask, (uint8_t *) &aux, 4);
00161         int32_t ndop = (int32_t) aux / 100;
00162         pack(46, mask, (uint8_t *) &aux, 4);
00163         int32_t edop = (int32_t) aux / 100;
00164         hdop = sqrt(ndop * ndop + edop * edop);
00165         sat = payload[48];
00166         uint8_t type = payload[50] ^ mask;
00167         uint8_t flag = payload[52] ^ mask;
00168         switch (type) {
00169             case 2:
00170                 fix = FIX_2D;
00171                 break;
00172             case 3:
00173                 fix = FIX_3D;
00174                 break;
00175             default:
00176                 fix = NO_FIX;
00177                 break;
00178         }
00179         if ((fix != NO_FIX) && (flag & 0x02)) {
00180             fix = FIX_DGPS;
00181         }
00182     }
00183
00184     // Decode compass data (not tilt compensated)
00185     else if (msgId == NAZA_MESSAGE_COMPASS) {
00186         uint8_t mask = payload[4];
00187         mask = (((mask ^ (mask >> 4)) & 0x0f) | ((mask << 3) & 0xf0)) ^ (((mask & 0x01) << 3) | ((mask
& 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);
00192         if (x > magXMax) {
00193             magXMax = x;
00194         }
00195         if (x < magXMin) {
00196             magXMin = x;
00197         }
00198         if (y > magYMax) {
00199             magYMax = y;
00200         }
00201         if (y < magYMin) {
00202             magYMin = y;
00203         }
00204         heading = -atan2(y - ((magYMax + magYMin) / 2), x - ((
magXMax + magXMin) / 2)) * 180.0 / M_PI;
00205         if (heading < 0) {
00206             heading += 360.0;
00207         }
00208     }
00209     return msgId;
00210 } else {

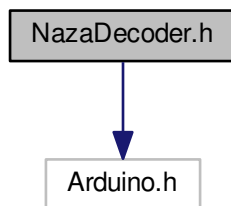
```

```
00211     return NAZA_MESSAGE_NONE;
00212   }
00213 }
```

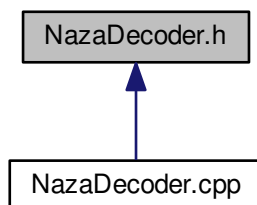
### 4.3 NazaDecoder.h File Reference

```
#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](#).

## 4.3.1.2 #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__
00009
00010 #include <Arduino.h>
00011
00012 #define NAZA_MESSAGE_NONE 0x00
00013 #define NAZA_MESSAGE_GPS 0x10
00014 #define NAZA_MESSAGE_COMPASS 0x20
00015
00016 class NazaDecoder {
00017 public:
00018     typedef enum {
00019         NO_FIX = 0,
00020         FIX_2D = 2,
00021         FIX_3D = 3,
00022         FIX_DGPS = 4
00023     } fixType;
00024
00025     NazaDecoder();
00026
00027     uint8_t decode(int16_t input);
00028     int32_t getLat();
00029     int32_t getLon();
00030     int32_t getGpsAlt();
00031     int32_t getSpeed();
00032     fixType getFixType();
00033     uint8_t getNumSat();
00034     int32_t getHeading();
00035     int32_t getCog();
00036     int32_t getGpsVsi();
00037     int32_t getHdop();
00038     int32_t getVdop();
00039     uint8_t getYear();
00040     uint8_t getMonth();
00041     uint8_t getDay();
00042
00043     // Note that for time between 16:00 and 23:59 the hour returned from GPS module is actually 00:00 -
00044     // 7:59.
00045     uint8_t getHour();
00046     uint8_t getMinute();
00047     uint8_t getSecond();
00048
00049 private:
00050     int16_t payload[58];
00051     int16_t seq;
00052     int16_t cnt;
00053     int16_t msgId;
00054     int16_t msgLen;
00055
00056     // checksum #1
00057     uint8_t cs1;
00058
00059     // checksum #2
00060     uint8_t cs2;
00061     int16_t magXMin;
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 int32_t gpsAlt;
00074
00075 // speed in m/s
00076 int32_t spd;
00077
00078 // fix type
00079 fixType fix;
00080
00081 // number of satellites
00082 uint8_t sat;
00083
00084 // heading (not tilt compensated) in degrees
00085 int32_t heading;
00086
00087 // 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
00093 // horizontal dilution of precision
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;
00101 uint8_t hour;
00102 uint8_t minute;
00103 uint8_t second;
00104
00105 void pack(uint8_t i, uint8_t mask, uint8_t *result, uint8_t len);
00106 void updateChecksum(int16_t input);
00107 };
00108
00109 #endif /* __ARDUINO_NAZA_DECODER_H__ */
```

## Index

- cnt
  - NazaDecoder, [5](#)
- cog
  - NazaDecoder, [5](#)
- cs1
  - NazaDecoder, [5](#)
- cs2
  - NazaDecoder, [5](#)
- day
  - NazaDecoder, [5](#)
- decode
  - NazaDecoder, [3](#)
- FIX\_2D
  - NazaDecoder, [3](#)
- FIX\_3D
  - NazaDecoder, [3](#)
- FIX\_DGPS
  - NazaDecoder, [3](#)
- fix
  - NazaDecoder, [5](#)
- fixType
  - NazaDecoder, [3](#)
- getCog
  - NazaDecoder, [3](#)
- getDay
  - NazaDecoder, [3](#)
- getFixType
  - NazaDecoder, [3](#)
- getGpsAlt
  - NazaDecoder, [3](#)
- getGpsVsi
  - NazaDecoder, [4](#)
- getHdop
  - NazaDecoder, [4](#)
- getHeading
  - NazaDecoder, [4](#)
- getHour
  - NazaDecoder, [4](#)
- getLat
  - NazaDecoder, [4](#)
- getLon
  - NazaDecoder, [4](#)
- getMinute
  - NazaDecoder, [4](#)
- getMonth
  - NazaDecoder, [4](#)
- getNumSat
  - NazaDecoder, [4](#)
- getSecond
  - NazaDecoder, [4](#)
- getSpeed
  - NazaDecoder, [4](#)
- getVdop
  - NazaDecoder, [4](#)
- getYear
  - NazaDecoder, [4](#)
- gpsAlt
  - NazaDecoder, [5](#)
- gpsVsi
  - NazaDecoder, [5](#)
- hdop
  - NazaDecoder, [5](#)
- heading
  - NazaDecoder, [5](#)
- hour
  - NazaDecoder, [5](#)
- lat
  - NazaDecoder, [5](#)
- lon
  - NazaDecoder, [5](#)
- magXMax
  - NazaDecoder, [5](#)
- magXMin
  - NazaDecoder, [5](#)
- magYMax
  - NazaDecoder, [6](#)
- magYMin
  - NazaDecoder, [6](#)
- minute
  - NazaDecoder, [6](#)
- month
  - NazaDecoder, [6](#)
- msgId
  - NazaDecoder, [6](#)
- msgLen
  - NazaDecoder, [6](#)
- NAZA\_MESSAGE\_COMPASS
  - NazaDecoder.h, [10](#)
- NAZA\_MESSAGE\_GPS
  - NazaDecoder.h, [10](#)
- NAZA\_MESSAGE\_NONE
  - NazaDecoder.h, [11](#)
- NO\_FIX
  - NazaDecoder, [3](#)
- NazaDecoder, [2](#)
  - cnt, [5](#)
  - cog, [5](#)
  - cs1, [5](#)
  - cs2, [5](#)
  - day, [5](#)
  - decode, [3](#)
  - FIX\_2D, [3](#)
  - FIX\_3D, [3](#)
  - FIX\_DGPS, [3](#)
  - fix, [5](#)

- fixType, [3](#)
- getCog, [3](#)
- getDay, [3](#)
- getFixType, [3](#)
- getGpsAlt, [3](#)
- getGpsVsi, [4](#)
- getHdop, [4](#)
- getHeading, [4](#)
- getHour, [4](#)
- getLat, [4](#)
- getLon, [4](#)
- getMinute, [4](#)
- getMonth, [4](#)
- 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](#)
- msgId, [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](#)
- NazaDecoder.cpp, [7](#)
- NazaDecoder.h, [10](#)
  - NAZA\_MESSAGE\_COMPASS, [10](#)
  - NAZA\_MESSAGE\_GPS, [10](#)
  - NAZA\_MESSAGE\_NONE, [11](#)
- pack
  - NazaDecoder, [4](#)
- payload
  - NazaDecoder, [6](#)
- sat
  - NazaDecoder, [6](#)
- second
  - NazaDecoder, [6](#)
- seq
  - NazaDecoder, [6](#)
- spd
  - NazaDecoder, [6](#)
- updateChecksum
  - NazaDecoder, [4](#)
- vdop
  - NazaDecoder, [6](#)
- year
  - NazaDecoder, [6](#)