DART Data Formats
and
DART System Messages

Bangkok Thailand, August 27-28, 2007

Guy Urban
West Coast and Alaska Tsunami Warning Center
910 S. Felton Street
Palmer, Alaska, 99645, USA

DART Data Types

- D$0 Position
- D$1 Standard Hourly
- D$2 Tsunami Report Mode
- D$3 Tsunami Report Mode Hourly
- D$MI Meteorological Data
- D$4 Deployment Data
- BATT Battery Data

Message Type = D$0

- D$0 Position
- Description

  Position messages are transmitted once per day, giving the location of the surface buoy.
- Message Breakdown

  “D$0 NS Date Time lat_deg lat_min N/S lon_deg lon_min E/W * checksum”
- * = checksum delimiter

  checksum = exclusive OR of all characters preceding "*", hexadecimal
- Example Message
- D$0 11/15/2006 13:05:28 3214.2972 N 12041.3991 W* 46

Message Type = D$1

- D$1 Standard Hourly
- Description

Standard hourly data messages are transmitted four times a day in six hour bundles.

- Message Breakdown : <cr> = 0x0D

"<cr>D$1C/I Date Time batv1 batv2 batv3 ht1 ht2 ht3 ht4 tries * checksum"
"<cr>D$1C/I Date Time batv1 batv2 batv3 ht1 ht2 ht3 ht4 tries * checksum"
"<cr>D$1C/I Date Time batv1 batv2 batv3 ht1 ht2 ht3 ht4 tries * checksum"
"<cr>D$1C/I Date Time batv1 batv2 batv3 ht1 ht2 ht3 ht4 tries * checksum"
"<cr>D$1C/I Date Time batv1 batv2 batv3 ht1 ht2 ht3 ht4 tries * checksum"
"<cr>D$1C/I Date Time batv1 batv2 batv3 ht1 ht2 ht3 ht4 tries * checksum"
<cr> = 0x0D

- D$1 = message ID
- C/I = message status; C=corrupted, I=Intact
- Date = mm/dd/yyyy
- Time [UTC] = hr:mn:se
- batv1 = BPR battery voltage in 10ths of a volt, or error code
- batv2 = acoustic Modem DSP battery in 10th of a volt
- batv3 = acoustic Modem battery in 10th of a volt
- ht1 … ht4 = water column height in millimeters
- tries = number of tries to deliver BPR data (up to 3)
- * = checksum delimiter
- checksum = exclusive OR of all characters preceding "*", hexadecimal

Message Type = D$1

- Example Message

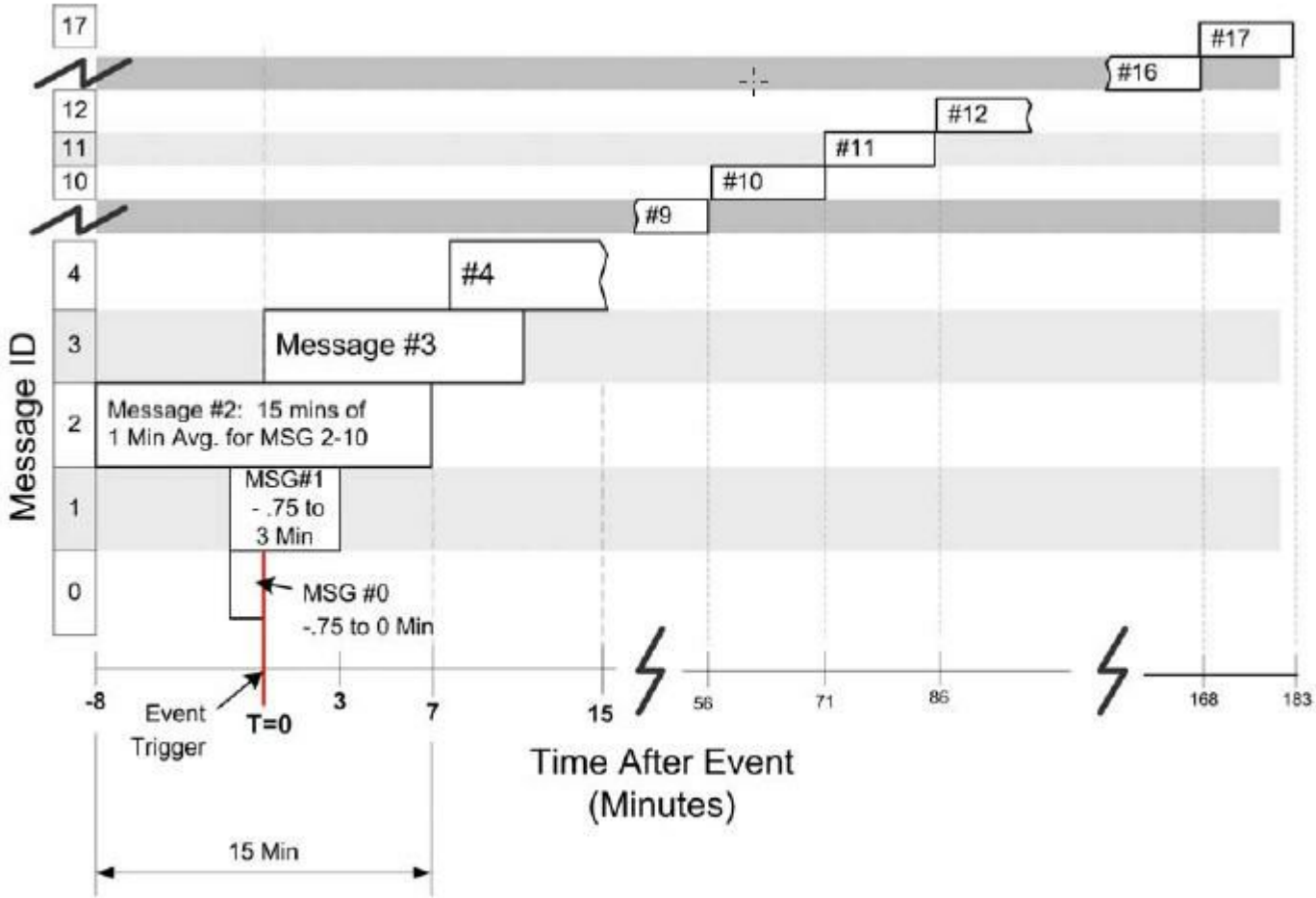
- D$1I 11/14/2006 18:15:00 1634146 3772376 3772344 3772313 3772294 1* 39
- D$1I 11/14/2006 19:15:00 1634146 3772275 3772262 3772251 3772249 1* 38
- D$1I 11/14/2006 20:15:00 1634146 3772249 3772257 3772271 3772293 1* 3E
- D$1I 11/14/2006 21:15:00 1634146 3772315 3772341 3772373 3772407 1* 39
- D$1I 11/14/2006 22:15:00 1634146 3772440 3772472 3772506 3772540 1* 3C
- D$1I 11/14/2006 23:15:00 1634146 3772572 3772603 3772631 3772657 1* 3B

Remember: water column height in millimeters

Message Type = D$2

D$2 Message Type has three (3) different formats

1. Message 00: four 15 second data points.
2. Message 01: sixteen 15 second data points.
3. Message 02 – end of event mode: sixteen 1 minute data points.



Timing diagram showing messages during Event Mode.

Message Type = D$2
Message Number 0

- D$2 Tsunami Report Mode

- Message Breakdown Message 0 "D$2C/I Msg# tt ttTime ts tsTime height dev1 dev2 dev3 tries * checksum"

- D$2 = message ID
- C/I = message status; C=corrupted, I=Intact
- Msg# = Sequential message number; this message is #00 of event mode
- tt = tsunami trigger designation
- ttTime = time tsunami detected; corresponds with 2nd elevated sample
- (observation #4)
- ts = time stamp designation
- tsTime = time of first data sample in message
- height = water column height in millimeters,
- dev1 = deviation from ht1 in millimeters, 4 hexadecimal characters
- dev2 = deviation from ht1 in millimeters, 4 hexadecimal characters
- dev3 = deviation from ht1 in millimeters, 4 hexadecimal characters
- tries = number of tries to deliver BPR data (up to 3)
- * = checksum delimiter
- checksum = exclusive OR of all characters preceding "*", hexadecimal

Message Type = D$2
Message Number 0

- Example Message

D$2I 00 tt 18:32:45 ts 18:32:00 3772311
00000063006201* 22

Time of Tsunami Trigger = 18:32:45
First Data Point Time Stamp (ts) = 18:32:00
Height 0 = 3772311 at ts
Height 1 = 3772311 + 0000 (0000H) ts + 15 seconds
Height 2 = 3772311 + 0099 (0063H) ts + 30 seconds
Height 3 = 3772311 + 0098 (0062H) at tt (event trigger) = ts + 45

Message Type = D$2
Message Number 1

- D$2 Tsunami Report Mode: Message #01
- Message Breakdown Message 1

D$2C/I Msg# tt ttTime ts tsTime height dev1 dev2 dev3 … dev15 tries * checksum"

Note: All data in message #1 (D$2C/I 01) are values integrated over the base sampling period of 15-seconds. The first four observations are repeats of message #0.

D$2 = message ID
C/I = message status; C=corrupted, I=Intact
Msg# = Sequential message number; this message is #01 of event mode
tt = tsunami trigger designation
ttTime = time tsunami detected; corresponds with 2nd elevated sample (observation #4)
ts = time stamp designation
tsTime = time of first data sample in message
height = water column height in millimeters, hexadecimal
<cr>dev1 = deviation from ht1 in millimeters, 4 hexadecimal characters
dev2 = deviation from ht1 in millimeters, 4 hexadecimal characters
…
…
dev15 = deviation from ht1 in millimeters, 4 hexadecimal characters
tries = number of tries to deliver BPR data (up to 3)

Message Type = D$2
Message Number 1

- Example Message

D$2I 01 tt 18:32:45 ts 18:32:00 3772311
0000006300620069006000061005f005ffffafff9fff8fff8fff7fff6fff401* 21

Time of Tsunami Trigger = 18:32:45
First Data Point Time Stamp (ts) = 18:32:00
Height 0 = 3772311 at ts
Height 1 = 3772311 + 0000 (0000H) ts + 15 seconds
Height 2 = 3772311 + 0099 (0063H) ts + 30 seconds
Height 3 = 3772311 + 0098 (0062H) at tt (event trigger) = ts + 45
Height 4 = 3772311 + 0105 (0069H) at ts + 60
…
Height 14 = 3772311 + (65526 – 65535) (fff6H) ts + 3m 30s
Height 15 = 3772311 + (65524 – 65535) (fff4H) ts + 3m 45s (225s)

Message Type = D$2
Message Number 2 – end of event mode

- Description

All D$2 data in messages 02 until end of event mode are 1-minute averages of four 15-second samples.

- Message Breakdown

"<cr>D$2C/I Msg# tt ttTime ts tsTime height dev1 dev2 dev3 … dev15 tries * checksum"

- D$2 = message ID

C/I = message status; C=corrupted, I=Intact
Msg# = Sequential message number beginning with #02
tt = tsunami trigger designation
ttTime = time tsunami detected; corresponds with 2nd elevated sample (observation #4)
ts = time stamp designation
tsTime = time of first data sample in message
height = water column height in millimeters, hexadecimal
<cr>dev1 = deviation from ht1 in millimeters, 4 hexadecimal characters
dev2 = deviation from ht1 in millimeters, 4 hexadecimal characters
…
…
dev15 = deviation from ht1 in millimeters, 4 hexadecimal characters
tries = number of tries to deliver BPR data (up to 3)

Message Type = D$2
Message Number 2 – end of event mode

- Example Message

D$2I 02 tt 18:32:45 ts 18:32:00 3772335
fffdfffafff7fff5fff1ffeeffea00190048ffe1ffddffdaffd8ffd5ffd101* 21

Time of Tsunami Trigger = 18:32:45
First Data Point Time Stamp (ts) = 18:32:00
Height 0 = 3772335 at ts = 18:32
Height 1 = 3772335 - 0002 (fffdH) at ts + 1 minute = 18:33
Height 2 = 3772335 - 0005 (fffaH) at ts + 2 minute = 18:34
…..
Height 14 = 3772335 - 0046 (ffd1H) at ts + 15 minutes = 18:47

Message Type = D$3

- D$3 Tsunami Report Mode Hourly
- Description

D$3 messages are reported on an hourly schedule.

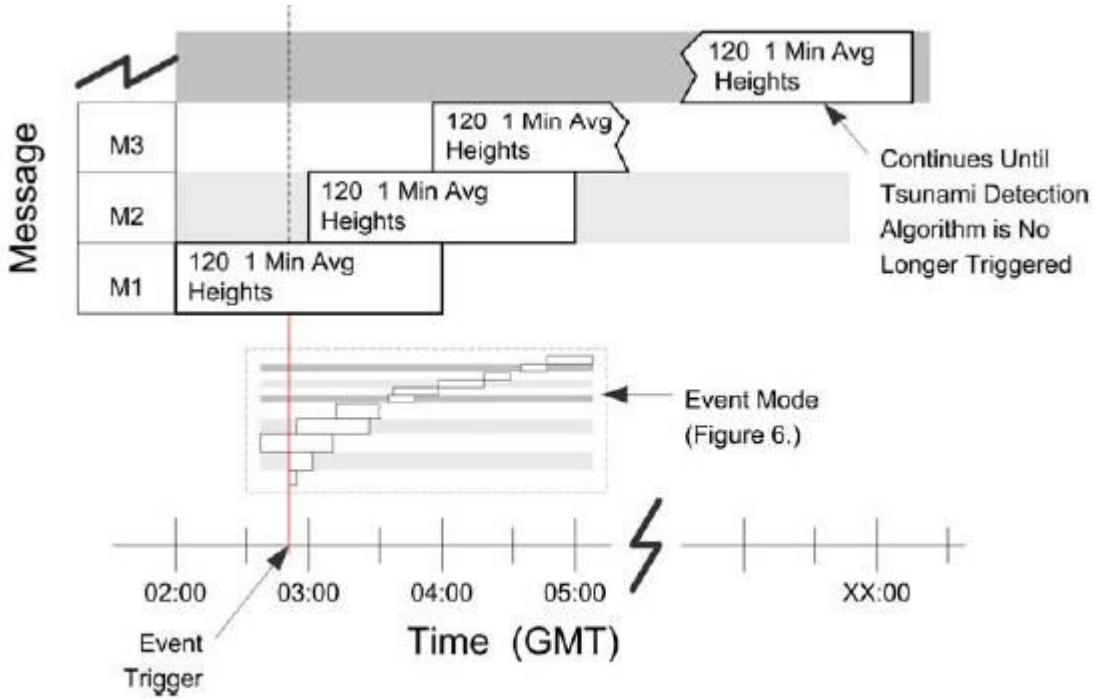
They are interspersed with D$2 messages while a unit is reporting in Event Mode.

120 one minute samples are reported;

60 previously reported samples provide redundancy and

60 ‘new’ samples are reported in each message.

- The 1-minute value time stamp ALWAYS corresponds with xx:xx:00.



Timing diagram during Extended-Reporting Mode

Message Type = D$3

- Message Breakdown

"<cr>D$3C/Itt ttTime ts tsTime height
<cr>dev1 dev2 dev3 … dev24
<cr>dev25 … dev48
<cr>dev49 … dev72
<cr>dev73 … dev96
<cr>dev97 … dev119 tries * checksum"

D$2 = message ID
C/I = message status; C=corrupted, I=Intact
tt = tsunami trigger designation
ttTime = time tsunami detected; corresponds with 2nd elevated sample (observation #4)
ts = time stamp designation
tsTime = time of first data sample in message
height = water column height in millimeters, hexadecimal
<cr>dev1 .. dev24 = deviation from ht1 in millimeters, 4 hexadecimal characters
<cr>dev25 .. dev48 = deviation from ht1 in millimeters, 4 hexadecimal characters
<cr>dev49 .. dev72 = deviation from ht1 in millimeters, 4 hexadecimal characters
<cr>dev73 .. dev96 = deviation from ht1 in millimeters, 4 hexadecimal characters
<cr>dev97 .. dev119 = deviation from ht1 in millimeters, 4 hexadecimal characters
tries = number of tries to deliver BPR data (up to 3)

USAID | ASIA
FROM THE AMERICAN PEOPLE

U.S. Contribution to the Indian Ocean Tsunami Warning System

NOAA

Message Type = D$3

- Example Message

D$3Itt 18:32:45 ts 17:00:00 3772621
fffdfffbfff6fff3fff0ffecffe9ffe6ffe3ffdfffddffd9ffd6ffd3ffcfffccffc9ffc5ffc2ffbfffbdffbaffb7ffb4
ffb1ffaeffabffa9ffa5ffa2ff9effcdfffcff95ff91ff8eff8cff89ff85ff84ff81ff7fff7cff7aff78ff75ff73ff6f
ff6cff68ff66ff63ff60ff5eff5bff58ff55ff52ff50ff4dff4aff49ff46ff42ff41ff3dff3bff38ff36ff34ff32ff2f
ff2dff2bff29ff28ff25ff23ff21ff1eff1cff1aff18ff17ff14ff12ff10ff0dff0dff0bff0aff07ff06ff04ff03ff00
fefefefdfefbfef9fef8fef5fef5fef3fef1fef0feeefeeefeedfeeafee9fee8fee6fee4fee3fee2fee1fedffedf01* 3D

Message Type = BATT

- Message Breakdown

BATT date time int
cpu acMo mode

<cr> = 0x0D
<cr>BATT = message ID
Date = mm/dd/yyyy
Time [UTC] = Time of sample: hr:mn:se
Int = Sampling Interval
cpu = central processing unit battery voltage at time of sample
acMo = acoustic modem battery voltage at time of sample
mode = system operating mode

- Example Message

BATT 05/21/2007 13:21:00 24:00:00
14.50 27.1 STANDARD MODE

Decoding the DART Data

- SXXX46 KWBC 231813

- DDDDDDD0 174181308
- D$1I 06/23/2007 12:15:00 1424141 4709912 4709891 4709877 4709863 1* 39
- D$1I 06/23/2007 13:15:00 1424141 4709851 4709843 4709837 4709829 1* 02
- D$1I 06/23/2007 14:15:00 1424141 4709830 4709829 4709834 4709836 1* 30
- D$1I 06/23/2007 15:15:00 1424141 4709843 4709854 4709871 4709884 1* 04
- D$1I 06/23/2007 16:15:00 1424141 4709898 4709919 4709941 4709957 1* 37
- D$1I 06/23/2007 17:15:00 1424141 4709976 4709999 4710019 4710039 1* 09 00-0NN 00E

Decoding the DART Data

Looking for Station (PlatformID) and the Julian Day and Time Stamp

```
countchar += 22;   // 22 characters from header

strncpy(PlatformID,&FileDataIn[countchar],8);

printf("platform id is %s\n",PlatformID);

        countchar += 9;
   strncpy(JDay,&FileDataIn[countchar],3);

   countchar += 3;
   strncpy(HR,&FileDataIn[countchar],2);

   countchar += 2;
   strncpy(MN,&FileDataIn[countchar],2);

   countchar += 2;
   strncpy(SC,&FileDataIn[countchar],2);
```

Decoding the DART Data
Looking for the Message Type

```
Dart2Switch = 0;
Dart2TriggerSwitch = 0;
Dart2HourSwitch = 0;

while (countchar < stringlength)
   {
      strncpy(MT,&FileDataIn[countchar],4);
       if (strncmp(MT,"D$1",3)==0)
              Dart2Switch = 1;
         if (strncmp(MT,"D$2",3)==0)
             Dart2TriggerSwitch = 1;
          if (strncmp(MT,"D$3",3)==0)
             Dart2HourSwitch = 1;
           countchar++;
   }  // end while
```

Decoding the DART Data
Looking for the Message Type

```
Dart2Count = 0;

 if (Dart2Switch == 1)
{
  for(n=0;n<Counter1;n++)
  {
    strncpy(MT,&FileDataIn[n],4);
  if (strncmp(MT,"D$1I",4)==0)
  {
    DartDataAt[Dart2Count] = n;
  Dart2Count++;
  }
  }
}
```

Decoding the DART D$1 Data
Counting the Number of Samples

- D$1I 06/23/2007 12:15:00 1424141 4709912 4709891 4709877 4709863 1* 39
- D$1I 06/23/2007 13:15:00 1424141 4709851 4709843 4709837 4709829 1* 02
- D$1I 06/23/2007 14:15:00 1424141 4709830 4709829 4709834 4709836 1* 30
- D$1I 06/23/2007 15:15:00 1424141 4709843 4709854 4709871 4709884 1* 04
- D$1I 06/23/2007 16:15:00 1424141 4709898 4709919 4709941 4709957 1* 37
- D$1I 06/23/2007 17:15:00 1424141 4709976 4709999 4710019 4710039 1* 09 00-0NN 00E

```
Dart2Count = 0;

  if (Dart2Switch == 1)
  {
     for(n=0;n<Counter1;n++)
     {
         strncpy(MT,&FileDataIn[n],4);
           if (strncmp(MT,"D$1I",4)==0)
             {
                DartDataAt[Dart2Count] = n;
          Dart2Count++;
         }
     }
  }
```

- There may be six lines of data or there could be more or less

Getting the Time and Height Data

```
if (Dart2Switch == 1)
                {
                  StartHeight = 0;
  for (n=0;n<Dart2Count;n++)
  {
                    if ((strncmp(PlatformID,"1404B7B8",8)==0) || (strncmp(PlatformID,"1404E7C4",8)==0))
              countchar = (DartDataAt[n] + 5);
                 else
                  countchar = (DartDataAt[n] + 5);

            memset(STempMonth,0x00,3);
              memset(STempDay,0x00,3);
              memset(STempYear,0x00,5);

              strncpy(STempMonth,&FileDataIn[countchar],2);
              STempMonth[2] = 0x00;
                   countchar += 3;
              strncpy(STempDay,&FileDataIn[countchar],2);
             STempDay[2] = 0x00;
              countchar += 3;
            strncpy(STempYear,&FileDataIn[countchar],4);
              STempDay[4] = 0x00;

              TempMonth[n] = atoi(STempMonth);
                   TempDay[n] = atoi(STempDay);
                   TempYear[n] = atoi(STempYear);
                   TempDartDay[n]=DartDateToDay(TempYear[n],TempMonth[n],TempDay[n]);
```

Getting the Time and Height Data (continued)

```
countchar = (DartDataAt[n] + 16);

 strncpy(TMHour,&FileDataIn[countchar],2);
DartTMHour[n]=atoi(TMHour);

 strncpy(teststring,&FileDataIn[countchar-4],10);

countchar = (DartDataAt[n] + 33);

for (m=0;m<4;m++)
{
        strncpy(TMHeight,&FileDataIn[countchar],7);
        DartHeight[4*n+m]=atof(TMHeight);
        countchar += 8;

if (DartTMHour[n]==23 && m==3)
        DartHour[4*n+m] = 0;
        else if (DartTMHour[n]<23 && m==3)
        DartHour[4*n+m] = DartTMHour[n] + 1;
        else
        DartHour[4*n+m] = DartTMHour[n];

         if (m==0)
           DartMin[4*n+m] = 15;
         else if (m==1)
           DartMin[4*n+m] = 30;
         else if (m==2)
           DartMin[4*n+m] = 45;
         else
           DartMin[4*n+m] = 0;

        if (atoi(JDay) == TempDartDay[n])
        Dartday[4*n+m] = atoi(JDay);
        else
        Dartday[4*n+m] = TempDartDay[n];

        if ((DartHour[4*n+m] == 0) && (DartMin[4*n+m] == 0))
        Dartday[4*n+m] = atoi(JDay);
        else
        Dartday[4*n+m] = TempDartDay[n];

jtime[4*n+m] = (float)(Dartday[4*n+m]) + (float)DartHour[4*n+m]/24 + (float)DartMin[4*n+m]/(24*60);

  } // for (m=0;m<4;m++)
} //  for (n=0;n<Dart2Count;n++)
```

Coding the DART D$2 Data

```
if (Dart2TriggerSwitch == 1)
{
    for (i=0;i<stringlength;i++)

                countchar = (DartTriggerDataAt + 5);
                strncpy(TMMsg,&FileDataIn[countchar],2);

        countchar += 6;
          strncpy(DetectTime,&FileDataIn[countchar],8);

                countchar += 12;
          strncpy(TMTime,&FileDataIn[countchar],8);

    countchar += 9;
          strncpy(TMHeight,&FileDataIn[countchar],7);

                StartHeight = atof(TMHeight);

    Deviation[0] = 0;

    MsgNumber = atoi(TMMsg);
```

Coding the DART D$2 Data
An Audio Alarm

```
if ((MsgNumber==0) || (MsgNumber==1))
{
        for (i=0;i<5;i++)
         PlaySound("dog.wav",NULL,SND_SYNC);
 }
```

Coding the DART D$2 Data
A Visual Alert

```
if (MsgNumber==0)
    {
     memset(darttempstring,0x00,6);

                                                    if (strncmp(PlatformID,"DDDDDD4C",8)==0)
    strncat(darttempstring, "32401",5);
                                  else if (strncmp(PlatformID,"DDDDDD4D",8)==0)
    strncat(darttempstring, "32401a",6);
                                  else if (strncmp(PlatformID,"DDDDDDD0",8)==0)
    strncat(darttempstring, "51407",5);
                                  else if (strncmp(PlatformID,"DDDDDDD2",8)==0)
    strncat(darttempstring, "51407a",6);
                                  else if (strncmp(PlatformID,"DDDDDD10",8)==0)
    strncat(darttempstring, "46403",5);
                                  else if (strncmp(PlatformID,"DDDDDD12",8)==0)
    strncat(darttempstring, "TDTEST",6);
                                  else if (strncmp(PlatformID,"DDDDDD28",8)==0)
    strncat(darttempstring, "TATEST",6);


     printf("DART 2 Buoy %s Triggered\n",darttempstring);

    }
```

Coding the DART D$2 Data
Getting the Deviations

```
countchar += 8;
 if (MsgNumber==0)
{
 for (i=1;i<4;i++)
 {
  Deviation[i] = 0;
  strncpy(TsuTemp,&FileDataIn[countchar],4);
   for (j=0;j<4;j++)
   {
            cvalue = (int)TsuTemp[j];
           tempvalue = hex_to_integer(cvalue);
             Deviation[i] += (double)(tempvalue*pow(16,3-j));
   }
              if (TsuTemp[0] == 'f')
                    Deviation[i] -= 65535;
                   countchar += 4;
    } // end for
 }
   else if (MsgNumber > 0)
    {
        for (i=1;i<16;i++)
        {
                    Deviation[i] = 0;
                    strncpy(TsuTemp,&FileDataIn[countchar],4);
                    for (j=0;j<4;j++)
                    {
             cvalue = (int)TsuTemp[j];

                     tempvalue = hex_to_integer(cvalue);

                   Deviation[i] += (double)(tempvalue*pow(16,3-j));
                    }
                  if (TsuTemp[0] == 'f')
                  Deviation[i] -= 65535;
                  countchar += 4;
        } // end for
 } // end if else
```

Coding the DART D$2 Data
Converting Time Info to Numeric Values

```
strncpy(stemp,&DetectTime[0],1);
strncat(stemp,&DetectTime[1],1);
detecthour = atoi(stemp);
 memset(stemp,0x00,3);

strncpy(stemp,&DetectTime[3],1);
strncat(stemp,&DetectTime[4],1);
detectmin = atoi(stemp);
 memset(stemp,0x00,3);

strncpy(stemp,&DetectTime[6],1);
strncat(stemp,&DetectTime[7],1);
detectsec = atoi(stemp);

 memset(stemp,0x00,3);

  strncpy(stemp,&TMTime[0],1);
strncat(stemp,&TMTime[1],1);
tsunamihour = atof(stemp);

  memset(stemp,0x00,3);
strncpy(stemp,&TMTime[3],1);
strncat(stemp,&TMTime[4],1);
tsunamimin = atof(stemp);

        memset(stemp,0x00,3);
strncpy(stemp,&TMTime[6],1);
strncat(stemp,&TMTime[7],1);
tsunamisec = atof(stemp);

 tsunamiday = atof(JDay);
 if (detecthour == 0 && tsunamihour == 23 &&    (MsgNumber == 0 || MsgNumber == 1 || MsgNumber == 2))
    tsunamiday -= 1;
```

Coding the DART D$2 Data
Getting Time and Height Values

```
                    if (MsgNumber==0)
                    {
                     for (i=0;i<4;i++)
                jtime[i] = tsunamiday + tsunamihour/24 + tsunamimin/(60*24)
                        + (tsunamisec + (float)(i*15))/(60*60*24); //count by 15 secs
                height[i] = StartHeight + Deviation[i];
                    }

                    if (MsgNumber > 0)
                    {
                     if (MsgNumber==1)
                {
                  for (i=0;i<16;i++)
                    jtime[i] = tsunamiday + tsunamihour/24 + tsunamimin/(60*24)
                                          + (tsunamisec + (float)(i*15))/(60*60*24); //count by 15 secs
                    height[i] = StartHeight + Deviation[i];

                }
                     if (MsgNumber!=1)
                {

                      for (i=0;i<16;i++)
                      jtime[i] = tsunamiday + tsunamihour/24
                                           + (tsunamimin + (float)i)/(60*24); // if i>1 1 minute increments
                    height[i] = StartHeight + Deviation[i];

                     }  // end if
                    } // end if (MsgNumber > 0)

        }  //end   if (Dart2TriggerSwitch == 1)
```

Coding the DART D$2 Data
Including alarms

```
 if ((MsgNumber==0) || (MsgNumber==1)) //|| (MsgNumber==2))
{
          for (i=0;i<5;i++)
             PlaySound("dog.wav",NULL,SND_SYNC);
 }

  if (MsgNumber==0)
   {
           stream3 = fopen(PAGER,"w");
           if (stream3 != NULL)                    // JMC 12/29/2006
            {
               fprintf(stream3,"DART 2 Buoy Triggered\n");
                fclose(stream3);
           }
          stream3 = fopen("p:\\seismic\\messages\\pagealarm.txt","w");
         if (stream3 != NULL)                     // JMC 12/29/2006
         {
            fprintf(stream3,"DART 2 Buoy %s Triggered\n",darttempstring);
       fclose(stream3);
           }
   }  // end  if
```

Coding the DART D$3 Data

The hourly data is done just like the D$2 trigger data, but is done for 119 deviation values.

```
for (i=1;i<120;i++)
 {
    Deviation[i] = 0;
    strncpy(TsuTemp,&FileDataIn[countchar],4);
    for (j=0;j<4;j++)
    {
        cvalue = (int)TsuTemp[j];
        tempvalue = hex_to_integer(cvalue);
         Deviation[i] += (double)(tempvalue*pow(16,3-j));
    }
   if (TsuTemp[0] == 'f')
      Deviation[i] -= 65535;
       countchar += 4;
    } // end for

Deviation[0] = 0;
for (i=0;i<120;i++)
{
   jtime[i] = tsunamiday + tsunamihour/24+ (tsunamimin + (float)i)/(60*24); // if i>0 1 minute increments
   height[i] = StartHeight + Deviation[i];
}
```

References

- *Tsunami Observation and Data Management, Appendix D: Data Formats*; NOAA, National Data Buoy Center NDBC), Stennis Space Center, MS, USA
- *Description of Real-time DART System Messages:* Revision # 2.01*;* NOAA, Pacific Marine Environmental Laboratory (PMEL); Engineering Development Division; Seattle, WA, USA
- *Socketport.*c; G. Urban and J. Carrick; West Coast and Alaska Tsunami Warning Center, Palmer, AK, USA

The End