Subsurface Data Acquisition in the North Sea

Max Schlereth

Thesis submitted to the Faculty of the
University of Washington
in partial fulfillment of the requirements for the degree of

Master of Science in Applied Physics

May 16th, 2014
Seattle, WA

Subsurface Data Acquisition in the North Sea

Abstract

From February to June of 2012 I had the privilege of participating in the development a telemetry system for a hydroplow cable burial machine. After completing my contribution to the development of this system I spent two months operating it in the North Sea on a cable laying vessel. I have used this experience in development and field work to prepare a Master's Thesis focusing on data acquisition in a submarine environment. While this telemetry system was specifically designed for cable burial, the techniques described in this paper can be generalized to many scientific and engineering applications.

The purpose of this thesis is to examine the physics of how sensors work in the context of the marine environment as well as review the interfacing electronics and software necessary to acquire this data. I will first provide a brief overview of the physics of the ocean and focus in on the actual practice of obtaining meaningful subsurface measurements. I will pay special attention to the use of acoustic and inductive instruments.

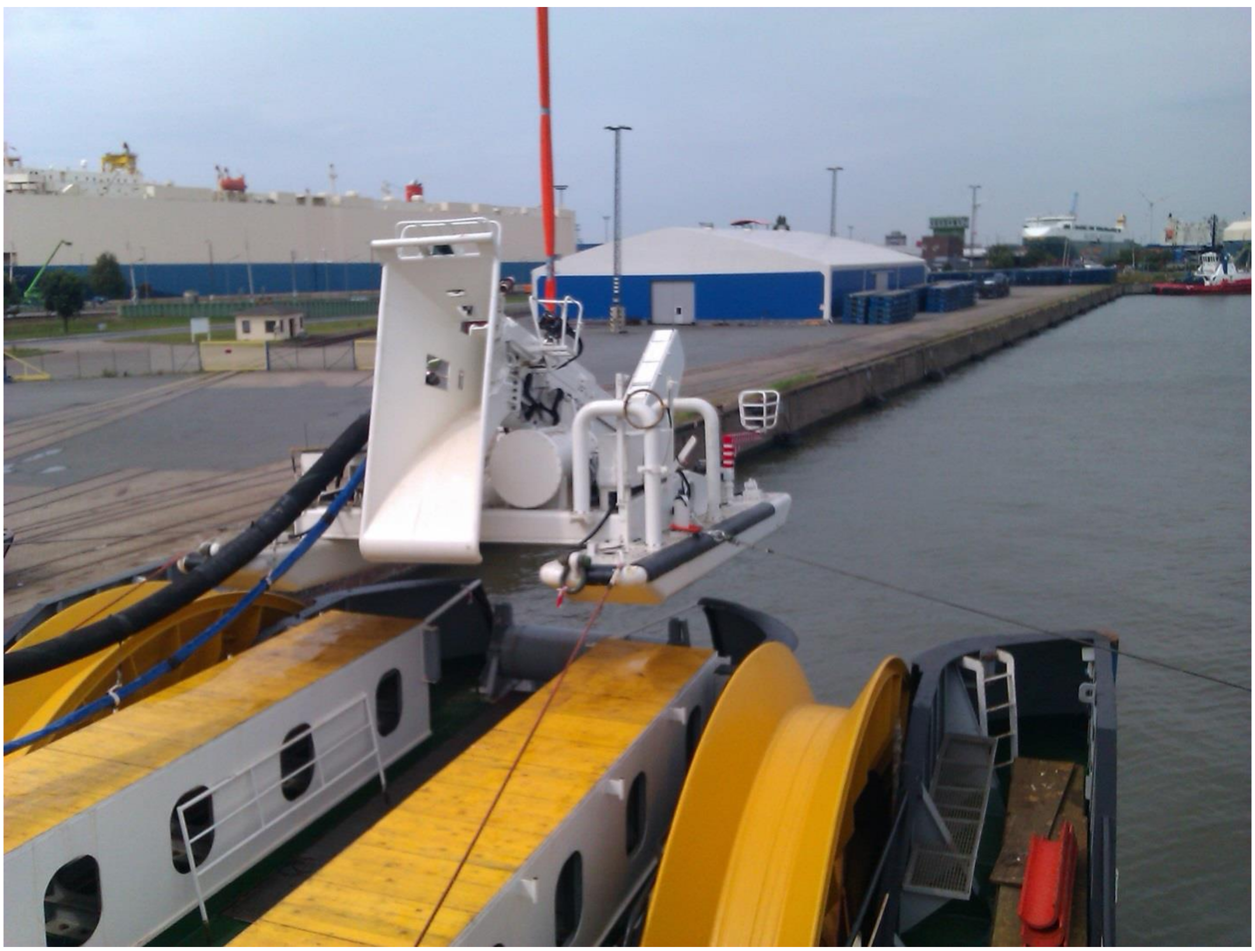

Figure 1

Contents

1 Physical Marine Environment

1.1 Overview

1.1.1 Surface interface, water column, seafloor

1.1.2 Salinity, Temperature, Pressure, Viscosity

1.1.3 Geological features of the seafloor

1.2 Electromagnetic fields, Optics, Scattering

1.2.1 Conductivity and depth

1.2.2 Ions per volume, salinity

1.2.3 Mobility of ions, dependence on pressure, temperature

1.2.4 Electric and magnetic fields from motional induction and earth's field

1.2.5 Index of refraction, use of optical instruments

1.3 Acoustical Oceanography

1.3.1 Characteristic acoustic impedance

1.3.2 Speed of sound in water

1.3.3 Equations (see CC Leroy 1969)

1.3.4 Graphic display, speed vs temperature, speed vs depth

2 Hydroplow and Instrumentation

2.1 Sonar Transducer

2.1.1 Piezoelectricity, electrostriction, magnetostriction

2.1.2 Altimeter, transmitter and processing circuit

2.1.4 Signal processing and sources of noise

2.3 Inductive Sensors

2.3.1 Heading, Pitch, Roll

2.3.2 Geomagnetic field, magneto inductive displacement

2.3.3 Errors due to geographic location and altitude

2.3.4 Magnetic field distortions

3 Data Processing

3.1 Electronics

3.2 Embedded Software

3.3 GUI

4 Conclusions and Future Work

4.1 Conclusions

4.2 Future Work

4.3 Addendum

References

Appendix One Embedded System Source Code

Appendix Two List of Figures

Appendix Three List of Tables

Chapter 1

Physical Marine Environment

1.1 Overview

To begin I will provide a very basic overview of ocean physics in order to introduce terminology and provide a theoretical framework for later technical material. It will also provide specific physical information on the southern region of the North Sea near the German coast where I operated the data acquisition techniques to be discussed herein.

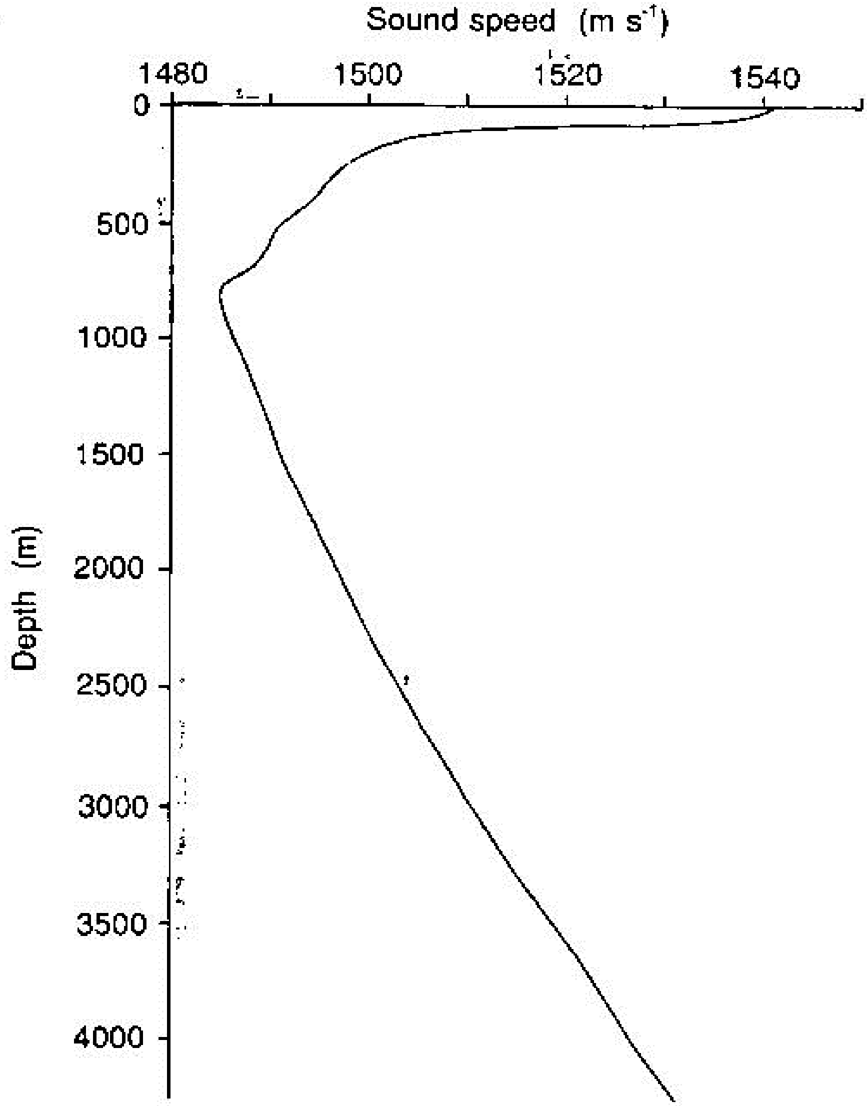

Figure 2

A water column profile describes a cross section of the ocean that allows us to identify the vertical distribution of physical properties such as temperature, salinity and pressure. This information is critical to understanding the structure of water masses and how that affects the process of collecting data. Please see **Figure 2** for an example of a water column profile illustrating speed of sound to depth.

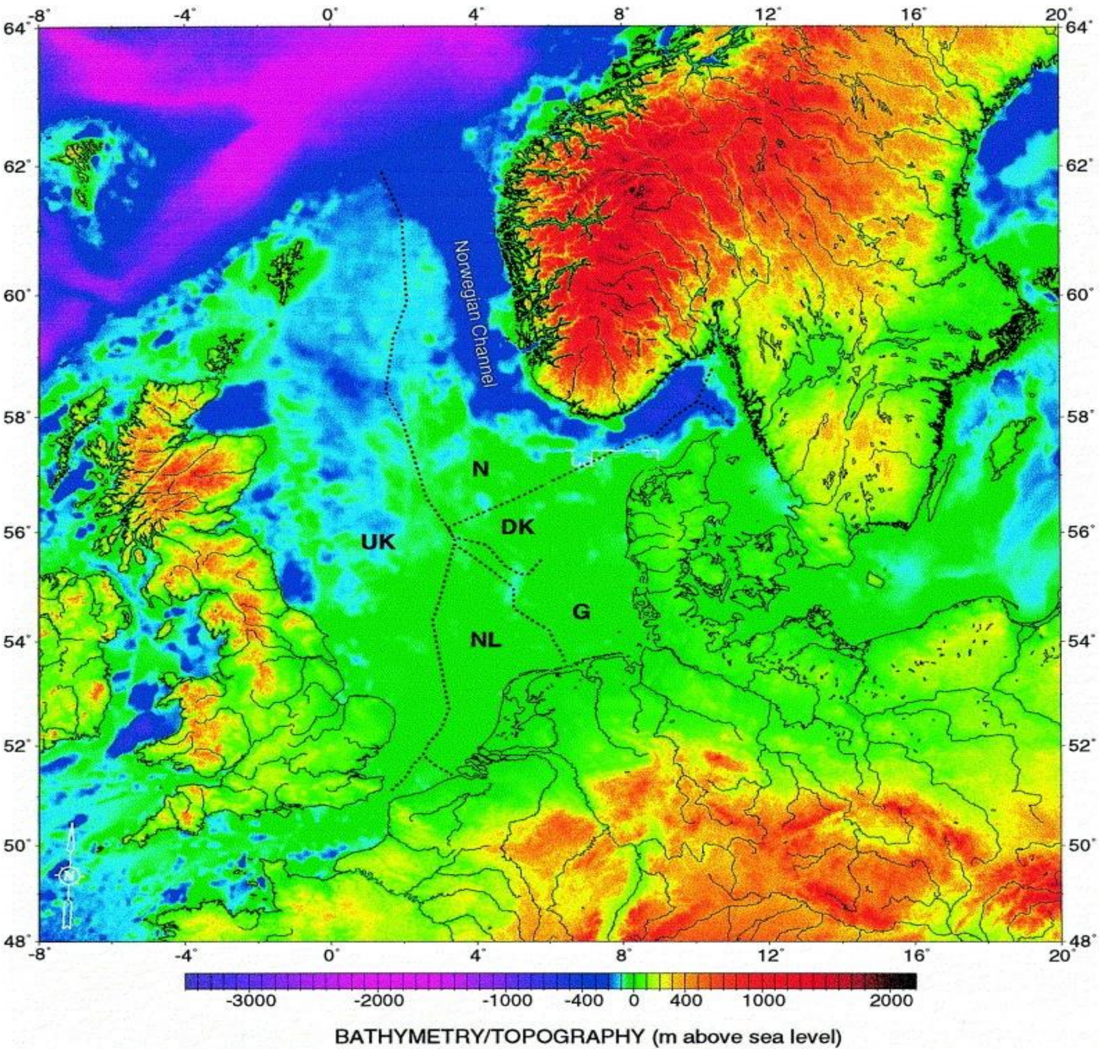

Figure 3

The southern shelf of the North Sea is relatively shallow with depths of 20 m and less being common. Incident solar radiation heats the surface providing thermal energy. Due to absorption and scattering, the total incident radiant energy attenuates to half its value with in the first half meter of water. [3] Given the shallow depth and constant mixing by turbulent waves, the thermal distribution is relatively uniform along the water column. Descending down the column pressure increases and conductivity and salinity fluctuate. The seafloor of this region, just north of the Wadden Sea is primarily soft mud with clay deposits. Strong currents sweep the sea floor preventing a static buildup of sediment.

1.2 Electromagnetic Fields and Optics

1.2.1 Conductivity

The conductivity of seawater is a function of the density and mobility of charge carriers. The charge carriers are ions produced from dissolved salts, i.e. a product of salinity. Temperature and pressure affect the mobility of these ions. Higher temperatures allow for greater ion mobility, while greater pressure leads to increased density. To compare the effects that these properties have on conductivity, consider that an increase in salinity by 1 gram per liter, a depth increase of 2000 m, or a temperature increase of 1^0C would each individually increase conductivity by roughly the same value. [4]

In general, temperature is the dominant contributing factor in conductivity levels of seawater. Deep sea conditions can yield a value around 2.5 S/m, while 6 S/m is common in shallow warm water. [4] The shallow depths and instability of the water column in the southern North Sea make profiling the conductivity very difficult.

1.2.2 Motional Induction

The ions produced from dissolved salt are pushed by currents through the Earth's magnetic field. Lorenz force deflects cations and anions in opposite directions causing a separation of charge. The resulting motional EMF then drives an electric current which results in additional magnetic fields. [4] Electronics must be carefully designed to operate in this complex inductive environment.

1.2.3 Optics

Ignoring the effects of biological populations and suspended sediment, the optical properties of pure seawater are remarkably uniform and depend weakly on salinity, temperature, wavelength and depth. [3] To an extent this simplifies the use of cameras and other optical instruments as the need to calibrate for different depths, temperatures etc., is not necessary. However, getting good resolution is chiefly limited by sediment and lack of light. Introducing electrical lighting and video camera equipment into this environment introduce a number of technical challenges which we will discuss later.

1.3 Acoustical Oceanography

Unlike electromagnetic and optical waves, acoustic waves propagate through the ocean extremely well. At low frequencies, sound waves can transmit across an entire ocean basin before they attenuate below background noise. Acoustical waves are longitudinally polarized, meaning that their displacements are parallel to their wave vectors. [3] Just below the surface of the ocean, sound travels at around 1500 m s^{-1}. [1] Variations in the speed of sound due to temperature, salinity and current variations result in gradual refraction which must be taken into account over significant distances.

Analytically, the speed of sound can be represented as a function of salinity, temperature, and depth. One such equation given below is believed to be accurate to within .2 m s^{-2}.

$$c = c(s,T,z) = c_0 + \alpha_0(T - 10) + \beta_0(T - 10)^2 + \gamma_0(T - 18)^2 + \delta_0(s - 35) + \epsilon_0(T - 18)(s - 35) + \zeta_0|z|$$

Here c, the speed of sound in water, depends linearly on salinity and quadratically on temperature. Greater depth increases speed, and its influence grows as pressure increases. In the shallow waters of the North Sea however, temperature is clearly the dominant factor.

Figure 4 is a sound speed profile based on data taken in the equatorial Atlantic in 1985 from the RSS Charles Darwin. [4] Although this is far from our environment of interest, it is a good illustration of the relationships just discussed. Notice in the top several hundreds of meters the strong dependence on temperature.

Below, **Figure 4** illustrates variations in temperature, salinity and the speed of sound.

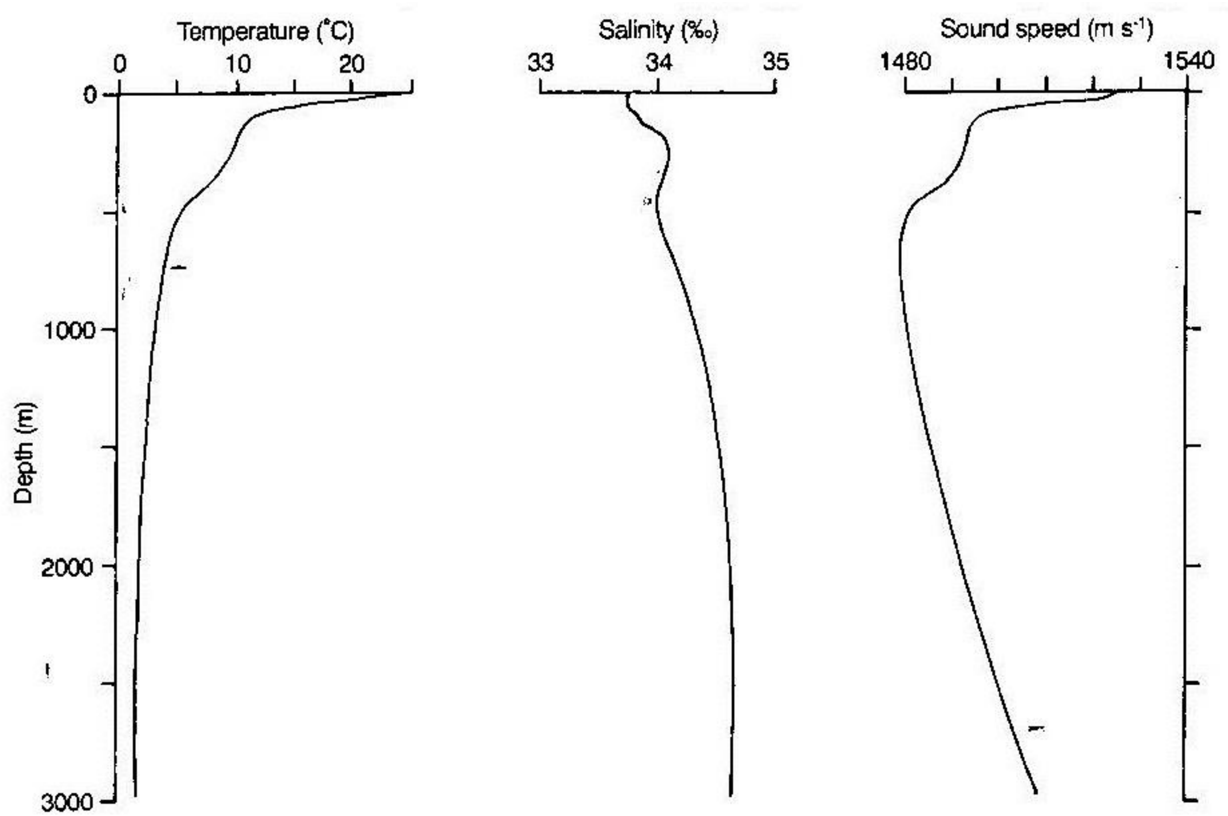

Figure 4

Chapter 2

Hydroplow and Instrumentation

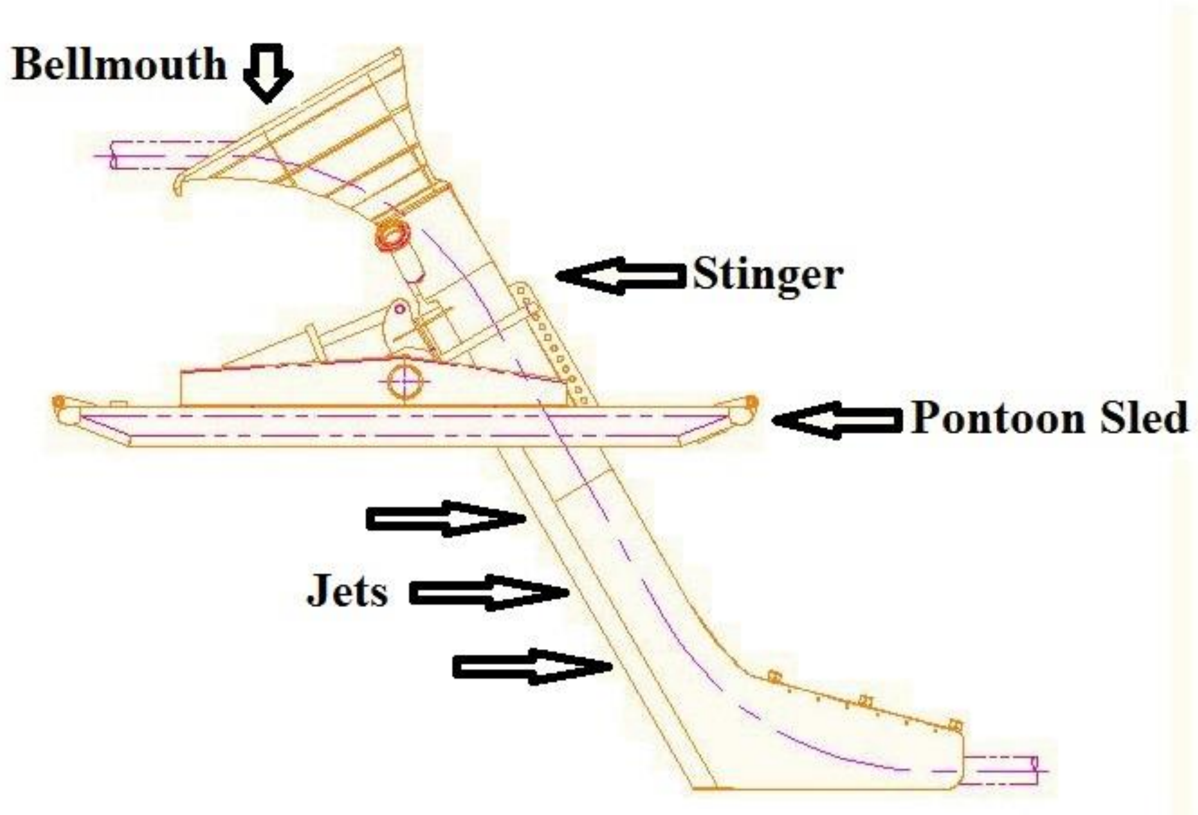

Figure 5

A hydroplow is a towed submersible vehicle that buries cable under the seabed. Please see **Figure 5**. The hydroplow is comprised of two main pieces: a stinger and a pontoon sled.

The stinger is a large movable piece that is mounted on the pontoon sled and can rotate between 0 and 90 degrees relative to the sled. Atop the stinger is a “bellmout”. This is a wide opening to a funnel which provides a path for the cable. During operation, cable is fed into the bellmouth of the stinger and deposited out of the keel.

The plow is towed via bridles mounted at the front of each pontoon. Two winches aboard the vessel control the tension at these contacts.

While not in operation, the plow is kept on the aft of the vessel with the stinger retracted to ~16 degrees. See **Figure 6** below.

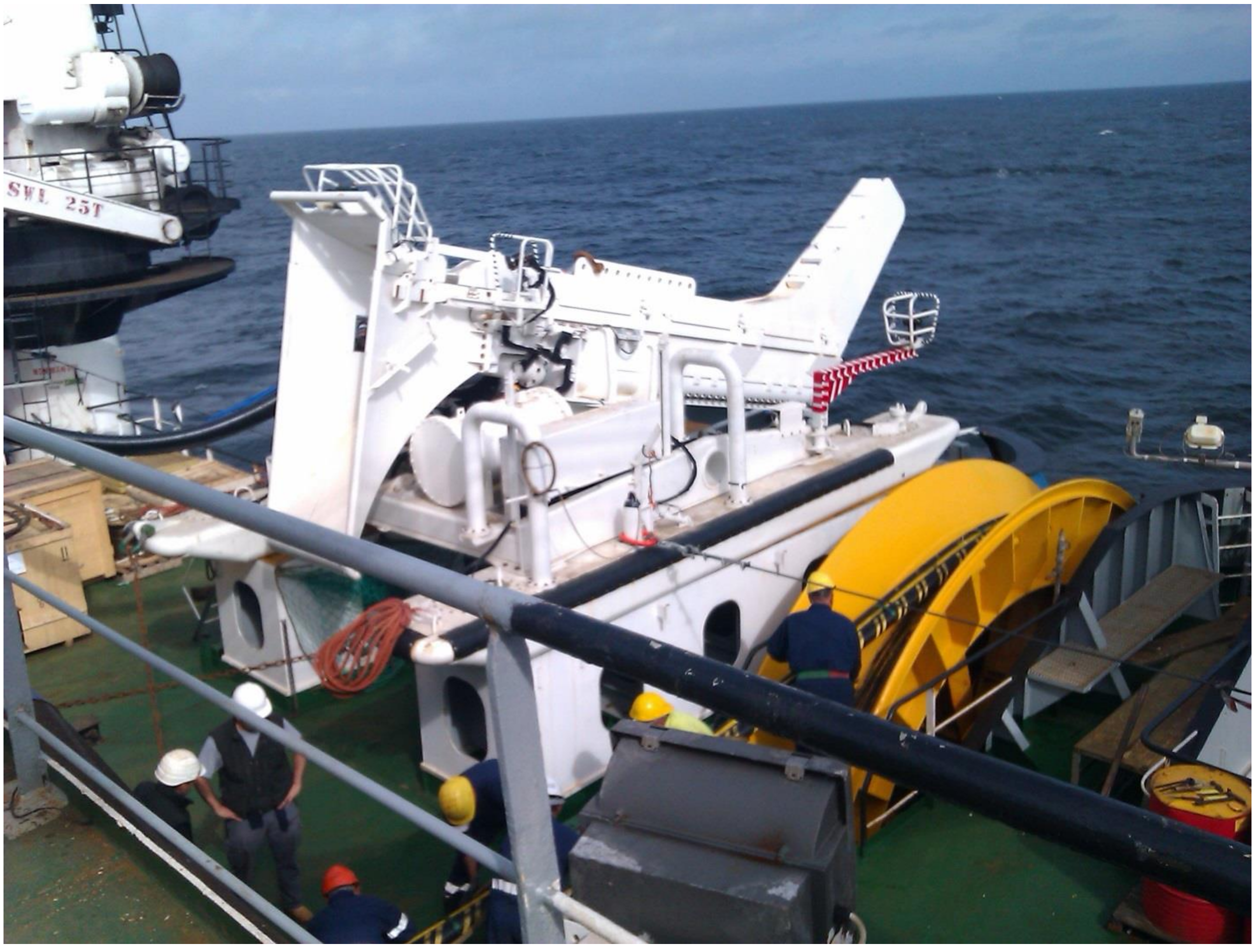

Figure 6

To begin operations the plow is launched from the back of the ship via crane and lowered to the seabed at 40m – 100m depth. See **Figure 1**.

Once on the seabed, top of the stinger is opened using remote controls provided by telemetry. The cable to be buried is carefully lowered into the open stinger. Once this cable is in place, the stinger is closed and burial may begin.

High pressure jets located along the front of the stinger are activated. The water pressure cuts through the seabed acting as a trencher. At this point the stinger may be rotated to about 40 degrees relative to the pontoon sled which deposits the cable to about 3M beneath the soil.

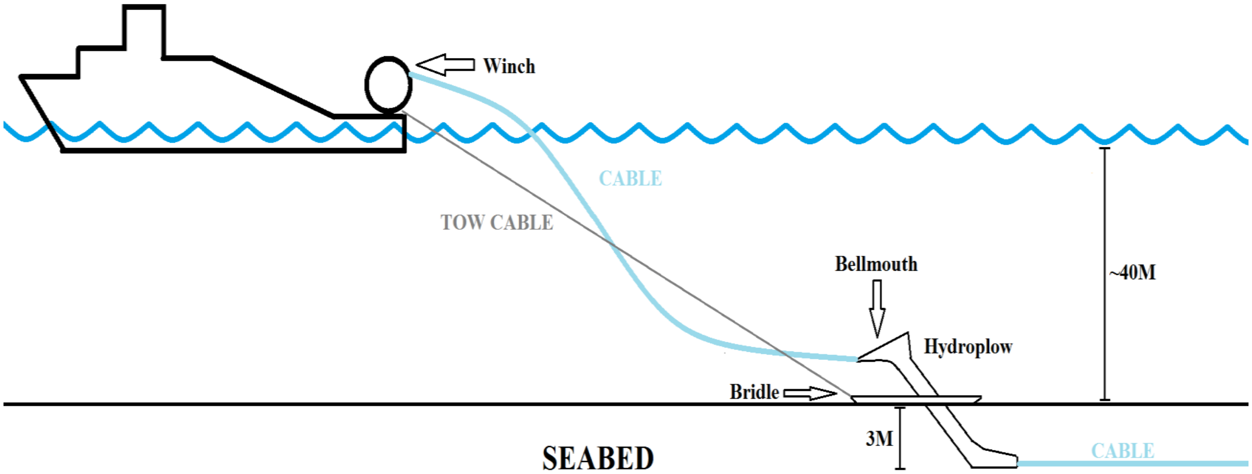

Figure 7

The vessel tows the plow along the seabed while paying out cable. Under favorable conditions 8km of cable may be buried in 24hrs.

See below a series of screen shots taken from a 3d graphical display designed and implemented by the author. This responds in real time to sensor data from the plow. Please observe how pitch and roll of the vehicle affect depth of the keel beneath the seabed and thus depth of cable burial. This is extremely interesting data as burial depth determines payment to our client.

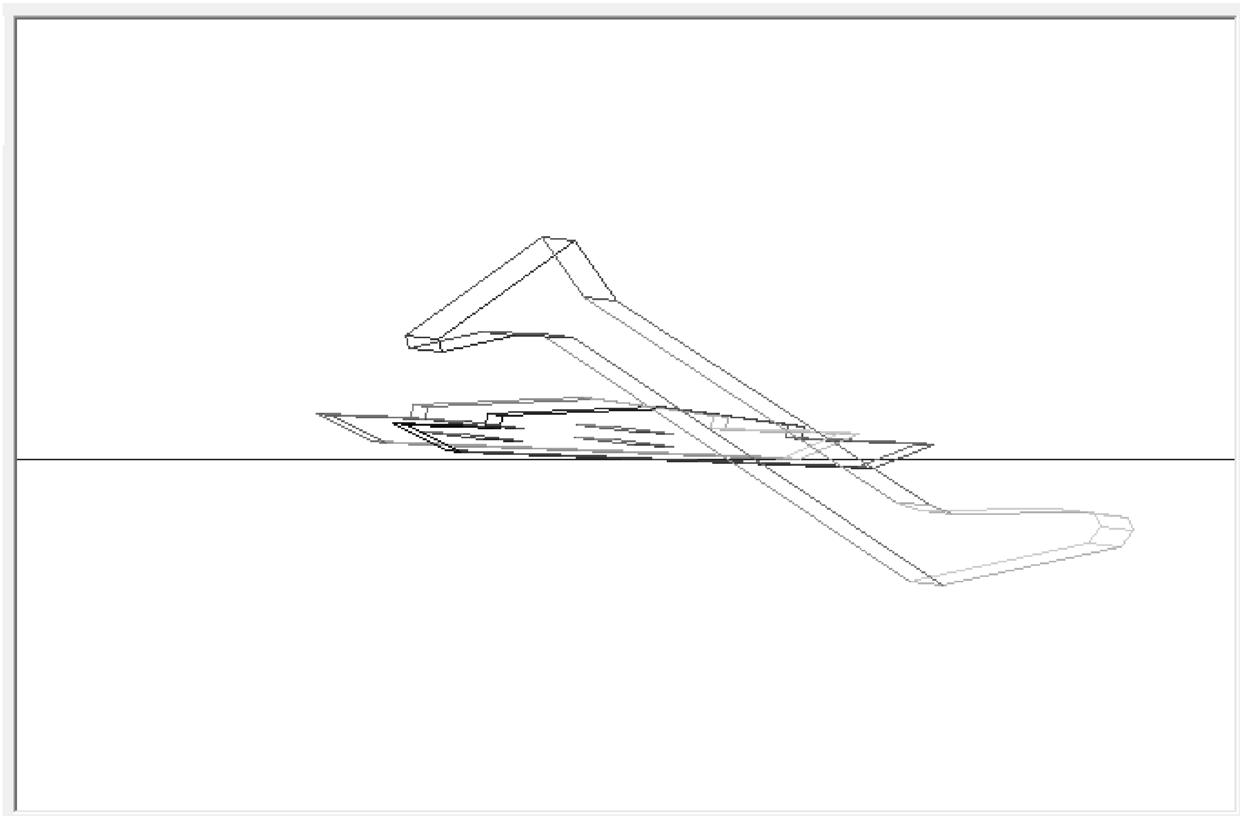

Figure 8

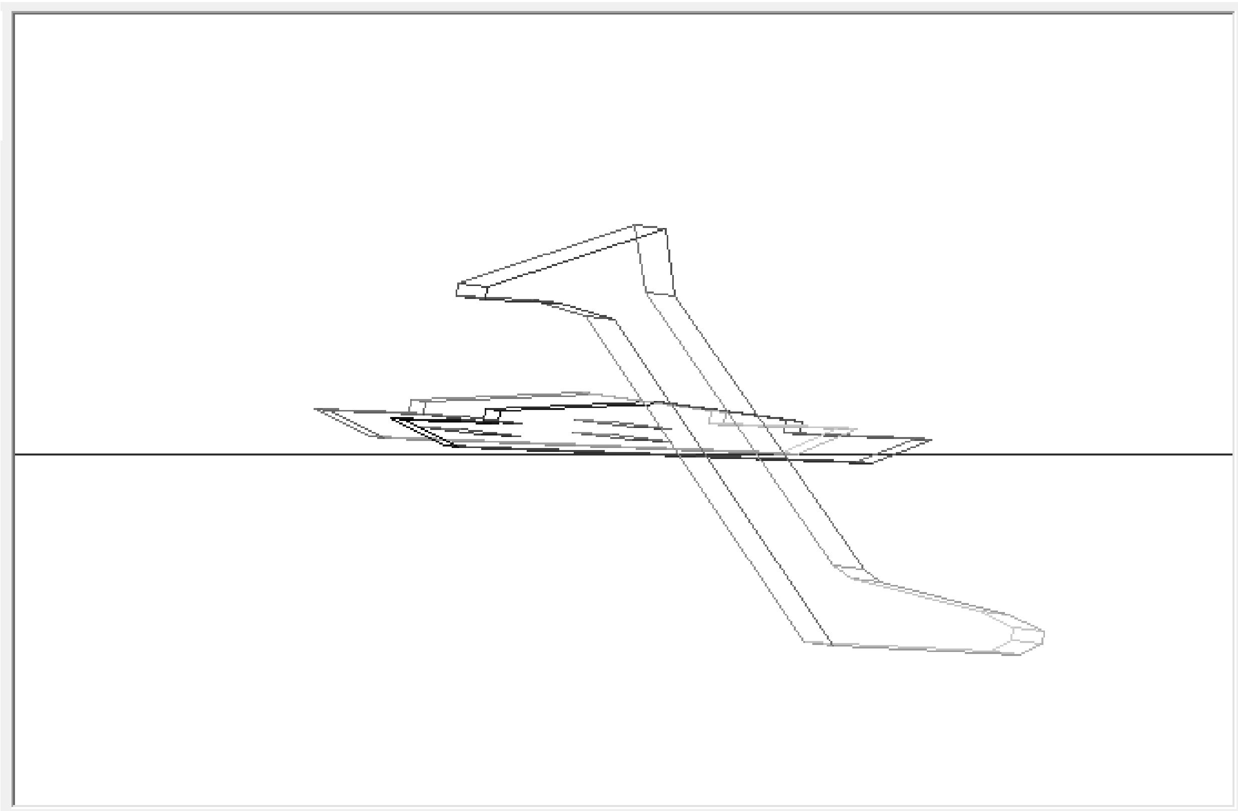

Figure 9

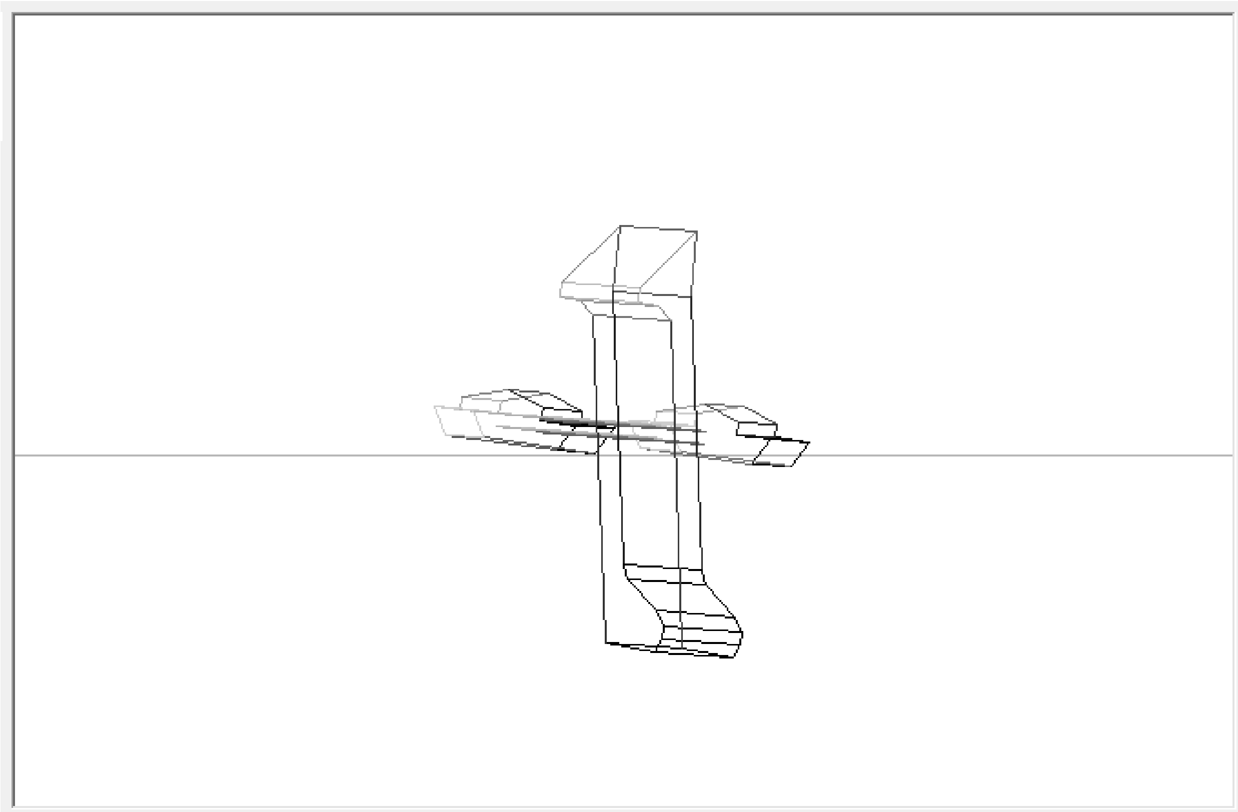

Figure 10

Burial data is recorded to a log file and exported via serial link to the client in real time. Attitude data is also critical to monitor stability of the plow. Capsizing results in significant interruption to operations.

Now that I have provided some environmental and operational background, let's talk about how to obtain meaningful physical measurements. **Figure 11** shows a processing block diagram of the telemetry system, including sensors, serial communications, and processor.

The telemetry system has ten distinct sensors along with several related control functions. In order to examine them in more detail, I will break down these sensors into operational categories; acoustic, pressure, and inductive. First, let's take a look at the use of sonar transducers.

2.1 Sonar Transducer

The system uses both off the shelf and custom acoustic sensing equipment. All employ active sonar techniques. This involves sending and receiving acoustic signals at specific frequencies. The tool that makes this possible is the sonar transducer. These are discrete components that can convert electrical signal into acoustic signal and vice versa. Three material properties are used to make transducers; piezoelectricity, electrostriction, and magnetostriction.

The piezoelectric effect has been exploited in sonar technology since the early twentieth century. An electric field applied to piezoelectric crystals will cause them to change dimensions. If the field is alternating, the deformations will occur at the same frequency as the applied field, creating an acoustic signal. This effect is reversible, where pressure applied to the crystal results in an electric charge.

During World War I, French physicist Paul Lanevin built a submarine detector that consisted of piezoelectric transducer crystals and a hydrophone. A high frequency electric field applied to the crystal resulted in a high frequency chirp. By measuring the response time with the hydrophone, the distance of an object could be calculated.

The magnitude of physical change is proportional to the electric field and can roughly be expressed as a relation of the electric displacement D, the amount of mechanical stress **s**, and two material specific constants, **r** and **k**.

$$\xi \equiv \mathbf{rs} + \mathbf{kD}$$

Critically, in the piezoelectric effect changes in the field direction result in changes in the deformation sign.

The electrorestrictive effect which is common in all dielectrics does not depend on the direction of materials applied field. In most cases, the deformation is too small to be useful. However in ferroelectric ceramic materials, deformation is significant and can be used in acoustic probe applications. The frequency which the material oscillates is double that of the electric field frequency.

Magnetorestrictive materials change dimensions when subjected to a magnetic field. The changes in dimension are independent of the direction of the field. This effect is the result of the structure of ferromagnetic materials. Internally these materials are divided into regions of uniform magnetic polarization. In the presence of a magnetic field these regions shift and rotate resulting in deformation.

2.2 Altimeter

The distance from the pontoons of the hyrdroplow to the seafloor is an important piece of information in determining burial performance. Combine with depth, pitch, roll, and stinger angle, we can use altitude to geometrically estimate the depth below the seafloor that the cable is being deposited.

The altimeter acts as a downward facing sonar, transmitting and receiving pings to the seafloor to determine the relative position of the hydroplow. It is mounted 50 cm above the bottom of the left pontoon, just behind a forward facing cross bar for shielding.

There are a number of technical complications of this set up. The signal transmitted by the depth sounder propagates conically downward resulting in spreading loss. Sediment in the water stirred up by the motion if the plow also results in attenuation. The soft muddy bottom of the North Sea is not an ideal sonar target given its high level of absorption. Total transmission loss and backscattering strength of the target influence the signal to noise ratio.

There are well developed equations that can help us understand this process in more detail. Consider the parameters of the equipment. There is the acoustic intensity of the original signal, source level SL. The directivity index is a measure of the focusing power of the device. This of course can apply to both the acoustic source DI_T, and the receiver DI_R. Detection threshold DT is the level the signal to noise ratio must exceed for the target to be seen. Then there is the medium through which the signal will travel. Here we expect transmission loss TL, a level of reverberation RL, and background noise NL. Target strength TS, describes how good of an acoustic reflector the target is. All of these quantities can be expressed on a log scale, dB.

The signal to noise ratio can thus be described by the following equation.

$$SL + DI_T + TS - 2TL - (NL - DI_R)$$

Notice that we are accounting for two way transmission loss. Using this we can define the detection threshold. This gives us the full equation for active sonar with noise limited performance.

$$DT = SL + DI_T + TS - 2TL - (NL - DI_R)$$

It is common that the acoustic transmitter also acts as a hydrophone which would make there acoustic focusing abilities equivalent, so $DI_T = DI_R$.

When looking at this it is easy to think that increasing the source level SL is the best way to improve signal to noise ratio. However, backscattering results in reverberation which can obscure the signal. This is particularly a problem in situations such as this where the target will be within a meter of the detector. This constitutes an upper limit on source level SL.

2.3 Inductive Sensors, Magnetic Compassing

2.3.1 Heading, Pitch, Roll

Pitch and roll are critical measurements in estimating burial depth and monitoring the stability of the hydroplow. One of the sensors used to collect this data is the *OS5000* from Ocean Server technologies, an off the shelf digital compass. The compass features two anisotropic magnetoresistance (AMR) sensors. One AMR sensor is two-axis for X and Y plane sensing, the other is single-axis for the Z plane. This in combination with a three axis accelerometer provides very reliable data.

2.3.2 Geomagnetic Field and Magnetoresistance

A material is magnetoresistant if its electrical resistivity changes when an external magnetic field is applied to it. In anisotropic magnetoresistance, the change in resistance is directly related to the angle between the direction of current and the applied magnetic field. Generally, electric resistance is greatest when the magnetic field and electric current are aligned.

Magnetic compassing is made possible by the earth's magnetic field. The direction of this field is dependent on geographic location. Around the equator the field will be horizontal. Approaching magnetic north, the field will progressively point downward, while approaching the south, the field will progressively point up.

The compass contains three magnetic sensors to measure three orthogonal vectors corresponding to the vertical and horizontal components of the earth's magnetic field. The accelerometer acts as a gimbal to measure gravitational direction. This gives us a tilt compensated value for pitch, roll and heading.

2.3.4 Magnetic field distortions

In the design process it is important to consider magnetic field distortions. Magnetized ferrous materials near the compass will result in hard-iron effects. This is a constant additive field that will change the values measured by the sensor. So long as the position and strength of the magnetic source is known, calibration and determining offset should be trivial.

Soft-iron effects, arising from the many electric currents and discrete components of an embedded system, result in distortions that are more difficult to deal with. For this reason, locating the sensor as far away from power sources, conducting wires and other soft-iron is the best solution. The compass was mounted inside a subsea electronics housing bottle, so the location could not be totally isolated from electrical noise.

Two *OS5000* digital compasses are used in the system. One measures pitch, roll and heading on the sled. The other is used to measure the pitch of the stinger. This is done so that the angle of the stinger relative to the plow can be calculated.

Chapter 3

Data Processing

My primary contribution to the development of the system was writing the software. This included software for the embedded system that was housed in the subsea electronics bottle as well as a Windows based GUI application for a PC on the vessel. In order to walk you through this process, I am first going to provide an overview of the electronics.

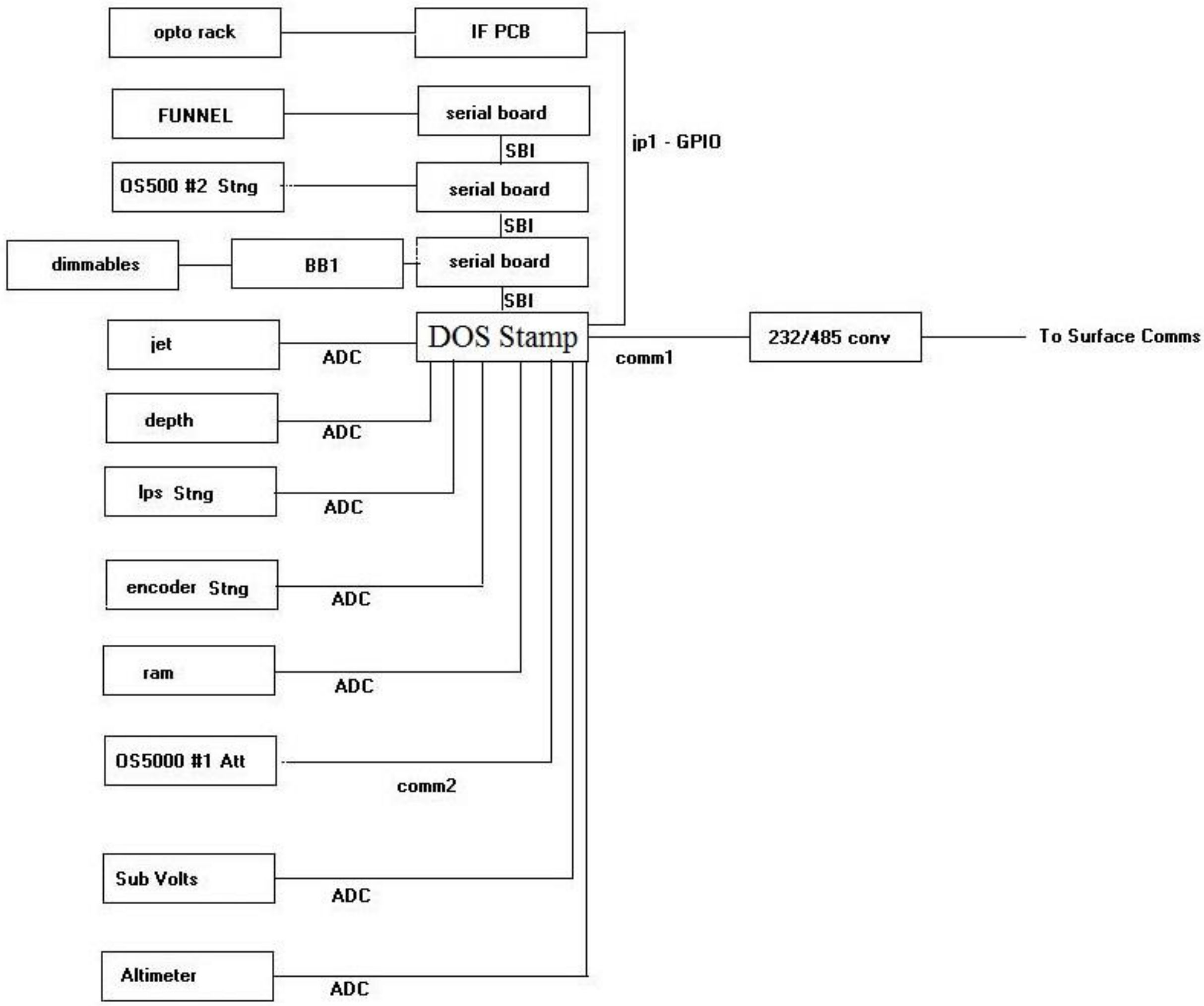

Figure 11

3.1 Electronics

The system uses an 8086 based microcontroller as its CPU. It is called a DOS Stamp™ and is produced by a company called Bagotronix™. This tiny computer runs MS DOS and has two on board RS232 serial ports for communication. One of these ports is dedicated to communicating with the surface interface. Given the number of electrical components on the system, three serial extension boards are necessary, all using RS232 standard.

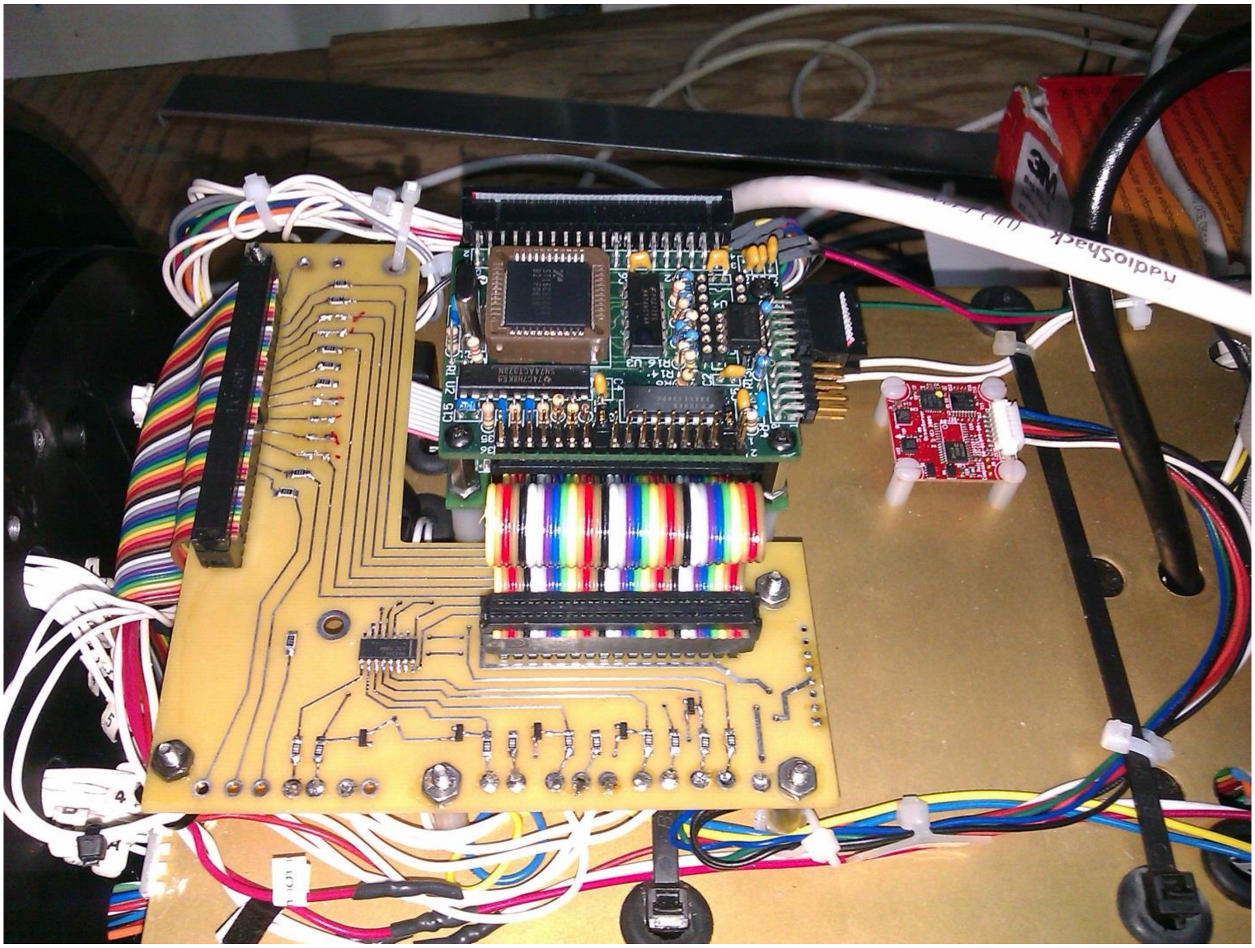

Figure 12

3.2 Embedded Software

The embedded software was written in C code and compiled using Borland 3.1. This language and specific compiler were necessary due to the design of the microcontroller. Other compilers, such as Microsoft's pad executable files with extra zeros in a way that is in compatible with the DOS Stamp.

There are eight sensors that produce an analog signal. Conveniently, the DOS Stamp has an on board eight channel ADC; altimeter, depth sensor, rotary encoder, jet pressure, linear position sensor, ram, bottle leak alarm, and voltmeter. The channel and pin assignments are as follows.

ADC channel	JP2	func	PIO	GPIO	JP1	volts
0	40	alt	18	11	5	0 to 10
1	39	depth	18	11	5	0 to 5
2	38	encoder	18	11	5	0 to 10
3	37	jet	18	11	5	0 to 10
4	36	lps	18	11	5	0 to 5
5	35	ram	18	11	5	0 to 10
6	34	leak	18	11	5	0 to 5
7	33	sub v	18	11	5	0 to 10

Figure 13

A code sample of the C function that handled these conversions is shown below.

```
void ADC()
{
        UWORD alt, depth, encoder, jet, leak, lps, ram, sub_v;
        double fval;

        ADCput(0, ADC_0_TO_10); // ALTIMETER
        while (!ADCready());    // WAIT FOR CONVERSION
        alt = ADCget();
        fval = (double)alt * ADC_10V_CONV;
        sprintf(s_altimeter, "%.2f", fval);

        ADCput(1, ADC_0_TO_5);  // DEPTH
        while (!ADCready());    // WAIT FOR CONVERSION
        depth = ADCget();
        fval = (double)depth * ADC_5V_CONV;
        sprintf(s_depth, "%.3f", fval);

        ADCput(2, ADC_0_TO_10); // ENCODER
        while (!ADCready());    // WAIT FOR CONVERSION
        encoder = ADCget();
        fval = (double)encoder * ADC_10V_CONV;
        sprintf(s_encoder, "%.2f", fval);

        ADCput(3, ADC_0_TO_10); // JET
        while (!ADCready());    // WAIT FOR CONVERSION
        jet = ADCget();
        fval = (double)jet * ADC_10V_CONV;
        sprintf(s_jet, "%.2f", fval);

        ADCput(4, ADC_0_TO_5);  // LPS
        while (!ADCready());    // WAIT FOR CONVERSION
        lps = ADCget();
        fval = (double)lps * ADC_5V_CONV;
        sprintf(s_lps, "%.2f", fval);
```

Figure 14

This function pings each channel with a bit which triggers a conversion of 0 – 10 volts to a twelve bit digital value. Once the conversion is ready, the digital information is cast into an ASCII character string. All sensor data is converted to this format for serial communication.

The two *OS5000* digital compasses support direct RS232 communication, so they can talk directly with the microcontroller via serial port. One had to run through a serial extension board which affected the timing such that the output sentences were garbled by the time they reached the DOS Stamp. To solve this I had to clear the UART (universal asynchronous receiver/transmitter) every time a sentence was transmitted.

Controls for the subsea dimmable lights proved more of a technical challenge. The luminescence of each light could be controlled by varying voltage on one of the pins. This required a digital to analog converter. For this we used a B&B Electronics module. Unfortunately it did not have a software library that was compatible with our system, thus I had to construct instructions as a 16 bit Boolean array in the C code and export them to the DAC directly.

The subsea executable ran a loop which would print out a character string containing all of the data being collected subsurface. Here is an example.

$H180P-5.2R-2.56T20D100L111C111J10F0001r10S1A1N100i100I100a100v1W0

This information is then piped to the surface via serial communication. RS232 signal attenuates after tens of meters, so conversion to RS485 is necessary. Multiple conversions led to a problem with cross talk. This was solved with a data filter implemented in the software.

3.3 GUI

The surface application is a Windows based GUI which displays data, creates logs, and has controls for the plow such as stinger movement. The client requested this be done in Visual Basic 6. The use of a relatively antiquated platform on a modern Windows operating system provided some technical difficulties.

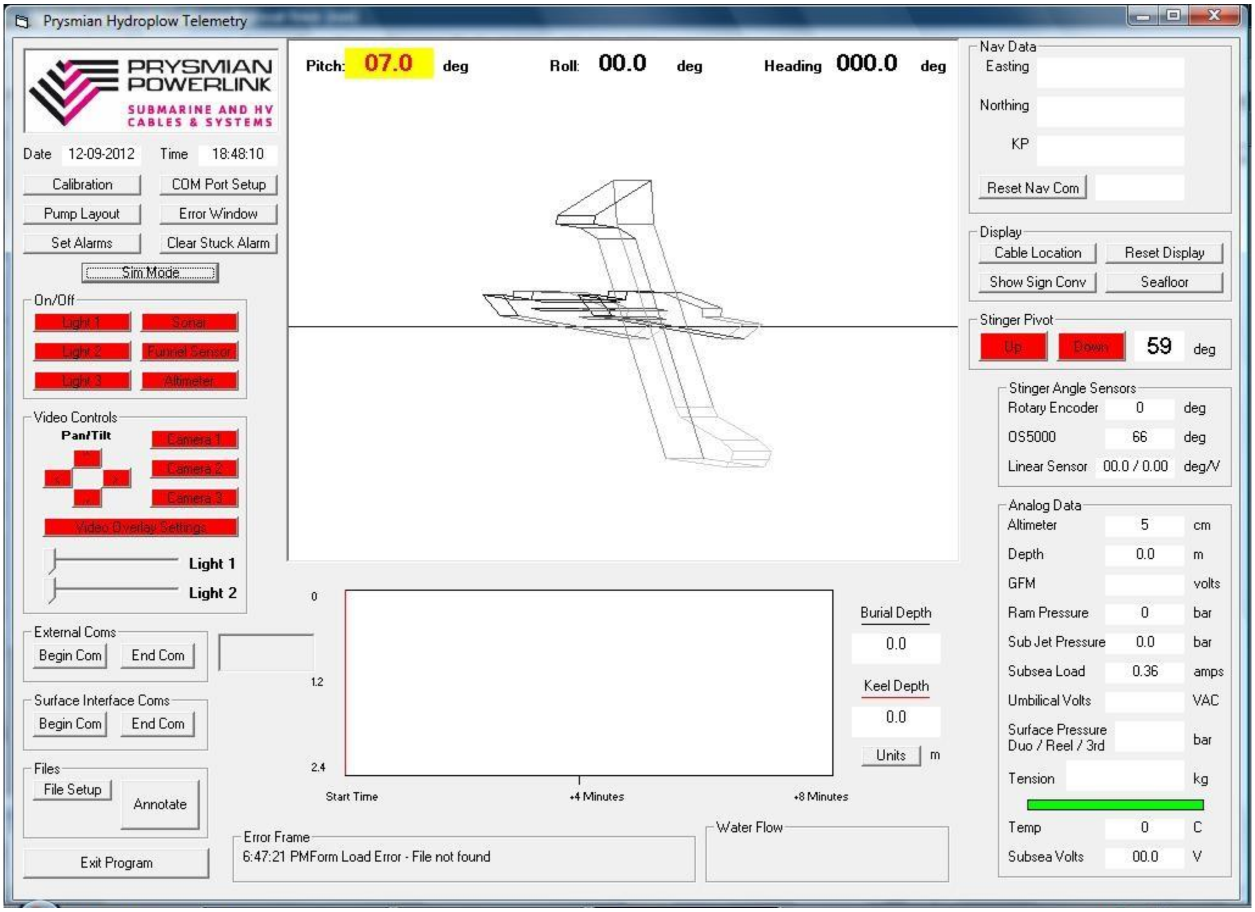

Figure 15

The client requested a three dimensional wireframe display of the plow. This would demonstrate pitch and roll in real time based on data collected from the digital compasses. The graphic could be manipulated by dragging and click. I will not go into the implementation of this being that it does not have much to do with physics. However, I would like to add that this ended up being a time consuming part of the development process.

Chapter 4

Conclusions and Future Work

4.1 Conclusions

My involvement with this project has been a tour de force in learning about the design, assembly and operation of physical measurement systems in a submarine environment. As it is my ambition to pursue a career in physical oceanography and technology development, these are valuable skills and experience that I can build on. In regards to this the project was a total success.

As far as the performance of the system in the field, there are a few suggestions I would make for future work. We had chronic problems with serial communication from the subsea bottle to the surface. These signals were carried over a twisted pair. We were able to add noise filtering with satisfactory results. However, in the future I think using fiber optics would be a more reliable solution. Also, using the downward facing sonar as an altimeter was compromised by the fact that it was fixed to the frame of the plow. As a result, the plow pitching up or down would obscure the results as the sonar would no longer be normal to the seafloor. This could be fixed by using a gimbal.

4.2 Future Work

I spent the first month of 2013 in Carmignano, Italy refitting the system for another deployment in the North Sea. Several significant changes were made to the system, including adding redundant sensors, rebuilding the power distribution, and taking steps to reduce noise in the serial communication to the surface.

In a couple of weeks I will head back to the North Sea for another cable laying operation. I intend to continue to take notes, conduct in the field research, and gain practical experience in taking physical measurements in a submarine environment.

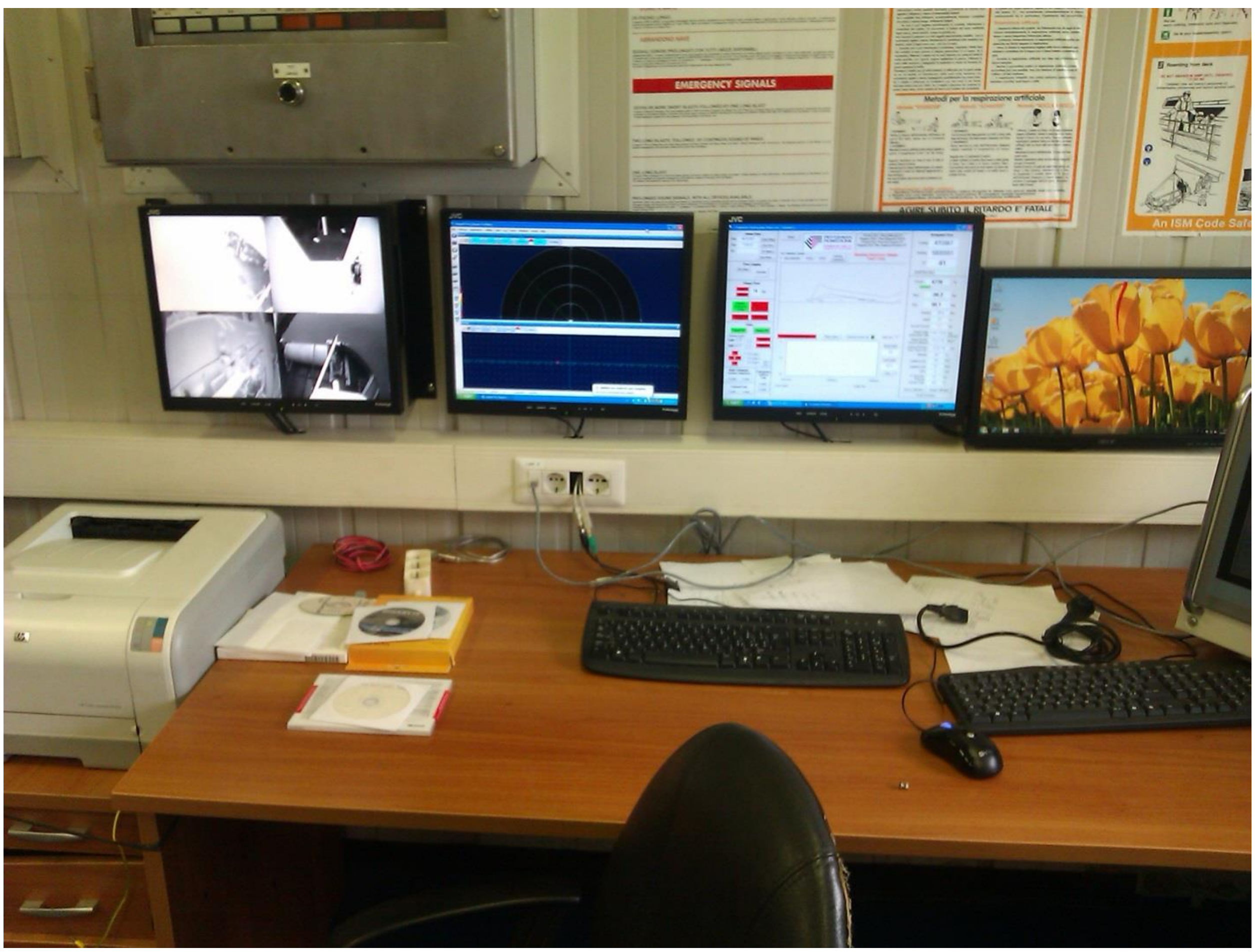

Figure 16

4.3 Addendum: Future 'Future Work'

Since the first draft of this paper I have been on three cruises and two additional design cycles with this machine. Among numerous technical upgrades, we have now more than doubled the operational depth of the cable burial machine to more than 100m. While this may not sound impressive compared to many submersible vehicles, it is a unique capability for seabed cable burial.

I am happy to mention that I have begun collaborating with BluHaptics, a startup firm hosted at UW's APL. We are working to integrate their data cloud technology into the technology I have been working on in order to develop a cutting edged telemetry and control system that will be unique in today's offshore industry.

Appendix One Embedded System Source Code

```
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
#include "stdinc.h"
#include "dosstamp.h"
#include "dsadc.h"
#include "dscomm.h"
#include "csfc8250.h"

#define ADC_5V_CONV         (5.0 / 4095.0)
#define ADC_10V_CONV        (10.0 / 4095.0)

IOPARMS cp1;
IOPARMS cp2;
IOPARMS cp3;

ULONG combaud = 9600L;
UWORD comdata = 8;
UWORD comparity = 0;
UWORD comstop = 1;
UWORD bauddiv;

void ADC(void);
void compass(void);
void compass2(void);
void funnel_sensor(void);
void light_1(void);
void light_2(void);

char s_altimeter[10], s_depth[6], s_encoder[6], s_funnel[3], s_jet[10], s_lps[10], s_ram[10], s_volts[5];
char s_compass[35];
```

```
char s_compass2[35];
char alt_stat, cam1, cam2, cam3, funnel, leakFlag, light1, light2, light3, sonar;

int level_1, level_2;       // MAX VALUE 255

void ADC()
{
        UWORD alt, depth, encoder, jet, leak, lps, ram, sub_v;
        double fval;

        ADCput(0, ADC_0_TO_10); // ALTIMETER
        while (!ADCready());    // WAIT FOR CONVERSION
        alt = ADCget();
        fval = (double)alt * ADC_10V_CONV;
        sprintf(s_altimeter, "%.2f", fval);

        ADCput(1, ADC_0_TO_5);      // DEPTH
        while (!ADCready());    // WAIT FOR CONVERSION
        depth = ADCget();
        fval = (double)depth * ADC_5V_CONV;
        sprintf(s_depth, "%.3f", fval);

        ADCput(2, ADC_0_TO_10);     // ENCODER
        while (!ADCready());    // WAIT FOR CONVERSION
        encoder = ADCget();
        fval = (double)encoder * ADC_10V_CONV;
        sprintf(s_encoder, "%.2f", fval);

        ADCput(3, ADC_0_TO_10);     // JET
        while (!ADCready());    // WAIT FOR CONVERSION
        jet = ADCget();
        fval = (double)jet * ADC_10V_CONV;
        sprintf(s_jet, "%.2f", fval);

        ADCput(4, ADC_0_TO_5);  // LPS
        while (!ADCready());    // WAIT FOR CONVERSION
        lps = ADCget();
        fval = (double)lps * ADC_5V_CONV;
        sprintf(s_lps, "%.2f", fval);

        ADCput(5, ADC_0_TO_10); // RAM
        while (!ADCready());    // WAIT FOR CONVERSION
        ram = ADCget();
        fval = (double)ram * ADC_10V_CONV;
        sprintf(s_ram, "%.2f", fval);

        ADCput(6, ADC_0_TO_5);      // LEAK
        while (!ADCready());    // WAIT FOR CONVERSION
        leak = ADCget();
        fval = (double)leak * ADC_5V_CONV;
        if (fval <2.5)
```

```
		leakFlag = '1';
	else leakFlag = '0';

	ADCput(7, ADC_0_TO_5);		// SUB VOLTS
	while (!ADCready());	// WAIT FOR CONVERSION
	sub_v = ADCget();
	fval = (double)sub_v * ADC_5V_CONV;
	sprintf(s_volts, "%.2f", fval);

}

void compass()
{
	UBYTE b, idx;

	idx=0;
	b=0;
	while (b!='$')	// GET TO BEGINING '$'
		if (SERrcvrready(1)) b = SERgetc(1);

	b=0;
	while (b!='*' && idx<30) {		// GET TO END '*'
		if (SERrcvrready(1)) {
			b = SERgetc(1);
			s_compass[idx++] = b;
		}
	}

	s_compass[idx-1]=0;	// GET RID OF '*'
	idx=0;
	while (s_compass[idx]) {
		if (s_compass[idx]=='C') {		// REPLACE C WITH H
			s_compass[idx]='H';
			break;
		}
		idx++;
	}
}

void compass2()
{
	BYTE b, idx;

	idx=0;
	b=0;
	while (b!='P')	// ISOLATE PITCH
		if (_8250_rcvrready (&cp2)) b = _8250_getc (&cp2);

	b=0;
	while (b!='R' && idx<30) {		// STOP READING AFTER PITCH
		while(!_8250_rcvrready (&cp2));
```

```
		b = _8250_getc (&cp2);
		s_compass2[idx++] = b;}


	s_compass2[idx-1]=0; // GET RID OF 'R'
	idx=0;
	while (s_compass2[idx]) {
		if (s_compass2[idx]=='P') {	// REPLACE 'P' WITH 'N'
			s_compass2[idx]='N';
			break;
		}
		idx++;
	}
}

void funnel_sensor()
{
	UBYTE e, idx;

	idx=0;
	e = 1;

	while(!_8250_xmtrready(&cp3));
	_8250_putc(&cp2, e);
	while (!_8250_xmtrdone(&cp3));

	while(idx < 4){
	while(!_8250_rcvrready (&cp3));
	e = _8250_getc (&cp3);
	s_funnel[idx] = e;
	idx++;}
}

void light_1()
{
	int code, i, j, n, bin[7], d[15], byte1, byte2;
	BYTE a;

	code = level_1;
	n = 128;
	i = 0;

	while(i < 8){
	if(code >= n){
	code = code - n;
	bin[i] = 1;}
	else bin[i] = 0;
	n = n/2;
	i++;}

	// POPULATE FINAL ARRAY
```

```
	i = 0;	// CHANNEL AND MULTIPLIER
	j = 0;
	while(i < 3){
	d[i] = 0;
	i++;}
	d[2]=1;

	while(i < 11){	// DIGITAL CODE
	d[i] = bin[j];
	i++;
	j++;}

	while(i < 16){	// IGNORED BITS
	d[i] = 0;
	i++;}

	// GET COMMAND BYTES
	byte1 = d[0]*128 + d[1]*64 + d[2]*32 + d[3]*16 + d[4]*8 + d[5]*4 + d[6]*2 + d[7]*1;
	byte2 = d[8]*128 + d[9]*64 + d[10]*32 + d[11]*16 + d[12]*8 + d[13]*4 + d[14]*2 + d[15]*1;

	// DUMP VALUES
	i=0;
	while (i < 8){
	bin[i] = '\0';
	i++;}

	i=0;
	while(i < 16){
	d[i] = '\0';
	i++;}

	a = '!';
	_8250_putc(&cp1, a);
	a = '0';
	_8250_putc(&cp1, a);
	a = 'S';
	_8250_putc(&cp1, a);
	a = 'V';
	_8250_putc(&cp1, a);
	_8250_putc(&cp1, byte1);
	_8250_putc(&cp1, byte2);
}

void light_2()
{
	int code, i, j, n, bin[7], d[15], byte1, byte2;
	BYTE a;

	code = level_2;
	n = 128;
	i = 0;
```

```
while(i < 8){
if(code >= n){
code = code - n;
bin[i] = 1;}
else bin[i] = 0;
n = n/2;
i++;}

// POPULATE FINAL ARRAY
i = 0;     // CHANNEL AND MULTIPLIER
j = 0;
while(i < 3){
d[i] = 0;
i++;}
d[1]=1;
d[2]=1;

while(i < 11){    // DIGITAL CODE
d[i] = bin[j];
i++;
j++;}

while(i < 16){    // IGNORED BITS
d[i] = 0;
i++;}

// GET COMMAND BYTES
byte1 = d[0]*128 + d[1]*64 + d[2]*32 + d[3]*16 + d[4]*8 + d[5]*4 + d[6]*2 + d[7]*1;
byte2 = d[8]*128 + d[9]*64 + d[10]*32 + d[11]*16 + d[12]*8 + d[13]*4 + d[14]*2 + d[15]*1;

// DUMP VALUES
i=0;
while (i < 8){
bin[i] = '\0';
i++;}

i=0;
while(i < 16){
d[i] = '\0';
i++;}

a = '!';
_8250_putc(&cp1, a);
a = '0';
_8250_putc(&cp1, a);
a = 'S';
_8250_putc(&cp1, a);
a = 'V';
_8250_putc(&cp1, a);
_8250_putc(&cp1, byte1);
```

```
	_8250_putc(&cp1, byte2);
}

void _16550_FifoCtl (IOPARMS *cp, BOOLEAN FifoEnable, BOOLEAN ResetRxFifo,
	BOOLEAN ResetTxFifo, UBYTE RxFifoThreshold) {

	outportb (cp->base+2, (RxFifoThreshold << 6) | (ResetTxFifo ? 0x04 : 0)
		| (ResetRxFifo ? 0x02 : 0) | (FifoEnable ? 0x01 : 0));
}

void main()
{
	WORD ch;
	char q, dim[3];
	int j, idx;

	cam1 = '0';
	cam2 = '0';
	cam3 = '0';
	level_1 = 0;
	level_2 = 0;
	light1 = '0';
	light2 = '0';
	light3 = '0';
	sonar = '0';
	idx = 0;

	//	INITIALIZE DOSSTAMP SERIAL
	SERsetbaud(1, B9600);
	SERinit(1, FALSE, FALSE, FALSE);
	PIOconfig(28, PIOCFG_NORMAL, 1);

	//	INITIALIZE SERIAL EXTENSION BOARDS
	//	address func jumpers
	//	===================
	//	800h	DAC	3-4
	//	820h	os5000	5-6
	//	840h	funnel	7-8


	PIOconfig (16, PIOCFG_NORMAL, 1); // CONFIGURE PIO AS CHIP SELECT

	cp1.base = 0x800;
	cp1.intr = 0;

	cp2.base = 0x820;
	cp2.intr = 0;

	cp3.base = 0x840;
	cp3.intr = 0;
```

```
	bauddiv = (UWORD)(115200L / combaud);

	// ENABLE FIFO cp1
	_16550_FifoCtl (&cp1, TRUE, TRUE, TRUE, 3);		// 3 -> threshold = 14

	// cp1 INITIALIZE 8250
	_8250_init (&cp1, bauddiv, comdata, comparity, comstop, FALSE, FALSE, FALSE, cp1.intr ?
TRUE : FALSE);

	// ENABLE FIFO cp2
	_16550_FifoCtl (&cp2, TRUE, TRUE, TRUE, 3);		// 3 -> threshold = 14

	// cp2 INITIALIZE 8250
	_8250_init (&cp2, bauddiv, comdata, comparity, comstop, FALSE, FALSE, FALSE, cp2.intr ?
TRUE : FALSE);

	// ENABLE FIFO cp3
	_16550_FifoCtl (&cp3, TRUE, TRUE, TRUE, 3);		// 3 -> threshold = 14

	// cp3 INITIALIZE 8250
	_8250_init (&cp3, bauddiv, comdata, comparity, comstop, FALSE, FALSE, FALSE, cp3.intr ?
TRUE : FALSE);

	// MAIN LOOP
	while (TRUE) {

		if(kbhit()){
			ch = getch();

			if (ch == 'a') {    // ALTIMETER OFF
				PIOconfig(30, PIOCFG_OUTPUT, 0);
				alt_stat = '0';
			}

			if (ch == 'A') {   // ALTIMETER ON
				PIOconfig(30, PIOCFG_OUTPUT, 1);
				alt_stat = '1';
			}

			if (ch == 'b') {    // CAMERA1 OFF
				PIOconfig(12, PIOCFG_OUTPUT, 0);
				cam1 = '0';
			}

			if (ch == 'B') {   // CAMERA1 ON
				PIOconfig(12, PIOCFG_OUTPUT, 1);
				cam1 = '1';
			}

			if (ch == 'c') {    // CAMERA2 OFF
				PIOconfig(13, PIOCFG_OUTPUT, 0);
```

```
        cam2 = '0';
}

if (ch == 'C') {   // CAMERA2 ON
        PIOconfig(13, PIOCFG_OUTPUT, 1);
        cam2 = '1';
}

if (ch == 'd') {   // CAMERA3 OFF
        PIOconfig(20, PIOCFG_OUTPUT, 0);
        cam3 = '0';
}

if (ch == 'D') {   // CAMERA3 ON
        PIOconfig(20, PIOCFG_OUTPUT, 1);
        cam3 = '1';
}

if (ch == 'e') {   // FUNNEL_SEN OFF
        PIOconfig(11, PIOCFG_OUTPUT, 0);
        funnel = '0';
}

if (ch == 'E') {   // FUNNEL_SEN ON
        PIOconfig(11, PIOCFG_OUTPUT, 1);
        funnel = '1';
}

if (ch == 'f') {   // LIGHT1 OFF
        PIOconfig(10, PIOCFG_OUTPUT, 0);
        light1 = '0';
}

if (ch == 'F') {   // LIGHT1 ON
        PIOconfig(10, PIOCFG_OUTPUT, 1);
        light1 = '1';
}

if (ch == 'g') {   // LIGHT2 OFF
        PIOconfig(0, PIOCFG_OUTPUT, 0);
        light2 = '0';
}

if (ch == 'G') {   // LIGHT2 ON
        PIOconfig(0, PIOCFG_OUTPUT, 1);
        light2 = '1';
}

if (ch == 'h') {   // LIGHT3 OFF
        PIOconfig(1, PIOCFG_OUTPUT, 0);
        light3 = '0';
```

```
}

if (ch == 'H') {   // LIGHT3 ON
        PIOconfig(1, PIOCFG_OUTPUT, 1);
        light3 = '1';
}

if (ch == 'i') {    // PT_UP STOP
        PIOconfig(31, PIOCFG_OUTPUT, 0);
}

if (ch == 'I') {    // PT_UP
        PIOconfig(31, PIOCFG_OUTPUT, 1);
}

if (ch == 'j') {    // PT_DOWN STOP
        PIOconfig(17, PIOCFG_OUTPUT, 0);
}

if (ch == 'J') {    // PT_DOWN
        PIOconfig(17, PIOCFG_OUTPUT, 1);
}

if (ch == 'k') {    // PT_LEFT STOP
        PIOconfig(18, PIOCFG_OUTPUT, 0);
}

if (ch == 'K') {   // PT_LEFT
        PIOconfig(18, PIOCFG_OUTPUT, 1);
}

if (ch == 'l') {    // PT_RIGHT STOP
        PIOconfig(19, PIOCFG_OUTPUT, 0);
}

if (ch == 'L') {   // PT_RIGHT
        PIOconfig(19, PIOCFG_OUTPUT, 1);
}

if (ch == 'm') {  // SONAR OFF
        PIOconfig(27, PIOCFG_OUTPUT, 0);
}

if (ch == 'M') {  // SONAR ON
        PIOconfig(27, PIOCFG_OUTPUT, 1);
}

idx = 0;
if (ch == 'n'){{   // DAC CHANNEL 0
        q = getch();
        {while(q!='x'){
```

```
                dim[idx] = q;
                q = getch();
                idx++;}}
j = atoi(dim);
level_1 = j;
light_1();
// DUMP VALUES
j = 0;
idx = 0;
while (idx < 4){
dim[idx] = '\0';
idx++;}}}
if (ch == 'N'){{  // DAC CHANNEL 1
        q = getch();
        {while(q!='x'){
                dim[idx] = q;
                q = getch();
                idx++;}}
j = atoi(dim);
level_2 = j;
light_2();
// DUMP VALUES
j = 0;
idx = 0;
while (idx < 4){
dim[idx] = '\0';
idx++;}}}
if (ch == 'z') {
ADC();
funnel_sensor();
compass();
compass2();
compass2();     // MUST RUN TWICE

// H,P,R,T  D   L       C
printf("$%s%c%s%c%c%c%c%c%c%c%c",
s_compass,      // OS5000
'D', s_depth,
'L', light1, light2, light3,
'C', cam1, cam2, cam3);

//      J   F   r   S   A   N   i   I   a   v   W
printf("%c%s%c%s%c%s%c%c%c%c%c%s%c%s%c%s%c%s%c%s%c%c\n",
'J', s_jet,
'F', s_funnel,
'r', s_ram,
'S', sonar,
'A', alt_stat,
'N', s_compass2,// 0s5000 PITCH ONLY
'i', s_encoder,    // ROTARY ENCODER
'I', s_lps,        // LINEAR SENSOR
```

```
                              'a', s_altimeter,
                              'v', s_volts,        // SUBVOLTS
                              'W', leakFlag);
                              // SAMPLE "$H180P-5.2R-
2.56T20D100L111C111J10F0001r10S1A1N100i100I100a100v1W0"
                              }
                    ch = 0;
                    }
          }

}/*end of main*/
```

Appendix Two List of Figures

Figure 1…Hydroplow in Emden, Germany August 2013

Figure 2…Sound speed profile of water column, Jones, E. J. W., *Marine Geophysics* **,** page 33

Figure 3…Bathymetric/Topographic map of North Sea

Figure 4…Salinity, temperature, and sound spend profiles in water column, *Marine Geophysics* **,** page 33

Figure 5…Portside scheme of Hydroplow

Figure 6…Hydroplow of aft of vessel, North Sea August/September 2013

Figure 7…Scheme of hydroplow burial operations

Figures 8,9,10…Screen shots of 3d display from GUI

Figures 11…Block Diagram of telemetry system, original work

Figure 12…Microcontroller and OS5000 digital compass

Figure 13…Analog to digital channel map of telemetry system

Figure 14…Microcontroller code sample for analog to digital conversion

Figure 15…Screen shot of GUI

Figure 16…Hydroplow operator station/control room

References

[1] Apel, J. R., *Principles of Ocean Physics,* Academic Press Inc., Orlando, Florida (1987).

[2] Jerlov, N. G., *Optical Oceanography*, Elsevier Publishing Company, New York, New York (1968).

[3] Jerlov, N. G., *Marine Optics*, Elsevier Scientific Publishing Co., Amsterdam, Netherlands (1976).

[4] Jones, E. J. W., *Marine Geophysics*, John Wiley & Sons, Ltd., West Sussex, UK (1999).

[5] Waite, A. D., *SONAR for Practicing Engineers*, John Wiley & Sons, Ltd., West Sussex, UK (2002).