ODYSSEY SALINITY / TEMP DATA LOGGER

Note that this only covers the trace mode as I have a lot more workto do on other modes of operation.

-------------------------------------------------------------------------------------------------------------------------------------

Trace Mode:

Baud Rate              4800

Parity                     None

Data Bits                8

Stop Bits                1

Input Buffer Size      32000

1. Software

To wake up the logger send a train of pulses i.e 0x01 01 01 01 01 01 01.

2. Logger

When the logger detects the above pulses it responds with the ascii string

'Hi!' (0x48, 69, 21).

3. Software

The software then confirms logger response and sends String 'T' (0x54)

followed by the ID number of the logger. For the conductivity logger the ID

is 0x1A (Decimal 26).

The software then needs to go into a state waiting for the logger to send

serial data.

4. Logger

Sends values to the software in what appears to be a constant data stream.

The software counts in each byte which is processed and displayed (see raw

data processing below).

If the logger reaches 64 readings it's sends a stop command to the software

(stop command TBD).

Raw Data Processing:

The logger sends data to the software representing the environment it is

monitoring.

This data is processed in such a way as to make the maximum value capable of

being recorded = 65535

Example:

Logger sends 0x09 13.

Binary conversion =  00001001   00010011

Decimal Conversion: MSB  = 9   LSB = 19

Software Conversion = MSB + (LSBx256)

i.e. 9 + (19 x 256) = (9 + 4864) = 4873

To exit the trace mode repeat steps 1 & 2 above. The logger will again

respond with the ascii string 'Hi!' (0x48, 69, 21) to confirm it has

stopped.

-------------------------------------------------------------------------------------------------------------------------------------

Hi Daniel,

Yes, we can retrieve the calibration figures if we have the serial numbers.

Engineer

You can get the software to download the data from our website:

<http://odysseydatarecording.com/index.php?route=product/category&path=66>

The manual is also available:

<http://odysseydatarecording.com/index.php?route=information/information&information_id=9>

You will also need the calibration files for the units, did they come with

the loggers?

Regards

Richard

Engineer

With reference to the photo supplied:

Green/White = Ground

Blue /White = Transmit (Tx)

Black/White = Recieve (Rx)

Hi Daniel,

Great to hear you are still working on the SHED project. I'm still doing sensor related work but more on the server side infrastructure. Our marine related work (in aquaculture) has stepped up one cost level and we now use embedded Linux and various high cost sensors like Tyco EC250 etc.

Your experience with Odyssey sensors matches exactly with ours. Their tech support is less than helpful. You are right we did reverse engineer the sensor communication. I believe we basically replay communication we captured from the windows software. I do remember thinking is was a bit odd.

Below is a code snippet that may help. This is basically replicating what we saw in our data capture. You should be able to port that across to what ever platform you are using. The last part about sending 256 0x01 bytes is what we saw in the capture. We were never quite sure why it was there! maybe some kind of reset mechanism.

Anyway, hopefully that helps.

Cheers

Chris

//write 0x01 to serial

  odyssey\_buf[0]=0x01;

  i = fos\_serial\_write(odyssey\_serial, odyssey\_buf, 1);

  if (i != 1) goto fail;

  // Brief pause of 0.8 second...

  fos\_vartimer\_sleep(800);

  //write 0x54, 0x1a to serial...

  odyssey\_buf[0]=0x54;

  i = fos\_serial\_write(odyssey\_serial, odyssey\_buf, 1);

  if (i != 1) goto fail;

  odyssey\_buf[0]=0x1a;

  i = fos\_serial\_write(odyssey\_serial, odyssey\_buf, 1);

  if (i != 1) goto fail;

  //next read 4 serial bytes or timeout...

  //first erase the odyssey\_buffer, flush the serial port ...

  for(i=0;i<4;i++)

    odyssey\_buf[i]=0;

  fos\_serial\_flush (odyssey\_serial);

  if(fos\_serial\_timed\_read(odyssey\_serial,odyssey\_buf,4,4)==4)

  {

    \*temperature = (uint16\_t)(odyssey\_buf[0]) | (((uint16\_t)(odyssey\_buf[1]))<<8);

    \*conductivity = (uint16\_t)(odyssey\_buf[2]) | (((uint16\_t)(odyssey\_buf[3]))<<8);

    success = FOS\_TRUE;

  } else {

    success = FOS\_FALSE;

  }

  //write 0x01 to serial about 256 times...

  for(i=0;i<4;i++)

    odyssey\_buf[i]=0x01;

  for(i=0;i<64;i++)

  {

    fos\_serial\_write(odyssey\_serial, odyssey\_buf, 4);

  }