Operating the Q330HR with an External Time Source

upgrading the GRSN DAQ at BFO from Q680 to Q330HR, Revision April 29, 2008

1 Introduction

This writeup describes how to synchronize a Q330 digitizer with universal time (UT) in a situation where no GPS signal is available, such as in a mine installation like BFO.

2 Basic timing requirements of the Q330

In order to record seismic data with time stamps based on Universal Time (UT) the Q330's internal clock has to be synchronized with Universal Time (UT). The Q330 offers three ways in which this can be accomplished:

A Connect an external GPS antenna directly to the Q330. The Q330 has a built-in GPS receiver which can be used to discipline the Q330's internal clock. This is the standard way of providing time to the Q330.

B If two Q330s are operated side-by-side, only one Q330 has to have a GPS antenna since every Q330 can also serve as time serves for other Q330s.

C If the Q330 is too far away from the surface such that the RF GPS signal from the antenna can no longer be transported over a coaxial cable other techniques must be used to provide the Q330 with a time signal. For this situation the Q330 can accept an external 1 pps (Pulse Per Second) electrical signal and a time telegram. Both of these signals are only poorly described in the Q330 manuals and are the topic of this writeup.

(B) and (C) are similar in that they both rely on an external GPS receiver. Furthermore the 1pps signal and the time telegrams have to be identical in (B) and (C). In fact, to find out what is needed in scenario (C) one can configure the Q330 to act as a time server and "listen" on the GPS EXT connector.

For underground installation these three methods of synchronizing the Q330 open up the possibility for two fundamentally different technical solutions:

I the broadband GPS signal is transported to the Q330 and the internal GPS receiver is used.

II an separate GPS receiver is operated at the Earths surface and this receiver generates a 1pps signal and a stream of time telegrams. These two signals are then transported to the Q330 and supplied via the the GPS EXT connector.

One commercial solution implementing strategy (I) is from Meinberg and uses their GPSGEN/GOAL products. In essence the GPS signal is demodulated, transmitted over a multimode fiber, and at the far end of the fiber the signal is again modulated onto the 1.57 GHz GPS carrier before it gets reradiated or fed via coaxial cable straight into the antenna input of the Q330.

Since transporting narrow band signals is easier we discuss here how the 1 pps signal an the time telegrams have to look like so that implementing strategy (II) becomes possible.

2.1 Electric definition of required input signals

The time telegrams are transmitted over a serial RS-232 line. The 1pps signal is also an RS-232 signal in the sense that it varies between $\pm$ 12V. Both these signals are referred to the same ground. The time telegrams are transmitted with 4800 baud and 8N1 (8 data bits, no parity and one stop bit). Note, the Q330 uses the falling edge to synchronize.

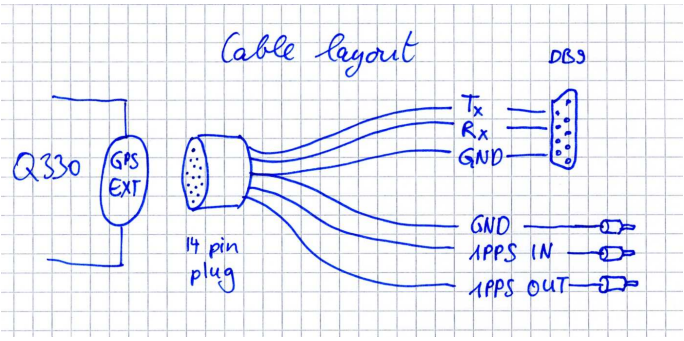

Figure 1: Possible layout of a cable to hook up the 1 pps signal and the RS232 serial line with the time telegrams to the Q330. Note that the 1 pps signal and the serial line refer to the same signal ground GND.

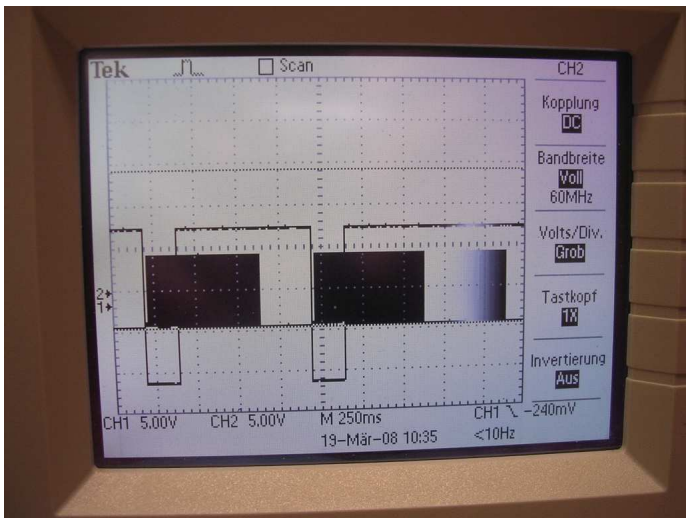

Figure 2: Overlay of the two signals produced by the M12 GPS receiver built into the Q330 as measured at the EXT GPS plug. The time axis covers 2.5 seconds. The amplitude of the 1 pps signal is 18V. The time pulse is 200ms long. Transmission of the time telegram (visible as black blocks) takes just about 600ms. Onset of the telegram is 10ms delayed with respect to the falling edge of the 1pps signal. The two signals are vertically offset for clarity. The amplitude of the telegram is 9V.

2.2 Connector layout for EXT GPS

A mating plug from ITT Cannon is the KPT06F12-14P. We have obtained it from `www.cosy.de`.

Pin	name	signal
A	X4TXD	GPS Tx RS232
B	X4RXD	GPS Tx RS232
C	X4GND	GPS Tx RS232
D	X41PPSOUT	GPS Tx RS232
E	X4PWR	GPS Tx RS232
F	X41PPSIN	GPS Tx RS232
G	X41PPS+	GPS Tx RS422
H	X41PPS-	GPS Tx RS422

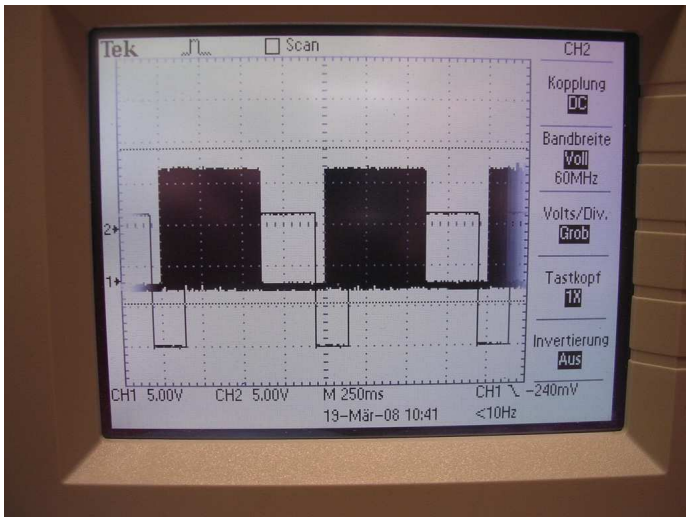

Figure 3: Overlay of the two signals fed into the Q330. The 1pps signal is from a DCF77 Meinberg PZF radio clock and the time telegram is generated by the NTP-to-M12.pl perl script (see appendix) which uses the PC clock as time reference. The time axis covers 2.5 seconds. The amplitude of the 1pps signal is 16V. The time pulse is 200ms long. Transmission of the time telegram (visible as black blocks) takes just about 600ms. Onset of the telegram is 50ms delayed with respect to the falling edge of the 1pps signal. The two signals are vertically offset for clarity. The amplitude of the time telegram is here 15V.

3 Our solution: technical details

We try two solutions:

1 We use our Meinberg GPS-167 observatory clock which contains a highly stable oscillator and produces both a 1 pps electrical signal as well as a stream of time telegrams. To transport the 1 pps signal into the mine we use a dedicated optical fiber. To transport the time telegram we use the LAN which also reaches via optical fibers into the mine.

2 Since our setup involves a PC104 Linux PC located right next to the Q330 we can use the system clock of the LinuxPC to generate the time telegrams and rely on NTP (Network Time Protocol) to keep the PC clock synced with UT.

3.1 Format of time telegrams

The Meinberg GPS-167 clock can be programmed to output NMEA-0187 GPRMC style telegrams. An example of six successive telegrams is given in appendix 1.

```
$GPRMC,153957.00,A,4819.76,N,00819.39,E,0.0,0.0,110308,0.0,E*50
$GPRMC,153958.00,A,4819.76,N,00819.39,E,0.0,0.0,110308,0.0,E*5F
$GPRMC,153959.00,A,4819.76,N,00819.39,E,0.0,0.0,110308,0.0,E*5E
$GPRMC,154000.00,A,4819.76,N,00819.39,E,0.0,0.0,110308,0.0,E*5C
$GPRMC,154001.00,A,4819.76,N,00819.39,E,0.0,0.0,110308,0.0,E*5D
$GPRMC,154002.00,A,4819.76,N,00819.39,E,0.0,0.0,110308,0.0,E*5E
```

The GPS receiver built into the Q330 outputs more than just time telegrams. It produces also telegrams specifying the geographic coordinates of the observer and a list of visible satellites, together with their coordinates and the signal strength. An example is given in appendix 2.

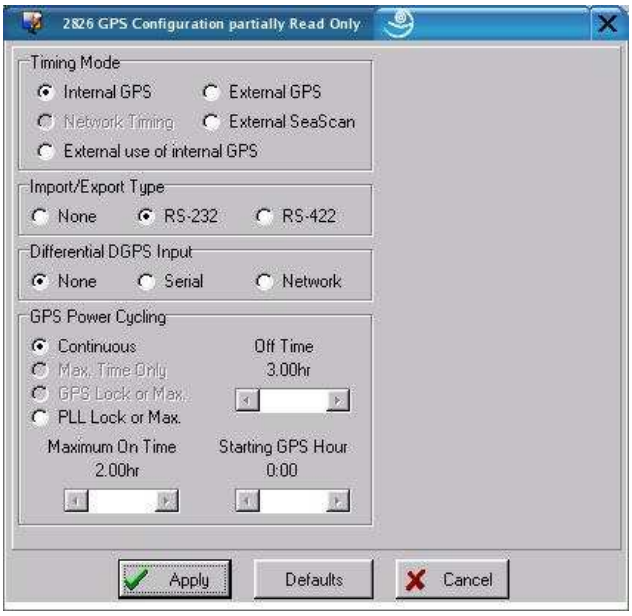

Figure 4: Q330 configured as a time server. With this configuration you can ”listen” on the GPS EXT connector to find out what the Q330 expects as inputs.

4 Configuring the Q330 for external timing

5 Testing the setup

```
$GPZDA,110958.00,27,02,2008,00,00*6F
$GPGGA,,4819.7681,N,00819.3761,E,0,02,,665.9,M,,M,,*72
$GPGSA,A,1,,,,11,,,28,,,,,,,,*14
$GPGSV,3,1,10,01,31,128,,03,33,158,,09,00,029,,11,42,284,45*76
$GPGSV,3,2,10,14,38,092,,19,66,168,,20,12,225,,22,34,054,*79
$GPGSV,3,3,10,28,19,314,38,32,65,069,*7A
$GPZDA,110959.00,27,02,2008,00,00*6E
$GPGGA,,4819.7681,N,00819.3761,E,0,02,,665.9,M,,M,,*72
$GPGSA,A,1,,,,11,,,28,,,,,,,,*14
$GPGSV,3,1,10,01,31,128,,03,33,158,,09,00,029,,11,42,284,46*75
$GPGSV,3,2,10,14,38,092,,19,66,168,,20,12,225,,22,34,054,*79
$GPGSV,3,3,10,28,19,314,44,32,65,069,*71
$GPZDA,111000.00,27,02,2008,00,00*6A
$GPGGA,,4819.7681,N,00819.3761,E,0,02,,665.9,M,,M,,*72
$GPGSA,A,1,,,,11,,,28,,,,,,,,*14
$GPGSV,3,1,10,01,31,128,,03,33,158,,09,00,029,,11,42,284,45*76
$GPGSV,3,2,10,14,38,092,,19,66,168,,20,12,225,,22,34,054,*79
$GPGSV,3,3,10,28,19,314,41,32,65,069,*74
$GPZDA,111001.00,27,02,2008,00,00*6B
$GPGGA,,4819.7681,N,00819.3761,E,0,02,,665.9,M,,M,,*72
$GPGSA,A,1,,,,11,,,28,,,,,,,,*14
$GPGSV,3,1,10,01,31,128,,03,33,158,,09,00,029,,11,42,284,46*75
$GPGSV,3,2,10,14,38,092,,19,66,168,,20,12,225,,22,34,054,*79
$GPGSV,3,3,10,28,19,314,41,32,65,069,*74
```

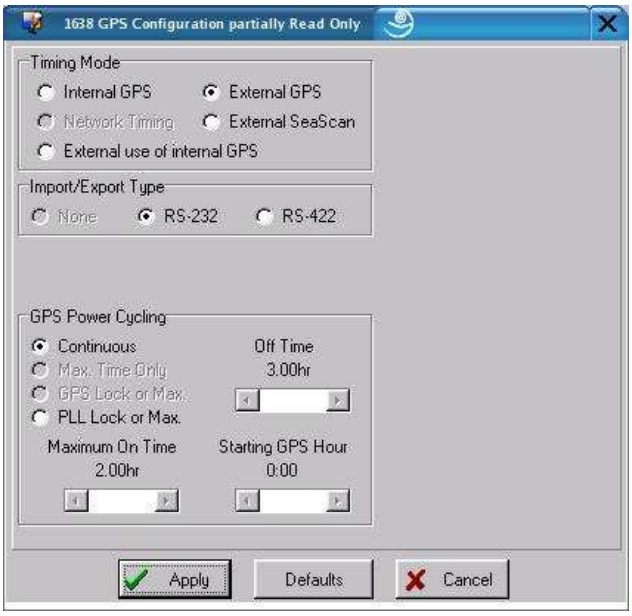

Figure 5: Q330 configured as a time client.

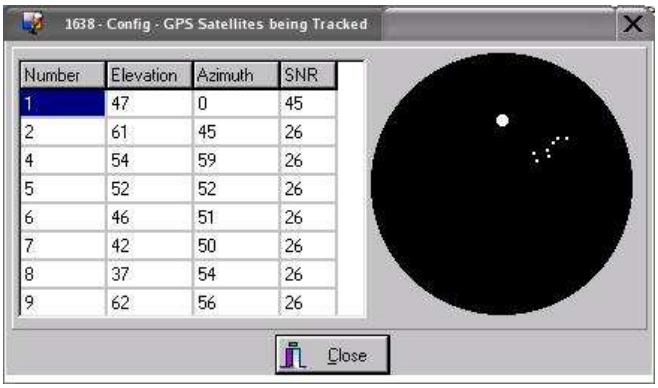

Figure 6: The artificial satellite configuration generated by the attached two perls scripts. If you see those the Q330 ingests the time telegrams.

5.1 Useful links

- For documentation of the Q330 and Q330HR: `www.q330.com`
- To decipher the NMEA telegrams: `http://home.mira.net/ gnb/gps/nmea.html`
- Our favorite supplier of exotic plugs (to connect to the Q330): `www.cosy.de`

6 Acknowledgments

Kinemetrics technical support. Jean-Yves (EOST),

```
#!/usr/bin/perl
#
#   Perl script to produce Q330-compatible NMEA-0183 time telegrams
#          based on the PC system clock.
#
#      We assume that NTP is used to discipline the PC clock.
#
#      For timing the Q330HR need also a 1pps signal which has
#           to be dealt with separately.
#
#      Every second we generate 6 telegrams:
#           one $GPZDA, $GPGGA, $GPGSA and three $GPGSV
#         ( For the meaning of these telegrams see:
#              http://home.mira.net/~gnb/gps/nmea.html )
#
#                           Rudolf Widmer-Schnidrig, BFO, 13.3.2008
#
$\ = "\n";              # set output record separator
#
# load Perl module HiRes for high resolution time manipulation.
#
use Time::HiRes qw( usleep gettimeofday );
#
require "flush.pl";         # Perl module to permit instant flushing of buffers
#
# station coordinates for BFO: these will be communicated to the Q330HR
#    and might show up in some SEED header. Change as needed.
#
      $lat  = "4819.7681";  # 48deg 19.7681'N
      $NS   = "N";
      $lon  = "00819.3761"; #  8deg 19.3761'E
      $EW   = "E";
      $elev = "665.9";      # elevation in meters

      $gpgga = sprintf("GPGGA,,%s,%s,%s,%s,1,08,,%s,M,,M,,",$lat,$NS,$lon,$EW,$elev);
      $line0 = sprintf("\$%s*%02X",$gpgga,&CHECKSUM($gpgga));
#
#  Enter some arbitrary GPS satelite constellation.
#    View the constellation under Willard -> Status -> GPS Satellites
#         Here: Polaris and Big Dipper constellation
#
      $gpgsa =sprintf("GPGSA,A,1,2,3,4,5,6,7,8,9,,,,,,,");
      $line1 =sprintf("\$%s*%02X",$gpgsa,&CHECKSUM($gpgsa));

      $gpgsv =sprintf("GPGSV,3,1,9,01,47,0,45,02,61,45,26,03,62,56,10,04,54,59,26");
      $line2 =sprintf("\$%s*%02X",$gpgsv,&CHECKSUM($gpgsv));

      $gpgsv =sprintf("GPGSV,3,2,9,05,52,52,26,06,46,51,26,07,42,50,26,08,37,54,26");
      $line3 =sprintf("\$%s*%02X",$gpgsv,&CHECKSUM($gpgsv));

      $gpgsv =sprintf("GPGSV,3,3,9,09,62,56,26");
      $line4 =sprintf("\$%s*%02X",$gpgsv,&CHECKSUM($gpgsv));
#
#  work through this while loop once every second
#
while (1)
  {
# get time of day
      $t0 = [gettimeofday];
      ($seconds, $microseconds) = gettimeofday;
      printf("Ready to go to sleep at %d usec.\n",$microseconds);
#
#  send next time telegram 50ms after the full second
#
      $sleep_for = 1000* (1000 + 50) - $microseconds;
#
#  sleep now
#
      usleep ($sleep_for);
#
#  did you wake up at the right time?
#
      $t0 = [gettimeofday];
      ($seconds, $microseconds) = gettimeofday;
      printf("\nWaking up %d usec after the full second.\n",$microseconds);
```

```
#
#  get UT time from PC clock and generate new telegram
#
      ($sec,$mn,$hr,$day,$mon,$yr,$wday,$jday,$dst) = gmtime;
#
#   manipulate time string just for testing
#
      $ss  = ((3600 * $hr) + 60 * $mn ) + $sec;

#     $ss  = $ss + 10;

      $hr  = int( $ss / 3600 );
      $ss  = $ss - 3600*$hr;

      $mn  = int( $ss / 60 );
      $ss  = $ss - 60*$mn;

      $now = sprintf("%02d%02d%02d.00",$hr,$mn,$ss);

      $gpzda = sprintf("GPZDA,%s,%02d,%02d,%4d,00,00",$now,$day,$mon,$yr);
      printf("\$%s*%02X\n",$gpzda,&CHECKSUM($gpzda));

      printf("%s\n%s\n%s\n%s\n%s",$line0,$line1,$line2,$line3,$line4);

      &flush(STDOUT);

     }
#
#  subroutine to compute checksum of GPS telegram according to NMEA
#
sub CHECKSUM {

        local($string) = $_[0];
        local($len, $i, $offset, $a);

        $chksum = 0;
        $len    = length($string);

        for ( $i = 1 ; $i <= $len ; $i++ )
          {
            $offset = $i - 1;
            $a = vec($string, $offset, 8);
            $chksum = $chksum ^ $a;         #  bitwise XOR operator
          }
        return ($chksum);
       }

#-----------------------------------end--------------------------------#
```

```
#!/usr/bin/perl -n
#
#   Perl script to reformat the Meinberg GPS-167 telegram
#       for use in the Q330HR
#
#      Note: we expect that the Meinberg is programmed to
#                     generate  NMEA GPRMC sentences.
#
#      for every GPRMC telegram we generate 5 telegrams:
#           one $GPZDA, $GPGGA, $GPGSA and two $GPGSV
#
#                        Rudolf Widmer-Schnidrig, BFO, 29.2.2008
#
$\ = "\n";              # set output record separator

require "flush.pl";     # flush output buffer immediately

s/\$//;                 # delete dollar sign from time string
@item=split(/,/);

if ( $item[0] eq "GPRMC" )
     {$hms = $item[1];
      $hr  = substr($hms,0,2);
      $mn  = substr($hms,2,2);
      $sec = substr($hms,4);
      $lat = $item[3]; $NS = $item[4];
      $lon = $item[5]; $EW = $item[6];
      $dmy = $item[9];
      $day = substr($dmy,0,2);
      $mon = substr($dmy,2,2);
      $yr  = substr($dmy,4,2) + 2000;

      $ss  = ((3600 * $hr) + 60 * $mn ) + $sec;

#   manipulate time string: add 10 seconds just to see if the
#      Q330 ingests the time telegrams

#     $ss  = $ss + 10;

      $hr  = int( $ss / 3600 );
      $ss  = $ss - 3600*$hr;

      $mn  = int( $ss / 60 );
      $ss  = $ss - 60*$mn;

      $now = sprintf("%02d%02d%02d.00",$hr,$mn,$ss);

      $gpzda = sprintf("GPZDA,%s,%02d,%02d,%4d,00,00",$now,$day,$mon,$yr);
      printf("\$%s*%02X\n",$gpzda,&CHECKSUM($gpzda));

      $gpgga = sprintf("GPGGA,,%s,%s,%s,%s,1,08,,665.9,M,,M,,",$lat,$NS,$lon,$EW);
      printf("\$%s*%02X\n",$gpgga,&CHECKSUM($gpgga));
#
#  Polaris and Big Dipper
#
      $gpgsa =sprintf("GPGSA,A,1,2,3,4,5,6,7,8,9,,,,,,,");
      printf("\$%s*%02X\n",$gpgsa,&CHECKSUM($gpgsa));

      $gpgsv =sprintf("GPGSV,3,1,9,01,47,0,45,02,61,45,26,03,62,56,10,04,54,59,26");
      printf("\$%s*%02X\n",$gpgsv,&CHECKSUM($gpgsv));

      $gpgsv =sprintf("GPGSV,3,2,9,05,52,52,26,06,46,51,26,07,42,50,26,08,37,54,26");
      printf("\$%s*%02X\n",$gpgsv,&CHECKSUM($gpgsv));

      $gpgsv =sprintf("GPGSV,3,3,9,09,62,56,26");
      printf("\$%s*%02X",$gpgsv,&CHECKSUM($gpgsv));

      &flush(STDOUT);

     }
#
#  subroutine to compute checksum of GPS telegram
#
sub CHECKSUM {
```

```
        local($string) = $_[0];
        local($len, $i, $offset, $a);

        $chksum = 0;
        $len    = length($string);

        for ( $i = 1 ; $i <= $len ; $i++ )
          {
            $offset = $i - 1;
            $a = vec($string, $offset, 8);
            $chksum = $chksum ^ $a;   #  bitwise XOR operator
          }
        return ($chksum);
       }
#------------------end----------------------------------------#
```