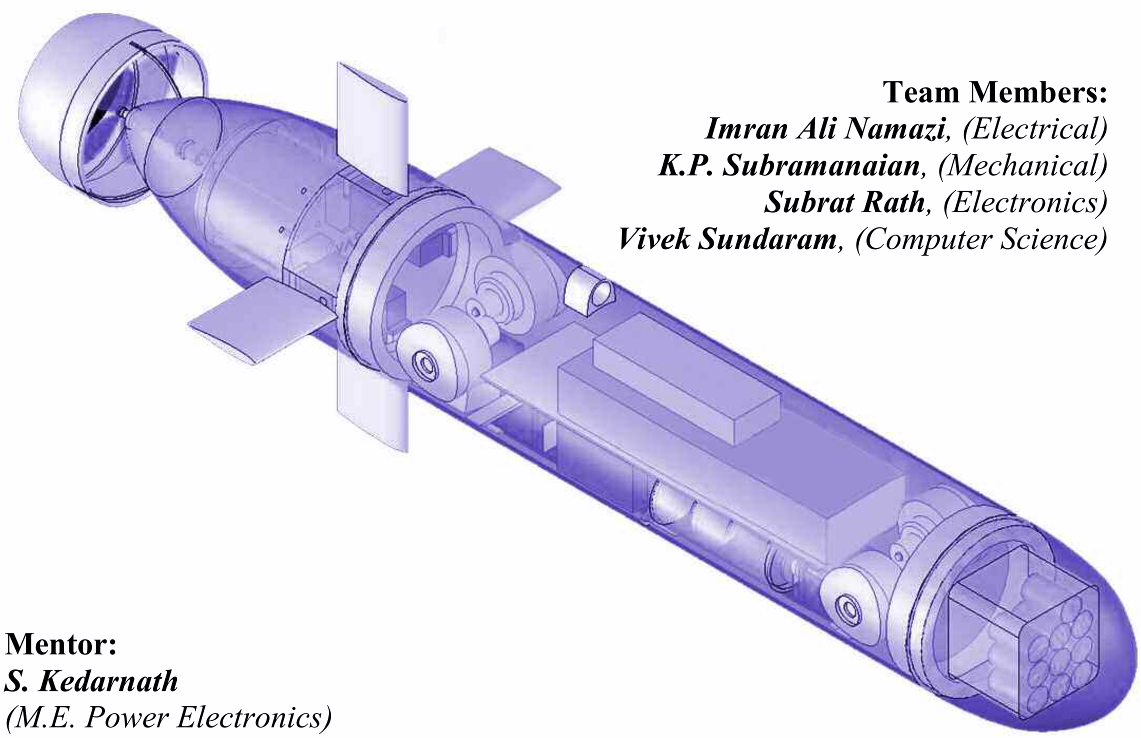
μ–Submarine

KCG College of Technology (affiliated to Anna University),
Chennai, India.



Team Members:
***Imran Ali Namazi**, (Electrical)*
***K.P. Subramanaian**, (Mechanical)*
***Subrat Rath**, (Electronics)*
***Vivek Sundaram**, (Computer Science)*

Mentor:
S. Kedarnath
(M.E. Power Electronics)

www.geocities.com/ianamazi/delphi

Abstract:

Technology has sky-rocketed as predicted since the times of Jules Verne. However, submarines have not yet been embraced by science to the extent they deserve. Design and manufacturing capabilities are contained within the technological fold of a mere 8 elite countries around the world. The years of research and billions of dollars required for fabrication has kept the rest of the nations from bringing this technological marvel into wide spread existence. Since these factors will always remain a juggernaut, a cost effective solution is indispensable to enable everyone to explore the enigmatic oceans, for commercial and defense purposes.

Delphis (Greek predecessor to the word **Dolphin**), serves practically every need for underwater activities, from **exploration** to **defense**. With features ranging across several engineering disciplines, such as **Sonar Mapping**, **Sonar Navigation**, **Contouring**, **Global Positioning System**, **Sound Recognition** (esp. for **Mine Detection**), **E**mergency **R**ecovery **S**ystem, Rescue Beacon, the project requires extensive research and brain storming. Hence, participation of 4 members from starkly contrasting fields. The Delphis model is directly connected to the **e-Box,** which acts as the sole center for processing the data provided by the **Multi Utility Submarine (μ Sub)**.

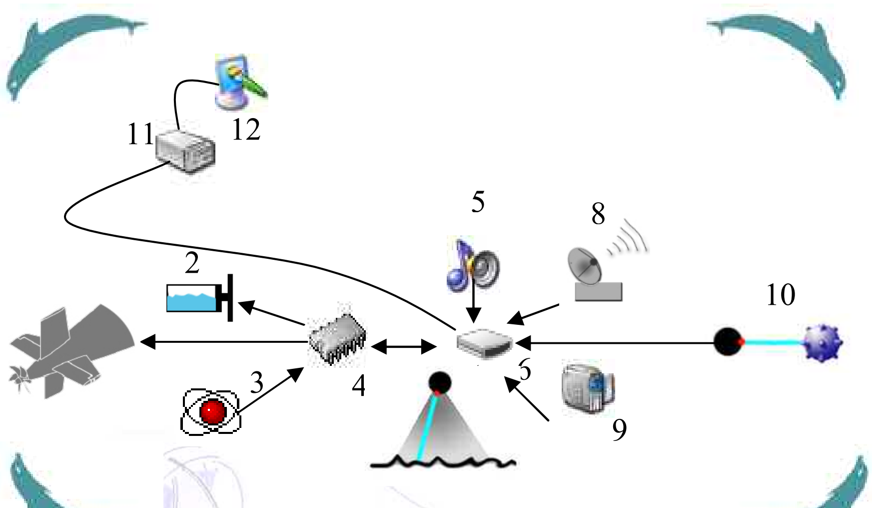
For the purpose of proving that this project is a requirement for the present day industries, we conducted a thorough study of several areas. Following is what we observed:

- The **Bosphorous Canal,** between the Black Sea and the Sea of Marmara, is one of the most heavily shipped lanes in the world. As the canal is between two major seas, there is enormous shifting of silt. So, if an accurate **Hydrographer's Map** of sand traps is not available, which often is the case, the consequence on a passing ship could be disastrous. Many tankers **have grounded** and caused oil spills.
- During the **Iran-Iraq War** from 1980-1988, several areas of the Persian Gulf and nearby waters were mined. On April 14, 1988, the **USS Samuel B. Roberts (FFG-58)** struck an M-08/39 mine in the central Gulf shipping lane, wounding 10 sailors. During the Gulf War, Iraqi naval mines severely damaged **USS Princeton (CG-59)** and **USS Tripoli (LPH-10)**. There have been many similar tragedies because of inefficient de-mining operations.
- Current methods of **mine disposal** are not effective enough for surface and sub-surface ships. Present day ROV's are incapable of being deployed from submarines as they would not fit inside the torpedo hatch.
- Ships piloting through **harbours** require **precise locations** of the various wrecks present in order to steer clear of them. Also, the enormous amount of silting at harbours needs **constant monitoring** to prevent crafts from running aground.
- Most underwater analysis and de-mining are performed by **divers**, thereby putting their lives in **grave danger**.

Delphis is meant to deal with such situations as well as the following:

- Underwater **Search** and **Rescue Beacon** for alerting Lifeguards.
- Nature Study (**Noiseless,** doesn't cause ecological disturbance).
- Oceanographic Study (like **Volcanic Activity**).
- Geological Study (**Topography** of areas such as fjords, due to **high maneuverability**)

1. System Overview:



Architecture of μ-Submarine

1. Rudder, Elevator, Propeller
2. Piston Tank
3. Instruments (Depth / Gyroscope)
4. Micro-controller
5. Piezo-electric Hydrophone
6. Modem (RF)
7. Scanning SONAR
8. GPS Receiver
9. USB Camera
10. Forward SONAR
11. e-Box
12. Display

Overall Design:

Our design methodology follows a **Bottom-up approach**, where every module is segregated into sub-systems depending on the field they belong to, namely, **Mechanical**, **Electrical & Electronics**, **Communication** and **Software**. Considering the **diversity** of fields of expertise required for a project of such proportions, this kind of an approach is inevitable. Collecting all the tasks at hand with respect to each field, their goals and performance requirements are as follows:

Mechanical Sub-System:

There were **4 major issues** that were to be considered prior to design of the μ-Submarine:

- Would design achieve neutral buoyancy?
- Was theoretical drag force negligible?
- Would material used withstand enough pressure?
- Was material easy to mould and cost effective?

In order to address all these questions, we chose **Poly Vinyl Chloride** as the core material. The hull should withstand up to 140 p.s.i. (**P**ounds per **S**quare **I**nch), for the submarine to be eligible to go as deep as 100 feet. The front nose cone alone will be fabricated out of **GRP** (**G**lass **R**einforced **P**lastic) in order to house the forward SONAR module. For the scanning SONAR, a GRP window is fitted to the bottom of the hull. Servo motors are used for maneuvering. A 16000 RPM DC motor is used for propulsion. A **retractable winch** for the GPS antenna is included in the mechanical design.

Electrical & Electronics Sub-System:

This involves **instrumentation feedback**, **drives** for the various components of the mechanical sub-system, control of the **ultrasonic equipment** and a **stepper motor.**

- Instrumentation constantly provides information on the status of the entire system. For e.g., a **gyroscope** is used to return **pitch** (Inclination) and **Yaw** (Magnetic Orientation).
- **PCM** (**P**ulse **C**ode **M**odulation) control is required for the servomotors as well as the drive for the motor that propels the submarine.
- The SONAR ranging system comprises of a coupled **emitter & sensor**, **amplifiers**, **heterodyning** circuitry for beating ultrasound to a workable frequency and a subsequent **window detector** to detect the arrival of an echo or "**ping**".
- The **stepper motor** and its corresponding drive circuits are essential for performing an **ultrasound scan**.

Communications Sub-System:

The requirement of communications from the Submarine (which might be a mile away) to the e-Box, is a task in itself. While considering the method of communication, we came across the following issues:

- Would the system work under water.
- Was it **flexible** enough, otherwise maneuverability would be hampered.
- Was there sufficient **bandwidth** to send all the instrumentation, audio and video data?
- Would it work with a **USB bridge** (for the camera)?

Communication via **wired FM** seemed to be the only answer to all these questions. We will be using a **433 MHz** wired RF modem as the communication controller, with a co-axial cable as antenna. A **USB bridge** is built for the camera, transported via FM and controlled by the host USB controller on the **Windows CE Operating System**.

Software Sub-System:

The Windows CE Image built on the e-Box will be the only controller of **Delphis**. There is an issue of computing power when all the work of interaction with the submarine as well as processing of incoming data (esp. Sound Recognition) has to be borne by a 200 MHz processor. But since the criterion was to use the e-Box for the project, we decided to optimize the software for the provided hardware. The following are the Software components of Delphis:

- **User Interface** & **Comprehensive control** over the maneuverability of the Submarine.
- Ultra Sound **mapping** & **Archival** software with **3D view generator**.
- **Sound Recognition** module.
- **Black Box** module (includes **GPS** control).
- Driver creation for **Serial**, **Audio-in** and **USB**.
- **E**mergency **R**ecovery **S**ystem (To be deployed onto Delphis).

The formidable task of evolving the various modules was performed across a plethora of programming tools such as **VB .NET**, **EVC++**, **VC# .NET** and **MATLAB**.

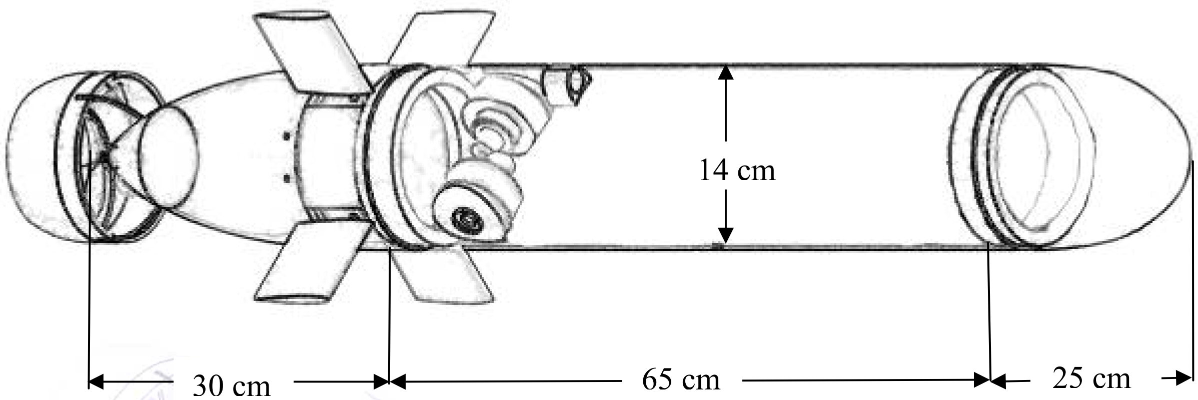
Innovative Design:

Though several countries have made mini-submarines, there are **none** capable of being deployed off **submarines**, or, for that matter, **equipped with GPS** that works without the need for surfacing. **E**mergency **R**ecovery **S**ystems and **Sound Recognition** are also unheard of. These systems are also not software based and there can be no image or **sonar library**.

2. Implementation and Engineering Considerations:

2.1 Mechanical Design:

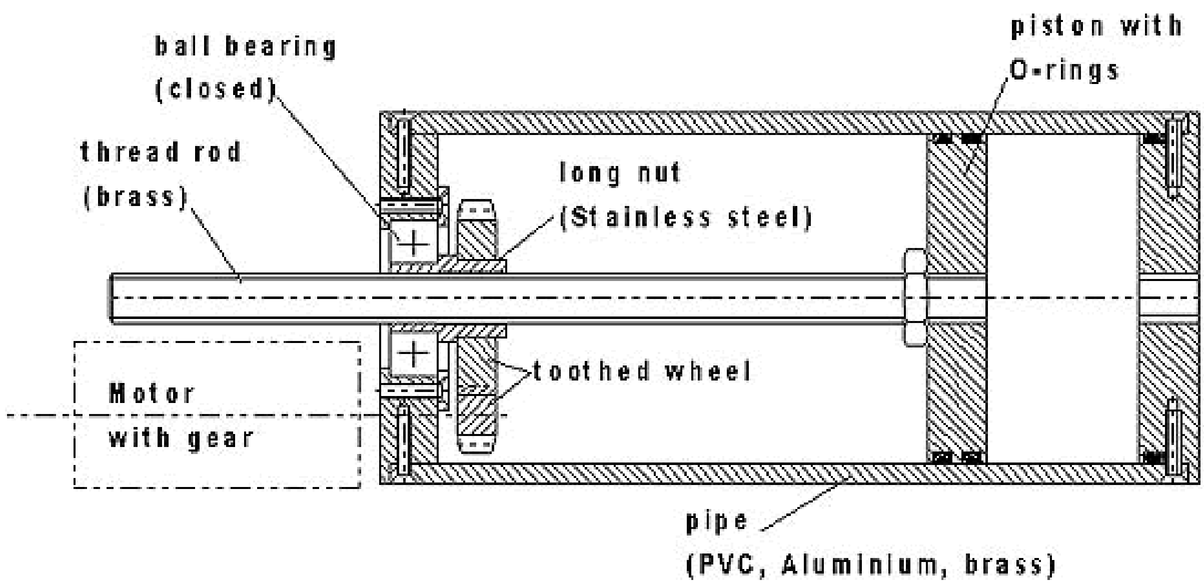
2.1.1 Hull Design:



Hull Specifications

The hull design chosen for this model is the **Myring hull contour** and will be made of **Poly Vinyl Chloride** of **4 mm thickness.** It has two compartments, namely, Nose-cone and Tail-piece; both these sections are joined to the centre by means of a pneumatically operated **bayonet lock,** and have got circular **GRP bulkheads**. In addition to that, the tailpiece has a bulkhead made of resin .The base dimensions of these sections are shown above.

2.1.2 Piston Tank:



The Piston Tank

Diving of a submarine by just **filling its tanks**, which does not require forward propulsion for remaining submerged is called **static diving**. The Delphis model must be capable of static diving; otherwise it will not be able to **hover underwater** at any particular depth. Hence, it has many advantages over a submarine that has to move itself forward to achieve diving capability (dynamic diving).

To precisely **control buoyancy** and diving, we use a **piston tank**. Although a compressed air system may be used to fill and blow the tanks, availability of air would be limited and compressors cannot easily be fitted on such a small submarine, so, using a piston tank is the best method to dive our submarine.

Specification:

Operating Volume	825 ml
Cylinder outer diameter	75 mm (2.95")
Cylinder length	265 mm (10.4")
Length overall min./max (full/empty):	347/505 mm (13.7/19.9")
Filling time approx.	18 Sec.
Burst pressure of cylinder	11 bar

Drive Motor:

Operating voltage	12 V
Voltage range	6-24 V

2.1.3 Control Surfaces:

The control surfaces (used for the rudder, elevators and bow-planes) were chosen with great care as they have to withstand the pressure underwater, typically at least **140 p.s.i** (**P**ounds per **S**quare **I**nch). For maximum efficiency and maneuverability, a standard **NACA 0012** airfoil was chosen. These surfaces will be made of resin, embedded with Aluminum for extra strength.

Specifications:

Chord: 5 cm.
Span: 3 cm. long
Servo shaft: 1/8 in.
Resin-Embedded Aluminum: excellent strength to weight ratios.
Servo selection: 15 deg. Control Surface deflection in 0.5 sec using Futaba™ S3003 servomotor.

2.1.4 Propeller:

The design and material of the propeller was also chosen with great care as it represents the mainstay of power. For maximum speed and minimum sound disturbances; we chose to use a 6 bladed brass scimitar propeller which is 2 inches in diameter. The reasons for our choice were:

1. **Reduced Vibration**
2. **Reduced Cavitation**

2.1.5 On board Power Systems:

The Delphi model will have standard 12 V Ni-Cd battery packs that will be mounted as close to the centre of gravity as possible to ensure proper trimming and stability.

Status of Fabrication:

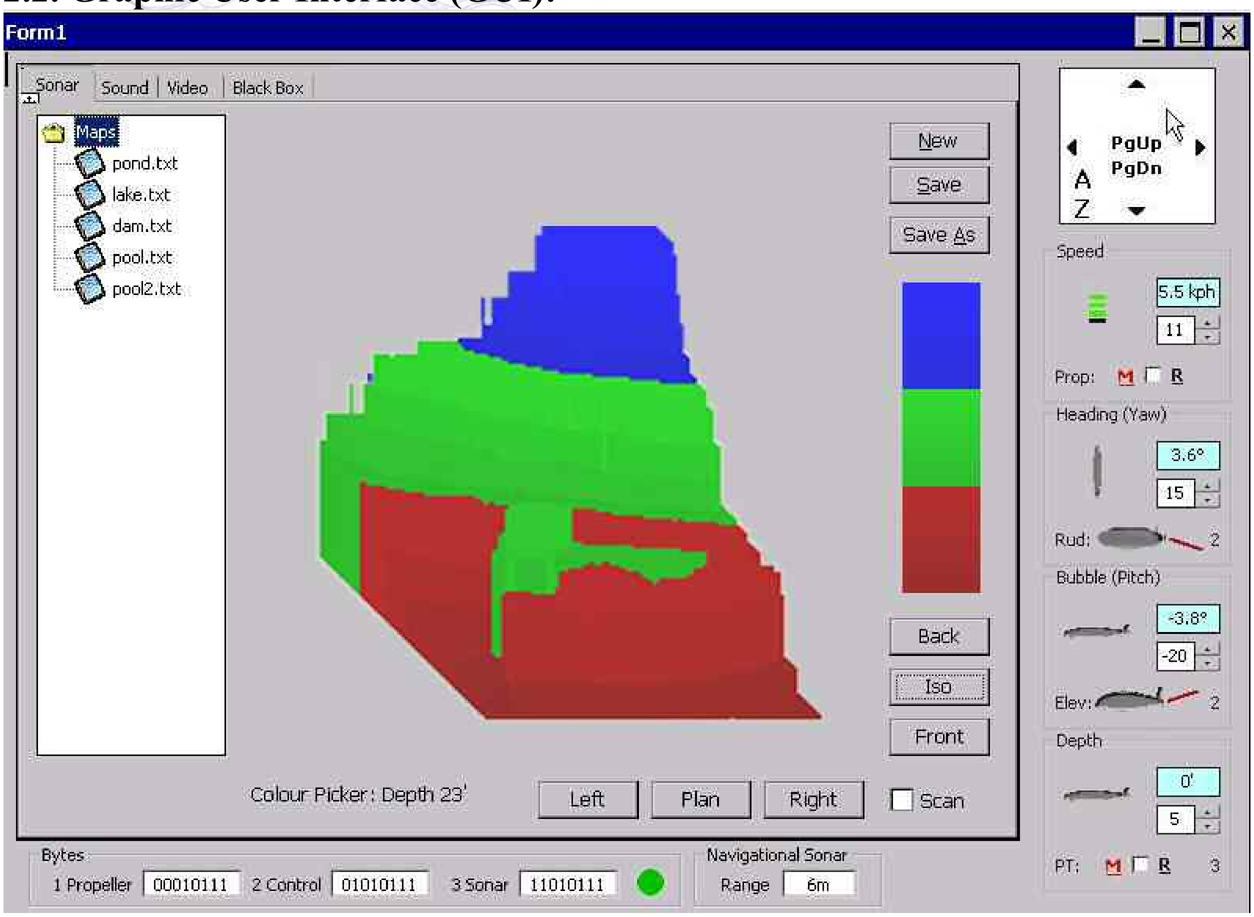
We have recently built and tested a **working, diving prototype** (right) made of GRP. Though it was designed for another purpose and cannot be used with the systems we are currently developing, it has proven invaluable to us in our mechanical design, and we are confident that **another system** can be built and deployed **within ten days**. The design is ready in all respects.

Our First System

Design Alternatives Considered:

1. **Steel** or **Aluminum** for the hull were considered, but cost, machining time and base workability put a stopper on those ideas. **G**lass **R**einforced **P**lastic was found to have excellent strength, but was expensive.
2. A "pick and place" **robotic arm** was considered and dropped on the grounds of fabrication time and cost.

2.2. Graphic User Interface (GUI):



The User Interface *(as viewed on emulator)*

2.2.1 Control System:

The Navigational control (Right Pane), consists of a **Virtual Joystick**, which is used to control **Throttle**, **Heading**, **Depth** (through Elevators) and **Buoyancy** (for Diving). Joystick input will change the set values for that control. To effect this change, the corresponding drive is turned on and will be propagated down the system to maneuver the submarine. When it reaches the set value (perceived via instrumentation), the particular drive shuts off. The HUD would prompt any **anomalous values** or values that would be harmful to the **submarine's performance**, such as **steep dive angle,** by flashing red.

2.2.2 SONAR:

2.2.2.1 Map Drawing: The map produced by the SONAR module can be viewed by selecting the SONAR tab. The **Depth Sorting Algorithm** is used to represent the contours in **six unique** views **viz. Front, Back, Left, Right, Plan and Isometric**. Each view is essential to observe variations in the elevation of the sea floor. The Isometric view provides a good **3D** perspective of the entire map. Each value of depth is assigned a particular color code (from red through green to blue) to enhance visual comprehension. The maps thus generated can be saved and loaded as and when required.

2.2.2.2 Navigational Sonar: This is the forward seeking SONAR. It measures the open area in front of the Submarine in order to avoid head-on collisions with objects and relief.

2.2.3 Sound:

The "Sound" Tab is activated in order to **Listen, Record, Play and Analyze** sounds produced under water and will be dealt with in detail in the section on **Wave Recognition**.

2.2.4 Video:

The "Video" Tab provides visual feedback of the **submarine's environment**. Useful in **navigation** and the observation of **geographic formations & aquatic organisms**, this component is invaluable to underwater study of any kind.

2.2.5 Black Box:

The "**Black Box"**, a creation inspired by those found on **airplanes**, is a **virtual device** that constantly records the position of the submarine. It can be invoked while **mapping contours** as well as during **general exploration**. It is tied to the **Global Positioning System**; hence becoming an integral part of **rescue operations** as well as **recovery** of the submarine should there be a failure of the communication link.

Performance Measurement & Testing:

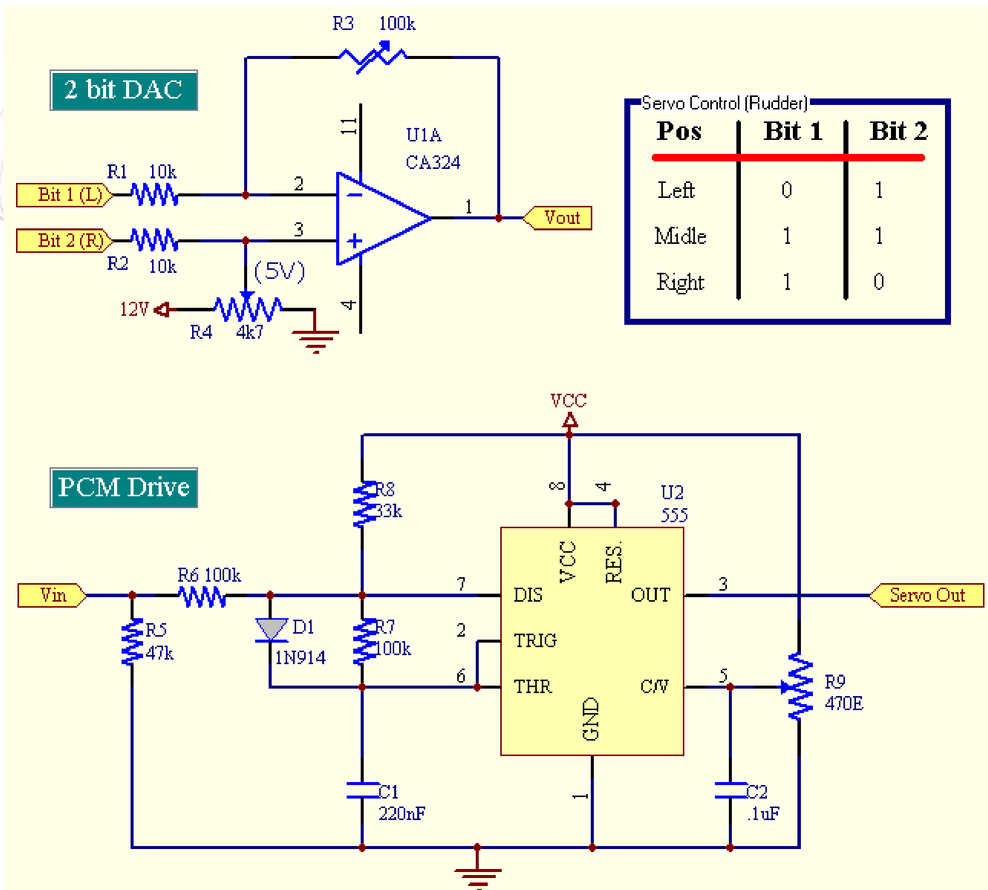
The performance of the UI completely relies on the processor capabilities. With about 71 controls on screen (during sonar mapping), there is a significant performance slow down. The load time (on the emulator) is 5.3 seconds and map drawing requires another 8 seconds. **This module has been completed**, and we are working on **optimizations** and **reduction in number** of controls for better performance on the e-Box.

2.3 Microcontroller:

The on-board electronics is built around an 8051 which is the main link between the software and the submarine. Around it are the various **drive circuits** and controls for ultrasound and the instruments. A "**battery low**" detection circuit triggers the **E**mergency **R**ecovery **S**ystem.

2.3.1 Servomotor Control:

Having chosen to reduce design complexity by using set angular steps (30º each side) for the **rudder** (and elevators), we had to design a way to translate the bits from software to 3 voltage levels (3.5,5 and 6.5V) that would be encoded by the standard PCM drive (555) to get the corresponding servomotor positions. For this we evolved the **2bit DAC** design shown below. The PCM drive works with a **standard 0-10V analog input** and by varying the charge time of C1, output's T_{ON} will vary from **1.25 to 1.75mS** as required by our servomotor.



Servomotor Control

2.3.2 Diving Control:

The **Piston Tank**, used for diving, is driven by a single DC motor that will be turned on using **two relays** (one for direction). Control has been implemented in the GUI. **Status** of the Piston Tank (whether it is empty or full) is available through a set of **micro switches** that are closed when the piston is at the corresponding end. This information is read and relayed by the micro-controller.

2.3.3 Instruments – Pitch & Yaw:

For instrumentation, what we've chosen to do is to **reverse engineer a mouse**. This gives us reliability and simplicity. The optical wheels inside the mouse will be mechanically coupled to a **compass and a pendulum** (for **Yaw and Pitch**). This information is read through Port C of the 8255, used for serial to parallel conversion (SIPO) and is sent directly to the software for interpretation.

PS/2 Mouse Information[1]:

	D7	D6	D5	D4	D3	D2	D1	D0
Byte1.	XV	XV	YS	XS	1	0	R	L
Byte2.	X7	X6	X5	X4	X3	X2	X1	X0
Byte3.	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

L, R	Left & Right button state (1 = pressed down)
X0-X7	Movement in X direction
Y0-Y7	Movement in Y direction
XS, YS	Movement data sign bits (1 = negative)
XV, YV	Movement data overflow bits (1 = overflow has occurred)

Resetting of this instrument must be done each time the submarine is powered on (like centering the mouse pointer on Windows startup), as readings will be taken with reference to this initial position. Calibration is a simple process and we expect resolution of within one degree.

2.3.4 Instruments - Depth:

Depth is perceived as pressure on the hull. This will cause a minute deformity of the PVC and can be sensed via strain gauges rigidly fixed to it. A **pair** of strain gauges is used to compensate for **temperature variation**. Using signal conditioning circuits, the **changes in their resistances** (as a function of strain and ultimately depth) will be converted into a corresponding analog signal. The 8255 (also used in conjunction with the mouse) will slave an **8 bit ADC** that will digitize this analog input. This information is relayed directly to the on-screen instrument.

Alternative Designs Considered:

1. Instrumentation is a vast field and the number of design options we faced was numerous. The compass and pendulum basically provide **angular displacement** and this had to be sensed somehow. Some of the alternatives considered were: a) Galvanometer, b) Optical sensors c) Mercury Switches d) Charge Coupled Device.
2. A tricky question was the **sensing of depth**. Theoretically, we could simply use a **U-Tube manometer** with an optical sensor. The sensing posed unnecessary complication concept is not amenable for operation in confined spaces. Another option considered was to use a column of air and a port (water inlet) below. The idea was to make the water pressure compress **air-filled bellows**. In theory, this system could be adjusted to give us **control over linearity** of instrument as well as range. However the fabrication and mechanical considerations for such systems were too complex and too many experiments would have had to be performed to implement either of these options.

[1] PS/2 mouse byte information was obtained from http://www.hut.fi/~then/mytexts/mouse.html

2.4 Navigation and Control:

Crucial to the operation of our submarine is the **Navigation** and **Control** system. The user gets **feedback** from the instruments on board, and with the accurate knowledge of location provided by the black-box; he determines the course of action and inputs it on screen. Instruction from software to the submarine, for navigation, is in the form of **2 control bytes** as:

		Prop. Speed (8 | 4 | 2 | 1)	
Byte 1	0 | 0 | X | R | 8 | 4 | 2 | 1	(Propeller)	
	Byte ID (0 | 0); Direction (Reverse) (R)		
Byte 2	0 | 1 | F | B | U | D | L | R	(Drives)	Piston Tank: Front,Back Elevator: Up, Down Rudder: Left,Right
	Byte ID (0 | 1)		

Control Byte Format

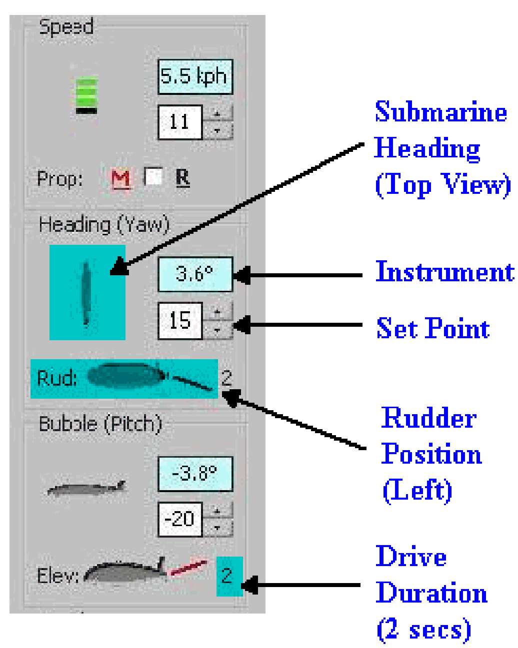
These bytes are written to the micro-controller each time there is a change. Testing of this has been done using **VB6** and the **MSCOMM** control. Drivers have been written for Windows CE

A case in study is the rudder control. As seen from the Control Pane (below), the Yaw is currently 3.6º. Set point however, is 15º. The rudder is being moved to the left to effect this change. For this to take place, the corresponding L & R bits should be set to 0 & 1 respectively (as per the truth table for the 2 bit DAC dealt with in the servomotor control section).

The rate at which the yaw changes is a function of rudder angle (30º) and propeller speed (5.5kph). These calculations have been taken into account for simulation of the instrument response and the entire control system.

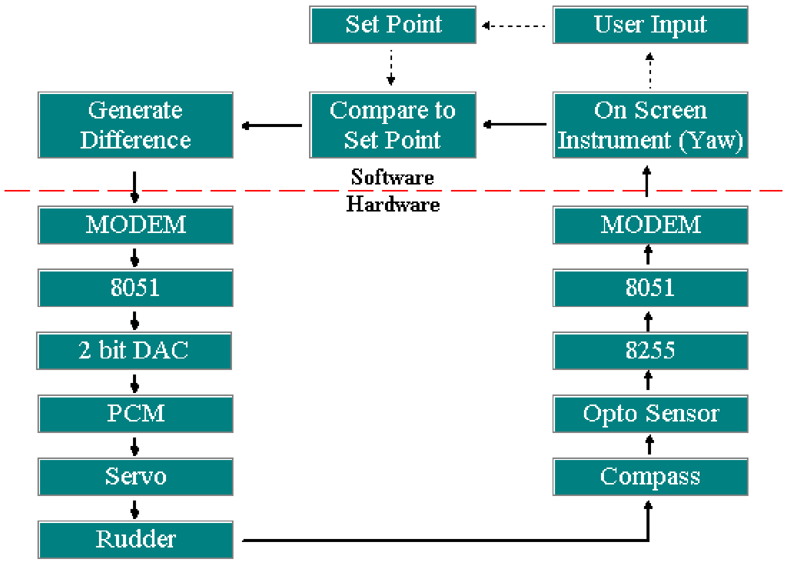
Also seen from the Control Pane is the set speed of the propeller, the value 11 (set point 1011_B) is the same lower nibble of Byte 1 of the control word. Direction is forward (as seen from the 'R' Checkbox, so the R bit (D4) is 0.

The elevator works in similar fashion, with the controls trying to bring it to 20º down bubble.



Navigational Control

As explained above in the screen-shot of the UI, our submarine is trying to track the changes made to the set points for yaw and bubble by controlling the corresponding servo motors. This operation is a closed loop as illustrated in the flowchart shown overleaf.



Flowchart for Rudder Control

For completing the **simulation** of this vital component of the software, we hard coded the response of our entire submarine using the following assumptions:

- Depth: One foot change for every 3 seconds of Piston Tank Running
- Elevators & Rudder: Angle = 2.5 x Speed / 7.5
  2.5º is the angular change per second for Flank Speed (7.5kph)
- Propeller: Acceleration .5kph/Sec. (This value doubles upon sudden reversal.)

Status of Completion:

Defining this system, designing the DACs, determining optimal control angle (30º) and outlining the byte format was the hard part of design. In various stages, we worked the PCM drive, the Serial Port communication, the propeller drive and designed the controls for diving.

For instrumentation, the micro-controller will relay the data from the mouse module, the strain gauge and the limit switches of the piston tank via Port C of the 8255 Programmable Peripheral Interface. This relaying will be paused while listening for an echo (on either sonar module) and when the "window detectors" find an echo, it will immediately be transmitted via an Interrupt Service Routine to keep the delay of the system a constant

2.5 Sonar Mapping:

The ability of Delphis to map the sea-floor elevation with an accuracy of 1 foot is one of its highlights. The reason why SONAR becomes a crucial component is that mine-detection is also made possible, hence complementing Sound Recognition. Our SONAR module works on the popular concept of echo-sounding, where the reflected ultra sound beam is sensed A window detector captures the "ping" and a timer (software) determines the duration of the rebound. Using time, the depth is calculated.

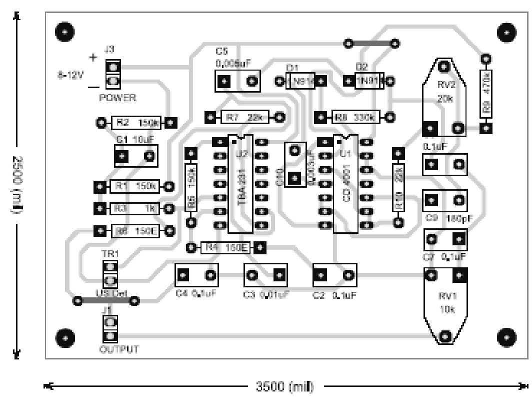
2.5.1 Coupled Emitter & Receiver:

We are using a MURATA 40KHz Matched Ultrasound emitter and receiver, housed within a GRP window. When the emitter is triggered, an ultrasonic beam is emitted for a specified duration. After reflecting of a surface (closest), the sensor picks up this signal and passes it on to the heterodyne. The sensor detects only the strongest reflection and hence multiple rebounds are ignored.

Ultrasound Emitter/Receiver

2.5.2 Need for Heterodyning:

Since ultrasound deals with frequencies above typical operating frequencies, in order to detect the "ping" by a standard window detector (using opamps), the frequency ought to be down-sampled to a workable value. For this purpose, **heterodyning** is required. We have designed and built a circuit which includes an in-built **two-stage amplifier**. The Top Overlay of the PCB is shown on the right.



Overlay of our Receiver PCB

2.5.3 Window detector:

After amplification and heterodyning, the exact point at which the ping occurs need to be detected and a flag bit is sent as soon as detection is done. The window detector consists of an IC 741 Op-Amp along with Resistor-Capacitor bridge networks to modify accepted frequency and threshold amplitude. A signal from the sensor that surpasses this threshold is considered to be a ping. A software timer on the e-Box picks up this ping, calibrates the offset for propagation delay and returns time for rebound.

2.5.4 Mapping algorithm:

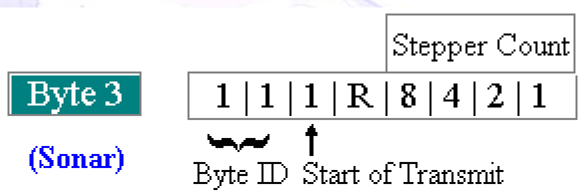
The following algorithm produces a 100 * 100 Matrix in order to calculate co-ordinates of the scanned points. The algorithm has a complexity of **Θ(n)**, where n is the number of mapped points. The algorithm **has been implemented** and is ready to be deployed on the **E-box** (**testing completed on emulator**).

Vz [][]	null	//Matrix to store Map
Heading	0	//Angle of taken by Sub w.r.t map
Speed	1000	//Speed of ultrasound beam (1000 feet/sec)
Row	50	//Starting point (X-axis)
Col	50	//Starting point (Y-axis)
Angle[]	Stepper Angles	//Angle of Stepper motor on which U/S is mounted
Depth	Curr. Depth	Current Depth of the Submarine from Strain Gauge

```
While map file exists
    Try
        Line ← Read line from file
        If heading<90 or heading>270
        Row ← Row + (l - col) / (Tan(PI / 180 * (90 - Heading)))
        Else
        Row ← Row - (l - col) / (Tan(PI / 180 * (90 - Heading)))
        End If
        Col ← Col + (line * speed * Sin(PI / 180 * angle) /
              Sqrt(1 + 1 / Pow(Tan(PI / 180 * (90 - Heading)), 2))
    Catch(Exception e)   //When angle is multiples of 90 (other than 0)
        If angle = 0
        Throw(e)
        Else
        Row ← Row ± (line * speed * Sin(PI / 180 * angle))
        Col ← Col ± (line * speed * Sin(PI / 180 * angle))
        // '±' Alternates for even and odd multiples of 90
        End If
    Catch(Exception e)          //When angle=0
        Row ← Row ± Sin(PI / 180 * angle)
        Col ← Col ± Cos(PI / 180 * angle)
        // '±' alternates for even and odd multiples of 90
    Vz(Round(Row),Round(Col)) ← Depth + (line / 200 * speed) * Cos(PI / 180 *
                                angle)
End While
```

2.5.5 Scanning Sequence:

This byte is written to the micro-controller at the instant of scanning. It turns on the emitter for a duration of 50mS, and turns the emitter to the next position by giving the stepper motor a number of pulses specified in the lower nibble and in the direction given by 'R'.



Control Byte for Mapping

Performance Measurement & Testing:

Except for propagation delay, this module has complexity that plainly depends on number of points mapped and time for file access (which is negligible too). **This module has been completed and is ready to be deployed.**

Alternative Designs Considered:

1. Initially we considered two sonar emitters and receivers that can scan simultaneously. But such a design requires two micro controllers working in conjunction. Due to space constraints and the need to reduce complexity, the idea was forgone.
2. The window detector was to be made as a software component through C++ coding. Every 50 pings were to be written as a WAVE file and analyzed for time delay. The

problem was, writing a WAVE file takes 2 seconds and reading it another 0.5 seconds. So during this time the submarine should be made to wait idly before resuming scan.

2.6 Sound Recognition:

Delphis uses sound recognition to detect mines when their mooring chains move to produce a characteristic **metallic clink**. It is programmed to identify such sounds and establish the source. Not only can mines be detected, but also any sound producing object

In order to analyze sound waves we have to first understand the two basic properties of it with respect to perception by any receiver:

- **Amplitude(**Intensity): The magnitude of sound energy
- **Frequency**(Pitch): The rate at which the wave repeats itself

The following analysis is done assuming that:

- The distance between the source and the receiver remains constant (in order to remove **Doppler effect**)
- The **temperature** remains constant (speed of sound waves varies with temperature)
- The **density of the medium** is constant

Amplitude: As far as perception of sound energy is concerned the intensity purely depends upon the distance between the source and receiver. If distance is less, the sound heard is louder and vice versa. Hence amplitude cannot be considered as the sole factor for analysis.

Frequency: Frequency on the other hand is a characteristic property of sound energy. It depends purely on the sound source and hence will be unique for each source.

Two mono-frequency sound sources can be adjudged different if they have different frequencies. However, the scenario for poly-frequency sources is a bit different. In this case we have to consider **the relative magnitudes** at different frequencies.

2.6.1 Algorithm for sound recognition:

The outline of the algorithm is given below:

Record the sound emitted by the source and then remove the noise components using an adaptive filter.

a[]	Array	//Handler for recorded WAVE file
f1[]	Array	//To Store frequency response of recorded sound
f2[]	Array	//Stored frequency response of sound in library
n1[]	Array	//To store frequency response of normalized wave
n2[]	Array	//Stored frequency response of normalized wave in library
result	*float*	*//Store comparison percentage*
a ⟵	*shift(a)*	*//Remove initial silence*
f1 ⟵	*fft(a)*	*//Obtain FFT*
n1 ⟵	*f1 / sum(f1)*	*//Normalize response by taking each values contribution the FFT*
n1 ⟵	*resize(n2)*	*//Bring first array to same size as second*

while ***array exists i 1:size(n1)***

if n1[i] > n2[i]

result ⟵ result + n1[i] / n2[i]

else

result ⟵ result + n2[i] / n1[i]

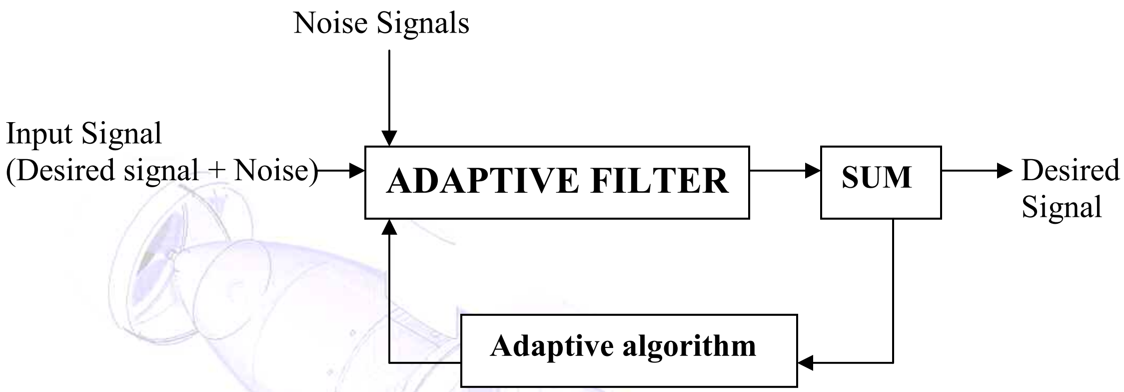
End While

Repeat above steps for recorded file with all files in sound library and Return max(Result)

2.6.2 Need for Filtering:

Recording is done with the help of a **hydrophone**. Underwater sound waves are coupled to the hydrophone by immersing it in a dielectric such as oil. Without this liquid, the sensitivity of the hydrophone will be largely reduced since sound travels much farther in liquids. The sounded, sensed as pressure variations are converted into electrical signals via a **piezo-electric transducer**. The recorded signal is sampled at a rate of **11025 Hz** and stored.

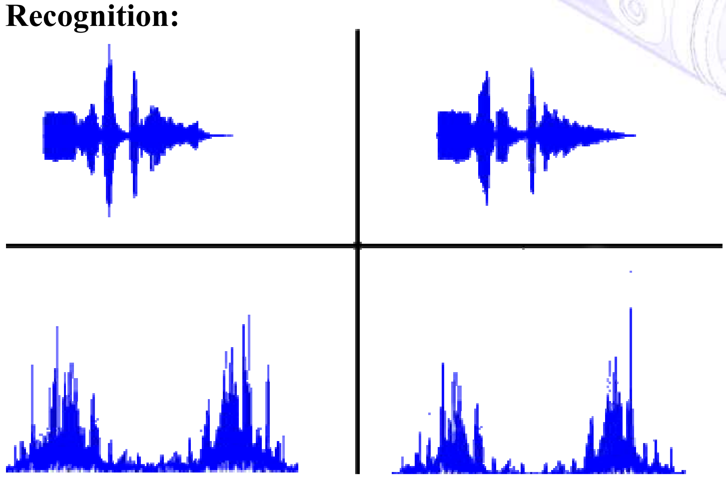
The environment in which recording is done will, besides containing the sound, also contain noise, which has to be discarded as their presence will affect the original signal. We have used adaptive filters to remove the noise components. Its functioning is as follows



Block diagram of the Adaptive filter

The noise expected from the environment is fed to the adaptive filter. The filter generates co-efficients and as the noise is removed the co-efficient values go on decreasing till they reach a minimum value predefined by us. This indicates that no more filtering of noise need be done. Training of the filter can only be done when no source is present. This recorded signal is nothing but the noise that will be present while recording the sound emitted by any source. The filter **learns to adapt** to the noise and hence its name.

Example for Wave Recognition:

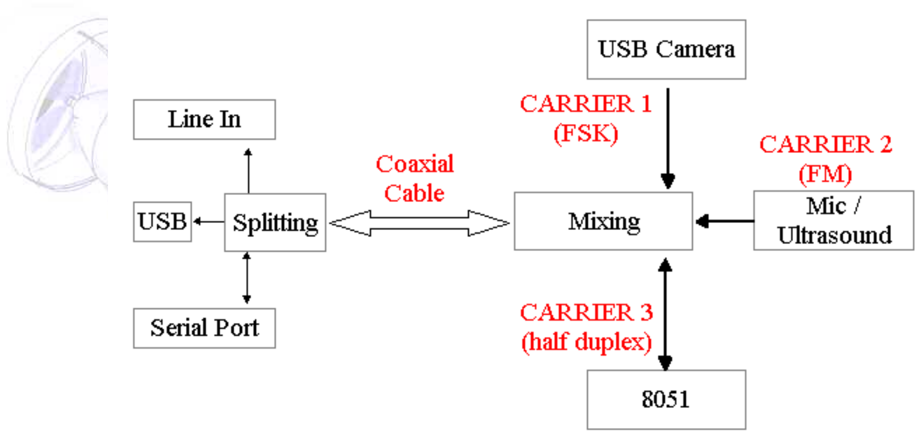


Amplitude and corresponding Fourier plots obtained by applying our algorithm to two similar sounds

Performance Measurement & Status:
In this method, speed depends a lot on the number of files to be compared with, the more the number of files less is the speed. But the fact that to get the best results maximum number of comparison should be carried out, cannot be ignored. In **extreme conditions**, with **40 different wave files** onboard, we obtained results on a **500 MHz processor**, in 3.2 **seconds**, using our own timers. It couldn't be tested on the **e-Box** since it didn't arrive on time. **This module has been completed and is ready to be deployed.**

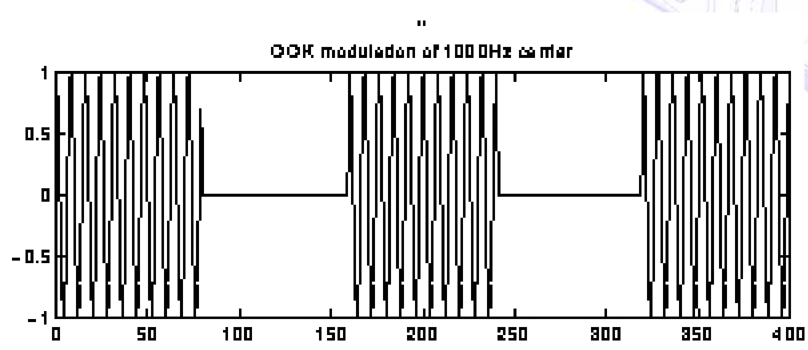
2.7. Communication:

The whole problem of communicating from the Submarine to the e-Box is a project in itself. With work on DSP successfully completed, it will be up to our electronics component leader to design a system that will meet these design specifications. The **USB Bridge** connects the USB camera to **Carrier1**, which is implemented on a **433 MHz** Transceiver.



Communication Block Diagram

For encoding of the digital signals (for the **RS 232 interface** and the **USB link**) we plan to use **On-Off Keying** illustrated in the figure below.

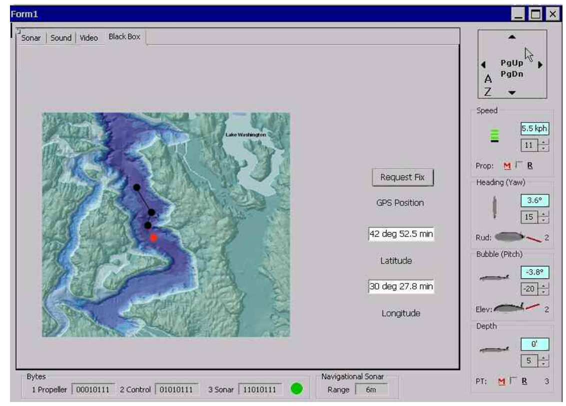


On-Off Keying (Byte 10101)

2.8 Black Box, GPS & Emergency Recovery System:

2.8.1 Black Box:

The black-box is a virtual device used to constantly **record position** of the submarine. It is tied to the GPS and has a separate Tab, where the current position of the submarine is represented in Red, and past positions in black. For better visual representation, the **geographical map** of the area may be loaded as background to our coordinate system. Request for **GPS fixes** are made here and require the submarine to come to a minimum depth so that the winch can **float the antenna**.



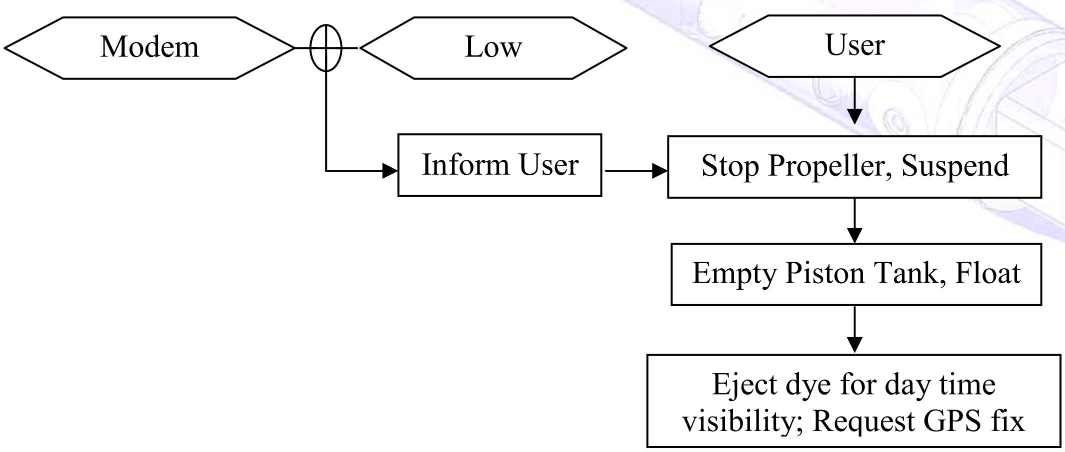
Black Box

2.8.2 Global Positioning System

The GPS system triangulates position via a minimum 3 GPS Satellites and is crucial to the integrity of our maps and black box operation. GPS fixes cannot be taken from underwater due to carrier attenuation. Negating the need to surface and disclose location, we have designed our µ-Sub with a retractable GPS antenna, housed in the conning tower.

2.8.3 Emergency Recovery System:

The Emergency Recovery System (ERS) is an essential function of Delphis, which prevents loss of submarine, should severing of the co-axial cable occur (detected on board by a connection time-out), or if battery supply is dangerously low. The process can also be initiated manually. The flowchart of the processes involved in this mode of recovery is as follows:



Flowchart for ERS

2.9 Role of e-Box:

The **Windows CE image** deployed on the e-Box will take **complete control** over Delphis. All processing software, including mapping and sound recognition, has been written for the Windows CE platform and tested on the **emulator**.

3. Summary:

The powerful features offered by the Delphis μ-Submarine, at an **affordable cost,** and in a **compact & robust** model makes it the ideal solution for many a problem currently faced by the maritime industry such as harbors, sea-goers, oceanographers and mining companies. The use of **embedded computing** on a scale never seen before will make this state-of-the-art creation a spectacular success. Enhancements can be easily made to comply with the needs of specific industry, since development is not based solely on proprietary hardware but is in the **software domain** and upgrades can be sought from third party solution vendors. Once the basic system is developed and its **versatility** a proven fact, user-specific development can be made for optimization of any particular task. Minor improvements to the performance, materials used and fabrication technology would make our craft **Military-Grade** and use as a tactical aid for real submarines in the defense of a nation.

While our first model focused on hydrodynamics and stealth, this Delphis Model is built as a full fledged **sea-worthy** embedded system with the following capabilities:

- ✓ Comprehensive Software Solution
- ✓ Navigation & Maneuverability
- ✓ On-board Instrumentation
- ✓ Mapping Capability
- ✓ Forward Seeking Sonar
- ✓ Underwater Deployable GPS Link
- ✓ Telemetry of Underwater Video
- ✓ Recognition of Sounds

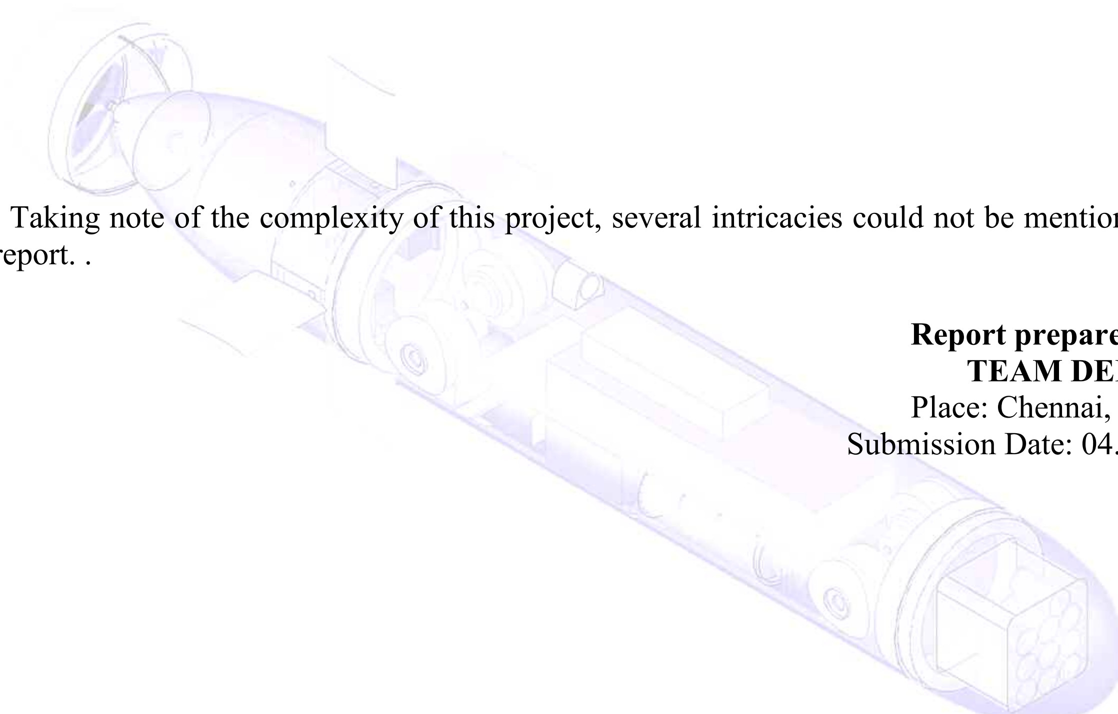
We are also looking to add the following features as further advancements to our second generation Delphis model. These would be implemented, **by June 16**, if time permits.

- **Landing gears** to enable the craft to ground on the sea-bed.
- Hydraulic drivers to extract up to two **sample cross-sections.**
- Three Hydrophones to **triangulate position,** enabling the system to determine the direction and range to the source of the sound, in addition to its nature (cognition).

Involving ourselves in diverse and exciting technologies such as this, developing on cutting-edge platforms and trying to leverage as many engineering fields as possible, we hope to continue to work in this area for a long time to come.

4. References:

1. John Sharp, Jon Jagger: Visual C# .Net – Step by Step
2. Cooley, J. W. and J. W. Tukey: "An Algorithm for the Machine Computation of the Complex Fourier Series," Mathematics of Computation, Vol. 19, April 1965, pp. 297-301.
3. Duhamel, P. and M. Vetterli: "Fast Fourier Transforms: A Tutorial Review and a State of the Art," Signal Processing, Vol. 19, April 1990, pp. 259-299.
4. Haykin, Simon, Adaptive Filter Theory, Prentice-Hall, Inc., 1996.
5. Scott Meyers: Effective C++, Addison-Wesley,1997.
6. CC1010 Single Chip Very Low Power RF Transceiver with 8051-compatible Microcontroller: Chipcon AS, 2003.



NB: Taking note of the complexity of this project, several intricacies could not be mentioned in the report. .

Report prepared by:
TEAM DELPHI
Place: Chennai, India.
Submission Date: 04.23.05