Sparton Navigation Modules
Programmer’s Quick Start Guide
Version 2.2

Applicable to Sparton GEDC-6, DC-4 and AHRS-8.
Programmer's Quick Start

This quick start is intended for use by someone who wishes to obtain data from the Navigation Module (NavMod). This document provides examples of each of the protocols that are supported by the NavMod plus an example of how to perform an in-field calibration. References are given to relevant sections in the user manuals so that the user only needs to read manual sections in the area of interest.

The documents referenced in this Quick Start may be found on the NavMod CD or https://spartonnavex.com/technical-support/downloads/.

Reference the Troubleshooting section at the end of the Software Interface User’s Manual if problems are encountered while following this Quick Start.

The NavMod supports four protocols where two are binary and two are ASCII. The binary protocols are Legacy Binary and RFS (Remote Function Select) and the ASCII protocols are NMEA and NorthTek. Reference: Protocol Introduction section in the Software Interface User’s Manual.

This Quick Start provides examples of the different methods to get magnetic heading (yaw).

Prerequisites

It is assumed that the instructions in the NDS-1 Quick Start Guide have been followed to set up the NavMod hardware such that it can communicate with the NDS-1 Host Application. A terminal emulator is required to follow this Programmer’s Quick Start procedure. See the Terminal Emulator Application Note for set-up examples or use the terminal emulator that is built into the NDS-1 Host Application. This Programmer’s Quick Start will assume that the NDS-1 Host Application Terminal Emulator is being used but other terminal emulators could be used as well.

Procedure Set-up

In the NDS-1 Host Application, pull down Options->Terminal Emulator or use <ctrl-t>.

In the terminal emulator window, hit the “Enter” key and there should be an “OK” response. By default, the NavMod does not respond until a request is entered. If it is desired that the NavMod output data upon start-up, reference the User Boot Program section in the Software Interface User’s Manual.

NorthTek Protocol


NorthTek is a programming language based on ANSI Forth. With NorthTek, the user has a lot of control in how the NavMod outputs its data using interpretive command line type instructions or by creating text files that define a NorthTek program script. NorthTek is case sensitive and lines are terminated with a carriage return.

NorthTek Display Command Word “di.”


The “di.” command can be used to display any of the variables listed in the long list of available variables (reference: the Variable Summary Table and the Variable Detailed Descriptions table in the Software Interface User’s Manual).

Enter:

```
yaw di.
```

Response example:

```
yaw = 215.102310
OK
```

NorthTek Streaming Sensor Data Functions

A set of functions have been created that output comma separated (and sometimes ‘;’ separated) data suitable for input to analysis tools such as Microsoft® Excel®, MATLAB® and Mathematica®. Terminal emulators usually supply a logging capability which can be used to capture the data.

Enter:

```
1 compass.p
```

Response example (repeats):

```
C,17324883, 2.45, -0.12, 214.26
```

The letter ‘C’ identifies the data set, the next number is the timestamp (milliseconds), followed by pitch, roll and yaw respectively (degrees).

Notation such as <ctrl-s> below represents pressing the control key and the s key at the same time.

To stop the data streaming, enter:

```
<ctrl-s>
0 compass.p
<ctrl-q>
```

The <ctrl-s> causes the streaming to pause so you can enter the next command. The <ctrl-s> is not necessary but it makes it easier to see what is being typed. The <ctrl-q> unpauses which allows streaming due to future commands.
NorthTek Custom Combination Print Streaming

Applies to GEDC-6 2.3.2 or later, DC-4 2.2.0 or later and any AHRS-8


This capability allows the user to select a custom combination of sensor data. In this example, suppose we want to have time, temperature and yaw and we want the data displayed every 10th time the yaw is computed.

First, we select what we want to display (see printmask below). NorthTek is based on Forth, so each item gets pushed on the stack and functions such as “or” operates on the top 2 items on the stack and puts the result on the stack. The “set” puts the result of the 2 “or”s into the variable “printmask”. The “drop” is needed to clear the status result of the “set” off of the stack. Reference: NPL Basics section in the NorthTek System Programming Manual

Enter:

Inhibit print streaming with <ctrl-S>, select what gets displayed (time, yaw, temperature):

```forth
<ctrl-s>
printmask
-time_trigger yaw_trigger or temp_trigger or set drop
```

Select the display rate (every 10 “triggers” specified by printtrigger below):

```forth
printmodulus 10 set drop
```

Select the display rate trigger (in this case, an update of yaw is set to be a trigger) and enable print streaming with <ctrl-q>:

```forth
printtrigger yaw_trigger set drop
<ctrl-q>
```

Response example (repeats):

```
P:,18754028,y,218.71,T,34.69
```

In the above response, the “P:” identifies the type of output. It is followed by the timestamp (milliseconds since power up), the ‘y’ identifies that the next number is yaw (degrees) and the ‘T’ identifies that the next number is the temperature (Celsius).

Note that if we print high rate data at a slower baud rate, it may not be possible to output all of the data due to the serial communication bandwidth limitations. In this case, entire lines of data will be dropped to ensure that the lines that do get output are complete. The data will still be computed at the normal rate within the NavMod.
To stop the streaming output, enter:

```plaintext
<ctrl-s>
printmask 0 set drop
printmodulus 0 set drop
printtrigger 0 set drop
<ctrl-q>
```

The NorthTek search command “grep” can be used to find other trigger symbols. **Reference:** Grep section in the *NorthTek System Programming Manual*. Enter:

```
grep trigger
```

Response example:

```
Wordlist : stuff
Wordlist : Forth
Wordlist : Core

Wordlist : Defines
  time_trigger temp_trigger quat_trigger yawt_trigger yaw_trigger
  gyrop_trigger gyror_trigger accelp_trigger accelr_trigger magp_trigger
  magr_trigger printtrigger
```

**NMEA**

**Reference:** NMEA section in the *Software Interface User’s Manual* for an introduction.

The acronym NMEA is used here to represent NMEA 0183 which is a communication standard for marine electronic devices. This standard is defined by the National Marine Electronics Association.

**NMEA Set-up**

Turn on key stroke “local echo” if you are using a terminal emulator OR issue the following command (once per session):

```
nmeaecho 1 set drop
```

If you are using a terminal emulator other than the one built into the NDS-1 Host Application, set up the terminal emulator to transmit a carriage return and line feed (i.e. <cr><lf> or hex 0d0a) upon hitting enter OR some terminal emulators like Tera Term will do so if you type <ctrl-j>.
**Standard NMEA command ($xxHDM):**

There is a 5 second timeout on entering each character in a NMEA command to prevent the NavMod from getting stuck in the protocol.

To get heading using the standard NMEA command, enter:

```
$xxHDM
```

Response example:

```
$HCHDM,300.4,M*2E
```

The 2E is the checksum and the ‘M’ indicates magnetic heading. **Reference: NMEA section in the Software Interface User’s Manual** for details on interpreting the response and for other NMEA protocol commands.

To get repeated values every 0.1 seconds, enter the command:

```
$xxHDM,RPT=0.1
```

To pause the output, enter:

```
<ctrl-s>
```

Enter any NMEA command to cancel the repeat, for example:

```
$xxHDM
<ctrl-q>
```

If you forget to enter the `<ctrl-q>`, then the next streaming command won’t output.

**Sparton Custom NMEA command:**

A NMEA command has been created that allows access to almost any of a long list of available NavMod variables (**reference**: the Variable Summary Table and the Variable Detailed Descriptions table in the Software Interface User’s Manual). For our magnetic heading example, the corresponding variable name is “yaw” (RPT can be applied to this one as well).

```
$PSRFS,yaw,get
```

Response example:

```
$PSRFS,yaw,286.672424*38
```

**Legacy Binary**


6 (Rev 2.2)
Legacy binary is intended to be backward compatible with the Sparton SP300X compasses when the DC4/6/AHRS8 is being used as a retrofit in an existing system. As a result, this protocol does not have access to any of the newer data such as the gyro data and is **not recommended** to be used with new designs.

Since it is a binary protocol, a terminal emulator that can send binary may be used or a custom interface program may be used to send the binary message.

Legacy Binary uses a header byte (0xA4) and a termination byte (0xA0). The 0x indicates the following 2 digits are in hexadecimal form (i.e. don’t send the 0x part). There is no checksum.

In this example, 0x09 is the identifier for magnetic heading. **Note:** The NDS-1 Host Application terminal emulator does not support this protocol. Use a terminal emulator that supports binary such as RealTerm.

The following example uses the RealTerm emulator. **Reference** the Terminal Emulator Application Note for RealTerm set-up instructions. In the Send tab, prefix the bytes with 0x and enter them in the “Send Numbers” window then press “Send Numbers”. The Legacy Binary example is shown here with a send of:

0xA4 0x09 0xA0

Note the binary response shows up in the black window.
The heading value in the response is 112 hex = 274 decimal.
To convert to degrees: $274 \times \frac{360}{4096} = 24.1$ degrees.

**RFS**

RFS is a flexible binary interface that provides a CRC for communication error checking. RFS can be used to access or control any of the variables listed in the long list of available variables (reference: the Variable Summary Table and the Variable Detailed Descriptions table in the Software Interface User’s Manual).

The big advantage of RFS is that it can be customized to efficiently provide just the data needed using “BitField”s (packed structures). There are three BitField variables defined, namely “position”, “composite2” and “composite3”. The “position” variable is unique in that it can be set up to repeat by using the “positionrate” variable to define the rate.

For an introduction and more examples, reference the RFS section in the Software Interface User’s Manual.


The following example uses the RealTerm terminal emulator and is followed by an interpretation of the data sent and received. You will need the RFS manual to follow the interpretation. Reference the Terminal Emulator Application Note for RealTerm set-up instructions.

RFS example to obtain yaw information:
For this Quick Start, we only desire the yaw data so we will show the one “Get” out of many that obtains the yaw information. Once the desired VID’s are obtained after NavMod start-up, then only “Get Value” commands need be used. The messages to obtain yaw data are detailed below with the binary (hexadecimal) values on the left and the descriptions of the binary data on the right.

Send a “Get” information request for VID 10 (yaw):

01  SOH - SAPP packet frame  
0B  SAPP packet body size  
40  Error Options/Protocol  
1081  SAPP Payload - Start of RFS packet  
00000000  DLE substitution for 01 - RFS Revision 1  
1081  RFS Payload size  
D8  sequence number  
0A  VID for yaw  
0A8C  2 CRC bytes for SAPP packet body  
03  ETX for SAPP packet frame

Receive the “GetResponse” message with a data description object plus the current value of yaw (see RealTerm figure above):

01  SOH - SAPP packet frame  
1F  SAPP packet body size  
60  Error Options/Protocol  
1081  SAPP Payload - Start of RFS packet  
00000014  DLE substitution for 01 - RFS Revision 1  
00  RFS Payload size  
D8  GetResponse (followed by Data Description plus Value)  
0A  sequence number of associated get request  
1090  VID for yaw  
1083  DLE substitution for 10hex - Scalar  
11  Description Field size 11hex = 17 decimal  
04  Text field length  
79617700  Name (Text Field): "yaw"  
00000000  Float Value Lower Limit  0.0  
43B40000  Float Value Upper Limit  360.0  
42F54EE9  Current Float32 Value  122.65  
5F3B  2 CRC bytes for SAPP packet body  
03  ETX for SAPP packet frame

Send a "Get Value" request for VID 10 (yaw):

01  SOH - SAPP packet frame  
0b  SAPP packet body size  
40  Error Options/Protocol  
1081  SAPP Payload - Start of RFS packet  
00000000  DLE substitution for 01 - RFS Revision 1  
08  RFS Payload size  
D9  sequence number  
0A  VID for yaw  
42F54EE9  End of RFS packet (SAPP payload)  
A72C  2 CRC bytes for SAPP packet body  
03  ETX for SAPP packet frame

Receive the “Value” message in response:

01  SOH - SAPP packet frame
In-field Calibration
Fixed magnetic interferences in the host device can be calibrated out using the in-field calibration.

The NDS-1 Host Application can be used to perform the in-field calibration but the in-field calibration can also be performed using NorthTek, NMEA commands or RFS.

Reference: In-Field Magnetic Calibration section in the NDS-1 User’s Manual
Reference: Magnetic Field Calibration section in the NMEA part of the Software Interface User’s Manual
Reference: 3D Compass Calibration section in the NorthTek part of the Software Interface User’s Manual

In this example, we shall use the cal3D NorthTek script that is available on the website downloads section or on the NavMod CD.

In the NDS-1 Terminal Emulator, click the “…” button, change the “Files of type:” at the bottom to “All Files”, and browse to the cal3D.4th script and click “Select”. Click the “Send File” button which will now download the NorthTek program titled “cal3D”. To confirm that the script loaded successfully, scroll back through the terminal emulator window and confirm that there are no “Huh”s.

Now run the script by entering:

    cal3D

Change the NavMod’s position and hit the space key. Repeat, attempting to achieve a variety of three dimensional orientations until calNumPoints is at least 4 and at most 12. Hit the Escape key when complete. When magErr has settled to a minimal value, hit the Escape key again to complete the procedure.

Someone who wishes to control in-field calibration from a host computer which is connected to the NavMod’s user port can emulate the cal3D script. There are two command variables to set (calmode and calCommand) and two status variables to read (calNumPoints and magErr). Reference the Variable Detailed Descriptions section in the appendix of the Software Interface User’s Manual for details on how to use these variables. These variables may be read or changed by any of the three protocols: NorthTek, NMEA and RFS.