

```
diff -uNrx build -x 'obj*' /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/CMakeLists.txt
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/CMakeLists.txt
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/CMakeLists.txt 2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/CMakeLists.txt 2016-03-21
20:52:05.018981617 +0100
@@ -110,7 +110,8 @@
     src/from_ownership.h
     src/wxJSON/jsonval.cpp
     src/wxJSON/jsonreader.cpp
-
+   src/rev_counter.cpp
+   src/rev_counter.h
+ )

SET(SRC_NMEA0183
@@ -178,6 +179,8 @@
     src/nmea0183/mda.hpp
     src/nmea0183/xdr.cpp
     src/nmea0183/xdr.hpp
+   src/nmea0183/rpm.cpp
+   src/nmea0183/rpm.hpp
+ )
INCLUDE_DIRECTORIES(src/nmea0183)

diff -uNrx build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/po/dashboard_pi.pot
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/po/dashboard_pi.pot
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/po/dashboard_pi.pot 2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/po/dashboard_pi.pot 2016-03-19
22:45:42.503345893 +0100
@@ -170,10 +170,14 @@
 msgid "Pitch"
 msgstr ""

-#: src/dashboard_pi.cpp:182
+#: src/dashboard_pi.cpp:186
 msgid "Heel"
 msgstr ""

+#: src/dashboard_pi.cpp:188
+msgid "Rev Counter"
+msgstr ""
+
+
+#: src/dashboard_pi.cpp:352 src/dashboard_pi.cpp:465 src/dashboard_pi.cpp:1397
+#: src/dashboard_pi.cpp:1406 src/dashboard_pi.cpp:1568
+#: src/dashboard_pi.cpp:1607 src/dashboard_pi.cpp:1871
diff -uNrx build -x 'obj*' /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/po/de.po
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/po/de.po
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/po/de.po 2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/po/de.po 2016-03-19
22:44:15.611344144 +0100
@@ -169,10 +169,14 @@
 msgid "Pitch"
 msgstr "Nickwinkel"
```

```

-#: src/dashboard_pi.cpp:182
+#: src/dashboard_pi.cpp:186
msgid "Heel"
msgstr "Kr  gung"

+#: src/dashboard_pi.cpp:188
+msgid "Rev Counter"
+msgstr "Drehzahlmesser"
+
#: src/dashboard_pi.cpp:352 src/dashboard_pi.cpp:465 src/dashboard_pi.cpp:1397
#: src/dashboard_pi.cpp:1406 src/dashboard_pi.cpp:1568
#: src/dashboard_pi.cpp:1607 src/dashboard_pi.cpp:1871
diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/dashboard_pi.cpp
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/dashboard_pi.cpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/dashboard_pi.cpp      2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/dashboard_pi.cpp 2016-03-22
19:13:44.359762738 +0100
@@ -83,7 +83,7 @@
    ID_DBP_D_RSA, ID_DBP_I_SAT, ID_DBP_D_GPS, ID_DBP_I_PTR, ID_DBP_I_CLK, ID_DBP_I_SUN,
    ID_DBP_D_MON, ID_DBP_I_ATMP, ID_DBP_I_AWA, ID_DBP_I_TWA, ID_DBP_I_TWD, ID_DBP_I_TWS,
    ID_DBP_D_TWD, ID_DBP_I_HDM, ID_DBP_D_HDT, ID_DBP_D_WDH, ID_DBP_I_VLW1,
ID_DBP_I_VLW2, ID_DBP_D_MDA, ID_DBP_I_MDA, ID_DBP_D_BPH, ID_DBP_I_FOS,
-    ID_DBP_M_COG, ID_DBP_I_PITCH, ID_DBP_I_HEEL, ID_DBP_D_AWA_TWA,
+    ID_DBP_M_COG, ID_DBP_I_PITCH, ID_DBP_I_HEEL, ID_DBP_D_AWA_TWA, ID_DBP_D_RPM,
    ID_DBP_LAST_ENTRY //this has a reference in one of the routines; defining a "LAST_ENTRY"
and setting the reference to it, is one codeline less to change (and find) when adding new
instruments :-)
};

@@ -120,8 +120,8 @@
    case ID_DBP_D_AW:
    case ID_DBP_D_AWA:
        return _("App. Wind Angle & Speed");
-    case ID_DBP_D_AWA_TWA:
-        return _("App & True Wind Angle");
+    case ID_DBP_D_AWA_TWA:
+        return _("App & True Wind Angle");
    case ID_DBP_I_AWS:
        return _("App. Wind Speed");
    case ID_DBP_D_AWS:
@@ -180,10 +180,12 @@
        return _("Sum Log");
    case ID_DBP_I_FOS:
        return _("From Ownship");
-    case ID_DBP_I_PITCH:
-        return _("Pitch");
-    case ID_DBP_I_HEEL:
-        return _("Heel");
+    case ID_DBP_I_PITCH:
+        return _("Pitch");
+    case ID_DBP_I_HEEL:
+        return _("Heel");
+    case ID_DBP_D_RPM:
+        return _("Rev Counter");

```

```

    }
    return _T("");
}
@@ -218,8 +220,8 @@
    case ID_DBP_I_VLW1:
    case ID_DBP_I_VLW2:
    case ID_DBP_I_FOS:
-    case ID_DBP_I_PITCH:
-    case ID_DBP_I_HEEL:
+    case ID_DBP_I_PITCH:
+    case ID_DBP_I_HEEL:
        item.SetImage( 0 );
        break;
    case ID_DBP_D_SOG:
@@ -228,7 +230,7 @@
    case ID_DBP_D_AWA:
    case ID_DBP_D_AWS:
    case ID_DBP_D_TW:
-    case ID_DBP_D_AWA_TWA:
+    case ID_DBP_D_AWA_TWA:
    case ID_DBP_D_TWD:
    case ID_DBP_D_DPT:
    case ID_DBP_D_MDA:
@@ -239,6 +241,6 @@
    case ID_DBP_D_MON:
    case ID_DBP_D_WDH:
    case ID_DBP_D_BPH:
+    case ID_DBP_D_RPM:
        item.SetImage( 1 );
        break;
    }
@@ -922,6 +925,16 @@
    }
}

+    else if ( m_NMEA0183.LastSentenceIDReceived == _T("RPM") ) {
+        if( m_NMEA0183.Parse() ) {
+            if( m_NMEA0183.Rpm.IsDataValid == NTrue ) {
+                SendSentenceToAllInstruments( OCPN_DBP_STC_RPM,
+m_NMEA0183.Rpm.Speed/1000,
+                _T("\u00B0") );
+                _T("x1000 RPM") );
+            }
+        }
+    }
+
    else if( m_NMEA0183.LastSentenceIDReceived == _T("RSA") ) {
        if( m_NMEA0183.Parse() ) {
            if( m_NMEA0183.Rsa.IsStarboardDataValid == NTrue ) {
@@ -1058,17 +1071,17 @@
                SendSentenceToAllInstruments(OCPN_DBP_STC_PITCH,
xdrdata, xdrunit);
            }
            // NKE style of XDR Heel
-            if (m_NMEA0183.Xdr.TransducerInfo[i].TransducerName ==
_T("ROLL")) {
-                if (m_NMEA0183.Xdr.TransducerInfo[i].MeasurementData > 0)

```

```

-                                     xdrunit = _T("\u00B0R");
-                                     else if (m_NMEA0183.Xdr.TransducerInfo[i].MeasurementData
< 0) {
-                                     xdrunit = _T("\u00B0L");
-                                     xdrdata *= -1;
-                                     }
-                                     else
-                                     xdrunit = _T("\u00B0");
-                                     SendSentenceToAllInstruments(OCPN_DBP_STC_HEEL, xdrdata,
xdrunit);
-                                     } //Nasa style water temp
+                                     //~ if (m_NMEA0183.Xdr.TransducerInfo[i].TransducerName ==
_T("ROLL")) {
+                                     //~ if (m_NMEA0183.Xdr.TransducerInfo[i].MeasurementData >
0)
+                                     //~ xdrunit = _T("\u00B0R");
+                                     //~ else if
(m_NMEA0183.Xdr.TransducerInfo[i].MeasurementData < 0) {
+                                     //~ xdrunit = _T("\u00B0L");
+                                     //~ xdrdata *= -1;
+                                     //~ }
+                                     //~ else
+                                     //~ xdrunit = _T("\u00B0");
+                                     //~ SendSentenceToAllInstruments(OCPN_DBP_STC_HEEL,
xdrdata, xdrunit);
+                                     //~ } //Nasa style water temp
+                                     if (m_NMEA0183.Xdr.TransducerInfo[i].TransducerName ==
_T("ENV_WATER_T")){
+                                     SendSentenceToAllInstruments(OCPN_DBP_STC_TMP,
m_NMEA0183.Xdr.TransducerInfo[i].MeasurementData,m_NMEA0183.Xdr.TransducerInfo[i].UnitOfM
easurement);
+                                     }
@@ -2442,13 +2455,17 @@
        instrument = new DashboardInstrument_FromOwnship( this, wxID_ANY,
        getInstrumentCaption( id ) );
        break;
-        case ID_DBP_I_PITCH:
-            instrument = new DashboardInstrument_Single(this, wxID_ANY,
-                getInstrumentCaption(id), OCPN_DBP_STC_PITCH, _T("%2.1f"));
-            break;
-        case ID_DBP_I_HEEL:
-            instrument = new DashboardInstrument_Single(this, wxID_ANY,
-                getInstrumentCaption(id), OCPN_DBP_STC_HEEL, _T("%2.1f"));
+        case ID_DBP_D_RPM:
+            instrument = new DashboardInstrument_RevCounter( this, wxID_ANY,
+                getInstrumentCaption( id ), OCPN_DBP_STC_RPM, 0,6);
+            break;
+        case ID_DBP_I_PITCH:
+            instrument = new DashboardInstrument_Single(this, wxID_ANY,
+                getInstrumentCaption(id), OCPN_DBP_STC_PITCH, _T("%2.1f"));
+            break;
+        case ID_DBP_I_HEEL:
+            instrument = new DashboardInstrument_Single(this, wxID_ANY,
+                getInstrumentCaption(id), OCPN_DBP_STC_HEEL, _T("%2.1f"));
+            }
        if( instrument ) {
            instrument->instrumentTypeId = id;

```

```

diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/dashboard_pi.h
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/dashboard_pi.h
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/dashboard_pi.h 2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/dashboard_pi.h 2016-03-19
21:44:32.519272005 +0100
@@ -62,6 +62,7 @@
#include "wind_history.h"
#include "baro_history.h"
#include "from_ownership.h"
+#include "rev_counter.h"

class DashboardWindow;
class DashboardWindowContainer;
diff -uNr build -x 'obj*' /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/instrument.h
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/instrument.h
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/instrument.h 2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/instrument.h 2016-03-19
21:44:40.815272172 +0100
@@ -90,8 +90,9 @@
    OCPN_DBP_STC_VLW2 = 1 << 27, // Sum Log
    OCPN_DBP_STC_MDA = 1 << 28, // Bareometric pressure
    OCPN_DBP_STC_MCOG = 1 << 29, // Magnetic Course over Ground
-    OCPN_DBP_STC_PITCH = 1 << 30, //Pitch
-    OCPN_DBP_STC_HEEL = 1 << 31 //Heel
+    OCPN_DBP_STC_PITCH = 1 << 30, //Pitch
+//    OCPN_DBP_STC_HEEL = 1 << 31, //Heel
+    OCPN_DBP_STC_RPM = 1 << 31 // Engine speed
};

class DashboardInstrument : public wxControl
diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/nmea0183.cpp
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/nmea0183.cpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/nmea0183.cpp
2016-02-03 18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/nmea0183.cpp
2016-03-19 23:21:09.587388718 +0100
@@ -98,8 +98,9 @@
    response_table.Append( (RESPONSE *) &Rmc );
/*
    response_table.Append( (RESPONSE *) &Rot );
-   response_table.Append( (RESPONSE *) &Rpm );
*/
+   response_table.Append( (RESPONSE *) &Rpm );
+
    response_table.Append( (RESPONSE *) &Rsa );
/*
    response_table.Append( (RESPONSE *) &Rsd );
diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/nmea0183.h
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/nmea0183.h
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/nmea0183.h
2016-02-03 18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/nmea0183.h 2016-03-19

```

```
07:17:33.654224702 +0100
@@ -84,6 +84,13 @@
    West
} EASTWEST;

+typedef enum _shaftengine
+{
+  SE_Unknown = 0,
+  Shaft,
+  Engine
+} SHAFTENGINE;
+
+typedef enum _northsouth
+{
+  NS_Unknown = 0,
diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/nmea0183.hpp
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/nmea0183.hpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/nmea0183.hpp
2016-02-03 18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/nmea0183.hpp
2016-03-21 20:59:59.026991160 +0100
@@ -114,6 +114,7 @@
#include "vlw.hpp"
#include "xdr.hpp" //transducer data
#include "xte.hpp"
+ #include "rpm.hpp"
/*
#include "ROT.hpp"
#include "RPM.hpp"
@@ -215,8 +216,9 @@
    RMC Rmc;
/*
    ROT Rot;
+ */
    RPM Rpm;
- */
+
    RSA Rsa;
/*
    RSD Rsd;
diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/rpm.cpp
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/rpm.cpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/rpm.cpp    1970-01-01
01:00:00.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/rpm.cpp    2016-03-21
21:11:40.007005273 +0100
@@ -0,0 +1,135 @@
+ /*****
+ *
+ * Project:  OpenCPN
+ * Purpose:  NMEA0183 Support Classes
+ * Author:   Samuel R. Blackburn, David S. Register
+ *
+ *****/
+ * Copyright (C) 2010 by Samuel R. Blackburn, David S Register *
```



```
+}
+
+bool RPM::Parse( const SENTENCE& sentence )
+{
+//  ASSERT_VALID( this );
+
+  /*
+  RPM - Revolutions
+
+      1 2 3  4  5 6
+      ||| | ||
+  $--RPM,a,x,x.x,x.x,A*hh<CR><LF>
+
+  Field Number:
+  1) Sourse, S = Shaft, E = Engine
+  2) Engine or shaft number
+  3) Speed, Revolutions per minute
+  4) Propeller pitch, % of maximum, "-" means astern
+  5) Status, A means data is valid
+  6) Checksum    ** First we check the checksum...
+  */
+
+  if ( sentence.IsChecksumBad( 6 ) == TRUE )
+  {
+    SetErrorMessage( _T("Invalid Checksum") );
+    return( FALSE );
+  }
+
+  Sourse      = sentence.ShaftOrEngine ( 1 );
+  Number      = sentence.Integer ( 2 );
+  Speed       = sentence.Double( 3 );
+  Pitch       = sentence.Double( 4 );
+  IsDataValid = sentence.Boolean( 5 );
+
+  return( TRUE );
+}
+
+bool RPM::Write( SENTENCE& sentence )
+{
+//  ASSERT_VALID( this );
+
+  /*
+  ** Let the parent do its thing
+  */
+
+  RESPONSE::Write( sentence );
+
+  sentence += Sourse;
+  sentence += Number;
+  sentence += Speed;
+  sentence += Pitch;
+  sentence += IsDataValid;
+
+  sentence.Finish();
+
+  return( TRUE );
+}
```



```
+
+const RPM& RPM::operator = ( const RPM& source )
+{
+//  ASSERT_VALID( this );
+
+  Source          = source.Source;
+  Number = source.Number;
+  Speed           = source.Speed;
+  IsDataValid     = source.IsDataValid;
+  Pitch = source.Pitch;
+
+  return( *this );
+}
diff -uNrx build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/rpm.hpp
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/rpm.hpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/rpm.hpp    1970-01-01
01:00:00.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/rpm.hpp    2016-03-21
20:08:43.034929231 +0100
@@ -0,0 +1,78 @@
+/******
+ *
+ * Project:  OpenCPN
+ * Purpose:  NMEA0183 Support Classes
+ * Author:   Gerald Supper copied changed from rsa.hpp
+ *
+ * *****/
+ * Copyright (C) 2016 Gerald Supper
+ *
+ * This program is free software; you can redistribute it and/or modify
+ * it under the terms of the GNU General Public License as published by
+ * the Free Software Foundation; either version 2 of the License, or
+ * (at your option) any later version.
+ *
+ * This program is distributed in the hope that it will be useful,
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
+ * GNU General Public License for more details.
+ *
+ * You should have received a copy of the GNU General Public License
+ * along with this program; if not, write to the
+ * Free Software Foundation, Inc.,
+ * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
+ * *****/
+ *
+ * S Blackburn's original source license:
+ * "You can use it any way you like."
+ * More recent (2010) license statement:
+ * "It is BSD license, do with it what you will"
+ */
+
+
+
+#if ! defined( RPM_CLASS_HEADER )
+#define RPM_CLASS_HEADER
+
+
+
+/**
```

```

+** Author: Samuel R. Blackburn
+** CId: 76300,326
+** Internet: sammy@sed.csc.com
+**
+** You can use it any way you like.
+*/
+
+class RPM : public RESPONSE
+{
+//  DECLARE_DYNAMIC( RPM )
+
+  public:
+
+    RPM();
+    ~RPM();
+
+    /*
+    ** Data
+    */
+
+    double      Speed;
+    NMEA0183_BOOLEAN IsDataValid;
+    double      Pitch;
+    SHAFTENGINE   Source;
+    int          Number;
+
+    /*
+    ** Methods
+    */
+
+    virtual void Empty( void );
+    virtual bool Parse( const SENTENCE& sentence );
+    virtual bool Write( SENTENCE& sentence );
+
+    /*
+    ** Operators
+    */
+
+    virtual const RPM& operator = ( const RPM& source );
+};
+
+#endif // RPM_CLASS_HEADER
diff -uNr build -x 'obj*'
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/rsa.cpp
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/rsa.cpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/rsa.cpp      2016-02-03
18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/rsa.cpp      2016-03-23
21:37:36.383746874 +0100
@@ -87,18 +87,53 @@
     ** First we check the checksum...
     */

-    if ( sentence.IsChecksumBad( 5 ) == TRUE )
-    {
-        SetErrorMessage( _T("Invalid Checksum") );
-        return( FALSE );

```

```

- }
+ int target_field_count = 2;
+
+ NMEA0183_BOOLEAN check = sentence.IsChecksumBad( 3 );
+
+ if ( check == NTrue )
+ {
+
+ /*
+ ** This may be an NMEA Version 2.3 sentence, with "Mode" field
+ */
+
+ wxString checksum_in_sentence = sentence.Field( 3 );
+ if(checksum_in_sentence.StartsWith(_T("*"))) // Field is a valid erroneous checksum
+ {
+     SetErrorMessage( _T("Invalid Checksum") );
+     return( FALSE );
+ }
+
+ else
+ {
+     target_field_count = 4;
+     check = sentence.IsChecksumBad( 5 );
+     if( check == NTrue )
+     {
+         SetErrorMessage( _T("Invalid Checksum") );
+         return( FALSE );
+     }
+ }
+ }
+
+ if ( sentence.GetNumberOfDataFields() != target_field_count )
+ {
+     SetErrorMessage( _T("Invalid FieldCount") );
+     return( FALSE );
+ }
+
+
+ Starboard      = sentence.Double( 1 );
+ IsStarboardDataValid = sentence.Boolean( 2 );
- Port          = sentence.Double( 3 );
- IsPortDataValid   = sentence.Boolean( 4 );
-
- return( TRUE );
+ if (target_field_count == 4) {
+     Port          = sentence.Double( 3 );
+     IsPortDataValid   = sentence.Boolean( 4 );
+ }
+ return( TRUE );
+
+ }

```

```
bool RSA::Write( SENTENCE& sentence )
```

```
diff -uNr build -x 'obj*'
```

```
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/sentence.cpp
```

```
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/sentence.cpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/sentence.cpp
2016-02-03 18:44:17.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/sentence.cpp
2016-03-22 19:21:37.223772258 +0100
```

```
@@ -1,4 +1,4 @@
```

```
-/*****
+ /*****
*
```

```
* Project: OpenCPN
```

```
* Purpose: NMEA0183 Support Classes
```

```
@@ -179,6 +179,28 @@
```

```
}
```

```
}
```

```
+SHAFTENGINE SENTENCE::ShaftOrEngine( int field_number ) const
```

```
+{
```

```
+//  ASSERT_VALID( this );
```

```
+
```

```
+  wxString field_data;
```

```
+
```

```
+  field_data = Field( field_number );
```

```
+
```

```
+  if ( field_data == _T("S") )
```

```
+  {
```

```
+    return( Shaft );
```

```
+  }
```

```
+  else if ( field_data == _T("E") )
```

```
+  {
```

```
+    return( Engine );
```

```
+  }
```

```
+  else
```

```
+  {
```

```
+    return( SE_Unknown );
```

```
+  }
```

```
+}
```

```
+
```

```
const wxString& SENTENCE::Field( int desired_field_number ) const
```

```
{
```

```
//  ASSERT_VALID( this );
```

```
diff -uNr build -x 'obj*'
```

```
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/Sentence.hpp
```

```
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/Sentence.hpp
```

```
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/nmea0183/Sentence.hpp
```

```
2016-02-03 18:44:17.000000000 +0100
```

```
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/nmea0183/Sentence.hpp
```

```
2016-03-19 07:17:33.662224702 +0100
```

```
@@ -74,6 +74,7 @@
```

```
virtual NMEA0183_BOOLEAN IsChecksumBad( int checksum_field_number ) const;
```

```
virtual LEFTRIGHT LeftOrRight( int field_number ) const;
```

```
virtual NORTHSOUTH NorthOrSouth( int field_number ) const;
```

```
+ virtual SHAFTENGINE ShaftOrEngine( int field_number ) const;
```

```
virtual REFERENCE Reference( int field_number ) const;
```

```
virtual TRANSDUCER_TYPE TransducerType( int field_number ) const;
```

```
diff -uNr build -x 'obj*'
```

```
/home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/rev_counter.cpp
```

```
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/rev_counter.cpp
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/rev_counter.cpp 1970-01-01
01:00:00.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/rev_counter.cpp 2016-03-24
21:07:39.654532825 +0100
@@ -0,0 +1,53 @@
+ /*****
+  * $Id: RevCounter.cpp, v1.0 2010/08/05 SethDart Exp $
+  *
+  * Project: OpenCPN
+  * Purpose: Dashboard Plugin
+  * Author: Jean-Eudes Onfray
+  *
+  *****/
+ * Copyright (C) 2010 by David S. Register *
+ *
+ * This program is free software; you can redistribute it and/or modify *
+ * it under the terms of the GNU General Public License as published by *
+ * the Free Software Foundation; either version 2 of the License, or *
+ * (at your option) any later version. *
+ *
+ * This program is distributed in the hope that it will be useful, *
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of *
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
+ * GNU General Public License for more details. *
+ *
+ * You should have received a copy of the GNU General Public License *
+ * along with this program; if not, write to the *
+ * Free Software Foundation, Inc., *
+ * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. *
+ *****/
+ */
+
+#include "rev_counter.h"
+
+// For compilers that support precompilation, includes "wx/wx.h".
+#include <wx/wxprec.h>
+
+#ifdef __BORLANDC__
+ #pragma hdrstop
+#endif
+
+// for all others, include the necessary headers (this file is usually all you
+// need because it includes almost all "standard" wxWidgets headers)
+#ifndef WX_PRECOMP
+ #include <wx/wx.h>
+#endif
+
+// Not much to do here most of the default dial values are fine.
+// Note the default AngleStart = 225 and AngleRange = 270 set here.
+
+DashboardInstrument_RevCounter::DashboardInstrument_RevCounter( wxWindow *parent,
wxWindowID id, wxString title, int cap_flag,
+ int s_value, int e_value) : DashboardInstrument_Dial( parent, id, title,
cap_flag, 225, 270, s_value, e_value)
+{
+ // We want the main value displayed inside the dial as well
```

```
+ // as the default arrow
+ SetOptionMainValue(_T("%.2f"), DIAL_POSITION_INSIDE);
+}
+
diff -uNrX build -x 'obj*' /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/rev_counter.h
/home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/rev_counter.h
--- /home/gerald/opencpn-4.2.0.orig/plugins/dashboard_pi/src/rev_counter.h    1970-01-01
01:00:00.000000000 +0100
+++ /home/gerald/opencpn-4.2.0/plugins/dashboard_pi/src/rev_counter.h    2016-03-19
06:58:12.330201321 +0100
@@ -0,0 +1,68 @@
+ /*****
+ * $Id: RevCounter.h, v1.0 2010/08/05 SethDart Exp $
+ *
+ * Project: OpenCPN
+ * Purpose: Dashboard Plugin
+ * Author: Jean-Eudes Onfray
+ *
+ *****/
+ * Copyright (C) 2010 by David S. Register *
+ *
+ * This program is free software; you can redistribute it and/or modify *
+ * it under the terms of the GNU General Public License as published by *
+ * the Free Software Foundation; either version 2 of the License, or *
+ * (at your option) any later version. *
+ *
+ * This program is distributed in the hope that it will be useful, *
+ * but WITHOUT ANY WARRANTY; without even the implied warranty of *
+ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the *
+ * GNU General Public License for more details. *
+ *
+ * You should have received a copy of the GNU General Public License *
+ * along with this program; if not, write to the *
+ * Free Software Foundation, Inc., *
+ * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. *
+ *****/
+ */
+
+#ifndef __RevCounter_H__
+#define __RevCounter_H__
+
+// For compilers that support precompilation, includes "wx/wx.h".
+#include <wx/wxprec.h>
+
+#ifdef __BORLANDC__
+ #pragma hdrstop
+#endif
+
+// for all others, include the necessary headers (this file is usually all you
+// need because it includes almost all "standard" wxWidgets headers)
+#ifndef WX_PRECOMP
+ #include <wx/wx.h>
+#endif
+
+#include "dial.h"
+
+//+-----
```

```
+//|
+//| CLASS:
+//|   DashboardInstrument_RevCounter
+//|
+//| DESCRIPTION:
+//|   This class creates a RevCounter style control
+//|
+//|+-----
+class DashboardInstrument_RevCounter: public DashboardInstrument_Dial
+{
+    public:
+        DashboardInstrument_RevCounter( wxWindow *parent, wxWindowID id, wxString title,
+int cap_flag,
+                                     int s_value, int e_value);
+
+        ~DashboardInstrument_RevCounter(void){}
+
+    private:
+
+};
+
+#endif // __RevCounter_H__
+
```