AN AUTONOMOUS ALARM SYSTEM FOR PREVENTING ACCIDENTS DUE TO OVERLOADING AND INSTABILITY OF A MARINE VESSEL

by

Muhibbur Rahman

POST GRADUATE DIPLOMA IN INFORMATION AND COMMUNICATION TECHNOLOGY

Institute of Information and Communication Technology
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

December 2018

The project titled “AN AUTONOMOUS ALARM SYSTEM FOR PREVENTING ACCIDENTS DUE TO OVERLOADING AND INSTABILITY OF A MARINE VESSEL” submitted by Muhibbur Rahman, Roll No.: 0417311001, Session: April/2017 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Post Graduate Diploma in Information and Communication Technology on 12 December, 2018.

BOARD OF EXAMINERS

1.	Dr. S. M. Lutful kabir Professor Institute of Information and Communication Technology BUET, Dhaka-1000, Bangladesh.	Chairman (Supervisor)
2.	Dr. Md. Liakot Ali Professor Institute of Information and Communication Technology BUET, Dhaka-1000, Bangladesh.	Member
3.	Dr. Mohammad Arifuzzaman Lecturer Institute of Information and Communication Technology BUET, Dhaka-1000, Bangladesh.	Member

CANDIDATE’S DECLARATION

It is hereby declared that this report or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Muhibbur Rahman
ID: 0417311001

Dedicated

To

My Parents and the Respected Teachers

Who Inspired Me All through My Life.

Table of Contents

List of Tables and Figures

List of Abbreviations of Technical Symbols and Terms

CG	Center of gravity
MCU	Microcontroller Unit
KG	Height of the ship's Center of Gravity the above keel
KM	The height of the initial metacenter M above the base plane.
KB	The height of the center of buoyancy above the base plane.
BM	Metacentric radius
GM	Metacentric height.

Acknowledgement

At first I would like to convey my gratitude to Almighty Allah for giving me the opportunity to accomplish this project. I would like to thank my supervisor Dr. S. M. Lutful Kabir, Professor, Institute of Information and Communication Technology (IICT), Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. He motivated me to take upon this interesting and useful project, which has an immense importance in our country. He has provided all sorts of support required for the project work. Without his proper guidance, advice, continual encouragement and active supervision in this process of this work, it would have not been possible to complete.

Special thanks goes to Mr. Rashedul Hasan, Research Engineer, IICT, BUET for his support in the project. I am indebted to all the teachers, officers and staffs of Information and Communication Technology (IICT) for giving me their kind support and information during the study. The lab facility in the "Embedded System Laboratory" was very helpful for carrying out this research.

I am also grateful to my parents whose continuous support all over my life has brought me this far in my career.

Finally, I tender again special thanks to the Almighty that I have been successful in my effort to complete the study.

Abstract

Passenger vessels in Bangladesh are a vital part of the inland waterways transports. A large number of travelers use these waterways regularly. Southern part of the country is connected to the capital mostly by water transports. Unfortunately, frequency of accidents in these waterways is also high. Rate of accidents increases during rainy season and the time of different festivals. The reason behind these accidents can be contributed to mainly faulty vessels, overloading and instability caused by the capsizing couple.

A vessel can be at three state of stability- positive stability, neutral stability and negative stability. When the CG of a vessel is below the metacenter a righting moment is formed. This righting moment keeps the vessel in positive stability condition. But when CG shifts vertically upward and move beyond metacenter, a capsizing couple is generated. This causes the vessel attain negative stability. As a consequence, if a vessel heels at some angle this capsizing couple will make that angle bigger and eventually capsizing the vessel.

This project deals with the real time tracking of the loading condition and movement of CG. To find out the CG in real time we investigated the possibility of mounting the load cells at important locations where the load tends to shift more. And after receiving the data from load cells placed in those locations, an estimate of total load and position of CG is suggested. If the total load exceeds the permissible limit, the alarm warms the vessel masters. And if the position of CG moves upward beyond metacenter another alarm will go off.

CHAPTER 1

Introduction

1.1 Introduction

1.1.1 Background

Safety is the primary concern for any type of marine vehicle. This concern in a passenger vessel is very high as it is designed for transporting people [1], [2]. Bangladesh is a riverine country. Consequently, a large number of people use inland waterways for transportation. But the number of passenger vessels is proportionately low. This results in the overloading of the passengers beyond the danger limit [3]. This trend is worsened during the peak time of a festival. It accounts for a large number of deaths per year in the inland waterways. During the year 2005-2015, nearly eighteen hundred people have died, injured and found missing due to inland waterways accidents in Bangladesh [4]. One of the key factors causing these accidents is the shifting of the center of gravity of the vessel [5], [6]. In overloaded vessel, a large number of passengers occupy the portion above the top deck. That causes the center of gravity (CG) of the vessel to shift upward.

Figure 1.1 Overloaded passenger vessel in Bangladesh waterways.

Generally, a stable equilibrium occurs when CG is below the metacenter [7]. In this case, a righting couple is formed which corrects the vessels inclining every time it occurs. So, the vessel remains stable and no capsizing occurs.
If the CG moves upward beyond the metacenter, the vessel becomes unstable. Then if the ship heels to one side there will be no righting couple to make vessel correct its heeling. Eventually, the vessel will capsize.

1.1.2 Literature Review

Some notable works have been performed to solve the issue. Zheng et al., [8] used level sensor for ship stability analysis and monitoring. They proposed an approach to measure the liquid level with accuracy. The vessel draft will be measured with this level sensor and stability will be estimated. Pennanen et al., [9] worked on integrated decision support system for the safety of passenger ship. Their work mainly discussed the decision support system of a passenger vessel during flooding emergency. They suggested integration of tank level data, flood level data, door status data and navigational data to develop the safety system. Surendran and Reddy [10] focused their work to numerically predict the ship stability for different waves. They undertook parametric investigation to identify the effect of number of key parameters like wave amplitude, wave frequency, metacentric height etc. Among the six motions of a ship, rolling motion generally causes the capsizing. Their study mainly focused the performance of a ship in a beam sea which causes rolling motion to a vessel.
Kreuzer and Wendt [11] did analysis on ship capsizing in following and quartering sea. They analyzed the stability of ships using advanced mathematical modeling and method. They showed that a ship’s dynamics strongly depends on the nonlinearities of the ship-fluid system. They also identified the critical wave heights to capsize a ship which is different in following and quartering sea. In quartering sea required wave height was found to be much lower than in following sea. Fan et al., [12] proposed algorithm for stowage in a large container ship. Their work analyzed the stowage plan (developed by their system) based on critical measurements such as crane intensity and ship stability. This stowage plan is vital for passenger ship also. In order to prevent

the capsizing, the passenger should have certain restriction on where they can settle on board.

1.1.3 Motivation

Generally during loading of a passenger vessel, safety is the utmost concern and overloading with passengers is simply not an option in normal situation. Though some works are found in regard to predicting the vessel's stability in various sea conditions but none of them dealt with the erratic moving of load within the vessel. No work has been found where the instability caused by the capsizing couple has been addressed. The reason for this absence can be attribute to uniqueness of the problem in Bangladesh and some very poor countries. To address this issue, a detection of overload and shifting of CG has to be developed.

1.2 Objectives and Possible Outcome

The objective of this project is to develop a wireless network based autonomous alarm system mounted on the marine vehicle with the following features:

1. To measure the weight at the different portion of the passenger vessels from sensors, each sensor can be considered as a node and the value of load measured by the sensors will be transmitted to a central module using Bluetooth.
2. To calculate total load after receiving data from all the sensors, and if total load is heavier than the permissible limit warn the master and passenger with alarm that the vessel is overloaded.
3. To calculate the shift in center of gravity of the whole vessel after receiving the load data from all the sensors and to trigger the alarm if the center of gravity moves upward beyond the metacenter of the vessel.

Possible outcome with the successful completion of this project will be development of an autonomous alarm system that will make the river journey safer for passengers. And hopefully, it will contribute to the elimination of tragic deaths caused by overloading in the inland waterways of Bangladesh.

1.3 Organization of the Project Report

The rest of the report is organized as follows:

Chapter 1 describes the overview and objectives of the project. Chapter 2 elucidates different important features of a marine vessel. Hardware description is presented in Chapter 3. In Chapter 4, a description is provided for the implementation of the project. Chapter 5 describes project strength, limitations and challenges of the project, conclusion and recommendations for future works of the system.

CHAPTER 2 Vessel Anatomy

2.1 Overview

During construction phase and also during cruising it is very important to have some knowledge about different parts of a vessel and some controlling points. These components and concepts of a vessel are discussed in this chapter.

2.2 Nautical Orientation

Before we discuss about the controlling points we should be familiar with the nautical orientation and directions. The forward most portion of a vessel is called bow and the aft ward portion at the extreme is called the stern. The middle portion between the bow and stern is called midship. If one stands on a vessel keeping the bow in the front, then the left side of the longitudinal centerline is called port and the right side is called starboard. When a vessel inclined at some angle at port or starboard side due to external force, it is called heeling. Figure 2.1 shows the bow, stern, port and starboard of a vessel.

Figure 2.1 Nautical orientation of a vessel.

2.3 Center of Gravity

The center of gravity (CG) of an object is the point at which weight is evenly distributed and all sides are in balance [13]. The coordinates of CG of an object is given by,

$$\bar{x} = \frac{x_1 m_1 + x_2 m_2 + x_3 m_3 + \cdots\cdots + x_n m_n}{m_1 + m_2 + m_3 + \cdots\cdots + m_n}$$

$$\bar{y} = \frac{y_1 m_1 + y_2 m_2 + y_3 m_3 + \cdots\cdots + y_n m_n}{m_1 + m_2 + m_3 + \cdots\cdots + m_n}$$

$$\bar{z} = \frac{z_1 m_1 + z_2 m_2 + z_3 m_3 + \cdots\cdots + z_n m_n}{m_1 + m_2 + m_3 + \cdots\cdots + m_n}$$

Where m_1, m_2, m_3……m_n are the masses of different components of an object. And x_1, x_2, x_3……. x_n are the coordinate of the corresponding components in the X direction.

y_1, y_2, y_3……. y_n are the coordinate of the corresponding components in the Y direction. z_1, z_2, z_3……. z_n are the coordinate of the corresponding components in the Z direction.

Center of Gravity of a ship is the point at which all forces of gravity acting on the ship can be considered to act. "G" is the center of mass of the vessel as in Figure 2.2. The position of "G" is dependent upon the distribution of weights within the ship [14]. As the distribution of weights is altered, the position of "G" will react as follows:

I. "G" moves towards a weight addition.
II. "G" moves away from a weight removal.
III. "G" moves in the same direction as a weight shift.

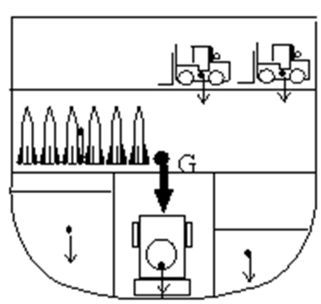


Figure 2.2 Center of gravity of a vessel [14].

2.4 Center of Buoyancy

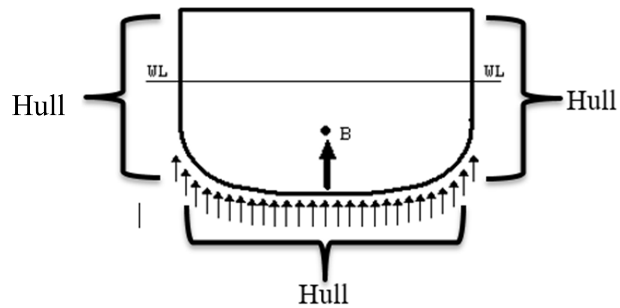


Figure 2.3 Center of buoyancy

Hull is the body of ship including shell plating, framing etc. For the purpose of this project we can consider the hull is the outer structure. The direction of the weight is downward and for a solid structure there is a point through which this weight is exerted, which is called CG. And buoyancy exerts same force equal to the weight of the immersed body. Acting upward the buoyancy also act along a point called center of buoyancy. Object can be wholly or partly immersed in the fluid. The position of center of buoyancy depends on the underwater contour of the immersed part of the object.

Center of Buoyancy, B as in Figure 2.3 is the geometric center of the ship's underwater hull body. It is the point at which all the forces of buoyancy may be considered to act in a vertically upward direction [14].

2.4.1 Lateral Shift in Center of Buoyancy

The Center of Buoyancy will move as the shape of the underwater portion of the hull body changes [14]. When the ship rolls to starboard, "B" moves to starboard, and when the ship rolls to port, "B" moves to port as shown in Figure 2.4

.

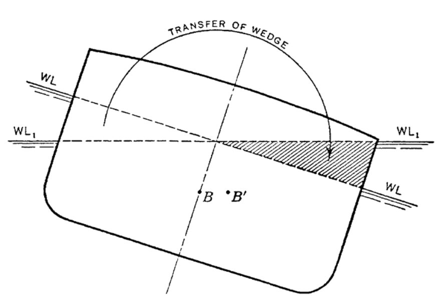


Figure 2.4 Lateral Shift in the position of center of buoyancy [14].

2.4.2 Vertical Shift in Center of Buoyancy

When the ship's hull is made heavier, the drafts increase as the ship sits deeper in the water. "B" will move up. When the ship's hull is lightened, the drafts decrease as the ship sits shallower in the water. "B" will move down [14]. The Center of Buoyancy moves in the same direction as the ship’s waterline as shown in Figure 2.5. The position of the metacenter is a function of the position of the center of buoyancy, thus a function of the displacement of the ship.

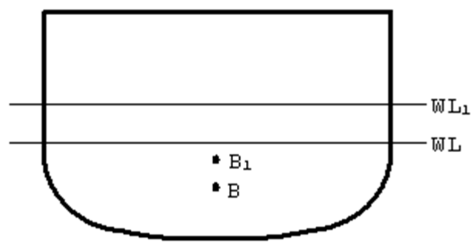


Figure 2.5 Vertical Shift in the position of center of buoyancy [14].

2.5 Metacenter

As shown in Figure 2.6, if a ship heels to any angle the portion of the lower side submerged under water and the portion of the upper side emerges out of the water.

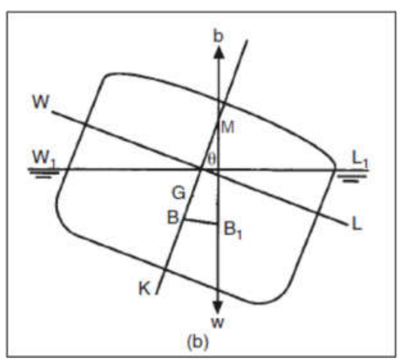


Figure 2.6 Position of stability reference points of a ship [16].

Due to this shift in the submerged volume the center of buoyancy shifts from the centerline to the side that is lowered as result of the heeling. A vertical line is extended through the shifted center of buoyancy, B_1. This line intersects the centerline (line through KG) at the metacenter, M. The position of "M" moves as follows: As the center of buoyancy moves up, the Metacenter moves down. As the Center of Buoyancy moves down, the Metacenter moves up.

2.6 Keel

Keel is situated at the bottom of the ship. The center line passing through from forward to aft of the vessel is the keel. In Figure 2.6, K is the keel. This is reference point from which all other parameter measurements are compared [14].

2.7 Stability Reference Points with the vessel inclined at a small angle

When a ship is floating at rest in still water, and at that time if it is inclined by an external force to a small angle φ, the center of buoyancy shifts from B to the new position – Bφ [15]. The point where the new line of buoyancy force intersects the initial line is denoted as the initial metacenter M shown in Figure 2.7. For practical purposes, in normal passenger vessel the point M does not change in position for inclination up to about 7-15 degrees.

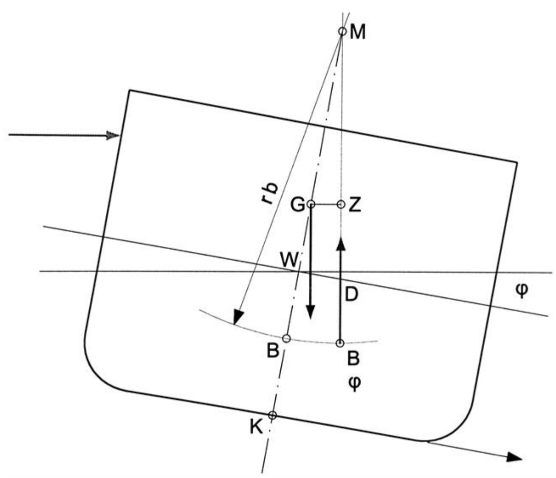


Figure 2.7 Position of CG and metacenter of a vessel [15].

KG - Height of the ship's Center of Gravity the above keel

KM – The height of the initial metacenter M above the base plane. It depends on a geometrical form of ship's submerged part. This measurement is found by using the Draft Diagram and Functions of Form Curves.

KM = KB + BM

KB – The height of the center of buoyancy above the base plane denoted as vertical center of buoyancy (VCB), BM – Metacentric radius

GM – The vertical distance between G and M is referred to as the metacentric height.

GM = KM - KG

2.8 Righting Arm

A ship generally undergoes a sea trial for several weeks just after the construction to calculate and calibrate various systems and machineries on board. There is an experiment called inclining experiment. In this experiment the position of CG at light weight is determined. At this stage the center of gravity of a ship is determined.

When a ship is inclined, the center of buoyancy shifts off centerline while the center of gravity remains in the same location. Since the forces of buoyancy and gravity are equal and act along parallel lines, but in opposite directions, a rotation is developed. This is demonstrated in Figure 2.8. This is called a couple, two moments acting simultaneously to produce rotation. This rotation returns the ship to where the forces of buoyancy and gravity balance out [14].

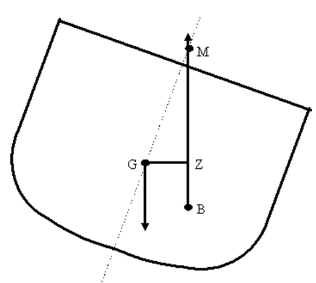


Figure 2.8(a) Righting couple [14].

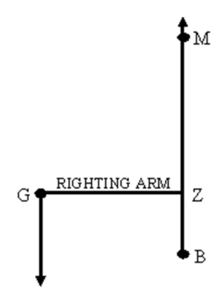
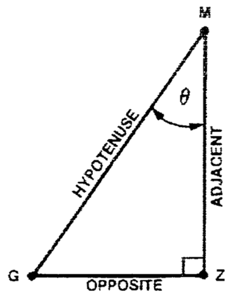


Figure 2.8(b) Righting arm [14].

The distance between the forces of buoyancy and gravity is known as the ship's righting arm. As shown above, the righting arm is a perpendicular line drawn from the center of gravity to the point of intersection on the force of buoyancy line. For small angles of heel (0 degree through 7 to 10 degree, metacenter doesn't move), the value for the ship's righting arm (GZ) may be found by using trigonometry as shown in Figure 2.9.



$$\sin\theta = \frac{Opposite}{Hypotenuse} = \frac{GZ}{GM}$$

$$\cos\theta = \frac{Adjacent}{Hypotenuse} = \frac{MZ}{GM}$$

$$\tan\theta = \frac{Opposite}{Adjacent} = \frac{GZ}{MZ}$$

Figure 2.9 Trigonometric implementation of righting arm [14].

2.9 Righting Moment (RM)

The Righting Moment is the best measure of a ship's overall stability. It describes the ship's true tendency to resist inclination and return to equilibrium. The Righting Moment is equal to the ship’s Righting Arm multiplied by the ship’s displacement [14].

$$RM = GZ \times W_F$$

Where, W_F is the total weight of the vessel.

2.10 Stability Condition

The positions of Gravity and the Metacenter will indicate the initial stability of a ship. The ship will assume one of the following three stability conditions as shown in Figure 2.10.

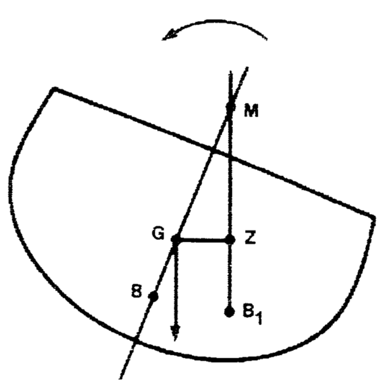


Figure 2.10(a). Positive Stability [14].

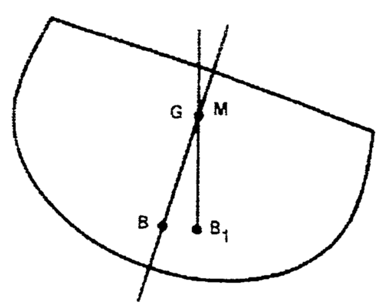


Figure 2.10(b). Neutral Stability [14].

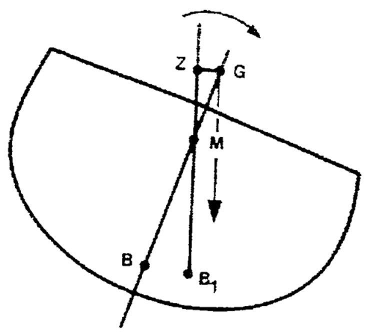


Figure 2.10(c). Negative Stability [14].

2.10.1 Positive Stability

The metacenter is located above the ship's center of gravity. As the ship is inclined, Righting Arms are created which tend to return the ship to its original, vertical position [14].

2.10.2 Neutral Stability

The metacenter and the ship's center of gravity are in the same location. As the ship is inclined, no Righting Arms are created. (until the metacenter starts to move after the ship is inclined past 7-10 degree) [14].

2.10.3 Negative Stability

The ship's center of gravity is located above the metacenter. As the ship is inclined, negative Righting Arms (called upsetting arms) are created which tend to capsize the ship [14].

2.11 Statical Stability Curve (Righting Arm Curve)

When a ship is inclined through all angles of heel, and the righting arm for each angle is measured, the statical stability curve is produced. At particular loading condition this curve is shown in Figure 2.11.

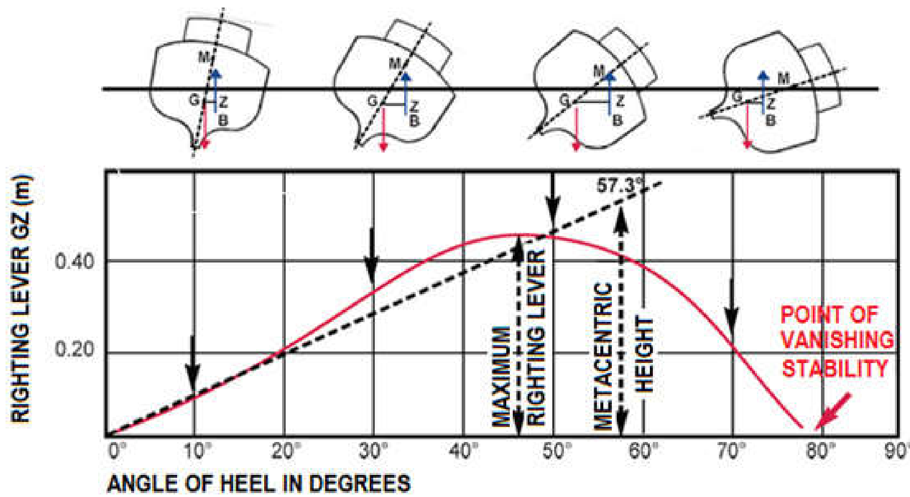


Figure 2.11. Snapshot of the ship's stability at that particular loading condition [17].

Much information can be obtained from this curve, including.

2.11.1 Range of Stability

This ship will generate Righting Arms when inclined from 0 degree to approximately 78 degree according to Figure 2.11. (This curve usually assumes that the entire superstructure is watertight) [14].

2.11.2 Maximum Righting Arm

Maximum righting arm is the largest separation between the forces of buoyancy and gravity. This is where the ship exerts the most energy to right itself [14].

2.11.3 Angle of Maximum Righting Arm

Angle of the maximum righting angle is the angle of inclination where the maximum Righting Arm occurs [14].

CHAPTER 3 Components of the Hardware Used

3.1 Overview

In this chapter the necessary components of the hardware, their specification, connections and feasibility are discussed.

3.2 System Functional Block Diagram

The system is designed for a passenger vessel with two decks. The passenger can occupy these top and bottom decks while on a journey. The weight of passengers is distributed over the entirety of a deck. To measure the load exerted on a single deck by the passenger, six load cells should be mounted symmetrically. There should be transmitter connected to each load cell. So, on the bottom and top deck a total of twelve load cells and associated transmitters are mounted. These transmitters will communicate with the central receiver placed at the Master's room. Figure 3.1 demonstrates the arrangement of the transmitter and the receiver.

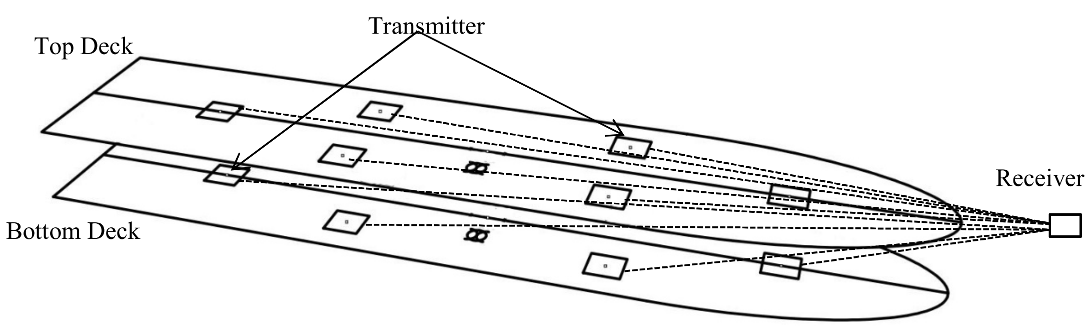


Figure 3.1 Transmitter and receiver arrangements of the two floors.

3.3 Microcontroller

3.3.1 Choice of Microcontroller

There are a range of microcontrollers like PIC, Raspberry Pi, and Atmel etc. For the purpose of this project we chose ATmega32. ATmega32 meets all requirement for the project.

3.3.2 Features of ATmega32

Using ATmega32 is similar to any other microcontroller. Figure 3.2 is real image of an ATmega32. For working with ATmeega32 the code has to be saved in the ATmega32 flash memory. After dumping this program code, the controller executes this code to create the response [18]. There are several IDE to write the code in C. When the code is error free the IDE generates HEX file. This HEX file has to be burnt in the flash memory of the ATmega32. Some of the basic features of ATmega32 are discussed.

Figure 3.2 ATmega32 Microcontroller [18].

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller.
- Advanced RISC Architecture [18].
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier

- High Endurance Non-volatile Memory segments [18].
    - 32Kbytes of In-System Self-Programmable Flash program memory
    - 1024Bytes EEPROM
    - 2Kbytes Internal SRAM
    - 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
- JTAG (IEEE std. 1149.1 Compliant) Interface [18].
    - Boundary-scan Capabilities According to the JTAG Standard
    - Extensive On-chip Debug Support
    - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features [18].
    - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
    - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
    - Real Time Counter with Separate Oscillator
    - Four PWM Channels
    - 8-channel, 10-bit ADC
        - 8 Single-ended Channels
        - 7 Differential Channels in TQFP Package Only
        - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
    - Byte-oriented Two-wire Serial Interface
    - Programmable Serial USART
    - Master/Slave SPI Serial Interface
    - Programmable Watchdog Timer with Separate On-chip Oscillator
    - On-chip Analog Comparator

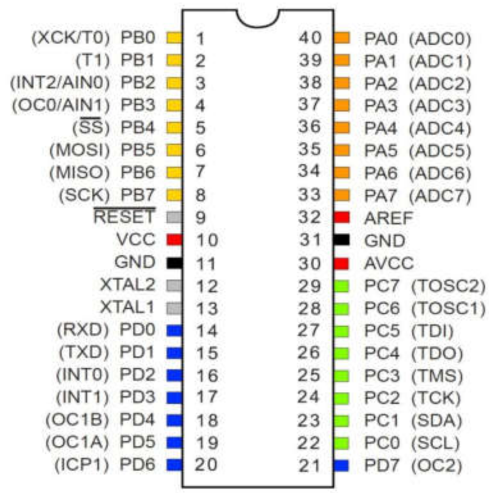


Figure 3.3 ATmega32 pin diagram [18].

- Special Microcontroller Features [18]
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages [18]. Figure 3.3 shows the pin diagram of ATmega32.
  - 32 Programmable I/O Lines
  - 40-pin PDIP
- Operating Voltages [18].
  - 2.7V – 5.5V
- Speed Grades [18].
  - 0 – 16MHz
- Power Consumption at 1MHz, 3V, 25°C [18].
  - Active: 0.6mA
  - Idle Mode: 0.2mA
  - Power-down Mode: < 1µA

3.4 Load Cell

3.4.1 Types of Load Cell

A load cell is a transducer that is used to create an electrical signal whose magnitude is directly proportional to the force being measured. The various types of load cell include hydraulic, pneumatic, and strain gauge type.

3.4.1.1 Hydraulic Load Cells

Hydraulic load cells use a conventional piston and cylinder arrangement to convey a change in pressure by the movement of the piston and a diaphragm arrangement which produces a change in the pressure on a Bourdon tube connected with the load cells. Figure 3.4 demonstrates the principle.

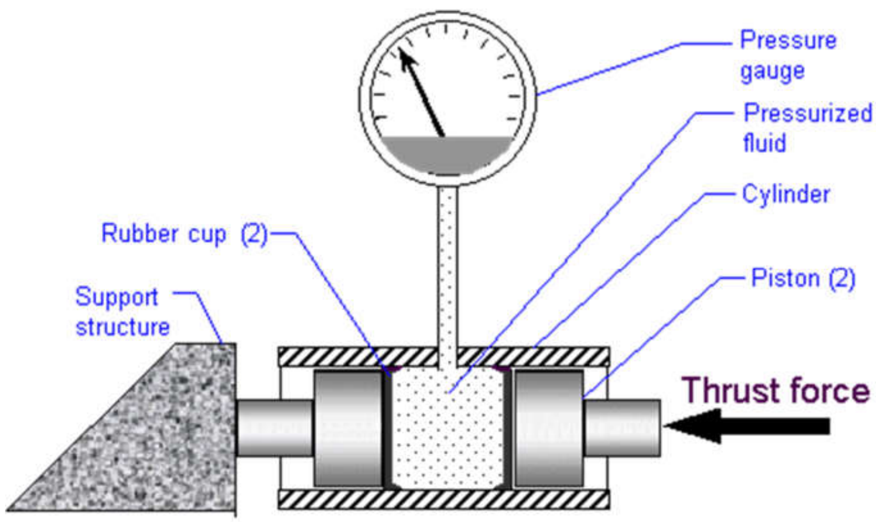


Figure 3.4 Hydraulic Load Cell [19].

3.4.1.2 Pneumatic Load Cells

Pneumatic load cells use air pressure applied to one end of a diaphragm, and it escapes through the nozzle placed at the bottom of the load cell, which has a pressure gauge inside of the cell [19]. A Diagram is shown in Figure 3.5, exhibits a representation of pneumatic load cell.

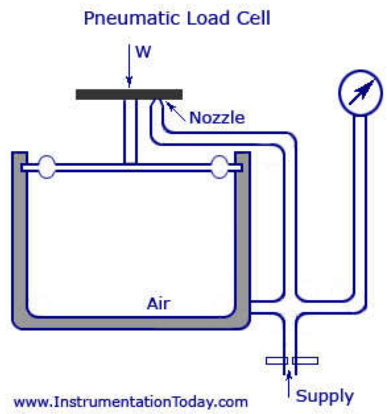


Figure 3.5 Pneumatic Load Cell [19].

3.4.1.3 Strain gauge load cell

Strain gauge load cells are the most common in industry. These load cells are particularly stiff, have very good resonance values, and tend to have long life cycles in application. Strain gauge load cells work on the principle that the strain gauge (a planar resistor) deforms when the material of the load cells deforms appropriately. Deformation of the strain gauge changes its electrical resistance, by an amount that can be related to the strain. The change in resistance of the strain gauge provides an electrical value change that is calibrated to the load placed on the load cell [19]. Figure 3.6 shows a schematic diagram of a bar strain gauge load cell.

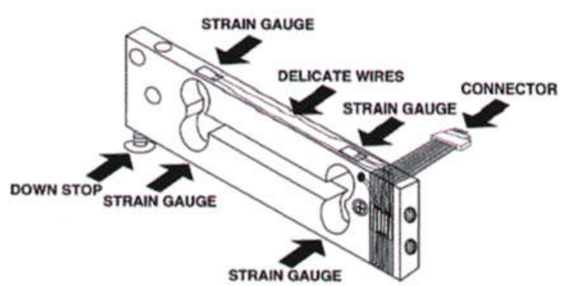


Figure 3.6 Schematic diagram of bar strain gauge load cell [19].

3.4.2 Load Cell Setup

Usually with larger, non-push button bar load cells, the load cell is set up between two plates in a “Z” shape, with fitting screws and spacers so that the strain can be correctly measured as shown in Figure 3.7.

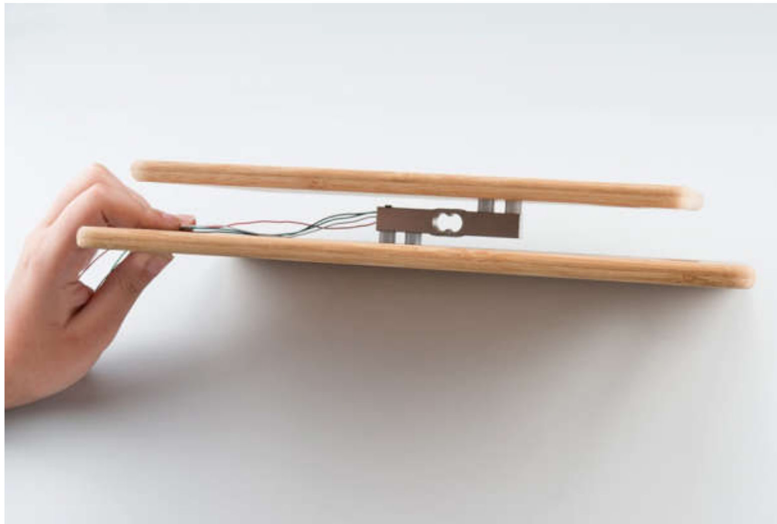


Figure 3.7 Bar strain gauge load cell set up in a "Z" formation between two plates [22].

Note that only one side of the load cell is screwed into each board. This provides a moment of force, or torque, on the strain gauge rather than just compression force, which is easier to measure and much more accurate.

3.4.3 Principle of the Chosen Load Cell

3.4.3.1 Measurement on the basis of Wheatstone bridge

A load cell usually consists of four strain gauges in a Wheatstone bridge configuration. Load cells of one strain gauge (Quarter Bridge) or two strain gauges (half bridge) are also available. The electrical signal output is typically in the order of a few millivolts (mV) and requires amplification by an instrumentation amplifier before it can be used. The output of the transducer can be scaled to calculate the force applied to the transducer. Sometimes a high resolution ADC, typically 24-bit, can be used directly.

A good way of taking small changes in resistance and turning it into something more measurable is using a Wheatstone bridge. A Wheatstone bridge is a configuration of four resistors with a known voltage applied like this:

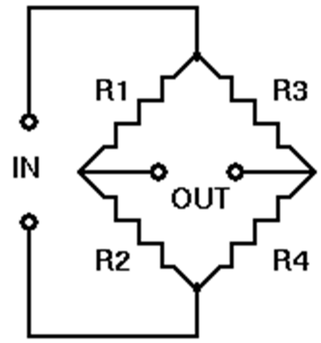


Figure 3.8 Wheatstone bridge [19].

According to Figure 3.8 V_{in} is a known constant voltage, and the resulting V_{out} is measured. If [R1/R2 = R3/R4] then V_{out} is 0, but if there is a change to the value of one of the resistors, V_{out} will have a resulting change that can be measured and is governed by the following equation using ohms law:

$$V_{out} = \left[\frac{R3}{R3+R4} - \frac{R2}{R1+R2}\right] \times V_{in}$$

By replacing one of the resistors in a Wheatstone bridge with a strain gauge, we can easily measure the change in V_{out} and use that to assess the force applied.

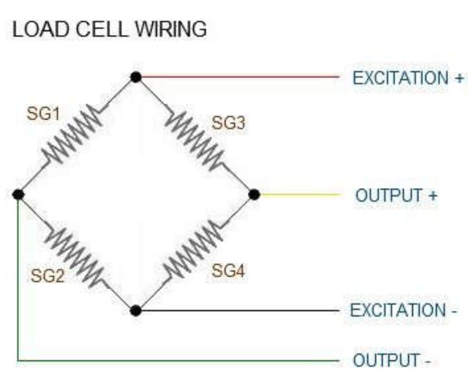


Figure 3.9 Load cell wiring, Wheatstone bridge formation

Four strain gauges (SG1 through 4) connected in a Wheatstone bridge formation. Some load cells might have slight variations in color coding such as blue instead of green or yellow instead of black or white if there are only four wires (meaning no wire used as an EMI buffer). Wiring is shown in Figure 3.9. From the colors available it may require to infer a little, but in general aforementioned color will appear.

3.4.4 Load Cell Amplifier

3.4.4.1 Load Cell Amplifier Specification

We are interfacing a 5 Kg load cell to the ATmega32 using the HX711 load cell amplifier module. HX711 is a precision 24-bit analog to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor. Figure 3.10 shows a commercially available HX711 amplifier.

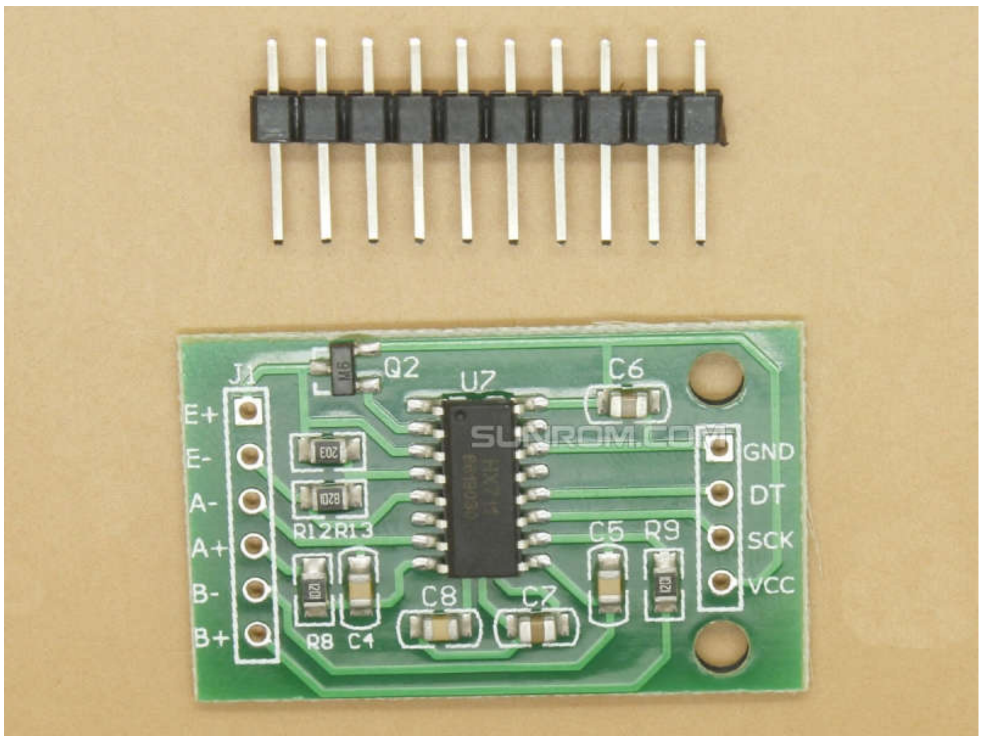


Figure 3.10 Load Cell Amplifier.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of ±20mV or ±40mV respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power on-reset circuitry simplifies digital interface initialization. The reading mechanism of the load data is described in the Appendix-I.

3.4.4.2 Load Cell Amplifier Wiring Connections

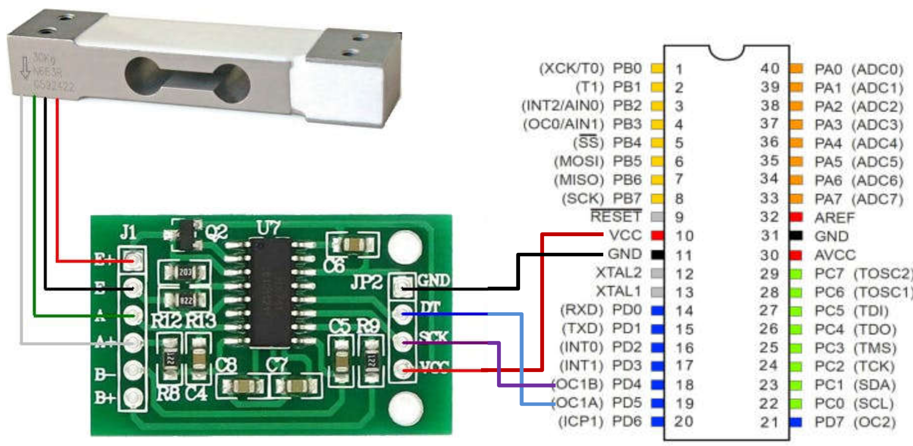


Figure 3.11 Load Cell wires connected to the HX711 Amplifier board and with ATmega32 [23]

Most load cells have four wires: red, black, green and white as shown in Figure 3.11. On the HX711 board E+, E-, A+, A- and B+, B- connections will be found. These four wires coming out from the Wheatstone bridge on the load cell are usually connected:

- Excitation+ (E+) or VCC is red
- Excitation- (E-) or ground is black
- Output+ (O+), Signal+ (S+)+ or Amplifier+ (A+) is white
- Output- (O-), Signal- (S-)+ or Amplifier- (A-) is green

HX711 module operates at 5V and communication is done using serial SDA and SCK pins.

The HX711 Load Cell Amplifier accepts five wires from the load cell. These pins are labeled with colors; RED, BLK, WHT, GRN, and YLW. These colors correspond to the conventional color coding of load cells, where red, black, green and white wires come from the strain gauge on the load cell and yellow is an optional ground wire that is not hooked up to the strain gauge but is there to ground any small outside EMI (electromagnetic interference). Sometimes instead of a yellow wire there is a larger black wire, foil, or loose wires to shield the signal wires to lessen EMI [22].

If the readings from the HX711 are opposite of what are expected (for example the values decrease as weight is increased) simply reversing the O+/O- wires will do the trick. Once the load cell connected to the amplifier, VDD, VCC, DAT, CLK, and GND can be connected to a ATmega32. VCC is the analog voltage to power the load cell. VDD is the digital supply voltage used to set the logic level. In many cases, VCC and VDD can be short together. If 3.3V logic is used in microcontroller however, connect VCC to 5V and VDD to 3.3V.

DAT and CLK hooked up to pin PINB.5 and PINB.4 respectively, but this is easily changed in the code. Any GPIO pin will work for either. Then VCC and VDD just need to be hooked up to 2.7-5V and GND to ground on microcontroller.

3.4.5 Other Considerations of Load Cell

Load cells are ranked, according to their overall performance capabilities, into differing accuracy classes or grades. A specific accuracy grade specifies an error envelope for certain parameters, such as linearity, hysteresis, temperature effects, creep, etc. In practice, certain system accuracy parameters depend considerably on the application of use, physical load introduction to the transducer and disturbing factors such as Zener barriers and surge protection devices. Load cells with different accuracy classes are required depending on the application. The chart provides an overview of typical applications, ranging from the lowest to the highest accuracy class [23].

3.4.5.1 Load Cell Accuracy

Load cell measurements can be off by +/- 5% due to a range of things including temperature, creep, vibration, drift, and other electrical and mechanical interferences. Before installing the scale, one should take a moment and design the system to allow for easy calibration or be able to adjust the code parameters to account for these variations [22].

Load cells with relatively low accuracy classified D1 to C2 are sufficient for simple building materials scales used to weigh sand, cement or water. Adding the right proportion of additives to building materials is essential. For this purpose, special building materials scales using accuracy class C3 load cells are available for mixing additives such as ash or sand. Accuracy class C3 load cells are widely used in machine construction as well. Here, scales contribute to quality assurance, for example, when ball bearings are checked.

However, increased accuracy is needed with shop-counter scales or scales used in filling machines. Grams or micrograms are required here. Load cells used in these applications comply with accuracy classes C3 to C6 [20].

3.4.5.2 Load Cell Calibration

Load cells are an integral part of most weighing systems in industrial, aerospace and automotive industries, enduring rigorous daily use. Over time, load cells will drift, age and misalign therefore they will need to be calibrated regularly to ensure accurate results are maintained. ISO9000 and most other standards specify a maximum period of around 18 months to 2 years between re-calibration procedures, dependent on the level of load cell deterioration. Annual re-calibration is considered best practice by many load cell users for ensuring the most accurate measurements [21].

3.5 Transceiver (Bluetooth HC-05)

3.5.1 Selection of Transceiver

As transceiver infra-red, WI-FI, Bluetooth can be used. In this project Bluetooth is used. There are also several choices for Bluetooth like HC-04, HC-06, HM-11, ESP32, CSR8645. For this project HC-05 Bluetooth module is used. HC-05 is low cost and easy to work with.

3.5.2 HC-05 Bluetooth module

The HC-05 is a very good module which can add two-way (full-duplex) wireless functionality to the projects. Figure 3.12 is a commercially available HC-05 Bluetooth Module. This module can be used to communicate between two microcontrollers or communicate with any device with Bluetooth functionality like a Phone or Laptop. There are many android applications that are already available which makes this process a lot easier. The module communicates with the help of USART at 9600 baud rate hence it is easy to interface with any microcontroller that supports USART. We can also configure the default values of the module by using the command mode. It can be used to transfer data from computer or mobile phone to microcontroller or vice versa. However, this module cannot transfer multimedia like photos or songs; CSR8645 module can be explored for those purposes [24].

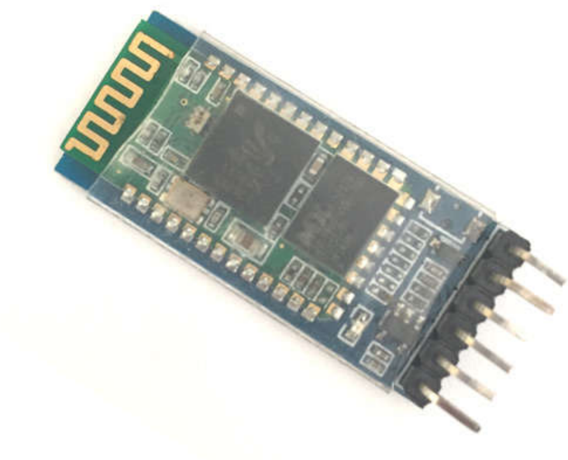


Figure 3.12 Bluetooth Module HC-05 [24].

3.5.3 HC-05 Default Settings [24]

- Default Bluetooth Name: “HC-05”
- Default Password: 1234 or 0000
- Default Communication: Slave
- Default Mode: Data Mode
- Data Mode Baud Rate: 9600, 8, N, 1
- Command Mode Baud Rate: 38400, 8, N, 1
- Default firmware: LINVOR

3.5.4 HC-05 Technical Specifications [24]

- Serial Bluetooth module for microcontrollers
- Operating Voltage: 4V to 6V (Typically +5V)
- Operating Current: 30mA
- Range: <100m
- Works with Serial communication (USART) and TTL compatible
- Follows IEEE 802.15.1 standardized protocol
- Uses Frequency-Hopping Spread spectrum (FHSS)
- Can operate in Master, Slave or Master/Slave mode

- Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

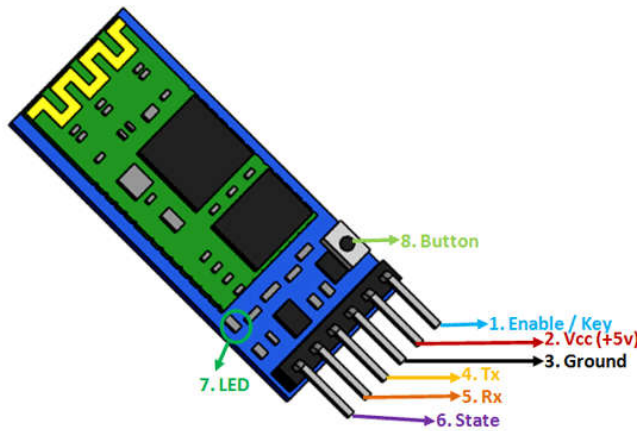


Figure 3.13 Bluetooth Module Pin-out [24].

3.5.5 HC-05 Bluetooth Module Guideline

The HC-05 has two operating modes, one is the Data mode in which it can send and receive data from other Bluetooth devices and the other is the AT Command mode where the default device settings can be changed. We can operate the device in either of these two modes by using the key pin as explained in the pin description [24]. Figure 31.3 show the Bluetooth pin-out of HC-05.

It is very easy to pair the HC-05 module with microcontrollers because it operates using the Serial Port Protocol (SPP). Simply power the module with +5V and connect the Rx pin of the module to the Tx of MCU and Tx pin of module to Rx of MCU as shown in Figure 3.14.

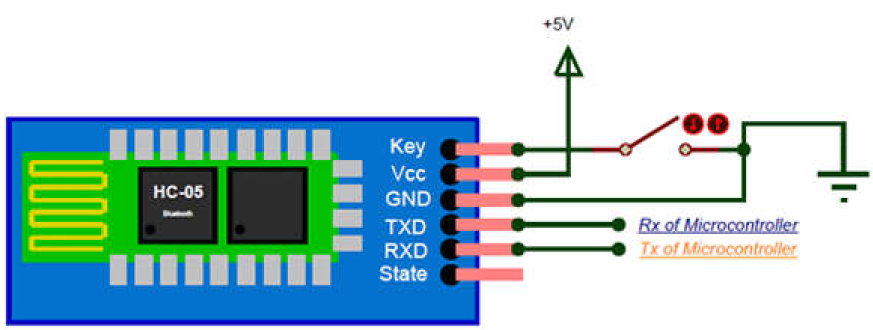


Figure 3.14 HC-05 connections [24].

During power up the key pin can be grounded to enter into Command mode, if left free it will by default enter into the data mode. As soon as the module is powered one should be able to discover the Bluetooth device as “HC-05” then connect with it using the default password 1234 and start communicating with it. The name password and other default parameters can be changed by entering into the command mode [24].

3.5.6 Bluetooth Applications [24]

- Wireless communication between two microcontrollers
- Communicate with Laptop, Desktops and mobile phones
- Data Logging application
- Consumer applications
- Wireless Robots
- Home Automation

CHAPTER 4 Implementation of the Proposed System

4.1 Overview

This chapter outlines the plan for mounting the hardware and the Bluetooth connectivity. Result and discussions are discussed at the end of the chapter. For the purpose of this study we considered a passenger vessel with two decks.

4.2 Transceiver Mounting Plan

As explained in the article 3.5.1, there should be six load-cells mounted at some predefined positions in the lower and upper deck and each load cell is connected with a transmitter which will transmit the load data to the receiver module at the center of the vessel. This receiver will calculate the total load of each of the decks and also determine the center of gravity of the vessel. Figure 4.1 demonstrates the probable position of the load cells in each deck. The receiver module at the Master's deck which will receive the data from the twelve transmitters of the two decks. The receiver will calculate the CG of the whole ship for particular loading conditions repeatedly.

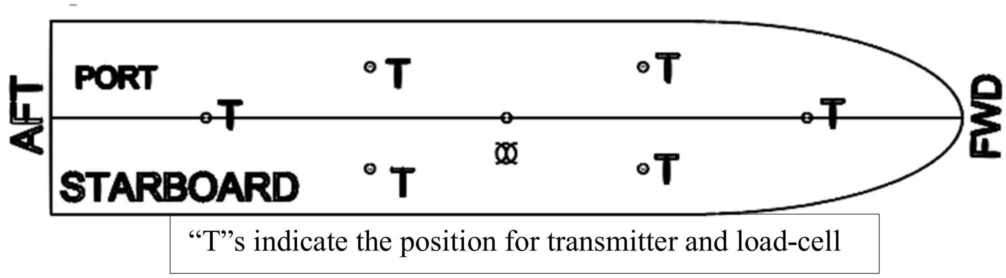


Figure 4.1 Position indicated for mounting Load Cells on a deck of a passenger vessel.

4.3 Interpreting Vessel Specifications

GM (Metcentric Height) of a fully loaded vessel should be neither too small nor too large. There are typical values for GM for different types of ship to be followed. A metacentric diagram must be consulted to find the KB and BM. From them KM can be calculated. If CG is calculated from the loadcell network, then KG is known and

KM is taken from graph (Figure 4.2). So the difference GM is calculated and can be determined if GM conforms the allowable limit.

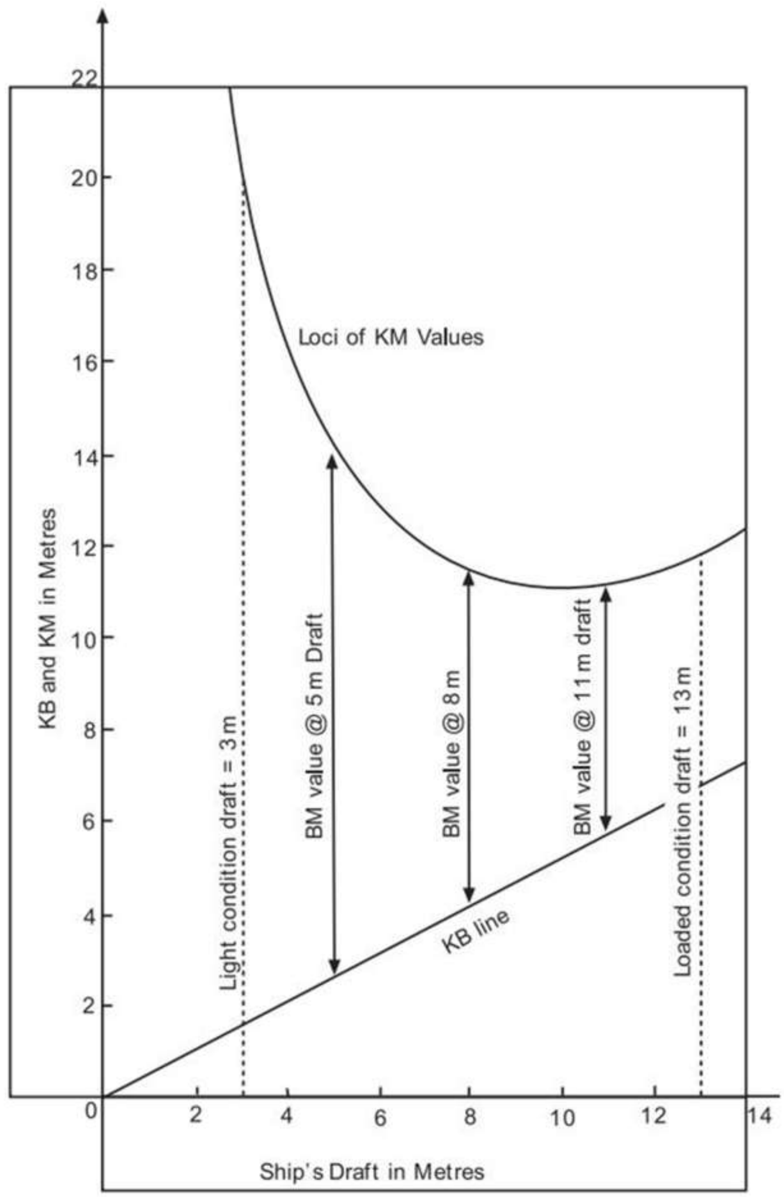


Figure 4.2 Metacentric diagram of a ship shaped vessel.

4.4 Processing Sensor Data

The microcontroller connected to the load cell at each of the decks calculates the load on that node. Then load data from all load cells is transmitted to the central module placed at the master's room. This module will receive the data from all the deck and calculate the total loading condition on the vessel and the vertical position of the vessels overall CG. After calculation the microcontroller decides if the ship is in safe loading condition. The microcontroller updates the Metacentric height from the vessels architectural diagram according to the loading condition and the heel. Then it will decide if the vessel is in Position, Neutral or Negative stability condition.

4.5 Alarm System

There will be two systems of alarm. One will be dedicated to the loading condition monitoring. And, other will be associated with the instability caused by the vertical shift in CG of the vessel. This loading will alarm will warn both the master's and the passengers if the vessel is overloaded. The alarm associated with instability will warn if the vessel is approaching the neutral stability and of course, if the negative stability occurs the alarm will be louder.

4.6 Experimental Setup

The experimental setup is arranged with three nodes. Two for nodes are transmitter and one node is receiver. Figure 4.3 shows the arrangement of the load cell, receiver and transmitter. Figure 4.4 shows the hardware emulation of the project.

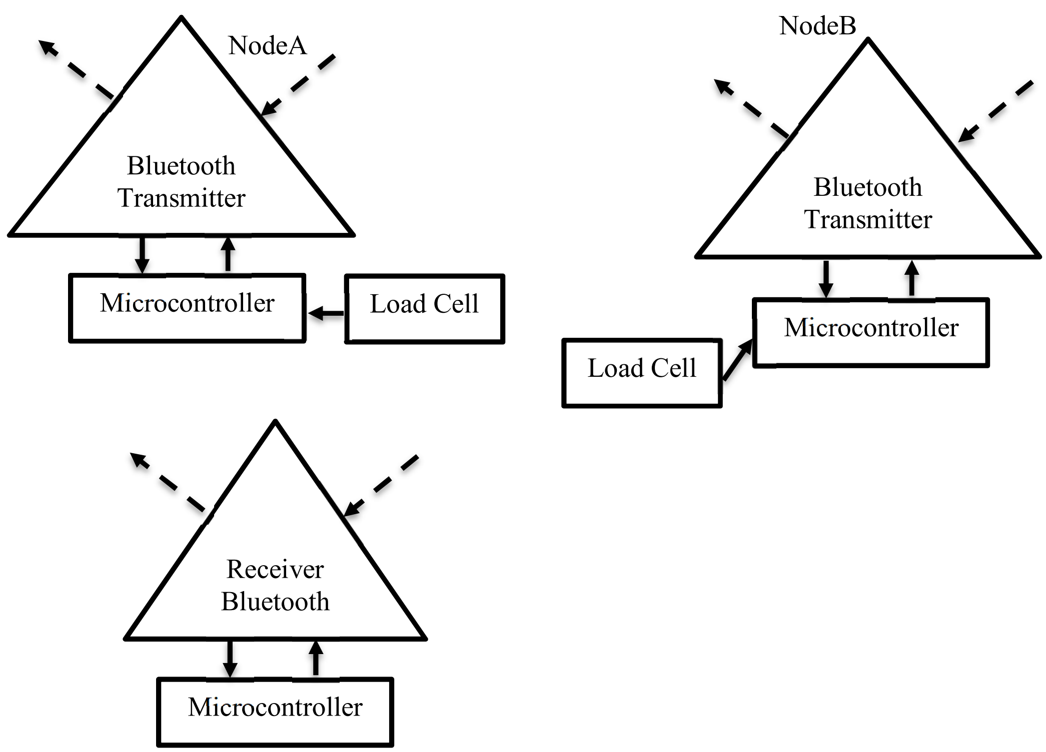


Figure 4.3 Block Diagram of the experimental setup.

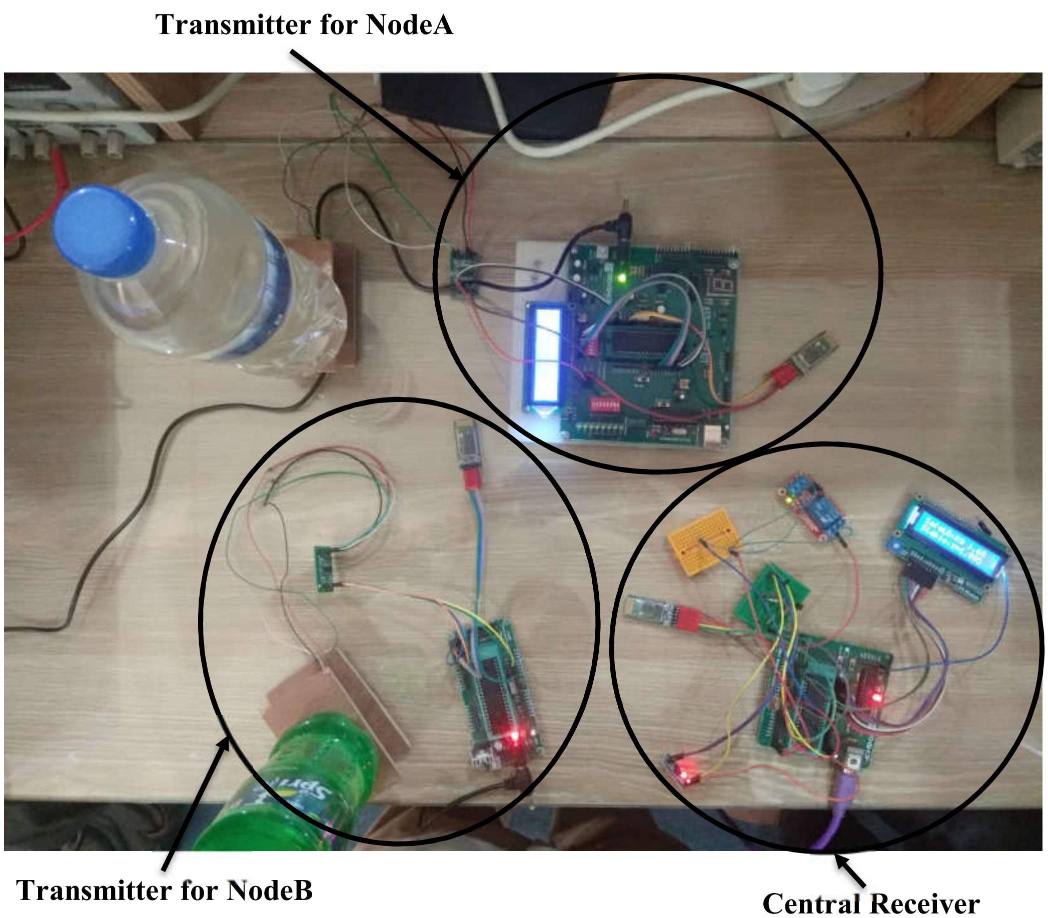


Figure 4.4 Hardware emulation of the Project.

4.6.1 Transmitter Setup

At the transmitter node each of the two load cells are connected with ATmega32. And the microcontroller is connected to Bluetooth module. We considered one load cell to be NodeA and other to be NodeB. Bluetooth at NodeA and NodeB is in slave mode and the receiver Bluetooth is in master mode. Baud rate of the all three Bluetooth set to be 38400.

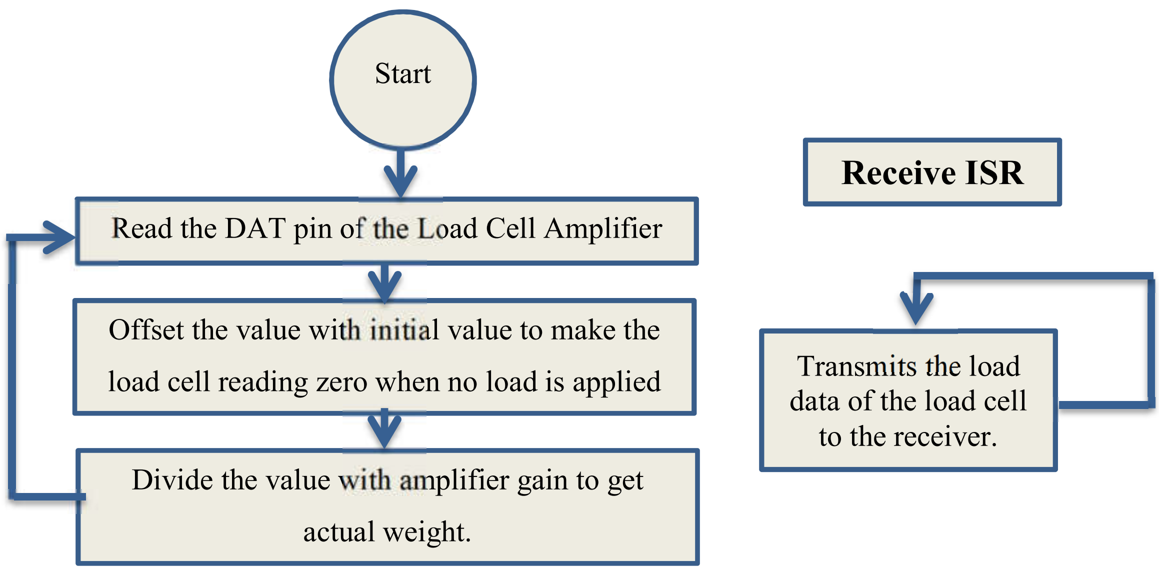
4.6.2 Receiver Module Setup

A receiver Bluetooth is connected with a ATmega32 which will receive the load data from two transmitters and calculate the total load and position of CG. This receiver node controls the alarm system.

The receiver Bluetooth set the Binding address in command mode using AT+BIND command. First, the receiver sets the binding address, the Bluetooth address of NodeA. Then it receives the data from NodeA. After that the receiver sets the binding address the, Bluetooth address of NodeB. Then it receives the data from NodeB.

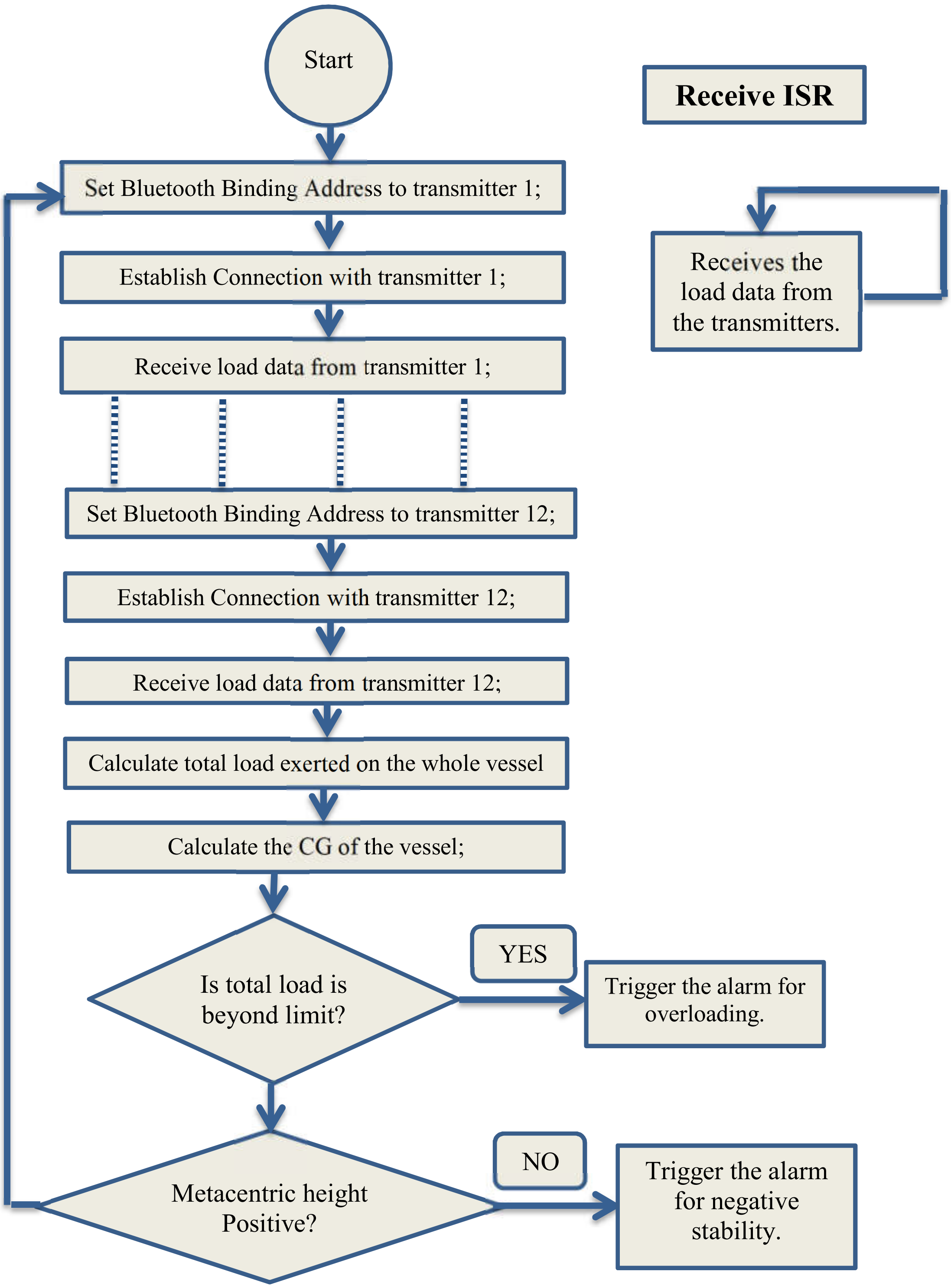
4.7 Top/Bottom Deck Transmitter Algorithm

Algorithm for transmitter at each floor is given below. Code is in the Appendix-II.



4.8 Master's Room Central Transceiver Algorithm

Algorithm for Receiver AT Master's Deck is given below. Code is in the Appendix-III.



4.9 Results and Findings

The coordinate system is considered to be three dimensional. The vessel is called lightweight, when for a passenger vessel all the load excluding passenger is considered. This is the initial condition of the vessel. At this point weight of machinery, oil, shell plating framing etc. determines the initial position of CG of the lightweight vessel. The total load at this point is fixed and known to Master. We considered the x, y, z coordinate of the initial position of CG is (0, 0, 0). So, now if the passengers come onboard the CG will shift from initial position.

We considered the total load of the upper deck is acting through the point where NodeA is mounted. Also, total load of the lower deck is acting through the point where NodeB is mounted. So, for the simulation purpose, considering the load at NodeA and NodeB will suffice to calculate total load of the vessel and the new position of CG of the vessel loaded with passengers. The results are shown on the LCD display. LCD display also shows the remarks about the load placed on the loadcells.

And coordinate of metacenter is taken to be fixed at (0, 0, 3). And the safe loading condition is below 5 kg. The position of NodeA is taken 4 meter above the CG. So, z-coordinate of NodeA is 4. And, the position of NodeB is taken 1 meter below the CG. So, z-coordinate of NodeA is -1.

4.9.1 Example Case 1

Now 2 kg is placed on the load cell at NoadA and 0.6 kg is placed on the load cell the at NodeB. So the total load is (2+0.6) =2.6 kg, within the safe loading condition.

From article 2.3,

$$\bar{z} = \frac{z_1 m_1 + z_2 m_2 + z_3 m_3 + \cdots\cdots\cdots + z_n m_n}{m_1 + m_2 + m_3 + \cdots\cdots\cdots + m_n}$$

The vertical position of CG becomes,

$$z = \frac{4 \times 4 + (-1) \times 0.6}{0.6 + 2}$$

=3.347

So the new position of CG is (0, 0, 3.347). The new position of CG is above the metacenter. So, this loading arrangement will create capsizing moment. To correct the negative couple one can either lower the load 4 kg from the position 4 meter above the

CG like below/nearer (0, 0, 0). Also removing the 4 kg load from NodeA will solve the problem.

4.9.2 Example Case 2

Again 2 kg is placed on the load cell at NoadA and 4 kg is placed on the load cell the at NodeB. The total load is (2+4) =6 kg, which is beyond the safe loading condition of 5 kg. So some load has to be removed from either of the node, but it is wise to remove load from the NoadA as this will facilitate achieving positive stability.

From article 2.3,

$$\bar{z} = \frac{z_1 m_1 + z_2 m_2 + z_3 m_3 + \cdots\cdots + z_n m_n}{m_1 + m_2 + m_3 + \cdots\cdots + m_n}$$

The vertical position of CG becomes,

$$z = \frac{4 \times 2 + (-1) \times 4}{2 + 4}$$

=0.6666

So the new position of CG is (0, 0, 0.6666). The new position of CG is below the metacenter. So, this loading arrangement is in positive stability condition.

4.9.3 Test result for different combination of weight

Table 4.1 Results found with different weights at NodeA and NodeB

Load on NodeA	Load on NodeB	Total Load	CG	Remarks
3 kg	3 kg	6 kg	1.5 m	Overload and positive stability. Some load should be removed
1 kg	3 kg	4 kg	0.25 m	Safe loading and positive stability
0.5 kg	4 kg	4.5 kg	-0.4444 m	Safe loading and positive stability
3 kg	1 kg	4 kg	2.75 m	Approaching Neutral Stability. Should shift the load lower from NodeA or remove the load.
4 kg	0.6 kg	4.6 kg	3.347 m	Safe loading but negative stability. Should shift the load from NodeA or remove the load.

4.9.4 Summary

In this project various possible scenario is tested. Safe loading condition, overloading is tested. Also positive stability, neutral stability and negative stability is demonstrated in the results.

CHAPTER 5 Conclusion

5.1 Conclusion

This project mainly focused on the hardware implementation of the alarm system. Implementation of this project shows some great prospects and also some real challenges. So, there should be some value judgments to decide if this project were to be implemented in the field. With the successful completion of this project, an autonomous alarm system may be developed that will make the river journey safer for passengers.

5.1.1 Strengths of the Project

- The proposed idea in this project addressed some vital issues of Bangladesh river transport systems.
- The implementation of this project will save thousands of lives.

5.1.2 Limitations

- This project is implemented in laboratory settings. It was not implemented in any practical field.
- Special investigation should be done whether there is any issue regarding the structural integrity of the vessel if the proposed hardware is mounted.

5.1.3 Challenges

There are some real challenges if this project ever were to be implemented.

1. The Metacenter is considered to be fixed for this project. But in reality metacenter shifts with the draft.
2. Load measurement is for a deck considered to be accumulated at a central point at the deck. But load is generally distributed over the deck.
3. In this project Bluetooth is used for the wireless network. But Bluetooth has some issues with range and connectivity. Bluetooth generally works in open air and within short distance. In that case WI-FI or GSM technology could be used.

5.2 Future Work

There are scopes of improvement in the project. Some of them have been described below.

1. While implementing this project special care should be taken to ensure the structural integrity of the vessel.
2. During implementation the metacenter should be readily updated after consulting the design graph and draft.
3. For estimating the load on the deck, at least more than 5 load cells should be mounted for a single deck.
4. It is highly recommended that, before implementing the idea we must test the system on a life size model to assess its effectiveness and viability.
5. This work mainly dealt in creating a Bluetooth network. But for larger vessel Wi-Fi/GSM network may be considered.

References

[1] Wang, J., “Offshore safety case approach and formal safety assessment of ships,” Journal of Safety Research, vol. 33(1), pp. 81-115, Spring. 2002.

[2] Akyuz, E., and Celik, M., “A hybrid decision-making approach to measure effectiveness of safety management system implementations on-board ships,” Safety Science, vol. 68, pp. 169-179, Oct. 2014.

[3] https://www.thedailystar.net/backpage/fear-and-frenzy-waterways14194002014 [last accessed date 27/10/2018].

[4] Uddin, M. I., Islam, M. R., Awal, Z. I., & Newaz, K. M. S., “An Analysis of Accidents in the Inland Waterways of Bangladesh: Lessons from a Decade (2005-2015),” Procedia engineering, 194 (2017), pp. 291-297, Aug. 2017.

[5] Eleftheria, E., Apostolos, P., & Markos, V., “Statistical analysis of ship accidents and review of safety level,” Safety science, vol. 85, pp. 282-292, June 2016.

[6] Kim, D. J., & Kwak, S. Y., “Evaluation of human factors in ship accidents in the domestic sea,” Journal of the Ergonomics Society of Korea, vol. 30(1), pp. 87-98, 2011.

[7] Barrass, B., & Derrett, C. D., “Ship stability for masters and mates,” in Elsevier (eds), chap. 6, 2011.

[8] Zheng, H., Huang, Y., & Ye, Y., "New level sensor system for ship stability analysis and monitor." IEEE Transactions on Instrumentation and Measurement, issue 6, vol. 48, pp. 1014-1017, 1999.

[9] Pennanen, P., Ruponen, P., & Ramm-Schmidt, H., “Integrated decision support system for increased passenger ship safety,” Damaged Ship III, Presented at the Royal Institution of Naval Architects, London, UK, Mar. 2015.

[10] Surendran, S., & Reddy, J. V. R., “Numerical simulation of ship stability for dynamic environment,” Ocean Engineering, vol. 30(10), 1305-1317, 2003.

[11] Kreuzer, E., & Wendt, M., “Ship capsizing analysis using advanced hydrodynamic modeling,” Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 358(1771), 1835-1851, 2000.

[12] Fan, L., Low, M. Y. H., Ying, H. S., Jing, H. W., Min, Z., & Aye, W. C., “Stowage planning of large containership with tradeoff between crane

workload balance and ship stability," In *Proceedings of the International Multi Conference of Engineers and Computers Scientists*, Vol. 3, pp. 1-7, March 2010.

[13] https://study.com/academy/lesson/what-is-center-of-gravity-definition-equati-on -examples.html [last accessed date 09/12/2018].

[14] https://fas.org/man/dod-101/navy/docs/swos/dca/stg4-01.html [last accessed date 09/12/2018].

[15] https://www.wartsila.com/encyclopedia/term/metacentre-m-(initial-transverse-metacentre) [last accessed date 09/12/2018].

[16] https://www.marineinsight.com/naval-architecture/ship-stability-introduction-hydrostatics-stability-surface-ships/ [last accessed date 09/12/2018].

[17] https://plato.is/stability_of_fishing_vessels/righting_lever/ [last accessed date 09/12/2018].

[18] https://protostack.com.au/shop/microcontrollers/atmega32a-pu-atmel-8-bit-32-avr-microcontroller/ [last accessed date 09/12/2018].

[19] https://learn.sparkfun.com/tutorials/getting-started-with-load-cells [last accessed date 09/12/2018].

[20] https://www.hbm.com/en/2637/oiml-accuracy-classes-explained/ [last accessed date 10/12/2018].

[21] https://www.upcscavenger.com/wiki/load_cell/#page=wiki [last accessed date 10/12/2018].

[22] https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hook-up-guide/all [last accessed date 09/12/2018].

[23] https://www.hackster.io/MOHAN_CHANDALURU/hx711-load-cell-amplif-ier-interface-with-arduino-fa47f3 [last accessed date 09/12/2018].

[24] https://components101.com/wireless/hc-05-bluetooth-module [last accessed date 09/12/2018].

APPENDIX-I

The load Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls. When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25th pulse at PD_SCK input will pull DOUT pin back to high (Figure 1). Input and gain selection is controlled by the number of the input PD_SCK pulses (Table 1). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

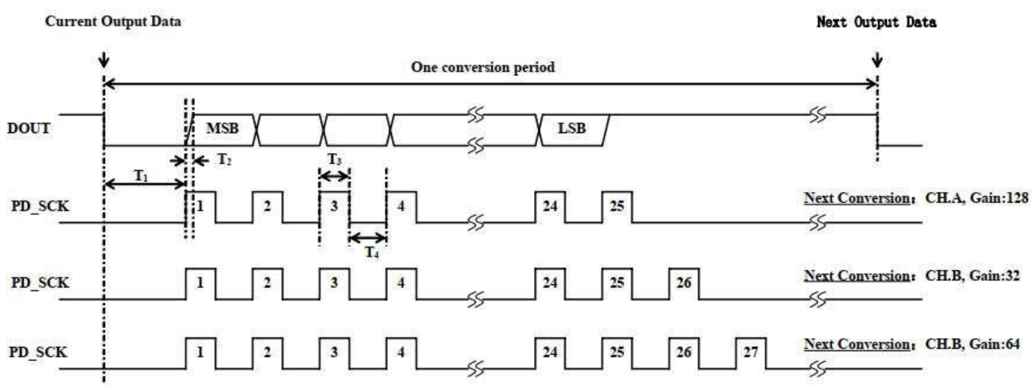


Figure 1 Data output, input, and gain selection timing and control.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 1 Input Channel and Gain Selection

APPENDIX-II

Code for transmitter at NodeA is given below

```
#include <mega32.h>
// Alphanumeric LCD functions
#include <alcd.h>
#include <delay.h>
#include <string.h>
#include <stdlib.h>
#define AD_DATA_PIN = PIND.5; // Pin mapping of microcontroller
#define AD_SCK_PIN = PIND.4; // Pin mapping of microcontroller

// Declare your global variables here
char mychar, loadData[4];       //To declare 8bit character variable

// Standard Input/Output functions
#include <stdio.h>
void usart_transmit(unsigned char ch )
{
        while ( !( UCSRA & (1<<UDRE)) );
        UDR = ch;
}

interrupt [USART_RXC] void myInterrupt(void)
{
        int j=0;
        mychar=getchar();
        if(mychar=='U')
        {
                lcd_clear();
                lcd_gotoxy(0, 0);
                lcd_puts("Transmitting");

                for (j=0; j<strlen(loadData); j++)
                {
                        usart_transmit(loadData[j]);
                }
                usart_transmit('M');
        }
}
```

```
unsigned long ReadCount()
{
        unsigned long Count;
        unsigned char i;
        PORTD.4=0;              //AD_SCK_PIN
        Count=0;
        while(PIND.5);          //AD_DATA_PIN
        for (i=0;i<24;i++)
        {
                PORTD.4=1;              //AD_SCK_PIN
                Count=Count<<1;
                PORTD.4=0;              //AD_SCK_PIN
                if(PIND.5) Count++;         //AD_DATA_PIN
        }
        Count=Count^0x800000;
        for(i=0;i<2; i++)
        {
                PORTD.4=1;                  //AD_SCK_PIN
                PORTD.4=0;                  //AD_SCK_PIN
        }
        return(Count);
}

void main(void)
{
// Declare your local variables here
char disp[16], loadTemp[4];
unsigned long data=0;
int w, finalWeight;
// Input/Output Ports initialization
DDRA=0xFF;
DDRD.4=1;

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
MCUCSR=0xFF;    //(0<<ISC2);
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
```

```
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) |
(0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) |
(1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x19;
#asm ("sei");
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16
lcd_init(16);
while (1)
{
        data=ReadCount();
        w=data/32;
        finalWeight=466;
        finalWeight+= w;
        itoa(finalWeight, disp);

        if(finalWeight/3<1000)
        {
                itoa(0,loadData);
                itoa(finalWeight/3,loadTemp);
                strcat(loadData, loadTemp);
        }
        else{
                itoa(finalWeight/3,loadData);
        }
        lcd_clear();
        lcd_gotoxy(0,0);
```

```
        lcd_puts(disp);
        lcd_gotoxy(0,1);
        lcd_puts(loadData);
        delay_ms(1000);
    }
}
```

APPENDIX-III

Code for Receiver is given below.

```
#include <mega32.h>
#include <alcd.h>
#include <delay.h>
#include <string.h>
#include <stdlib.h>
#define bt_vcc PORTB.5
#define bt_en PORTB.6
#define bt_state PORTB.7

// Declare your global variables here
unsigned char at_rmaad[8], at_bind_A[26], at_bind_B[26];
unsigned char recData[5];       //To declare 8bit character variable
char i=0, b=0;    //To declare single bit

// Standard Input/Output functions
#include <stdio.h>

void usart_transmit(unsigned char ch )
{
        while ( !( UCSRA & (1<<UDRE)) );
        UDR = ch;
}

interrupt [USART_RXC] void myInterrupt(void)
{
        recData[i] = getchar();    //Get Character from UDR
        if(recData[i]=='M')
        {
                recData[i]=0x00;
                b=1;
        }

        i++;
        if(i>=5)
        {
                i=0;
        }
```

```
}

void pair_with_bluetooth_A(void)
{
	int j=0;
	//UCSRB=(0<<RXCIE)|(0<<RXEN);
	PORTB.6=0;
	PORTB.5=0;
	delay_ms(200);

	PORTB.6=1;
	delay_ms(100);
	PORTB.5=1;
	for (j=0; j<strlen(at_rmaad); j++)
	{
		usart_transmit(at_rmaad[j]);
	}
	delay_ms(1000);

	for (j=0; j<strlen(at_bind_A); j++)
	{
		usart_transmit(at_bind_A[j]);
	}
	delay_ms(1000);

	PORTB.6=0;
	PORTB.5=0;
	delay_ms(200);
	PORTB.5=1;
}

void pair_with_bluetooth_B(void)
{
	int j=0;
	PORTB.6=0;
	PORTB.5=0;
	delay_ms(200);

	PORTB.6=1;
	delay_ms(100);
	PORTB.5=1;
```

```
        for (j=0; j<strlen(at_rmaad); j++)
        {
                usart_transmit(at_rmaad[j]);
        }
        delay_ms(1000);

        for (j=0; j<strlen(at_bind_B); j++)
        {
                usart_transmit(at_bind_B[j]);
        }
        delay_ms(1000);

        PORTB.6=0;
        PORTB.5=0;

        delay_ms(200);
        PORTB.5=1;
}

void main(void)
{
// Declare your local variables here
char disp1[16], disp2[16];
unsigned char loadA[5], loadB[5];
float A=0, B=0, Tot=0;
float x=0, y=0, z=0;
int clk=0;
char coordX[5], coordY[5], coordZ[5], totalLoad[5];

// Input/Output Ports initialization
DDRA=0xFF;

// Port B initialization
// Function: Bit7=In Bit6=Out Bit5=Out Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
DDRB=(0<<DDB7) | (1<<DDB6) | (1<<DDB5) | (0<<DDB4) | (0<<DDB3) |
(0<<DDB2) | (0<<DDB1) | (0<<DDB0);
// State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
PORTB=(0<<PORTB7) | (0<<PORTB6) | (0<<PORTB5) | (0<<PORTB4) |
(0<<PORTB3) | (0<<PORTB2) | (0<<PORTB1) | (0<<PORTB0);

MCUCR=(0<<ISC11) | (0<<ISC10) | (0<<ISC01) | (0<<ISC00);
```

```
MCUCSR=0xFF;  //(0<<ISC2);
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: Off
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=(0<<RXC) | (0<<TXC) | (0<<UDRE) | (0<<FE) | (0<<DOR) | (0<<UPE) |
(0<<U2X) | (0<<MPCM);
UCSRB=(1<<RXCIE) | (0<<TXCIE) | (0<<UDRIE) | (1<<RXEN) | (1<<TXEN) |
(0<<UCSZ2) | (0<<RXB8) | (0<<TXB8);
UCSRC=(1<<URSEL) | (0<<UMSEL) | (0<<UPM1) | (0<<UPM0) | (0<<USBS) |
(1<<UCSZ1) | (1<<UCSZ0) | (0<<UCPOL);
UBRRH=0x00;
UBRRL=0x19;

#asm ("sei");
// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16
lcd_init(16);
strcpy(at_rmaad, "AT+RMAAD\r\n");
strcpy(at_bind_A, "AT+BIND=98D3,A1,F5B548\r\n");
strcpy(at_bind_B, "AT+BIND=98D3,21,F73B81\r\n");
PORTB.5=1;

while (1)
    {
        //Receiving from Bluetooth transmitter A
        //coordinate of the loadcell: x=10m, y=5m, z=4m
                pair_with_bluetooth_A();
                while(PINB.7==0);
                delay_ms(2000);
```

```
        usart_transmit('U');
        while(b==0 && clk<50)
        {
            i=0;
            usart_transmit('U');
            delay_ms(100);
            clk++;
        }
        strcpy(loadA, recData);
        clk=0;
        b=0;
        i=0;
//Receiving from Bluetooth transmitter B
//coordinate of the loadcell: x=-8m, y=3m, z=-1m
        pair_with_bluetooth_B();
        while(PINB.7==0);
        delay_ms(2000);
        usart_transmit('V');
        while(b==0 && clk<50)
        {
            i=0;
        usart_transmit('V');
        delay_ms(100);
        clk++;
        }
        strcpy(loadB, recData);
        b=0;
        i=0;
        A=atof(loadA);
        B=atof(loadB);
        if(A<100)
        {
            A=0;
        }
        if(B<100)
        {
            B=0;
        }
        A/=1000;
        B/=1000;
        x=(10*A-8*B)/(A+B);
```

```
        y=(5*A+3*B)/(A+B);
        z=(4*A-B)/(A+B);
        if(A==0 && B==0)
        {
                x=0;
                y=0;
                z=0;
        }
        Tot=A+B;
        if(Tot>5.0)
        {
                strcpy(disp1, "OverLD,Kg ");
                ftoa(Tot,2, totalLoad);
                strcat(disp1, totalLoad);
        }
        else
        {
                strcpy(disp1, "SafeLD,Kg ");
                ftoa(Tot,2, totalLoad);
                strcat(disp1, totalLoad);
        }
//Metacenter z-coordinate 3m
        if(z>3.0)
        {
                strcpy(disp2, "Instable,z=");
                ftoa(z,3, coordZ);
                strcat(disp2, coordZ);
        }
        else if(z==3.0)
        {
                strcpy(disp2, "Neutral,z=");
                ftoa(z,3, coordZ);
                strcat(disp2, coordZ);
                }
        else
        {
                strcpy(disp2, "Stable,z=");
                ftoa(z,3, coordZ);
                strcat(disp2, coordZ);
        }
lcd_clear();
```

```
        lcd_gotoxy(0, 0);
        lcd_puts(disp1);
        lcd_gotoxy(0, 1);
        lcd_puts(disp2);
    }
}
```