

Major Project Document

by Daniel Lloyd Rees

FILE	21105_DANIEL_LLOYD_REES_MAJOR_PROJECT_DOCUMENT_78196_77981701.DOCX (16.05M)		
TIME SUBMITTED	21-APR-2016 11:47AM	WORD COUNT	17875
SUBMISSION ID	56188849	CHARACTER COUNT	92075



PRIFYSGOL CYMRU
Y Drindod Dewi Sant
UNIVERSITY OF WALES
Trinity Saint David

B-Eng Major Project

Underground Waste Water Tank Level System

Daniel Lloyd Rees : P144240

Supervisor: Gareth Jones

Second Marker: Gary Jones

April 2016

Project submitted as part of the requirements for the award

BEng Electrical and Electronic Engineering



DECLARATION

I, ... declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed :

Date :



FORM OF CONSENT

I hereby consent that my Project, submitted in candidature for the BEng Electrical and Electronic Engineering degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

Date :



Abstract

There are many methods of measuring the level of water or liquid in many different arrays of tanks or silos. With this knowledge being known, what is the most efficient way of measuring the level of liquid in a tank and also of controlling the level of water that is in the tank at any one time?

The aim of this project is to research the different types of level measurement that are available on the market for the uses that have been asked for by the company NPT Recycling. Following finding a suitable sensor for the system, there will then be the need to design an electronic and electrical system that can incorporate this method of tank level management. As the tank which requires the level control is an inaccessible underground tank, the methods of level control differ from those which could be used in a tank that can be accessed easily.

Using ultrasonic measurement techniques was chosen to be the easiest way of level measurement in the underground tank, but this method of measurement is not without its own downfalls. With ultrasonic sensors using sound waves there are a number of variables which can give back false or inaccurate readings such as different temperatures or humidity inside the tank and also if there is not a clear surface for the ultrasonic pulse to rebound from this can cause inaccurate readings also. Also industrial sensors for this type of application can be very costly for a small unit and simple level measurement.

The system is based around the use of a microcontroller controlling the system, and the microcontroller used is the PIC18F2455. This was determined to be able to control the system with the incorporation of an ultrasonic sensor, and be able to relay data back to operators and technicians alike. The code for the system is written using C++ language and



is used to display data back from the results to an LCD display and also through means of RS232 Communications.

Throughout testing of the design and system, it has been found that non-contact ultrasonic sensors are more than capable of completing distance measurement, and that the sensor which was chosen worked correctly and as was thought from research undertaken. Further tests will have to be carried out to be sure that it is this most efficient system for underwater tank level measurement and control.



Table of Contents

Abstract	4
Chapter 1: Introduction	8
1.1 Aim	8
1.2 Objectives.....	9
1.3 Structure plan	9
Chapter 2: Research	11
2.1 Research and literature review	11
Float Level Switches	14
Conductivity Sensor	15
2.2 Ultrasonic Sensors.....	16
2.3 Comparison of Sensors	18
2.4 Choice of sensor	21
Chapter 3: Development Methodology	23
3.1 Software Development	28
Chapter 4: Design & Implementation	30
4.1 Design.....	30
4.1.1 System Control.....	30
Microcontroller Specification	31
4.1.2 Display Specification	36
4.1.3 Ultrasonic Sensor Choice	36
4.1.4 Pump Control	39



4.1.5 Circuit Design	40
4.1.6 Programme Flow Chart	42
4.2 Implementation	43
4.2.1 Microcontroller and Ultrasonic Interfacing	45
4.2.2 LCD display with readings	49
4.2.3 Circuit Design, Programme and Simulation	53
4.2.4 System Build and Test	59
Chapter 5: Evaluation.....	69
Chapter 6: Conclusion & Recommendations	72
Chapter 7: Reflection	75
Table of Figures.....	76
References	78
Bibliography	80
Appendices.....	81



Chapter 1: Introduction

This project is aimed at designing and implementing a water level measurement system that will have the capabilities of automatic tank level control. This project is of a work based nature and is seen to be a much needed feature in the control of underground waste water tanks. At the present moment in time the current method of measuring the volume of water in the tank is by using analogue float switches. The design of the system at the present time is that there are three float switches within the underground tank. The first switch will active to show that you have a low level of water in the tank. The next is suspended near to the halfway point of the tank; this is to give an indication to an operator that the tank is half full. The final float switch is suspended 25" from the top of the tank to give you a Hi-Level alarm indication, showing that the tank is at its maximum capacity and the pump needs to be manually activated to lower the level in the tank. These float switches then in turn, light up a panel mounted LED on a control panel to alert the operator that the tank is at fill level 1, 2 or 3. These float switches do not automatically energise the pump in the tank to reduce the water level. There are a number of down sides to this method of underground tank monitoring such as, the floats becoming entwined with each other therefore false readings being given on the control panel. Also there is the problem of operator error, not checking the control panel to see if the tank is full, thus leading to the tank over flowing.

1.1 Aim

What is hoping to be achieved is the design and implementation of a system that will eliminate the chance of human error with regards to the tank over filling, and also have an overall more reliable measurement system that will give an exact amount, to within 1cm, of how much water is actually in the tank at any given time. The overall summary of the main aim to this project is to be able to design, build and implement a fully automated



system which will regulate the amount of waste water in the desired tank at all times. It is also wished that there will be a display for operators of the plant to see how much waste water is in the tank at any one time for readings and data logging purposes.

1.2 Objectives

The objectives of this project to be achieved in correspondence with the aims are as follows:

The system is aimed to be able to achieve the following operations:

1. Display a reading of how much water is in the tank at any one time on an LCD panel mounted display.
2. Once a set high level reading is reached, energise the control circuit for the pump so that the level in the tank can be reduced to an ample amount of water.
3. Log the amount of times in a 7 day week the pump circuit is energised.
4. Calculate the flow of water that has been pumped out of the tank over the 7 day week.

If all of these operations are achieved within this project and design, the overall reliability and detail of data, for the water measurement and pumping system will be of a much higher standard.

1.3 Structure plan

With the aims and objectives of the project set out and understood, next is to start researching what is out there at the moment, what can be implemented and what kind of systems are used for this type of problem within the industry. With adequate background research it will be possible to move onto the further stages of the project of the design and implementation sections and overall, achieve the solving of the problem that at the moment is so monotonously continuous.



There will be in depth research being carried out in the field of underground tank level measurement systems, with different systems being compared and pros and cons of each being stated. Following this research will be the decision on which type of system will be used to achieve the aims of the project. With this section being completed the design and implementation of the system can then begin. This will have its own section of research carried out to see, what is the best system that industry has utilised to make these systems work at a competitive and reliable rate.

Chapter 2: Research

2.1 Research and literature review

For there to be a full understanding of the field and subject of this major project there is an element of research that needs to be obtained from different sources and types of research. For this to be achieved successfully what is going to be applied is the so called method of the “research onion”. Below is a diagram of the research onion model that will be used.

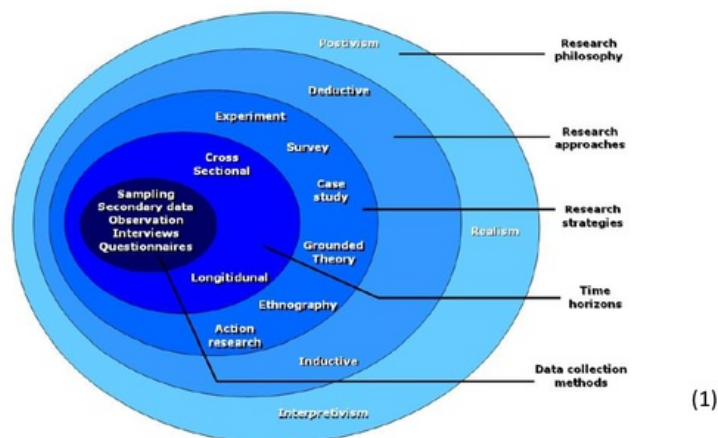


Figure 1 - Research onion

Research Philosophy: As this project has different elements to it such as its engineering foundation and also different types of engineering systems the research comes under positivist research section. This however is not where all of the project proposal and project research will be done. As this is at the beginning of the project, there will be research done into different fields of possibilities to achieve the aims.

Research Methods: For the research that is needed at this time and the nature of the subject being researched it will be that is secondary research being carried out. This will allow a better understanding of the systems and designs that are looking to be implicated



and also what type of systems are out there already. This will allow for comparison to other similar designs and proven working systems and how they work. This will also allow the insight into the most reliable way of tank measurement and level control in this system design. For this research there are different sources that will be used and different means of research and knowledge acquisition.

These types of sources are:

- Project area textbooks
- Project area specific journals and information guides
- Project area internet pages and information sources
- Project area consumer references.

When the research for the basics and understanding of the project area is completed there must be a different type of research brought into action, which will be primary research, and this will lead to research being done in such places as:

- Technical information for this medium of tank level control
- Design drawings for these types of systems
- Systems already in use, their outputs and what feedback is from them on how they react in this field of study.
- University lectures and how these types of systems work and components and hardware/software that may be able to be used.
- Component and hardware data sheets.



For the project to be able to be implemented, evaluated, checked and tested the knowledge from previous and current studies must be applied but also with the use of the primary research undertaken so that full understanding of the system is achieved. Due to the length of this project and it spanning over 18 months, the research falls under the longitudinal section of the "Research Onion". From company records and information stored in hardware on the plant, it can be determined what is the starting point of this project in the cross-sectional section, to where at the end, once all completed and the solution implemented, if there has been any change in the reliability of the tank level control system.

Within this section will be different types of tank level measurement researched for the problem of this major project. There are three different types of sensors that have been researched.

Float Level Switches

Float or Buoyancy switches come in many different shapes and sizes. This certain type of level indicator is very versatile in which applications of liquid level measurement they are intended for. The number of different designs that are on the market give these switches the ability to fit into almost any tank no matter the shape or the size. The design within this type of switch most commonly used is the simple utilization of a reed switch. This switch is activated by an embedded magnet within the float, which moves with the level of the liquid. This is most simply explained as, when the tank is at a low level, the float is at a vertical angle in respect to the tank. As the water level rises, the float will start to become more horizontal until the switch is at a made state. This explanation can be seen in the figures below.

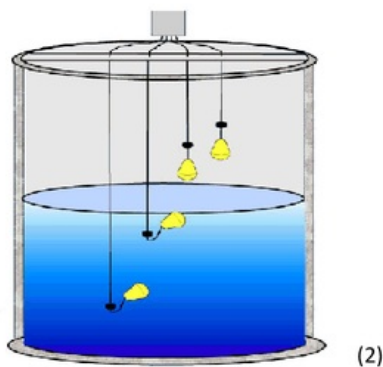


Figure 2 - Float level switch

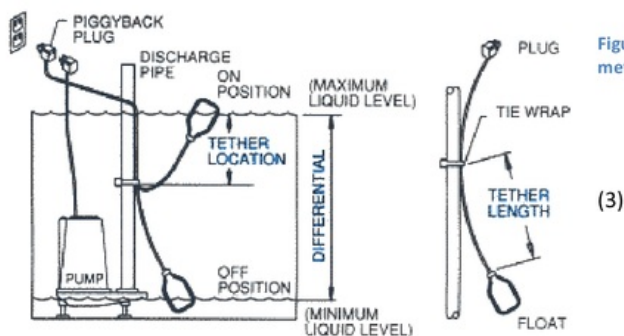


Figure 3 - Float level switch methodology

Conductivity Sensor

The conductivity sensor works by relying on the conductivity of the liquid it is there to measure. This sensor consists of two metallic probes which are extended down to a certain depth of the tank that is wished to be measured with the metallic probes being of different lengths. The probe which is longest will carry a low voltage and this also acts as a ground leg. The shorter length probe is cut at the tip at the actuation point. Once the tank fills so that both probes are in touching the liquid, the liquid acts as a conductor, and thus a switch effect is observed by the current flowing through the both probes. This type of

sensor will require an electronic controller for it to be used. Below is an image of a conductivity switch.

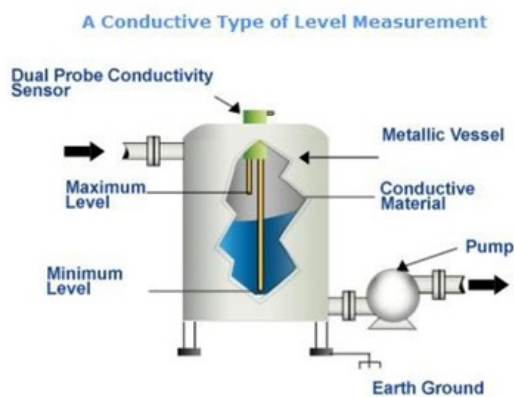


Figure 4 - conductivity sensor working

(4)

2.2 Ultrasonic Sensors

Ultrasonic sensors for distance measurement have been utilised for a number of years, with the one of the first uses being sonar. "It was employed on a large scale by the United States Navy during World War II to detect enemy submarines. Sonar operates by bouncing a series of high frequency, concentrated sound wave beams off a target and then recording the echo. Because the speed of sound in water is known, it is an easy matter to calculate the distance of the target." (5) As can be seen from this information it has been being used for 40 plus years for depth measurement. In industry at this present moment ultrasonic sensors come in a variety of times such as, contact, non-contact, invasive and non-invasive. No matter which type ultrasonic sensors are very accurate and can also be very small units.

Non-Contact Ultrasonic Sensors

Non-contact sensors use technology that converts electricity or electrical energy into acoustic energy. For measurement within a tank, the sensors send out an ultrasonic sound wave from the transducer, which in turn bounces off the water's surface back to a receiver. The level is determined by the length of time the 'Echo' signal takes to return back to the sensor. These sensors can be used for applications of continuous level sensing, or

hi/low level switching.



Figure 5 - Ultrasonic sensor

(6)

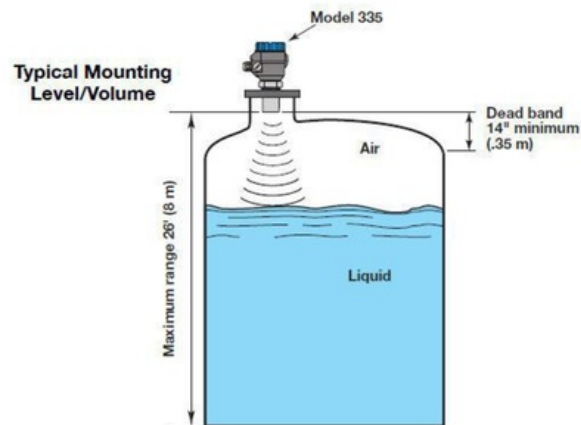


Figure 6 - Ultrasonic sensor theory and design

(7)

Contact Ultrasonic Sensor

Contact ultrasonic sensors use sound transmissions to detect the liquid within the tank. Within this version of sensor a transmitting crystal sends a sound wave to its counterpart receiving crystal. If there is a water presence between the two crystals then the signal will be greatly amplified. If this sensor is just in free space, the ultrasonic waves are attenuated greatly. Thus the circuitry within the sensor must then calculate and analyse the signal strength controlling the switch appropriately. Below is an example of a contact ultrasonic sensor.

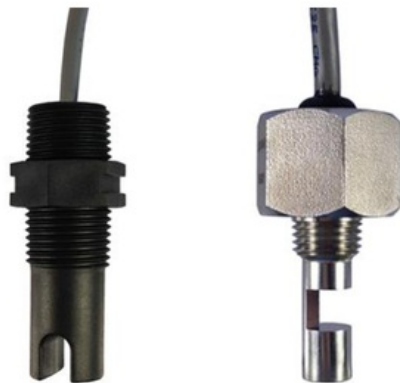


Figure 7 - Contact ultrasonic sensor

(8)



2.3 Comparison of Sensors

Within this section of the project will be advantages and disadvantages of each type of sensors from different sources and also from Knowledge that the author has obtained from the workplace and industry.

Advantages and Disadvantages of float level switch

Advantages:

- Non-powered unit.
- Reliable.
- Direct Indication.
- Relatively cheap unit.

Disadvantages:

- In-contact with fluids.
- Moving parts within switch.
- If there are multiple sensors within a tank, there is a tendency for sensors to become entwined with one and another. (Current problem that is trying to be overcome.)
- If it is only a small tank, water displacement can become an issue with the float being within the liquid.

As can be taken from this above information, if the overall aim of the project was just to highlight a Hi-level alarm that a float level switch would more than likely suffice. This method is the most basic of the three types of sensor researched. Although these units are relatively cheap, there are problems which associate with them such as corrosion with them being in contact with the liquid they are measuring. Also with the moving parts which are found within these switches, you can have component breakdown faults quite often, as has also been seen by the author numerous times.



Advantages and Disadvantages of Conductivity Sensors

Advantages:

- These Units have no moving parts.
- Low Cost.
- Easy to configure.
- Minimum amount of water displacement with the 'thin Rods'.

Disadvantages:

- Probe erosion.
- Measurement liquid must be conductive.
- Receiver entry hole can become blocked from dirt or other foreign objects within the tank.

With the conductivity sensor there are many advantages of using this design. Also there are many disadvantages to it also. These units are more reliable than the float switch as there is no moving parts within the sensor, yet the disadvantage to go along side this is that you will get probe erosion over time with the probes being submerged in the liquid constantly. Also, with these sensors the liquid it is measuring must be conductive for the circuit to be made. Another problem that can arise, all be it maybe not that often, is that of the receiver hole on the shorter rod can become blocked by dirty or foreign bodies. This is another disadvantage similar to that of the float switches becoming entangled with each other.



Advantages and Disadvantages of Ultrasonic Sensors

Advantages:

- No moving parts.
- Typically as small unit.
- Very accurate measurements.
- Can be used for tank depth and also as a switching device.
- Not immersed in liquid being measured, therefore minimal chance of corrosion.
- As said by 'Allen-Bradley' "An ultrasonic sensor's response is not dependent upon the surface colour or optical reflectivity of the object. For example, the sensing of a clear glass plate, a brown pottery plate, a white plastic plate, and a shiny aluminium plate is the same." (9)
- Very Robust units for use in areas where they could get damaged.
- Good immunity to background noise (9).
- Easily programmable with multiple output options.

Disadvantages:

- Can be fairly expensive to buy if their only for hi-level measurement.
- Require a power source for operation, for example a 24V power supply.
- Ultrasonic sensors do have a minimum sensing distance, which is a small distance from the sensor, which it cannot read the echo due to the transducer still sending out its pulse signal.
- Changes in environment, such as the temperature of a tank, pressure within the vessel, relative humidity, can affect the ultrasonic wave being transmitted back.
- Contact ultrasonic sensors need to be placed into the tank normally through the side wall, which for an underground tank can become obviously, very problematic.



- Contact sensors can also give false readings from air pockets or bubbles in the tank.

Finally, coming to the advantages and disadvantages of the ultrasonic sensor, there is again a number of each that has been found. Overall they are conveyed to the consumer as the best type of level measurement for within a tank. With them not being in contact with the liquid, there is the least chance of all of them breaking down with corrosion. Yet for a simple Hi-level measurement system, these sensors are very much overpriced and a matter of overkill. The ultrasonic sensors can also be used for numerous jobs such as level measurement, some as accurate as to within 0.5mm, with some having a sink / source option of switching for pump control within a tank. For this kind of application, it is clearly conveyed why they could be seen as the best, although the downside with this is that they need a power supply to run them and also can be quite an expensive unit.

2.4 Choice of sensor

After the research had been completed into what type of sensors there were already in place in the industry of tank level measurement, there was a decision to be made of which would be the best suited for the application at NPT Recycling. The final choice was to go with an ultrasonic sensor for the measurement of the underground tank. There was a number of factors that were considered before this choice was finalised upon. The pros and Cons of each type of sensor were taken into account and broken down rigidly to see what would be the best for the company, most reliable and most efficient for the level measurement, and be able to incorporate the pump level control.

As was mentioned within the pros and cons previously in this document, the ultrasonic sensors have a wide variety of different applications built into them. These are such things as, pure level measurement, the distance that the water level is away from the top of the tank. It was seen that the greatest asset found within the ultrasonic sensor



capabilities is the sink/source feature that some sensors are capable of for the running of a pump to mediate that level of water in this tank. Many types of ultrasonic sensors were looked at for this project, with each having their own different function, price range and even completely different designs all together. Also with the use of an ultrasonic sensor, it has been seen throughout the research that they can be incorporated into existing systems, or new systems. It is shown that the sensors, are seemingly easy to set up and that if they were needed to be incorporated into an existing system, they can be adapted to the application simply. The final choice of sensor being used will follow later in this document with the reasoning behind it fully explained.



Chapter 3: Development Methodology

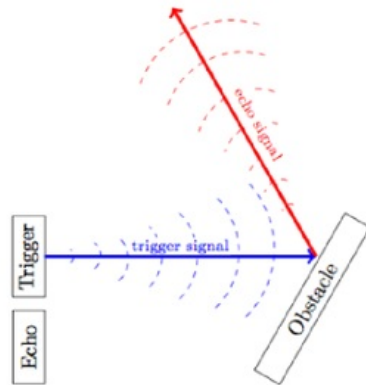
As has been mentioned in previous chapters, the overall aim of this project is to design, build and implement an accurate system for measuring the water level in a sealed underground tank which will also control the water level by an underground pump should the water level raise to high. The aim is to build a system that incorporates the use of an ultrasonic sensor, so that the level can be measured and the system can be fully automatic without the need of operators to be physically emptying the tank and manual control of the submersed pump. Also being developed will be the electrical control side of the pump, with the use of a control panel, and the electrical components needed for motor or pump control.

With the decision being made to use an ultrasonic sensor, there must be good reasoning for it to be finalised on. There are two main types of ultrasonic sensor, you have Non-contact ultrasonic sensors and then you have contact ultrasonic sensors. These both types of sensor are a good choice for the level within a tank and are both very accurate in the “perfect world” should we say. For example, the contact ultrasonic sensor will tell you when the water level has reached where it is situated within the tank as the water will amplify the ultrasonic burst that is being sent between the two crystals. The problem that NPT Recycling face is that the tank wished to be measured is underground and sealed under a re-enforced concrete floor, thus the contact ultrasonic sensors would only give a reading of the full high level. This leads to this method of Contact ultrasonic measurement being impractical and nearly impossible as the sensor can only be situated at the top and cannot be placed any lower down the tank wall.

As for the Non-contact system for ultrasonic measurement, this is much more practical for the application it is being used for at NPT Recycling. This is because of the way that the ultrasonic sensor works, with sending out an ultrasonic burst, whilst then waiting

for the return echo which will determine the distance that the signal has travelled. As the sensor will be mounted outside of the tank, it will be directly above the tank floor, meaning that the full ultrasonic wave will be returned from the water's surface back to the sensor.

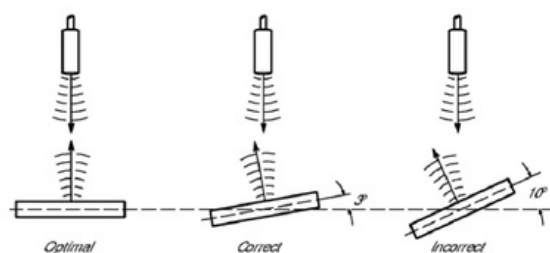
This is shown in the figure below:



(10)

Figure 8 - Ultrasonic signal

As you can see in the above figure, if the sensor is not directly opposite the object it is trying to detect, the ultrasonic burst can be deflected away from the receiver which reads the echo. This will lead to you not having a reading outputted from the sensor itself, or if it is just slightly out of line, a small section of the return echo can be lost making the ultrasonic burst weaker. Below is figure 9 showing how when the sensor is completely opposite the object it wants to detect, you will receive the full echo signal back to the receiver.



As you can see here, the optimal position for the sensor to be positioned is directly opposite. (11)



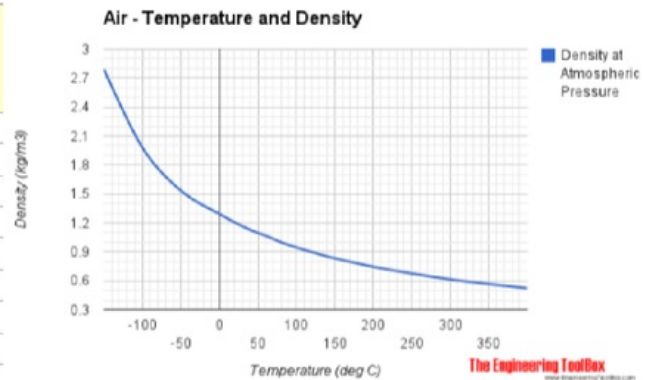
Figure 9 - Optimum sensor position

Also ultrasonic sensors can be used well for distance detection as sound, the ultrasonic burst, has a linear response to distance. This means that it can be used for accurate distance measurement with the correct formula's being implemented.

Although that the contactless method has been the chosen as the design of ultrasonic sensor to be used for this level measuring system, it does not come without its disadvantages. These disadvantages consist of such problems as, simply the sensor itself becoming dirty. This can cause, in the worst case, serious problems with the transmission and receiving of the ultrasonic burst if a cleaning schedule is not maintained. There is also the problem in some instances of the need for a separate power supply to the sensor itself as most industrial ultrasonic sensors seem to run at 10-30 VDC. This can be overcome by using a 12VDC battery in the design, but if what is wanted is a permanent fixture that is to be running around the clock, this is not always the best practice for this type of application.

Another problem that can arise is the effect of temperature differentiation alongside with humidity of the area where you are measuring and the effect these two outside variables can have on the ultrasonic burst. As you can see in the below table and graph, this is the effect temperature has on air density. Simply, as air becomes cooler, its density becomes greater and vice versa. These fluctuations can cause a change in the time the ultrasonic burst echo will return back to the sensor.

Temperature - t - (°C)	Density - ρ - (kg/m³)
-150	2.793
-100	1.980
-50	1.534
0	1.293
20	1.205
40	1.127
60	1.067
80	1.000
100	0.946



(12)

(12)

Figure 10 - Air temperature to density table and graph

As with humidity, as the humidity of the air increases, the air becomes less dense, thus the ultrasonic sound speed through air will change.

“Lower density translates into faster sound wave travel, so sound waves travel faster at high humidity. The increase in speed, however, is very small, so for most everyday purposes you can ignore it. In room temperature air at sea level, for example, sound travels about 0.35 percent faster in 100 percent humidity (very humid air) than it does in 0 percent humidity (completely dry air).” (13)

To solve this possible problem, the temperature and humidity inside of the tank can be measured, and then the results can be utilised and implemented as a co-efficient in the

$$Speed = \frac{Distance}{Time}$$

Equation for the most accurate reading.

For powering the system there was a number of possibilities that were taken into account as the system will be based primarily on site at NPT recycling, where there are no restrictions in what power supply could be used for the system. It was decided that a battery wouldn't suffice for this system, although it would make the system more portable, it would mean that the supply would have to be checked more often to make sure the battery



had not run flat. With this system being in place and running constantly a battery just was not deemed feasible.

Next was the use of an 110V to 24V transformer based inside the control panel. This seemed to be a good idea, but then there was the problem of the efficiency of the control in the circuit. With the 24V being an ideal power supply for the running of the pump contactor also it did seem like the best power supply. The problem then arose that, taking the 24VDC down to 5VDC for the PIC using a voltage regulator, would take the efficiency of the circuit down to a very low level and there would be major losses in heat dissipation at the regulator. This method of power supply also puts the restriction on the system that it can only be run in a vicinity that can cater for 110V, meaning either extra transformers or hardwire 110V into the system.

The final decision was made that, within the enclosure, there would be a 230V to 5V transformer mounted, and this would be the supply for the system. Even though the step down from 230v to 5v is much greater than 25VDC to 5VDC, the transformer is a much more efficient way of stepping down the voltage than the use of the voltage regulator. As the supply to the system is 230V, this gives the system the opportunity to be set up anywhere on the site of NPT recycling where there is access to a 230V supply. As this is the standard voltage in the United Kingdom, the system could be utilised for many more jobs rather than just the level and pump control of the underground tank.



3.1 Software Development

For this system design, it has been decided that a microcontroller, or ' μC ', will be used. This means that there is software that needs to be utilised to be able to programme the PIC microcontroller. There are many different types of software out there for this activity, but it was decided that CCS C would be used. CCS C is a very powerful and also exceedingly helpful piece of software. It is a type of compiling software, which works with C++ coding language.

This software is a very useful tool, as it has an outstanding project wizard function, which you can set up any analogue inputs or out puts, timers, clocks, communication ports, LCD incorporation and many more tools. It also has a very good help function which allows the user to search for the function they want to complete, and will display to the user examples of how that function can be used. As the programme uses C++ language, it makes coding much simpler as it is much better for wording the code rather than using normal C language, which will be shown in example code later in the document. This is a brilliant piece of software for a beginner at programme writing with all of its example code and help files, but is also powerful enough for the most experienced of coders to be able to use.

Also included in the development of the software is the software, Proteus. This is a piece of software that will be utilised for the simulation of the system. Also in this system, the PIC can be set up, and the programme written in here. As this software supports CCS C as a compiler, it is a very useful and powerful programme. With Proteus, the code can be written straight into the 'Main.C' file, and then compiled to show if there are any errors with the code that has been produced. If this compiles successfully utilising the CCS C software, the system can be simulated and tested to see what piece of the software is doing what, if there are signals being set and received, if your "Tell Tales" are working and how the hardware will be connected to the microcontroller using what pins.



For the actual programming of the PIC, writing the code that has been achieved onto the PIC, the software that will be used is called Pickit 3. This software is developed by MP Lab, which is part of the company Microchip. The Pickit 3 software has the ability to write to and erase from a PIC, so it gives you the opportunity to easily make changes in the code, should there be mistakes found or if there is added functions to written. The programming of the chip is completed by using the Pickit 3 programmer connector, which will plug into the laptop or computer of the person who would wish to programme the chip. Then by using the simple steps of the software, in a short space of time your PIC microcontroller is ready to be included in the circuit fully loaded with the code that has been written to it.

The PIC μ C that was decided on to use was the PIC18F2455 chip. With regards to what is needed for this design the PIC18F2455 has more than enough capabilities to run the system and complete the tasks that are needed of it. It has enough memory for the programme, and will run on an internal oscillator by far fast enough to complete its commands throughout the running of the design.



Chapter 4: Design & Implementation

4.1 Design

4.1.1 System Control

The aim to be achieved by this system is to allow NPT Recycling, the freedom and ease, of having automated control of the level of water in an underground tank. With this freedom it allows the company to be able to have a reliable system in place and not have to rely on operators running the pump manually, possibly forgetting, or worst case running the pump when there is no water within the tank. For this system to be able to work there was a need for a choice of control that was going to be used. As it was seen the company had two choices, incorporate a PLC into the system and control it via this method, or use a microcontroller.

It was decided upon that a microcontroller would be used for this system, although that there was no issue with space or size of a PLC, NPT Recycling wanted this as a trial of pump automation and did not want to spend an extortionate amount of money on PLC's and software in the case of there not being a great deal of difference in efficiency of controlling the tank level. As a microcontroller was seen to have the capability to be able to run a system of this nature, it was chosen as it was the most feasible method of control. Also the choice was made for using a microcontroller as there was software available to the writer and, components that would be used within the circuit and system would be relatively cheap compared to that of the use of a PLC.

For the choice of the actual microcontroller that was going to be utilised for this system, knowledge gained from the university pointed towards using a PIC microcontroller. With the software that was to be used and for ease of programming the PIC was most certainly the best choice of μC . All microcontroller manufacturers have their own different



language that the code must be written in and C++ is the most user friendly, especially with the assistance from the software that it was to be written on.

Previous to being able to make a decision on which PIC would be used, there are a number of variables, which must be taken into account for the correct one to be chosen. Below is a small list of basic needs that must be taken into account before the final choice can be made:

- Does the PIC need a certain No. of A-D converters and at what resolution should these be?
- Number of pins needed for the running of the system and its components.
- Does the PIC need an external crystal oscillator or would the internal oscillator run and a great enough speed?
- Does the System require the PIC to have internal timers?
- How much programme memory is going to be needed for this system?
- Is there going to be data stored from the system? If so has the PIC got the capabilities to offer this storage?

As you can see there are many things that must be taken into account before the correct PIC microcontroller can be finalised on.

Microcontroller Specification

Firstly, for the choice of the microcontroller, it was needed to be determined whether any A-D converters were needed. Whilst researching different types of ultrasonic sensors, there were many different types of signal that were being sent back to the PIC as the echo. The majority of these ultrasonic sensors were found to give a linear 0-5V signal back as the echo in accordance to the distance and time the pulse had travelled for. There it was seen that at least a minimum of 1 A-D converter, 10-bit resolution was needed. As this is the only input to the PIC, only one A-D channel needed to be considered.



The next decision that had to be made was the number of pins that would be needed on the microcontroller for all of the components of the circuit and system. With looking at the circuit design and what components were going to be needed in the system, it was seen that the microcontroller to be used would need a minimum of 13 Pins, without counting for VDD and VSS. The components to be connected to the PIC and their individual pin count needs are shown in the table below.

Component	Pin Count
Trigger Pin Ultrasonic	1
Echo Pin Ultrasonic	1
Flashing LED 'Tell-Tale'	1
Control signal for pump contactor	1
LCD Display	7
RS232 communications	2
Total Minimum Pin Count	13

Figure 11- Pin count needed

So after looking at the minimum pin count of 13, there must be sufficient overhead given for any future modifications to the system and design, so a minimum of 20 Pins was decided upon. The next step was to find out if the PIC has an internal oscillator, and if so what speed would be enough for the running of the system. If there was not a PIC that could cope with the system, what size external crystal would need to be used? The final clock speed that was chosen to be used and needed was 8MHz. This speed was chosen due to a small number of calculations being carried out, which follow below:

- 30 instructions per reading
- 14470 readings in 1 hour
- 4 readings per second
- $30 \text{ instructions/reading} = 30 \times 4 = 120$
- $4 \text{ Clock pulses per instruction} = 120 \times 4 = 480$
- Therefore, a total of 480/S
- Overhead taken into account and clock speeds looked at, 8MHZ will be more than suffice.

These calculations were made and chosen after what had been learnt and picked up by the author whilst in university and also from what had been used in previous system designs.



Will the system require the microcontroller to have any number of internal timers that would need to be utilised for the running of the programme? Due to the nature of the system and the components that were being used, it was decided that there would surely need to be at-least one internal timer on the PIC. This was said because of the nature of the ultrasonic sensor. If the sensor did not detect a water level in the tank, or the sensor was faulty, the timer would time out after a set time period and return an error code to the LCD screen and via the RS232 Communications port. Without this timer and without it being set up properly, if the PIC did not receive an ECHO signal from the sensor it would get to a certain point of the programme code and just freeze and sit in limbo indefinitely waiting for the ECHO signal. In simple terms, without the timer incorporated, the system would just stall.

The next decision that had to be made before a specific microcontroller could be chosen was that of how much programme memory would the PIC need to be able to store all of the code for this system? For the full coding of the system, previous systems were taken into account and the amount of programme memory decided upon them was looked at. With the amount of code needed for this system and the number of steps that would be needed for it to complete its full cycle, an educated estimate of 4K was chosen. To allow for sufficient overhead and make sure there was enough room for further additions to the code, 16K was settled on as the final amount of programme memory needed for this system.

Another selection and decision that had to be made was that if there was to be any data going to be stored either on the chip or externally, has the chip got the capabilities to be able to make this possible. For this system it was decided that there wouldn't be data stores as such, but rather the data would be outputted to an LCD screen and also to virtual terminal software such as "Term Term" or "Putty" via RS232. If in the future the system

was wanted to be modified to store data, there are ways and means by using an SD card which would require the PIC to be able to offer an SPI mode, or via USB which would mean the PIC would have to be able to support a USB mode.

Once all of these variables were considered for the system, then came to the actual decision of which microcontroller would be used for the system. As now all of the major needs of the system had been defined, the microchip sites search engine was utilised to finalise on the choice of PIC to be used. Below is a screen shot of the starting point of the search engine and what choices of variables the user has to edit and finalise on their desired μ C. Also finally it was desired that the specified PIC would run at an operating voltage of 5V for ease of circuit design and to minimise the amount of components on the PCB and circuit design.

Figure 12 - Microchip selection

ly/Core	Program Memory Size (Kbytes)	Program Memory Size (KWords)	Bootload capable	RAM (bytes)	EEPROM / HEF	I/O Pins	Pin count	Max. CPU Speed MHz	Peripheral Pin Select (PPS)	Vbat for battery backup	Brown-Out Reset (BOR)	Low Voltage Detection	Power On Reset (POR)	Int Os
8-bit	0.125 0.250 0.375 0.500 0.750 1.000 1.500 2.000 3.000	0.125 0.250 0.375 0.500 0.750 1.000 1.500 2.000 3.000	No Yes	16 32 64 128 256 512 1024 2048 4096	0 128 256 512 1024 2048 4096 8192 16384	4 6 8 10 12 14 16 18 20	4 6 8 10 12 14 16 18 20	4 8 16 32 64 128 256 512 1024	No Yes	No Yes	None Programmable (BOR)	None Programmable (LVD)	Yes No	4 MHz 8 MHz 16 MHz 32 MHz 64 MHz 128 MHz 256 MHz 512 MHz 1024 MHz
8-bit	0.375	0.25	No	16	0	4	6	4	No	No	None	None	Yes	4 MHz

(14)

As you can see these are not all of the choices of variable available to the user, but it has a very simple set up, and makes the search for the ideal microcontroller much easier. Below is a screen shot of the PIC that was decided on for the running of this system once all of the variables mentioned previously were taken into account and entered into the search.

Product	Buy	Status	Documents	5K Pricing	Family/Core	Program Memory Size (Kbytes)	Program Memory Size (KWords)	Bootload capable	RAM (bytes)	EEPROM / HEF	I/O Pins	Pin count
[Reset] PIC18F2550 PIC18F2551 PIC18F2552 PIC18F2553 PIC18F2554 PIC18F2555 PIC18F2556 PIC18F2557 PIC18F2558 PIC18F2559 PIC18F2560 PIC18F2561 PIC18F2562 PIC18F2563 PIC18F2564 PIC18F2565 PIC18F2566 PIC18F2567 PIC18F2568 PIC18F2569 PIC18F2570 PIC18F2571 PIC18F2572 PIC18F2573 PIC18F2574 PIC18F2575 PIC18F2576 PIC18F2577 PIC18F2578 PIC18F2579 PIC18F2580 PIC18F2581 PIC18F2582 PIC18F2583 PIC18F2584 PIC18F2585 PIC18F2586 PIC18F2587 PIC18F2588 PIC18F2589 PIC18F2590 PIC18F2591 PIC18F2592 PIC18F2593 PIC18F2594 PIC18F2595 PIC18F2596 PIC18F2597 PIC18F2598 PIC18F2599 PIC18F2600 PIC18F2601 PIC18F2602 PIC18F2603 PIC18F2604 PIC18F2605 PIC18F2606 PIC18F2607 PIC18F2608 PIC18F2609 PIC18F2610 PIC18F2611 PIC18F2612 PIC18F2613 PIC18F2614 PIC18F2615 PIC18F2616 PIC18F2617 PIC18F2618 PIC18F2619 PIC18F2620 PIC18F2621 PIC18F2622 PIC18F2623 PIC18F2624 PIC18F2625 PIC18F2626 PIC18F2627 PIC18F2628 PIC18F2629 PIC18F2630 PIC18F2631 PIC18F2632 PIC18F2633 PIC18F2634 PIC18F2635 PIC18F2636 PIC18F2637 PIC18F2638 PIC18F2639 PIC18F2640 PIC18F2641 PIC18F2642 PIC18F2643 PIC18F2644 PIC18F2645 PIC18F2646 PIC18F2647 PIC18F2648 PIC18F2649 PIC18F2650 PIC18F2651 PIC18F2652 PIC18F2653 PIC18F2654 PIC18F2655 PIC18F2656 PIC18F2657 PIC18F2658 PIC18F2659 PIC18F2660 PIC18F2661 PIC18F2662 PIC18F2663 PIC18F2664 PIC18F2665 PIC18F2666 PIC18F2667 PIC18F2668 PIC18F2669 PIC18F2670 PIC18F2671 PIC18F2672 PIC18F2673 PIC18F2674 PIC18F2675 PIC18F2676 PIC18F2677 PIC18F2678 PIC18F2679 PIC18F2680 PIC18F2681 PIC18F2682 PIC18F2683 PIC18F2684 PIC18F2685 PIC18F2686 PIC18F2687 PIC18F2688 PIC18F2689 PIC18F2690 PIC18F2691 PIC18F2692 PIC18F2693 PIC18F2694 PIC18F2695 PIC18F2696 PIC18F2697 PIC18F2698 PIC18F2699 PIC18F2700 PIC18F2701 PIC18F2702 PIC18F2703 PIC18F2704 PIC18F2705 PIC18F2706 PIC18F2707 PIC18F2708 PIC18F2709 PIC18F2710 PIC18F2711 PIC18F2712 PIC18F2713 PIC18F2714 PIC18F2715 PIC18F2716 PIC18F2717 PIC18F2718 PIC18F2719 PIC18F2720 PIC18F2721 PIC18F2722 PIC18F2723 PIC18F2724 PIC18F2725 PIC18F2726 PIC18F2727 PIC18F2728 PIC18F2729 PIC18F2730 PIC18F2731 PIC18F2732 PIC18F2733 PIC18F2734 PIC18F2735 PIC18F2736 PIC18F2737 PIC18F2738 PIC18F2739 PIC18F2740 PIC18F2741 PIC18F2742 PIC18F2743 PIC18F2744 PIC18F2745 PIC18F2746 PIC18F2747 PIC18F2748 PIC18F2749 PIC18F2750 PIC18F2751 PIC18F2752 PIC18F2753 PIC18F2754 PIC18F2755 PIC18F2756 PIC18F2757 PIC18F2758 PIC18F2759 PIC18F2760 PIC18F2761 PIC18F2762 PIC18F2763 PIC18F2764 PIC18F2765 PIC18F2766 PIC18F2767 PIC18F2768 PIC18F2769 PIC18F2770 PIC18F2771 PIC18F2772 PIC18F2773 PIC18F2774 PIC18F2775 PIC18F2776 PIC18F2777 PIC18F2778 PIC18F2779 PIC18F2780 PIC18F2781 PIC18F2782 PIC18F2783 PIC18F2784 PIC18F2785 PIC18F2786 PIC18F2787 PIC18F2788 PIC18F2789 PIC18F2790 PIC18F2791 PIC18F2792 PIC18F2793 PIC18F2794 PIC18F2795 PIC18F2796 PIC18F2797 PIC18F2798 PIC18F2799 PIC18F2800 PIC18F2801 PIC18F2802 PIC18F2803 PIC18F2804 PIC18F2805 PIC18F2806 PIC18F2807 PIC18F2808 PIC18F2809 PIC18F2810 PIC18F2811 PIC18F2812 PIC18F2813 PIC18F2814 PIC18F2815 PIC18F2816 PIC18F2817 PIC18F2818 PIC18F2819 PIC18F2820 PIC18F2821 PIC18F2822 PIC18F2823 PIC18F2824 PIC18F2825 PIC18F2826 PIC18F2827 PIC18F2828 PIC18F2829 PIC18F2830 PIC18F2831 PIC18F2832 PIC18F2833 PIC18F2834 PIC18F2835 PIC18F2836 PIC18F2837 PIC18F2838 PIC18F2839 PIC18F2840 PIC18F2841 PIC18F2842 PIC18F2843 PIC18F2844 PIC18F2845 PIC18F2846 PIC18F2847 PIC18F2848 PIC18F2849 PIC18F2850 PIC18F2851 PIC18F2852 PIC18F2853 PIC18F2854 PIC18												

The final choice was to use the PIC18F2455 Microcontroller as this PIC met all of the needs of the system, and had a great quantity of overhead if the system was ever wanted to be expanded in the future without the need for having to purchase and redesign a new PIC and system. Below is a comparison table showing what are the minimum needs of this system, compared to what the 18F2455 will offer.

Variable	Needed	PIC18F2455
A/D Channels	1	10
A/D Resolution	12 Bit	10 Bit
Clock Speed	480/s Decided 8MHZ	8MHz / 32kHz
USB	Yes	Yes
SPI	Yes	Yes
Programme Memory	16K	24K
RAM	64Bytes	2048 Bytes
Timers	Minimum of 1 x 8-bit	1 x 8-bit, 3 x 16 bit
No. of Pins	20 Pins	28 Pins
Desired Operating Voltage	5V	2 – 5.5V

As can clearly be seen in the about table, the PIC18F2455 will be more than capable of handling this system and has an excess amount of overhead in most areas where the system can happily be adjusted and added to. This choice of microcontroller also gives the user the capability, if wanted at a later date, of logging the data that it creates as there is an SPI mode available and also a USB interface if they are wished to be utilised.



4.1.2 Display Specification

For the design of this system, what was wanted by NPT Recycling was a system that could read, control and display the level of water in the underground tank. As for the display, there were many ways of displaying the level in the tank such as the utilisation of an array of LED's, a constant alarm being sounded if the pump was running, or the use of an LCD display. After discussing with the management at NPT recycling, it was decided that a 16x2 LCD display would be used. This decision was made because, not only did they want to see the level of the tank, but how many times that pump had run over their desired period of time be it one day or one month. Also agreed was that for an operator or a technician to be able to see the pump running, a panel mount light would be used.

The final choice of LCD display that was to be used was the Display-Tech 162B display, which was readily available from online retailer's RS for next day dispatch. This display would then update for the operators of NPT recycling after every cycle of the programme. This gives the operators an up to date reading of the tank level and No. of pump counts since the system was initiated. This information will then be logged on a log sheet and given to a member of the management team.

4.1.3 Ultrasonic Sensor Choice

For the choice of ultrasonic sensor, there were many different types, designs and companies looked at before the final decision was settled upon. As was mentioned in the research, firstly what had to be decided was the type of ultrasonic sensor which would be used for this system and fit in with what NPT Recycling's needs were. There was the choice of contact or non-contact sensors to be considered and which was the best choice for the type of system it was to be implemented into. After lots of discussion and looking at research of what was on the market and which type was best for the measurement of a



tank level in an inaccessible tank it was decided on that, the design of sensor which would be implemented was to be the Non-Contact sensors.

Once this decision had been made, next was to decide what design of ultrasonic sensor would be used and what the specifications that the sensor had to meet were. What was needed that a sensor that had the capability of measuring the distance of up to 3 meters, and be able to incorporate the output from the sensor into the microcontroller system with ease. There were many sensors found, and below is a comparison table of a number of sensors that were considered.

Sensor	Distance	Output	Price
microsonic mic+340/D/TC	3,400 mm (3.4 m)	PNP	£128.44
ToughSonic 30 Ultrasonic	6.1 m	0-10VDC / 4-20mA	£450.67
PVC ToughSonic 50 Ultrasonic	10 m	0-10VDC / 4-20mA	£589.56
Rockwell Ultrasonic 18mm Proximity Sensor	1500mm	PNP	£324.52
HC-SR04 Ultrasonic Module	Up to 4m	5VDC	£1.45

Figure 15 - Comparison of ultrasonic sensors

As can be seen clearly from the above table, if the choice to go for an ultrasonic sensor which would be capable of being left in an industrial condition, the price of sensors is well over the £100 mark. There are cheaper available industrial sensors available on the market, but as the price of the unit comes down so does the distance that the sensors can work at decreases. With this system being implemented at this moment in time purely as a trial to see if they system does make a difference to the efficiency of the measuring and control of the tank, NPT recycling was not prepared to be spending an extortionate amount of money on just one sensor. It was decided to try and find a cheap alternative to the industrial style sensors that would be able to handle the requirements of this system.

This is when the ultrasonic distance measurement modules were discovered, specifically the HC-SR04 module. As this sensor is designed firstly for the use in robotics operations and distance measurement in that field, it has the ability to be incorporated into a microcontroller based system very easily, and without the need for massive adjustments. The module itself needs a 5VDC supply to run it, which is ideal for this system as that is what the microcontroller itself is running at. It requires the PIC to send it a $10\mu\text{s}$ pulse to initiate the modules internal controls for it so send its PING from the module to the surface it is measuring. It then waits to receive an echo back to the receiver on the module, which in turn sends a 5V signal back to the PIC, which varies in time depending on the distance it has measured. This is a compact unit which does not require much space, is extremely cheap to acquire, and is easy to incorporate into this system.



Figure 16 - HC-SR04 Module

(16)

Below is a small table showing what specifications the HC-SR04 module has:

Sensor	Input Voltage	Output Type	No. of pins needed on PIC	Measurement Distance	Price
HC-SR04	5VDC	5vDC Pulse Width Variable	2	2cm – 400cm With accuracy to 3mm	£1.45

Figure 17 - Specification of HC-SR04 module



It can be seen clearly from the above table that this module will be more than capable of doing what is needed for this system, and has very easy incorporation in to circuitry of the system as it works at 5VDC. It can also be incorporated into the PIC easily, although it doesn't give a simple 0-5V DC analogue signal back to the microcontroller, it sends a High Level Pulse from the echo pin back to the microcontroller which is directly in proportion to the time it takes for the echo signal to be received. This then can be written into the programme using a timer to calculate the distance and level of the underground tank. As has been mentioned previously, with this only being a trial system with the opportunity to maybe extended it to a full time system, this ultrasonic ranging module is ideal for now.

4.1.4 Pump Control

For the control of the pump, a control panel will be built and implemented so that it can be controlled by an output from the microcontroller when a Hi-Level is reached. As the pump is a 3-phase pump, the panel will require a supply taken from a nearby distribution board local to where the tank is situated. The panel will include a 3-phase isolator, which has the ability to be locked off should any work need to be done on the system, a 4-10amp overload as the pump is rated to run at a maximum of 8amps and finally a 3-phase contactor which will have a 5VDC coil in it so it can be directly energised from an output of the microcontroller. Also in this panel will be a single pole MCB which will supply 24V to a safety and or manual control push button circuit should the system fail and the pump need to be activated manually. Within the panel, for the control of the pump manually, will be a 24V to 5V transformer so that the 5V coil of the contactor can be used.

4.1.5 Circuit Design

For the design of the circuit, the Proteus Professional software was the chosen method. This was decided upon because of the fact, as certain parts of the design are built, it can be simulated to see if it will work correctly. This is good for such things as the LCD display and also for making sure that the PIC can talk to an RS232 output, and that the code that has been used in the software is correct. This powerful software gives the opportunity that parts of the code and circuitry are written and connected properly, before actually building anything in hardware on such things as a bread board or PCB. Below is a screen shot of this final circuit design that was completed on Proteus, which was the working simulation of the system that had been designed.

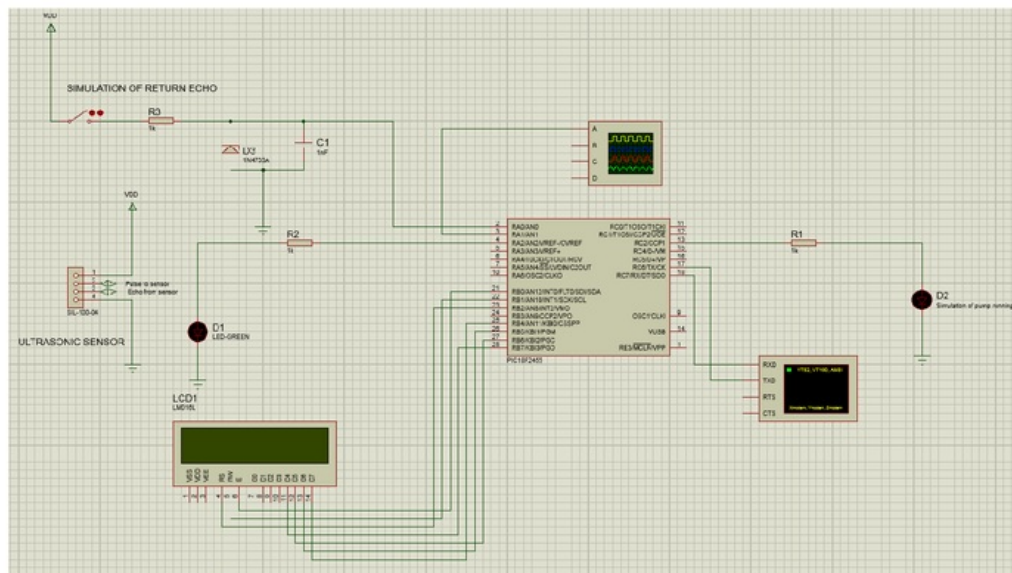


Figure 18 - Screen shot of system design

4.1.5.1 System Design Block Diagram

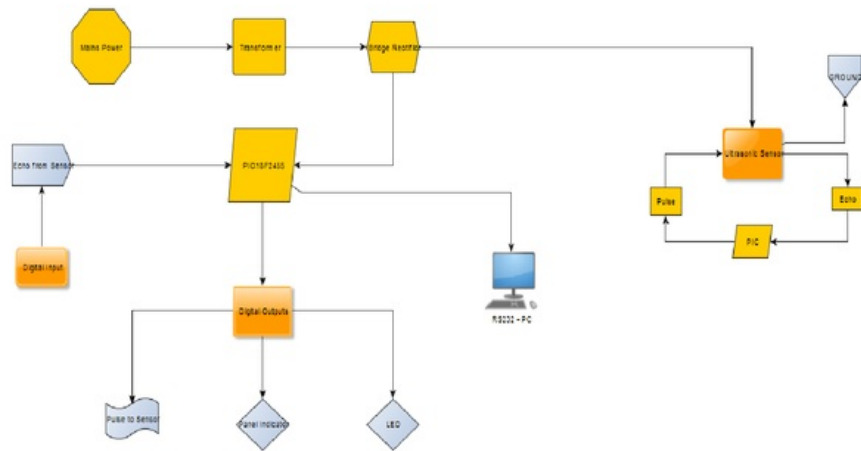


Figure 19 - System Block Diagram

4.1.6 Programme Flow Chart

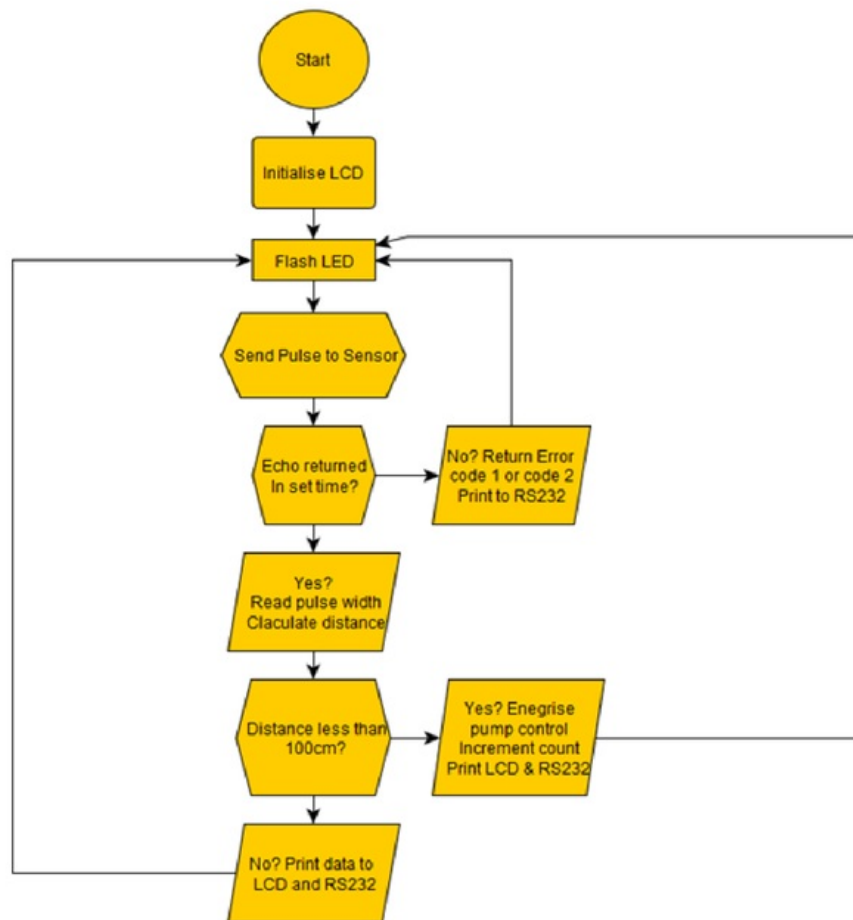


Figure 20 - Programme Flow Chart

The above flow diagram is to show that the steps the programme will step through and in which order throughout the running of the programme.

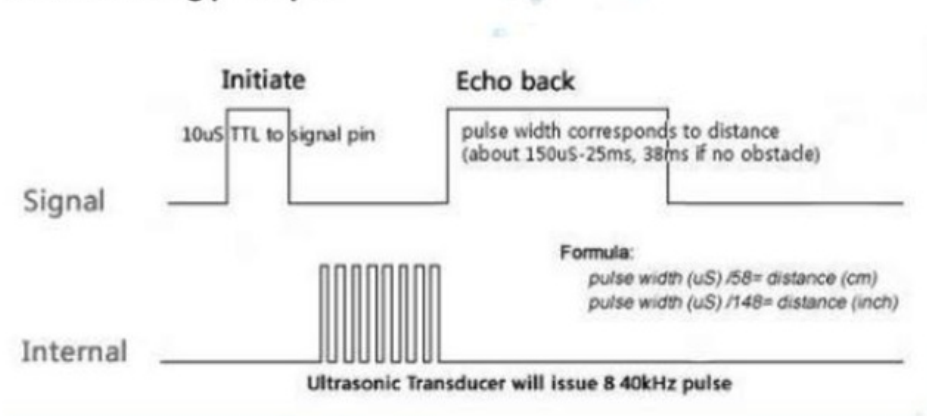


4.2 Implementation

For this project, it was decided that what was needed was a new way of measuring the level of water in an inaccessible underground tank on the site of NPT Recycling. What was needed was a new method and system of measuring the water level in the tank which is three meters deep, a depth that was acquired from site drawings, and also able to control a submersible pump to be able to control the level of the tank. As this system, at the moment, is only a trial system it was wished by the management at the company to be able to do this as cheaply but accurately as possible so results could be monitored and then a decision could be made whether to implement this system in a more industrial standard further down the line.

For this system it was decided that the HC-SR04 ultrasonic ranging module would be used as it is more than capable of being able to measure over the distance that is required, and also will work well in a system that is based around the use of a microcontroller. The microcontroller can be programmed quite simply so that it can control the sensor, and also the sensor sends back a different sized pulse width which varies directly depending on this distance it has measured. This distance is measured depending on the time it takes for the receiver on the module to receive an echo signal from that sent out by the modules transducer. From looking at the data sheet of this sensor, it has a minimum time it can receive an echo of $115\mu\text{s}$ which is equal to 2cm, up to 25mS which is that maximum level of 4 meters the sensor can read up to. If no object is seen in front of the sensor, or something has gone aerie, sensor will read up to 38mS, giving you an error signal. The times mentioned above, are the length of time the module will send back a pulse of varied width to the microcontroller directly in accordance to the distance that it has measured. Below is a diagram from the data sheet explaining this.

Figure 21 - Working principle of HC-SR04

Basic working principle:

(17)

Also from the above diagram it can be seen that for the module to work in needs a 10µs pulse to be sent to it from the microcontroller. This pulse will then let the module know to send the 8 / 40 KHz pulse and wait to receive an echo signal back to the module. Essentially, the 10µs pulse from the controller to the ultrasonic module is an initiation pulse for the internal circuitry of the sensor to begin their steps.

The display for the level measurement made by the sensor will be an LCD display that is directly connected to set output pins on the microcontroller. The display is to first off display two lines of text reading " F.Prep Water / Tank Level". This text should be displayed on the system as soon as the microcontroller starts stepping through its programme. Also as this system is not specifically being used to store any data, it was decided that an readings being sent to the LCD screen could also be sent via RS232 to a communication port of a computer so that any technicians working on the system could see if it was giving true readings to both displays and there were no discrepancies. So again as the system was initialised, the same text would be sent via the RS232 connections to a connected computer with designated communication software.



4.2.1 Microcontroller and Ultrasonic Interfacing

As has been mentioned previously in this chapter it was decided that the HC-SR04 module would be used as the method of ultrasonic distance measurement in this system. The HC-SR04 needs a number of different variables for it to be able to function correctly. Firstly it needs a 5VDC permanent supply being given to it so that the internal circuitry of the module has power. Then for it to know when to start stepping through the programme it has embedded within itself, it needs an initiation signal to be sent to it from the microcontroller. This was done by a simple few lines of code being written in the Proteus C.Main file at the beginning of this part of the code. Below is the code that is written in C++ for the initialisation of the module:

```
output_high(Trig);           // ping the sensor  
  
delay_us(10);               // sending 10us pulse  
  
output_low(Trig);           // Take sensor low
```

As can be seen, three simple lines of code are written, firstly to send the pin "Trigger" high, meaning that a 5v or binary 1 would be sent to the module. There is then a delay for 10 μ S where the Trigger pin will stay high, and finally set Trigger pin low back to 0V or binary 0. The (Trig) was defined in the H.File of the Proteus software, and this was set on the pin which was connected to the trigger pin of the module which is PIN_A1.

Once the previous steps of the programme had been executed it was then time to wait for the echo signal to be returned to the Echo Pin of the ultrasonic module after it had been rebounded off the surface of the water of the tank. This was done by the setting of a timeout=0 function and a number of "If" and "while" loops. These loops were introduced because, when the code was trying to be written to simulate what was actually happening, the programme was stalling in a specific spot and not moving along any further. As there



was no time out on the signal returning to the microcontroller, the system would not step to the next section of code and would sit in an infinite loop waiting for the return echo signal. If this had been left like this in the real life application and the distance had been at too greater distance for the sensor or, the sensor had a fault on it, the code would never move out of this loop. Also there was error codes incorporated into this section of code to allow the system technician to be able to see what error code was showing on the RS232 communication port. The following lines of code show the programme waiting for the return of the Echo signal to the microcontroller, and then showing error would should it exceed its time limit that has been set.

timeout=0;

```
while(!input(ECHO))                // wait for high state of echo pin  
  
    {  
  
        timeout++;  
  
        if(timeout==12000)  
  
            {  
  
                printf("error 1\n\r");        // No Echo Return Error Code 1  
  
                break;  
  
            }  
  
    }
```

Maybe not clearly conveyed in these few lines of code, but what its being achieved by this “while” loop is that firstly, the time out counter is set to 0. Then whilst the echo pin



is low, hence the 'while(input(echo))', increment the timeout count which is achieved by the 'timeout ++' line of code. Then the following "If" loop of if(timeout==12000), which means the count had incremented to 12000, print to the RS232 ERROR 1. This is then followed by a break command, which was found in the CCS C help window, and is used to exit out of a control loop early. It enables the ability to move the programme on from this line, and followed by the two '}' brackets to finish both the "while" and "If" loops. The exclamation mark that is before input in this code means 'not'. Simplistically, if this was written in words, whilst echo pin is not an input.

Though, if the Error 1 code is never made, and the count didn't increment to that level, the programme will continue doing what it should be doing. If the echo pin is to become an input and go High before the 12000 increment from the first set of loops, there are more lines of code which start to work on other functions in the system. The next loops in the programme and code are there for once the pin does go high, to be able to calculate the time, or count incrimination from when the pin goes high to when it goes low. Essentially what this means is depending on the width of the pulse sent back to the microcontroller from the ultrasonic module, this will define the number that the count within this while loop will increment to. As before in the previous lines of code, there is an internal "if" loop within the while loop, which is incrementing constantly whilst the pin is high with its own maximum level set to supply another error code to the RS232 communications. Below is the example of that code and it briefly explained by the comments next to the code.

```
timeout=0;           // Reset timeout count to zero  
  
while(input(ECHO))   // Echo pin is high  
  
{
```



```
        timeout++;           // Increment timeout count

        if(timeout==65000) // Timer overrun- stops programme getting caught in
loop

{

    printf("error 2\n\r");    // Display Error code 2 on RS232 -- Return pulse width
too long

    break;                   // Exits out of control loop early.

}

}
```

The three sections of code shown and explained above are the full sections of what is needed in the code of this system, for the ultrasonic sensor to be able to firstly activate, the system to be able to continue running whilst it is waiting for a return echo signal and finally be able to display if there is a fault on the system or simply return the correct data that has been supplied back to the microcontroller from the ultrasonic sensors in a pulse that varies in width. Without the error codes and separate while loops, the system has the ability to be able to freeze in one place, waiting for an echo signal, and with no way of being able to access the data the system is stopping on as there is no error visible. This was a fault that was discovered when the PIC was programmed and a real life test was carried out. There was a loose connection on the signal wire returning to the PIC from the Echo pin of the module, thus leaving the programme in an infinite wait for the Echo signal to return.



Once the code was adjusted to match what is in the code above, the system worked perfectly running through its sequence time and time again.

4.2.2 LCD display with readings

For this system it was decided that an LCD display would be used to display the data to the plant operators at NPT Recycling so that they would have access to the number of times the pump had been energised over a certain period of time. This display was also decided to be used to be able to show the operator, or anybody who was interested in the tank, at what level the water was be it one meter from the top or three meters away, being the bottom of the tank, thus the tank being empty. For this it was decided that the type of LCD display to be used would be a 16 character by 2 line display. This was sourced from RS components in the shape of the Display-tech 16x2 162B LCD display.

Firstly, before any hardware was purchased or even finally decided upon, the setup of the LCD display had to be completed. By using the CCS C software previously mentioned in this report, and utilising the project wizard this is a very simple step to set up. Going through the steps to set up the microcontroller for a number of things in this project wizard is very easy, and is a spectacular tool for someone who is a beginner to using CCS C or just basically C++ programme language. To setup the LCD in the project wizard, firstly you must open a new project wizard from the menu. Once this has opened and it has been saved to its required destination the first thing that must be done is to choose the microcontroller that is being used, for this system it is the 18F2455. This is done as shown in the figure below.

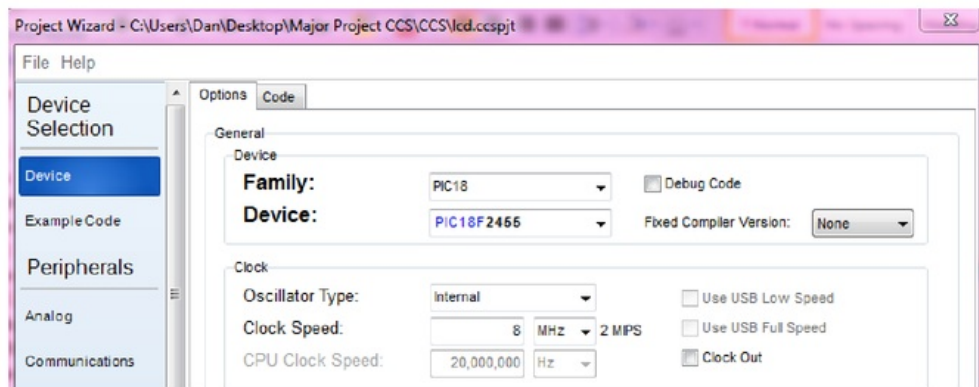


Figure 22 - Choice of PIC in Project wizard

Once this step has been completed the next step to be taken is to scroll down the list on the left hand side of the window to the external LCD selection pane. Once this has opened, the first step to take is to select the use LCD tick box, this opens up the drop downs so you can assign what pins you want to be going to where on the LCD or vice versa. For this system the pin assignments were:

Enable = Pin B2, RS= Pin B0, RW= Pin B1, Data pin 4 = Pin B4, Data pin 5= Pin B5, Data pin 6= Pin B6 and Data pin 7= Pin B7. All of this can be seen from the two figures below showing the choice of pin outs for the PIC and LCD and also the code which goes with what has been chosen. There is also the choice to add example code which could be used in this programme but it was decided not to be used in this instance.

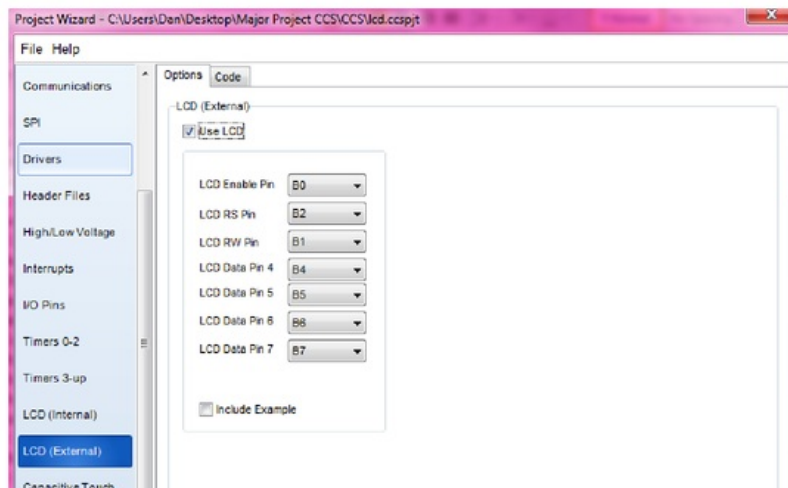


Figure 23 - Pin-outs of LCD

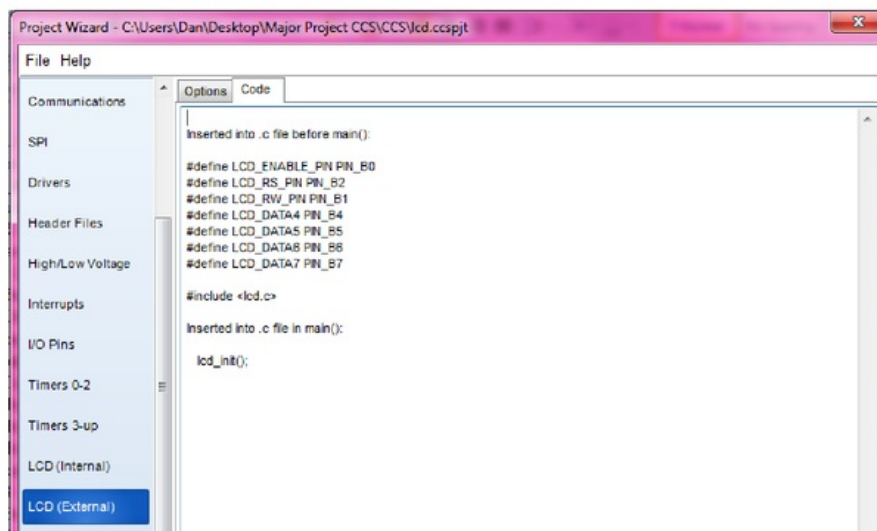


Figure 24 - Code that will be inserted into .C file before the start of the main programme.

Once these two steps have been completed in the project wizard the LCD is completely set up ready to be connected into the circuit within Proteus ready for the simulation part of the system. This information will not change for when it comes to actually programming the microcontroller later on in the implementation. To be able to have the LCD reading the desired data, there are lines of code which must be inserted into

the code so that this can be made possible. At the start of the code in this system, there is so to say a welcome message which reads, "F.Prepare Water Tank Level." For this to be displayed on the LCD display there is a simple line of code which must be written. Below is the line of code and the breakdown of what certain sections of it actually mean.

```
printf(LCD_PUTC, "\f F.Prepare Water \n Tank Level" );
```

The 'printf' statement means that the function that is going to be completed following it is to print a certain string of data somewhere and the 'LCD_PUTC', states where the data is going to be written to. Next within the " " the \f means to return to the first line and the \n means to move down to the next line. If you have the \f at the beginning of any "printf" statement, it means that it will return to the top left hand side, the starting point and will clear and data already in that position. In essence what this command does is clears the screen and starts again. Once this line of code is written, that is it, the LCD will show F.Prepare water on the first line and Tank level on the next. This can be seen in the below figure in the simulation.

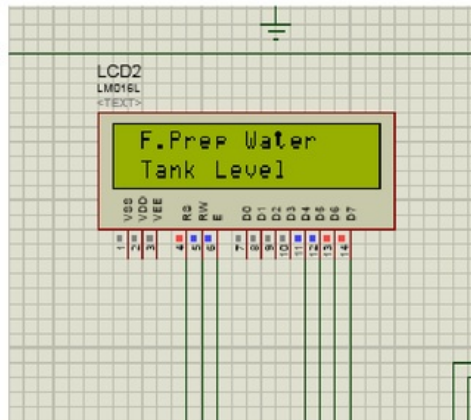


Figure 25 - Initial LCD start up display

As this initial display does not actually have to write any data from the system, the line of code is a little bit similar than that later in the code for writing the pump count and the tank level readings. Again for these two commands, the code is very much similar with



a couple of more added commands. Below is the code that was written to be able to display the distance and count to the LCD display.

```
printf(LCD_PUTC, "\f Pump Count =%u \n Distance=%ucm ",count,distance);
```

As can be seen, the command starts the same but this time, there is the command `=%u`, which means that what it is printing is an unsigned integer. The pump count and Distance are defined as unsigned integers at the beginning of the code before getting into the main programme section. Once the text comes are closed, the `'(",count,distance);'` defines what data is being written to what part of the command. The count is to be printed as the first 'unsigned int', and the distance reading is to be printed as the second unsigned int. For the LCD display this is all that was needed to be written as it was displaying only these two pieces of data. This data will update every time the programme cycles through to the `Printf(LCD_PutC, command` within the main programme.

4.2.3 Circuit Design, Programme and Simulation

For the circuit design and simulation of the system, as previously mentioned the Proteus Professional software was to be used. Also within the Proteus software, the majority of the code to be written would be done here. As Proteus can be set up to compile with CCS C, it allows the user to be able to write the code in Proteus, and then compile in this window also to make sure that the code is correct. If there is a fault with the compiling of the code that has been written, at the bottom of the Proteus window it will tell you in which file, whether it be in the .C file or the .H file and also which line of code that is not able to be compiled. This makes fault finding of the code much easier for beginner code writers and more advanced programmers alike.

Setting up of the Proteus software again is a quick and easy process. Proteus itself has a project wizard, just like CCS C, for setting up the design and firmware for the system

wishing to be built. This wizard is user friendly to use for anyone, be it a professional who is using the software every day, or someone who has possibly never seen it before. By following a number of simple steps everything is set up for you. Firstly it must be decided where to save the design that is to be built, followed by the choice of size of schematic that is wished to be used. The final step of setting up the Proteus design then is to decide what firmware, if any is wished to be used. At this step the PIC18F2455 was entered and the compiler choice was CCS C. In figure 26 below it shows the setup screen of the wizard from Proteus. Once Proteus was set up, the circuit design was able to be started, and Pins were able to be assigned to their desired targets on the schematic ready for setting up the programme code in the CCS C compiler wizard.

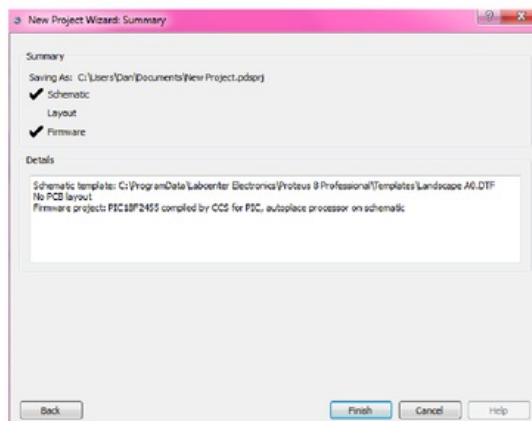


Figure 26 - Project wizard setup Proteus.

The CCS C software is also a very powerful tool, and quite simplistic to use and set up. With the help of the project wizard CCS C makes setting up the major parts of the programme much quicker and easier. With the help of the project wizard there are a number of commands and functions that can be set up with example code also being placed into the .C and .H files for the user. This makes for a much faster and easier way of setting up the programme and defining certain aspects of the code. If you are a beginner, this tool is brilliant to help you understand what lines of code mean as possibly, without



seeing the example code that it produces it could take a number of hours to work out a simple command.

Within the project wizard the functions you are able to set up are such as, firstly choosing the PIC you are going to use and then in an example code window you are able to set up a "Tell-Tale" in a flashing LED command. This is four simple lines of code, yet having this on the circuit gives the ability to be able to see if the chip is running. Also in this wizard, the communications if any can be set up with ease. With this system it was decided to use RS232 communications, and this is the section where this was set up previous to writing any code at all. Next was to set up the LCD external, which has already been talked about and explained in the design section of this document.

Finally in this wizard, the fuses for the controller can be set up here and the code for these will be written into the .H file without any trouble and ready to be used. For this system the only fuses that were selected were a fail-safe clock monitor, Power up timer and a reset if a brownout is detected, which was set to 2.7V as the microcontroller runs at 5V that was deemed to be a sufficient dip in voltage being supplied to the controller. In the below figure is the fuses setup page in the wizard and also a shot of the way the code is set up in the wizard also.

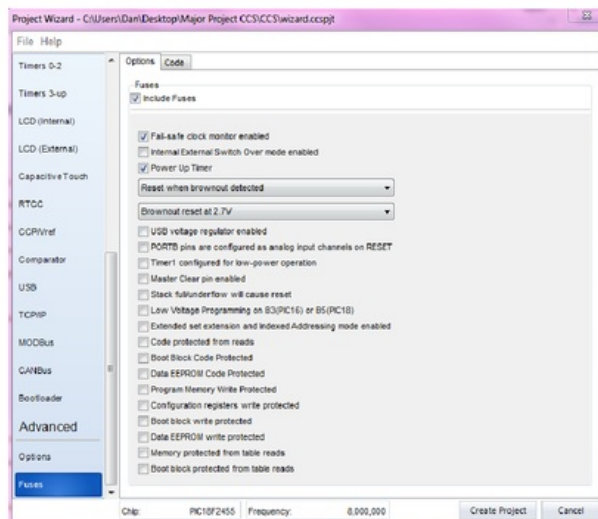


Figure 27 - Fuse setup in CCS

In figure 28 below, not only does this give the fuses that are needed for the system, but also the fuses which are not being used which the PIC18F2455 is capable of using.

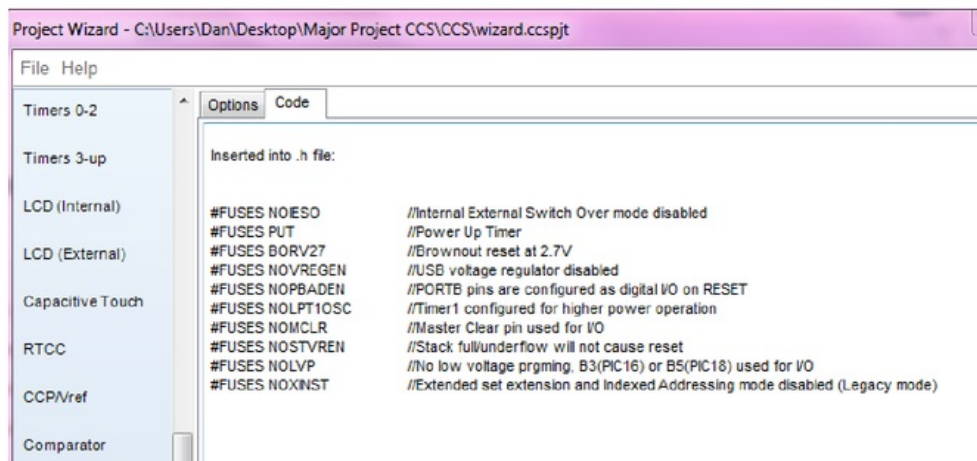


Figure 28 - Code setting up fuses

Once this wizard was set up, it was a case of clicking the create project button and the setup of the Microcontroller and certain functions in the programme was completed and able to be imported into the Proteus software in their .C and .H files respectfully.

Following the circuit designs completion and the completion of writing the code for the system, the next stage was to be able to run a simulation using the Proteus software to

see if all of the sections of code worked as were desired. The major aspects of the system that were desired to be seen to simulate correctly were the printf commands to both the LCD display and the RS232 communications, the microcontroller sending the initialisation 10 μ S pulse to the ultrasonic module and should the maximum and minimum levels of the range be reached to energise the pump until the lower limits were reached. The LCD display printf command has been shown as working in the previous section, and below in figure 29, the LCD and RS232 are shown to be showing the same readings with the RS232 holding its data also.

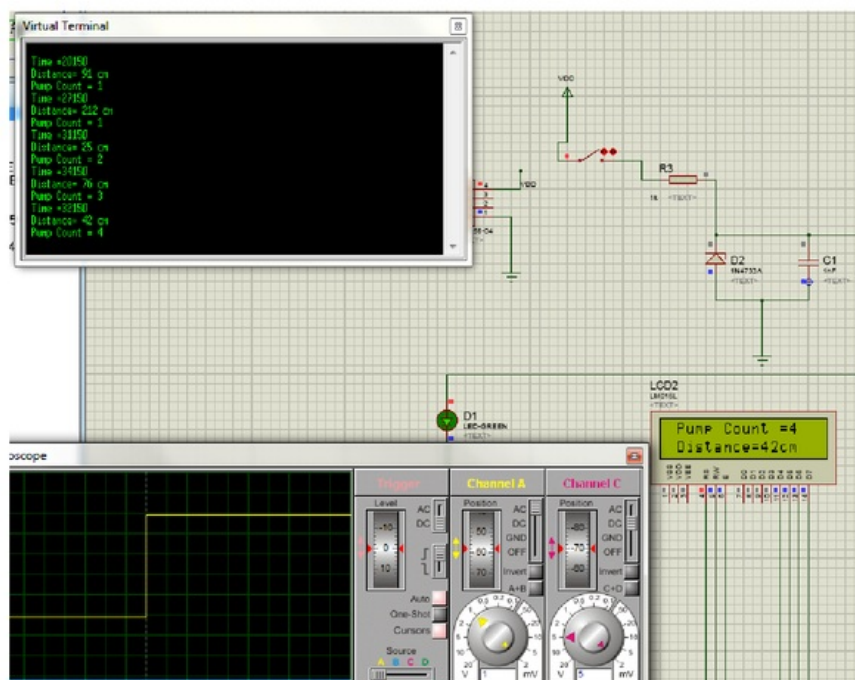


Figure 29 - RS232 receiving data, LCD display matching RS232 data & microcontroller sending initiation pulse to ultrasonic module.

As can be seen from the above simulation figure, for the RS232 Communications port the virtual terminal in Proteus was utilised and this was connected via the RX and TX pins on the microcontroller. From the above figure, it is seen that the water level is below the 100cm set point for which the pump is wished to be energised, so it was wished that the microcontroller would have outputted a high signal or binary 1 from PIN_C2 to engage

the pump to lower the level. The below picture in figure 30 shows that the system did simulate successfully.

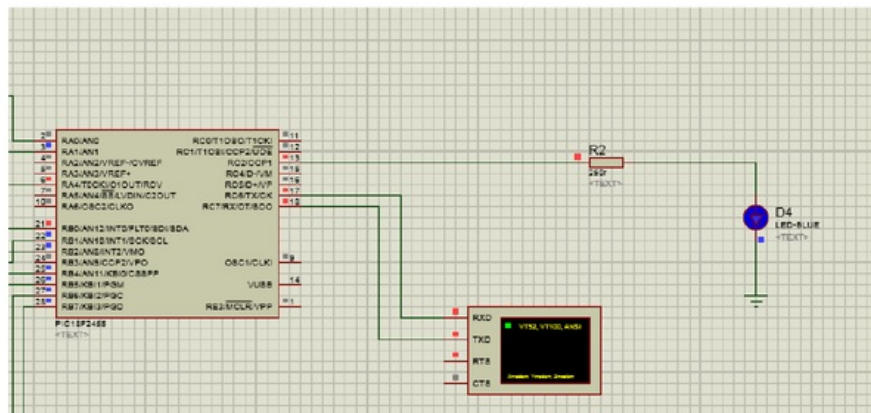


Figure 30 - Blue LED lit simulating pump output was set high correctly.

There was a specific section of code for this operation to be made possible at the end of the programme where the set points were defined for the pump output of PIN_C2 to go high and when it was wished that it would switch back to low. Figure 31 below shows this section of the code in Proteus.

```

if(distance<100)           // If the distance is less than 100 Increment Count
    count++;               // This means that the pump has run

if (distance>200)          // If the distance is greater than 200 turn pump off
{
    delay_us(10);
    output_low(pump);      // If the distance is greater than 200 turn pump off
}

if (distance<100)          // If the distance is less than 100 turn pump on
{
    delay_us(10);
    output_high(pump);     // If the distance is less than 100 turn pump on
}
  
```

Figure 31 - Pump set points defined

The comments at the side of each 'if' command are very self explanatory, if the pump runs increment the count by 1, if the distance drops below 100cm or 1m switch the pump on, if the level is more than 200cm or 2m switch the pump back off. With the simulation of this section there was a problem with how the code was written with



concerns to the count incrimination. As it was only a 5V switch which was manually being closed in the simulation to simulate the return echo to the controller, every time that the distance was less than 100cm the count would increment. In the real world application of this system it is hoped that this can be solved by means of a possible delay between how often the system is to go through the programme allowing for enough time for the pump to have taken the water level past the 1m mark thus, not to affect the count in this manner.

4.2.4 System Build and Test

Once the simulation was completed and was seen to be working as expected, the next stage was to download the programme onto the PIC and run the system to see how it worked compared to the simulation results that were acquired. To download the programme to the PIC firstly a .HEX file had to be generated from Proteus to be able to use the MPLAB IPE download software. This software is easily available from Microchip, the manufacture of the PIC microcontroller family and is a fast and easy piece of software to use. To be able to actually programme the microchip there was the need for a Pickit3 downloader module which works with the MPLab IPE software to download the programme onto the microcontroller. The MPLab software and PICKIT3 module were readily and easily accessible in the laboratories at the university so this made programming the chip easy.

When programming the chip there was one error that was stated by the software that the target device was not found as there was not supply voltage to the VDD of the PIC. This problem was solved once the fault was looked up on the internet and stated that the PIC had to be supplied its 5V from the PICKIT 3 device to be able to download the programme onto it. This was completed by going into the advanced settings section of the MPLab software and selecting the "Power target from tool" tick box to supply the 5V that VDD needed. Figure 32 below shows this selection.

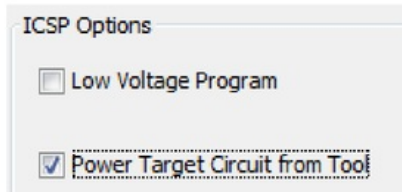


Figure 32 - Selection of power target circuit

Once this had been completed the download was completed successfully and the microcontroller was ready to be placed into the circuit to test the system out for the first time in a real life application. Before actually giving the system any power, the decision was made to actually test what voltage was being supplied to the bread board power rail from the 240V to 5V transformer that was mounted in the enclosure for the circuit. It was found that the transformer was supplying 5V AC to the power rail which was not what was desired for this circuit. This was the first problem that was encountered during the testing of the system. A bridge rectifier was inserted into the circuitry to rectify the 5VAC voltage being supplied to 5VDC. Once this was incorporated into the system, VDD of the controller was connected to the power 5VDC power rail of the breadboard and the system was given power.

At this point the system was expected to start to run, or at a minimum what was wished to be seen was the flashing LED loop being run. There was a fault on the system as none of the components seemed to respond to power being supplied to the system. After further testing of the supply voltage it was found that the bridge rectifier was not giving any voltage to the power rail on the breadboard, meaning the circuit was not getting any supply voltage anywhere. It was at this point it was decided for the purposes of initial testing to use a 5VDC power supply that was accessible in the university lab to power the system. Once this was used as the power supply to the system the LED flashed once but then nothing else happened.

The LCD was lit at the back light but was showing no data or readings. The decision was made to connect the USB to TTL lead to the RS232 connections so that if the system was streaming any data it could be seen on the “putty” software that had been downloaded, which is a virtual terminal software for PC. This again showed no signs of anything within the system running. The next step of the fault finding was to make sure all of the connections were correct on the LCD display as in the simulation with the RS and RW connections the wrong way round the LCD display showed no display. This was done but still to no avail. After researching on the internet for similar problems with systems that incorporated an LCD display it was suggested to remove the LCD commands from the code, re-programme the PIC and see if this made any difference to the system.

After this was completed and the new code was compiled and downloaded onto the PIC the “Tell-Tale” of the flashing LED was running as it was in the simulation. The conclusion to this fault was made by the author that there was a fault with the LCD display itself and the microcontroller was not receiving what’s known as a ‘hand shake’ from the display saying that it had received the data and to move on to the next step of the programme. Below in figure 33 is a photograph of the design set up with the LED lit showing the system is running.

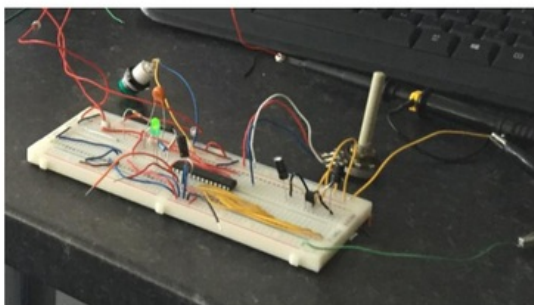


Figure 33 - LED lit showing system running

As now the LCD display had been removed from the system, the RS232 communication connection was the only means to which it could be seen what data the

system was receiving back from the ultrasonic sensor and what the distance calculation the microcontroller was making. So the next step was to connect the USB to TTL lead into the system, open up the putty virtual terminal and see what data was being sent from the PIC. At this point yet another problem arose. The data that was being sent out from the PIC seemed to be corrupted for some unknown reason. As the communication had worked in the simulation it was not a fault in the code that was the problem but something different all together. Even though the data being sent was corrupted, it was being sent as it should be every time the programme got to the commands at the end of the programme loop. So this determined that the code was correct and the controller was sending the data as it should be. Below in figure 34 is a photograph of the data which was being received by the Putty terminal, but also showing that the code was correct as the terminal was still recording the data and not re writing over it. This was a fault that was found when simulating the system in Proteus. This fault was found to be caused by the “\f” at the start of the command as what it means is first. This command will take you back to the top left hand corner of the display and start writing again from the beginning.

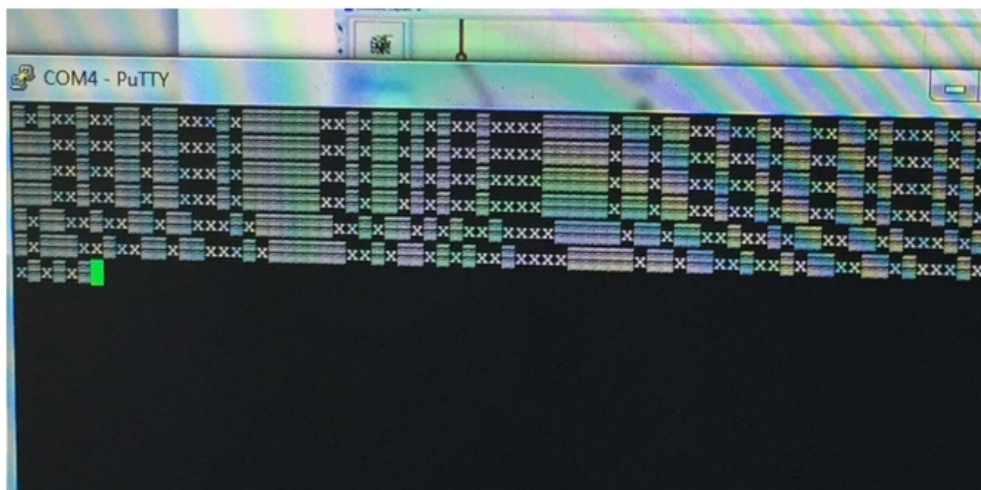


Figure 34 - corrupted data being send via RS232 Communications.

As there were problems with both of the dedicated ways of seeing the data that the system was producing the next step was to make sure that the microcontroller and ultrasonic module were both working correctly as they should be and that it was nothing to do with a simple fault of the module returning its pulse. So the first test undertaken was to attach to cable which was supplying the initiate 10 μ S pulse to the module to make sure that the microcontroller was completing this step completely. Below in Figure 35 it can be seen that this was working correctly.

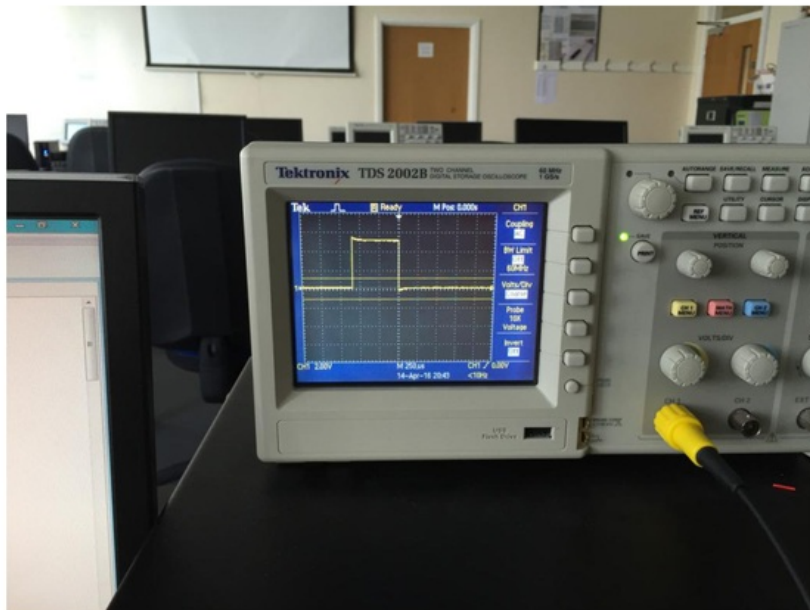


Figure 35 - Pulse being sent to ultrasonic module from microcontroller – scope set at 250 μ s/div – 2v/div

After seeing this result from the testing, it was known that the system was running correctly from the beginning to the end as this pulse shows the beginning of the code, and the data being written to the RS232 shows the end of the code. The next test was to see if the ultrasonic module was sending back a pulse of varying width depending on distance to the microcontroller. For this test to be completed the cable that was supplying the pulse to the microcontroller was connected up to the oscilloscope and the system was set to run to see what resulting pulse widths were received depending on the variation of distance.

Below in figure 36 is a photograph of the pulse sent back to the controller at an unknown distance with the scope set at 25ms/div and at 2v/div.

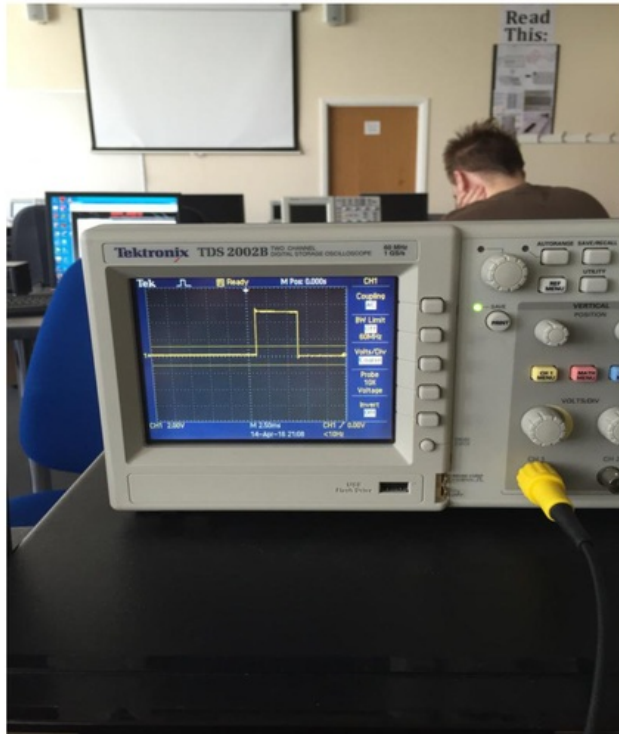


Figure 36 - the pulse sent back to the controller at an unknown distance with the scope set at 2.5ms/div and at 2v/div.

As can be clearly seen, the module is sending a pulse back to the PIC with regards to the distance it has measured using its internal controls. Following seeing this result, and knowing that the ultrasonic sensor was sending back a pulse signal to the micro controller, it was decided to test if the pulse width did change with a change in distance measured. The next test, which is displayed in figure 37, was with an object placed 20cm away from the sensor but with the oscilloscope settings held the same. The oscilloscope settings were being set the same to make sure the width of the pulse did vary in response to the change in distance.

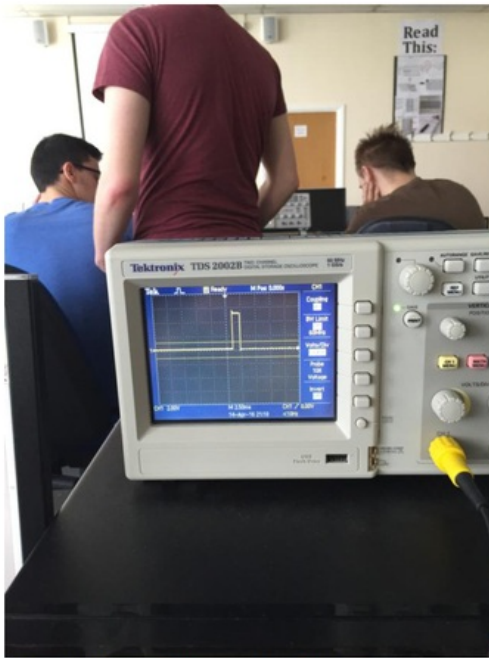


Figure 37 - Object placed 20cm away from sensor, with scope settings remaining the same.

The photograph which is figure 37 shows that the theory of the pulse width changing directly proportionally to the change in distance is true. This is the result that was expected after research was carried out to what should happen. Not taking any single result for granted it was decided to do another distance to pulse width test to make sure that this wasn't the only width pulse that was being sent back to the microcontroller.

So the next test, which is displayed in figure 38, is the pulse width that was recorded with an object place 50cm away from the ultrasonic sensor. What was expected to happen in this test was for the pulse width to be wider in response to the time the echo had to travel to and back from the object it was detecting.

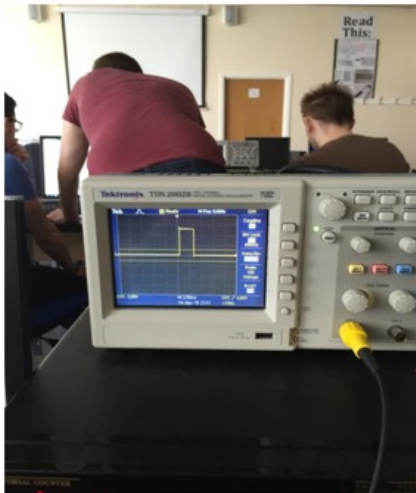


Figure 38 - Pulse width result on scope with object placed at 50cm away from sensor, but all scope settings kept the same.

This test completed the testing of the theory that as the distance from the sensor increases, so does the pulse width that is sent back to the microcontroller. Also this test also showed that the ultrasonic sensor module was working correctly as it should have been varying the pulse width in regards to the distance the ultrasonic burst had travelled. Below in figure 39 is a photograph of the test rig that was set up for these test to be completed. All be it a basic simple set up, it worked efficiently and gave out the expected results.

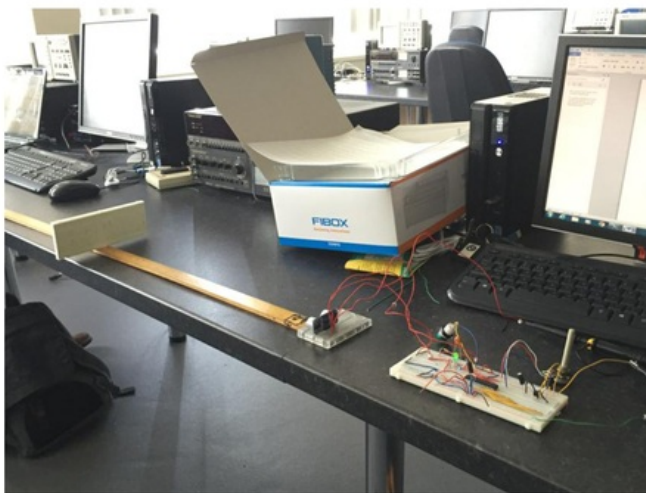


Figure 39 - Test rig set up



During the testing of the system, there was one aspect of the code seeming not to work, and that was the microcontroller giving a signal to the pump to run, as all of the tests were under the 100cm set point. After testing Pin with a multi-meter to see if PIN_C2 had gone high, it was found that the output was only 2.3VDC. During the testing of this system, the output pin assigned to the pump was connected to a panel indicator lamp, to simulate the pump running and it getting its signal. At first it was thought that the voltage the microcontroller was sending out was under voltage, yet for no apparent reason. Once the supply cable to in the indicator had been removed and PIN_C2 was tested again it showed that the pin was actually giving out a 5VDC voltage. This was working correctly.

It was clear that the panel indicator was for some reason dragging down the voltage being sent out from the microcontroller. The indicator was first tested by giving it a 5VDC supply and worked perfectly. So the first step taken in trying to rectify this fault was to place a 250Ω resistor in line with the supply to the indicator to see if this would solve the problem. This did not work as it seemed the 250 Ω resistance was too much and the lamp would not fire from the PIC supply or even from a separate 5VDC supply. Next was to try and incorporate a transistor into the circuit. This would simply act as a switch, being able to handle the current being pulled by the panel indicator on the initial firing of the lamp and being able to keep it on.

The first transistor that was tried was acquired from the store room in the university after explaining what it was going to be used for. This transistor was placed onto the breadboard, and once finding the data sheet with the Pin-out of this certain type, connected up to the light with still no joy. After testing with a multi meter, it was found that it was now being supplied with 5VDC from the controller, but was not switching as was hope. When looking into the data sheet further it was found that this actual transistor had too high a resistance, again giving the problem of not being able to fire the lamp inside the

indicator. The decision was made to look for a switching transistor with as low a resistance as possible.

It was decided to use a Mosfet for this application as they are capable of carrying high currents but are to have a low resistance. Once this Mostfet was incorporated into the system, and the system was powered up, the panel indicator lit up as was designed to when the distance was under 100cm, simulating the pump running. Below in figure 40 is a photograph of the indicator lamp lit up with the distance at around 55cm. The Mosfet itself was wired up so that the signal from the controller would switch the Mosfet, and Sink the voltage to ground completing the circuit of the indicator. This would be a suffice way for controlling the actual pump control when taken to site.

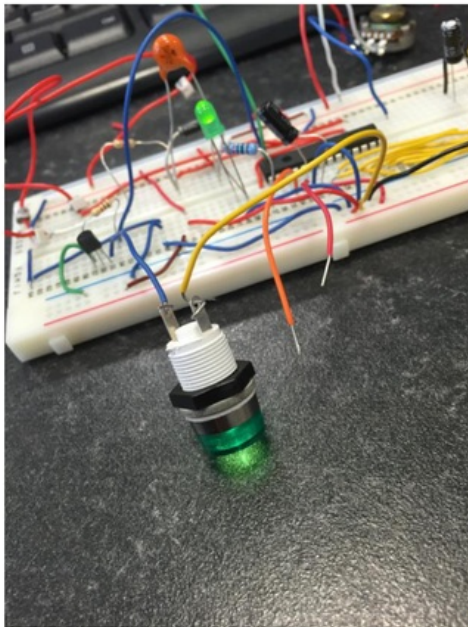


Figure 40 - Indicator illuminated with distance at around 55cm. Wired up to sink the voltage.

Once this task had been completed, and was deemed to be working correctly as should be and it was seen that the system was running as it should be, but not without the two problems of the LCD not showing data and the RS232 comms fault.



Chapter 5: Evaluation

The system that was to be designed and implemented which was stated in the abstract and the introduction was done with success. What was required was a system that has the ability to measure the level of water in an inaccessible underground tank and be able to control the level of water in that tank also. From simulation of the system and bench tests that were undertaken it showed that the ultrasonic module would vary the size of its pulse width in direct proportion to the distance the ultrasonic signal has travelled.

Unfortunately to this point not all of the system is working correctly in the LCD displaying data, as a fault was found to be on the display itself and also the RS232 Communications sending data correctly, but this data being of a corrupt nature. It is being worked towards that these faults can be overcome in the near future with a new LCD display being purchased and trying to overcome the communication fault with the RS232.

Another fault is still present in that of the 240V AC to 5V AC transformer. To overcome this the decision was made to use a 5VDC power supply for testing purposes to see if this would be ample to run the system as it is to be a permanent fixture at NPT Recycling and a battery just wouldn't suffice. With this being found to be a more than capable way of powering the system, the next component to be ordered is a 240v to 5VDC panel mount power supply rather than an AC-AC transformer. This will also make the circuitry easier to understand as it will remove any need for bridge rectifiers in the system.

At the moment, with not being able to see any data out of the system, it is still unknown if the formula being used in the code for calculating the distance is correct or not. It is not the best of situations to be in, yet the formula that has been used was taken from the datasheet of the HC-SR04 module, so this should be correct. This is a disappointment, but it is still being worked towards to be corrected as soon as possible.



Although the system did present its faults when it was being tested, this gave for good opportunity to be able to test different components and sections of the system, to determine whether the system and components were working correctly in their own sections. This was a new type of fault finding for the author as it is normally industrial electrical fault finding that is undertaken which is a completely different array of problems, though some knowledge was able to be implemented such as testing for certain voltages and continuity testing in certain components.

From the testing of the ultrasonic module it was able to be seen that from what was said earlier in the document and research of the sensor that it did work correctly as it should. As the distance increased or decreased so too did the width of the pulse the module was generating. Also from testing the output of the microcontroller to the ultrasonic sensor it was seen that this aspect of the programme worked correctly as wished.

The thought is that the system does, as a whole, work and the implementation and writing of the code was completed correctly with proof of this within the simulation inside the Proteus software running as should be expected. Also within the code there were problems and faults found originally that were put right by the implementation of simple "While" and "If" commands and loops. The Software used throughout the project was found to be very powerful, but also very helpful and user friendly to an inexperienced electronic system designer and also a very inexperienced programmer.

Overall, the aim of designing a system that has the capabilities to control the level of an underground water tank, display the measurement of the water in the tank and have the accessibility to a technician to see if there are faults or problems with the system were met. If this project was to be done again, there wouldn't be a different choice of system control; the microcontroller is a nice and compact way of controlling the system. As for the



ultrasonic sensor, for what the system is, a trial, was ample, cheap and had the capabilities of measuring the range that was needed for the system. If there was the scope to have a bigger budget to work to, the possibility of using a more industrial sensor would be considered greatly.



Chapter 6: Conclusion & Recommendations

Overall, it is seen that the design and implementation of the tank level system was a success. The aim and objective was to move away from the current method of tank level measurement which was the use of float switches and move to something more reliable. The choice of the non-contact ultrasonic sensor was seen to be the best option for this method of tank measurement and control, especially as the tank is situated underground. This project has got further avenues for research as it is believed that there are a number of other ways that this system could be incorporated into the controlling of the tank. Either by using a more industrial type of ultrasonic sensor or looking to incorporate this system into all of the underground tanks on site for overall automated tank control. Within the research there were a number of different methods of researched and considered for this system design, between float switches up to the more advanced ultrasonic distance measurement sensors. Once it was decided that the non-contact ultrasonic method was to be used, next came the research into which sensor would be utilised in the system and comparing different types by their functions and also a factor which NPT Recycling wanted taken into account the units actual price.

Following this was the design of the circuit for this system using the Proteus software and writing the code for the system between both the CCS C compiler software and within the Proteus software. The system was designed around a suitable microcontroller for this system, which was chosen from using the Microchip search tool and entering specific needs that were required for this system returning the PIC18F2455 as the best choice for the system. Following the design and the writing of the code, using the Proteus software, the system was simulated to test the code that had been written and see the system was displaying results that were expected to be returned. During the simulation



a number of minor errors were encountered but were quickly eradicated by the use of 'While' and 'If' loops within the source code for the system.

Once these faults had been rectified, during the hardware build and testing of certain components some more difficulties arose but the majority of these were overcome and the system worked as was expected from information found in the research and previous knowledge. After all of the component testing was completed and the system had been run, it was seen that the system met all of the needs for being able to measure and control the level of an underground tank. It will be capable of being used by NPT recycling for a trial to see if it is worth the investment to upgrade the system further. Further work to be undertaken in the coming number of days is to rectify the problem of the LCD display and also the problem of the communication fault with the RS232 only displaying corrupted data.

There are a number of recommendations that can be put into the design if there was to be an investment into expanding the system and making it more accessible and up market. Below is a small list of a number of these.

1. WIFI Accessibility – This could be incorporated so that you could access the system from your laptop/tablet/smart-phone at any time rather than having to plug the system into a computer Via RS232.
2. System accessibility – With this the use of possibly a HMI screen which a technician can have access to so that the pump control set points can be altered easily and fault finding can be made easier.
3. Using a more industrial built ultrasonic sensor so that it can be left as a permanent fixture at any point in any plant not just at NPT Recycling.



4. GPRS module – Again working on the same method of ease of accessibility, this module could also upload data from the system to a cloud account so that technicians or management can view what the tank is doing when, what is the tank level at any time, how long does the tank take to fill above the maximum set point.
5. Utilising the use of another piece of software which was discovered, Easy PC, designing a PCB board to incorporate in the system rather than just using a breadboard for the circuitry. Also using this software the design can be sent off and the possibility to have a PCB board made by an outside contractor.



Chapter 7: Reflection

During this project that I have been working on for the previous number of months there have been many things that I have come across and not seen before. There have been problems that I have never encountered before due to the simple fact of the very small amount of work I have done with electronic system design previous to the major project. It has given me much insight into how powerful microcontrollers can be and the amount of different types of systems and designs they can be implemented in. Another part of this project that has shown me something powerful is certain types of software and their capabilities. Previous to this project I had never heard of the software that has been utilised in the design and implementation let alone used any of it.

It has honestly shown me that even though if you encounter a problem when working on a system, or anything in that matter, nothing is fully impossible to overcome. At points I myself never thought that I would be able to get the system running to the extent that I had managed to get it to.

One major issue for me whilst working on this project, like many of my peers around me have also said, was time management. This project has taught me that when you are designing a system of any type, variation or control to a deadline, never ever leave it until the pressure has been turned on you to have it completed. This will 100% set me in better stead for time management of projects or even the simplest of jobs in my further working career. It also made me use techniques that I have picked up working in the electrical industry for testing equipment. Overall I did enjoy working on this system design and project and I have gained major amounts of knowledge about electronic system design which I did not have before. I would finally like to thank my Company NPT Recycling for putting me through my degree and also the lectures at the University for the Help that has been received throughout the overall course.



Table of Figures

Figure 1 - Research onion	11
Figure 2 - Float level switch	14
Figure 3 - Float level switch methodology	14
Figure 4 - conductivity sensor working	15
Figure 5 - Ultrasonic sensor	16
Figure 6 - Ultrasonic sensor theory and design	17
Figure 7 - Contact ultrasonic sensor	17
Figure 8 - Ultrasonic signal.....	24
Figure 9 - Optimum sensor position	25
Figure 10 - Air temperatruue to density table and graph	26
Figure 11- Pin count needed.....	32
Figure 12 - Microchip selection.....	34
Figure 13 - Selection of variables	35
Figure 14 - Comparison of microcontroller.....	35
Figure 15 - Comparison of ultrasonic sensors.....	37
Figure 16 - HC-SR04 Module	38
Figure 17 - Specification of HC-SR04 module	38
Figure 18 - Screen shot of system design	40
Figure 19 - System Block Diagram.....	41
Figure 20 - Programme Flow Chart.....	42
Figure 21 - Working principle of HC-SR04.....	44
Figure 22 - Choice of PIC in Project wizard	50
Figure 23 - Pin-outs of LCD.....	51
Figure 24 - Code that will be inserted into .C file before the start of the main programme.	51
Figure 25 - Initial LCD start up display	52



Figure 26 - Project wizard setup Proteus.	54
Figure 27 - Fuse setup in CCSC	56
Figure 28 - Code setting up fuses.....	56
Figure 29 - RS232 receiving data, LCD display matching RS232 data & microcontroller sending initiation pulse to ultrasonic module.	57
Figure 30 - Blue LED lit simulating pump output was set high correctly.	58
Figure 31 - Pump set points defined	58
Figure 32 - Selection of power target circuit	60
Figure 33 - LED lit showing system running	61
Figure 34 - corrupted data being send via RS232 Communications.	62
Figure 35 - Pulse being sent to ultrasonic module from microcontroller – scope set at 250µs/div – 2v/div	63
Figure 36 - the pulse sent back to the controller at an unknown distance with the scope set at 2.5ms/div and at 2v/div.	64
Figure 37 - Object placed 20cm away from sensor, with scope settings remaining the same.	65
Figure 38 - Pulse width result on scope with object placed at 50cm away from sensor, but all scope settings kept the same.	66
Figure 39 - Test rig set up.....	66
Figure 40 - Indicator illuminated with distance at around 55cm. Wired up to sink the voltage.	68



References

1. Research onion. *Uk Essays*. [Online] [Cited: 25 11 2015.]
https://www.ukessays.com/images/essays/the_research_onion.jpg.
2. Level control floats. *btreeit*. [Online] [Cited: 12 feb 2016.]
http://www.btreeit.com/images/level_control_floats.png.
3. plumbing supply pump info. *www.plumbingsupply.com*. [Online] [Cited: 14 Feb 2016.]
<https://www.plumbingsupply.com/images/pumpinfo.gif>.
4. Conductive level sensor. *www.automationwiki.com*. [Online] [Cited: 14 Feb 2015.]
http://automationwiki.com/images/4/40/Conductive_Level_Sensor.jpg.
5. *Design and implementation of water depth measurment and object detection model using ultrasonic signal system*. Unknown. 3, s.l. : International Journal of Engineering Research and Development, 2012, Vol. 4.
6. ultrasonic sensor. *Directindustry.com*. [Online] [Cited: 16 Feb 2015.]
http://img.directindustry.com/images_di/photo-g/894-7685437.jpg.
7. Ultrasonic tank sensor. *directindustry.com*. [Online] [Cited: 14 Feb 2015.]
http://img.directindustry.com/images_di/photo-g/6020-8671117.jpg.
8. Ultrasonic Liquid Level Sensors from SMD Fluid Controls. *www.fluidswitch.com*. [Online] [Cited: 15 Feb 2015.] <http://www.fluidswitch.com/blog/wp-content/uploads/2015/10/Ultrasonic-Liquid-Level-Sensors-from-SMD-Fluid-Controls.png>.
9. Ultrasonic advantages and disadvantages. *www.ab.com*. [Online] [Cited: 15 Feb 2015.]
<http://www.ab.com/en/epub/catalogs/12772/6543185/12041221/12041229/Ultrasonic-Advantages-and-Disadvantages.html>.



10. ultrasonic workshop. *Robotics*. [Online] [Cited: 28 Feb 2016.]

<http://robotics.jcu.io/images/ultrasonic/SensorSignalDeflection.png>.

11. Google images. *Allen Bradley*. [Online] [Cited: 2 March 2016.]

<http://epub1.rockwellautomation.com/images/web-proof-large/GL/48721.gif>.

12. Air Properties. *Engineering Tool Box*. [Online] [Cited: 10 March 2016.]

http://www.engineeringtoolbox.com/air-properties-d_156.html.

13. How Does Humidity Affect Speed of Sound? *Opposing views*. [Online] [Cited: 10

March 2016.] [http://science.opposingviews.com/humidity-affect-speed-sound-](http://science.opposingviews.com/humidity-affect-speed-sound-22777.html)

[22777.html](http://science.opposingviews.com/humidity-affect-speed-sound-22777.html).



Bibliography

1. <http://www.sensorsmag.com/sensors/leak-level/a-dozen-ways-measure-fluid-level-and-how-they-work-1067>
2. <http://hycontrol.com/level-measurement/>
3. <http://www2.emersonprocess.com/en-us/brands/rosemount/level/ultrasonic-transmitters/3100-series-ultrasonic/pages/index.aspx>
4. <http://new.abb.com/products/measurement-products/level/ultrasonic-level-transmitters-and-switches/kmicro>



Appendices

Any documentation following this point of the document is classed as being in the appendices section. This section will be supplied on a separate flash drive for ease of access and reading.