Volume XXIV 2021
ISSUE no.2
MBNA Publishing House Constanta 2021

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

The practical design of automation and monitoring petroleum products transfer system from a tankship

To cite this article: P. Burlacu, V. Mocanu, V. Dobref, O. Cristea, C. Clinci, E. Robe, Scientific Bulletin of Naval Academy, Vol. XXIV 2021, pg. 152-158.

Submitted: 08.12.2021
Revised: 15.01.2022
Accepted: 20.01.2022

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-21-I2-015

The practical design of automation and monitoring petroleum products transfer system from a tankship

P Burlacu[1], V Mocanu[2], V Dobref[3], O Cristea[4], C Clinci[5], E Robe[6]

[1]Associate Professor PhD Eng, Naval Academy "Mircea cel Bătrân", Constanța, RO
[2]Asist. Univ. PhD Candidate, Naval Academy "Mircea cel Bătrân", Constanța, RO
[3]Professor PhD Eng, Naval Academy "Mircea cel Bătrân", Constanța, RO
[4]Lecturer PhD Eng, Naval Academy "Mircea cel Bătrân", Constanța, RO
[5]Lecturer PhD Eng, Naval Academy "Mircea cel Bătrân", Constanța, RO
[6]PhD Eng, Naval Academy "Mircea cel Bătrân", Constanța, RO

E-mail: m.vladmocanu@gmail.com

Abstract. This project is designed to simulate a transfer of oil liquids from an oil tanker. This practical project presents an easy method of realizing such a system with the observance of the safety conditions. The software that underlies the operation of the Arduino Mega development board used in the project is exposed.

1. Introduction

The tanker is a specialized vessel for the transport of crude oil and petroleum products. These products are loaded into special rooms of the ship, called tanks. The number of tanks varies depending on the size of the ship. Large oil tankers transport oil from the extraction site to the refineries and the smaller ones transport the refining products from the refineries to the distribution network.

The rapid increase in global fuel consumption and the long distances between extraction, crude oil processing, and consumption areas, have led to a considerable development of this type of ship, the trend being towards the construction of supertankers, reaching the capacity of load up to 400,000 tdw [1].

2. Technical details

To carry out this project, the Arduino Mega development board was used, which has the role of processing the values sent by the ultrasonic sensor HC-SR04, calculates and displays the liquid level in the service tank, and sends commands to the relay to start and stop the pumps, the simplified diagram is exposed in figure 1.

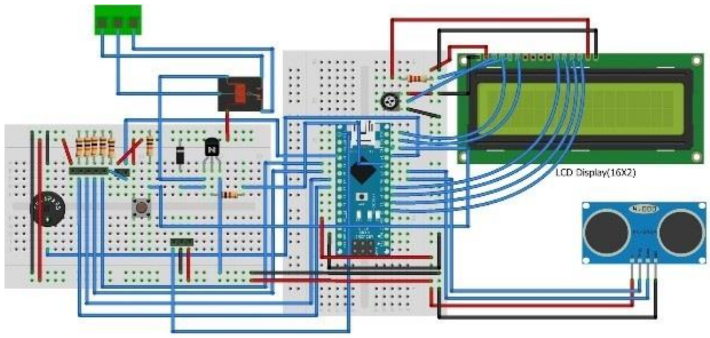

Figure 1. The simplified diagram of project.

The HC-SR04 ultrasonic sensor is used to measure the distance between 2 cm and 400 cm with an accuracy of 3 mm. The module consists of an ultrasound transmitter, receiver, and control circuit [2].

The ultrasonic sensor works on the natural phenomenon of the echo.

A pulse is sent by the transmitter for about 10 μs. Then the module automatically sends 8 cycles of 40 kHz ultrasound signal.

The signal returns to the receiver after hitting an obstacle. Thus, the distance of the obstacle from the sensor is calculated by the following formula, as exposed in figure 2.

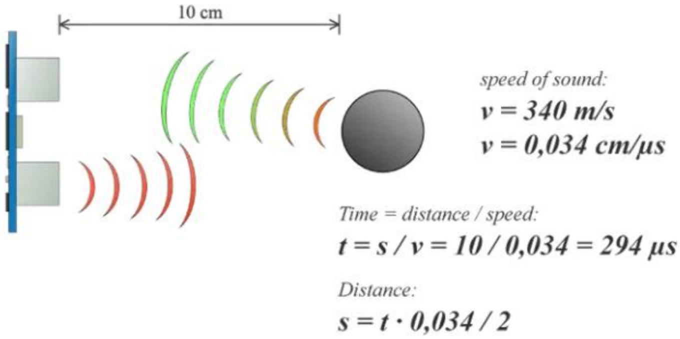

Figure 2. The distance calculation of ultrasonic sensor [2].

2.1 System operation

When tank 2 empties to 20% of its capacity, the pump is started, which transfers liquid from tank 1 and fills tank 2. When the liquid level in tank 2 reaches 90% of its working capacity, the pump stops. The pumps is powered by 3-12V d.c. and is controlled by the Arduino Mega development board via the control relays, as exposed in figure 3.

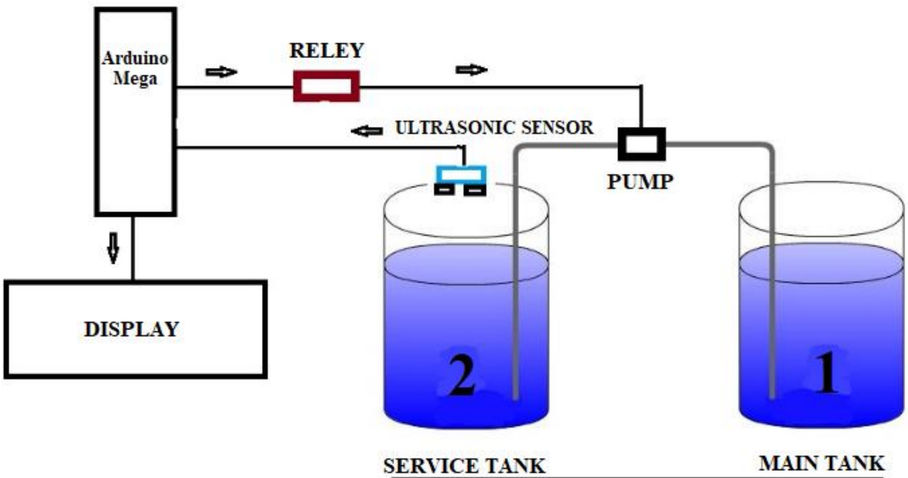

Figure 3. Working principle of liquid tank transfer.

2.2 Project software

The interpretation of the data transmitted by the Arduino development board is performed with the help of a program installed on the PC via a USB connection. For the programming of the Arduino Mega development board, the open-source software of the system was used, for which the following code was written:

```
#include <LiquidCrystal_I2C.h> // Includes the library used by I2C mode [4].
LiquidCrystal_I2C lcd = LiquidCrystal_I2C(0x27,20,4); // Dysplay address setting (0x27) and dysplay resolution (20.4)
#include <NewPing.h> // Includes the library used for distance measurement using the HC-SR 04 sonar sensor
#define TRIGGER_PIN  2  // Defines the receiver pin used
#define ECHO_PIN     3  // Defines the pin for the transmitter
#define MAX_DISTANCE 500 // Set the maximum distance in centimeters
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing configuration of pins and maximum distance
#define RELAYPIN 11// Arduino pin tied to Relaypin of the Relay Circuit
#define EXTRELAYPIN 12 // Arduino pin tied to vin pin of the External Relay Circuit
#define BUZZER 9 // Arduino pin tied to +ve terminal of the Buzzer
float val;
#include <EEPROM.h>
int addr = 0;
int addr2 = 1;
int flag;
byte readval;
#define buttonPin 8
int buttonState = 0;         // Variable for reading the status of the button
float TankHeight, MaxWaterLevel, EmptySpace, SonarReading, ActualReading,  Temp;
int percentage;
int SpmpSensorPin = A0;    // Set pin A0 as the minimum tank level sensor 1
int SpmpsensorValue = 0;  // Variable to store the value from the sensor
int counter = 0;
int readtank = 0;
int temppercentage = 0;
int errorcorrection = 0;
void setup()
{
  Serial.begin(9600);
  Serial.println(flag);
  lcd.begin();
  lcd.backlight();
  pinMode(RELAYPIN,OUTPUT);// Relay pin as output pin
  pinMode(EXTRELAYPIN,OUTPUT);// External Relay pin as output pin
  digitalWrite(RELAYPIN,LOW); //Turn off the relay
  digitalWrite(EXTRELAYPIN,LOW); //Turn off the external relay
  pinMode(BUZZER,OUTPUT);// Buzzer pin as output pin
  digitalWrite(BUZZER,LOW);  //Turn off the Buzzer
  lcd.print("   MONITORING    "); // Display the project name on display
  lcd.setCursor(0,1);
  lcd.print("   LEVEL FROM    ");
  lcd.setCursor(0,2);
  lcd.print("cargo tanks");
```

```
  lcd.setCursor(0,3);
  lcd.print("at an oil tanker");
  delay(5000);
  lcd.clear();
  lcd.print("Long press the button to trigger the level");
  delay(1000);
  delay(1000);
  lcd.clear();
  lcd.print("Tank 2 empty!");
  delay(1000);
  delay(1000);
  lcd.clear();
  lcd.print("The height is calculated");
                    for (int i=0; i<=5; i++)
                      {
                      lcd.setCursor(0,1);
                      lcd.print("  in: ");
                      lcd.print(5-i);
                      lcd.print("Seconds");
                      buttonState = digitalRead(buttonPin);
                           if (buttonState == HIGH)
                             {
                             TankHeight =sonar.ping_cm();
                             EEPROM.write(addr, TankHeight);
                             }
                      delay(1000);
                      }
  TankHeight= EEPROM.read(addr);
  lcd.clear();
  lcd.print("Memorized height:");
  lcd.setCursor(0,1);
  lcd.print("    ");
  lcd.print(TankHeight);
  lcd.print(" cm   ");
  delay(2000);
  lcd.clear();
  MaxWaterLevel=0.85*TankHeight;
  EmptySpace=TankHeight-MaxWaterLevel;
}
void loop()
{
  delay(50); // 100ms delay between pings (approximately 20 pings / sec). 29 ms should be the shortest
delay between pings.
  SonarReading=sonar.ping_cm();
  SpmpsensorValue=analogRead(SpmpSensorPin);
  Serial.println(SpmpsensorValue);
  Serial.println(percentage);
  Serial.println(readtank);
  Temp= SonarReading-EmptySpace;
  ActualReading= MaxWaterLevel-Temp;
```

```
 readtank=(ActualReading/MaxWaterLevel*100);          // Algorithm for correcting the read values.
temppercentage=(temppercentage+readtank)/2;
 counter=counter+1;
 if (counter = 100000000000 && temppercentage > 0)
   {
    errorcorrection=(temppercentage+percentage)/2;
    counter=1;
   }
  if ( errorcorrection<percentage+10 && errorcorrection > percentage-10 )
  {
   percentage = errorcorrection;
  }
 lcd.setCursor(0,0);
 lcd.print("   Nivel 2 :");
 lcd.print(percentage);
 lcd.print("%    ");
          if(SpmpsensorValue<=100)
           {
                    if(percentage<=90)
                         {
                         lcd.setCursor(0,1);
                         lcd.print("Pump on");
                         digitalWrite(RELAYPIN,HIGH);
                         digitalWrite(EXTRELAYPIN,LOW);
                         flag=1;
                         EEPROM.write(addr2, flag);
                         flag= EEPROM.read(addr2);
                         }
                else if(percentage>20 && percentage<=85)
                 {
                  flag= EEPROM.read(addr2);
                  if(percentage>20 && percentage<=85 && flag ==1)
                        {
                        digitalWrite(RELAYPIN,HIGH);
                        digitalWrite(EXTRELAYPIN,LOW);
                        lcd.setCursor(0,1);
                        lcd.print("Pump on");
                        }
               else if(percentage>20 && percentage<=85 && flag ==0)
                  {
                   digitalWrite(RELAYPIN,LOW);
                   digitalWrite(EXTRELAYPIN,LOW);
                   lcd.setCursor(0,1);
                   lcd.print("Drain pump stopped");
                   }
                 }
                else if(percentage>85)
                  {
                   delay(500);
                   lcd.setCursor(0,1);
                   lcd.print("Pump stopped");
```

```
                lcd.setCursor(0,0);
                lcd.print("   Nivel 2 :");
                lcd.print("100");
                lcd.print("%    ");
                digitalWrite(RELAYPIN,LOW);
                digitalWrite(EXTRELAYPIN,LOW);
                flag=0;
                EEPROM.write(addr2, flag);
                flag= EEPROM.read(addr2);
               }
      }
               else if(SpmpsensorValue>=100)
               {
                flag= EEPROM.read(addr2);
                if(flag==1)
                   {
                    lcd.clear();
                    lcd.setCursor(0,0);
                    lcd.print("*Tank 1 Goal!*");
                    lcd.setCursor(0,1);
                    lcd.print("*The pump is stopped!*");
                    digitalWrite(BUZZER,HIGH);
                    digitalWrite(RELAYPIN, LOW);
                    digitalWrite(EXTRELAYPIN, HIGH);
                   delay(100);
                    digitalWrite(BUZZER,LOW);
                    delay(100);
                   }
                   }

                 else if(flag==0)
                   {
                    lcd.setCursor(0,1);
                    lcd.print("Tank 1 empty!");
                    }
                }
```

3. Practical realization

For the practical realization, experiments and tests were performed in the laboratory with the help of a breadboard, as can be seen in the figure 4.

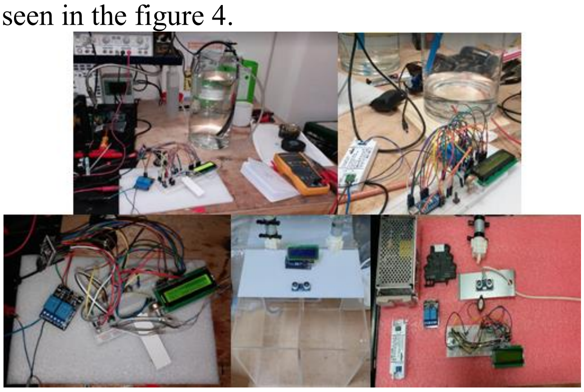

Figure 4. Tests and experiments during the project realization.

Following the exact establishment of the connections, the transfer installation of the two transfer tanks was performed, including the development plate, the relays, the ultrasonic sensor, the transfer pumps, the display, and the two transfer tanks made of plexiglass, as can be seen in figure 5.

Figure 5. Final configuration of project.

4. Conclusions

This project is a good basis for a fully automated oil tanker. Following the realization of this prototype for the transfer of liquids from oil transport tanks, we can conclude that an automatic tank loading system can be made with minimal costs. At the same time, this project can be easily adapted and used in other activities that involve the transfer of liquids from various tanks.

Referances

[1] https://zeymarine.com/largest-oil-tankers-ever-built/ - link accesed on 20.10.2021

[2] https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6 - link accesed on 22.10.2021

[3] https://create.arduino.cc/projecthub/Rushabh/fluid-level-indicator-and-controller-ca6773 - link accesed on 25.10.2021

[4] https://www.youtube.com/watch?v=0XvcTOI3-1Y - link accesed on 25.10.2021