Chapter One

Introduction

1.1 Introduction

Almost all aspects of human life have undergone rapid development. This development is supported by the advance of electronics and information technology. The job can be performed on schedule precisely and efficiently by adopting this advance technology [1].

An achievement in computer technology is used not only in business and industry but has also covers almost all fields, including control system where a computer system can be used to control the hardware in a flexible way. Therefore, computer based control system becomes more common in recent development of control system [1].

Computer-based control system also can be implemented for optimizing river flow management to minimize flood caused by water overflow. Management can be performed based on elevation of water level on the river as an input data and control the sluices along the river stream based on these data [1].

The benefits of being able to control devices from one particular location has become imperative as it saves time and effort. Wireless technology in recent years has become very useful in the field of control and automation especially in remote control operations. The merits of embedded system are now being utilized into remote monitoring and control systems because of lots of advantages. Remote monitoring is an effective method in order to avoid interfering with the environment and improve efficiency.

1.2 Problem statement

There are continuous variations in water level, by increase or decrease the level that cause effect in environment. So there are great need for records it continuous and take decision in the records. In the past the method that is used to measure the water level is construction graduate columns in water, this method is consume time, low precision and large errors ratio.

1.3 proposed solution

To solve this problem I used two circuits, one for the transmitter and the other for the receiver. The transmitter circuit is put it near the water, and the receiver circuit is put it in the control room. The important components of the two circuits are ultrasonic sensor and Arduino used to measure the water level without any contact with environment, after that display the current measure in LCD monitor and compare it with previous record. This method is very active, easy to use and do not consume time.

1.4 Objectives

There are some objectives needed to be achieved in order to accomplish this research. These objectives are:

1. To design wireless water level monitoring system to understand the changes that take place in environments.
2. To collect the data of water level quickly as it saves time and effort.
3. To avoid a hazard if the water that is monitored is hot or cold.

1.5 Methodology

In order to complete this research smoothly, a lot of research work on flow monitoring systems was required such as going through reference books, journals, internet resources and components datasheets so that will assist in the success of the research. The system consists of hardware and software parts. The Hardware components which are needed to construct the system are:

1) The Ultrasonic ranging module HC - SR04.
2) RF Modules.
3) Arduino.
4) The Liquid Crystal Display (LCD).
5) The Light Emitting Diodes (LEDs) and the buzzer.

The Arduino is the “brain” of the whole system. It receives the input signals from the sensor and displays the water level on the Liquid Crystal Display (LCD), Light Emitting Diodes (LEDs) and the buzzer. A signal carrying the data should be send and received via a suitable system built for this purpose. Or processes done by this system should be under control of specific software program. The Software parts subprograms, perform the job, the language selected in this research is Arduino C program because it’s easy to use and the instructions are not complex.

1.6 Thesis layout

This research contains 6 chapters and they are organized as below:

Chapter 1: This chapter explains the introduction that includes overview, motivation, objectives, and methodology.

Chapter 2: This chapter describes the background and previous works about literature review.

Chapter 3: This chapter provides description the main components are used in this dissertation such as Ultrasonic sensor, Arduino, RF module, Encoder, Decoder, LCD, LEDs, and buzzer and also describes software program used in this system.

Chapter 4: This chapter describes the Hardware design that which is describes the connection between components.

Chapter 5: This chapter explains the results and discussion of the system.

Chapter 6: This chapter summarizes the overall conclusion and explains about features recommendation for this research.

1.7 Summary

In this chapter we exposed to the introduction of the thesis, and we mentioned the problem statement that is motivated to write this thesis, and then we explained the proposed solution to this problem, and we exposed to the objectives of the thesis, and also we explained the methodology, and the last is thesis organization.

Chapter Two

Literature Review

2.1 Introduction

The monitoring water level in a river or in a reservoir is important in the applications related to agriculture, flood prevention, and fishing industry, etc. The schemes developed for measuring water level can be categorized as four types based on the measuring features such as pressure, ultrasonic waves, heat, and image [1].

An ultrasonic level or sensing system requires no contact with the target. For many processes in the medical, pharmaceutical, military and general industries this is an advantage over inline sensors that may contaminate the liquids inside a vessel or tube [1]. In this chapter we used the concepts of the ultrasonic waves, and we mentioned the previous works about it.

2.2 Background

The first use for the concept of the ultrasonic waves was done by U.S. researcher Dr. Floyd Firestone of the University of Michigan applies for a U.S. invention patent for the first practical ultrasonic testing method. The patent is granted on 1942 under the title "Flaw Detecting Device and Measuring Instrument". Extracts from the first two paragraphs of the patent for this entirely new non-destructive testing method succinctly describe the basics of such ultrasonic testing. "His invention pertains to a device for detecting the presence of in homogeneities of density or elasticity in materials. For instance if a casting has a hole or a crack within it, His device allows the presence of the flaw to be detected and its position located, even though the flaw lies entirely within the casting and no portion of it extends out to the surface. The general principle of

His device consists of sending high frequency vibrations into the part to be inspected, and the determination of the time intervals of arrival of the direct and reflected vibrations at one or more stations on the surface of the part" [2].

Another use for the concept of the ultrasonic waves was done by James F. McNulty of Automation Industries, then, in El Segundo, California, an early improver of the many foibles and limits of this and other non-destructive testing methods, teaches in further detail on ultrasonic testing in his U.S. Patent (application filed in 1962, granted in 1966, titled “Ultrasonic Testing Apparatus and Method”)that Basically ultrasonic testing is performed by applying to a piezoelectric crystal transducer periodic electrical pulses of ultrasonic frequency.

The crystal vibrates at the ultrasonic frequency and is mechanically coupled to the surface of the specimen to be tested. This coupling may be affected by immersion of both the transducer and the specimen in a body of liquid or by actual contact through a thin film of liquid such as oil. The ultrasonic vibrations pass through the specimen and are reflected by any discontinuities which may be encountered. The echo pulses that are reflected are received by the same or by a different transducer and are converted into electrical signals which indicate the presence of the defect.

In ultrasonic testing, an ultrasound transducer connected to a diagnostic machine is passed over the object being inspected. The transducer is typically separated from the test object by a couplant (such as oil) or by water, as in immersion testing. However, when ultrasonic testing is conducted with an Electromagnetic Acoustic Transducer (EMAT) the use of couplant is not required [3].

2.3 Previous Works

In [4], S.Jatmiko proposed, Water level detection system is designed to facilitate human in collecting water levels data that can be performed in real-time. Ping sensor is used as a distance sensor for detecting water level by measuring distance between sensor and water surfaces. The system consists of two modules, transmitter and receiver. Transmitter module performs water level detection and transmits it to the receiver module as a data collector. Receiver module then displays the data on the screen. This system can be used as a part of the system that need the water level detection which can be collect remotely, such as, flood control system.

In [5], Jirapon Sunkpho proposed, Flooding is one of the major disasters occurring in various parts of the world. The system for real-time monitoring of water conditions: water level; flow; and precipitation level, was developed to be employed in monitoring flood in Nakhon Si Thammarat, a southern province in Thailand. The two main objectives of the developed system is to serve 1) as information channel for flooding between the involved authorities and experts to enhance their responsibilities and collaboration and 2) as a web based information source for the public, responding to their need for information on water condition and flooding. The developed system is composed of three major components: sensor network, processing/transmission unit, and database/application server. These real-time data of water condition can be monitored remotely by utilizing wireless sensors network that utilizes the mobile General Packet Radio Service (GPRS) communication in order to transmit measured data to the application server.

In [6], Samarth Viswanath proposed, Monitoring Systems are necessary to understand the changes that take place in environments. Remote monitoring and data collection systems are useful and effective tools to collect information from

bulk storage tanks and to monitor the same. The measurement of liquid inside the tank is most important and such systems are useful in industries which are categorized as safety critical systems. This paper presents the architecture and initial testing results of a low power wireless system for tank level monitoring using ultrasonic sensors. The data acquisition is done by the sensors used to sense the changes in the liquid level of the tank and is stored in the system's memory. A server collects the information sent from the onboard micro-controller through a GSM modem in the tank; saves it to a database and displays it on a website graphically.

In [7], Mahmoud Meribout proposed, an ultrasonic technique has been developed to examine the propagation of ultrasonic waves in the oil, water, and mixed oil-water liquids. The technique is expanded to determine the oil, emulsion, and water levels in an oil tank. A dedicated compact, low-cost, and programmable ultrasound-based Multi-layer level measurement (MLLM) device has been designed and implemented. The advantages of the new method over the current methods include contactless distance measurement, higher accuracy, lower cost, user friendly, simpler setup, and using non-nuclear rays. Additionally, the use of ultrasonic waves for the measurement has the advantage over light-based methods of being insensitive to dusty and smoky environment and almost independent of the object material and surface Preliminary experiments have been conducted on the device.

2.4 Summary

In this chapter we exposed to the background about the concepts of the ultrasonic waves, and explained the first use of concepts. Also we mentioned the researchers who used this work to measure the water level or other measures.

Chapter Three

System and Components

3.1 Introduction

The aim of this research is to develop prototype of water level detection that can be viewed as a part of control system of river flow management system. The system consist of two parts, transmitter and receiver modules. The transmitter module detects water level automatically, then transmits the data to the receiver. Ultrasonic sensor is used to detect the distance between sensor and the water surface. The water level detection is performed without physical contact between the sensor and water surface. The Ultrasonic sensors utilize the principle of sound reflection to measure the level of the water. Elapsed time required to transmit and receive the reflected ultrasonic wave is multiplied by the rapid propagation of sound in water in order to obtain the distance value [1]. The receiver module has supply voltage 5v and the maximum antenna length is 24cm the value who is received is connected to display system and this chapter discusses the components use to construct the system.

3.2 Component Description

The main components used in this dissertation are Ultrasonic sensor, Arduino, RF module, Encoder, Decoder, LCD, LEDs and buzzer.

3.2.1 Background of ultrasonic distance measurement

Ultrasonic distance measurement is based on the speed property of sound. The system transmits multiple sound waves that travel out into the air. These sound waves reflect off from any objects they impact and return back as an echo to the location from which they originated. The system detects these reflected sound waves (that is, echoes). The time between the transmission of the sound waves and the detection of the echo is measured, as shown in Figure 3.1. At time t0, the transducer creates the sound waves. At time t1, the sound waves impact an object. At time t2, the waves have reflected off from the object and are travelling back toward the transducer. At time t3, the echo has impacted the transducer, which detects these waves. The system subtracts t0 from t3 to calculate the total travel time of the sound.

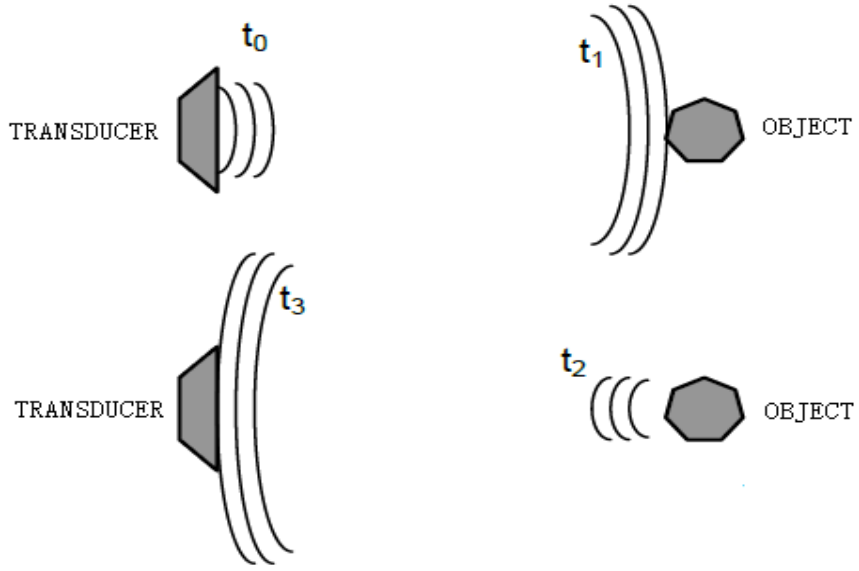

Figure 3.1 The time between the transmission sound waves and the detection of the echo

The sound's travel time is multiplied by the speed of the sound to calculate the total distance that the sound waves travelled. This distance is divided by two to calculate the distance of the object that caused the echo. These calculations

are shown in Equation 2.1, where S is the distance between the transducer and the detected object, V is the speed of sound, and t is the measured time between sound wave transmission and detection of the echo [8].

$$\text{The total distance(s)} = \frac{v \cdot t}{2} \qquad \text{Equation (2.1)}$$

3.2.1.1 The HC-SR04 Ultrasonic sensor

There are numerous types of ultrasonic range sensors available with key differences in frequency and power consumptions. Ultrasonic sensor with high frequency will have a sharper beam width and can detect obstacles in longer range. Also some of the new sensors have similar range detection as previous models but with less power consumption.

In this research, the ultrasonic sensor must be able to detect obstacles and objects from 2cm to 400cm. Since the whole system power supply will be taken from battery pack, the less power consumption is crucial and must be able to operate at low voltage. SR04 meets the criteria of this standard to detect the obstacles in a short period after the long research work was done to select between the SR04 and others Ultrasonic sensors [9].

3.2.1.2 HC - SR04 features

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The HC-SR04 Ultrasonic sensor is shown in Figure 3.2. The basic principles of work are:

(1) Using IO trigger for at least 10μs high level signal,

(2) The Module automatically sends eight 40 kHz signals and detect whether there is a pulse signal back.

(3) IF the signal comes back, through high level, time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340m/s) / 2 [10].

Figure 3.2 HC-SR04 ultrasonic sensor

The HC-SR04 ultrasonic sensor action according to the electrical parameters is shown is Table 3.1. Wire connecting direct as following:

- 5V power supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Table 3.1 electrical parameters of HC-SR04 ultrasonic sensor

Working voltage	DC 5V
Working current	15 mA
Working frequency	40Hz
Max range	4m
Min range	2 cm
Measuring angle	15 degree
Trigger input signal	10 µs TTL pulse
Echo output signal	Input TTL lever signal and the range in proportion
Dimension	450*20*15 mm

3.2.1.3 HC - SR04 Timing diagram

The timing diagram of HC-SR04 is shown in figure 3.3. To start the measurement, Trig of SR04 must receive a pulse of high (5V) for at least 10µs, this will initiate the sensor to transmit out 8 cycles of ultrasonic burst at 40 kHz and wait for the reflected ultrasonic burst. When the sensor detected ultrasonic from the receiver, it will set the Echo pin to high (5V) and delay for a period (width) which is proportional to distance. We can calculate the range = high level time * velocity (340meter/sec) / 2; it is suggested to use over 60ms measurement cycle, in order to prevent trigger signal from the echo signal [10].

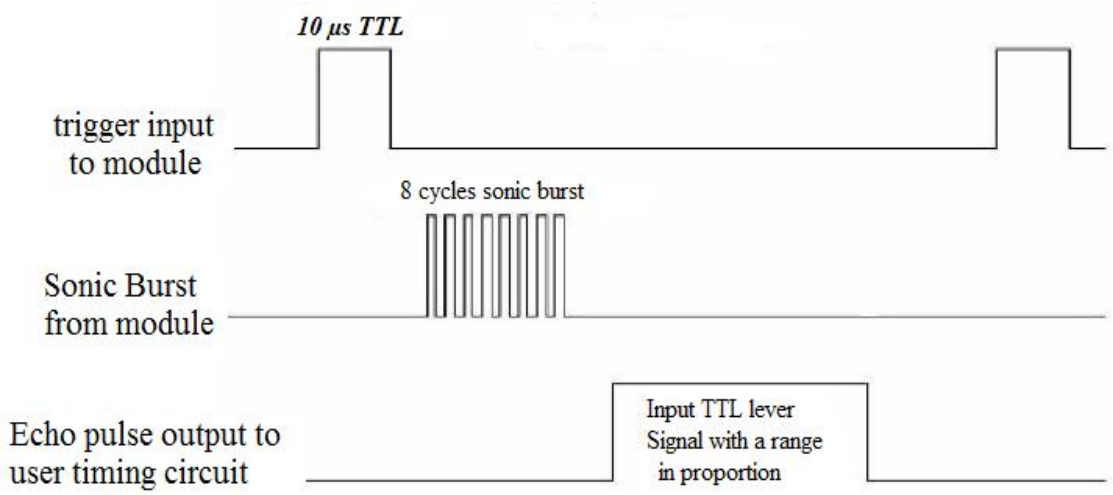

Figure 3.3 HC-SR04 Timing diagram

3.2.2 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The pin diagram of Arduino Uno is shown in Figure 3.4. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter."Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

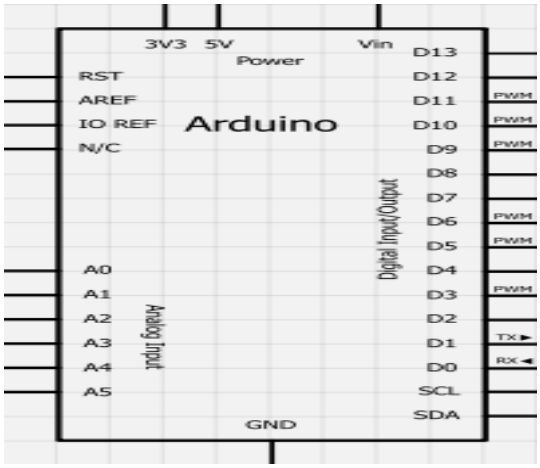

Figure 3.4 pin diagram of Arduino Uno

3.2.2.1 Pin Descriptions

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- The Atmega328 has 32 KB of flash memory for storing code, It has also 2 KB of SRAM and 1 KB of EEPROM.
- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3**.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.

- **PWM:** 3, 5, 6, 9, 10, and 11**.** Provide 8-bit PWM output.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13**.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I2C:** 4 (SDA) and 5 (SCL). Support I2C (TWI) communication.
- **AREF:** Reference voltage for the analog inputs.
- **Reset:** Bring this line low to reset the microcontroller. Typically used to add a reset button to Shields which block the one on the board.

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

3.2.2.2 Arduino Advantages

There are a lot of advantage of Arduino can be mentioned below:

1. Ready to Use: The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5v regulator, a burner, an oscillator, a micro-controller, serial communication interfaces, LED

and headers for the connections. You don’t have to think about programmer connections for programming or any other interface.

2. Examples of codes: it's have a library of examples present inside the software of Arduino.

3. Effortless functions: During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability.

4. large community: there are many forums present on the internet in which people are talking about the Arduino, Engineers, hobbyists and professionals are making their projects through Arduino. You can easily find help about everything.

3.2.2.3 Arduino Disadvantages

Despite it’s a lot of advantage, The Arduino has the following disadvantage:

1. Structure: the structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible. But with the big structures of Arduino we have to stick with big sized PCB’s.

2. Cost: the most important factor which you cannot deny is cost. This is the problem which every hobbyist, Engineer or professional has to face.

3. Easy to use: in my opinion, if you started your journey of micro-controllers with Arduino then it will be very difficult for you to make the complex intelligent circuitries in future.

3.2.3 RF Modules

The RF module operates at Radio Frequency. The corresponding frequency range varies between 30 kHz & 300 GHz. In this RF system, the digital data is

represented as variations in the amplitude of carrier wave. This kind of modulation is known as Amplitude Shift Keying (ASK). Transmission through RF is better than IR (infrared) because of many reasons. Firstly, signals through RF can travel through larger distances making it suitable for long range applications. Also, while IR mostly operates in line-of-sight mode, RF signals can travel even when there is an obstruction between transmitter and receiver. Next, RF transmission is more strong and reliable than IR transmission. RF communication uses a specific frequency unlike IR signals which are affected by other IR emitting sources. This RF module comprises of an RF Transmitter and an RF Receiver. The transmitter/receiver (Tx/Rx) pair operates at a frequency of 434 MHz. The pin diagram of RF modules (Tx/Rx) is shown in Figure 3.5. An RF transmitter receives serial data and transmits it wirelessly through RF through its antenna connected at pin4. The transmission occurs at the rate of 1Kbps - 10Kbps.The transmitted data is received by an RF receiver operating at the same frequency as that of the transmitter.

The RF module is often used along with a pair of encoder/decoder. The encoder is used for encoding parallel data for transmission feed while reception is decoded by a decoder. HT12E- HT12D, HT640-HT648, etc. are some commonly used encoder/decoder pair ICs.

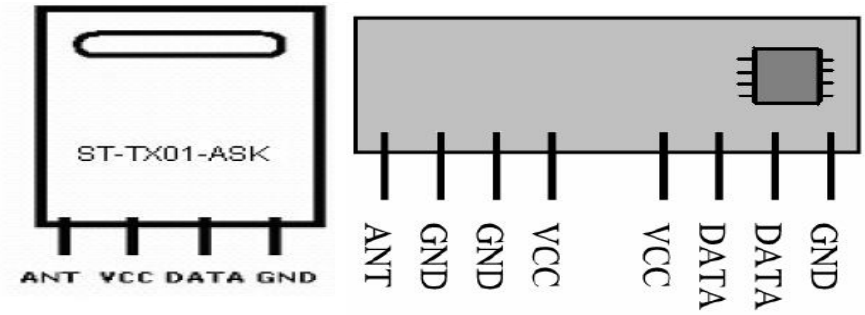

Figure 3.5 pin diagram of RF modules (a) Tx (b) Rx

3.2.3.1 HT12E Encoder

HT12E is an encoder integrated circuit of 212 series of encoders. They are paired with 212 series of decoders for use in remote control system applications. It is mainly used in interfacing RF and infrared circuits. The chosen pair of encoder/decoder should have same number of addresses and data format. Simply put, HT12E converts the parallel inputs into serial output. It encodes the 12 bit parallel data into serial for transmission through an RF transmitter. These 12 bits are divided into 8 address bits and 4 data bits. HT12E has a transmission enable pin which is active low. When a trigger signal is received on TE pin, the programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium. HT12E begins a 4-word transmission cycle upon receipt of a transmission enable. This cycle is repeated as long as TE is kept low. As soon as TE returns to high, the encoder output completes its final cycle and then stops. The pin diagram of HT12E Encoder is shown in Figure 3.6

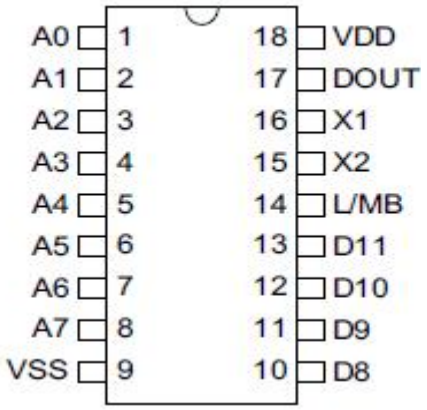

Figure 3.6 pin diagram of HT12E Encoder

3.2.3.2 HT12D Decoder

HT12D is a decoder integrated circuit that belongs to 212 series of decoders. This series of decoders are mainly used for remote control system applications, like burglar alarm, car door controller, security system etc. It is mainly provided

to interface RF and infrared circuits. They are paired with 212 series of encoders. The chosen pair of encoder/decoder should have same number of addresses and data format. In simple terms, HT12D converts the serial input into parallel outputs. It decodes the serial addresses and data received by, say, an RF receiver, into parallel data and sends them to output data pins. The serial input data is compared with the local addresses three times continuously. The input data code is decoded when no error or unmatched codes are found. A valid transmission in indicated by a high signal at VT pin. HT12D is capable of decoding 12 bits, of which 8 are address bits and 4 are data bits. The data on 4 bit latch type output pins remain unchanged until new is received. The pin diagram of HT12D Decoder is shown in Figure 3.7

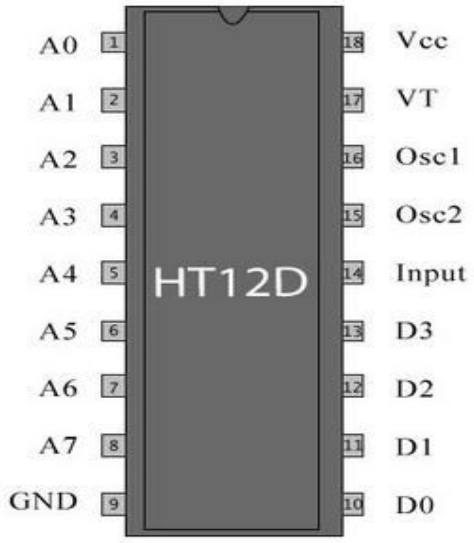

Figure 3.7 pin diagram of HT12d Decoder

3.2.4 Liquid Crystal Display

Nowadays, most commonly used LCDs found in the market are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 characters, where as LCDs supporting more than 80 characters make use of 2 HD44780 controllers. In my research I have used LCD (16*2) to display the water level. The block diagram of LCD (16*2) is shown in Figure 3.8

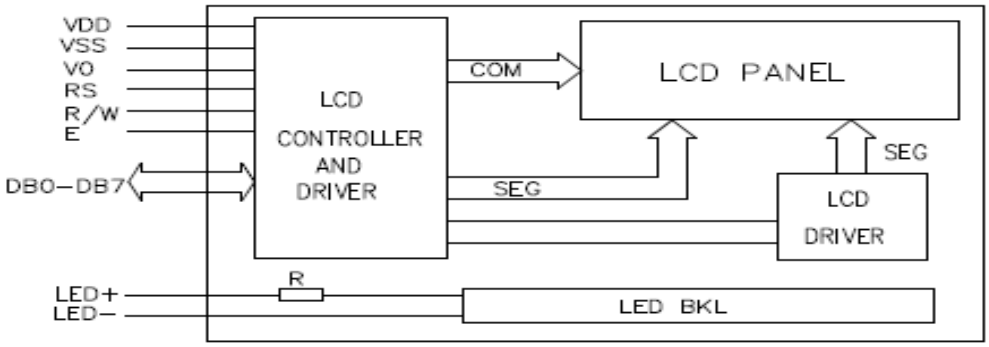

Figure 3.8 block diagram of LCD (16*2)

3.2.5 Light Emitting Diode (LED)

A LED is a semiconductor diode that emits light when an electrical voltage is applied in the forward direction of the device. LED is shown in Figure 3.9. When LED anode lead has a voltage that is more positive than its cathode lead by at least the LED forward voltage drop thus current flows. Electrons are able to recombine with holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the colour of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor.

In this research, the LEDs will be used as indicators at the prototype; thus, I have used one green LED, one yellow LED and one red LED which indicate the water level.

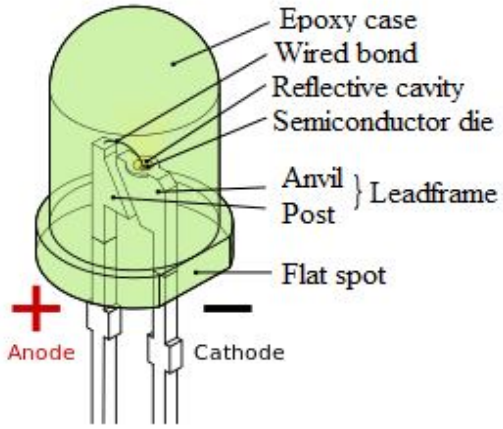

Figure 3.9 LED

3.2.6 Buzzer

A buzzer is an audio signalling device which may be mechanical or electronic. It can be used as an alarm, timer or confirmation of user input. The sound output from the buzzer may be continuous or intermittent. As the output is typically at least 75dB, it will provide sufficient sound aid for the user. Thus, I have used a buzzer in my system to give an alarm if the water level changed rapidly and considerably dangerous. Figure 3.10 is shown the buzzer.

Figure 3.10 The Buzzer

3.3 Software Used

The software development begins when the hardware implementation is close to completion. In order to determine if the hardware design works as expected, programming codes are required to test the hardware design modules individually. For instance, programming code to test solely on the LCD if it is functioning or not. Upon the completion of the hardware design, full program instructions will be planned out, written and programmed into the Arduino microcontroller. Figure 3.11 is shown the software development flowchart as a guideline.

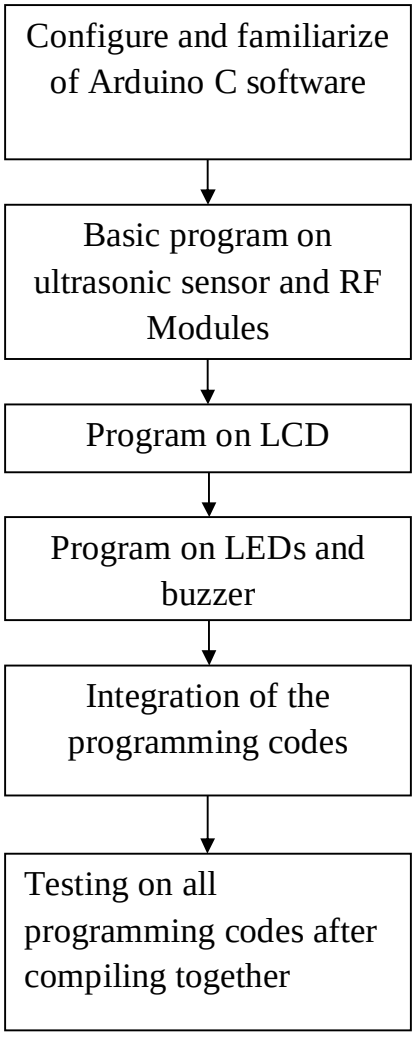

Figure 3.11 The software development flowchart

3.3.1 Arduino (IDE)

The IDE (Integrated Development Environment) is a special program running on your computer that allows you to write sketches for the Arduino board in a simple language. If you click the triangular Verify button on the Arduino IDE, this actually attempts to compile the C that you have written without trying to send the code to the Arduino IDE. After Verify the Arduino IDE has looked at your efforts at writing code and found them to be acceptable. It tells you this by saying “Done Compiling” and reporting the size of the sketch to you 450 bytes. The IDE is also telling you that the maximum size is 32,256 bytes, so you still have lots of room to make your sketch bigger. When you press the button that uploads the sketch to the board: the code that you have written is translated into the C language (which is generally quite hard for a beginner to use), and is passed to the avr-gcc compiler, an important piece of open source software that makes the final translation into the language understood by the microcontroller. This last step is quite important, because it’s where Arduino makes your life simple by hiding away as much as possible of the complexities of programming microcontrollers.

The programming cycle on Arduino is basically as follows:

1- Plug your board into a USB port on your computer.

2- Write a sketch that will bring the board to life.

3- Upload this sketch to the board through the USB connection and wait a couple of seconds for the board to restart.

4- The board executes the sketch that you wrote.

3.4 Summary

In this chapter we explained it the overall components that have been used to complete the system, and how important this components, also we Spoke about the software that is used to programs the Arduino.

Chapter Four

Hardware Design and its associated Software

4.1 Introduction

This chapter discusses the hardware implementation of the system. It will be divided into various parts which are essential to complete the system.

The following is the block diagram and a general scheme of the system it represents the connections between the Arduino and all the part of system.

4.2 The whole Hardware Design and Implementation

The block diagram of the whole design is shown in Figure 4.1. The main blocks are Arduino, ultrasonic sensor, RF module, Encoder, Decoder, LCD, LEDs and buzzer.

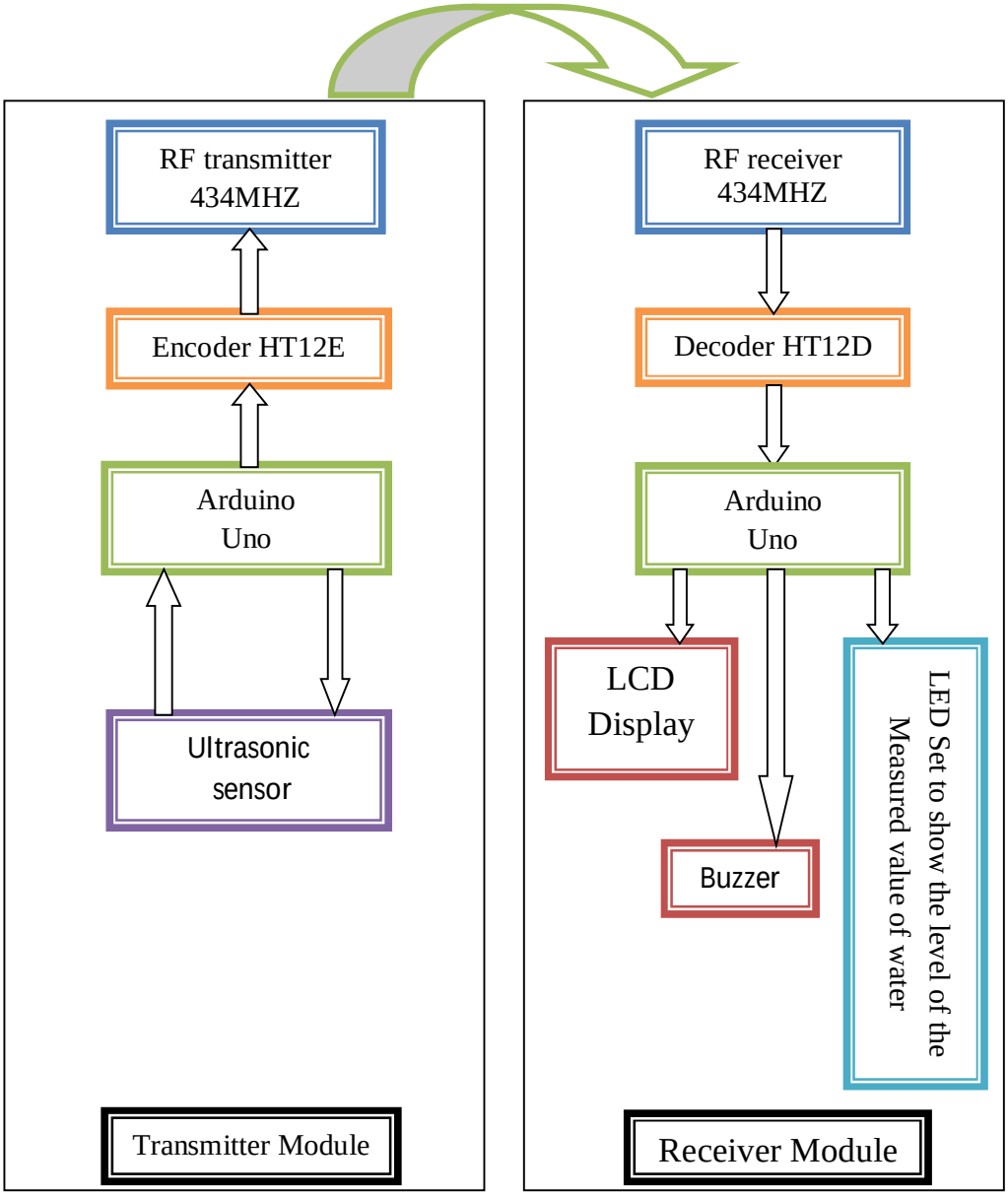

Figure 4.1 block diagram of the monitoring system

The block diagram of the monitoring system consist of two parts, transmitter and receiver modules, transmitter module detect water level automatically, ultrasonic sensor is detect the distance between sensor and water surface [1]. Then the data is passed to the Arduino to perform the calculation, Here HT12E and HT12D have been used as encoder and decoder respectively. The encoder converts the parallel inputs (from the Arduino) into serial set of signals. These

signals are serially transferred through RF to the reception point. The decoder is used after the RF receiver to decode the serial format and retrieve the original signals as outputs. These outputs are passed to another Arduino to display the water level in LCD a set of LEDs to show that the current value of the water level located in which area, then if water level is changed rapidly and considerably dangerous, the buzzer will be activated.

4.2.1 Circuit follow up

The schematic circuit diagram of the system is shown in Figure 4.2 and figure 4.3 for the transmitter module and the receiver module in series. It represents the connections between the Arduino and all the parts of the system. From this schematic, we can easily describe the functionality of the product developed.

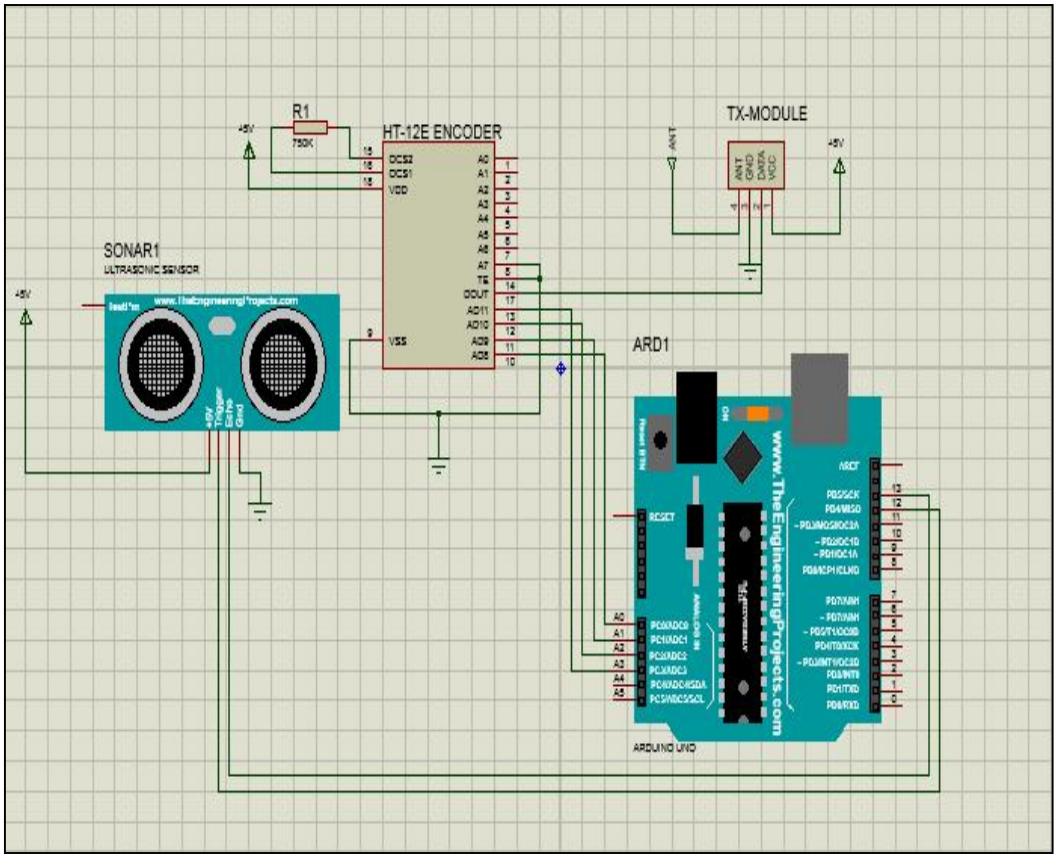

Figure 4.2 schematic circuit diagram of the system (transmitter module)

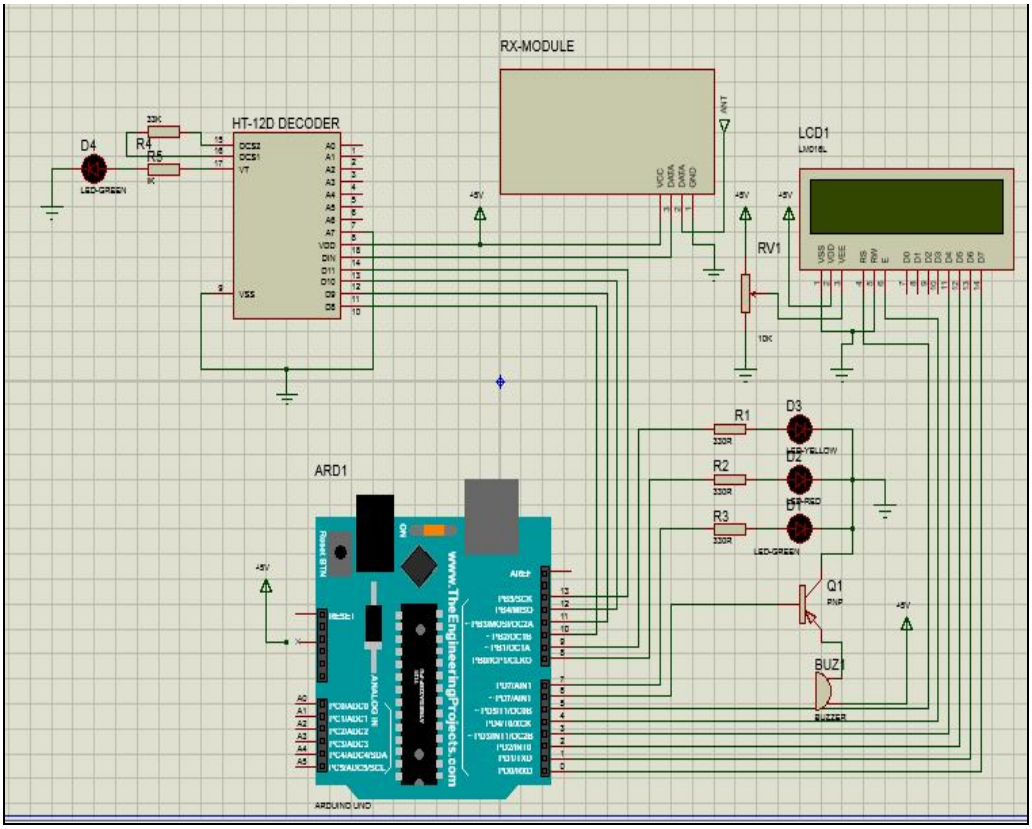

Figure 4.3 schematic circuit diagram of the system (receiver module)

The Encoder IC (HT12E) receives parallel data in the form of address bits and control bits. The control signals from Arduino along with 8 address bits constitute a set of 12 parallel signals. The encoder HT12E encodes these parallel signals into serial bits. Transmission is enabled by providing ground to pin (14) which is active low. The serial data is fed to the RF transmitter through pin (17) of HT12E. Transmitter, upon receiving serial data from encoder IC (HT12E), transmits it wirelessly to the RF receiver. The receiver, upon receiving these signals, sends them to the decoder IC (HT12D) through pin2. The serial data is received at the data pin (14) of HT12D. The decoder then retrieves the original parallel format from the received serial data. When no signal is received at data pin of HT12D, it remains in standby mode and consumes very less current (less than 1μA) for a voltage of 5V. When signal is received by receiver, it is given to pin (14) of HT12D. On reception of signal, oscillator of HT12D gets activated.

IC HT12D then decodes the serial data and checks the address bits three times. If these bits match with the local address pins (1-8) of HT12D, then it puts the data bits on its data pins (10-13) and makes the VT pin high. An LED is connected to VT pin (17) of the decoder. This LED works as an indicator to indicate a valid transmission. The corresponding output is thus generated at the data pins of decoder IC.

The Arduino in this research was Arduino Uno and I used two Arduino Uno one in the transmitter module and the another in the receiver module:

Arduino in the transmitter module it connected pin (12), pin (13) to the trigger, echo pins of ultrasonic sensor respectively, and pins (A0,A1,A2,A3) to the pins (10,11,12,13) of HT12E Encoder respectively, and 5v,Gnd of Arduino to VCC and Gnd of ultrasonic sensor.

Arduino in the receiver module it connected 5v, Ground (Gnd) of Arduino to pins (1,2,15,16) and pins (0,1,2,3,4,5) to pins (14,13,12,11,6,4) of LCD respectively. Also pin (6) to the base of transistor, pins (7,8,9) to resisters (1k ohms) to avoid damage the LEDs, pins (10,11,12,13) to pins (10,11,12,13) of HT12D Decoder.

4.3 The system flowchart

Figure 4.4 is the flowchart can be describes as follows: When the water level is detected by ultrasonic sensor it will be sended via the receiver and then check the value that is received in three cases and display it’s in LCD and explain the case by LEDs.

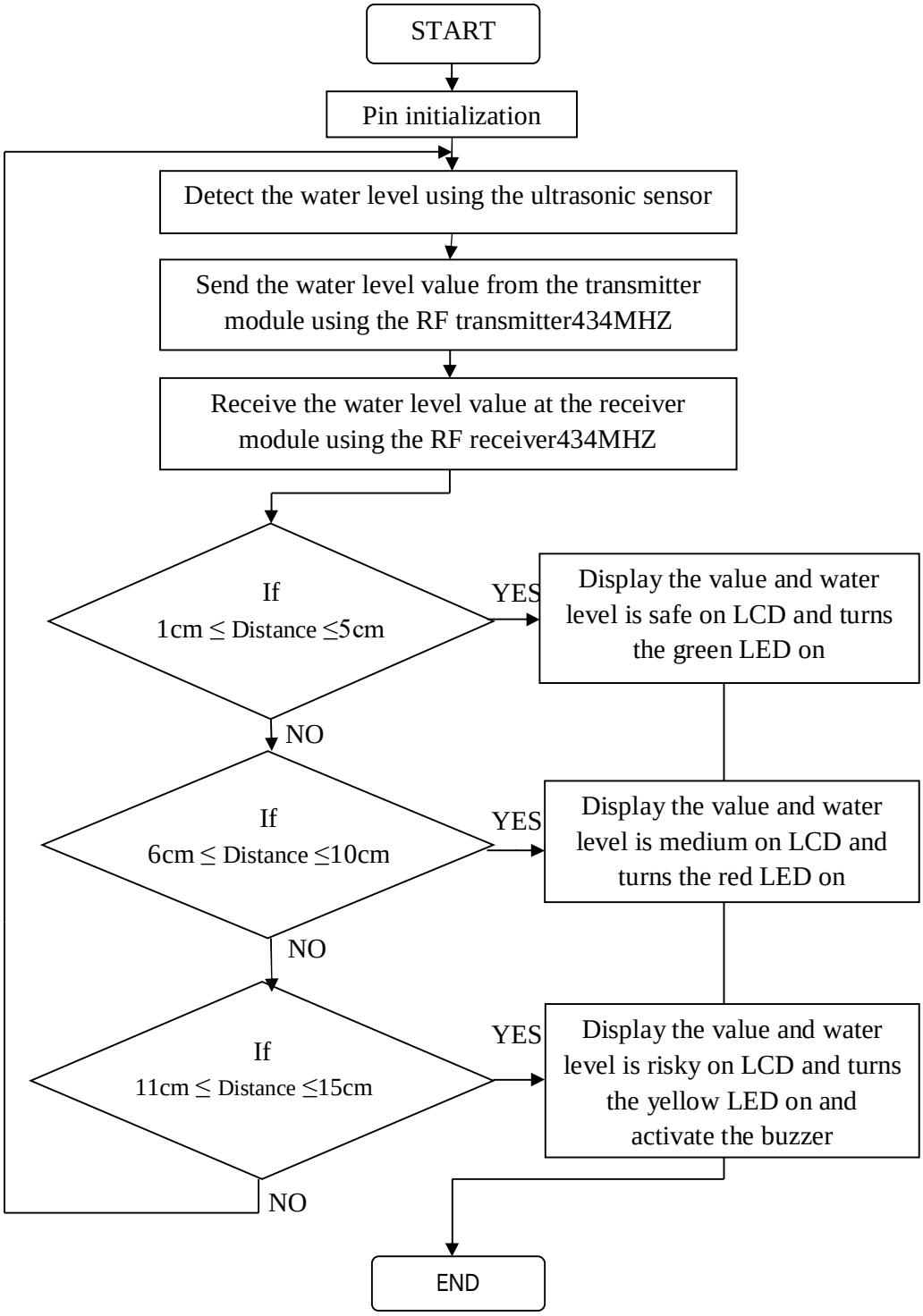

Figure 4.4 the system flowchart

4.4 Summary

We exposed in this chapter the block diagram of the whole design and explained all the parts in it, also followed up the schematic circuit diagram of the system, and lastly we described the system flowchart.

Chapter Five

Results and Discussion

5.1 Introduction

After building every component and modules on prototype board, it is important to test the individual circuit as well as the functionality of overall system. The whole Integrated System has put to a functionality test to confirm that all the hardware and software coding and interfacing are work well.

The whole verifed in this research was written in C language. It was verify by (Arduino IDE) and then uploaded by USB to the Arduino Board. The program, computer and hardware prototype were properly connected before the program is uploaded. It is important to make sure that the components are connected correctly on the board to avoid disruption during the testing and debugging.

5.2 The System Integration

The figures 5.1 and 5.2 next page show the full set up of the whole system prototype for both transmitter and receiver module.

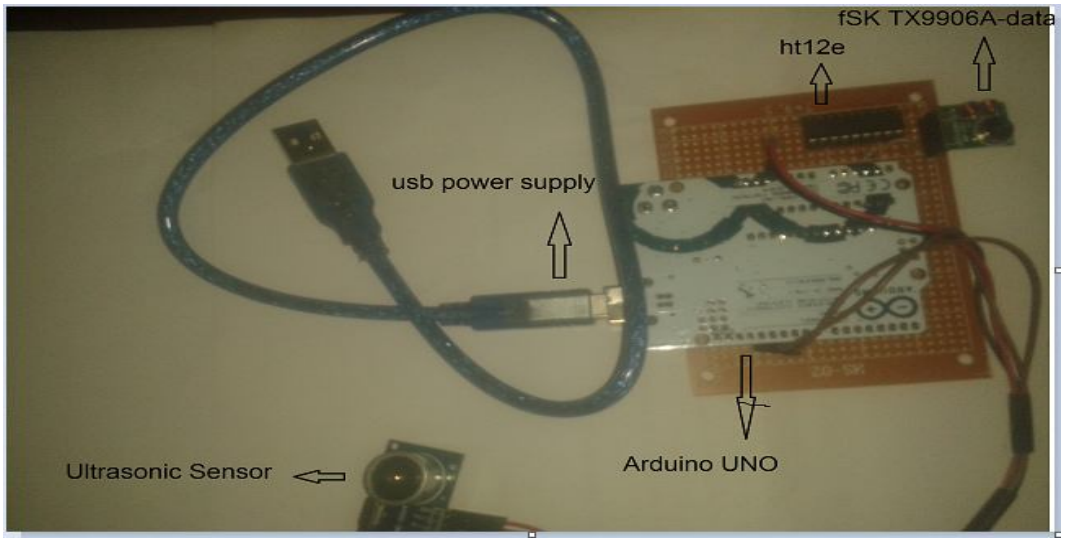

Figure 5.1 Transmitter module prototype

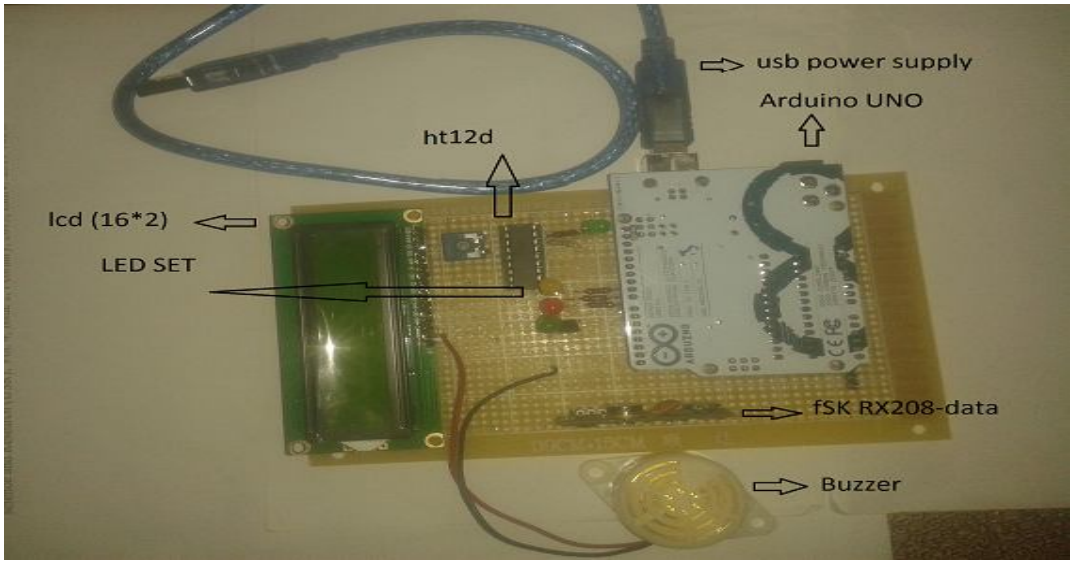

Figure 5.2 Receiver module prototype

5.3 The System results and discussion

The system was built to automatically sense the water level using the HC-SR04 ultrasonic sensor at the transmitter module and then send those values through the RF transmitter and then receive those values and show them using the LCD (16*2) at the receiver module.

The calculation of distance is performed by high level language program and resides in the Arduino microcontroller. We also have a set of LEDs to show that

the current value of the water level located in which area. We determined three cases for the water level, the first case from (1cm to 5cm), the second case from (6cm to 10cm), and third case from (11cm to 15cm). We indicated these cases below:

Case 1: if the distance from object is (1cm to 5cm), then display the water level value in LCD and the level is safe, then the LED with the green colour will set ON as we show in figure 5.3 below.

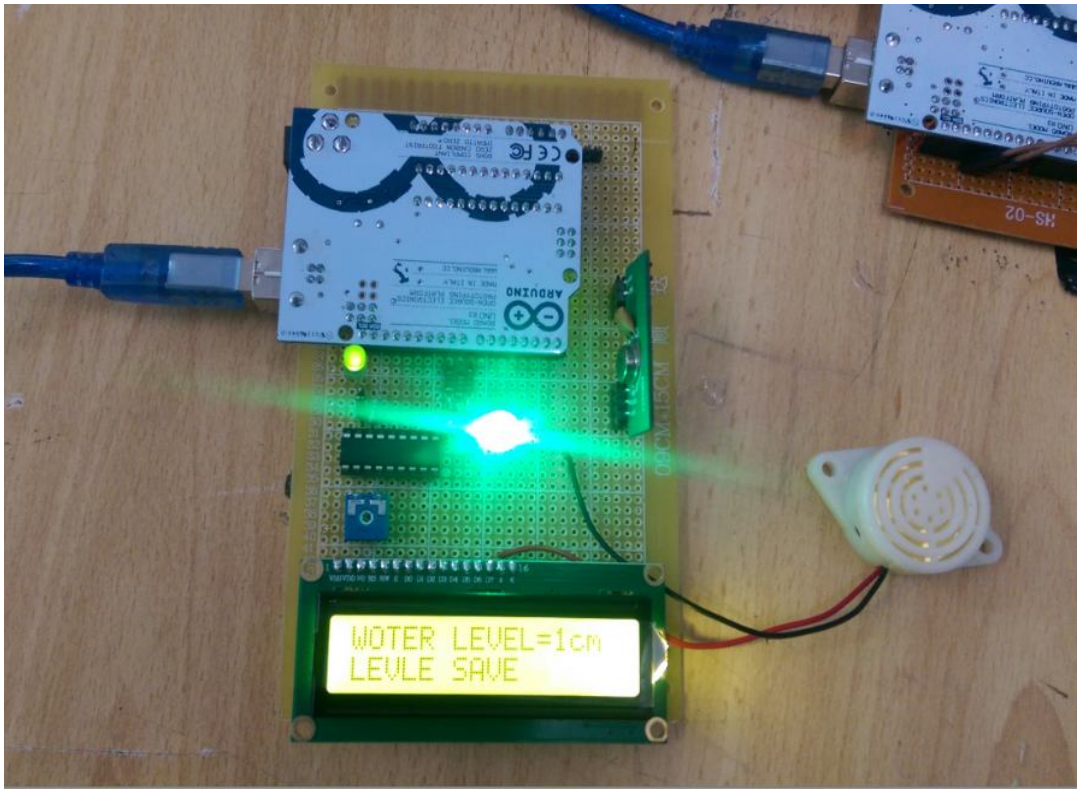

Figure 5.3 Case 1: the water at the safe level

Case 2: if the distance from object is (6cm to 10cm), then display the water level value in LCD and the level is medium, then the LED with the Red colour will set ON as we show in figure 5.4 next page.

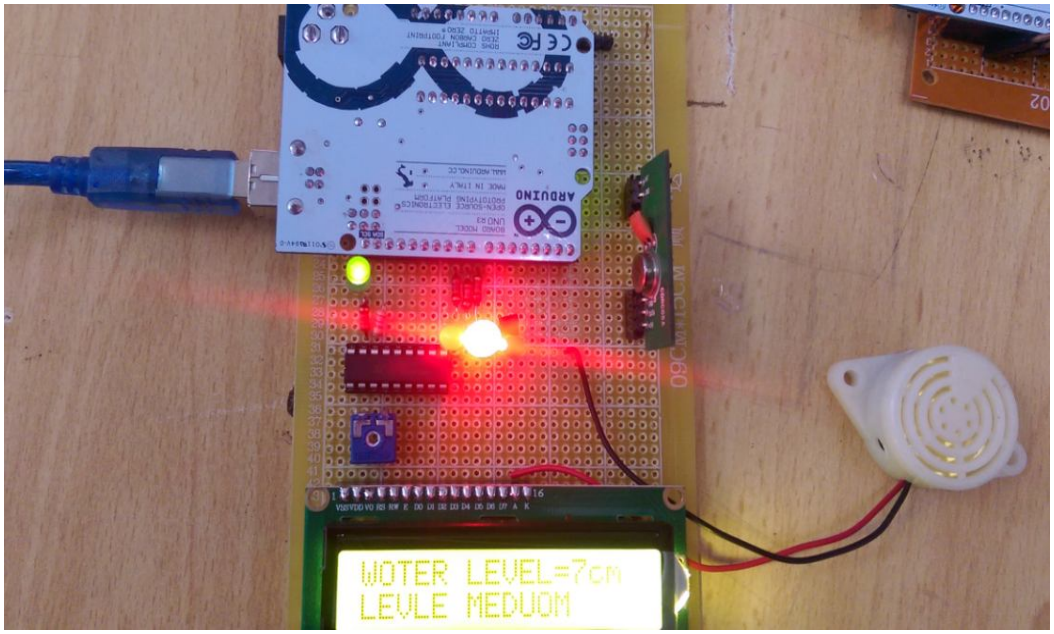

Figure 5.4 Case 2: the water at the medium level

Case 3: if the distance from object is (11cm to 15cm), then display the water level value in LCD and the level is risky, then the LED with the yellow colour will set ON and the buzzer will be activated as we show in figure 5.5 next page.

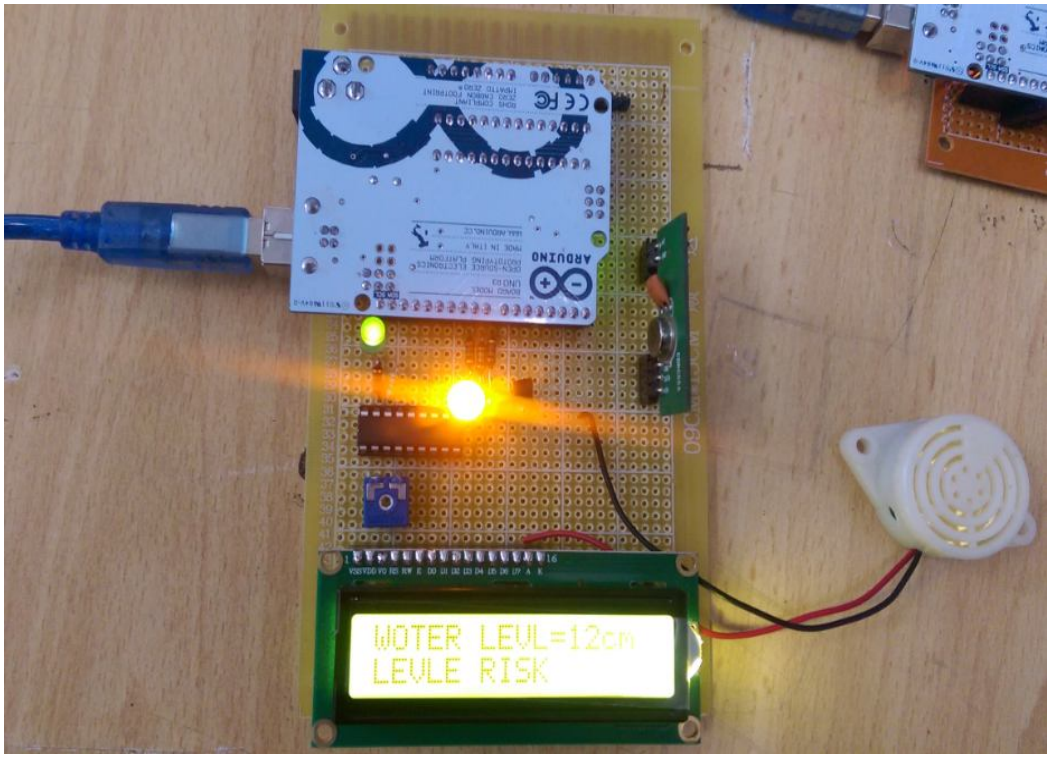

Figure 5.5 Case 3: the water at the risky level

Also we conducted experiment to measure the water level and that by use two type of measure, by normal method and this called (gradient method) measured by ruler, and the second method is used the ultrasonic sensor waves, we worked compare between these methods to determine the resolution of measure, and then presented this measure in table 5.1 as shown in the next page.

Table 5.1 Comparison between values measured by HC-SR04 ultrasonic sensor and reference values by ruler.

Reference value by ruler (milli meter)	Measured value by ultrasonic sensor (milli meter)
10	13
20	22
30	31
40	41
50	50
60	60
70	69
80	79
90	92
100	101
110	109
120	121
130	130
140	141
150	151

Max Deviation value: 3mm

Note: the distance between ultrasonic sensor and object not less than 2cm we dissolve it by raise sensor fixed place to increase the distance between high water level and sensor surface.

Ultrasonic waves are prone to absorption and refraction, like any types of waves, it will have impacts if using with water on the contrary used with solids, with solid materials it will better than an accuracy of the water.

5.4 Summary

In this chapter, firstly we showed the full set up of the whole system prototype, and secondly we explained the three cases for the water level.

Chapter Six

Conclusion and Recommendations

6.1 CONCLUSION

Prototype of Water Level Detection System has been tested and reasonably good performance is shown based on the test results. One of the main contributions of this research is the ultrasonic sensor calibration by adjusting calculations of distance based on actual data. Testing need to be carried out for the real fluctuated water surface condition to get the system performance in the real circumstances.

The water level data is successfully displayed locally and remotely, therefore this prototype can be used as part of the bigger system, such as river flow management system which controls the stream to minimize the flood. The receiver acts as a water level data feeder which can transmit the data remotely to the server if we use the computer as a part of receiver module, therefore more sophisticated system can be developed to display and analyze time series water level data, instead of only displaying the current water level data.

The hardware part of the research gave me a chance to understand Arduino, ultrasonic sensors, radio frequency modules and some other components and their functions in industrial applications. I also had the chance to practice and exercise on circuit designing, electronic prototyping, PCB designing. (its pin diagram), mounting of electrical components using soldering process and interfacing of the hardware circuit with the computer.

Software part helped me to have a better understanding about the embedded system programming and Arduino C programming language, circuit designing, troubleshooting and project management skills which are vital in my vocational

career. This prototype is developed and tested with satisfactory working condition and all in all it works correctly for what it was designed for.

I have completed the research on time and matched the research objectives. It helps me to improve my hardware design skills, programming skills and planning skills. I am sure that the experience that I have obtained from this research will help me throughout my future career.

6.2 Recommendations

This system is so open to be improved to provide better performance, and here are some suggestions to be included in any Recommendation:

- The better radio frequency module can be used, in order to reach longer distances.
- This tool can be developed to measure water depth using the other types of ultrasonic sensors such as SRF02 or SRF08.
- The better receiver module can be developed to receive the water level data from multiple transmitter modules.
- Can be applied this work to follow the river in order to avoid flooding.

References

[1] Abubakr Rahmtalla Abdalla Mohamed, Real Time Wireless Flood Monitoring System using Ultrasonic Waves, International Journal of Science and Research, vol. 3, pp.2319-7064, August 2014.

[2] patented Apr.21, 1942 United States patent "Flaw Detecting Device and Measuring Instrument", Floyed A.Firestone, ann Arbor, mich. Application May 27, 1940.

[3] United States patent “Ultrasonic Testing Apparatus and Method”. James F.Mcnulty, westbury, N.y, assignor to Automation industries, incorporated, Elsegundo, California field Dec 21,1962.

[4] S.Jatmiko, A.B.Mutiara, Prototype of Water Level Detection System With Wireless, Journal of Theoretical and Applied Information Technology, Vol. 37, pp. 1817-3195, March 2012.

[5] Jirapon Sunkpho, and Chaiwat Ootamakorn, Real-time flood monitoring and warning system, Songklanakarin Journal of Science and Technology, pp. 227-235, 29 March 2011.

[6] Samarth Viswanath, Marco Belcastro, John Barton, Brendan O’Flynn, Low-Power Wireless Liquid Monitoring System Using Ultrasonic, Sensors,International Journal on Smart Sensing and Intelligent System, Vol. 8, pp. 1178-5608, March 2015.

[7] Mahmoud Meribout, Mohamed Habli, Ahmed Al-naamany, A New Ultrasonic-based device for Accurate Measurement of Oil, Emulsion, and Water Levels in Oil Tanks, Internal Report, May 2003.

[8] Valeriy kyrynyuk, ben kropf, balaji M V, “Automotive ultrasonic Distance Measurement for Park Assist System”, AN76530.

[9] www.micropik.com.

Appendixes:

Appendix (A): Program code

The transmitter module code

```
#include <Wire.h>

#include <LiquidCrystal.h>

const int buttonPin = 13;

const int button1Pin = 12;

const int button2Pin = 11;

const int button3Pin = 10;

int a=0;

int buttonState = 0;

int button1State = 0;

int button2State = 0;

int button3State = 0;

LiquidCrystal lcd(5, 4, 3, 2, 1, 0);

void setup() {

  lcd.begin(16, 2);

     pinMode(7, OUTPUT);

     pinMode(8, OUTPUT);
```

```
    pinMode(9, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(buttonPin, INPUT);
    pinMode(button1Pin, INPUT);
    pinMode(button2Pin, INPUT);
    pinMode(button3Pin, INPUT);
}
void loop() {
  buttonState = digitalRead(buttonPin);
   button1State = digitalRead(button1Pin);
    button2State = digitalRead(button2Pin);
    button3State = digitalRead(button3Pin);
if (buttonState == HIGH && button1State == HIGH && button2State == HIGH
&& button3State == HIGH)
{
    lcd.print("WOTER LEVEL=1cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE SAVE        ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
```

```
    digitalWrite(7, HIGH);
     digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      delay(1000);
}
if (buttonState == HIGH && button1State == HIGH && button2State == HIGH
&& button3State == LOW)
{
     lcd.print("WOTER LEVEL=2cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE SAVE         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
     digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      delay(1000);

}
```

```
if (buttonState == HIGH && button1State == HIGH && button2State == LOW
&& button3State == HIGH)
{
     lcd.print("WOTER LEVEL=3cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE SAVE         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
     digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      delay(1000);
}
if (buttonState == HIGH && button1State == LOW && button2State == HIGH
&& button3State == HIGH)
{
     lcd.print("WOTER LEVEL=4cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE SAVE         ");
   lcd.setCursor(0, 0);
```

```
  digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
     digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      delay(1000);
}
if (buttonState == LOW && button1State == HIGH && button2State == HIGH
&& button3State == HIGH)
{
     lcd.print("WOTER LEVEL=5cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE SAVE         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, HIGH);
     digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      delay(1000);
}
```

```
if (buttonState == LOW && button1State == LOW && button2State == LOW &&
button3State == LOW)
{
     lcd.print("               ");
   lcd.setCursor(0, 1);
   lcd.print("                 ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, LOW);
      digitalWrite(9, LOW);
      delay(1000);
}
////////////////////////////////////////////////////////////////////////////////
//////////////////////
if (buttonState == LOW && button1State == HIGH && button2State == HIGH
&& button3State == HIGH)
{

     lcd.print("WOTER LEVEL=6cm");
```

```
   lcd.setCursor(0, 1);
   lcd.print("LEVLE MEDUOM         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
      delay(1000);
}
if (buttonState == HIGH && button1State == LOW && button2State == HIGH
&& button3State == HIGH)
{
     lcd.print("WOTER LEVEL=7cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE MEDUOM         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
```

```
      delay(1000);
}
if (buttonState == HIGH && button1State == HIGH && button2State == LOW
&& button3State == LOW)
{
     lcd.print("WOTER LEVEL=8cm");
   lcd.setCursor(0, 1);
  lcd.print("LEVLE MEDUOM         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
      delay(1000);
}
if (buttonState == HIGH && button1State == LOW && button2State == LOW &&
button3State == HIGH)
{
     lcd.print("WOTER LEVEL=9cm");
   lcd.setCursor(0, 1);
```

```
    lcd.print("LEVLE MEDUOM          ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
      delay(1000);
}
if (buttonState == HIGH && button1State == LOW && button2State == LOW &&
button3State == LOW)
{
     lcd.print("WOTER LEVL=10cm");
   lcd.setCursor(0, 1);
      lcd.print("LEVLE MEDUOM          ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
      delay(1000);
```

```
}
////////////////////////////////////////////////////////////////////////////////
//////////////////////////
////////////////////////////////////////////////////////////////////////////////
//////////////////////////
if (buttonState == LOW && button1State == HIGH && button2State == HIGH
&& button3State == LOW)
{
     lcd.print("WOTER LEVL=11cm");
   lcd.setCursor(0, 1);
   lcd.print("LEVLE RISK         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, LOW);
    digitalWrite(7, LOW);
     digitalWrite(8, LOW);
      digitalWrite(9, HIGH);
      delay(1000);
}
if (buttonState == HIGH && button1State == LOW && button2State == HIGH
&& button3State == LOW)
{
```

```
     lcd.print("WOTER LEVL=12cm");
    lcd.setCursor(0, 1);
    lcd.print("LEVLE RISK         ");
    lcd.setCursor(0, 0);
   digitalWrite(6, LOW);
     digitalWrite(7, LOW);
      digitalWrite(8, LOW);
       digitalWrite(9, HIGH);
       delay(1000);
}
if (buttonState == LOW && button1State == LOW && button2State == HIGH &&
button3State == LOW)
{
      lcd.print("WOTER LEVL=13cm");
    lcd.setCursor(0, 1);
    lcd.print("LEVLE RISK         ");
    lcd.setCursor(0, 0);
   digitalWrite(6, LOW);
     digitalWrite(7, LOW);
      digitalWrite(8, LOW);
```

```
      digitalWrite(9, HIGH);
      delay(1000);
}
if (buttonState == LOW && button1State == HIGH && button2State == LOW &&
button3State == LOW)
{
     lcd.print("WOTER LEVL=14cm");
   lcd.setCursor(0, 1);
    lcd.print("LEVLE RISK         ");
   lcd.setCursor(0, 0);
  digitalWrite(6, HIGH);
    digitalWrite(7, LOW);
     digitalWrite(8, LOW);
      digitalWrite(9, HIGH);
      delay(1000);

}
if (buttonState == LOW && button1State == LOW && button2State == LOW &&
button3State == HIGH)
{
```

```
      lcd.print("WOTER LEVL=15cm");
    lcd.setCursor(0, 1);
      lcd.print("LEVLE RISK         ");
    lcd.setCursor(0, 0);
   digitalWrite(6, HIGH);
     digitalWrite(7, LOW);
      digitalWrite(8, LOW);
       digitalWrite(9, HIGH);
       delay(1000);
}
if (buttonState == LOW && button1State == LOW && button2State == LOW &&
button3State == LOW)
{
      lcd.print("               ");
    lcd.setCursor(0, 1);
    lcd.print("                  ");
    lcd.setCursor(0, 0);
   digitalWrite(6, LOW);
     digitalWrite(7, LOW);
      digitalWrite(8, LOW);
```

```
    digitalWrite(9, LOW);
    delay(1000);
}



}}
```

The receiver module code

```
/* Ping))) Sensor
  This sketch reads a PING))) ultrasonic rangefinder and returns the
  distance to the closest object in range. To do this, it sends a pulse
  to the sensor to initiate a reading, then listens for a pulse
  to return.  The length of the returning pulse is proportional to
  the distance of the object from the sensor.

  The circuit:
      * +V connection of the PING))) attached to +5V
      * GND connection of the PING))) attached to ground
      * SIG connection of the PING))) attached to digital pin 7
  http://www.arduino.cc/en/Tutorial/Ping
```

```
  created 3 Nov 2008
  by David A. Mellis
  modified 30 Aug 2011
  by Tom Igoe
  This example code is in the public domain.
 */
// this constant won't change.  It's the pin number
// of the sensor's output:
const int ledPin =  A0;
const int led1Pin =  A1;
const int led2Pin =  A2;
const int led3Pin =  A3;
const int pingPin = 12;
const int ping1Pin = 13 ;
const int pinjPin = 11;
const int pinj1Pin = 10 ;
void setup() {
  // initialize seri' al communication:
     pinMode(ledPin, OUTPUT);
     pinMode(led1Pin, OUTPUT);
```

```
    pinMode(led2Pin, OUTPUT);
    pinMode(led3Pin, OUTPUT);
     digitalWrite(ledPin, LOW);
   digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, LOW);
  Serial.begin(9600);
}
void loop()
{
  // establish variables for duration of the ping,
  // and the distance result in inches and centimeters:
  long duration, inches, cm;
    long duration1, inches1, cm1;
  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
```

```
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

// The same pin is used to read the signal from the PING))): a HIGH
// pulse whose duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
pinMode(ping1Pin, INPUT);
duration = pulseIn(ping1Pin, HIGH);
 pinMode(pinjPin, OUTPUT);
digitalWrite(pinjPin, LOW);
delayMicroseconds(2);
digitalWrite(pinjPin, HIGH);
delayMicroseconds(5);
digitalWrite(pinjPin, LOW);
// The same pin is used to read the signal from the PING))): a HIGH
// pulse whose duration is the time (in microseconds) from the sending
// of the ping to the reception of its echo off of an object.
pinMode(pinj1Pin, INPUT);
duration1 = pulseIn(pinj1Pin, HIGH);
// convert the time into a distance
```

```
inches = microsecondsToInches(duration);
cm = microsecondsToCentimeters(duration);
 inches1 = microsecondsToInches(duration1);
cm1 = microsecondsToCentimeters(duration1);
Serial.print(cm1);
Serial.print("cm, ");
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
if (cm<2 && cm>0) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, HIGH);
}
else if (cm<3 && cm>1) {
  // turn LED on:
  digitalWrite(ledPin, LOW);
```

```
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, HIGH);
}
else if (cm<4 && cm>2) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, HIGH);
}
else if (cm<5 && cm>3) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, HIGH);
}
 else if (cm<6 && cm>4) {
  // turn LED on:
```

```
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, LOW);
}
////////////////////////////////////////////////
else if (cm<7 && cm>5) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, LOW);
}
else if (cm<8 && cm>6) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, HIGH);
}
```

```
else if (cm<9 && cm>7) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, LOW);
}
else if (cm<10 && cm>8) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, HIGH);
}
 else if (cm<11 && cm>9) {
  // turn LED on:
  digitalWrite(ledPin, HIGH);
   digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, LOW);
```

```
}
//////////////////////////////////////////////

//////////////////////////////////////////////
else if (cm<12 && cm>10) {
  // turn LED on:
  digitalWrite(ledPin, LOW);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, LOW);
}
else if (cm<13 && cm>11) {
  // turn LED on:
  digitalWrite(ledPin, LOW);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, HIGH);
}
else if (cm<14 && cm>12) {
  // turn LED on:
```

```
  digitalWrite(ledPin, LOW);
   digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, LOW);
}
else if (cm<15 && cm>13) {
  // turn LED on:
  digitalWrite(ledPin, LOW);
   digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, HIGH);
     digitalWrite(led3Pin, LOW);
}
 else if (cm<16 && cm>14) {
  // turn LED on:
  digitalWrite(ledPin, LOW);
   digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, LOW);
     digitalWrite(led3Pin, HIGH);
}
//////////////////////////////////////////////
```

```
  else {
    // turn LED on:
    digitalWrite(ledPin, LOW);
     digitalWrite(led1Pin, LOW);
      digitalWrite(led2Pin, LOW);
       digitalWrite(led3Pin, LOW);
  }
}
long microsecondsToInches(long microseconds)
{
  // According to Parallax's datasheet for the PING))), there are
  // 73.746 microseconds per inch (i.e. sound travels at 1130 feet per
  // second).  This gives the distance travelled by the ping, outbound
  // and return, so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf
  return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
```

```
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```

Appendix (B): RF Modules Pin Functions

Pin Description

Transmitter Module

Pin Number	Function	Name
1	Ground (0V)	GND
2	Serial Data Input Pin	DATA
3	Supply Voltage (5V)	VCC
4	Antenna Output Pin	ANT

Receiver Module

Pin Number	Function	Name
1	Ground (0V)	GND
2	Serial Data Output Pin	DATA
3	Linear Output Pin; Not Connected	NC
4	Supply Voltage (5V)	VCC
5	Supply Voltage (5V)	VCC
6	Ground (0V)	GND
7	Ground (0V)	GND
8	Antenna Input Pin	ANT

Appendix (C): HT12E Encoder Pin Functions

Pin Description

Pin Number	Function	Name
1	8 BIT ADDRESS PINS FOR INPUT	A0
2		A1
3		A2
4		A3
5		A4
6		A5
7		A6
8		A7
9	GROUND (0V)	GROUND
10	4 BIT DATA/ADDRESS PINS FOR INPUT	D0
11		D1
12		D2
13		D3

14	TRANSMISSION ENABLE (ACTIVE LOW)	TE
15	OSCILLATOR OUTPUT	OSC 2
16	OSCILLATOR INPUT	OSC 1
17	VALID TRANSMISSION, ACTIVE HIGH	VT
18	SUPPLY VOLTAGE; 5V (2.4 – 12V)	Vcc

Appendix (D): HT12D Decoder Pin Functions

Pin Description

Pin Number	Function	Name
1	8 BIT ADDRESS PINS FOR INPUT	A0
2		A1
3		A2
4		A3
5		A4
6		A5
7		A6
8		A7
9	GROUND (0V)	GROUND
10	4 BIT DATA/ADDRESS PINS FOR OUTPUT	D0
11		D1
12		D2
13		D3
14	SERIAL DATA INPUT	INPUT
15	OSCILLATOR OUTPUT	OSC 2
16	OSCILLATOR INPUT	OSC 1
17	VALID TRANSMISSION, ACTIVE HIGH	VT
18	SUPPLY VOLTAGE; 5V (2.4 – 12V)	Vcc

Appendix (E): Data sheet of LCD(16*2)

Pin description

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Absolute maximum ratings

Item	Symbol	Standard			Unit
Power voltage	V_{DD}-V_{SS}	0	-	7.0	V
Input voltage	V_{IN}	VSS	-	VDD	
Operating temperature range	V_{OP}	0	-	+50	℃
Storage temperature range	V_{ST}	-10	-	+60	