Apparatus and System For Search and Rescue Operations At Sea

FARHAD SHAMS

Master of Science Thesis
Stockholm, Sweden 2011

TRITA-ICT-EX-2011:23
Apparatus and system for search
And rescue operations at sea

Master Thesis
Last Revised: 2011-06-8

Farhad Shams
farhads@kth.se

Examiner & Supervisor
Professor Mark T. Smith
ABSTRACT

The aim of this thesis is to design and implement of an independent rescue system that includes a transmitter part which can be embedded in the life jacket of boat passengers and a receiver part. The system includes GPS, Radio besides a 3-axis magnetometer in order to set the heading of receiver part to transmitter part. The overall system has been designed with some feasible tests in order to support proposed system in completeness and effectiveness.
## Contents

1. Introduction ..................................................................................................................... 1  
   1.1 Research Purpose ................................................................................................. 1  
   1.2 Problem statement ............................................................................................... 1  
   1.3 Existing Products .................................................................................................. 3  
      1.3.1 ACR's Mini B 300 ILS H2ON ........................................................................ 3  
      1.3.2 Alert 2 Man-Overboard Alarm: ..................................................................... 3  
      1.3.3 Life Tag Wireless Man Overboard Monitoring System .............................. 3  
      1.3.4 Mobilarm V100: ............................................................................................ 3  
      1.3.5 First Light Solutions mermaid id: ............................................................... 4  
   1.4 Target Specification ............................................................................................... 5  
   1.5 Research Methodology .......................................................................................... 7  
2. Hardware Overview ........................................................................................................ 8  
   2.1 Transmitter ........................................................................................................... 8  
      2.1.1. Microcontroller ............................................................................................ 8  
      2.1.2. GPS Module ............................................................................................... 10  
      2.1.3. Radio .......................................................................................................... 11  
   2.2 Receiver ............................................................................................................... 12  
      2.2.1. Microcontroller ............................................................................................ 12  
      2.2.2. Compass .................................................................................................... 13  
      2.2.3. Radio .......................................................................................................... 14  
      2.2.4 GPS Module ............................................................................................... 14  
3. Software Overview ....................................................................................................... 15  
   3.1 Transmitter ........................................................................................................... 15  
      3.1.1 Retrieve and calculate Device Coordination ................................................. 15  
      3.1.2 Sending message via radio to receiver side .................................................. 17  
      3.1.3 Timer ........................................................................................................... 20  
   3.2 Receiver ............................................................................................................... 22  
      3.2.1 Receive Data ................................................................................................. 22  
      3.2.2 Reading device coordination ....................................................................... 25  
      3.2.3 Calculating distance ..................................................................................... 25  
      3.2.4 Calculate heading of the device ................................................................... 28  
4. Test and implementations ............................................................................................. 31  
   4.1.1 Haversine’s formula V.S. Vincenty’s formula ................................................. 31  
   4.1.2 Radio Test ........................................................................................................ 31  
5. Conclusion & further works ........................................................................................... 33  
6. Bibliography ................................................................................................................. 34  
Appendix A- USART Communication in MSP430 ....................................................... 37  
Appendix B- SirfBinary protocol’s checksum calculation ........................................... 38  
Appendix C- ADF7021 write to registers ..................................................................... 38  
Appendix D - CRC16 calculation .................................................................................. 38  
Appendix E – I2C communication .................................................................................. 39  
Appendix F - Test Haversine’s formula V.S. Vincenty’s formula .................................. 39
Table of figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>General Functionality of MOB device</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Transmitter’s schematic</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Performance Per Watt (PIC Vs MSP Vs AVR)</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Performance Per Watt Per Dollar (PIC Vs MSP Vs AVR)</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Receiver's schematic</td>
<td>12</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Performance Per Watt (ST Vs Maxim)</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Performance Per Watt Per Dollar (ST Vs Maxim)</td>
<td>13</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Overview of transmitter device functionality</td>
<td>15</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Transmitter registers setup procedure</td>
<td>18</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>An example of Manchester encoding</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>MSP430’s timer up mode</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>Packet transmission time measured by Oscilloscope</td>
<td>21</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Overview of receiver device functionality</td>
<td>22</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>Receiver registers setup procedure</td>
<td>24</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Structure of receiving bytes via radio chip</td>
<td>24</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>Pythagorean Theorem</td>
<td>25</td>
</tr>
<tr>
<td>Figure 3.10</td>
<td>Haversine Theorem</td>
<td>26</td>
</tr>
<tr>
<td>Figure 3.11</td>
<td>Circle offsets</td>
<td>29</td>
</tr>
<tr>
<td>Figure 3.12</td>
<td>Tilt compensatsion</td>
<td>29</td>
</tr>
</tbody>
</table>
Tables

Table 1.1 General MOB device target specification ........................................................... 5
Table 1.2 Transmitter processor target specification .......................................................... 5
Table 1.3 Receiver processor target specification .............................................................. 6
Table 1.4 Radio target specification ................................................................................... 6
Table 1.5 GPS target specification ...................................................................................... 6
Table 2.1 MSP430F2122 features ...................................................................................... 9
Table 2.2 Transmitter nominated processors comparison .................................................. 9
Table 2.3 Fastrax IT321 features ...................................................................................... 10
Table 2.4 Fastrax IT321 features ...................................................................................... 11
Table 2.5 Receiver nominated processors comparison ..................................................... 12
Table 2.6 STM32 features ................................................................................................. 13
Table 2.7 Yas529 features ................................................................................................. 14
Table 2.8 ADF7021 features ............................................................................................. 14
Table 3.1 Set Serial Port Data Format and change mode of NMEA to SirfBinary .......... 16
Table 3.2 SirfBinary protocol message structure .............................................................. 16
Table 3.3 SirfBinary protocol input message 144 payload structure ................................. 17
Table 3.4 Transmission Packet Structure ........................................................................ 18
Table 3.5 MSP430 Operation modes ................................................................................ 20
1. Introduction

1.1 Research Purpose

As semiconductor technology steams ahead and chips with more and more functions become cheaper every day, exciting new product development.

One of the areas with great improvement opportunity is safety at sea. Men over board (MOB) accidents are quiet common and can result in loss of life more often than not. Any system that can cut the time off search and rescue operation and assist the involved parties in any way would be welcomed specially one which is affordable and user friendly.

It is the purpose of this thesis to design and implement such a system.

As mentioned above the most significant aspects of the system are affordability and ease of use. Affordable because even though its about saving lives there is always a price tag and if it is deemed too high it will not be widely use. And simple to use because the system is aimed at professional as well as non professional users and during an emergency and under stress people tend to forget the instructions. The system should be simple enough for anybody to use under any condition.

We are proposing a transmitter small enough that it could be integrated in an ordinary life jacket. The transmitter when either submerged or otherwise activated will start transmitting its position and the assigned ID number. The receiver upon reception of a valid signal from any transmitter will activate an alarm as well as showing the distance, direction and identity of the MOB.

1.2 Problem statement

This section tries to answer some questions:

- What is the problem?
  The main problem is finding a way to improve the speed of process finding and rescuing MOB at sea. This can be done by merging the other alternatives which already exist, add some new ideas to them and implement the new system in a simple to use manner.

- What are the alternatives so far (related work)?
  Traditional devices like personal ELTs (Emergency Locator Transmitter) [1] or handheld devices such as rescue laser flare are now becoming more common and, specialized but most of the current alternatives suffer from more or less serious short comings.

- What is the new solution?
  The new solution must be activated automatically (by sensing water) and/ or manually and uses GPS in order to locate the exact location of MOB to be rescued and send it to receiver side. Then the receiver, after parsing the received message, and by using GPS and compass heading should indicate the distance and direction to the MOB. The
system must handle virtually unlimited number of MOB. It should always indicate the
data for the nearest MOB but log the data for all MOBs for later processing.

Transmitter:
- Retrieve the coordinate of the MOB, convert it to the radians and send it.
- Send the time since the MOB device is activated.
- Send a bit in order to indicate the existence of the satellite fix. This option can
indicates that is the coordinate is updated or old.
- 16 bit CRC (Cyclic Redundancy Check) in order to validate the data.

Receiver:
- Receive the coordinate of the MOB devices.
- Calculate distance to the MOB device.
- Calculate the direction to the MOB devices simultaneously.
- And some other simple features like showing system clock, velocity, current
and information of each MOB device.

![Figure 1.1 General Functionality of MOB device](image)

Usually MOB devices works in two different ways:

I. **Active**: in this method MOB device always send a message back to the receiver in
order to inform that the wearer is safe. In case of danger (e.g. man over board, fire,
flooding) the connection will be aborted (degraded) or another type of message will
be sent which trigger alarm system.

II. **Passive**: in this method receiver device always listens to MOB devices without
transmitting. Once a massage is sent by transmitter emergency system will be
activated. This method is in use in this product.
1.3 Existing Products

Prior to finalizing the detail specification for our system a market research with the aim of identifying and characterizing the competition has been done.

1.3.1 ACR’s Mini B 300 ILS H2ON [2]

This is a floatable, water/manual activated device with the working frequency of 121.5 MHz. embedded inline speaker emits audio warble in order to confirm distress signal transmission or notifying the receiver which there is a MOB condition. An LED is also trying to confirm signal transmission. It will be complemented with an ACR Vecta2 direction finder. The important thing about this device is using the 10 years storage life replaceable lithium camera battery. Operating life is at least 24 hours in -10°C.

1.3.2 Alert 2 Man-Overboard Alarm [3]:

This device consisting water activated transmitter with the effective range of 1 mile and working frequency of 418 MHz. In addition to audible alarm the receiver could wire to the engine in order to shut down in the MOB condition also by connecting a NMEA-compatible chart plotter the location of transmitter device can be coordinated. Transmitter device use two 3.6 volt lithium batteries. The remarkable thing of this model is the support of unlimited number of transmitters while others have.

1.3.3 Life Tag Wireless Man Overboard Monitoring System [4]:

Like the other devices this manufacturer produces a set of device which consists of transmitter and receiver. The transmitter send a code back to the receiver permanently in order to confirm wearer is safe when the pendant beacon immerse to the water the signal will be degraded so it won’t be safe anymore.

In the receiver part when the signal is degraded the alarm will be turned on, and the sinking person position monitored. In addition some critical information like the time of sinking or personal information of the sinking crew could be provided.

Transmitter beacon uses CR2 lithium batteries with the estimation of 200 working hours.

1.3.4 Mobilarm V100 [5]:

The Mobilarm V100 is the first Maritime Survivor Locator Device (MSLD) [6]. which is like most of the other MOBs in the market can activated either manually or by sensing water. As soon as a man falls over board, it will send MAYDAY (distress message signal) [7] to whomever that is equipped by DSC (Digital Selective Calling) [8] System in the range of 10 nautical miles.

Some of the most important features of this device are as follows:

- Minimum battery life of 12 hours at -10°C.
- Positive buoyancy and water proof design plus a status LED make it visible from long distance.
- Use of standard VHF channel 70 of international maritime frequencies [9] to send distress messages.

1.3.5 First Light Solutions mermaid id [10]:
This system consists of three integrated parts: (1) ‘id-pods’ which are to be worn by crew members, (2) ‘thru hull hydrophone’ a listening transducer and (3) a ‘helm’ that will attach to the dashboard of the boat.

There are two pins on the id-pods when this two pin sense to water constantly for one second the device activates and send the coordination point to the second piece. While the listening transducer receives the message relay that to the helm and helm raise the audible and visual alarms additionally try to track the position of id-pod in the display.

Important features of this system are:
- The system can work with up to 30 id-pods.
- The helm has a NMEA output port that can connect to the other instruments like PC and GPS plotters for other usages.
- Light weight and durable. The system has been used for the vessels in the Volvo Ocean Race.
1.4 Target Specification

Without any doubt, first thing that should be concerned is how the device should be powered up and how much power can be supplied with respect to the working hours and dimensions of the board; consequence of choosing power supply can impact the rest of process of target specification in various aspects (ex. speed, price, dimension). As the Transmitter (MOB) device should be wearable and not heavy, four alkaline AAAA batteries will be used. Some advantages of this type of battery are listed below:

- Tiny design with the 8.3 mm diameter.
- 625 mAh capacity and supplying 1.5 voltages.
- Shelf life around 10 years.

So components of the transmitter device should power up by supplying voltage of less than or equal 6VDC.

The target specifications of MOB device are listed in table below:

<table>
<thead>
<tr>
<th>Working hours</th>
<th>Around 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage range</td>
<td>Around 5 kilometers</td>
</tr>
<tr>
<td>Dimensions</td>
<td>45mm<em>35mm</em>25mm</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>4.5V to 6V</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>Active Mode: 0.5A up to 2 A</td>
</tr>
<tr>
<td>Water resistance</td>
<td>Yes</td>
</tr>
<tr>
<td>Working area</td>
<td>Anywhere including water</td>
</tr>
</tbody>
</table>

Table 1.1 General MOB device target specification

Transmitter Processor:

Due to light tasks which this side is responsible for, there is no need for a high speed processor but because of limited amount of power, the chosen processor should be ultra low power. On the other hand because this part works as long as device works and is responsible for controlling other modules, it is good to have some operation modes to change the state of processor whenever its functions finished in order saving as much power as we can. Beside those, due to lack of space and as we need timer this microcontroller should have an accurate timer with an internal crystal. Table 1.2 Listed target specification of the Transmitter (MOB) device processor.

<table>
<thead>
<tr>
<th>Supply voltage</th>
<th>&lt;= 3 V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Consumption</td>
<td>&lt;= 1 mA</td>
</tr>
<tr>
<td>Serial Communication</td>
<td>USART , SPI</td>
</tr>
<tr>
<td>Timers</td>
<td>Watchdog and one 16-bit</td>
</tr>
<tr>
<td>Mode</td>
<td>Standby, active</td>
</tr>
<tr>
<td>GPIO</td>
<td>&gt;=20</td>
</tr>
<tr>
<td>Speed</td>
<td>&gt; 1MIPS</td>
</tr>
<tr>
<td>Price</td>
<td>&lt;=3 €</td>
</tr>
</tbody>
</table>

Table 1.2 Transmitter processor target specification
**Receiver Processor:**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>&lt;= 3 V</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>&lt;= 200 mA</td>
</tr>
<tr>
<td>Serial Communication</td>
<td>USART, I2C</td>
</tr>
<tr>
<td>Timers</td>
<td>Watchdog and RTC (Real Time Clock)</td>
</tr>
<tr>
<td>Mode</td>
<td>Standby, active</td>
</tr>
<tr>
<td>GPIO</td>
<td>&gt;= 30</td>
</tr>
<tr>
<td>Speed</td>
<td>&gt; 30 MIPS</td>
</tr>
<tr>
<td>Price</td>
<td>&lt;= 12 €</td>
</tr>
</tbody>
</table>

*Table 1.3 Receiver processor target specification*

**Radio:**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>&lt;= 3 V</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>&lt;= 40 mA for 10 dBm</td>
</tr>
<tr>
<td>Frequency range</td>
<td>International maritime frequencies Channel 70(156.525 MHz)</td>
</tr>
<tr>
<td>Price</td>
<td>&lt;= 1 €</td>
</tr>
</tbody>
</table>

*Table 1.4 Radio target specification*

**GPS:**

The most important criteria that should be concerned about GPS module, is to support of NMEA and Sirf Binary protocol. In this phase of project the Sirf binary protocol is used while in other versions that would be upgraded to NMEA protocol. Data interface format is not a great deal because nearly all available modules in the market support serial interface. See table 1.5 for the overall target specification of needed GPS component.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>&lt;= 3 V</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>&lt;= 100 mA</td>
</tr>
<tr>
<td>Data interface format</td>
<td>Serial interface should be available (UART or SPI).</td>
</tr>
<tr>
<td>Data protocol</td>
<td>NMEA and Sirf binary protocols should be supported.</td>
</tr>
<tr>
<td>Price</td>
<td>&lt;= 15 €</td>
</tr>
</tbody>
</table>

*Table 1.5 GPS target specification*
1.5 Research Methodology
The process of designing and Implementation of this project will start by introducing the ICs (Integrated Circuit) that are in used in this study and the way that they are used. Third chapter focuses on software algorithms and the way of interacting with software. The process of test and implementation will be reviewed in the forth chapter. Finally conclusion as well as the further work will come on the chapter five.

This research has four main stages, first stage includes information gathering; for this stage literature survey as a part of qualitative approach has been done for understanding the market needs and what has been done in the same area as well as the related work. The second stage is dedicated to planning and preparing the environment for experiment. In the third stage quantitative approach is chosen because we deal with some numerical analysis and mathematical calculations and statistical analysis (please refer to chapter three in order to find out what has been done for mathematical calculations and the test and implementation chapter for details of statistical analysis).
2. Hardware Overview

2.1 Transmitter

Figure 2.1 shows the schematic of the transmitter. The main concern in choosing different parts for transmitter has been power efficiency. Following the schematic, a detailed analysis is given for choosing the main parts.

![Figure 2.1 Transmitter’s schematic](image)

### 2.1.1. Microcontroller

Perhaps one of the most important issues of the growing technology in these days is how to increase the speed of the processor while optimizing the energy consumption. So choosing the microprocessor for any product is not a matter of speed anymore but it’s the compromising between power, speed and price.

The MSP430F2122 from T.I Company [11] has been chosen because of its ultra low power consumption, low price, and suitable speed. Below are some specifications of the microcontroller.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply-Voltage Range</td>
<td>1.8 V . . . 3.6 V</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>Active Mode: 250 μA at 1 MHz, 2.2V</td>
</tr>
<tr>
<td></td>
<td>Standby Mode: 0.7 μA</td>
</tr>
<tr>
<td></td>
<td>Off Mode (RAM Retention): 0.1 μA</td>
</tr>
<tr>
<td>Five Power-Saving Modes</td>
<td></td>
</tr>
<tr>
<td>Wake-Up From Standby Mode</td>
<td>less than 1 μs</td>
</tr>
</tbody>
</table>
### 16-Bit RISC Architecture
- 62.5-ns Instruction Cycle Time

### 12-Bit A/D Converter With Internal Reference, Feature
- Sample-and-Hold and Autoscan

### Serial Communication Interface (USART)
- Functions as Asynchronous UART or Synchronous SPI Interface

### Flash (KB)
- 4

### RAM (KB)
- 512

### Frequency (MHz)
- 8

### GPIO
- 24

### Timers
- 1 Watchdog/Interval, 1 16-bit (3CCR), 1 16-bit (7CCR)

### Integrated Peripherals
- Analog Comparator, Hardware Multiplier

<table>
<thead>
<tr>
<th>Table 2.1 MSP430F2122 [11] features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16-Bit RISC Architecture</strong></td>
</tr>
<tr>
<td><strong>12-Bit A/D Converter With Internal Reference, Feature</strong></td>
</tr>
<tr>
<td><strong>Serial Communication Interface (USART)</strong></td>
</tr>
<tr>
<td><strong>Flash (KB)</strong></td>
</tr>
<tr>
<td><strong>RAM (KB)</strong></td>
</tr>
<tr>
<td><strong>Frequency (MHz)</strong></td>
</tr>
<tr>
<td><strong>GPIO</strong></td>
</tr>
<tr>
<td><strong>Timers</strong></td>
</tr>
<tr>
<td><strong>Integrated Peripherals</strong></td>
</tr>
</tbody>
</table>

Now its time to show the comparison between 3 microcontrollers which are among most in used today.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power Consumption (WATT)</strong></td>
<td>223mA*5V</td>
<td>250uA<em>8</em>2.2V</td>
<td>18.4mA*3.2V</td>
</tr>
<tr>
<td><strong>Speed (MIPS)</strong></td>
<td>30</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td><strong>Performance Per WATT</strong></td>
<td>27</td>
<td>609</td>
<td>18</td>
</tr>
<tr>
<td><strong>Price (€)</strong></td>
<td>1.91</td>
<td>4.61</td>
<td>5.47</td>
</tr>
</tbody>
</table>

Table 2.2 Transmitter nominated processors comparison

![Figure 2.2 Performance Per Watt (PIC Vs MSP Vs AVR)](image-url)
Figure 2.3 Performance Per Watt Per Dollar (PIC Vs MSP Vs AVR)

After reviewing, “performance per Watt” and “Performance per Watt per Dollar” charts. Obviously MSP430 is the best choice right now for our transmitter device but there are some additional advantages which are listed below:

- Very good data sheet: everything can be found easily in the data sheet.
- Easy to program in compare with others (Jtag and serial port). With the kit several sample code is provided. Also GCC (GNU Compiler Collection) which is free of charge is supported.

NOTE: Current version (3.2.3) of GCC compiler for MSP430 processors, include full set of header files. In this stable version all the data types except floating numbers with double precision are supported. Also during installation it’s necessary to download some sort of patches from GNU website.

2.1.2. GPS Module

Another device which is used in the transmitter part is the Fastrax IT321 [14] GPS module in order to send the exact coordinate of the person in place of danger. The module specification has been shown in the Table3.

<table>
<thead>
<tr>
<th>Chip set &amp; Sensitivity</th>
<th>SiRF GSC3LT -159dBm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channels</td>
<td>20 physical (limited to 12 tracking by firmware)</td>
</tr>
<tr>
<td>Update rate</td>
<td>1 Hz default (fix rate configurable)</td>
</tr>
<tr>
<td>Supply voltage, VDD</td>
<td>+3.25V…+5.5 V</td>
</tr>
<tr>
<td>Power consumption, VDD</td>
<td>90 mW typical @ 3.3V</td>
</tr>
<tr>
<td>Operating temperature range</td>
<td>-30°C…+85°C</td>
</tr>
<tr>
<td>Serial port protocol (default)</td>
<td>NMEA 3.01 (switchable to SiRF binary)</td>
</tr>
<tr>
<td>Serial data format</td>
<td>8 bits, no parity, 1 stop bit</td>
</tr>
<tr>
<td>Serial data speed (default)</td>
<td>4800 baud (configurable)</td>
</tr>
<tr>
<td>Embedded antenna gain</td>
<td>2.8dBi (linear pol.) @ 80mm ground plane width</td>
</tr>
</tbody>
</table>

Table2.3 Fastrax IT321 features

Some advantages of this module are:
The size of this module is 10.4mm (Width) * 14.0mm (length)* 2.3mm (Height), so enabling smaller PCB for transmitter.

NMEA (The National Marine Electronics Association) and Sirf Binary protocols are both supported. In the following chapter a brief explanation and differences of these protocols will be given.

Serial interface speed can be set up to 115200 bps.

A part of board acting as the antenna so there is no need for soldering extra wires.

2.1.3. Radio
The radio chip used in the transmitter, “ADF7021” [15] is a high performance, low power, low cost, and highly integrated transceiver with a wide working frequency range. The specification of this chip which is described in Table 4(Transmitter part) and Table8 (Receiver part) made this IC somewhat unique in the market. "The ADF7021 is the only IC that meets the strict FCC Part 90 regulatory standard. In addition, it has excellent radio performance with best-in-class receiver sensitivity—6 dB better than the nearest competitor—which translates into longer range," [16]. Several modulation schemes are supported (FSK, 2FSk, 4FSK, MSK and GFSK) also the frequency agile PLL allows the chip to be used in frequency hopping systems.

<table>
<thead>
<tr>
<th>Operating voltage range</th>
<th>2.3…3.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature range</td>
<td>-40°C...+85°C</td>
</tr>
<tr>
<td>Maximum transmit power</td>
<td>+13 dBm</td>
</tr>
<tr>
<td>Frequency range(MHz)</td>
<td>80.650</td>
</tr>
<tr>
<td></td>
<td>862.950</td>
</tr>
<tr>
<td>Data rate</td>
<td>2FSK, FSK 0.5..25 Kbps</td>
</tr>
<tr>
<td>Transmit current consumption(mA)</td>
<td>868MHz</td>
</tr>
<tr>
<td></td>
<td>0dbm</td>
</tr>
<tr>
<td></td>
<td>5dbm</td>
</tr>
<tr>
<td></td>
<td>10dbm</td>
</tr>
<tr>
<td>Price (£)</td>
<td>1.91 per units in quantities of 1000</td>
</tr>
</tbody>
</table>

Table2.4 Fastrax IT321 features [14]
2.2 Receiver
Many tasks had been left for receiver due to limitation of power and speed in the transmitter part. Figure 2.4 shows receiver schematic.

![Figure 2.4 Receiver’s schematic](image)

2.2.1. Microcontroller
Like section 2.1.1 with respect to target specification and by comparing some common products in the market, the target microcontroller will be choosen.

<table>
<thead>
<tr>
<th></th>
<th>ST (STM32F103R8T6)[17]</th>
<th>Maxim (DS89C450)[18]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Consumption(WATT)</td>
<td>150mA*3V</td>
<td>75mA*5V</td>
</tr>
<tr>
<td>Speed(MIPS)</td>
<td>90</td>
<td>32</td>
</tr>
<tr>
<td>Performance Per WATT</td>
<td>200</td>
<td>85</td>
</tr>
<tr>
<td>Price(€)</td>
<td>8.29</td>
<td>12.59</td>
</tr>
</tbody>
</table>

Table 2.5 Receiver nominated processors comparison

Figure 2.5 and 2.6 illustrating the comparison between STM32F103R8T6 from ST Company and DS89C450 from maxim.
Table 2.6 describes the specification of STM32:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Processor</td>
<td>ARM® Cortex-M3™</td>
</tr>
<tr>
<td>Processor Name</td>
<td>STM32F103RET6</td>
</tr>
<tr>
<td>Core Size</td>
<td>32-Bit</td>
</tr>
<tr>
<td>Data Converters</td>
<td>A/D 16x12b</td>
</tr>
<tr>
<td>Interface</td>
<td>I²C, CAN, SPI, USART, USB</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>40°C – 85°C</td>
</tr>
<tr>
<td>Speed</td>
<td>72MHz</td>
</tr>
<tr>
<td>Standard operating voltage</td>
<td>2... 3.6</td>
</tr>
<tr>
<td>Supply current in Run mode @ 72MHz</td>
<td>50mA</td>
</tr>
<tr>
<td>Serial data speed (default)</td>
<td>4800 baud (configurable)</td>
</tr>
<tr>
<td>Embedded antenna gain</td>
<td>2.8dBi (linear pol.) @ 80mm ground plane width</td>
</tr>
<tr>
<td>Price(£)</td>
<td>8.29</td>
</tr>
</tbody>
</table>

Table 2.6 STM32 features

2.2.2. Compass

There are many compass modules and ICs in the market but what make are choice narrower is the need of tilt compensated compass sensor for this project. We found 2 perfect matches for using in this device one of them was HMC1055 [19] while the other one which has been chosen is Yas529 [20] from Yamaha Company. The features of this sensor are listed in the following table (Table2.7).
Supply voltage for core(V) | YAS529 | 2.5...3.6 | HMC1055 | 2.7...5.25
---|---|---|---|---
Operating temperature(°C) | -40...+95 | -40...125
Current consumption(mA) | 4 | 4.9 at 2.7 V
Measuring magnetic field range(µT) | ±300 | ±60.000
Resolution | ≤0.6 µT/count(X,Y) | ≤1.2 µT/count(Z) | 12 nT
Magnetic sensor measurement | Within 10 ms | Not available
Interface | I²C slave | Analog output
Dimensions | 2mm x 2mm x 1mm | 1.5mm x 4 x 7
Price(€) | 0.0072 | 21

Table 2.7 Yas529 features

Also there are some features which make YAS529 unique:
- The world smallest three-axis geomagnetic sensor.
- The price which is only a fraction of other competitors..

### 2.2.3. Radio

The chip which is in used in this part is same as transmitter part but the receiver specifications are described in the following table:

<table>
<thead>
<tr>
<th>Operating voltage range</th>
<th>2.3...3.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating temperature range</td>
<td>-40°C...+85°C</td>
</tr>
<tr>
<td>Maximum transmit power</td>
<td>+13 dBm</td>
</tr>
<tr>
<td>Frequency range(MHz)</td>
<td>80.650</td>
</tr>
<tr>
<td>862...950</td>
<td></td>
</tr>
<tr>
<td>Sensitivity(dBm)</td>
<td>Sensitivity at 0.1 Kbps: -130</td>
</tr>
<tr>
<td></td>
<td>Sensitivity at 0.25 Kbps: -127</td>
</tr>
<tr>
<td></td>
<td>Sensitivity at 1 Kbps: -122</td>
</tr>
<tr>
<td></td>
<td>Sensitivity at 9.6 Kbps: -115</td>
</tr>
<tr>
<td></td>
<td>Sensitivity at 25 Kbps: -110</td>
</tr>
<tr>
<td>Input IP3(dBm)</td>
<td>Low Gain Enhanced Linearity Mode: -3</td>
</tr>
<tr>
<td></td>
<td>Medium Gain Mode: -13.5</td>
</tr>
<tr>
<td></td>
<td>High Sensitivity Mode: -24</td>
</tr>
<tr>
<td>ADJACENT CHANNEL REJECTION @ 868 MHz(dB)</td>
<td>12.5 KHz Channel spacing: 25</td>
</tr>
<tr>
<td></td>
<td>25 KHz Channel spacing: 27</td>
</tr>
<tr>
<td>Price (€)</td>
<td>1.91 per units in quantities of 1000</td>
</tr>
</tbody>
</table>

Table 2.8 ADF7021 features

### 2.2.4 GPS Module

Like transmitter part, GPS module is needed to indicate the coordination points of receiver device and calculate the distance to the MOB device. Please refer to section 2.1.2 for detailed specification.
3. Software Overview
This section tries to clarify the problem and find a simple solution for both sides of system. Another aim is to discuss the ways that we can combine components (that are introduced in previous section) in order to get the desired functionality.

3.1 Transmitter
Figure 3.1 illustrates the overview of the transmitter device. First of all it is getting ON manually or by sensing water then until it finds a satellite and a validated coordination point sends the device ID and time since the device turned ON, once it finds more than 4 GPS satellite the message will be updated by coordination points of the device. In the following sections functionality of each IC and its dependencies with other ICs will be discussed.

![Figure 3.1 Overview of transmitter device functionality](image)

3.1.1 Retrieve and calculate Device Coordination
There are two standard protocols available in the Fastrax IT321 module NMEA [21] and Sirf binary protocol [22] after handling NMEA message we found out, Binary protocol is much faster mainly because the type of message that can be retrieved. While latitude and longitude in NMEA message need about 20 bytes of memory, Sirf binary needs 8 signed bytes in the memory. Furthermore by using NMEA data should be transformed from byte to integer while in the Sirf binary protocol just shifting the data is sufficient.

So we decided to use Sirf binary protocol in this project. To use this protocol, USART (Appendix A) port of MSP430 should be initialized properly. The setting of the processor is as follows

- Word length: 8bits
- Stop bit: 1bit
- Parity bit: NO
- Hardware Flow Control: NO
- Baud rate: 4800bps
The module is in the NMEA mode as default, we have to send PSRF 100 input command of NMEA message in order to change to Sirf binary protocol mode. Command syntax followed by command description is written below.

$PSRF100, 0, 4800, 8, 1, 0,*0F, 0x0D, 0x0A

<table>
<thead>
<tr>
<th>Name</th>
<th>Example</th>
<th>Unit Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>$PSRF100</td>
<td>PSRF100 protocol header</td>
</tr>
<tr>
<td>Protocol</td>
<td>0</td>
<td>0=SiRF binary, 1=NMEA</td>
</tr>
<tr>
<td>Baud</td>
<td>4800</td>
<td>1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200</td>
</tr>
<tr>
<td>DataBits</td>
<td>8</td>
<td>8,7</td>
</tr>
<tr>
<td>StopBits</td>
<td>1</td>
<td>0,1</td>
</tr>
<tr>
<td>Parity</td>
<td>0</td>
<td>0=None, 1=Odd, 2=Even</td>
</tr>
<tr>
<td>Checksum</td>
<td>*0F</td>
<td></td>
</tr>
<tr>
<td>&lt;CR&gt; &lt;LF&gt;</td>
<td></td>
<td>End of message termination</td>
</tr>
</tbody>
</table>

Table 3.1 Set Serial Port Data Format and change mode of NMEA to SirfBinary[21]

In order to provide better understanding and brighter image of the work that has been done in this section it is good to get familiar with structure of SirfBinary protocol message. Each message contains five parts as it’s depicted below:

<table>
<thead>
<tr>
<th>Start Sequence</th>
<th>Payload Length</th>
<th>Payload</th>
<th>Message Checksum</th>
<th>End Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xA0, 0xA2</td>
<td>Two-bytes (15-bits)</td>
<td>Up to 210-1(&lt;1023)</td>
<td>Two-bytes (15-bits)*</td>
<td>0xB0, 0xB3</td>
</tr>
</tbody>
</table>

Table 3.2 SirfBinary protocol message structure [22]

*Description about how to calculate checksum can be found at Appendix B

Message data rate and configuration should be set by sending the input message number of 166. The detailed description of message payload is listed in table 3.3.

<table>
<thead>
<tr>
<th>Name</th>
<th>Number of Bytes</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>1</td>
<td>A6</td>
<td>decimal 166</td>
</tr>
<tr>
<td>Mode</td>
<td>1</td>
<td>01</td>
<td>00: enable/disable one message 01: poll one message instantly 02: enable/disable all messages 03: enable/disable default navigation messages (Message ID 2 and 4) 04: enable/disable default debug messages (Message ID 9 and 255) 05: enable/disable navigation debug messages (Message ID 7, 28, 29, 30, and 31)</td>
</tr>
<tr>
<td>Message ID to be set</td>
<td>1</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td>Update Rate</td>
<td>1</td>
<td>01(second)</td>
<td>Range = 0 – 30</td>
</tr>
</tbody>
</table>
At first it was decided to set mode parameter to zero and a specific update rate (5 second) an interrupt generates in order to retrieve the desired data but as we progressed, it has been realized that there will be an interference between radio chip packet transmission and USART interrupt (there should be no interrupt while radio tries to send the message) so mode 1 (poll one message instantly) used for this purpose with the message id of 144. Message number 144 used because this message contains the latitude, longitude and number of satellites in range. The input message is

\[
0xA0,0xA2,0x00,0x08,0xA6,0x01,0x29,0x01,0x00,0x00,0x00,0x00,0xD1,0xb0,0xb3
\]

The message length of the output message 144 is 91 bytes. But as we just need three parameters from this message every time we only read up to byte 32, if the calculated checksum value is equal to message checksum value, they will be stored in a long value (32 bits signed value) then they should be transformed from degree to radian by this formula:

\[
\text{Angle in radians} = \text{angle in degrees} \times \frac{\pi}{180}
\]

### 3.1.2 Sending message via radio to receiver side

Sending data via radio has 2 main parts: turning on the radio and registers initialization, sending packet and turning off. These steps will be discussed in the following sections.

**Turning on the radio and registers initialization**

By changing the state of the CE pin of the ADF7021 radio chip to one the whole chip will be enabled and ready to work. For the purpose of enhance the effective range of the radio; an amplifier is used beside the radio chip which should be turned on after 500 microseconds.

In order to configure ADF7021 we are dealing with 16, 32 bit registers. First 4 LSB (Least Significant Bit) of each registers are dedicated for the register addresses. The model of writing to the register which is a bit complicated has been explained in Appendix C. Transmitter registers setup procedure is depicted in figure 3.1. The calculation of register values has been done with the great help of ADF7021 Evaluation Software.
Sending packet and turning off
In order to operate packet transmission properly, we need to manage data by defining a packet structure. The criteria that should be concerned for this purpose are:

- Due to power consumption of the radio part compare with other chips on our board, transmission packet should be as short as possible because, longer data packet means the longer time that radio is on which results in consuming more power.
- On the other hand packet should contain all the essential information in addition to the information that makes the packet recognizable.

By concerning these two criteria, our packet structure is illustrated in the following table.

<table>
<thead>
<tr>
<th>Device ID</th>
<th>Kind and time of the message</th>
<th>Latitude and longitude</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 bits</td>
<td>16 bits</td>
<td>48 bits</td>
<td>16 bits</td>
</tr>
</tbody>
</table>

Table 3.4 Transmission Packet Structure
But before sending each packet, 40 bits of preamble and 2 bytes of start of frame (0x1906) will be sent. As the names indicates device ID is a unique ID which attach to the...
MOB device during the production process. The second part of messages has 16 bits which first 12 bit from left indicates the **number of minutes** since the MOB is activated or in other word time in water. Next 3 bits belong to **kind of the message**. Right now in this version we have only one kind of message so these 3 bits are constants and equal to be 6 and the last bit is indicating that whether MOB has **access to the valid satellites** or not. if this bit set to one it means that device has a clear view to the GPS satellites and if its set to zero it means that is not the case. The next piece of the message contains the **latitude and longitude** that was calculated before when the GPS module retrieves the data from the satellites but as it takes about 3 minutes in order two find the satellites and receive the validated data the default value for these two byte are 0xFFFF which will not coordinate anywhere in the earth. Last 16 bits are the **CRC16** (information about CRC16 can be found in Appendix D).

**Manchester encoding**

Manchester encoding [23] has been applied to the process of data transmission in this project. This section will discuss the actual needs of Manchester encoding and provides some information about it.

In the beginning, the simple way of data transmission was used which means, high voltage represents as 1(or high bit) and low voltage represents as 0(or low bit). But after a while we found out that sometimes some bits are missing in the receiver side (calculated CRC was not same as transmitted packet CRC). After testing transmitted data and received data by Oscilloscope the problem was revealed. Under some circumstances the transmit packet contains long runs similar bits (more than 5 consecutive 1s or 0s) and this will cause a DC offset in the demodulator and results in bit error. This problem has been solved by using Manchester encoding, in this method ‘1’ represents as a high to low transition while ‘0’ is low to high transition. By changing the state of the output data the data transmission will be facilitated and reduce the chance of data failure. See Figure 3.3.

![Manchester Encoding](image)

**Figure 3.3** an example of Manchester encoding

Along with the mentioned advantages, the main disadvantage of Manchester encoding is increasing time of transmission. By using this method the number of transmission bits will be twice so the time of data transmission increases respectively and the speed divide by 2.
3.1.3 Timer

Because of following reasons, timer is essential for this project:

1. In order to save as much power as possible, each time after sending the packets all the components in the device will go in standby mode for some specific amount of time.

2. As it was mentioned previously, the time since the transmitter was activated should be send to the receiver side continuously.

For these purposes internal timer of MSP4302122 in up mode has been used, as figure 3.3 shows in this mode counter increments from zero to the desired value which can be set in the TACCR0 register repeatedely, when it reaches to the this value an interrupt will be generated and the counter value restarts automatically. We decided to get interrupt in each 10 milliseconds, because this version of project runs under speeds of 1 MHz speed every 10000 clock cycle will be 10 milliseconds so the TACCR0 register value should be set to 10000. So each hundred times (1 minute) the times in water value increments by 1.

![Figure 3.4 MSP430's timer up mode](image)

But for changing the state of device to standby mode we need to stop the processor in the specific amount of time. MSP430 processor has 5 modes except active mode which everything is active, 4 low power modes exist. Because the internal clock of msp430 (SMCLK) is used in this project we had to choose a low power mode which internal DCO and SMCLK being active always even in standby mode. Table 3.5 shows the different functions of operation modes of processor.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Mode</td>
<td>CPU is Active and all peripheral modules are active</td>
</tr>
<tr>
<td>LPM0</td>
<td>CPU Off, MCLK Off, SMCLK ON, DCO ON</td>
</tr>
<tr>
<td>LPM1</td>
<td>CPU Off, MCLK Off, DCO Off, SMCLK On, ACLK On</td>
</tr>
<tr>
<td>LPM2</td>
<td>CPU Off, MCLK Off, SMCLK Off, DCO Off, ACLK On</td>
</tr>
<tr>
<td>LPM3</td>
<td>CPU Off, MCLK Off, SMCLK Off, DCO Off, ACLK On</td>
</tr>
<tr>
<td>LPM4</td>
<td>CPU Off, MCLK Off, DCO Off, SMCLK Off, ACLK Off</td>
</tr>
</tbody>
</table>

Table 3.5 MSP430 Operation modes [24]

By having a look at the table 3.5 it will be obvious that the right choice is Low Power Mode 0 (LPM0) because that’s the only mode which both SMCLK and DCO are active at the same time.

As previously mentioned, the reason to use timer in this project is to save the power. The actual working hours of the device should be 24 hours. But the actual time that device can work properly with 4 AAAA battery which is in used for the mob device is:
Capacity of AAAA battery (mAh)/Power Consumption of MOB (mAh) = 625/2000 = 0.3125 (h)
So the actual time is 60 * 0.3125 = 18.75 minutes.

But transmitting time of each packet (that includes 80 bits preamble + 32 bits start of frame + 192 bits data) with 304 bits payload with the data rate of 4.8 kHz is calculated below and show in figure 3.5:

Number of packets that can be sent in each second = 4800 / 304 = 15.78 ~ 16
The transmission time of each packet = 1000 (ms) / 16 = 66 (ms).

Figure 3.5 packet transmission time measured by Oscilloscope
3.2 Receiver

It is previously mentioned many tasks because of limited source of power in the transmitter, was left to the receiver side. Because of communication with more chips and more calculation this side of the system is more complex. As it is illustrated in figure 3.6 the device always listens to the desired frequency (in this case channel 70 of international maritime frequency which is 156.525) and once a validated packet is received, the direction and distance to MOB will be calculated.

![Flowchart of Receiver Functionality](image)

3.2.1 Receive Data

In contrary to the transmitter the receiver Radio chip is always on in receive mode. The transceiver listens to the set frequency continuously.

Register initialization

According to [15], in order to prepare and set the state of the ADF7021 chip to receiver, five register should be programmed. These register and explanation about what are their functions with respect to their turn in the sequence of writing can be found below and depicted in figure 3.7 (as previously mentioned the four least significant bits of each register are dedicated to register address):

- Register 1: value that is set for this register is 0x535011; Fifth bit should be one because there is no need to divide input frequency to smaller number, since we don’t use clock divider, buffer bits 7 to 10 should be set to 0. There is possibility to multiply value of the reference crystal by 2 but it is not used in this project the 11th bit is set to 0. By changing the state of 12th bit to high the oscillator circuit of the chip will be powered up the main reason for programming this register was this bit which in fact power up the chip. Other important bit in this registers are bits 20 to
24 which refer to oscillator current bias. This value according to [15] should be set to 2 for our working frequency.

- Register 3: value of this register is 0x2A4CC093; According to [15] bit 4 and 5 represents baseband offset clock frequency divider, while the baseband offset clock frequency must be greater than 1 MHz and less than 2 MHz so the relation is:

  Baseband clock frequency = Crystal frequency / Baseband offset clock frequency divider = 14.7456*10^6/8 = 1.8432*10^6

The next step is to set up demodulation clock divider (bits 6 to 10) which should be between 2 MHz and 15 MHz where:

  Demodulation Clock = Crystal frequency / Demodulation Clock divider = 14.7456*10^6/2 = 7.3728*10^6

Following 8 bits are dedicated data/clock recovery frequency divider and the value of the data/clock recovery frequency should be within 2% of (32*symbol rate) so in our case divider value is 48.

Next 8 bits are called Sequencer clock divider and should be close to 100 kHz and it will be set by following formula:

  Sequencer clock = Crystal frequency / Sequencer clock divider = 14.7456*10^6/93 = 158.554*10^3

Last 8 bits are called AGC clock divider and it is the time allowed for each AGC step to update, the value should be close to 10 kHz so the divider value is 10.

- Register 5: value of this register is 0x24F5; this register is dedicated to IF filter setup. Coarse IF filter calibration is performed when the bit 5 is set. The IF filter calibration is automatically performed after the coarse calibration. The coarse calibration value should be around 50 kHz where the relation is:

  Crystal frequency / IF filter divider = 14.7456*10^6/167 = 88.297*10^3

Bits 6 to 14 are referred to IF filter divider which should be set to 167. Since image rejection calibration and rest of the bits should be set to zero.

- Register 0: the value that should be set for this register is 0x895376C0; the main responsibility of this register is to setup the radio frequency of the chip. This will be done by setting the values of bits 5 to 19 which called FRACTIONAL_N and bit 20 to 27 seven which in datasheet called INTEGER_N with respect to the following formula:

  Local oscillator frequency / Synthesizer frequency = PFD * (INTEGER_N + FRACTIONAL_N / 2^15)

  Local oscillator frequency / Synthesizer frequency = 3.6864*10^6 * (42 + 14188 / 2^15) = 156.424950MHz

According to [15], the output frequency of synthesizer should be 100 kHz below the output frequency so:

  Output frequency = 156.524950MHz

Receiver mode of chip will be active when the 28th bit is set. Since the standard mode is used in this project, the 29 bit should be set to zero.

- Register 4: the value that should be set for this register is 0x895376C0;
Before going to next section, it’s good to know how the Manchester data has been decoded. Actually there is no piece of code in this part which handles decoding of data; instead the data rate is set to twice of the data rate of transmitter and falling edge of the clock is skipped.

**How to receive data?**

As it has been said in the first chapter, one of the goals of receiver device is the ability of dealing with unlimited number of mobs. In the other hand there are some other tasks that should be done simultaneously. By the help of interrupt request (IRQ) both criteria has been met. IRQ has been set to the falling edge of radio chip clock. So by each falling edge of clock one bit will be received.

In addition to start of frame which is 16 bits, number of bits which should be stored for a complete packet is 112 bits. For this purpose 2, 64 bits variables have been defined, Each time a bit received, it will be added to the first temporary variable and all of the values will be shifted to left by 1. Figure 3.8 shows this structure in a smaller scale.

<table>
<thead>
<tr>
<th>Step</th>
<th>Byte 2</th>
<th>Step</th>
<th>Byte 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>Step 1</td>
<td>0 0 0 0 0 0 0 0 0 1</td>
</tr>
<tr>
<td>Step 2</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>Step 2</td>
<td>0 0 0 0 0 0 0 0 1 0</td>
</tr>
<tr>
<td>Step 3</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>Step 3</td>
<td>0 0 0 0 0 0 1 0 1</td>
</tr>
<tr>
<td>Step 4</td>
<td>0 0 0 0 0 0 0 0 0</td>
<td>Step 4</td>
<td>0 0 0 0 0 0 1 0 1</td>
</tr>
<tr>
<td>Step 8</td>
<td>0 0 0 0 0 0 0 0 1</td>
<td>Step 8</td>
<td>0 0 0 0 0 1 1 0 1</td>
</tr>
<tr>
<td>Step 9</td>
<td>0 0 0 0 0 0 0 1 0</td>
<td>Step 9</td>
<td>0 0 0 0 1 1 0 1</td>
</tr>
<tr>
<td>Step 10</td>
<td>0 0 0 0 0 0 1 0</td>
<td>Step 10</td>
<td>0 0 0 1 1 0 1</td>
</tr>
<tr>
<td>Step 11</td>
<td>0 0 0 0 0 1 0</td>
<td>Step 11</td>
<td>0 0 1 1 0 1</td>
</tr>
<tr>
<td>Step 14</td>
<td>0 0 0 0 1 0</td>
<td>Step 14</td>
<td>0 0 1 1 0 1</td>
</tr>
<tr>
<td>Step 15</td>
<td>0 0 0 1 0 0 1</td>
<td>Step 15</td>
<td>0 0 1 1 0 1</td>
</tr>
<tr>
<td>Step 16</td>
<td>0 0 1 0 0 0 1</td>
<td>Step 16</td>
<td>0 0 1 1 0 1</td>
</tr>
</tbody>
</table>

**Figure 3.8 structure of receiving bytes via radio chip**

Note: received byte value is equal to 0xA6C3
By using this structure, the last 16 bits of second temporary variable should always be watched and if it is equal to 0x1906, the rest of the data goes to validating phase. Received data will be validated if their calculated CRC-16 value is equal to received CRC-16 value.

### 3.2.2 Reading device coordination

For communication with chip instead of handling and using COM port a new method which presented recently by Microsoft corporation has been used. This method called “GPS Intermediate Driver Application development” [25] has been used. Some benefits are listed in [24].

For creating connection with GPS intermediate driver, GPSOpenDevice function should be called with two input arguments (handles). Actually these two handles have been created with CreateEvent function. The first event is responsible to pass signal whenever a new location is retrieved and the second one is responsible for passing the signal whenever state of the device is changed.

After a connection is created, it is time to retrieve the location information, including latitude and longitude. One benefit of this method is returning the coordinate values in NMEA format so once the data is fetched; there is no need to parse it. That will be done by calling function GPSGetPosition [26]. Second argument of this function is the most important one, by passing a pointer from GPS_POSITION [27] structure, on return, this structure is filled with all coordinate data which is obtained by GPS intermediate Driver.

### 3.2.3 Calculating distance

There are many different approaches to calculating the distance between two points on any surface, these approaches split in 3 groups while the difference between them is tradeoff between accuracy and speed and depending on how an application chooses to model the Earth.

**Flat-model approach:** presuming earth as a flat surface then Pythagorean Theorem can be used to calculate straight line distance. In order to implement this method, latitude and longitude coordinates should be converted to Cartesian coordinates which express the distances in two different directions (north to south and east to west). If a right triangle with two short sides of south-east and east-west is created the straight line distance can then be thought of as the long side of the triangle (See figure 3.9).

![Figure 3.9 Pythagorean Theorem](image-url)
Pythagorean Theorem formula is: \[ d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \]
Where \( d \) is the distance the \( X_2 - X_1 \) the difference in the X-axis and \( Y_2 - Y_1 \) is the difference in the Y-axis.

There are some drawbacks in this approach. Near the poles when longitude lines reach each other and circumference of latitudes become smaller and smaller, differences between in locations along these directions can lead to large error. Flat earth formulas start to show noticeable errors when the distance is more than 20 kilometers, Otherwise result in an error of:
- Less than 30 meters for latitudes less than 70 degrees,
- Less than 20 meters for latitudes less than 50 degrees,
- Less than 9 meters for latitudes less than 30 degrees.
(Note: These error statements reflect both the convergence of the meridians and the curvature of the parallels.)

Since we are dealing with the latitudes and longitudes, for using this method they should be converted to Cartesian coordinates then applying flat Earth method. The time that this calculation takes will exceed the spherical model which is more accurate.

**Spherical model approach:** presuming Earth as a Spherical surface with radius \( R \) (Earth radius is equal to 6367Km) then Haversine formula can be used to calculate straight line distance.

Haversine Formula [28]:
\[
\text{haversin}(d/R) = \text{haversin}(\text{lat}_2 - \text{lat}_1) + \cos(\text{lat}_1) + \cos(\text{lat}_2) \times \text{haversin}(\text{dlon})
\]
Where:
- haversin is equal to \( \sin^2(\theta/2) \),
- \( d \) is the distance between two points,
- \( R \) is the radius of the sphere (in this case should be Earth sphere),
- \( \text{lat}_1 \) is the latitude of point 1,
- \( \text{lat}_2 \) is the latitude of point 2,
- \( \text{dlon} \) is the difference between longitude of point 2 and longitude of point 1,

Then:
\[
d = 2R \times \text{acrsin}(h)
\]
where \( h \) is equal to \( d/r \).

![Figure 3.10 Haversine Theorem][35]
Considering figure 3.10, according to Haversine formula if the given sphere is assumed a unit sphere, the length of “a”, “b” and “c” are equal to angles of them in terms of radian. If “u” considers as North Pole and v and w are two points which their distance is required then “lat1” and “lat2” are equal to \( \cos(a) \) and \( \cos(b) \), “C” is “dlon” and “c” is equal to \( d/R \). Haversine Theorem is more accurate than Pythagorean Theorem while execution time for both of them is approximately the same.

**Ellipsoidal model approach:** Finally, presuming Earth as an ellipsoidal surface. For implementing this model Vincenty formula can be used which can obtain highly accurate distances (down to 0.5 millimeters) on the ellipsoid. According to [29] this formula has two methods:

- **Direct method:** by providing latitude and longitude of a starting point in the surface along with the starting azimuth and the distance to the destination point. This formula calculates the coordination points of destination point.
- **Inverse method:** in this method latitude and longitude of two points should be given, the geodesic distance, forward and reverse azimuth will be found.

In our case inverse method can be used with the following formula:

\[
\begin{align*}
\sin^2 \beta &= (\cos U_2 \sin \lambda_2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda_2))^2 \\
\cos^2 \gamma &= \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos \lambda_2 \\
\tan \sigma &= \frac{\sin \sigma}{\cos \sigma} \\
\sin \alpha &= \cos U_1 \cos U_2 \sin \lambda / \sin \sigma \\
\cos 2\alpha &= \cos \sigma - (2 \sin U_1 \sin U_2 / \cos \alpha) \\
C &= (f/16) \cos 2\alpha [4 + f (4 - 3 \cos 2\alpha)] \\
\lambda &= \omega + (1 - C) \sin \{\sigma + C \sin \sigma [\cos 2\alpha m + C \cos \sigma (-1 + 2 \cos 2\sigma m)]\}
\end{align*}
\]

Then:

\[
\begin{align*}
u_2 &= \cos 2 \alpha (a_2 - b_2) / b_2 \\
A &= 1 + (u_2 / 16384) \{4096 + u_2 [-768 + u_2 \{320 - 175u_2 \}]} \\
B &= (u_2 / 1024) \{256 + u_2 \{128 + u_2 (74 - 47u_2) \}]} \\
\Delta \sigma &= BS \sin \sigma \{\cos 2\sigma m + (B/4) [\cos \sigma (-1 + 2 \cos 2\sigma m) - (B/6) \cos 2\sigma m (-3 + 4 \cos 2\sigma m)]\} \\
S &= b A (\sigma - \Delta \sigma)
\end{align*}
\]

\( \Phi_1, \Phi_2 \) Latitude of the points
\( \lambda_1, \lambda_2 \) Longitude of the points
\( \tan U_1 = (1-f) \tan \Phi_1 \)
\( \tan U_2 = (1-f) \tan \Phi_2 \)
\( \omega = \lambda - \lambda_2 \)

S is equal to ellipsoidal distance between the two points
When two points are antipodal, both Harvesin and inverse Vincenty formula may result in error. Consequently, by comparing complexity and execution time of Harvesin and Vincenty formulas, Harvesin formula has been chosen to use in this project (test and comparison which has been implemented between these two formula, will be presented in next chapter).

### 3.2.4 Calculate heading of the device

In order to determine the magnetic north, a three-axis magnetometer is cooperating with a 3-axis accelerometer. For accessing to the magnetic compass registers (information) I2C interface has been implemented which is explained in Appendix E.

The problem is magnetic compass does not only measure earth magnetic field, but some additional fields that are created by the sensor environment. So in order to have correct magnetic field value data should be calibrated. There are always two factors that disturb earth magnetic field [30]:

- Permanent perturbation results from permanent magnetizations and DC currents in any parts of compass or its installation environment. As a result this value is always constant and depends on the compass and its installations.
- Induced perturbation is attributable to magnetizable soft iron material in the vicinity of the compass. Induced perturbation is variable in both direction and magnitude and is proportional to the earth magnetic field.

Thus 2 model of calculation should be done in this project one manual calibration in order to determine permanent perturbation and auto calibration that should compute induced perturbation continuously.

#### Manual calibration

This process will be implemented only once at initialization for instance first time that the device is being installed somewhere. The process consists of rotating the device in each 3-axis. While the device is rotating it will measure the magnetic fields since the other two axes are constant (in the case that it rotates horizontally only Y axis is changed) the centered value of each axes will be obtained by the help of averaging correct values and filtering those values that are over limited. After this phase correction offsets that sets the measuring values to the center of magnetic field are ready, in other words permanent perturbation will be omitted by using these offsets.

#### Automatic Calibration

According to [31], every time after measuring earth magnetic field, in order to find the induced perturbation the values should be calibrated by the following formula:

\[
X_{\text{new}} = X_{\text{old}} + \Delta \cdot \cos(\tan^{-1}(H_y/H_x)) \\
Y_{\text{new}} = Y_{\text{old}} + \Delta \cdot \sin(\tan^{-1}(H_y/H_x)) \\
\Delta = \sqrt{H_x^2 + H_y^2} - R_{\text{mean}}
\]
Where $H_x$ and $H_y$ are the corrected x and y sensor value, $\Delta$ denote the offset length and $R_{\text{mean}}$ represents radius of the magnetic circle.

**Figure 3.11 Circle offset [36]**

**Direction Calculation**

Electric compasses measure direction to the north magnetic pole which is at 79º latitude, 115 º W longitudes and oriented 115 degrees away from actual North Pole. The easiest way that is in use in 2D electronic compass for calculating direction is: Heading = $\arctan(x/y)$.

Since position of the device is not fixed and it can be tilted, the important role of Z axis should be considered. So X and Y should be transferred according to their tilt angle (compensation):

**Figure 3.12 Tilt compensation**
According to [31], assume $G_x, G_y$ and $G_z$ are accelerometer corresponding values and $g$ is the gravity acceleration then:

$G_x = -g \cdot \sin(\theta)$
$G_y = g \cdot \sin(\Phi) \cdot \cos(\theta)$
$G_z = g \cdot \cos(\Phi) \cdot \cos(\theta)$

Then the roll angle is:

$\Phi = \arctan \left( \frac{G_y}{G_x} \right)$

And pitch angle is:

$\Theta = \arctan \left( \frac{-G_x/G_y \cdot \sin(\Phi) + G_z \cdot \cos(\theta)}{} \right)$

Now by assuming $H_X, H_Y$ and $H_Z$ as the calibrated values that obtained from compass the tilt compensated values ($H_{X_{true}}, H_{Y_{true}}$ and $H_{Z_{true}}$) are:

$$
\begin{bmatrix}
H_{X_{true}} \\
H_{Y_{true}} \\
H_{Z_{true}}
\end{bmatrix} =
\begin{bmatrix}
C\theta & S\theta \Phi & S\theta C\Phi \\
0 & C\Phi & -S\Phi \\
-S\theta & C\theta S\Phi & C\theta C\Phi
\end{bmatrix}
\begin{bmatrix}
H_X \\
H_Y \\
H_Z
\end{bmatrix}
$$

Then direction is:

Direction = $\arctan(H_{Y_{true}}/H_{X_{true}})$.
4. Test and implementations

So far target specification, hardware and software solutions in addition to some theoretical formula have been discussed. Now it is the time to implement and test them in order to make sure, they meet the critical requirements.

4.1.1 Haversine’s formula V.S. Vincenty’s formula

As it was mentioned before, a test between Haversine’s and Vincenty’s formula has been done. In order to compare both output distances and their accuracy. The test was done under following conditions:

- First point is located at latitude: 59.24 and longitude: 17.56.
- Second point latitude made by adding 0.01 of degree to the first point latitude and longitude made by adding 0.01 of degree to the first longitude point.
- After each 5 time iteration of upper step the first point latitude and longitude is increased by 5 degree.

*NOTE: This test also has been done in reverse mode. This means that instead of adding 0.01 degree and 5 degree they were subtracted.

*NOTE: the detailed result of output data can be found in Appendix F.

The result of this experiment shows that as long as we get far from our origin point, Haversine’s formula accuracy reduces but in our case these deviations at most is between 15- 20 meters when the distance is about 5 kilometer. On the other hand complexity and elapsed time that takes in the handheld devices to calculate distance via Vincenty’s formula is very high.

4.1.2 Radio Test

In this part two model of testing considered, theoretical and practical. But before going through the tests, since working environment is over sea surface, some important points should be considered. Radio wave propagation includes everything that can happen to an energy radiated from a transmitting antenna to a receiving antenna. It includes the radiating properties of both antennas such as gain, directivity and polarization. It also includes free space attenuation of radio wave with distance in addition to factors such as refraction, reflection, noise interference, diffraction, atmospheric absorption and scatter. Propagation is therefore dependent upon the properties of transmission and boundary media [32]. So both theoretical and practical parts of this survey are just predictions, because especially on the sea, there are many environmental characteristics that change widely over time which will affect the propagation in unpredictable ways.

Theoretical

The transmit power of the transmitter is about 3 Watt, so the power in dBm is:

\[
\text{dBm} = 10 \log [\text{P/1mW}] = 10 \log[3000/1] = 34.77
\]

On the other hand the receiver sensitivity is -110 dBm, so the maximum permissible path loss will be equal to:

\[
\text{Path loss} = 35 - (-110) = 145 \text{ dB}
\]

Path loss can be expressed as a function of distance and frequency:

\[
\text{Path loss} = 32.4 + 20 \log (f) + 20 \log(d)[33]
\]
Where \( f \) is the radio frequency in MHz and \( d \) is the effective distance in kilometers (km). By applying above formula, the effective range of our device with the path loss of 145 dB and working frequency of 170 MHz is 5.5185 km.

**Practical**
For this test, a receiver placed in the room in the 3rd floor of a building and a person carried transmitter out of the building. Unfortunately boards only could communicate within 1 km.
5. Conclusion & further works

In this thesis, design and implementation of a system that facilitate the process of finding and rescuing victims of man over board accidents is presented. During test and implementation of this project following issues has been happened:

- Receiver PCB boards contains metal, and this results in a high permanent noise for magnetometer sensor. As it was discussed earlier there are two types of noise that should be calibrated for this sensor. First type was permanent that only needs to be calibrated once but the noise magnitude is so high in this case that it results in data overflow during measuring permanent perturbation.
- Due to the working environment (sea), used frequency, type of antenna etc. the effective range of the device is considerably lower than expected.

For the next release above problems should be addressed by better antenna design, decrease the receiver PCB noise, shielding magnetometer area or perhaps changing magnetometer sensor to another sensor which provide more resistance to noise. Furthermore there are some areas that the system could be improves. These areas are listed below:

- In order to be compatible with other existing products in the market, ITU-R protocol should be applied to both receiver and transmitter part. For applying this protocol, existing GPS protocol (binary) should be changed to NMEA protocol; Manchester encoding should be ignored in addition to changing the packet structure.
- Because coordination points, direction and desired speed are all provided. Receiver device can easily take the control of the boat to departure in the case of emergency. Obviously this needs to design and implement a new system which works parallel with the current product.
6. Bibliography

http://faculty.uml.edu/jweitzen/16.548/ClassNotes/A%20PAINLESS%20GUIDE%20TO%20CRC%20ERROR%20DETECTION%20ALGORITHMS.pdf.


[36] Based on: Chin E.Lin, Chan Wei Hsu. Avionics System Integration on Autonomous And Remote Control Enhancement(2008), Tainan, Taiwan, R.O.C: Dissertation of Doctor of Philosophy Department of Aeronautics and Astronautics National Cheng kung University. (pp.194)
Appendix A- USART Communication in MSP430

Following configuration has been applied to port USART0 of MSP430f2122 microprocessor:

UCA0CTL1 = UCSSEL_2;
UCA0CTL0 =0x00;
UCA0BR0 =208;
UCA0BR1 =0;
UCA0MCTL=UCBRS1 + UCBRS0;
UCA0CTL1&=~UCSWRST;

This section tries to clarify the above register values.

**UCA0CTL1:**

<table>
<thead>
<tr>
<th>Bit no</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCSSELx</td>
<td>UCRXEIE</td>
<td>UCBRKIE</td>
<td>UCDORM</td>
<td>UCTXADDR</td>
<td>UCTXBRK</td>
<td>UCSWRST</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**UCSSELx:** by assigning 2 to last two bits of this register(UCSSEL_2=0x2), SMCLK(System Module Clock) will be choosen as the USART control clock.

**UCA0CTL0:**

<table>
<thead>
<tr>
<th>Bit no</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2..1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCPEN</td>
<td>UCPAR</td>
<td>UCMSB</td>
<td>UC7BIT</td>
<td>UCSPB</td>
<td>UCMODEX</td>
<td>UCSYNC</td>
<td></td>
</tr>
</tbody>
</table>

This register contains port setting values of the choosen USART. Desired port setting is:

**Parity:** None

**Data flow direction:** LSB(Least Significant Bit) first

**Number of data bits:** 8

**Number of stop bits:** 1

**USART Mode:** Asynchronous mode

**UCA0BR0 and UCA0BR1:**

These two register are sum of 16 bits that store baud rate value of the USART communication. In this microcontroller baud rate calculates in the following way:

**Baud rate** = Frequency of clock source/baudrate = $10^6/4800 = 208$

In order to start USART operation UCSWRST of UCA0CTL1 should set to Zero.
Appendix B- SirfBinary protocol’s checksum calculation [22]

As it was mentioned before, SirfBinary protocol uses a 16 bits checksum. Checksum value calculates as a sum of all payload data, following flow chart tries to clarify this method.

Index=0
Checksum=0
While index<MsgLength
   Checksum+=Message [index++]
   Checksum&=0x7FFF
End while

While Message is the array of receive/transmit bytes and MsgLength is size of that array.

Appendix C- ADF7021 write to registers [15]

For writing to the register 4 pins in the ADF7021 chip are interfered. SCLK (System clock), SLE (System Latch enable), SDATA (data line) and SREAD (Sent data read back) pin which is not in used in this implementation.

Data is clocked in rising edge of SCLK with MSB order (Most Significant Bit), while the SLE pin is low. After all data transfer to the shift register SLE pin should set to high in order to save data. Following algorithm shows writing data to registers in ADF7021 chip:

i=0
j=4
data[4]
While i<4
   While j<8
      SCLK=0
      if (data[3-i]>>8&1)
         SDATA=1
      else
         SDATA=0
      END While
      j++
   END While
   i++
End while

Appendix D - CRC16 calculation [33]

Cyclic Redundancy Check (CRC) is a way of detecting accidental changes in transmitting messages. The key point for calculating CRC is choosing polynomial. In this project CRC-16-IBM has been choose with the polynomial of x^{16}+x^{15}+x^2+1 or 0x8005. The following algorithm shows a simple way to calculate CRC:

i=j=0
crc = 0
poly = 0x8005
while i< msg_size
   msg &< 8
while j<8
  if((msg ^ crc) >> 15) crc = (crc << 1) ^ poly
  else crc <<= 1
  msg <<= 1
end while

j=0;
end while

Appendix E – I2C communication [34]
I2C has been created by Philips semiconductors and allows communication of data between devices only by use of two wires. Because only two wires are interfered in this protocol a unique address should be assigned to each device in order to identify them on the bus. For implementation of this system at least one master is needed while several slave devices can exist. Data are transfers serially over data line (SDA) and clock is provided over clock wire (SCL). Master device is always in charge of controlling the bus and generating the clock.

Following steps are general transfer sequence of I2C:

- **Generating start:** start command always issue by master device. This will be done by force the SCL pin to high and falling edge of SDA pin.
- **Send the slave address:** I2C communication address has two parts the 7 most significant bits are the actual address of the slave device while the last bit indicate whether master device expects to receive(last bit=0) or transmit(last bit=1).
- **Wait for acknowledge bit to send or receive:** if the master device wants to writes data into the bust then slave waits to receive acknowledge bit and vice versa. Finally if everything goes well and connection creates with slave successfully acknowledge bit is 0.
- **Send/receive 8 bit data and wait for acknowledge:** this can be done inside a loop until the end of transmission.
- **Sending stop bit:** when all of data transferred stop condition generated by setting SCL pin to high and rising edge of SDA pin.

Appendix F - Test Haversine’s formula V.S. Vincenty’s formula

<table>
<thead>
<tr>
<th></th>
<th>Vincenty’s (KM)</th>
<th>Haversine</th>
<th>Difference(Meter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.lat=59.240000,long=17.560000 lat2=59.240000,lon2=17.560000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>2.lat=59.240000,long=17.560000 lat2=59.250000,lon2=17.560000</td>
<td>1.113000</td>
<td>1.111000</td>
<td>2</td>
</tr>
<tr>
<td>3.lat=59.240000,long=17.560000 lat2=59.260000,lon2=17.570000</td>
<td>2.299000</td>
<td>2.295000</td>
<td>4</td>
</tr>
<tr>
<td>4.lat=59.240000,long=17.560000 lat2=59.270000,lon2=17.570000</td>
<td>3.390000</td>
<td>3.383000</td>
<td>7</td>
</tr>
<tr>
<td>5.lat=59.240000,long=17.560000 lat2=59.280000,lon2=17.580000</td>
<td>4.599000</td>
<td>4.590000</td>
<td>9</td>
</tr>
<tr>
<td>6.lat=64.240000,long=22.560000 lat2=64.240000,lon2=22.560000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0</td>
</tr>
<tr>
<td>7.lat=64.240000,long=22.560000 lat2=64.250000,lon2=22.570000</td>
<td>1.215000</td>
<td>1.212000</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>lat=59.240000,long=17.560000</td>
<td>lat2=59.220000,lon2=17.550000</td>
<td>2.299000</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------</td>
<td>---------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>3.</td>
<td>lat=59.240000,long=17.560000</td>
<td>lat2=59.210000,lon2=17.550000</td>
<td>3.390000</td>
</tr>
<tr>
<td>4.</td>
<td>lat=59.240000,long=17.560000</td>
<td>lat2=59.200000,lon2=17.540000</td>
<td>4.600000</td>
</tr>
<tr>
<td>5.</td>
<td>lat=59.240000,long=12.560000</td>
<td>lat2=54.240000,lon2=12.560000</td>
<td>0.000000</td>
</tr>
<tr>
<td>6.</td>
<td>lat=59.240000,long=12.560000</td>
<td>lat2=54.230000,lon2=12.550000</td>
<td>1.290000</td>
</tr>
<tr>
<td>7.</td>
<td>lat=59.240000,long=12.560000</td>
<td>lat2=54.220000,lon2=12.550000</td>
<td>2.319000</td>
</tr>
<tr>
<td>8.</td>
<td>lat=59.240000,long=12.560000</td>
<td>lat2=54.210000,lon2=12.540000</td>
<td>3.585000</td>
</tr>
<tr>
<td>9.</td>
<td>lat=59.240000,long=12.560000</td>
<td>lat2=54.200000,lon2=12.540000</td>
<td>4.639000</td>
</tr>
<tr>
<td>10.</td>
<td>lat=49.240000,long=7.560000</td>
<td>lat2=49.210000,lon2=7.550000</td>
<td>0.000000</td>
</tr>
<tr>
<td>11.</td>
<td>lat=49.240000,long=7.560000</td>
<td>lat2=49.230000,lon2=7.560000</td>
<td>1.112000</td>
</tr>
<tr>
<td>12.</td>
<td>lat=49.240000,long=7.560000</td>
<td>lat2=49.220000,lon2=7.550000</td>
<td>2.340000</td>
</tr>
<tr>
<td>13.</td>
<td>lat=49.240000,long=7.560000</td>
<td>lat2=49.210000,lon2=7.550000</td>
<td>3.415000</td>
</tr>
<tr>
<td>14.</td>
<td>lat=49.240000,long=7.560000</td>
<td>lat2=49.200000,lon2=7.540000</td>
<td>4.681000</td>
</tr>
<tr>
<td>15.</td>
<td>lat=44.240000,long=2.560000</td>
<td>lat2=44.240000,lon2=2.560000</td>
<td>0.000000</td>
</tr>
<tr>
<td>16.</td>
<td>lat=44.240000,long=2.560000</td>
<td>lat2=44.230000,lon2=2.560000</td>
<td>1.368000</td>
</tr>
<tr>
<td>17.</td>
<td>lat=44.240000,long=2.560000</td>
<td>lat2=44.220000,lon2=2.550000</td>
<td>2.361000</td>
</tr>
<tr>
<td>18.</td>
<td>lat=44.240000,long=2.560000</td>
<td>lat2=44.210000,lon2=2.540000</td>
<td>3.696000</td>
</tr>
<tr>
<td>19.</td>
<td>lat=44.240000,long=2.560000</td>
<td>lat2=44.200000,lon2=2.540000</td>
<td>4.723000</td>
</tr>
<tr>
<td>20.</td>
<td>lat=44.240000,long=2.560000</td>
<td>lat2=44.200000,lon2=2.540000</td>
<td>4.723000</td>
</tr>
<tr>
<td>21.</td>
<td>lat=39.240000,long=2.440000</td>
<td>lat2=39.230000,lon2=2.440000</td>
<td>0.000000</td>
</tr>
<tr>
<td>22.</td>
<td>lat=39.240000,long=2.440000</td>
<td>lat2=39.210000,lon2=2.440000</td>
<td>1.111000</td>
</tr>
<tr>
<td>23.</td>
<td>lat=39.240000,long=2.440000</td>
<td>lat2=39.220000,lon2=2.450000</td>
<td>2.384000</td>
</tr>
<tr>
<td>24.</td>
<td>lat=39.240000,long=2.440000</td>
<td>lat2=39.210000,lon2=2.450000</td>
<td>3.444000</td>
</tr>
<tr>
<td>25.</td>
<td>lat=39.240000,long=2.440000</td>
<td>lat2=39.200000,lon2=2.560000</td>
<td>4.762000</td>
</tr>
<tr>
<td>26.</td>
<td>lat=34.240000,long=7.440000</td>
<td>lat2=34.240000,lon2=7.440000</td>
<td>0.000000</td>
</tr>
<tr>
<td>27.</td>
<td>lat=34.240000,long=7.440000</td>
<td>lat2=34.230000,lon2=7.450000</td>
<td>1.441000</td>
</tr>
<tr>
<td>28.</td>
<td>lat=34.240000,long=7.440000</td>
<td>lat2=34.220000,lon2=7.450000</td>
<td>2.406000</td>
</tr>
<tr>
<td>29.</td>
<td>lat=34.240000,long=7.440000</td>
<td>lat2=34.210000,lon2=7.460000</td>
<td>3.809000</td>
</tr>
<tr>
<td>30.</td>
<td>lat=34.240000,long=7.440000</td>
<td>lat2=34.200000,lon2=7.460000</td>
<td>4.812000</td>
</tr>
<tr>
<td>31.</td>
<td>lat=29.240000,long=12.440000</td>
<td>lat2=29.240000,lon2=12.440000</td>
<td>0.000000</td>
</tr>
<tr>
<td>32.</td>
<td>lat=29.240000,long=12.440000</td>
<td>lat2=29.230000,lon2=12.440000</td>
<td>1.108000</td>
</tr>
<tr>
<td>33.</td>
<td>lat=29.240000,long=12.440000</td>
<td>lat2=29.220000,lon2=12.450000</td>
<td>2.426000</td>
</tr>
<tr>
<td>34.</td>
<td>lat=29.240000,long=12.440000</td>
<td>lat2=29.210000,lon2=12.450000</td>
<td>3.474000</td>
</tr>
<tr>
<td>35.</td>
<td>lat=29.240000,long=12.440000</td>
<td>lat2=29.200000,lon2=12.460000</td>
<td>4.851000</td>
</tr>
</tbody>
</table>