Sparton Navigation Modules

# DC-4E Version 5.3.4 Release Notes

**Note that DC-4E software will only execute properly on DC-4E hardware. It will not work on the DC-4 hardware.**

This document details the differences between:

* DC-4E Version 4.4.1 and DC-4E Version 5.3.4

The version creation dates are:

* DC-4E Version 4.4.1 – 1/12/2016
* DC-4E Version 5.3.4 – 5/4/2016

# In-field Calibration Improvements

***3-D and 2-D Field Calibration Convergence Speed and Accuracy***

Improvements were made in the speed and accuracy of the 3-D field calibration convergence. The 3-D field calibration now works well even for points on in a 2-D plane.

***3-D and 2-D Field Calibration Option for Faster Convergence***

The convergence time can be reduced roughly in half by turning off the compass computations during the convergence process using the new variable compassOnForCalConvergence (defaults to On). This setting is stored in EEPROM (remembered through power cycles). The compass data will still be computed during the point selection. The command to speed up convergence is: “compassOnForCalConvergence 1 set drop”

***Fixed Field Calibration Compass Data Jump After First Point Selected***

In previous versions, the field calibration data was reset to defaults just after the first point was collected which sometimes resulted in a “jump” in the compass data being used to select the points. This version was changed to post-pone setting the old field calibration data to factory defaults until after all the points are selected. Note, that in the case where the old field calibration is invalid, the user should command the field calibration data back to defaults prior to entering calibration mode using “restoreFieldCal 1 set drop” in which case this change would show no difference.

***New Field Calibration Error Estimate - estimatedPk2PkYawError***

This score gives a better correlation for heading error. The variable estimatedPk2PkYawError gives an estimate of the peak to peak error for yaw.

***Instructions for Users Who Have Changed AdaptNav Configuration Variables***

Note that if you have previously changed the value of AdaptNav configuration variables such as infieldCalMu\* and calDeqDelta, then consider using the new default values since they have been improved. To use the default values, command the following:

0 eesectorerase

restorefactorycal 1 set drop

// then resend any other custom configurations since they will be cleared in the eesectorerase

# EEPROM Bug Fix

***EEPROM Based Configuration and Calibration Data is Now Validated Prior to Reading***

The user copies (not prefixed with f\_) of configuration and calibration data have always been double buffered in EEPROM to accommodate reset and power interruptions. Previous versions read the most recently written copy of the double buffer, but this has been changed to perform a CRC check and read the most recent and correct buffer.

***EEPROM Based Configuration and Calibration Data Failure Accommodation***

In the event that one of the double buffers is corrupted due to a reset or power interruption, the previous versions would overwrite the most recently written buffer first, where a second reset or power interruption would cause a loss of user configuration and calibration data. This version will write the corrupted buffer first so there will always be a valid buffer. Should they both be failed, then the user data will automatically revert to the factory (f\_) settings.

***EEPROM Event Log Created***

New event data is stored in EEPROM, separate from the EEPROM database data, to record EEPROM events and a start-up counter has been added as well. Each log data item is a 3 element integer array where the first element is a counter of the event (startUpCount), the second is the startUpCount at the most recent occurrence of the event, and the third element is the time in milliseconds of the last event. The event data can be cleared using “clearEEPROMLog 1 set drop”. The eeLogRestoreFactoryAuto event has a counter of the number of times that user data was automatically restored to factory settings due to the failure of both user EEPROM buffers. The eeLogRestoreFactoryUser event has a counter of the number of times that a user has intentionally commanded “restorefactorycal 1 set drop”. Please note that eeLogUserDB1Fail and eeLogUserDB2Fail will always count the same failure twice at startup because the first is due to the initial read attempt and the second is due to the write attempt. The event data is copied into the RAM based database and can be viewed using the following commands:

startUpCount di.

eeLogUserDB1Fail di.

eeLogUserDB2Fail di.

eeLogRestoreFactoryAuto di.

eeLogRestoreFactoryUser di.

***EEPROM Based Configuration and Calibration Data Failure Accommodation during Production***

The production calibration data (names start with f\_) is not double buffered since it is only intended to be written during production and there is a hardware write protection pin. It has been found that it is possible to lose factory configuration and calibration data if a reset or power down happens to occur during a write to the single buffer while at the factory. Changes to the production process have been made to prevent this as well as confirmation that the factory settings in EEPROM are correct.

***Fixed Bug That Occurred for Resets after Low Power Mode***

Fixed bug where user configuration and calibration may be defaults if a reset is performed after coming out of Low Power mode. This bug did not occur for power cycles after Low Power mode and did not affect the data stored in EEPROM.

***Write Delay After Permanent Variable Changes***

User changes to permanent variables invoke a database write to EEPROM and often multiple variables are changed in close succession. Previous versions delayed one second after the first variable change and this version now delays one second after the last variable change.

***Fixed Bug Where the Unit Could Lock Up When the User Writes a NorthTek Script to EEPROM***

The user has the option to write NorthTek scripts to EEPROM to be executed on start-up (see User Boot Program section in the user manual). In previous versions, it is possible that the write of the script could occur at the same time as an internal database write to EEPROM and cause a corruption of the EEPROM and potentially lock up the unit. The internal database writes occur periodically every 10 minutes and upon any permanent variable change by the user (with one second delay). Version 5.2.5 coordinates the writes and fixes this issue. Note for users with previous versions, Sparton has a process that can be used to repair the unit in the event a customer encounters the lockup problem. Another note for users with previous versions, if the initial script write succeeds, then no problem is to be expected later.

***Variable eepromWriteInProgress Now Applies to All EEPROM Operations***

In previous versions, eepromWriteInProgress could be used by the user to determine if an EEPROM database write was in progress in order to delay a power down and insure that the user’s configuration changes were saved. Version 5.2.5 now applies that variable to all EEPROM operations, such as writing a NorthTek script to EEPROM, to ensure the write was completed before powering down the unit. The intent is that, after the user makes configuration or EEPROM script changes, the user polls eepromWriteInProgress for 0 before powering down the unit if the power down is less than 6 seconds after the EEPROM changes.

***Fixed NorthTek Command “ee!” For Writes Across EEPROM Page Boundaries***

In previous versions, the ee! NorthTek command that crossed the EEPROM’s 256 byte page boundary would be wrapped around to the beginning of the page instead of on the next page as would be expected. This version fixes that problem.

# New NorthTek Command

***Added NorthTek command di.choice***

The new di.choice command can be used with database variables with the Ordinal type to list the options for that variable. See the example below and note that the number after the name is the value that must be used to change the variable.

orientation di.choice

Horizontal[0]

Vertical[1]

Left Edge[2]

Right Edge[3]

Inverted[4]

**5/5/16**