**A LOW COST COSPAS-SARSAT**

**TRANSPONDER**

*A Project Report*

*Submitted by*

**KEERTHI M ( Reg. No : MAC15EC072 )**

**NIRMAL ELDHO JOSE(Reg. No : MAC15EC092)**

**ATHIRA K(Reg. No : MAC15EC044)**

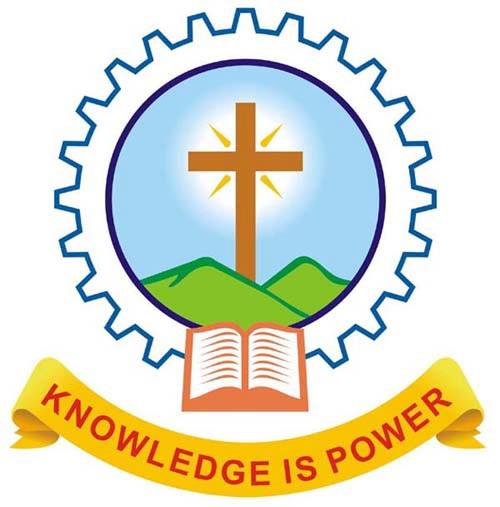
**S KRISHNA PRASAD(Reg. No : MAC15EC108)**

*In partial fulfillment of the requirements for the award of the Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION**

**ENGINEERING**

**MAR ATHANASIUS COLLEGE OF ENGINEERING**

**KOTHAMANGALAM, KERALA, INDIA 686 666**

*Affiliated to*

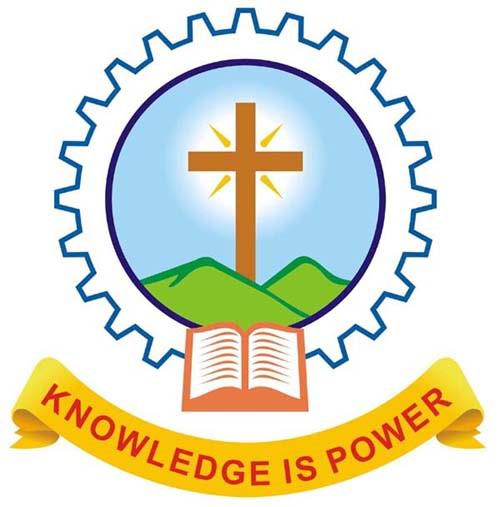
**APJ Abdul Kalam Technological University**

**Trivandram, Kerala, india**

**JUNE 2019**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**MAR ATHANASIUS COLLEGE OF ENGINEERING KOTHAMANGALAM**



**CERTIFICATE**

This is to certify that the report entitled A LOW COST COSPAS-SARSAT TRANSPONDER submitted by Miss.Keerthi M, Mr.Nirmal Eldho Jose, Mr.S Krishna Prasad, Miss.Athira K to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Electronics & Communication Engineering is a bonafide record of the project report carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose

Prof. Athira Prasad Prof.Thomas George Prof. BabuP

Kuriakose

**Project Coordinator Project Guide Head of Dept.**

**Place :**

**Date :**

**ACKNOWLEDGEMENT**

It is a great pleasure to acknowledge all those who have assisted and supported me for successfully completing our project.

We are deeply indebted to Dr. Mathew K, Principal, Mar Athanasius College of Engineering for his encouragement and support.

We thank God Almighty for his blessings as it is only through his grace that we were able to complete our project successfully.

We express our deep sense of gratitude to Prof.Babu P Kuriakose, Head of Electronics & Communication Engineering Department.

We also extend our deep sense of gratitude to our Project Guide Prof.Thomas George and Faculty Advisor, Prof. Athira Prasad, and Dr. Jinsa Kuruvilla, Professor, Electronics & Communication Engineering Department for their constant support and immense contribution for the success of our project.

We also express our gratitude to Mr.Jinesh and Mr.Lochan George for their support and suggestions throughout the project.

We whole - heartedly thank all our classmates, for their valuable suggestions and for the spirit of healthy competition that exists between us.

**ABSTRACT**

Every year calamities attack a big number of fishermen at the sea and coastal areas. Many fishermen already lost their lives and some were able to save lives. With the advancement of technology, nowadays there are various kind of Emergency Position Indicating Radio Beacon, SAR Transponder, Emergency Personnel Locators available in the market. One of the leading and most successful EPIRB is COSPAS-SARSAT; a satellite based SAR Transponder which is available for maritime use. But these devices are expensive and not affordable by the fishermen living their life below poverty line . This emerges the need of a SAR Transponder, which is relatively low in cost and also open format for platform independent and readable by any radio operator on board of a ship or shore. In this paper, we will discuss and propose of such type of an alternate solution to the COSPAS-SARSAT transponders, which is based on GPS and Radio Beacons, relatively low in cost yet reliable.

CONTENTS

[CHAPTER 1 9](#_Toc10802574)

[INTRODUCTION 9](#_Toc10802575)

[CHAPTER 2 11](#_Toc10802576)

[LITERATURE SURVEY 11](#_Toc10802577)

[CHAPTER 3 19](#_Toc10802578)

[CURRENT AVAILABLE MARITIME DISTRESS SIGNALING 19](#_Toc10802579)

[3.1 COSPAS SARSAT 19](#_Toc10802580)

[3.2 GDMSS 20](#_Toc10802581)

[CHAPTER 4 22](#_Toc10802582)

[PROJECT REQIREMENTS 22](#_Toc10802583)

[4.1 SYSTEM REQUIREMENTS 22](#_Toc10802584)

[4.2 TEST REQUIREMENTS 23](#_Toc10802585)

[CHAPTER 5 24](#_Toc10802586)

[BLOCK DIAGRAM 24](#_Toc10802587)

[CHAPTER 6 27](#_Toc10802588)

[HARDWARE DESIGN 27](#_Toc10802589)

[6.1 MICROCONTROLLER: ATMEGA328P 28](#_Toc10802590)

[6.2 UBLOX NEO 6M GPS MODULE 30](#_Toc10802591)

[6.3 LM741 32](#_Toc10802592)

[6.4 AD633 34](#_Toc10802593)

[6.5 MODULATION 35](#_Toc10802594)

[CHAPTER 7 36](#_Toc10802595)

[SOFTWARE DESIGN 36](#_Toc10802596)

[7.1 NEO 6M CONFIGURATION 36](#_Toc10802597)

[7.2 CONTROLLER CODING 37](#_Toc10802598)

[CHAPTER 8 43](#_Toc10802599)

[BUDGET 43](#_Toc10802600)

[CHAPTER 9 44](#_Toc10802601)

[RESULTS 44](#_Toc10802602)

[CHAPTER 10 46](#_Toc10802603)

[CONCLUSION 46](#_Toc10802604)

[REFERENCES 47](#_Toc10802605)

[Appendices 48](#_Toc10802606)

[CONTROLLER CODE 49](#_Toc10802607)

LIST OF FIGURES

No. Title Page No.

2.1 INSAT 16

2.2 INMCC,RCC and SPOCS 17

3.1 COSPAS SARSAT devices 20

3.2 GMDSS Equipment 20

5.1 Block Diagram 25

6.1 Schematic of EPIRB 27

6.2 Atmega328P 28

6.3 Pin Diagram of Atmega328P 30

6.4 NEO 6M GPS Module with integrated antenna 31

6.5 Pin Configuration of LM741 32

6.6 Opam Waveforms 33

6.7 AD633 IC 34

6.8 Pin Configuration of AD633 35

7.1 Ublox Readings Before Lock 36

7.2 Ublox Readings After Lock 37

7.3 GPRMC Fields 38

7.4 Preamble of Long Message 38

7.5 Standard Location Protocol MMSI 38

7.6 Coordinate Conversion 39

7.7 Data Encoding and Modulation Sense 42

9.1 Hardware 44

9.2 Modulator Output 45

9.3 U-CENTER output 45

# CHAPTER 1

## INTRODUCTION

The total length of the world's coastlines is about 504,000 km, enough to circle the Equator 12 times. Coastal areas comprise 20% of the earth's surface yet contain over 50 per cent of the entire human population. By the year 2025, coastal populations are expected to account for 75% of the total world population. More than 70% of the world's mega cities (greater than 8 million inhabitants) are located in coastal areas. Half of the world's cities with more than one million people are sited in and around estuaries. The threats to coastal communities include extreme natural events such as hurricanes, coastal storms, tsunamis, and landslides, as well as longer-term risks of coastal erosion and sea level rise. The total number of commercial fishermen and fish farmers is estimated to be 38 million around the world, past decade they are continuously threatened by marine calamities, sea pirate attacks, natural and man-made disaster also. According to the International Maritime Bureau the number of sea pirate attacks in Bangladesh jumped three-fold to 11 in 2017. Every year they face attack from the pirates of Bay of Bengal which made the life of fishermen at sea and coastal area very tough. Most commercial fishermen are low-caste Hindus who eke out the barest subsistence working under primitive and dangerous conditions. Every time they set out to the sea to fish, they gamble with their own lives. A total of 11 piracy events took place o the coast of Bangladesh in 2012.In the last five years; pirates have killed at least 411 fishermen and wounded at least 1,000 more. Not only Bangladesh, but a lot of countries around the world suffer from marine distress as hurricanes, tropical storms, tsunamis, and landslides have the potential to generate a tremendous amount of marine debris. Recent marine distress occurred at Kerala over 483 people died, and 14 are missing.one-sixth of the total population of Kerala had been directly affected by the floods and related incidents.

This report gives us a brief idea on the system block for the alternative EPIRB Transponder. Also discuss on the message format that we will send to the shore as a distress code. We included several proposals on RF and message encoding. .We also have the current stage of the prototype and result of the first test run at field. Finally we will close our discussion by our conclusion on the topic, which will help us to design a low cost maritime EPIRB, which may save a lot of life of those fishermen in sea.

# CHAPTER 2

## LITERATURE SURVEY

The International COSPAS-SARSAT Programme is a satellite-aided search and rescue initiative. It is organized as a treaty-based, non-pro t, intergovernmental, humanitarian cooperative of 45 nations and agencies. It is dedicated to detecting and locating radio beacons activated by persons, aircraft or vessels in distress, and forwarding this alert information to authorities that can take action for rescue. The system utilizes a net-work of satellites that provide coverage anywhere on Earth. Distress alerts are detected, located and forwarded to over 200 countries and territories at no cost to beacon owners or the receiving government agencies. COSPAS-SARSAT was conceived and initiated by Canada, France, the United States, and the former Soviet Union in 1979. The first rescue using the technology of COSPAS-SARSAT occurred in September 1982. The definitive agreement of the organization was signed on 1 July 1988.Between September 1982 and December 2017 the COSPAS-SARSAT System provided assistance in rescuing at least 46,553 people in 13,627 SAR events. COSPAS-SARSAT cooperates with United Nations-affiliated agencies, such as the International Civil Aviation Organization (ICAO), the International Maritime Organization (IMO), and the International Telecommunication Union (ITU), among other international organizations, to ensure the compatibility of the COSPAS-SARSAT distress alerting services with the needs, the standards and the applicable recommendations of the global community. COSPAS-SARSAT is an element of the IMO's Global Maritime Distress Safety System (GMDSS), and is expected to become a component of ICAO's Global Aeronautical Distress and Safety System (GADSS). The IMO requires automatic-activating COSPAS-SARSAT beacons (EPIRBs) on all vessels subject to requirements of the International Convention for the Safety of Life at Sea (so-called SOLAS-class vessels), commercial fishing vessels, and all passenger ships in international waters. Similarly, ICAO requires COSPAS-SARSAT beacons aboard aircraft on international flights. National administrations often impose requirements in addition to the international requirements of those agencies. COSPAS-SARSAT only monitors for alerts from digital distress beacons that transmit on 406 MHz (so-called 406 beacons). Older beacons that transmit only a legacy analogue signal on 121.5 MHz or 243 MHz rely on being received only by nearby aircraft or rescue personnel. For satellite reception of alerts by COSPAS-SARSAT the beacon must be a model that transmits at 406 MHz. COSPAS-SARSAT has received many honours for its humanitarian work. One recent honour was induction into the Space Foundation's Space Technology Hall of Fame for space technologies improving the quality of life for all humanity. A COSPAS-SARSAT distress beacon is a digital 406-MHz radio transmitter that can be activated in a life-threatening emergency to summon assistance from government authorities. Beacons are manufactured and sold by dozens of vendors. They are classified in three main types. A 406-MHz beacon designed for use in an aircraft is known as an emergency locator transmitter (ELT). One designed for use aboard a marine vessel is called an emergency position-indicating radio beacon (EPIRB). And one that is designed to be carried by an individual is known as a personal locator beacon (PLB). Sometimes PLBs are carried aboard aircraft or vessels, but whether this satisfies safety requirements depends on local regulations. A COSPAS-SARSAT 406-MHz beacon does not transmit until it is activated in an emergency (or when certain testing features are activated by the user). Some beacons are designed to be manually activated by a person pressing a button, and some others are designed for automatic activation in certain circumstances (e.g., ELTs may be automatically activated by a physical shock, such as in a crash, and EPIRBs may be automatically activated by contact with water). There are no subscriptions or other costs imposed by COSPAS-SARSAT for beacon ownership or use. (Some countries may impose licensing and or registration charges for beacon ownership, and some jurisdictions may assess costs for rescue operations.)

The COSPAS-SARSAT system space segment consists of SARR and,or SARP instruments aboard: 1.Five satellites in polar low-altitude Earth orbit called LEOSARs 2.Seven satellites in geostationary Earth orbit called GEOSARs 3.Over 35 satellites in medium-altitude Earth orbit called MEOSARs .A SARR or SARP instrument is a secondary payload and associated antennas attached to those satellites as an adjunct to the primary satellite mission. A SARR instrument retransmits a beacon distress signal to a satellite ground station in real time. A SARP instrument records the data from the distress signal so that the information can later be gathered by a ground station when the satellite passes overhead. The satellites are monitored by receiving ground stations equipped to track (point at and follow) the satellites using satellite dishes or phased antenna arrays called local user terminals (LUT). LUTs are installed by individual national administrations or agencies. The distress messages received by a LUT are transferred to an associated mission control centre which uses a detailed set of computer algorithms to route the messages to rescue coordination centres worldwide. When a distress beacon is activated, the COSPAS-SARSAT system, decodes the binary coded message of the beacon, which contains information such as the identity of the vessel, aircraft and, for beacons equipped with the feature, the location of the beacon derived from a local navigation source (such as a GPS receiver incorporated into the beacon's design)performs a mathematical analysis of the signal to calculate the location of the beacon, even if the beacon's location is not reported in the distress message. The COSPAS-SARSAT system is the only satellite distress alerting system that is capable of this dual, redundant means of locating an activated distress beacon. The SARR and or SARP instrument typically is attached to a satellite that is being launched primarily for another purpose. The primary mission of all of the LEOSAR and GEOSAR satellites is meteorological (gathering of weather data).The primary mission of all of the MEOSAR satellites is navigation. The first system satellite, "COSPAS-1" (Kosmos 1383), was launched from Plesetsk Cosmodrome on June 29, 1982. COSPAS-SARSAT began tracking the two original types of distress beacons, EPIRBs and ELTs, in September, 1982. The first persons were rescued with the assistance of COSPAS-SARSAT when the distress signal from a small plane was relayed by the COSPAS-1 satellite to a then-experimental ground station in Ottawa, Ontario, Canada. The story has been related by the plane's pilot, Jonathan Ziegelheim, who rescue authorities judged would probably have died of his injuries if it were not for COSPAS-SARSAT. In the early 2000s (in 2003 in the USA) a new type of distress beacon, the personal locator beacon (PLB), became available for use by individuals who cannot contact emergency services through normal telephone-originated services, such as 1-1-2 or 9-1-1. Typically PLBs are used by people engaged in recreational activities in remote areas, and by small-aircraft pilots and mariners as an adjunct to (or, when permitted, a substitute for) an ELT or EPIRB. The four founding Party States led development of the 406-MHz marine EPIRB for detection by the system. The EPIRB was seen as a key advancement in SAR technology in the perilous maritime environment. Prior to the founding of COSPAS-SARSAT, the civilian aviation community had already been using the 121.5 MHz frequency for distress, while the military aviation community utilized 243.0 MHz as the primary distress frequency with the 121.5 MHz frequency as the alternate. ELTs for general aviation aircraft were constructed to transmit on 121.5 MHz, a frequency monitored by airliners and other aircraft. Military aircraft beacons were manufactured to transmit at 243.0 MHz, in the band commonly used by military aviation.

Early in its history, the COSPAS-SARSAT system was engineered to detect beacon-alerts transmitted at 406 MHz, 121.5 MHz and 243.0 MHz. Because of a large number of false alerts, and the inability to uniquely identify such beacons because of their old, analogue technology, the COSPAS-SARSAT system beginning in 2009 stopped receiving alerts from beacons operating at 121.5 MHz and 243.0 MHz, and now only receives and processes alerts from modern, digital 406-MHz beacons. Many ELTs include both a 406-MHz transmitter for satellite detection and a 121.5 MHz transmitter that can be received by local search crews using direction-finding equipment. The design of distress beacons as a whole has evolved significantly since 1982. The newest 406-MHz beacons incorporate GPS receivers. Such beacons transmit in their distress message highly accurate position reports. The distress alert and location are forwarded almost instantly to SAR agencies via COSPAS-SARSAT satellites. This provides a second method for COSPAS-SARSAT to know the location of the distress, in addition to the calculations independently done by COSPAS-SARSAT LUTs to determine the location. This two-tiered reliability and global coverage of the system has inspired the current motto of SAR agencies: "Taking the 'Search' out of Search and Rescue". COSPAS (КОСПАС) is an acronym for the Russian words "Cosmicheskaya Sistema Poiska Avariynyh Sudov" (Космическая Система Поиска Аварийных Судов), which translates to "Space System for the Search of Vessels in Distress". SARSAT is an acronym for Search And Rescue Satellite-Aided Tracking.

**HANDBOOK OF BEACON REGULATIONS**: This document provides a summary of regulations issued by COSPAS-SARSAT Participants regarding the carriage of 406 MHz beacons. It also includes practical information on coding and registration requirements in each country, where such information was made available to the COSPAS-SARSAT Secretariat. It also includes the following information: The list of type approved beacons, Details on points of contact for beacon matters , And beacon test facilities information. Examples of beacon registration cards are not included in the document, but are available at the Secretariat and can be provided on request. This document is based mainly on information provided by Participants at COSPAS-SARSAT meetings and in reports on System status and operations. Some information was provided by non-COSPAS-SARSAT Participants. However, regulations are likely to evolve and the attached information should not be regarded as an official record of their current status. Participants are invited to provide the COSPAS-SARSAT Secretariat with updates as appropriate. India is a member of the international COSPAS-SARSAT programme for providing distress alert and position location service through LEOSAR (Low Earth Orbit Search And Rescue) satellite system. Under this programme, India has established two Local User Terminals (LUTs), one at Lucknow and the other at Bengaluru. The Indian Mission Control Centre (INMCC) is located at ISTRAC, Bengaluru. The system is operational from the past 23 years.INSAT-3A, located at 93.50 East and INSAT-3D located at 820 East, are equipped with a 406 MHz Search and Rescue payload that picks up and relays alert signals originating from the distress beacons of maritime, aviation and land users. Indian LUTs provide coverage to a large part of the Indian Ocean region rendering distress alert services to Bangladesh, Bhutan, Maldives, Nepal, Seychelles, Sri Lanka and Tanzania. The operations of INMCC,LUT are funded by the participating agencies, namely, Coast Guard, Airports Authority of India (AAI), and Directorate General of Shipping and Services. INSAT GEOSAR Local User Terminal (GEO LUT) is established at ISTRAC, Bengaluru and integrated with INMCC. The distress alert messages concerning the Indian service area, detected at INMCC, are passed on to Maritime Rescue Coordination Centres (MRCCs) of Indian Coast Guard (Mumbai, Chennai, Port Blair), and Rescue Coordination Centres (RCCs) of AAI (Mumbai, Kolkata, Delhi, Chennai). The search and rescue activities are carried out by Coast Guard, Navy and Air Force. INMCC is linked to the RCCs, MRCCs, SPOCs (Search and Rescue Points of Contact) and other International MCCs (Mission Control Centres) through Aeronautical Fixed Telecommunication Network (AFTN). The Indian LUTs and MCC provide round the clock service and maintain the database of all 406 MHz registered beacons carried on-board Indian ships and aircraft. During the year 2013, INMCC provided search and rescue support to 14 distress incidents in Indian service area through Indian system and contributed to saving 94 human lives. During 2013, about 962 new radio beacons were added in Indian database (most of them for maritime applications). Till date, there are about 716 registered user agencies (Maritime Aviation) in India with an Indian beacon population of more than 13,636 in our database.INSAT-3D, the advanced metrological satellite located at 820 East longitude carries a Search and Rescue (SAR) Payload. The satellite was launched on July 26, 2013 and the in-orbit-testing of SAR payload was completed in Aug 2013. INSAT-3D will provide the continuity to SAR service.



Figure 2.1: INSAT

The Indian Mission Control Centre (INMCC), which is responsible for providing distress alerting services to National Aeronautical and Maritime Rescue Coordination Centres (ARCCs and MRCCs), Search and Rescue Points of Contact (SPOC) in seven neighbouring countries and other international Mission Control Centres (MCCs), is co-located with the Bangalore LUT. Following diagram represents the INMCC service area covering 4 national Search and Rescue zones supported by ARCCs and MRCs, and 7 SPOCs.

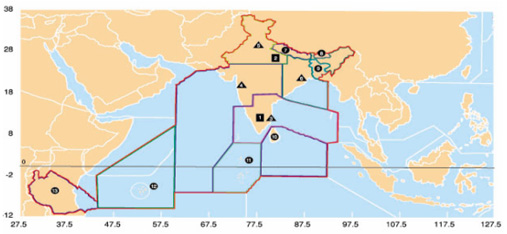


Figure 2.2: INMCC,RCC and SPOCS

INDIAN LUTS AND MCC

1. Bangalore (LEOLUT, GEOLUT, INMCC)
2. Lucknow (LEOLUT)

INDIA RCCS

* Chennai
* Mumbai
* Delh
* iCalcutta

INMCC SPOCs

1. Nepal
2. Bhutan
3. Bangladesh
4. Srilanka
5. Maldives
6. Seychelles
7. Tanzania

REGISTRATION PROCEDURE FOR INDIAN BEACONS:

The registration of 406 MHz distress Beacon is mandatory with INMCC as per national (DG Shipping and DGCA) and international (IMO, ICAO) regulations. It is free of charge and the help is extended with no loss of time in case of distress. Non-registration and inadvertent activation of beacons may be subjected to necessary disciplinary action. Beacon registration data includes primary and alternate emergency points of contact that SAR forces can call when a distress signal is received. Additional information is also included. For EPIRBs the vessel name, type, length, color, capacity and homeport are listed as well as type of communication equipment and call sign. For ELTs, the aircraft number, make, model, colour, capacity and home airport, FBO are listed. Both registration forms also contain room for additional information that may help locate the vessel or aircraft sooner. The registration of beacon helps discriminate false alarms quickly saving SAR resources and efforts by the SAR forces. The registration information is stored securely at the INMCC and used only for search and rescue purposes.

# CHAPTER 3

## CURRENT AVAILABLE MARITIME DISTRESS SIGNALING

### 3.1 COSPAS SARSAT

COSPAS-SARSAT is a satellite based Search And Rescue (SAR) program for distress information, which was established on 1979 by Canada, France, USA and former Soviet Union. Fig. 1 shows the most common type of products for this type of device. These devices are very handy to use, battery operated. These normally does not require any switch to turn on and on contact of sea water, they gets turn on and get the position of the beacon terminal using a GPS receiver embedded in it. It then sends the position data to the satellite on 406 MHz with an EIRP of 5W (a 100 kHz band with centre frequency of406.05MHz). The transmitted distress message can be 112 bits or 144 bits (short or long) of data [4]. Also some device transmits a homing beacon on 121.5 MHz, which is for secondary basis as 121.5MHz is primarily used by civil aircraft distress radio beacons. Ref [4] has very detail of COSPAS-SARSAT with its working principle and system composition. Regardless of ease of use, each of these device price ranges from Us dollar 400.00 and these requires registration to COSPAS in order to identify any terminal activated in distress. Nowadays most of the merchant marine vessels are equipped with it, but local fishermen in Bangladesh are not equipped with it yet.



Figure 3.1: COSPAS SARSAT devices

### 3.2 GDMSS

The Global Maritime Distress and Safety System (GMDSS) is another form of distress signaling for marine vessels. GMDSS also works with the assistance of GPS. It uses Digital Selective Calling (DSC) in marine HF and VHF channels. These are basically HF and VHF Radio Transceivers with the capability of sending text and distress signals over narrow band RadioTelex.



Figure 3.2: GMDSS Equipment

These devices have a single button that requires to be pressed during distress, so that any distress message along with the GPS coordinates will be instantly sent as a text message for direct printing to another GMDSS equipment. Ref [5] and [6] has very detail about this technology and products available. Ref [6] and [7] tells that, a GMDSS station works on marine VHF Ch 16 (156.80MHz) for voice and Ch 70 (156.525MHz) for DSC signaling. Also for HF GMDSS radios,4125 kHz, 6215 kHz, 8291 kHz, 12.290 MHz and 16.420 MHz is used for long distance voice distress calls and 2187.5 kHz, 4207.5 kHz, 6312 kHz, 8414.5 kHz, 12577 kHz and 16804.5kHz is used for DSC signaling. GMDSS has total four areas of operation. Sea Area A1covers 30 nautical miles (56 km) to 40 nautical miles (74 km) from the shore station. VHF Ch 70 is used in this area for distress operation with DSC. Sea Area A2 uses MF, 2187.5kHz in DSC and covers 180 nautical miles (330 km) of the offshore during daytime and in night time it extends to up to 400 nautical miles (740 km) due to good MF/HF propagation. Sea Area A2 excludes coverage of Sea Area A1. The other two areas, Sea Area A3 and A4 are for more long distance with exclusion of A1, A2 for A3 and A1, A2, A3 for A4.Bangladeshi fishermen works mostly in Sea Area A1 and some go beyond A1, but does not reach A2.Many of the merchant marine vessels nowadays are equipped with GMDSS radios, but still it is not so popular in Bangladesh due to need of registration of each station. Also GMDSS enabled radios are costly in comparison to Bangladesh economy and especially for local fishermen.

# CHAPTER 4

## PROJECT REQIREMENTS

The COSPAS SARSAT documents are tailored to obtain the system or functional requirements and the test requirements.

### 4.1 SYSTEM REQUIREMENTS

* The equipment shall be compatible with the COSPAS-SARSAT system as per appropriate tailoring from the HANDBOOK OF BEACON REGULATIONS [REF-1] and specifically the regulations for India [REF-2].
* The tailoring document for compliance to [REF-1] and [REF-2] with any nested requirements shall be established with necessary justifications. Note: This document should become part of the Specification for the equipment.
* The equipment shall include a 406 MHz satellite uplink with geo-location received from a built-in GPS receiver.
* The prototype design shall be compatible to be used in marine environments.
* The equipment should be waterproof.
* The prototype design shall be compatible to be used in land environments. The equipment should be able to withstand a fall from 20m.
* The prototype should not cost more than 10,000 INR to make.
* The prototype should fit in a casing smaller than 10cm x 10cm x 10cm.
* The prototype should not weigh more than 250 gm.
* The prototype battery should last for 2 years, without activation.
* The prototype should support a minimum of 20 activations.
* Once activated, the beacon must be cancels within 1 minute.
* The beacon should be protected against unintended activation either by accident.
* The equipment shall include a provision for coded message in addition to the geo-location received from the GPS receiver.
* The GPS Receiver shall be designed with location accuracy of 10m.

### 4.2 TEST REQUIREMENTS

* The 406 MHz beacons shall be designed with a self-test capability for evaluating key performance characteristics.
* Initiating the beacon self-test function must not generate a distress alert in the COSPAS-SARSAT System.

# CHAPTER 5

## BLOCK DIAGRAM

As stated earlier, the goal is to send the position data of a fishing boat in distress to shore over RF, the EPIRB Transponder will be installed on a fishing boat as a single box device with built-in power source. The device will have only one switch, which will be used to turn ON the device to signal a distress. An additional switch will be there for a TEST beacon, which will also work in the same manner, but will not send

Distress instead it will exclusively send test beacon. The all-in-a- box device will only have at least two RF connectors, where on is for receiving GPS signals and the other one is to the beacon transmitting antenna. In case of multiple transmissions like one for distress position reporting and other for local homing beacon (like COSPAS-SARSAT’s 406MHz and 121.5MHz pair), there may be multiple RF-OUT connectors for different radiating antennas. On pressing the distress button, a fully functional EPIRB will be powered on and will lock itself with the GPS satellites. Once locked, the GPS data (position coordinates) will be transferred to the onboard processor and processor will convert the position data into regular format from NMEA 0183 format [as most GPS gives readout in NMEA 0183 format]. The processor will also prepare the distress message. Then the processor will encode the position report to the signalling method of desire like PSK or OOK, etc. After this, the encoded data will be transmitted on the air using the radio transmitter(s) for sending the position data.

The proposed EPIRB will be comprised of the following blocks:

GPS SUB-SYSTEM:

This sub-system is a plain GPS module capable of detecting GPS satellites for Lat-Lon coordinates and local time. The output is NMEA 0183 format, so that it is universal and easy to interface with any system. While selecting a GPS module, parameters like start-up time, satellite locking speed, power consumption, etc. should be carefully considered for price vs. performance.

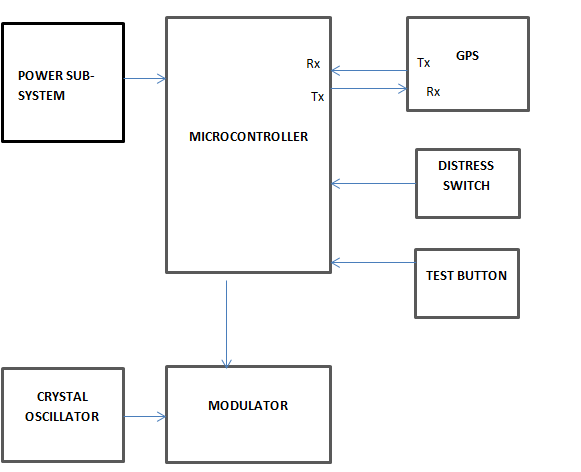


Figure 5.1: Block Diagram

CONTROLLER SUB-SYSTEM:

This is the heart of the whole system, which will be connected to other sub-systems of the whole device. This is responsible for entire message creation and encoding along with sending data to transmitter and to act like a transponder by sending position data at an interval.

MODULATOR SUB-SYSTEM:

To modulate the encoded information with carrier and transmit it to the RF section of the system, Here the carrier must be a high frequency signal to avoid signal attenuation. Output of the modulator is a digital PSK signal so that we will be able to distinguish the message signal from the noise at the receiver end.

CRYSTAL OSCILLATOR:

Temperature compensated crystal oscillator (TCXO) are used to achieve high level of accurate and stable high frequency carrier signal. It provides a means of counter acting the frequency change caused by temperature change in a crystal oscillator.

POWER SUB-SYSTEM:

The power sub-section is self-explanatory. This will provide power to the entire transponder to operate. This can be any DC power source, inbuilt to make the entire transponder a single box device. This will be comprised of battery banks as power storage and also power supply circuits and power regulators for various sub-systems. For our particular case, we are not proposing any PV array as an alternative power source for keep the battery standby. As the device will be turned on during distress only, the power in the device will be stored for a long time. But as batteries have self-discharge, we propose here for firstly Li-Ion batteries for extended life and less weight. Also we are proposing to avoid hi grade Li-Ion batteries to reduce price. Instead, a charging mechanism can be either inbuilt or a separate charger can be used to charge the transponder during boat's maintenance time after each sail.

# CHAPTER 6

## HARDWARE DESIGN

As it is difficult to implement and test the system at a high frequency of 406 MHz it is favorable to scale down the frequency to develop a proof of concept circuit design of EPIRB and the homing section of EPIRB can be avoided in order to reduce the price of the device and the probability of false alarm. By doing sufficient scaling and satisfying the hardware requirement tailored by COSPASS SARSAT regulations a schematics has been developed.

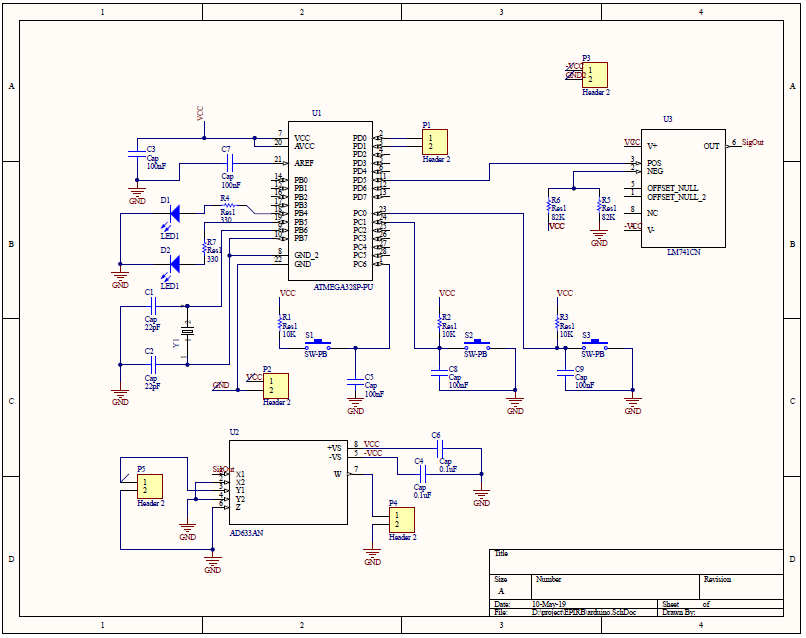


Figure 6.1: Schematic of EPIRB

The details of each component used in the above circuit design will be explained in detail below:

### 6.1 MICROCONTROLLER: ATMEGA328P

The ATmega328 is a single-chip micro controller created by Atmel in the mega AVR family (later Microchip Technology acquired Atmel in 2016). It has a modified Harvard architecture 8-bit RISC processor core. The Atmel 8-bit AVR RISC-based micro controller combines 32kB ISP flash memory with read-while-write capabilities, 1kB EEPROM, 2kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz.



Figure 6.2: Atmega328P

FEATURES:

* High Performance, Low Power AVR® 8-Bit Micro controller
* Advanced RISC Architecture
* 131 Powerful Instructions – Most Single Clock Cycle Execution
* 32 x 8 General Purpose Working Registers
* Fully Static Operation
* Up to 20 MIPS Throughput at 20 MHz
* On-chip 2-cycle Multiplier
* High Endurance Non-volatile Memory Segments
* 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48P/88P/168P/328P)
* 256/512/512/1K Bytes EEPROM (ATmega48P/88P/168P/328P)
* 512/1K/1K/2K Bytes Internal SRAM (ATmega48P/88P/168P/328P)
* Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
* Data retention: 20 years at 85°C/100 years at 25°C(1)
* Optional Boot Code Section with Independent Lock Bits
* In-System Programming by On-chip Boot Program
* True Read-While-Write Operation
* Programming Lock for Software Security

Peripheral Features:

* Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
* One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
* Real Time Counter with Separate Oscillator
* Six PWM Channels
* 8-channel 10-bit ADC in TQFP and QFN/MLF package Temperature Measurement
* 6-channel 10-bit ADC in PDIP Package Temperature Measurement
* Programmable Serial USART
* Master/Slave SPI Serial Interface
* Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
* Programmable Watchdog Timer with Separate On-chip Oscillator
* On-chip Analog Comparator
* Interrupt and Wake-up on Pin Change

Special Micro controller Features:

* Power-on Reset and Programmable Brown-out Detection
* Internal Calibrated Oscillator
* External and Internal Interrupt Sources
* Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

Operating Voltage:

* 1.8 - 5.5V for ATmega328P

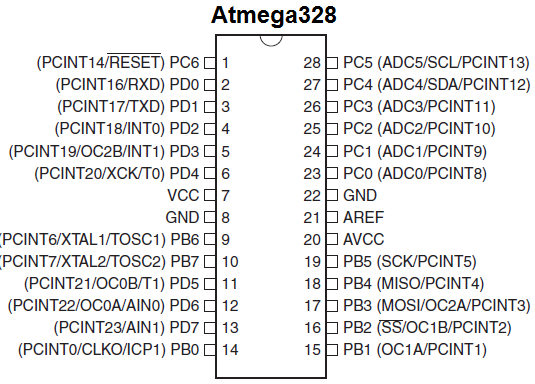


Figure 6.3: Pin Diagram of Atmega328P

### 6.2 UBLOX NEO 6M GPS MODULE

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 module is ideal for battery operated mobile devices with very strict cost and space constraints. The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.



Figure 6.4: NEO 6M GPS Module with integrated antenna

Specifications:

* Supply Voltage:2.7 to 3.6V
* Supply current:67 mA
* Antenna gain:50 dB
* Operating temperature:-40 to 85°C
* Antenna Type: Passive and active antenna
* Sensitivity
  + Tracking and Navigation:-160 dBm
  + Reacquisition:-160 dBm
  + Cold Start (Autonomous):-146 dBm

Pin Description:

* Vcc-Supply Voltage
* Gnd-Ground pin
* TX and RX-These 2 pins acts as an UART interface for communication

### 6.3 LM741

The LM741 series are general purpose operational amplifiers. It is intended for a wide range of analog applications. The high gain and wide range of operating voltage provide superior performance in integrator, summing amplifier, and general feedback applications.

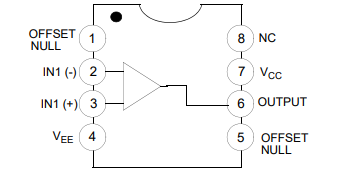


Figure 6.5: Pin Configuration of LM741

The amplifier's differential inputs consist of a non-inverting input (+) with voltage V+ and an inverting input (-) with voltage V- ideally the op-amp amplifies only the difference in voltage between the two, which is called the differential input voltage. The output voltage of the op-amp Vo is given by the equation :

where A0 is the open-loop gain of the amplifier (the term "open-loop" refers to the absence of a feedback loop from the output to the input).

Open-loop amplifier:

The magnitude of A0 is typically very large (100,000 or more for integrated circuit op-amps), and therefore even a quite small difference between and drives the amplifier output nearly to the supply voltage. Situations in which the output voltage is equal to or greater than the supply voltage are referred to as saturation of the amplifier. The magnitude of Ao is not well controlled by the manufacturing process, and so it is impractical to use an open-loop amplifier as a stand-alone differential amplifier.

Without negative feedback, and perhaps with positive feedback for regeneration, an op-amp acts as a comparator. If the inverting input is held at ground (0 V) directly or by a resistor Rg and the input voltage Vin applied to the non-inverting input is positive, the output will be maximum positive; if Vin is negative, the output will be maximum negative. Since there is no feedback from the output to either input, this is an open-loop circuit acting as a comparator.

Closed Loop Amplifier : If predictable operation is desired, negative feedback is used, by applying a portion of the output voltage to the inverting input. The closed-loop feedback greatly reduces the gain of the circuit.

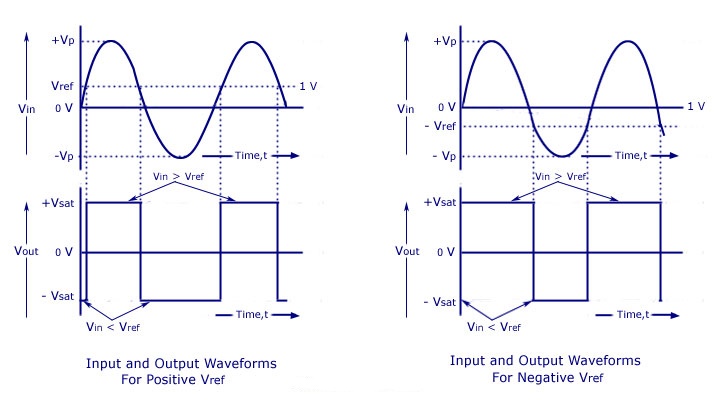


Figure 6.6: Opam Waveforms

### 6.4 AD633

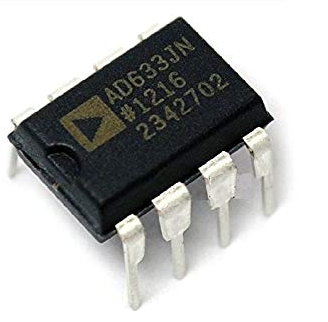


Figure 6.7: AD633 IC

The AD633 is a functionally complete, four-quadrant, analog multiplier. It includes high impedance, differential X and Y inputs, and a high impedance summing input (Z). The low impedance output voltage is a nominal 10 V full scale provided by a buried Zener. The AD633 is the first product to offer these features in modestly priced 8-lead PDIP and SOIC packages. The AD633 is laser calibrated to a guaranteed total accuracy of 2% of full scale. Nonlinearity for the Y input is typically less than 0.1% and noise referred to the output is typically less than 100 μV rms in a 10 Hz to 10 kHz bandwidth. A 1 MHz bandwidth, 20 V/μs slew rate, and the ability to drive capacitive loads make the AD633 useful in a wide variety of applications where simplicity and cost are key concerns. The versatility of the AD633 is not compromised by its simplicity. The Z input provides access to the output buffer amplifier enabling the user to sum the outputs of two or more multipliers, increase the multiplier gain, convert the output voltage to a current, and configure a variety of applications. For further information, see the Multiplier Application Guide. The AD633 is available in 8-lead PDIP and SOIC packages. It is specified to operate over the 0°C to 70°C commercial temperature range (J Grade) or the −40°C to +85°C industrial temperature range (A Grade).

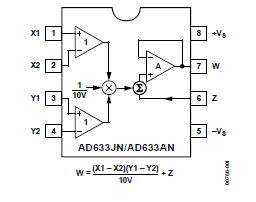


Figure 6.8: Pin Configuration of AD633

FEATURES:

* 4-quadrant multiplication
* Low cost, 8-lead SOIC and PDIP packages
* Complete—no external components required
* Laser-trimmed accuracy and stability
* Total error within 2\% of full scale
* Differential high impedance X and Y inputs
* High impedance unity-gain summing input
* Laser-trimmed 10 V scaling reference

APPLICATIONS:

* Multiplication, division, squaring
* Modulation/demodulation, phase detection
* Voltage-controlled amplifiers/attenuators/filters

### 6.5 MODULATION

The carrier shall be phase modulated positive and negative (1.1) +/- (0.1) radians peak, referenced to an un modulated carrier. Positive phase shift refers to a phase advance relative to nominal phase. The rise (tr ) and fall (tf )times of the modulated waveform, shall be (150 + 100) us.

Modulation symmetry shall be such that: 0.05

# CHAPTER 7

## SOFTWARE DESIGN

The software design is accomplished in modular design and consists of different functions for message generation, BCH encoding, Bi Phase L conversion and the random delay modules. The software is developed in Arduino platform. Configuration NEO 6M module is done using U CENTER software.

The details are as follows:

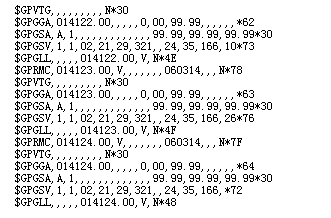


Figure 7.1: Ublox Readings Before Lock

### 7.1 NEO 6M CONFIGURATION

Connect the UART GPS NEO-6M module to a serial module. FT232 is applied as the serial module in this document. The connection between the GPS module and the serial module is listed as the table below. Connect the serial module to the computer. Start the serial debugging assistant, and set the serial port as below:Select the corresponding serial port in the Port No option. In this example, COM8 is selected.

Baud rate: 9600(default).8 data bits, 1 stop bit, no parity and no flow control.After finishing the settings, you can see the serial debugging assistant will showman data in its window, as the figure shows below.

The data shown in the figure above means that the connection is established successfully, but it is unable to perform positioning.

In this case, please place the GPS module to the balcony or near the window, or outdoors for testing, because GPS is less stable in searching signal indoors.

After waiting for a while, if the serial debugging assistant lists similar data inits window as the figure shows below, it means GPS has performed positioning.And you can see that the LED on the module, which remains on when GPS is unable to perform the positioning, is flickering now.

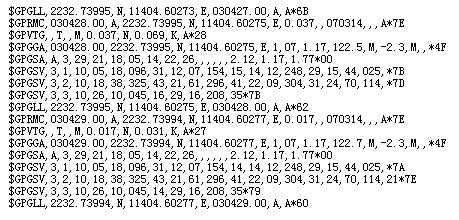


Figure 7.2: Ublox Readings After Lock

### 7.2 CONTROLLER CODING

#### 7.2.1 GPS EXTRACTION

For this project we have used GPRMC data. We will illustrate these information with an example of GPRMC,030742.00,A,2232.73830,N,11404.58520,E,0.356,,070314,,,A\*77

RECOMMENDED MINIMUM COORDINATES:

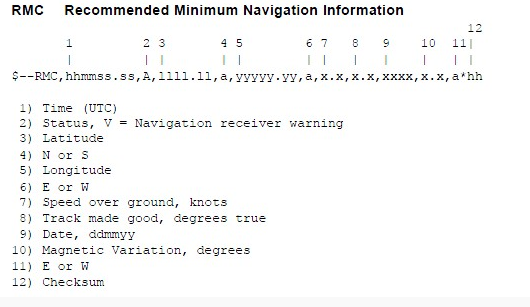


Figure 7.3: GPRMC Fields

#### 7.2.3 DIGITAL MESSAGE GENERATOR

Repetition Period:

The repetition period shall not be so stable that any two transmitters appear to be

synchronized closer than a few seconds over a 5-minute period. The intent is that no two beacons will have all of their bursts coincident. The period shall be randomised around a mean value of 50 seconds, so that time intervals between transmissions are randomly distributed on the interval 47.5 to 52.5 seconds.

Total Transmission Time:

The total transmission time, measured at the 90 per cent power points, shall be 440 ms +1 per cent for the short message and 520 ms +1 perc ent for the long message.

Un modulated Carrier:

The initial 160 ms +1 per cent of the transmitted signal shall consist of an un modulated carrier at the transmitter frequency measured between the 90 percent power point and the beginning of the modulation.

Long Message:

The final 360 ms +1 per cent of the transmitted signal shall contain a 144-bit message at a bit rate of 400 bps +1 per cent.

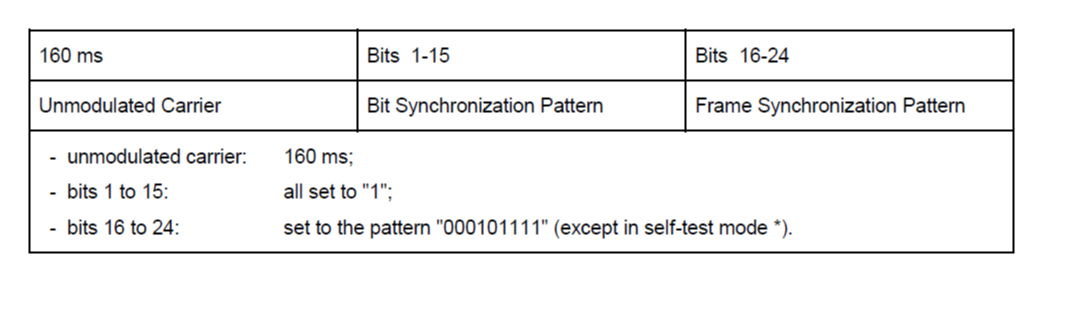


Figure 7.4: Preamble of Long Message

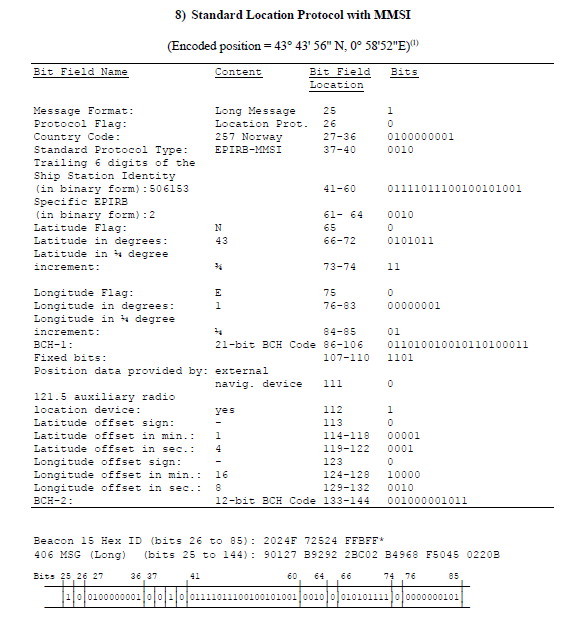


Figure 7.5: Standard Location Protocol MMSI

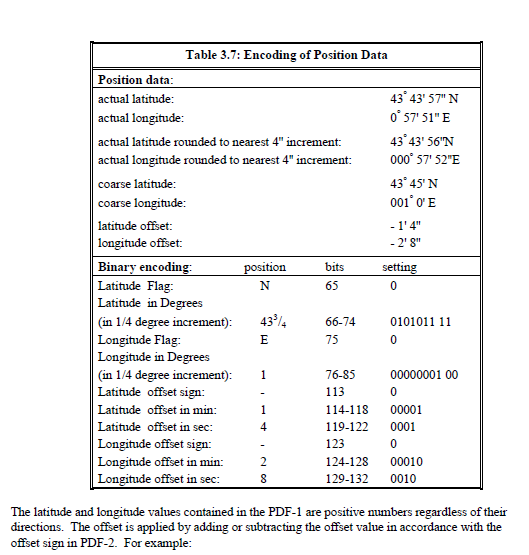


Figure 7.6: Coordinate Conversion

#### 7.2.3 BCH CODING

21-BIT BCH CODE CALCULATION:

The error-correcting code used in the first protected field of all 406 MHz messages is a shortened form of a (127,106) Bose-Chaudhuri-Hocquenghem (BCH) code. The shortened form (82,61) consists of 61 bits of data followed by a 21-bit triple error-correcting code. The code is used to detect and correct up to three errors in the entire 82-bit pattern (bits 25 through 106 of the 406-MHz message). Note: For the purpose of error correction, all calculations shall be performed with the full length code. Therefore, 45 zeros are placed before the 61 data bits to form the 106 bit pattern of the (127,106) BCH code. These padding zeros do not affect the generation of the BCH code as described below.

For the (82,61) BCH code, a generator polynomial g(X) (the same as for (127,106) BCH code) is defined as follows:

g(X) = LCM (m1 (X) , m3 (X) , m5 (X))

where LCM = Least Common Multiple.

In the above case:

m1 (X) = X7 + X3 + 1

m3 (X) = X7 + X3 + X2 + X + 1

m5 (X) = X7 + X4 + X3 + X2 + 1

from which,

g(X) = m1 (X) m3 (X) m5 (X)

g(X) = X21 + X18 + X17 + X15 + X14 + X12 + X11+ X8 + X7 + X6 + X5 + X + 1

a determination of g(X) results in the following 22-bit binary number:

g(X) = 1001101101100111100011

To generate the BCH code, an information polynomial, m(x) is formed from the 61 data bits as follows:

m(X) = b1 X 60 + b 2 X 59 + .... + b60 X + b61

where b1 is the first bit (i.e. format flag), and b61 is the last bit of PDF-1. B-2 C/S T.001 – Issue 4 – Rev. 2 February 2018

m (X) is then extended to 82 bits by filling the least significant bits with 21 "0". The resulting 82-bit binary string is then divided by g(X) and the remainder, r(X), becomes the BCH code (the quotient portion of the result of the module-2 binary division is discarded).

SAMPLE 12-BIT BCH CODE CALCULATION :

The BCH error correcting code (bits 133-144) used in the second protected field of the long message is capable of detecting and correcting up to two bit errors in the bits 107-144. The generator polynomial used as a basis for this code is:

g(x) = (1 + x + x6) (1 + x + x2 + x4+ x6)

g(x) = (1 + x3 + x4 + x5 + x8 + x10 + x12)

#### 7.2.4 BIPHASE-L ENCODING

The Data Encoding and the modulation sense should be as follows:

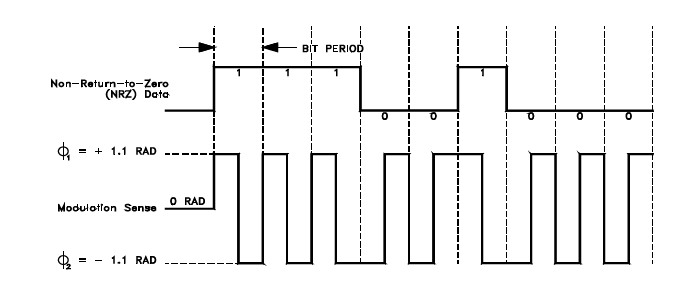
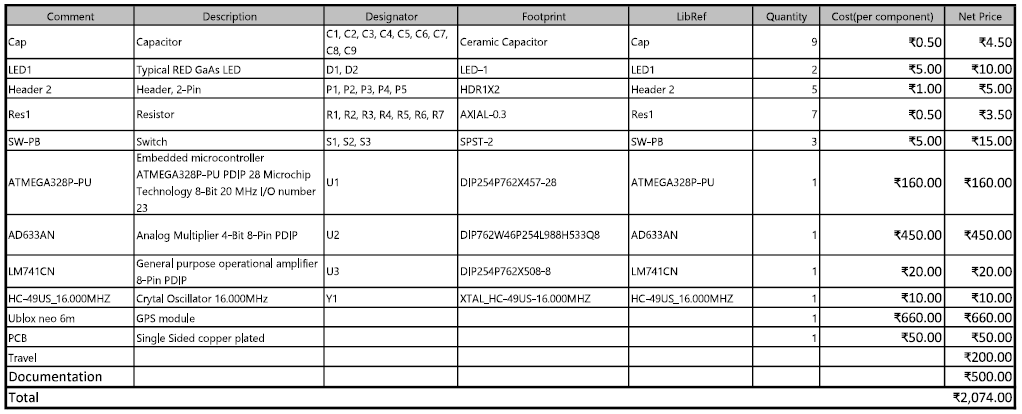


Figure 7.7: Data Encoding and Modulation Sense

# CHAPTER 8

## BUDGET

The Budget of this project id presented below:



# CHAPTER 9

## RESULTS

As per the regulations specified in COSPAS SARSAT Hand Book, the controller coding is done.

For the Testing the EPIRB, the test switch has to be activated, which results in code control entering the test message generation module. The frame synchronisation is added to the message and id BCH encoded and transmitted to Biphase generating circuit after adding a offset delay

On the activation of distress switch, the GPS coordinates are extracted into the specified format. The location data is added to the message and is BCH encoded. The final message is converted to software Biphase and with the addition of offset it is transmitted to the Biphase circuit.

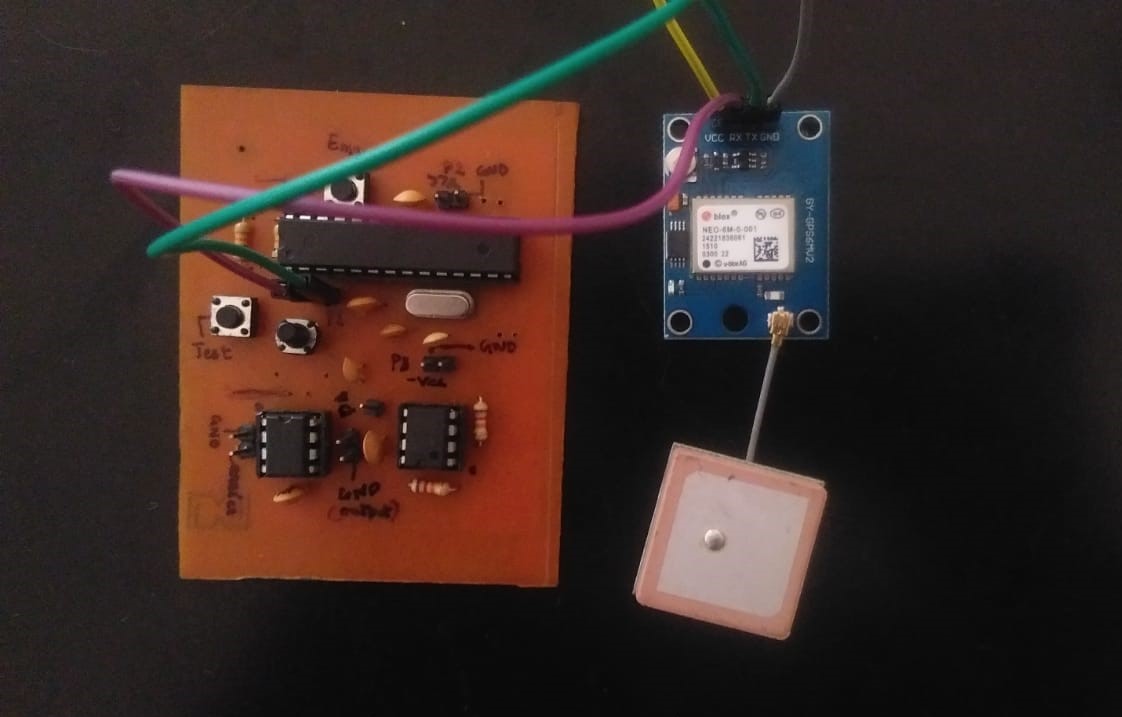


Figure 9.1: Hardware

The circuit is working till the following ranges:

* frequency range (carrier) : up to 10kHz
* supply voltage : -5V and +5V
* controller output : 0V and 5V
* Biphae L signal (LM741) : -5V and +5V

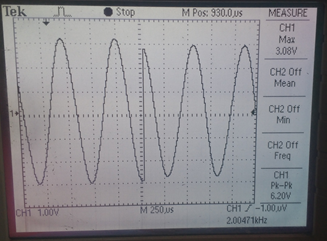


Figure 9.2: Modulator Output

UBLOX NEO 6M GPS coordinates are verified with the U-CENTER software.The obtained result is as follows:

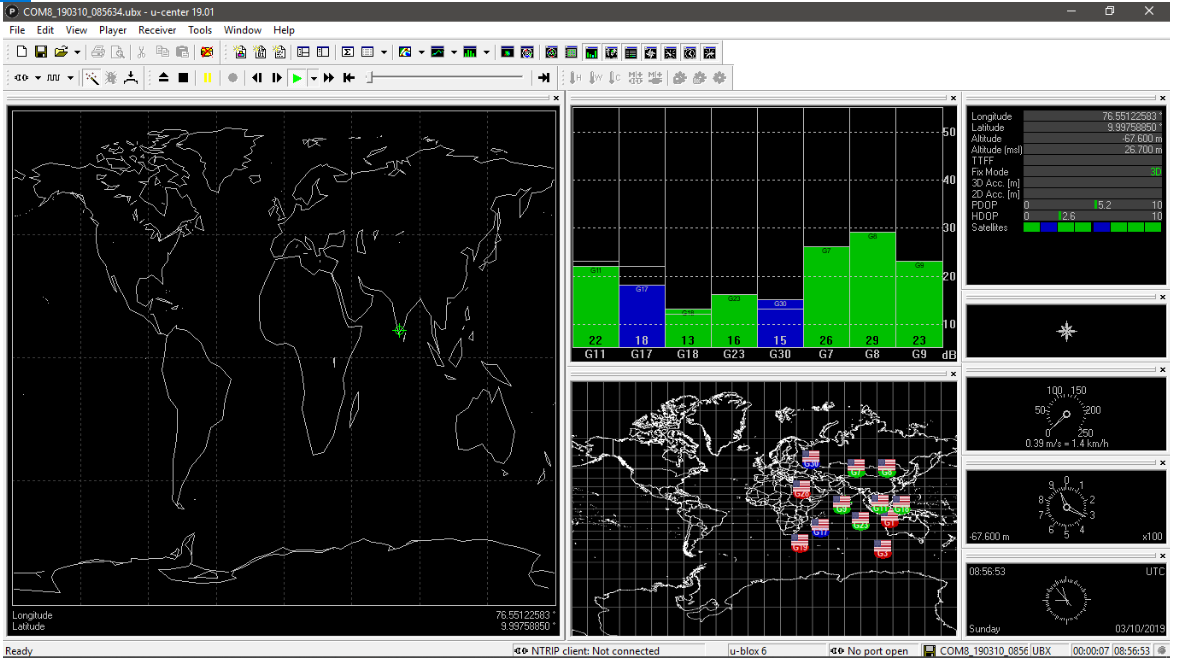


Figure 9.3: U-CENTER output

# CHAPTER 10

## CONCLUSION

The low cost transponder of COSPAS-SARSAT will serve as a locally developed transponder in low cost. As this transponder will use the GPS for positioning data, a boat in distress can easily send its position to monitoring authorities on a single push of a button. Also for medium distance transmission, when HF will be selected, the coverage range will be a broad range and if multiple MCCs are available or MCCs with Aux-MCCs available, all will get almost the same data, hence they can double check for false alarm. Also since signal may be received by more than one MCC, error in data can be easily detected, as in this case at least one MCC will have always better chance to receive stronger signal than rest others. When each transponder

will have their own ID, maintaining a database and registration process will make identification of the boat more user-friendly. In a formatted distress message, for 3rd party listeners like merchant vessels, amateur radio operators or other disciplines,

it will be also easy to read the distress and relay the message to concerning authority.

# REFERENCES

[1]. SPECIFICATION FOR COSPAS-SARSAT 406 MHz DISTRESS BEACONS

*C/S T.001 Issue 4 Revision 2 February 2018*

[2]. HANDBOOK OF BEACON REGULATIONS

*C/S S.007 Issue 2 February 2018*

[3]. HANDBOOK OF BEACON REGULATIONS IND - Section

C/S S.007 Issue 2 February 2018

[4]. UBLOX NEO 6M The Wikipedia Website

Available: http://en.wikipedia.org/wiki/ublox-neo6M

[5]. AD633,IC The Wikipedia Website

Available: http://en.wikipedia.org/wiki/Ad633,analog devices

[6]. LM741 The Wikipedia Website

Available: http://en.wikipedia.org/wiki/texas,lm741

[7]. International COSPAS-SARSAT Programme. The Wikipedia Website

Available: <https://en.wikipedia.org/wiki/International> Cospas-Sarsat Programme

[8]. Global Maritime Distress and Safety System, (2014) The Wikipedia Website

Available: https://en.wikipedia.org/wiki/Global Maritime

Distress and Safety System

[9]. International Distress Frequency, The Wikipedia Website.

Available: http://en.wikipedia.org/wiki/International distress frequency

[10]. GPS nmea information.

Available: <http://www.gpsinformation.org/dale/nmea.html>

[11].NMEA 0183, The Wikipedia Website.

Available: http://en.wikipedia.org/wiki/NMEA 0183

# 

# Appendices

Appendix A

## CONTROLLER CODE

*////////////////////////////////////////////////////////////////////////*

*//////////////////////////////// EPIRB CODE ///////////////////////////*

*///////////////////////////////////////////////////////////////////////*

*#include<SoftwareSerial.h>*

*SoftwareSerial GPSModule( 0 ,1 ); // RX,TX // 10,11 initial*

*/////////////////////////////////// I/O pins*

*const byte testBotton = A1; // 24*

*const byte emergencyBotton = A0 ; // 23*

*const byte testLed = 8 ;*

*const byte emgLed = 9 ;*

*const byte TX\_OUT = 5 ;*

*//////////////////////////// variables for BCH ////////////////////*

*const int messageLength = 144 ;*

*const int biphaseMessageLength = 288 ;*

*bool transmitMessage[ messageLength ]; // transmit buffer*

*bool biphaseMessage[ biphaseMessageLength ];*

*uint8\_t msgLength\_1 = 82 ;*

*uint8\_t genLength\_1 = 22 ;*

*uint8\_t emgLength\_1 = 61 ;*

*uint8\_t msg\_offset\_1 = 24 ;*

*uint8\_t msgLength\_2 = 38 ;*

*uint8\_t genLength\_2 = 13 ;*

*uint8\_t emgLength\_2 = 26 ;*

*uint8\_t msg\_offset\_2 = 106 ;*

*int ran ;*

*uint8\_t Value\_1 ;*

*uint8\_t Value\_2 ;*

*////////////////////////// variables for gps //////////////////////*

*uint8\_t lat\_degree;*

*uint8\_t lat\_minute;*

*uint8\_t lat\_sec;*

*uint16\_t long\_degree;*

*uint8\_t long\_minute;*

*uint8\_t long\_sec;*

*bool north\_pole = false;*

*bool east\_pole = false;*

*uint8\_t offsetMinute;*

*uint8\_t offsetSec;*

*bool sign;*

*uint8\_t value ;*

*uint8\_t stringStart = 0;*

*String nmea[15];*

*/////////////////////////////////////////////////////////////////*

*////////////////////////////// void setup //////////////////////*

*/////////////////////////////////////////////////////////////////*

*void setup(){*

*pinMode( TX\_OUT , OUTPUT);*

*pinMode( testLed , OUTPUT);*

*pinMode( emgLed , OUTPUT);*

*pinMode ( emergencyBotton , INPUT ) ;*

*pinMode ( testBotton , INPUT ) ;*

*//attachInterrupt( digitalPinToInterrupt( emergencyBotton ) , emgMessage , LOW );*

*//attachInterrupt( digitalPinToInterrupt( testBotton ) , testMessage , LOW );*

*Serial.begin(9600);*

*}*

*///////////////////////////////////////////////////////////////////*

*////////////////////////////// void loop /////////////////////////*

*///////////////////////////////////////////////////////////////////*

*void loop(){*

*digitalWrite( emgLed , LOW );*

*digitalWrite( testLed , LOW );*

*gpsExtract( ) ;*

*message\_assign( transmitMessage );*

*converttoLat( nmea , &north\_pole , &lat\_degree , &lat\_minute , &lat\_sec );*

*converttoLon( nmea , &east\_pole , &long\_degree , &long\_minute , &long\_sec );*

*Value\_1 = digitalRead(emergencyBotton);*

*//Serial.println(Value\_1);*

*if( Value\_1 == 0){*

*emgMessage ();*

*}*

*Value\_2 = digitalRead(testBotton);*

*//Serial.println(Value\_2);*

*if ( Value\_2 == 0){*

*testMessage();*

*}*

*}*

*////////////////////////////////////////////////////////////////////////////*

*////////////////////////// function definitions /////////////////////*

*///////////////////////////////////////////////////////////////////////////*

*void emgMessage (){ ///////// interrupt 0*

*//unsigned long start = micros(); //// check the time*

*digitalWrite( emgLed , HIGH );*

*//Serial.println("emergency");*

*//gpsExtract( ) ;*

*//message\_assign( transmitMessage );*

*//converttoLat( nmea , &north\_pole , &lat\_degree , &lat\_minute , &lat\_sec );*

*//converttoLon( nmea , &east\_pole , &long\_degree , &long\_minute , &long\_sec );*

*roundCoordinate( lat\_minute , lat\_sec , &offsetMinute , &offsetSec , &sign);*

*offtoBitsLat();*

*roundCoordinate( long\_minute , long\_sec , &offsetMinute , &offsetSec , &sign);*

*offtoBitsLong();*

*bch\_gen( transmitMessage , msgLength\_1 , genLength\_1 , emgLength\_1 , msg\_offset\_1 );*

*bch\_gen( transmitMessage , msgLength\_2 , genLength\_2 , emgLength\_2 , msg\_offset\_2 );*

*biphaseMessageGeneration();*

*//unsigned long end = micros();*

*//unsigned long delta = end - start;*

*//Serial.println(delta);*

*delay( 158 );*

*delayMicroseconds(540);*

*delayMicroseconds(175); /// 1kHz --- 175usec /// 2kHz --- 87.5usec /// 5kHz --- 17.5usec*

*while(1)*

*{*

*digitalWrite( emgLed , HIGH );*

*ran = random( 0 , 6 );*

*delay( (ran\*10) );*

*delay( 160 );*

*for (int i = 0; i < biphaseMessageLength ; ++i)*

*{*

*//Serial.print( biphaseMessage[ i ] );*

*digitalWrite( TX\_OUT , biphaseMessage[ i ] );*

*delayMicroseconds(5000);*

*}*

*digitalWrite( emgLed , LOW );*

*}*

*}*

*void testMessage(){ ///////// interrupt 1*

*//unsigned long start = micros(); //// check the time*

*digitalWrite( testLed , HIGH );*

*//Serial.println(" test ");*

*message\_assign( transmitMessage );*

*testMessageAssignment( transmitMessage );*

*bch\_gen( transmitMessage , msgLength\_1 , genLength\_1 , emgLength\_1 , msg\_offset\_1 );*

*bch\_gen( transmitMessage , msgLength\_2 , genLength\_2 , emgLength\_2 , msg\_offset\_2 );*

*biphaseMessageGeneration();*

*//unsigned long end = micros();*

*//unsigned long delta = end - start;*

*//Serial.println(delta);*

*delay( 158 );*

*delayMicroseconds(564);*

*delayMicroseconds(175); /// 1kHz --- 175 usec /// 2kHz --- 87.5 usec /// 5kHz --- 17.5 usec*

*while(1){*

*for (int i = 0; i < biphaseMessageLength ; ++i)*

*{*

*//Serial.print( biphaseMessage[ i ] );*

*digitalWrite( TX\_OUT , biphaseMessage[i] );*

*delayMicroseconds(5000);*

*}*

*//digitalWrite( testLed , LOW );*

*}*

*//while(1);*

*}*

*////////////////////////////////////////////////////////////////////////*

*/////////////////////////// GPS extraction ///////////////////////////*

*////////////////////////////////////////////////////////////////////////*

*void gpsExtract( ){*

*uint8\_t index\_nmea = 0;*

*//Serial.println(" gps extract ");*

*Serial.flush();*

*GPSModule.flush();*

*while (GPSModule.available() > 0)*

*{*

*GPSModule.read();*

*}*

*if (GPSModule.find("$GPRMC,"))*

*{*

*String tempMsg = GPSModule.readStringUntil('\n');*

*for (int i = 0; i < tempMsg.length(); i++)*

*{*

*if (tempMsg.substring(i, i + 1) == ",")*

*{*

*nmea[index\_nmea] = tempMsg.substring(stringStart, i);*

*stringStart = i + 1;*

*index\_nmea++;*

*}*

*if (i == tempMsg.length() - 1)*

*{*

*nmea[index\_nmea] = tempMsg.substring(stringStart, i);*

*}*

*}*

*}*

*}*

*//////////////////////////////////////////////////////////////////////////*

*////////////////////////////// extracting latitude ///////////////////////*

*///////////////////////////////////////////////////////////////////////////*

*void converttoLat( String \* nmea , bool\* north\_pole , uint8\_t\* lat\_degree , uint8\_t\* lat\_minute , uint8\_t\* lat\_sec ){*

*//Serial.println(" convert to lat ");*

*//Serial.println(" converttoLat ");*

*if (nmea[3] == "N")*

*{*

*\*north\_pole = false;*

*}*

*else*

*{*

*\*north\_pole = true;*

*}*

*for (int i = 0 ; i < nmea[2].length(); i++ )*

*{*

*if (nmea[2].substring(i, i + 1) == ".")*

*{*

*\*lat\_degree = nmea[2].substring(0, i - 2).toInt();*

*\*lat\_minute = nmea[2].substring(i - 2,i - 1).toInt();*

*\*lat\_sec =nmea[2].substring(i+1).toInt();*

*}*

*}*

*}*

*////////////////////////////////////////////////////////////////////////////*

*///////////////////////////// extracting longitude /////////////////////////*

*///////////////////////////////////////////////////////////////////////////*

*void converttoLon( String \* nmea , bool\* east\_pole , uint16\_t\* long\_degree , uint8\_t\* long\_minute , uint8\_t\* long\_sec ){*

*//Serial.println(" convert to long ");*

*//Serial.println(" converttoLon ");*

*if (nmea[5] == "E")*

*{*

*\*east\_pole = false;*

*}*

*else*

*{*

*\*east\_pole = true;*

*}*

*for (int i = 0; i < nmea[4].length(); i++)*

*{*

*if (nmea[2].substring(i, i + 1) == ".")*

*{*

*\*long\_degree = nmea[4].substring(0, i - 2).toInt();*

*\*long\_minute = nmea[4].substring(i - 2,i - 1).toInt();*

*\*long\_sec =nmea[4].substring(i+1).toInt();*

*}*

*}*

*}*

*////////////////////////////////////////////////////////////////////////////*

*////////////////////////// round coordinates /////////////////////////////*

*///////////////////////////////////////////////////////////////////////////*

*void roundCoordinate( uint8\_t valCoordinate , uint8\_t secCoordinate , uint8\_t\* offsetMinute , uint8\_t\* offsetSec , bool\* sign){*

*//Serial.println(" roundCoordinate ");*

*if ( ( valCoordinate >= 7 ) && ( valCoordinate < 23 ) )*

*{*

*value = 15 ;*

*if ( ( value - valCoordinate ) >= 0)*

*{*

*\*sign = false ;*

*\*offsetMinute = value - valCoordinate ;*

*}*

*else*

*{*

*\*sign = true;*

*\*offsetMinute = valCoordinate - value;*

*}*

*}*

*else*

*{*

*if ( ( valCoordinate >= 23 ) && ( valCoordinate < 37 ) )*

*{*

*value = 30 ;*

*if ( ( value - valCoordinate ) >= 0)*

*{*

*\*sign = false ;*

*\*offsetMinute = value - valCoordinate ;*

*}*

*else*

*{*

*\*sign = true;*

*\*offsetMinute = valCoordinate - value;*

*}*

*}*

*else*

*{*

*if ( ( valCoordinate >= 37 ) && ( valCoordinate < 52 ) )*

*{*

*value= 45 ;*

*if ( ( value - valCoordinate ) >= 0)*

*{*

*\*sign = false ;*

*\*offsetMinute = value - valCoordinate ;*

*}*

*else*

*{*

*\*sign = true ;*

*\*offsetMinute = valCoordinate - value;*

*}*

*}*

*else*

*{*

*value = 0 ;*

*if ( valCoordinate < 7)*

*{*

*\*sign = false ;*

*\*offsetMinute = valCoordinate ;*

*}*

*else*

*{*

*\*sign = true ;*

*\*offsetMinute = 60 - valCoordinate ;*

*}*

*}*

*}*

*}*

*secCoordinate = ( secCoordinate / 4 ) \* 4 ;*

*if ( \*sign )*

*{*

*offsetSec = secCoordinate;*

*}*

*else*

*{*

*\*offsetSec = 60 - secCoordinate;*

*\*offsetMinute = \*offsetMinute - 1 ;*

*}*

*}*

*///////////////////////////////////////////////////////////////////////////////////*

*///////////////////////////////// bch message generation /////////////////////////*

*///////////////////////////////////////////////////////////////////////////////////*

*void bch\_gen( bool \* emg\_msg , uint8\_t msgLength, uint8\_t genLength , uint8\_t emgLength , uint8\_t msg\_offset ){*

*//Serial.println(" bch\_gen ");*

*bool msgPolynomial [ msgLength ];*

*bool genPolynomial [ genLength ];*

*for( int k = 0; k < emgLength ; k++ )*

*{*

*msgPolynomial[ k ] = emg\_msg[ msg\_offset + k ];*

*}*

*for( int k = emgLength ; k < msgLength ; k++)*

*{*

*msgPolynomial[k]= false ;*

*}*

*if( genLength == genLength\_1 )*

*{*

*bch21\_Generator( genPolynomial);*

*}*

*else*

*{*

*bch12\_Generator( genPolynomial);*

*}*

*for (int i = 0; i < (msgLength - genLength +1) ; i++ )*

*{*

*if ( msgPolynomial[ i ] == true )*

*{*

*int temp = i;*

*for (int k = temp ; k < ( genLength + temp ) ; k++ )*

*{*

*msgPolynomial[ k ] = ( ( msgPolynomial[ k ] ) ^ ( genPolynomial[ k - temp ] ) );*

*}*

*}*

*}*

*for( int k = 0; k < genLength ; k++ )*

*{*

*emg\_msg[ k + emgLength + msg\_offset ] = msgPolynomial[ k + emgLength - 1 ];*

*}*

*}*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*/////////////////////////////////////////////////// generator polynomial for bch 1 //////////////////////////*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*void bch21\_Generator(bool\* genPolynomial){*

*genPolynomial[ 0 ] = true ;*

*genPolynomial[ 1 ] = false ;*

*genPolynomial[ 2 ] = false ;*

*genPolynomial[ 3 ] = true ;*

*genPolynomial[ 4 ] = true ;*

*genPolynomial[ 5 ] = false ;*

*genPolynomial[ 6 ] = true ;*

*genPolynomial[ 7 ] = true ;*

*genPolynomial[ 8 ] = false ;*

*genPolynomial[ 9 ] = true ;*

*genPolynomial[ 10 ] = true ;*

*genPolynomial[ 11 ] = false ;*

*genPolynomial[ 12 ] = false ;*

*genPolynomial[ 13 ] = true ;*

*genPolynomial[ 14 ] = true ;*

*genPolynomial[ 15 ] = true ;*

*genPolynomial[ 16 ] = true ;*

*genPolynomial[ 17 ] = false ;*

*genPolynomial[ 18 ] = false ;*

*genPolynomial[ 19 ] = false ;*

*genPolynomial[ 20 ] = true ;*

*genPolynomial[ 21 ] = true ;*

*}*

*///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*///////////////////////////////////////////////////////// generator for bch 2 ////////////////////////////////////////*

*///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*void bch12\_Generator(bool\* genPolynomial){*

*genPolynomial[ 0 ] = true ;*

*genPolynomial[ 1 ] = false ;*

*genPolynomial[ 2 ] = true ;*

*genPolynomial[ 3 ] = false ;*

*genPolynomial[ 4 ] = false ;*

*genPolynomial[ 5 ] = true ;*

*genPolynomial[ 6 ] = true ;*

*genPolynomial[ 7 ] = false ;*

*genPolynomial[ 8 ] = true ;*

*genPolynomial[ 9 ] = true ;*

*genPolynomial[ 10 ] = false ;*

*genPolynomial[ 11 ] = false ;*

*genPolynomial[ 11 ] = false ;*

*}*

*/////////////////////////////////////////////////////////////////////////////////////////////////*

*///////////////////////////////// standard location protocol with mmsi /////////////////////*

*/////////////////////////////////////////////////////////////////////////////////////////////////*

*void message\_assign(bool\* message ){*

*int i; ///////////////////////////////////////////// bit synchronisation 0 - 14 - all 1s*

*for(i=0;i<15;i++)*

*{*

*message[i]= true;*

*}*

*message[15]= false; //////////////////////////////////////////// frame synchro 15- 23*

*message[16]= false;*

*message[17]= false;*

*message[18]= true;*

*message[19]= false;*

*message[20]= true;*

*message[21]= true;*

*message[22]= true;*

*message[23]= true; //*

*message[24]= true; // long message //////////////////////////////////////////// std location with MMSI*

*message[25]= false; // location protocol //*

*message[26]= false; /////////////////////////////////////////// country code india 419*

*message[27]= true;*

*message[28]= true;*

*message[29]= false;*

*message[30]= true;*

*message[31]= false;*

*message[32]= false;*

*message[33]= false;*

*message[34]= true;*

*message[35]= true; //*

*message[36]= false; /////////////////////////////////////////////// std protocol type ; EPIRB-MMSI 0110*

*message[37]= true;*

*message[38]= true;*

*message[39]= false; //*

*for(i=40;i<49;i++) //////////////////////////////////////////// 10 bit cospass sarsat approva certificate no*

*{*

*message[i]= false;*

*}*

*for(i=50;i<63;i++) //////////////////////////////////////////////// 14 bit serial number*

*{*

*message[i]= false;*

*}*

*message[64]= false; //////////////////////////////////////////////// latitude flag (N/S : 0/1)*

*for(i=65;i<72;i++) /////////////////////////////////////////////// latitude in degree (0 - 90 with .25 increment)*

*{*

*message[i]= false;*

*}*

*message[72]= true; ///////////////////////////////////////////// lattitude in 1/4 increment*

*message[73]= true; //*

*message[74]= false; /////////////////////////////////////////////// logitude flag (E/W : 0/1)*

*for(i=75;i<83;i++) ////////////////////////////////////////////// logitude in degree (0 - 180 with .25 increment)*

*{*

*message[i]= false;*

*}*

*message[83]= true; ///////////////////////////////////////////// logitude in 1/4 increment*

*message[84]= true;*

*for(i=85;i<105;i++) ///////////////////////////////////////////// BCH\_1 - 21 bit*

*{*

*message[i]= false;*

*}*

*message[106]= true; //////////////////////////////////////////// fixed bits (1101)*

*message[107]= true;*

*message[108]= false;*

*message[109]= true;*

*message[110]= false; /////////////////////////////////////////// positional data provided by external device (yes/no) ; (0/1)*

*message[111]= false; /////////////////////////////////////////// auxiliary device :121.5 m hz (y/n) : (1/0)*

*message[112]= false; /////////////////////////////////////////// latitue offset sign (-/+) : (0/1)*

*message[113]= false; /////////////////////////////////////////// latitude off set in min :(0-30 in 1 min increment)*

*message[114]= false;*

*message[115]= false;*

*message[116]= false;*

*message[117]= true;*

*message[118]= false; /////////////////////////////////////////// latitude off set in sec : (0-30 in 1 min increment)*

*message[119]= false;*

*message[120]= false;*

*message[121]= true;*

*message[122]= false; /////////////////////////////////////////// longitude offset sign (-/+) : (0/1)*

*message[123]= true; /////////////////////////////////////////// longitude off set in min : (0-30 in 1 min increment)*

*message[124]= false;*

*message[125]= false;*

*message[126]= false;*

*message[127]= false;*

*message[128]= false; /////////////////////////////////////////// longitude off set in sec : (0-30 in 1 min increment)*

*message[129]= false;*

*message[130]= true;*

*message[131]= false;*

*for(i=132;i<144;i++) ////////////////////////////////////////////// BCH\_2 - 12 bit*

*{*

*message[i]= false;*

*}*

*}*

*////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*//////////////////////////////////////// selftest Message /////////////////////////////////////////////////////////////*

*///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*void testMessageAssignment(bool\* message ){*

*message[15]= 0; //////////////////////////////////////////// frame synchro 15- 23*

*message[16]= 1;*

*message[17]= 1;*

*message[18]= 0;*

*message[19]= 1;*

*message[20]= 0;*

*message[21]= 0;*

*message[22]= 0;*

*message[23]= 0;*

*}*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*////////////////////////////////////////// offset to bits in latitude //////////////////////////////////////////////*

*////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*void offtoBitsLat(){*

*transmitMessage[ 64 ] = north\_pole ;*

*for (int i = 0; i < 8; i++)*

*{*

*transmitMessage[ 65 + i ] = bitRead(lat\_degree,i);*

*}*

*if ( value == 0)*

*{*

*transmitMessage[ 72 ] = 0 ;*

*transmitMessage[ 73 ] = 0 ;*

*}*

*else*

*{*

*if ( value == 15)*

*{*

*transmitMessage[ 72 ] = 0 ;*

*transmitMessage[ 73 ] = 1 ;*

*}*

*else*

*{*

*if ( value == 30)*

*{*

*transmitMessage[ 72 ] = 1 ;*

*transmitMessage[ 73 ] = 0 ;*

*}*

*else*

*{*

*transmitMessage[ 72 ] = 1 ;*

*transmitMessage[ 73 ] = 1 ;*

*}*

*}*

*}*

*transmitMessage[ 112 ] = sign ;*

*for (int i = 0; i < 4; i++)*

*{*

*transmitMessage[ 113 + i ] = bitRead( offsetMinute , i );*

*}*

*offsetSec = offsetSec / 4 ;*

*for (int i = 0; i < 3; i++)*

*{*

*transmitMessage[ 118 + i ] = bitRead( offsetSec , i );*

*}*

*}*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*////////////////////////////////////////// offset to bits in longitude //////////////////////////////////////////////*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*void offtoBitsLong(){*

*transmitMessage[ 74 ] = east\_pole ;*

*for (int i = 0; i < 9; i++)*

*{*

*transmitMessage[ 75 + i ] = bitRead(long\_degree,i);*

*}*

*if ( value == 0)*

*{*

*transmitMessage[ 83 ] = 0 ;*

*transmitMessage[ 84 ] = 0 ;*

*}*

*else*

*{*

*if ( value == 15)*

*{*

*transmitMessage[ 83 ] = 0 ;*

*transmitMessage[ 84 ] = 1 ;*

*}*

*else*

*{*

*if ( value == 30)*

*{*

*transmitMessage[ 83 ] = 1 ;*

*transmitMessage[ 84 ] = 0 ;*

*}*

*else*

*{*

*transmitMessage[ 83 ] = 1 ;*

*transmitMessage[ 84 ] = 1 ;*

*}*

*}*

*}*

*transmitMessage[ 122 ] = sign ;*

*for (int i = 0; i < 4; i++)*

*{*

*transmitMessage[ 123 + i ] = bitRead( offsetMinute , i );*

*}*

*offsetSec = offsetSec / 4 ;*

*for (int i = 0; i < 3; i++)*

*{*

*transmitMessage[ 128 + i ] = bitRead( offsetSec , i );*

*}*

*}*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*//////////////////////////////////////////////// Biphase ////////////////////////////////////////////////////////*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*void biphaseMessageGeneration(){*

*for (int i = 0; i < messageLength ; ++i)*

*{*

*biphaseMessage [ (2\*i) ] = transmitMessage[ i ];*

*biphaseMessage [ (2\*i) + 1 ] = ~ transmitMessage[ i ];*

*}*

*}*

*/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*

*/////////////////////////////////////////////////// END ////////////////////////////////////////////////////////////*

*////////////////////////////////////////////////////////////////////////////////////////////////////////////////////*