Design and Implementation of Water pump automation system

A Project Submitted
By

Rajib, Dey	(ID: 10-17225-2)
Sudipto, Roy	(ID: 10-17252-2)
Chowdhury Mohammed , Riajul Islam	(ID: 10-16258-1)
Aditi Anika Wardia	(ID: 10-15542-1)

Under the supervision of

Kamrul Hassan
Lecturer, Faculty of Engineering
American International University - Bangladesh

Electrical and Electronic Engineering Department
Faculty of Engineering
American International University - Bangladesh

Fall Semester 2013-2014

December 2013

American International University - Bangladesh

Design and Implementation of Water pump automation system

A Project submitted to the Electrical and Electronic Engineering Department of the Engineering Faculty, American International University - Bangladesh (AIUB) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

A Project submitted by

Rajib , Dey	(ID: 10-17225-2)
Sudipto , Roy	(ID: 10-17252-2)
Chowdhury Mohammed , Riajul Islam	(ID: 10-16258-1)
Aditi Anika Wardia	(ID: 10-15542-1)

Electrical and Electronic Engineering Department
Faculty of Engineering
American International University - Bangladesh

Fall Semester 2013-204
December 2013

American International University - Bangladesh

DECLARATION

This is to certify that this project and thesis is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. Any material reproduced in this project has been properly acknowledged.

Students' names & Signatures

1. Rajib, Dey
(ID: 10-17225-2)
Dept: Electrical and Electronic Engineering

2. Sudipto, Roy
(ID: 10-17252-2)
Dept: Electrical and Electronic Engineering

3. Chowdhury Mohammed ,Riajul Islam
(ID: 10-16258-1)
Dept: Electrical and Electronic Engineering

4. Aditi Anika Wardia
(ID: 10-15542-1)
Dept: Electrical and Electronic Engineering

APPROVAL

The Project titled "Design and Implementation of Water pump Automation System" has been submitted to the following respected members of the Board of Examiners of the Faculty of Engineering in partial fulfillment of the requirements for the degree of Bachelor of Electrical and Electronic Engineering on December 1, 2013 by the following students and has been accepted as satisfactory.

1. Rajib , Dey	(ID: 10-17225-2)
2. Sudipto , Roy	(ID: 10-17252-2)
3. Chowdhury Mohammed , Riajul Islam	(ID: 10-16258-1)
4. Aditi Anika Wardia	(ID: 10-15542-1)

Kamrul Hassan
Supervisor
Lecturer, Faculty of Engineering
American International University - Bangladesh

Dr. M. Tanseer Ali
External Supervisor
Assistant Professor, Faculty of Engineering
American International University - Bangladesh

Professor Dr. ABM Siddique Hossain
Dean, Faculty of Engineering
American International University - Bangladesh

Dr. Carmen Z. Lamagna
Vice Chancellor
American International University - Bangladesh

ACKNOWLEDGEMENT

First and foremost, we would like to express our special thanks to our supervisor Kamrul Hassan, Lecturer, Faculty of Engineering, American International University - Bangladesh for giving us enormous support, motivation and invaluable advises regarding this thesis. He has been our idol and role model in the last couple of months and we are grateful to the Almighty for giving us the opportunity to learn under such a great supervisor.

Secondly, we would like to express our special thanks to our External Supervisor Dr. M. Tanseer Ali , Assistant Professor, Faculty of Engineering, American International University - Bangladesh for giving us remark and invaluable advises for further extension of this thesis.

We would also like to show gratitude towards our honorable Vice Chancellor Dr. Carmen Z. Lamagna, honorable Dean, Professor Dr. ABM Siddique Hossain, American International University - Bangladesh, for approving our thesis.

We would like to thank our families and friends, without whose patience and support we would have never reached our goal.

TABLE OF CONTENTS

List of Figures

ABSTRACT

Water is one of our most valuable resources. The importance of water in our daily life can not be emphasized enough. Water scarcity is the serious issue in major cities. Wasting of this resource is not acceptable in any way. It is a common problem which is faced by every house owner, that when his tank is empty he has to switch on the motor and switch the motor off when it is full. Every day the water tank on our roof-top over flows and wastes 10-20 or more liters of water before we switch the water pump off. That is why an efficient way to pump the water from the water supply line to the roof-top tank of ours is needed. Due to the busy life it is common that the tank usually overflows without notice. One has to keep on observing his tank water level to switch off the motor once it is switched on. And sometimes this also can happen that the motor coil burns because of absence of water in the sump. So these are the everyday problem that motivated us in coming up with an affordable, automatic water level control system that doesn’t need any attention once it is installed.

In this paper we have discussed about design and implementation of water level control system which is microcontroller based, automatic, cost effective and a reliable system. It uses an Ultrasonic Distance Sensor along with a motor controller circuit which is installed at the tank. Ultrasonic Distance Sensor is basically a transceiver is used for detecting the water level in the water tank. It is completely automated with the help of a microcontroller. Also it doesn’t touch the water and stays atleast a few inches over it , so there is no chance of any kind of germ contamination from the sensor to the water or any kind of damage to the device itself due to it being submerged in the water . While creating this project we took special care so that the sensor and the device never touches the water , that is why we programmed the device to turn the motor off when the water in the tank reaches 90% . We also took care that for no period of time the user should feel that there is scarcity of water in the tank , that is why we have programmed the microcontroller to turn the motor on when the water level decreases below 10%.So this means , after someone is done installing the device on their rooftop tank he/she can literally forget about this.It will work on it’s own till the power to the device is there.

Chapter 1
Introduction

1.1 Introduction

Automation or automatic control is the use of various control systems for operating equipment such as machinery, processes in factories and other applications with minimal or reduced human intervention. The biggest benefit of automation is that it saves labor. However, it is also used to save energy and materials and to improve quality, accuracy and precision.

Water pump automation system is the technology of automatically collecting status data from the water tank metering devices and transferring that data to a water pump controlling system. Which will turn the motor on or off depending upon the water level in the tank. This technology mainly saves the hassle to physically check the water tank on our roof-top and turn the water-pump on/off , which in addition helps us avoid wastage of water.

1.2 Historical Overview:

The term automation, inspired by the earlier word automatic was not widely used before 1947, when General Motors established the automation department.[1] It was during this time that industry was rapidly adopting feedback controllers, which were introduced in the 1930s.[2]
Automation has been achieved by various means including mechanical, hydraulic, pneumatic, electrical, electronic and computers, usually in combination. Complicated systems, such as modern factories, airplanes and ships typically use all these combined techniques

The oldest form of automation to provide continuous corrective action is the centrifugal governor, which dates to the last quarter of the 18th century. The centrifugal governor was used to adjust the gap between millstones and also to adjust the vanes so they faced the wind.[3] The centrifugal governor was also used in the automatic flour mill developed by Oliver Evans in 1785, making it the first completely automated industrial process. The governor was adopted by James Watt for use on a steam an engine in 1788 after Watt's partner Boulton saw one at a flour mill Boulton & Watt were building [3].

Relay logic was introduced with factory electrification, which underwent rapid adaption from 1900 though the 1920s. Central electric power stations were also undergoing rapid growth and operation of new high pressure boilers, steam turbines and electrical substations created a large demand for instruments and controls.

Central control rooms became common in the 1920s, but as late as the early 1930s, most process control was on-off. Operators typically monitored charts drawn by recorders that plotted data from instruments. To make corrections, operators manually opened or closed valves or turned switches on or off. Control rooms also used color coded lights to send signals to workers in the plant to manually make certain changes [2].

Controllers, which were able to make calculated changes in response to deviations from a set point rather than on-off control, began being introduced the 1930s. Controllers allowed manufacturing to continue showing productivity gains to offset the declining influence of factory electrification.[4]

In 1959 Texaco’s Port Arthur refinery became the first chemical plant to use digital control. Conversion of factories to digital control began to spread rapidly in the 1970s as the price of computer hardware fell.[1]

1.3 Overview of the Technology

In our project the inbuilt automated water height measuring system powered by ultrasonic distance sensor monitors the water height in the tank.If it senses that there is insufficient amount of water in the tank then the system can turn the water pump motor ON and also can turn it OFF when the water height in the rooftop tank becomes near-full through a relay. The device also has a LCD display with it. This will display the water height live.User can monitor the changes in the water level through this display, without opening the hatch of the tank.

1.4 Objective of this work

In our country access to water becomes a big problem in dry seasons. Specially in big cities and suburban areas. But when we get more than enough water to pump in our rooftop tank we waste so much of that precious water by overflowing the tank.It is kind of very hard to stop that as we are already very busy with our life. We hardly get time to check if our rooftop water tank is full or not.That is why we need a new system of water pumping, which will save water and stop making us guilty of wasting a valuable natural resource.That is the exact thing what we tried to achieve in this project.We tried to make this project as simple as possible , but we did not negotiate on efficiency.It will be very easy to install .Without any actual knowledge of the device itself ,one can install it in their rooftop tank by just pressing a button.

The aim of this project was to Design and Implementation of Water pump automation system which is

- **Produced using locally available components and resources:** This is essential for making it feasible for manufacturing by local firms.
- **Reliable and cost efficient in long term:** The proposed design may require a larger initial investment but would save money in the long run if additional features are added.
- **Immune to tampering:** If the device is not immune to tampering then it’s longevity will be greatly hampered.
- **Marketable:** Water pump automation system is not a project of academic interest only. It is something that we really need today. So, it is very practical and can be immediately deployed in the real world.

1.5 Introduction of Project

This project is about the design and implementation of Water pump automation system. Water is an essential resource for all living things. The importance of an effective and efficient water pumping system cannot be over emphasized. This is very critical as non-efficient water pump system can lead to substantial loss of water and electricity also. Thus, it is essential to have an efficient and effective water pump automation system.

This project is about the design and construction of a highly efficient water pump automation system. It is made up of integrated circuits, active and passive hardware devices. The design monitors the amount of water in the water tank. The LCD display displays the height of the water in the tank . When the water in the tank reached 90% then the motor turns off and when the water decreases below 10% then the motor turns on.

The work presented in this project is organized in five main chapters. These five chapters are structured as follows:

Chapter 1 is entitled “Introduction". It introduces historical overview and overview of the technology used in the project. Objectives of this work are also discussed.

Chapter 2 is entitled "Idea Generation and Concept". The typical water pump system , Project Idea generation and Concept of the project are discussed in this chapter.

Chapter 3 is entitled "Hardware". It is devoted to introduce all the hardwires that is used in this project.

Chapter 4 is named "Project Implementation and Result Analysis ". This chapter is focused on the Circuit Diagrams and Block diagrams for this project. It discusses how we managed to do all the things that we claimed to have done effectively. Finally, in the Result Analysis one Experimental setup of the system are introduced.

Chapter 5 is entitled "Software Interfacing ". It is devoted to introduce software architecture as well as language of compilation of the system. Working methods of the software are also presented in this chapter.

Chapter 6 is entitled "Discussion & Conclusion ", Here all the achievements are summarized and appropriate conclusions are drawn. This chapter is focused on the further development of the project. Limitations of this project are discussed and different methods for improving the main drawbacks are introduced.

Chapter 2

Idea Generation and Concept

2.1 Typical Water pump system:

Water supply in our country is heavily dependent on groundwater extraction. In Dhaka city along with some of the biggest cities in our country groundwater extraction started from a depth of 100 meters and in some extreme condition the well goes up to 300 meters to reach the main aquifer [5].82% of the water supply is abstracted from groundwater that is free of arsenic, while three surface water treatment plants provide the remaining 18% [6]. For pumping up the water, different types of motor pumps are used. Centrifugal pumps are the most common type of pump used to move water through a piping system. Centrifugal pumps are used to transport fluids by the conversion of rotational kinetic energy to the hydrodynamic energy of the fluid flow. The rotational energy typically comes from an engine or Electric motor.

Firstly pumped water is collected in reserve water tank in the residence then water is lifted up to a tank situated on the roof. For usage of water, tank must be filled with water. So, these two water tanks have to be simultaneously checked whether there is enough water or not. Which is full of hassales. If the tank is full and overflowing then it has to be turned off , on the other hand if it's empty then it has to be turned on manually by someone. Which is very time consuming and full of hassles.

In a typical scenario, In almost every house there is a water tank in the rooftop and water pump in the ground floor to fill it up. One family member or the security person has to do the painful duty of checking whether the tank is full or empty. If it's empty then he/she will have to turn the water pump on and wait for the tank to fill up when it's time to switch the pump off. In the meantime, he/she will have to be extra alert to avoid any kind of wastage of water and power due to overflow of water in the tank. Which is a demanding task due to our busy daily life nowadays.

2.2 Project Idea Generation:

Access to water is a basic human right and it is a crosscutting issue for sustainable development. Water resources have unlimited importance including human survival, socio-economic stability and environmental sustainability. Dhaka city dwellers, following an unsystematic urban sprawl, are deprived of basic urban amenities where water supply has appeared as the most critical issue. Dhaka Water Supply and Sewerage Authority (DWASA) has projected water demand as 150 liters per person per day (l/p/d). Empirical evidence shows that one-third of the city dwellers receive only 40 l/p/d and they have to manage their daily activities with this little amount of water. Only 5.1 percent of total population of Dhaka city receives more than 60 l/p/d. On an average, 42.8 percent of the respondents can receive basic requirement of 50 l/p/d and the rest (57.8 percent) are suffering from water scarcity despite piped connection [5]. Water being a crosscutting resource, the urban water management involves complicated and systematic process that includes planning, research, design, engineering, regulation, and administration.

In everyday life, there must be some physical elements that need to be controlled in order for them to perform their expected behaviors. A control system therefore can be defined as a device, or set of devices, that manages, commands, directs or regulates the behavior of other device(s) or system(s). Consequently, automatic controlling involves designing a control system to function with minimal or no human interference. Intelligent systems are being used in a wide range of fields including from medical sciences to financial sciences, education, law, and so on. Several of them are embedded in the design of everyday devices.

Our typical water pump system in household has some drawbacks:

1. Waste of electrical energy, dwellers fails to switch off the motor pumps when the tank filled up.
2. Waste of valuable natural resource water due to the overflow.
3. Waste of money.

2.3 Concept:

When at first we thought of doing Automation on a water pump system , the first thing that comes to mind is how to measure the water height in the water tank.There can be many ways to do this . Such as , by placing electrodes in the water at certain levels and one electrodes will just float on top of the water.Whenever the water top electrode touches one the electrodes at different levels , we can know that the water has reached that certain level.But this process will not last for a long time because of water will cause damage to the electrodes.The same way placing laser in the water tank also will not last long time.As we need to put the laser light in such way that when the tank is full , the water must block the path of the laser light. So basically we need to submerge the laser light in water.Considering all these , we chose to use Ultrasonic sensor to measure the water height.As we will not have to submerge anything in the water ,that is why the longevity of the device and cleanliness of the water will not be hampered.

In this project , we have used an ultrasonic distance sensor to measure the water height in the tank , which will stay in a fixed position right above the water level and will not touch the water at any point of time.To drive this sensor and displaying the water height values in the LCD we used a very cost effective Microcontroller Atmega8.Which will also drive the relay . This relay will turn the machine ON or OFF depending on the water height in the tank.

This paper aimed at presenting our project in embedding a control system into an automatic water pump controller. One of the motivations for this research was the need to bring a solution to the problem of water shortage in various places by eliminating the major culprit; waste of water during pumping and dispensing into overhead thanks. We believe that creating a barrier to wastage will not only provide more financial gains and energy saving, but will also help the environment and water cycle which in turn ensures that we save water for our future.

Chapter 3
Hardware

3.1 Introduction:

In this project we have used a few components according to the needs of this project. We tried to make the project as simple as possible and kept the cost within a manageable amount. Following the block diagram of our project we used an Ultrasonic Distance Sensor to measure the water height using ultrasonic sound wave reflection. According to the water height measured by the Ultrasonic sensor ,to drive the LCD and the relay (which will control the motor) we used a cost effective microcontroller ATmega8.

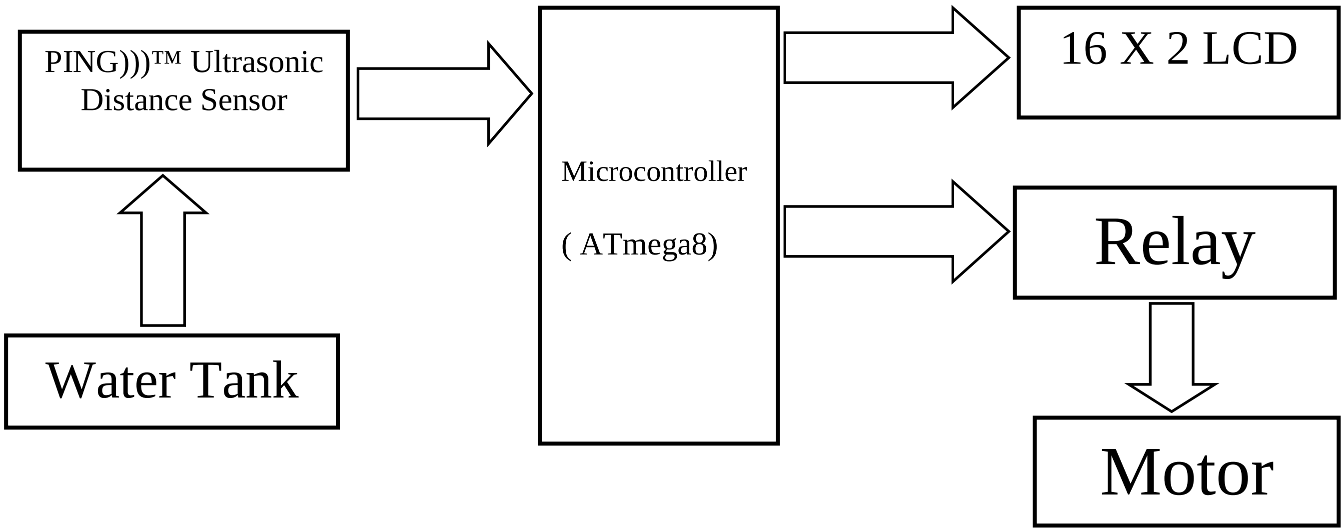


Figure 3.1.1: Block Diagram of Water pump automation system

The description of all of these hardware parts are given below.

3.2 Ultrasonic sensor:

Ultrasonic sensors (also known as transceivers when they both send and receive, but more generally called transducers) work on a principle similar to radar or sonar which evaluates attributes of a target by interpreting the echoes from radio or sound waves respectively. Ultrasonic sensors generate high frequency sound waves and evaluate the echo which is received back by the sensor. Sensors calculate the time interval between sending the signal and receiving the echo to determine the distance to an object.

This technology can be used for measuring wind speed and direction (anemometer), tank or channel level, and speed through air or water. For measuring speed or direction a device uses multiple detectors and calculates the speed from the relative distances to particulates in the air or water. To measure tank or channel level, the sensor measures the distance to the surface of the fluid. Further applications include: humidifiers, sonar, medical ultrasonography, burglar alarms and non-destructive testing.
Systems typically use a transducer which generates sound waves in the ultrasonic range, above 18,000 hertz, by turning electrical energy into sound, then upon receiving the echo turn the sound waves into electrical energy which can be measured and displayed.
The technology is limited by the shapes of surfaces and the density or consistency of the material.

3.2.1 PING)))™ Ultrasonic Distance Sensor (#28015)

The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to BASIC Stamp® or Javelin Stamp microcontrollers, requiring only one I/O pin.
The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width the distance to target can easily be calculated. [7]

Figure 3.2.1 : Physical appereance of the PING)))™ Ultrasonic Distance Sensor [7]

3.2.1.2 Features:

• Supply Voltage – 5 VDC

• Supply Current – 30 mA typ; 35 mA max

• Range – 2 cm to 3 m (0.8 in to 3.3 yrds)

• Input Trigger – positive TTL pulse, 2 uS min, 5 µs typ.

• Echo Pulse – positive TTL pulse, 115 uS to 18.5 ms

• Echo Hold-off – 750 µs from fall of Trigger pulse

• Burst Frequency – 40 kHz for 200 µs

• Burst Indicator LED shows sensor activity

• Delay before next measurement – 200 µs

• Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)

3.2.1.3 Theory of Operation

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air at about 1130 feet per second, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target [7].

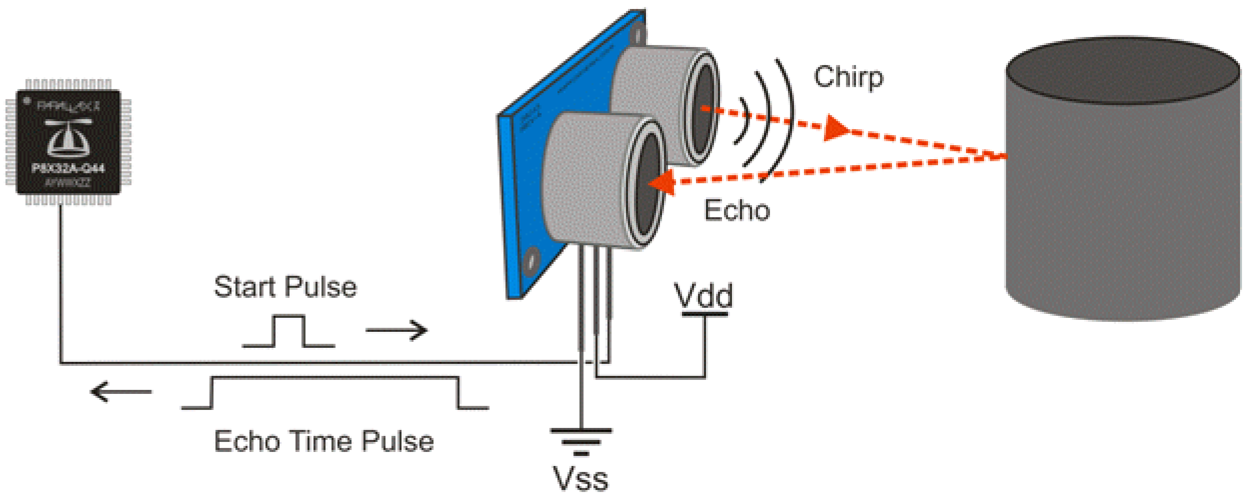


Figure 3.2.2: Working Principle of Ultrasonic Distance Sensor [8]

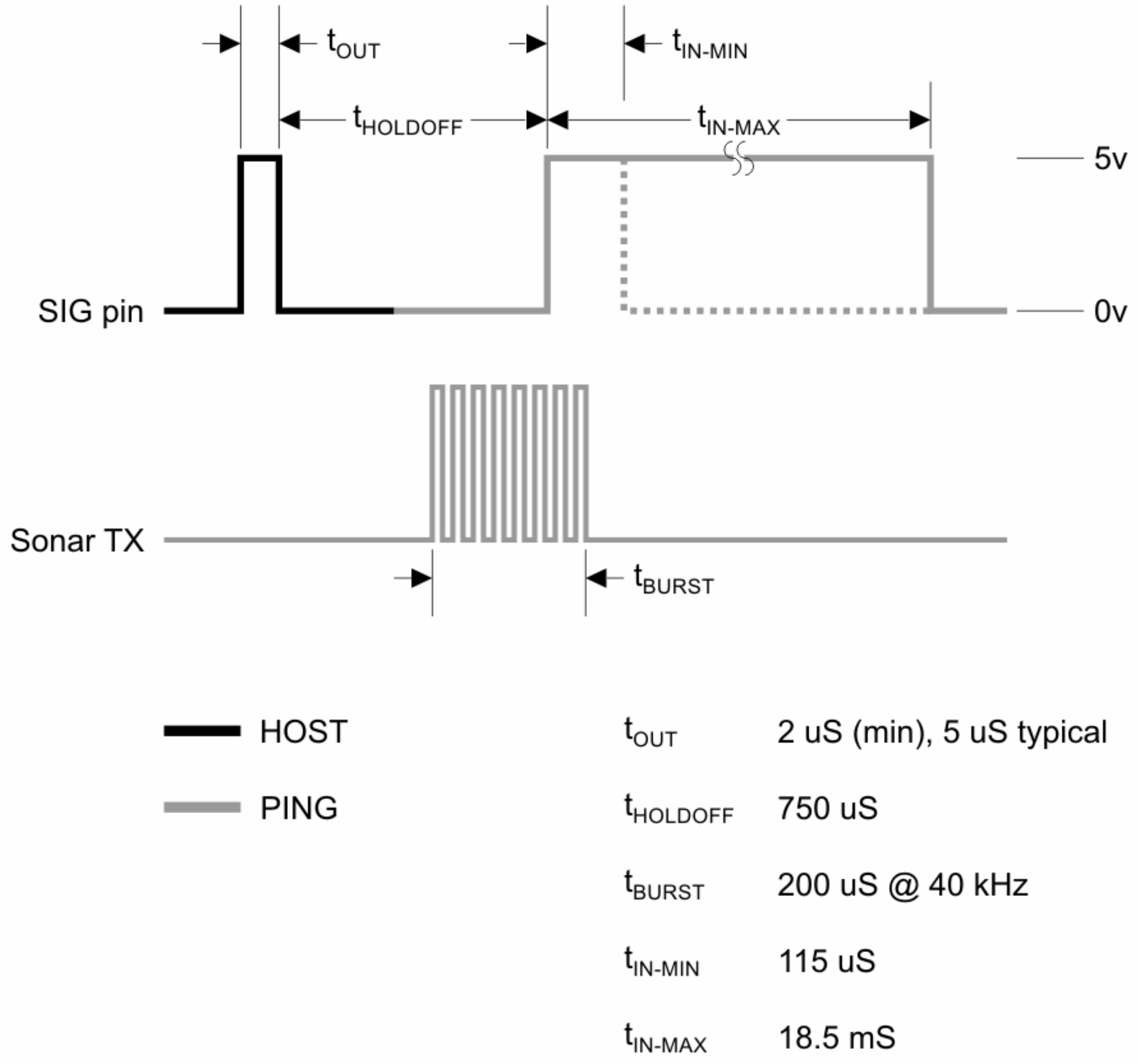


Figure 3.2.3: Time Diagram of Ultrasonic Distance Sensor [7]

3.3 Microcontroller

3.3.1 Introduction:

A microcontroller (sometimes abbreviated µC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use 4-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption (single-digit milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

Figure 3.3.1: 2 ATmegaxx micro controllers

3.3.2 Embedded design:

A microcontroller can be considered a self-contained system with a processor, memory and peripherals and can be used as an embedded system [9]. The majority of microcontrollers in use today are embedded in other machinery, such as automobiles, telephones, appliances, and peripherals for computer systems. While some embedded systems are very sophisticated, many have minimal requirements for memory and program length, with no operating system, and low software complexity. Typical input and output devices include switches, relays, solenoids, LEDs, small or custom LCD displays, radio frequency devices, and sensors for data such as temperature, humidity, light level etc. Embedded systems usually have no keyboard, screen, disks, printers, or other recognizable I/O devices of a personal computer, and may lack human interaction devices of any kind.

3.3.3 Overview(ATmega8):

The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8 provides the following features: 8 Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1 Kbyte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Twowire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Powerdown mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel’s high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel

ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.The ATmega8 AVR is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits [10].

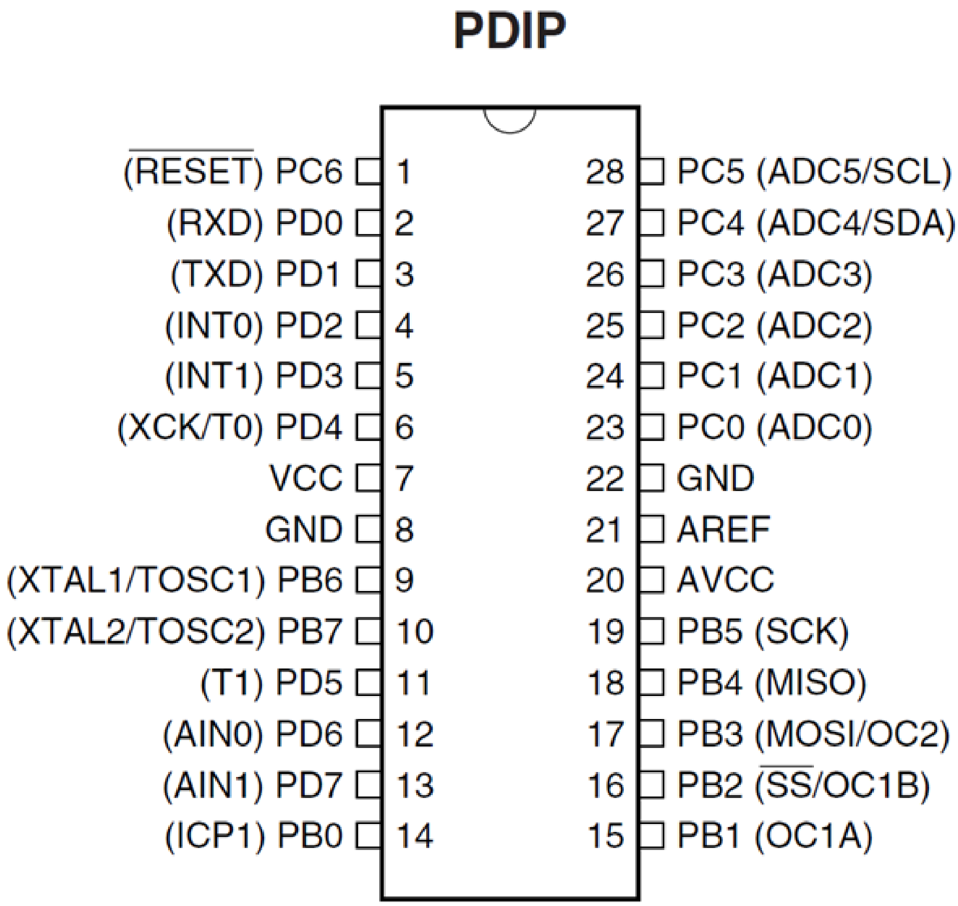


Figure 3.3.2: Pin diagram of Atmega8 microcontroller [10]

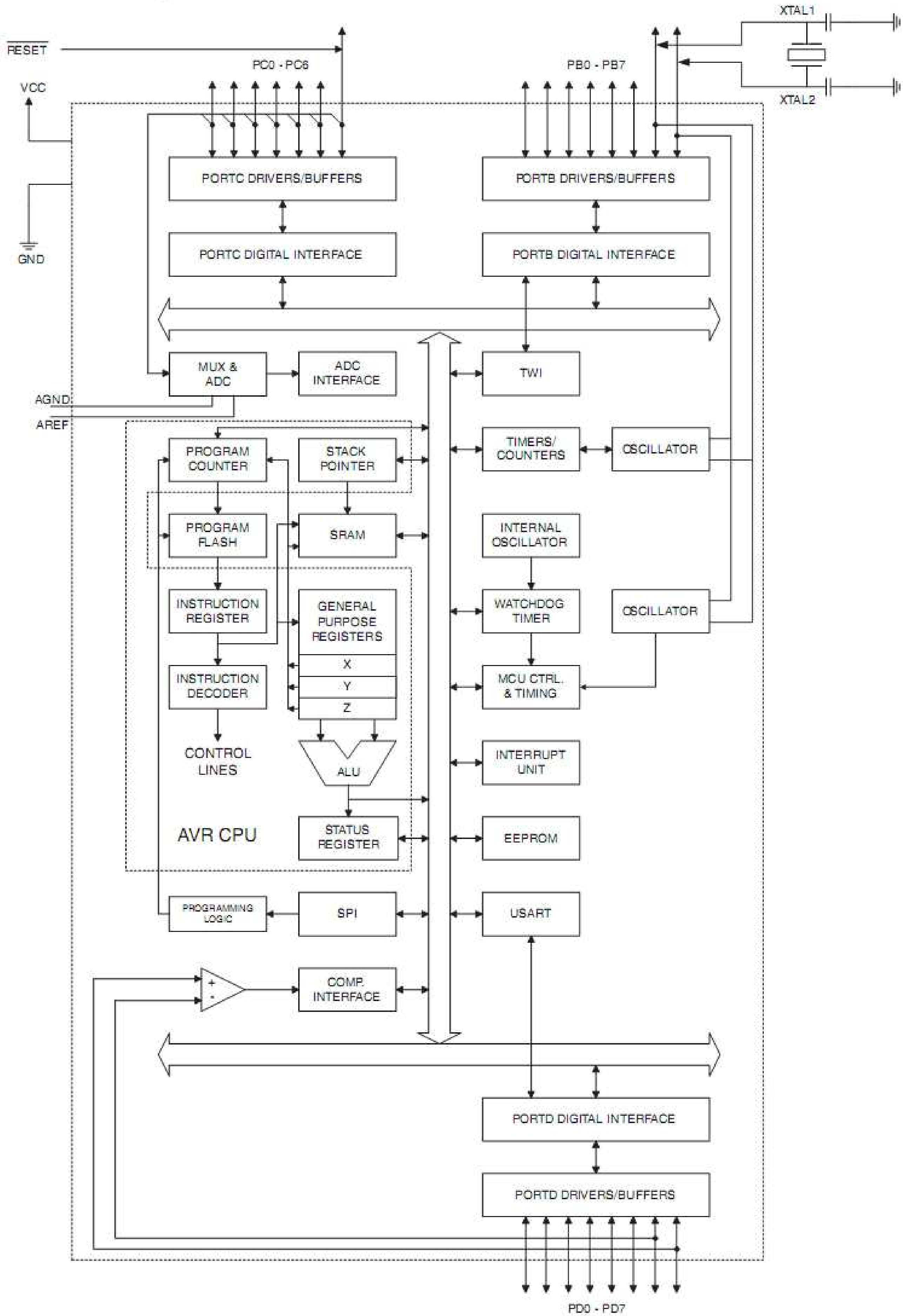


Figure 3.3.3: Block diagram of Atmega8 microcontroller [10]

3.3.3.1 Features:

• **High-performance, Low-power Atmel®AVR® 8-bit Microcontroller**

• **Advanced RISC Architecture**

– 130 Powerful Instructions – Most Single-clock Cycle Execution

– 32 × 8 General Purpose Working Registers

– Fully Static Operation

– Up to 16 MIPS Throughput at 16MHz

– On-chip 2-cycle Multiplier

• **High Endurance Non-volatile Memory segments**

– 8Kbytes of In-System Self-programmable Flash program memory

– 512Bytes EEPROM

– 1Kbyte Internal SRAM

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM

– Data retention: 20 years at 85°C/100 years at 25°C(1)

– Optional Boot Code Section with Independent Lock Bits

In-System Programming by On-chip Boot Program

True Read-While-Write Operation

– Programming Lock for Software Security

• **Peripheral Features**

– Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode

– Real Time Counter with Separate Oscillator

– Three PWM Channels

– 8-channel ADC in TQFP and QFN/MLF package

Eight Channels 10-bit Accuracy

– 6-channel ADC in PDIP package

Six Channels 10-bit Accuracy

– Byte-oriented Two-wire Serial Interface

– Programmable Serial USART

– Master/Slave SPI Serial Interface

– Programmable Watchdog Timer with Separate On-chip Oscillator

– On-chip Analog Comparator

• **Special Microcontroller Features**
– Power-on Reset and Programmable Brown-out Detection
– Internal Calibrated RC Oscillator
– External and Internal Interrupt Sources
– Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

• **I/O and Packages**
– 23 Programmable I/O Lines
– 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF

• **Operating Voltages**
– 2.7V - 5.5V (ATmega8L)
– 4.5V - 5.5V (ATmega8)
• **Speed Grades**
– 0 - 8MHz (ATmega8L)
– 0 - 16MHz (ATmega8)

• **Power Consumption** at 4Mhz, 3V, 25°C
– Active: 3.6mA
– Idle Mode: 1.0mA
– Power-down Mode: 0.5µA

3.4 Relay:

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays were used extensively in telephone exchanges and early computers to perform logical operations.
A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and

sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Figure 3.4.1: DPDT 5V Relay [11]

3.4.1 Basic design and operation

A simple electromagnetic relay consists of a coil of wire wrapped around a soft iron core, an iron yoke which provides a low reluctance path for magnetic flux, a movable iron armature, and one or more sets of contacts (there are two in the relay pictured). The armature is hinged to the yoke and mechanically linked to one or more sets of moving contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and the circuit track on the printed circuit board (PCB) via the yoke, which is soldered to the PCB.

When an electric current is passed through the coil it generates a magnetic field that activates the armature and the consequent movement of the movable contact either makes or breaks (depending upon construction) a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low-voltage application this reduces noise; in a high voltage or current application it reduces arcing.

When the coil is energized with direct current, a diode is often placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to semiconductor circuit components. Some automotive relays include a diode inside the relay case. Alternatively, a contact protection network consisting of a capacitor and resistor in series (snubber circuit) may absorb the surge. If the coil is designed to be energized with alternating current (AC), a small copper "shading ring" can be crimped to the end of the solenoid, creating a small out-of-phase current which increases the minimum pull on the armature during the AC cycle [12].

A solid-state relay uses a thyristor or other solid-state switching device, activated by the control signal, to switch the controlled load, instead of a solenoid. An optocoupler (a light-emitting diode (LED) coupled with a photo transistor) can be used to isolate control and controlled circuits.

3.4.2 Advantages and Disadvantages of Relay

The main advantages and disadvantages of relays are listed below:

Advantages:

- Relays can switch AC and DC, transistors can only switch DC.
- Relays can switch higher voltages than standard transistors.
- Relays are often a better choice for switching large currents (> 5A).
- Relays can switch many contacts at once.

Disadvantages:

- Relays are bulkier than transistors for switching small currents.
- Relays cannot switch rapidly (except reed relays), transistors can switch many times per second.
- Relays use more power due to the current flowing through their coil.
- Relays require more current than many ICs can provide, so a low power transistor may be needed to switch the current for the relay's coil [13].

3.4.3 Relay Application Considerations

Selection of an appropriate relay for a particular application requires evaluation of many different factors:

- Number and type of contacts – normally open, normally closed, (double throw).

- Contact sequence – "Make before Break" or "Break before Make". For example, the old style telephone exchanges required Make before break so that the connection didn't get dropped while dialing the number.
- Rating of contacts – small relays switch a few amperes, large contactors are rated for up to 3000 amperes, alternating or direct current.
- Voltage rating of contacts – typical control relays rated 300 V AC or 600 V AC, automotive types to 50 V DC, special high voltage relays to about 15 000 V.
- Operating lifetime, useful life - the number of times the relay can be expected to operate reliably. There is both a mechanical life and a contact life. The contact life is naturally affected by the kind of load being switched: switching while "wet" (under load) causes undesired arcing between the contacts, eventually leading to contacts that weld shut or contacts that fail due to a build up of contact surface damage caused by the destructive arc energy [14].
- Coil voltage – machine tool relays usually 24 VAC, 120 or 250 VAC, relays for switchgear may have 125 V or 250 VDC coils, "sensitive" relays operate on a few milliamperes
- Coil current including minimum current required to operate reliably and minimum current to hold. Also effects of power dissipation on coil temperature at various duty cycles.
- Package/enclosure – open, touch safe, double voltage for isolation between circuits, explosion proof, outdoor, oil and splash resistant, washable for printed circuit board assembly.
- Operating environment minimum and maximum operating temperatures and other environmental considerations such as effects of humidity and salt.
- Assembly – Some relays feature a sticker that keeps the enclosure sealed to allow PCB post soldering cleaning, which is removed once assembly is complete.
- Mounting – sockets, plug board, rail mount, panel mount, through panel mount, enclosure for mounting on walls or equipment.
- Switching time – where high speed is required.
- "Dry" contacts – when switching very low level signals, special contact materials may be needed such as gold plated contacts.
- Contact protection – suppress arcing in very inductive circuits.
- Coil protection – suppress the surge voltage produced when switching the coil current.
- Isolation between coil circuit and contacts.
- Aerospace or radiation resistant testing, special quality assurance.
- Expected mechanical loads due to acceleration – some relays used in aerospace applications are designed to function in shock loads of 50 g or more.

- Size - smaller relays often resist mechanical vibration and shock better than larger relays, because of the lower inertia of the moving parts and the higher natural frequencies of smaller parts [15]. Larger relays often handle higher voltage and current than smaller relays.

There are many considerations involved in the correct selection of a control relay for a particular application. These considerations include factors such as speed of operation, sensitivity and hysteresis. Although typical control relays operate in the 5 ms to 20 ms range, relays with switching speeds as fast as 100 us are available. Reed relays which are actuated by low currents and switch fast are suitable for controlling small currents.

As for any switch, the current through the relay contacts (unrelated to the current through the coil) must not exceed a certain value to avoid damage. In the particular case of high inductance circuits such as motors, other issues must be addressed. When a power source is connected to an inductance, an input surge current which may be several times larger than the steady current exists. When the circuit is broken, the current cannot change instantaneously, which creates a potentially damaging spark across the separating contacts.

Consequently for relays which may be used to control inductive loads we must specify the maximum current that may flow through the relay contacts when it actuates, the make rating, the continuous rating and the break rating. The make rating may be several times larger than the continuous rating, which is itself larger than the break rating.

3.5 16x2 LCD:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD [16].

Figure 3.5.1: A 16x2 LCD [17]

3.5.1 Features

• 5 x 8 dots with cursor

• Built-in controller (KS 0066 or Equivalent)

• + 5V power supply (Also available for + 3V)

• 1/16 duty cycle

• B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)

• N.V. optional for + 3V power supply

Figure 3.5.2: A 16x2 LCD displaying the sentence " Hello , World! " and the digit 5171 [18]

3.5.2 Pin configuration:

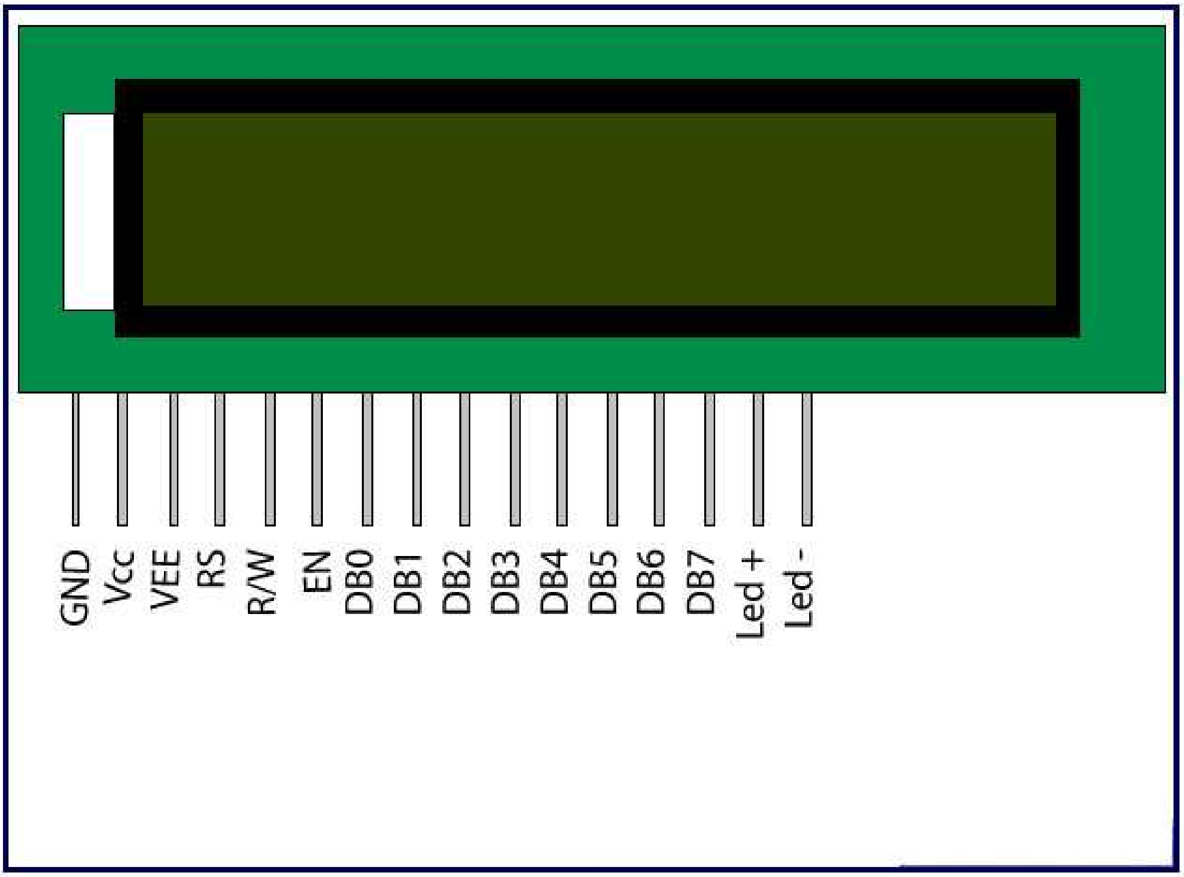


Figure 3.5.3: Pin Configuration of 16x2 LCD [16]

3.5.3 Pin Description:

Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-

Chapter 4

Project Implementation and Result Analysis

4.1 Introduction

In this chapter, we have elaborately demonstrated the block diagram, circuit diagram, and all the technical details that is related to this project. Apparently this chapter gives the basic idea of this project and helps to understand the working principle of the "Water pump automation system".

4.2 Basic principle of the Project:

In this project we have used a Ultrasonic distance sensor to measure the water height in the water tank , then the Microcontroller displays the water height (In Inches) in the LCD and while the water height changes in the tank it then displays the water height in percentage also . According to this percentage value of the height of the water, the Microcontroller drives the motor on/off through the relay.

4.3 Block diagram of The Project

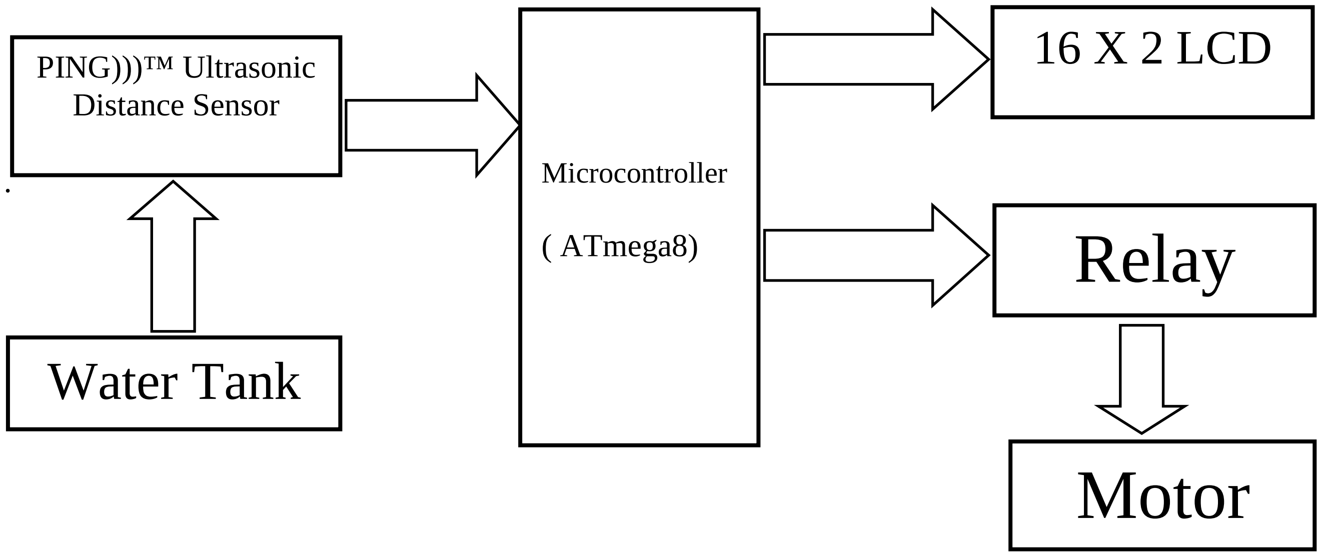


Figure 4.3.1: Block Diagram of Water pump automation system

4.4 Circuit Diagram:

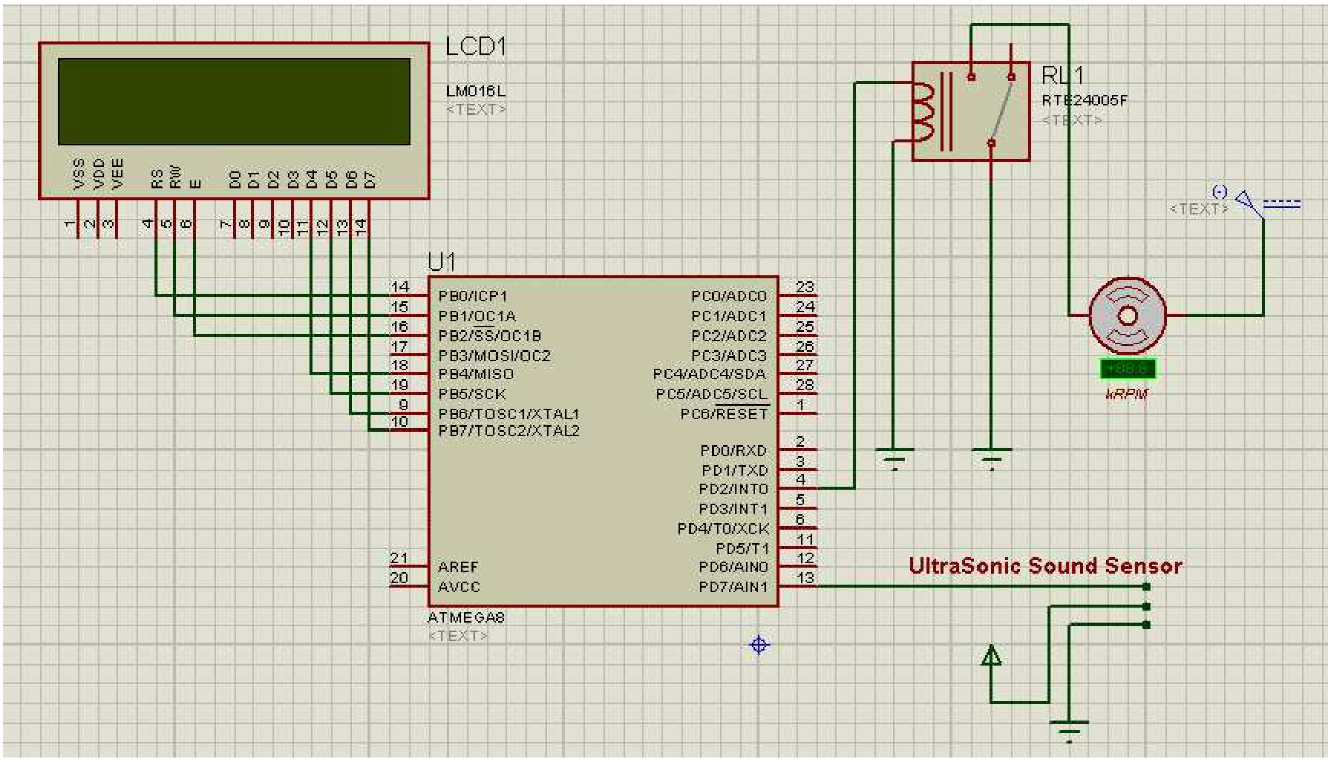


Figure 4.4.1: Circuit Diagram of Water pump automation system

Figure 4.4.2: Physical appearance of the Water pump automation system

4.5 Working Principle:

In this project we place the ultrasonic distance sensor right on top of the water tank. This ultrasonic distance sensor emits a 40 kHz (ultrasonic) burst. This burst travels through the air at about 1130 feet (13560 Inch) per second, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target. The generally accepted value for the speed-of-sound is 1130 feet per second. This works out to 13,560 inches per second or 1 inch in 73.746 microseconds.

4.5.1 Measuring the height:

When the sensor starts transmitting then the sonar sensor sends a “high” to the microcontroller. While the ultrasonic wave is bounced back to the receiver part of the sonar sensor then the sensor makes this “high” signal into a “low” one.The microcontroller calculates the time between this one high and low pulse . But this whole time is for the ultrasonic wave to travel twice the distance of the water height in the tank. That is why we divide this time by two. We are calling this time as Elapsed time. If we divide this elapsed time by 73.74 then we will get the height in Inch. We have to remember that, all our calculations here are done in microseconds and inches.

4.5.2 Setting up the project for one particular water tank:

When we are done with measuring the tank height (empty water tank) in inches then we will save this value onto an EEPROM by pressing the store button and after this the LCD will display the sentence “Data Stored”. So that if there is a load-shedding or the power goes off due to some reason then we won’t have to measure the tank height again and again. So, basically we are fixing one set of this project for one water tank at a time. We can use one set of this project in many water tanks , but at first we have to do the measuring and saving the empty water tank height to the EEPROM first. After this the LCD will display the tank height in the first column right after the text “Water Level”

4.5.3 Measuring the water height:

To measure the water height all the procedures described in 4.5.1 is followed. After this the water height is displayed in the LCD in the second column, and after that the percentage value of the water height according to the tank height is displayed. To calculate the percentage value we used the following simple equation:

$$Percantage\ value = \frac{Water\ Height}{Tank\ Height} \times 100\%$$

4.5.4 Making the relay on/off:

Through microcontroller we have programmed the microcontroller to switch the relay on when the water height goes down below 10% and switch the relay off when the water height is above 90% . We are switching the relay off at 90% and not when the tank is 100% full to save our sensor and the circuit which will be in the tank itself.

4.5.5 Power supply in microcontroller:

The power for microcontroller was taken from a 9v Battery source. A voltage regulator IC 7805 used to keep constant value of the voltage at 5V.But we can also power the microcontroller from the main power supply, to do that we need to use a step-down transformer to make the AC 220V connection to 12V/9V DC line. Which then will be connected to IC 7805 to make a 5V constant DC line.

4.6 PCB Circuit Design:

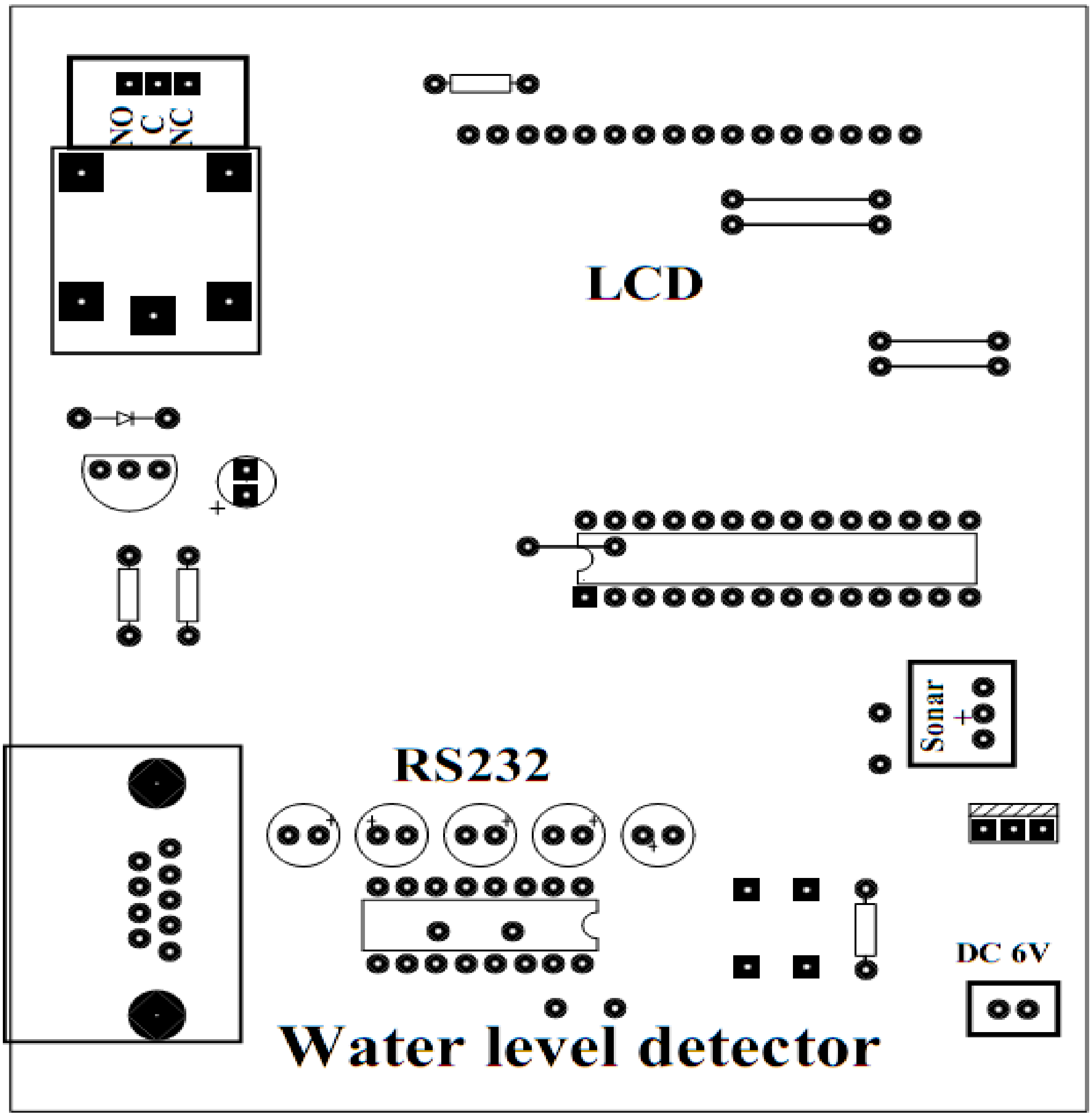


Figure 4.6.1: Top part of PCB Circuit Design

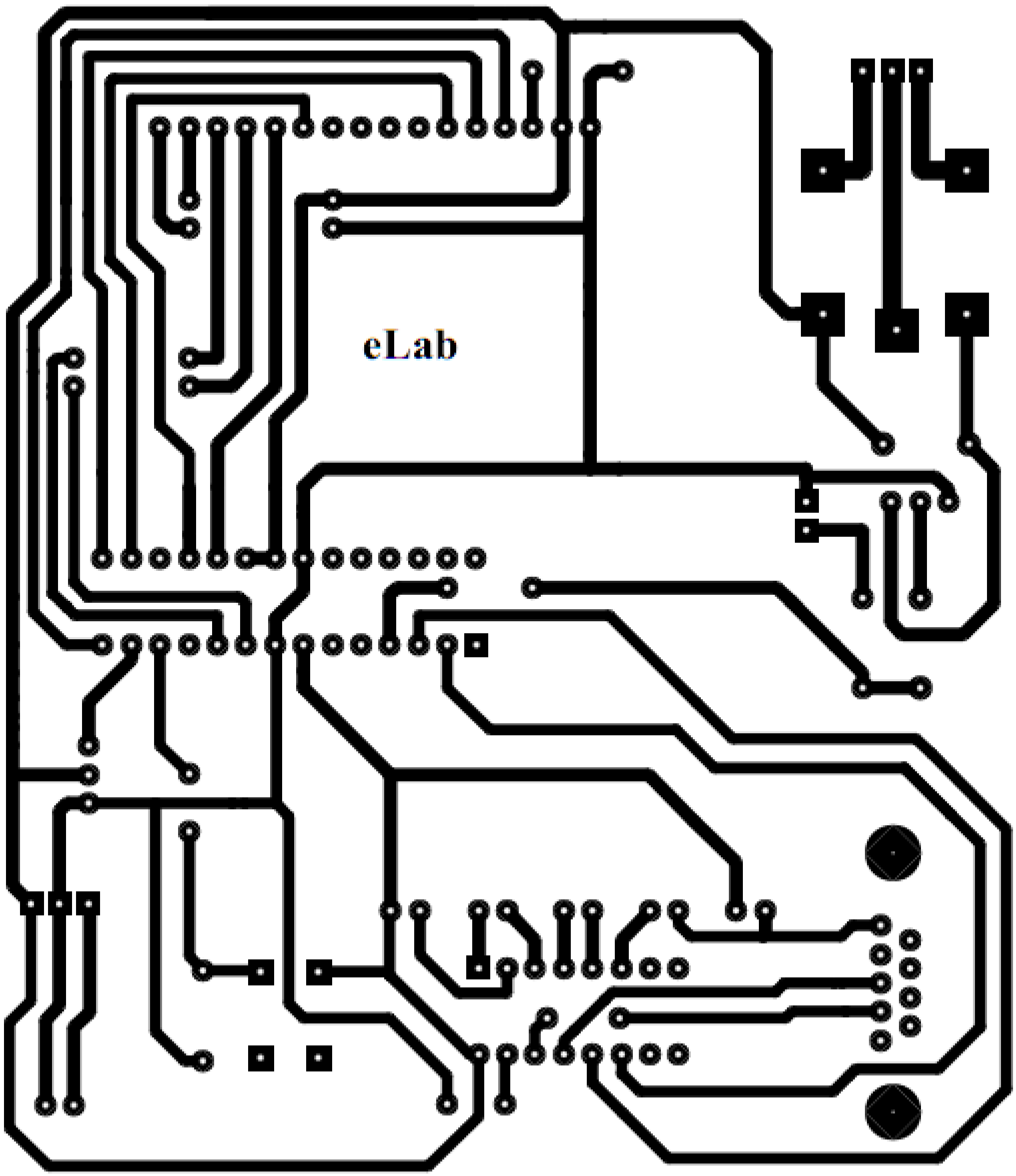


Figure 4.6.2: Bottom part of the PCB Circuit Design

4.7 Result Analysis:

We tested the project in rooftop water tanks and it worked just fine.Here is a demonstration of the project on a plain floor ,which will represent the water surface in the water tank.

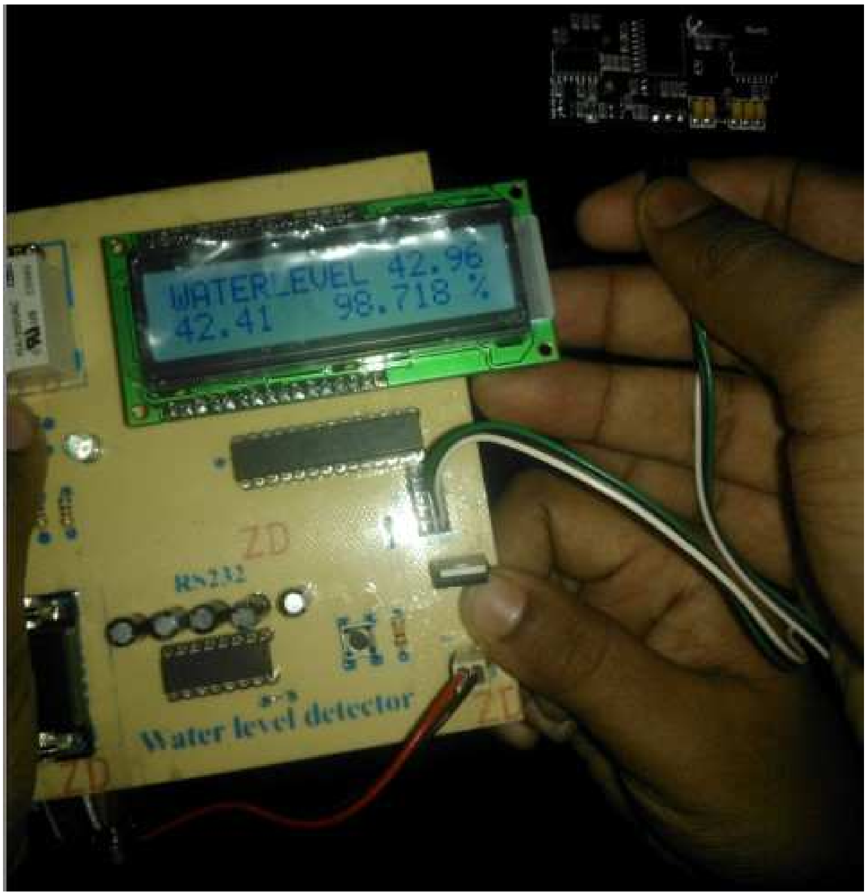


Figure 4.7.1: An almost empty Water tank

This represent an almost empty tank , which is 98.718% empty.The tank height is shown on the top right (in inches). The percentage value is shown in the bottom right corner.

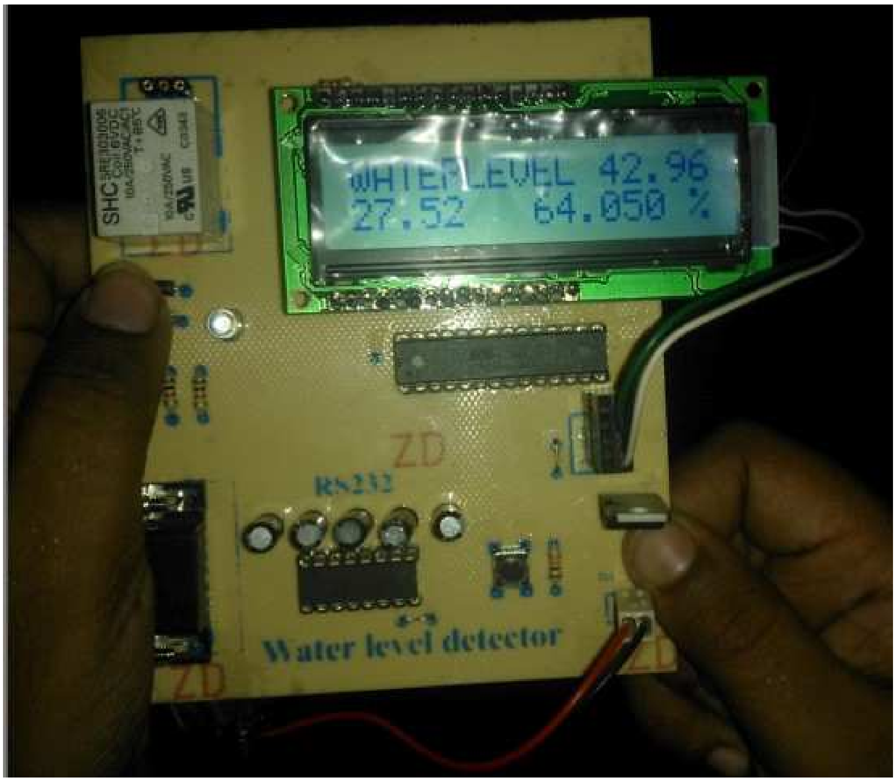


Figure 4.7.2: A semi-filled Water tank

After a few mins the water will fill the water tank a bit. This represents a semi-filled water tank. Which is 64.050% empty.

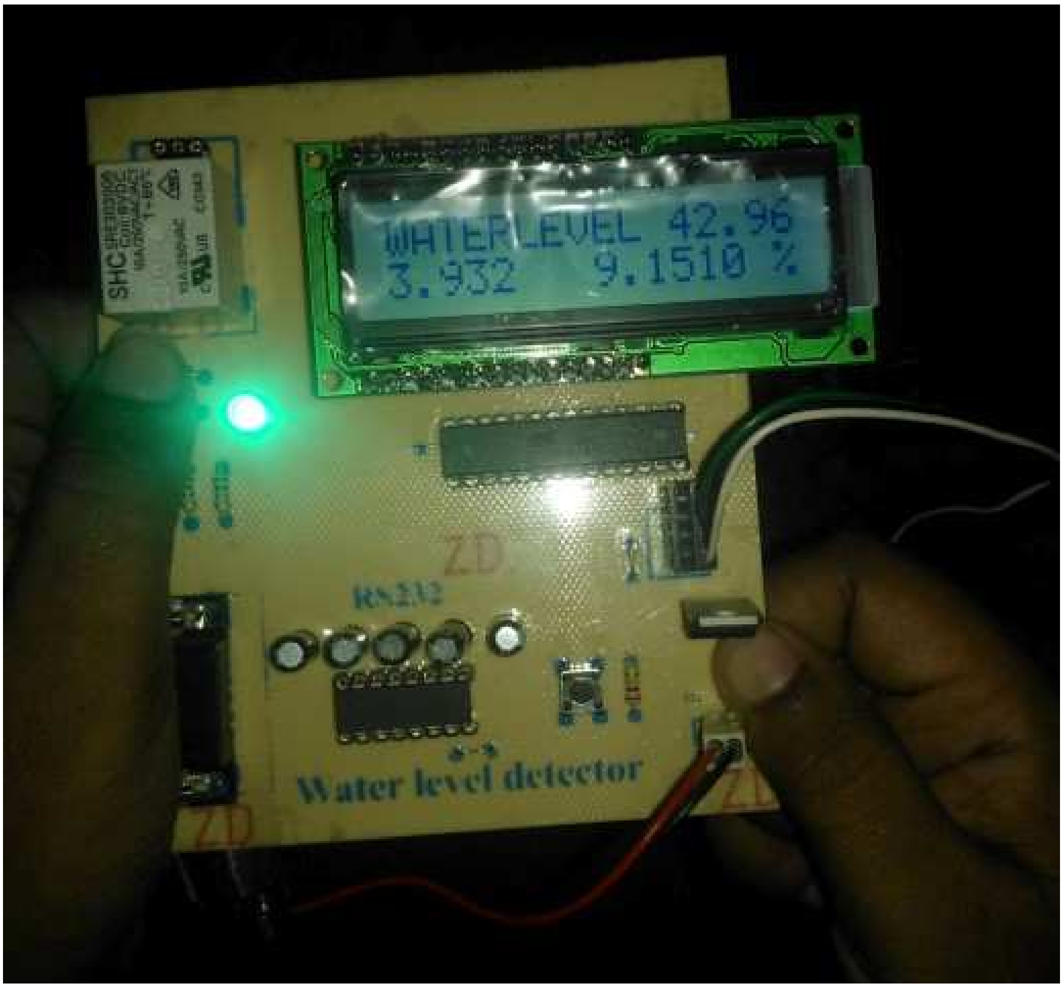


Figure 4.7.3: An almost filled Water tank

And at last , this represents an almost filled water tank with only 9.1510% left to fill. By the microprocessor we have coded the relay to turn the water pump off when it's 90%+ full. That is why we can see the Green LED light turning on, which indicates that the Motor is OFF.

	Tank Height (Inch)	Current height value (Inch)	Water Height (Tank Height – Current value) (Inch)	Percantage value (How much of the tank is empty)	Machine ON or OFF
1.	42.96	42.41	0.55	98.718%	**ON**
2.	42.96	27.52	15.44	64.050%	**ON**
3.	42.96	3.932	39.028	9.1510%	**OFF**

Chapter 5
Software Interfacing
5.1 Introduction

A software interface may refer to a range of different types of interface at different "levels".an operating system may interface with pieces of hardware. Applications or programs running on the operating system may need to interact via streams, and in object oriented programs, objects within an application may need to interact via methods.

5.2 Design of Software Architecture

Software application architecture is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application [19].

5.3 Types of Compiled Languages

A compiled language is a programming language whose implementations are typically compilers (translators which generate machine code from source code), and not interpreters (step-by-step executors of source code, where no pre-runtime translation takes place).The term is somewhat vague; in principle any language can be implemented with a compiler or with an interpreter. A combination of both solutions is also increasingly common: a compiler can translate the Source code into some intermediate form (often called bytecode),which is then passed to an interpreter which executes it.

5.4 C:

In computing, C is a general-purpose programming language initially developed by Dennis Ritchie between 1969 and 1973 at AT&T Bell Labs. Like most imperative languages in the ALGOL tradition, C has facilities for structured programming and allows lexical variable scope and recursion, while a static type system prevents many unintended operations. Its design provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, most notably system software like the Unix computer operating system [20].

C is one of the most widely used programming languages of all time, and C compilers are available for the majority of available computer architectures and operating systems.

C is an imperative (procedural) language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. C was therefore useful for many applications that had formerly been coded in assembly language, such as in system programming.

Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers.

5.4.1 Characteristics

Like most imperative languages in the ALGOL tradition, C has facilities for structured programming and allows lexical variable scope and recursion, while a static type system prevents many unintended operations. In C, all executable code is contained within subroutines, which are called "functions" (although not in the strict sense of functional programming). Function parameters are always passed by value. Pass-by-reference is simulated in C by explicitly passing pointer values. C program source text is free-format, using the semicolon as a statement terminator and curly braces for grouping blocks of statements.

The C language also exhibits the following characteristics:

- There is a small, fixed number of keywords, including a full set of flow of control primitives: for, if/else, while, switch, and do/while. There is one namespace, and user-defined names are not distinguished from keywords by any kind of sigil.
- There are a large number of arithmetical and logical operators, such as +, +=, ++, &, ~, etc.
- More than one assignment may be performed in a single statement.
- Function return values can be ignored when not needed.
- Typing is static, but weakly enforced: all data has a type, but implicit conversions can be performed; for instance, characters can be used as integers.
- Declaration syntax mimics usage context. C has no "define" keyword; instead, a statement beginning with the name of a type is taken as a declaration. There is no "function" keyword; instead, a function is indicated by the parentheses of an argument list.

- User-defined (typedef) and compound types are possible.
- Heterogeneous aggregate data types (struct) allow related data elements to be accessed and assigned as a unit.
- Array indexing is a secondary notion, defined in terms of pointer arithmetic. Unlike structs, arrays are not first-class objects; they cannot be assigned or compared using single built-in operators. There is no "array" keyword, in use or definition; instead, square brackets indicate arrays syntactically, e.g. month.
- Enumerated types are possible with the enum keyword. They are not tagged, and are freely interconvertible with integers.
- Strings are not a separate data type, but are conventionally implemented as null-terminated arrays of characters.
- Low-level access to computer memory is possible by converting machine addresses to typed pointers.
- Procedures (subroutines not returning values) are a special case of function, with an untyped return type void.
- Functions may not be defined within the lexical scope of other functions.
- Function and data pointers permit ad hoc run-time polymorphism.
- A preprocessor performs macro definition, source code file inclusion, and conditional compilation.
- There is a basic form of modularity: files can be compiled separately and linked together, with control over which functions and data objects are visible to other files via static and extern attributes.
- Complex functionality such as I/O, string manipulation, and mathematical functions are consistently delegated to library routines.
- C does not include some features found in newer, more modern high-level languages, including object orientation and garbage collection.

5.5 Code Vision AVR

CodeVision AVR is a C cross-compiler, integrated development environment and automatic program generator designed for the Atmel AVR family of microcontrollers.

The C cross-compiler implements all the elements of the ANSI C language, as allowed by the AVR architecture, with some features added to take advantage of specificity of the AVR architecture and the embedded system needs.

The compiled COFF object files can be C source level debugged, with variable watching, using the

atmel studio and AVR Studio debiggers.
The Integrated Development Environment (IDE) Has built-in AVR chip in-system programmer software that enables the automatic transfer of the program to the microcontroller chip after successful compilation/assembly. The In-system Programmer software is designed to work in conjunction with many development boards.For debugging embedded systems, which employ serial communication, the IDE has a built-in terminal.

CodeVisionAVR can be also used as an extension in ATmel Studio 6.1 or later, allowing seamless editing, compiling and debugginh projects in this IDE.
Besided the standard C Libraries, the CodeVisionAVR C compiler has dedicated libraries for:

- Alphanumeric and Graphic LCD modules
- Philips I2c bus
- National Semiconductor LM75 Temperature sensor
- Maximum/Dallas semiconductor 1 wire Protocol
- SPI
- Power Management
- Delays
- Gray Code conversion
- FAT access on MMC/SD/SDHC FLASH memory cards etc

CodeVisionAVR also contains the CodeWizerdAVR Automatic Program Generator, that allows you to write, in a matter of minutes, all the code needed for implementing the following functions:

- External Memory access setup
- Chip reset source identification
- Input/output port initialization
- External interrupts initialization
- timers/counters initialization
- watchdog timer initialization
- UART (USART) initialj and interrupt driven buffered serial communication
- Analog Comparator initialization
- SPI Interface initialization
- Alphanumeric and graphic display module initialization etc [21].

5.5.1 Features:

- Easy to use Integrated Development Environment and ANSI C compatible Compiler
- Editor with auto indentation, syntax highlighting for both C and AVR assembler, function parameters and structure/union members autocomplete
- Besides it's own IDE, CodeVisionAVR can be used an Extension integrated in Atmel Studio 6.1 or later
- Supported data types: bit, bool, char, int, short, long, float
- Fast floating point library with hardware multiplier and enhanced core instructions support for all the new ATmega chips
- AVR specific extensions for:
    - Transparent, easy accessing of the EEPROM & FLASH memory areas, without the need of special functions like in other AVR compilers
    - Bit level access to I/O registers
    - Interrupt support
    - Support for placing bit variables in the General Purpose I/O Registers (GPIOR) available in the new chips (ATtiny2313, ATmega48/88/168, ATmega165/169/325/3250/329/3290/645/6450/649/6490, ATmega1280/1281/2560/2561/640, ATmega406 and others)
- Compiler optimizations:
    - Peephole optimizer
    - Advanced variables to register allocator, allows very efficient use of the AVR architecture
    - Common Block Subroutine Packing (what our competition calls "Code Compressor"), replaces repetitive code sequences with calls to subroutines. This optimizer is available as Standard in CodeVisionAVR, at no additional costs, not like in our competitor's products.
    - Common sub-expression elimination
    - Loop optimization
    - Branch optimization
    - Subroutine call optimization

  - Cross-jumping optimization
  - Constant folding
  - Constant literal strings merging
  - Store-copy optimization
  - Dead code removing optimization
  - 4 memory models: TINY (8 bit data pointers for chips with up to 256 bytes of RAM), SMALL (16 bit data pointers for chips with more than 256 bytes of RAM), MEDIUM (for chips with 128k of FLASH) and LARGE (for chips with 256k or more FLASH). The MEDIUM and LARGE memory models allow full FLASH addressing for chips like ATmega128, ATmega1280, ATmega2560, etc, the compiler handling the RAMPZ register totally transparently for the programmer. This feature is available as Standard in CodeVisionAVR, at no additional costs, not like in our competitor's products.
  - User selectable optimization for code Size or Speed
- Possibility to insert inline assembler code directly in the C source file
- VERY EFFICIENT USE OF RAM: Constant literal strings are stored only in FLASH memory and aren't copied to RAM and accessed from there, like in other compilers for the AVR
- C Source level debugging, with COFF symbol file generation, allows variable watching (including structures and unions) in Atmel Studio 6.1 and AVR Studio 4.19 debuggers
- Fully compatible with Atmel's In-Circuit Emulators: AVR JTAG-ICE, AVR Dragon, etc [22].

Chapter 6

Discussion and Future Work

6.1 Discussion

Since the inception of motor, we have invented new ways to use this device to meet various kinds of our needs. Pumping water from ground or other sources is one of them. As this technology is getting cheaper and cheaper day by day almost every single house will have one of this to pump water onto their rooftop tank. But most of the time we can see these rooftop water tanks is overflowing and water is being wasted. Water is one of the precious natural resource we find without paying anything. It is our duty to protect this resource for our future generation .As so many people have predicted that next world war might just be a “Water War” between nations.

The typical water pump system in our household is very old and time consuming.This project of our is hassle free after the installation and can be operated without any human intervention except when the battery (power source) needs a changing.The benefits of this projects are –

- Saves water
- Saves time
- No need of any human intervention for a long time after the installation
- Saves Electricity by switching the motor off at the correct time
- Helps preventing a mess by stopping the overflow of water

6.2 Future Work

Now we are working to improve our technology.In future we want to add some extra features like –

Auto and Manual start:

Sometimes for some reasons we might need to stop the circuit and need to manually start the motor.That is why we want to include this features where the user will have the option to choose between Auto and Manual start.

Wireless Sensor : To avoid connecting the sensor and microcontroller part with the relay with a long wire (which will be very long in high rise buildings) we want to include wireless sensor feature in our project. In this feature, the sensor and the microcontroller part in the rooftop water tank will be connected to the relay wirelessly . We can use GSM , ZigBee Network or RF Network . In these three modes RF network will be the cheapest but less reliable , where ZigBee network will be a bit costly and the best possible way to do it as this uses microwave network and creates an wireless ad hoc network around (10-100 Meters) it [29].

For reserve tank or Multiple tanks : For reserve tank we need 2 Ultrasonic sensor in the circuit. If in case we have a multiple tank setup then we will need multiple Ultrasonic sensors added to the circuit. One single Atmega8 microcontroller can accommodate upto 5 ultrasonic distance sensor.

For Multiple Motors : For multiple motor setting we will need multiple single phase motor driver circuit. One single Atmega8 microcontroller can accommodate upto 5 SPDT (Single pole double Throw) Relay.

6.3 Limitations :

This project of ours is a prototype. It can measure the water height in the water tank and display it in the LCD along with the percentage value perfectly but to connect an actual motor with the motor driver circuit we need to add the following things to the circuit :

- Low Voltage Protection Circuit
- Surge Protection Circuit

This project of ours shows a wide range of inaccuracy when we put the sensor into small water tanker/pots ranging below 20 Inches. Due to the extra echoes it gets from the very near walls of the tanker/pots.If the tank/pot is very narrow then also the senor gets confused by the extra echos.So we can not put this sensor in those kinds of tanker or pots . But it works very well in typical 2000 Liters tank that most of the households have in their rooftop.

The ultrasonic sensor available in the market is not that efficient, if we want to increase our accuracy then we need to use more efficient sensor. Which will be a bit too costly but will give very accurate readings of the water height in the water tank.

6.4 Extension of the Project: As an extension of the project we have added a feature through which the water level can be observed on User's PC through the USB Serial port via a PC Program written exclusively for this project on C#. [31]

6.5 Conclusion:

Water is one of the most important basic needs for all living beings. But unfortunately a huge amount of water is being wasted by uncontrolled use. Some other automated water level monitoring system is also offered so far but most of the method has some shortness in practice. We tried to overcome these problems and implemented an efficient automated water level monitoring and controlling system. Our intension in this project was to establish a flexible, economical and easy configurable system which can solve our water losing problem. We have used a low cost ATmega8 microcontroller and Ultrasonic sensor in this system which is the key point to reduce cost.

We have successfully experiment the system and analyzed the results. The microcontroller-based water pump automation system provided a very satisfactory performance with a minimal percentage of error. Using this automation system, one can save manpower. This could have a substantial benefit from this project for efficient management of water.

We have used a low cost ATmega8 microcontroller and Ultrasonic sensor in this system along with different component. Pcb plays an integrated part to compact the whole system which make the. system more reliable and portable. All other automation on water pump system does not always reflect the proper water level detection and lagged when switching the water pump. But we successfully overcome all the obstrucle in our project titled water pump automation.

Appendix A

Main coding For the Microcontroller:

```
#include <mega8.h>
#include <stdio.h>
#include <stdlib.h>
#include <delay.h>
// Alphanumeric LCD functions
#include <alcd.h>

#define sonarpin PORTD.4
#define sonarpindata PIND.4

// Declare your global variables here
float distance=0,coefficient=0,temp=0,level=0;
unsigned int timerCoefficient=0,i=0;
//unsigned long int elapsedTime1=0;
unsigned int elapsedTime=0;
```

```
//unsigned long int elapsedTime=0;
unsigned char buffer[5];
eeprom float threshold=20;

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
// Place your code here
   timerCoefficient++;
   TCNT0=0;
}

void TimerStart();
void TimerStop();
void TimerReset();

void main(void)
{

unsigned char buffer1[10];
unsigned int level1=0;
PORTB=0x00;
DDRB=0x00;

PORTC=0x00;
DDRC=0x00;

PORTD=0x00;
DDRD=0xE0;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 1000.000 kHz
TCCR0=0x00;
TCNT0=0x00;
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x01;
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x33;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTB Bit 0
// RD - PORTB Bit 1
// EN - PORTB Bit 2
// D4 - PORTB Bit 4
// D5 - PORTB Bit 5
// D6 - PORTB Bit 6
// D7 - PORTB Bit 7
// Characters/line: 8
lcd_init(16);
// Global enable interrupts
#asm("sei")
coefficient=73.74;
temp=threshold;
delay_ms(2000);
while (1)
   {
      elapsedTime=0;
      //elapsedTime1=0;
      // Place your code here
      TimerReset();
```

```
//declare the pin as output pin
DDRD=0X10;
//make the pin logic high
sonarpin=1;
delay_us(5);
//make the pin logic low
sonarpin=0;
delay_us(500);
   //Declare the pin here as input
DDRD=0x00;
delay_us(60);
//wait here as long as pin is logic 0
while(sonarpindata == 0 )  { }
//here count start
TimerStart();
//wait here as long as pin is logic 1
while(sonarpindata == 1) {  }
//count stop when the signal is 0
TimerStop();
elapsedTime= timerCoefficient*256+TCNT0;
elapsedTime=elapsedTime/2;
//elapsedTime=elapsedTime+elapsedTime1;
sonarpin=0;
delay_ms(20);
sonarpin=0;

// elapsedTime=elapsedTime/5;
   //Converting elapsed time into float , this is typecasting
distance=(float)elapsedTime/coefficient;

 // Save the new threshold value onto eeprom
if(PINC.0 == 1)
{
   threshold=distance;
   lcd_clear();
   lcd_gotoxy(0,0);
```

```
    lcd_putsf("Data stored");
    delay_ms(2000);
    temp=threshold;
    lcd_gotoxy(0,0);
    lcd_putsf("                ");
    //lcd_gotoxy(0,1);
    //lcd_putse("")
  }
    //To show the water level in percentage

  level=100*distance;
  level=level/temp;

    //This is the code where we define when to turn the relay on or off,
  if(level < 10.00) PORTD.6=1;
  if(level > 90.00) PORTD.6=0;

  delay_ms(500);
  lcd_gotoxy(0,0);//This will set the LCD display point to 0 Column and 0 Row
  lcd_putsf("WATERLEVEL");//This will display the word "Water level" at 0,0
  lcd_gotoxy(11,0);//This will set the LCD display point to 11 Column and 0 Row , and will show
the threshold value ( Tank height ) to the display
  ftoa(temp,2,buffer);
  lcd_puts(buffer);
  lcd_gotoxy(0,1);//This will set the LCD display point to 0 Column and 1 Row
  ftoa(distance,2,buffer);//This will display the water height in the tank
  lcd_puts(buffer);
  lcd_gotoxy(8,1);//This will set the LCD display point to 8 Column and 1 Row
  ftoa(level,2,buffer);
  lcd_puts(buffer);
  lcd_gotoxy(15,1);
  lcd_putsf("%");
  delay_ms(500);
    //The following lines are used to send data to computer

  level1=(int)level;
```

```
        sprintf(buffer1,"%d,2",level1);
        puts(buffer1);
        delay_ms(100);

    }
}
void TimerStart()
{
   TCNT0=0x00;
   TCCR0=0x02;
}
void TimerStop()
{
    TCCR0=0x00;
}
void TimerReset()
{
   TCNT0=0x00;
   TCCR0=0x00;
   timerCoefficient=0;
}
```

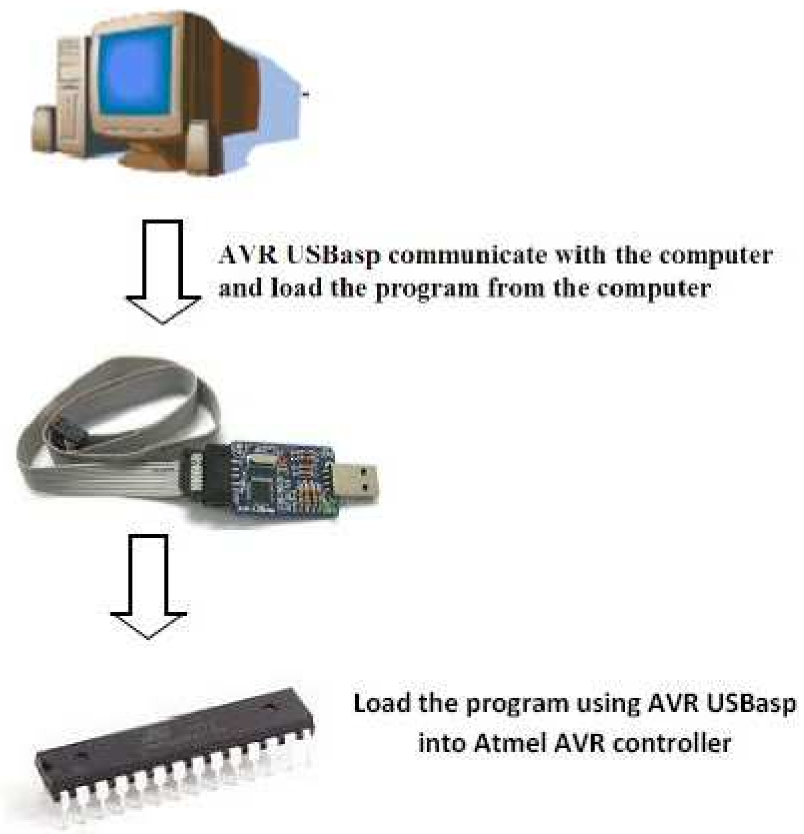
Loading the Program in a Microcontroller:

Introduction

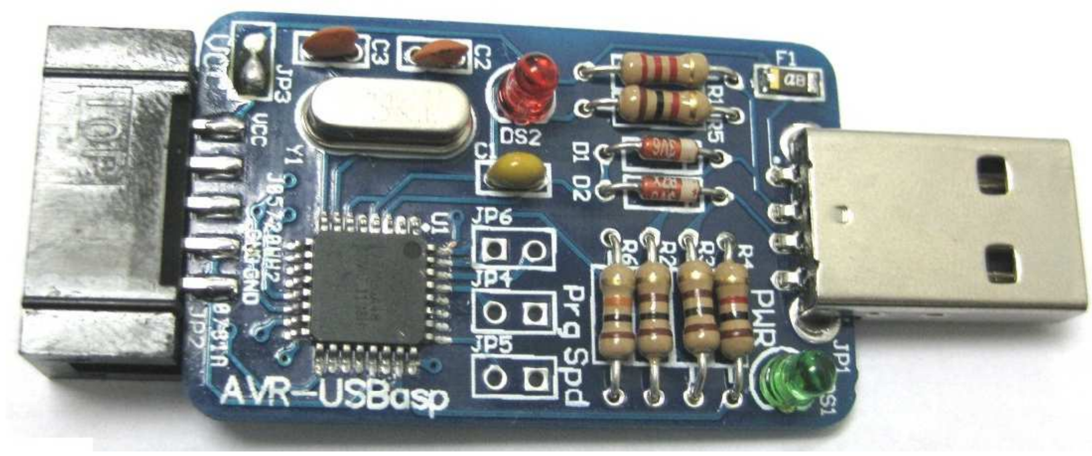
AVR USBasp is a USB in-circuit programmer and it can use to program most of the Atmel AVR controllers. It simply consists of an ATMega8 and a couple of passive components such as resistors, capacitors, LEDs and ect. The programmer uses a firmware only USB driver and there is no special USB controller is needed. By using AVR USBasp, itis easier and simpler, it just needs one step to finish the process which is to connect the AVR USBasp with computer and with microcontroller, then program it. AVR USBasp has beendesigned with capabilities and features of:

- USBasp works under multiple platforms. Linux, Mac OS X and Windows are tested.
- Its speed for the programming is up to 5kBytes/sec.
- Its SCK option is supported to the targets with low clock speed (<1.5Mhz).

USBasp Method



Board Layout and Specification



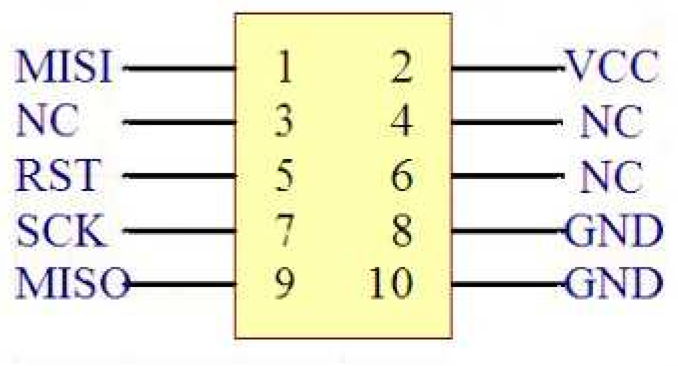
Label	Function
A	USB A type (male)
B	LEDs
C	10 pins IDC connector for interface to microcontroller
D	JP4 and JP5 for programmer and speed
E	Vcc supply

A – USB A type (male). This is for USB connection.

B – The USBASP programmer has 2 LEDs. The function is listed below:

- LED 1 (Green LED) – Power
- LED 2 (Red LED) – Programmer communicating with the target device

C – The 10 pins ISP connection provides an interface to the microcontroller. This interface uses a 10 pin IDC connector and the pin out is shown in figure below.



D – JP4 is for programmer and JP5 is for speed. Normally we are not using JP4 because JP4 function is to update the firmware on it. JP5 is only used for new microcontroller. It will not be used for others than new microcontroller because we want to increase the speed of the process of programming.

E- Vcc supply is to supply power from USB to ATMEGA target board.

Placement

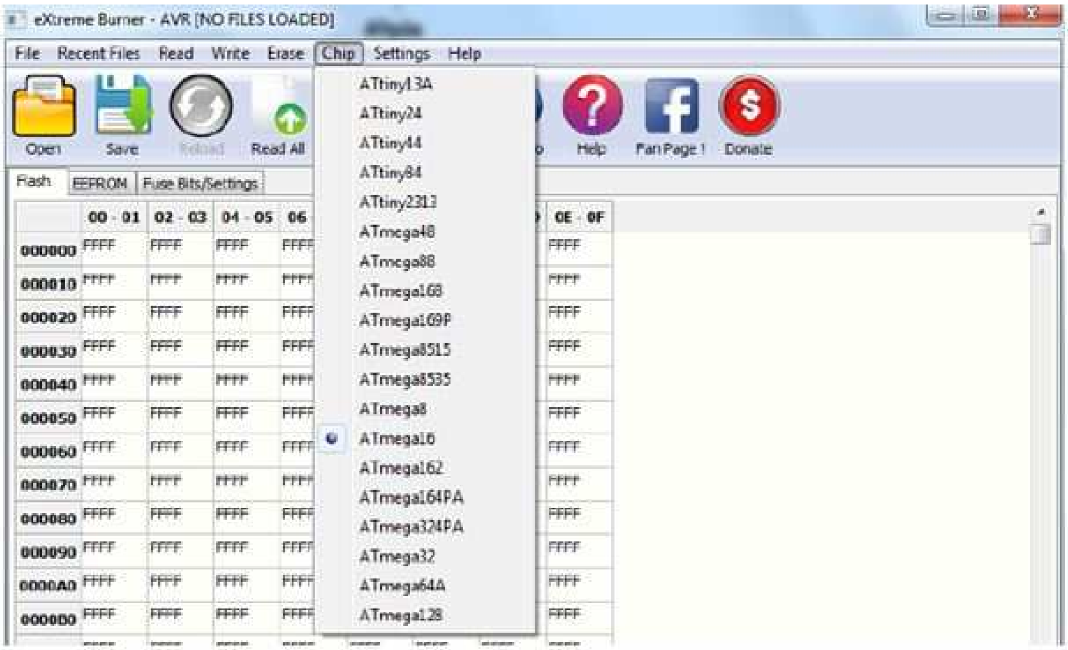
For 40pin AVR microcontroller press the button (B1) and yellow LED will lit. For other AVR microcontroller release the button and red LED will lit. This Programmer supports all the 5 volt AVR Microcontroller.

Burner Software

Programming steps:

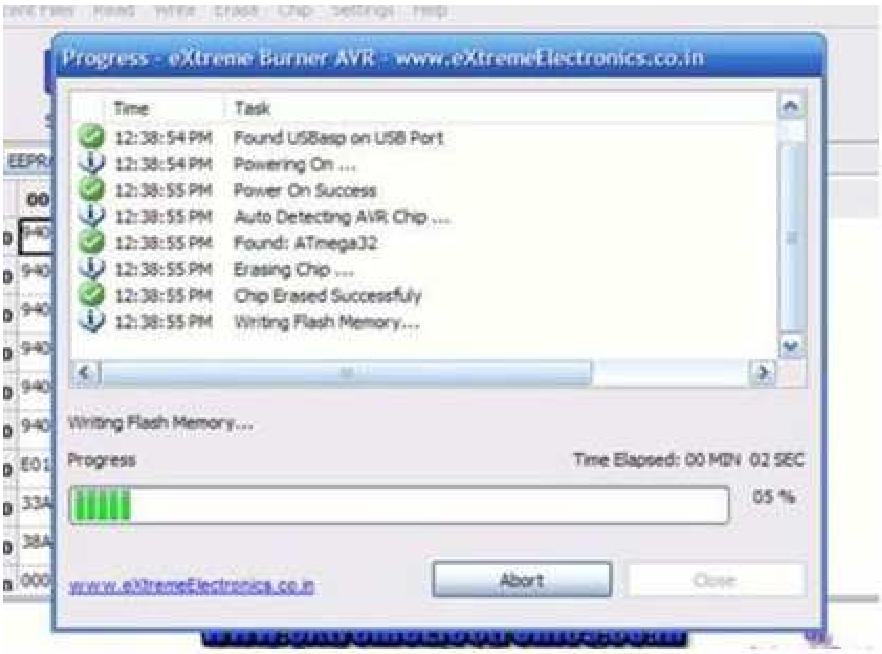
- Plug the AVR Programmer into windows PC.

- Install the “Extreme Burner V1.2”. Driver will be installed with the installation process.
- Open Extreme Burner.
- Under drop-down menu choose your desired chip.

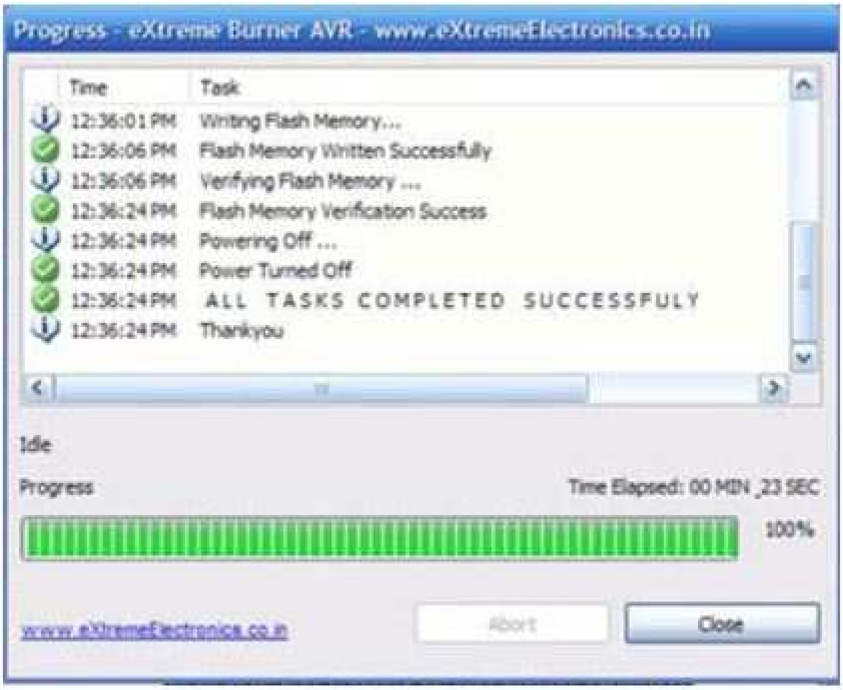


- Click on the Fuse Bits/Setting tab and choose “Read All” for read your chip’s Fuse Bytes.

- If you want to write Fuse Bytes into your chip, write down it on the blank boxes, check the “write”, and then press write.
- For loading Hex file press “Open” and browse the hex file. From the drop-down menu choose Write>Flash



- If programming is successful you will see just like below: [23][24]



Appendix B

Hardwareparts

Introduction of Microcontroller:

A microcontroller (sometimes abbreviated µC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. It is a simplified CPU, plus some amount of RAM, plus some amount of (re)programmable ROM, plus some I/O ports (including some analog I/O ports), and all of this in a single small chip.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, and toys. These are also used at many electronic devices, including microwave ovens and washing machines. They are simple and useful enough to be used in many DIY (do it yourself) projects. They usually can run at a clock rate of a few MHz.

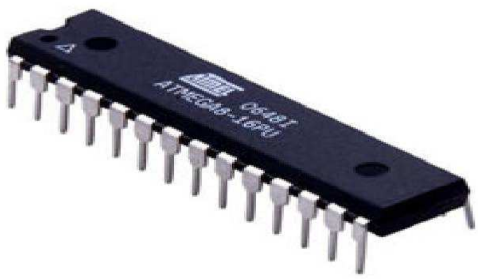


Figure :A microcontroller

The controller that we have used for our project is an Atmega8 microcontroller which acts as our Project's brain.

Physical configuration:

To assembled a microcontroller into an electrical component where it will access inner blocks through the outside pins. The picture below shows what a microcontroller looks like inside.

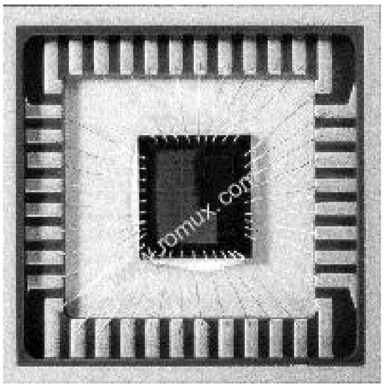


Figure: Physical configuration of the interior of a microcontroller

Internal connection of a microcontroller:

The following chart represents the center section of a microcontroller.Thin lines which lead from the center towards the sides of the microcontroller represent wires connecting inner blocks with the pins on the housing of the microcontroller so called bonding lines.

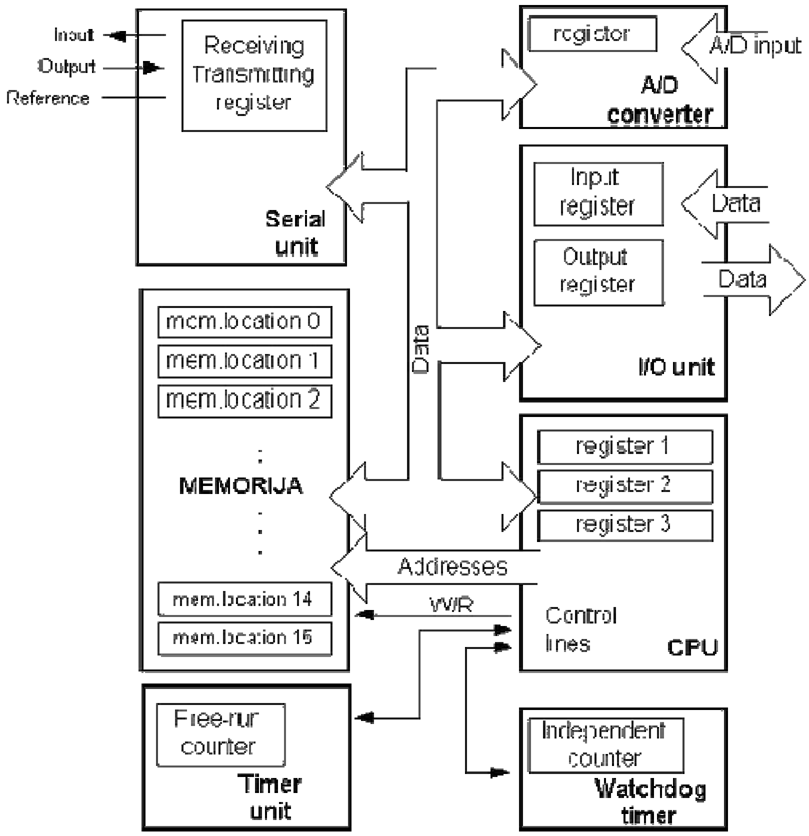


Figure: Internal connection of a microcontroller

Pin Description of ATmega8 Microcontroller:

- **VCC** Digital supply voltage.
- **GND** Ground.
- **Port B (PB7...PB0)**

XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit).The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Depending on the clock selection fusesettings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

- **Port C (PC5..PC0)**

Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

- **PC6/RESET**

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

- **Port D (PD7..PD0)**

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega8.

- **RESET**

Reset input.A low level on this pin for longer than the minimum pulse length will generate reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

- **AVCC**

AVCC is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that Port C (5..4) use digital supply voltage, VCC.

- **AREF**

AREF is the analog reference pin for the A/D Converter.

- **ADC7..6 (TQFP andQFN/MLF PackageOnly)**

In the TQFP and QFN/MLF package, ADC7..6 serve as analog inputs to the A/D converter.These pins are powered from the analog supply and serve as 10-bit ADC channels.

- **I/O-Register parts:**

General

- the ATmega8 has **three I/O register**
  - io-register B
  - io-register C
  - io-register D

- each register has a size of **one byte** (eight bits)
- every bit of these registers are physically **pins** *(0V = 0; 5V = 1)*
- every pin can use either as **input** or **output** pin

- the *direction* (in or out) of each pin can specified via the **DDRx** register (Data Direction Register)
  - **0** = read (receive)
  - **1** = write (send)

- every I/O register has a data direction register
  - io-register B => **DDRB**

- io-register C => **DDRC**
- io-register D => **DDRD**

- if a register an **out register** (means all pins are "out pins"), you can write whole bytes using the **PORTx** register
  - io-register B => **PORTB**
  - io-register C => **PORTC**
  - io-register D => **PORTD**

- if a register an **in register** (means all pins are "in pins"), you can read whole bytes using the **PINx** register
  - io-register B => **PINB**
  - io-register C => **PINC**
  - io-register D => **PIND**

Memory: It has 8 Kb of Flash program memory (10,000 Write/Erase cycles durability), 512 Bytes of EEPROM (100,000 Write/Erase Cycles). 1Kbyte Internal SRAM

I/O Ports: 23 I/ line can be obtained from three ports; namely Port B, Port C and Port D.

Interrupts: Two External Interrupt source, located at port D. 19 different interrupt vectors supporting 19 events generated by internal peripherals.

Timer/Counter: Three Internal Timers are available, two 8 bit, one 16 bit, offering various operating modes and supporting internal or external clocking.

SPI (Serial Peripheral interface): ATmega8 holds three communication devices integrated. One of them is Serial Peripheral Interface. Four pins are assigned to Atmega8 to implement this scheme of communication.

USART: One of the most powerful communication solutions is USART and ATmega8 supports both synchronous and asynchronous data transfer schemes. It has three pins assigned for that. In many projects, this module is extensively used for PC-Micro controller communication.

TWI (Two Wire Interface): Another communication device that is present in ATmega8 is Two Wire Interface. It allows designers to set up a commutation between two devices using just two wires along with a common ground connection, As the TWI output is made by means of open collector outputs, thus external pull up resistors are required to make the circuit.

Analog Comparator: A comparator module is integrated in the IC that provides comparison facility between two voltages connected to the two inputs of the Analog comparator via External pins attached to the micro controller.

Analog to Digital Converter: Inbuilt analog to digital converter can convert an analog input signal into digital data of 10bit resolution. For most of the low end application, this much resolution is enough.

Resistors:

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. The current through a resistor is in direct proportion to the voltage across the resistor's terminals.

In our project we connect 1K ohm resistor with each LEDs. Because LED should never connect directly to a battery or power supply. Unless it will destroy almost instantly because too much current will pass through and burn it out.LEDs must have a resistor in series to limit the current to a safe value, for quick testing purpose a1kΩ resistor is suitable for most LEDs if our supply voltage is 12V or less.

Figure: A 1K ohm resistor

We also use a 10k ohm resistors to connect between vdd and MCLR' microcontroller pin in our circuit. Used as a weak pull up resistor in switch circuits and timed R/C.

Capacitor:

A capacitor (originally known as condenser) is a passive two-terminal electrical component used to store energy in an electric field. The forms of practical capacitors vary widely, but all contain at least two electrical conductors separated by adielectric (insulator); for example, one common construction consists of metal foils separated by a thin layer of insulating film. Capacitors are widely used as parts of electrical circuits in many common electrical devices.

When there is a potential difference (voltage) across the conductors, a static electric fielddevelops across the dielectric, causing positive charge to collect on one plate and negative charge on the other plate. Energy is stored in the electrostatic field. An ideal capacitor is characterized by a single constant value, capacitance, measured in farads. This is the ratio of the electric charge on each conductor to the potential difference between them.

A capacitor is composed of two conductors separated by an insulating material called a DIELECTRIC. The dielectric can be paper, plastic film, ceramic, air or a vacuum. The plates can be aluminum discs, aluminum foil or a thin film of metal applied to opposite sides of a solid dielectric.

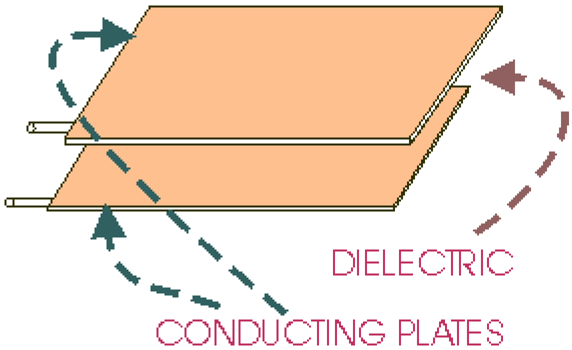


Figure : Internal parts of a capacitor

Capacitors are widely used in electronic circuits for blocking direct current while allowing alternating current to pass, in filter networks, for smoothing the output of power supplies, in the resonant circuits that tune radios to particular frequencies, in electric power transmission systems for stabilizing voltage and power flow, and for many other purposes.

Figure: Different types of capacitor

Linear regulator IC:

Integrated circuit (IC) linear voltage regulators use an active pass element to reduce the input voltage to the regulated output voltage. The use of voltage-controlled sources enables IC linear voltage regulators to force a fixed voltage to appear at the output terminal. Control circuitry monitors the output voltage and adjusts the current source accordingly.

IC linear voltage regulators can provide positive, negative, or both positive and negative polarity. With positive polarity, the output voltage is in phase (positive) with the input voltage. With negative polarity, the output voltage is out of phase (negative) with the input voltage.

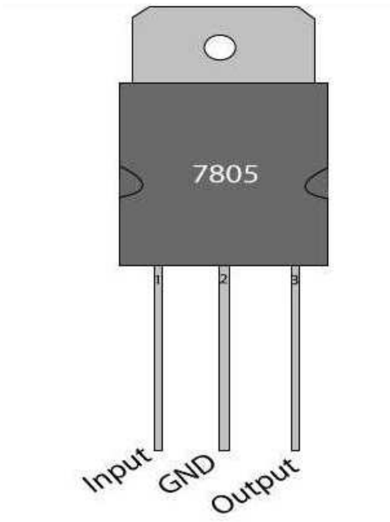


Figure: Linear Regulator IC 7805

IC linear voltage regulators are available in a variety of package types and life stage levels. They are used in industrial,automotive, aerospace, and military applications, as well as in consumer electronics and telecommunications. 7805 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value. The xx in 78xx indicates the fixed output voltage it is designed to provide. 7805 provides +5V regulated power supply. Capacitors of suitable values can be connected at input and output pins depending upon the respective voltage levels.

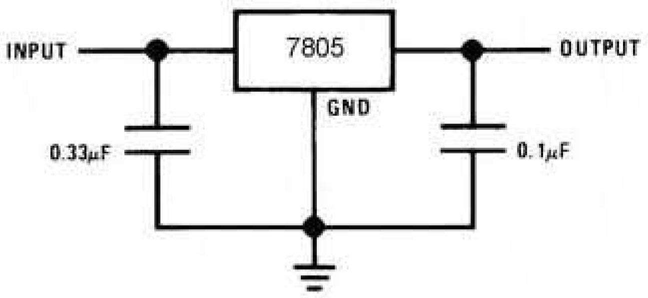


Figure: Internal circuit diagram of 7805 IC

Pin Description

Pin No	Function	Name
1	Input voltage (5V-18V)	Input
2	Ground (0V)	Ground
3	Regulated output; 5V (4.8V-5.2V)	Output

In the IC 7805, Excess energy is dissipated as heat. This elegantly simplistic regulation mechanism pays dearly in terms of lost power. Because of this, linear regulators are associated with excessive dissipation, in efficiency, high operating temperatures and large heat sinks.

LCD display:

A liquid-crystal display (LCD) is a flat panel display, electronic visual display, or video display that uses the light modulating properties of liquid crystals. Liquid crystals do not emit light directly.LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images which can be displayed or hidden, such as preset words, digits, and 7-segment displays as in a digital clock. They use the same basic technology, except that arbitrary images are made up of a large number of small pixels, while other displays have larger elements.In our project we have used it for monitoring unit.

Figure: 8-bit a liquid-crystal display (LCD)

Printed circuit board (PCB):

PCB stands for "Printed Circuit Board." A PCB is a thin board made of fiberglass, composite epoxy, or other laminate material. Conductive pathways are etched or "printed" onto board, connecting different components on the PCB, such as transistors, resistors, and integrated circuits. A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCB's can be single sided (one copper layer), double sided (two copper layers) or multi-layer. Conductors on

different layers are connected with plated-through holes called vias. Advanced PCB's may contain components - capacitors, resistors or active devices - embedded in the substrate.
Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs are more costly to design but allow automated manufacturing and assembly. Products are then faster and cheaper to manufacture, and potentially more reliable.

Design

Printed circuit board artwork generation was initially a fully manual process done on clear mylar sheets at a scale of usually 2 or 4 times the desired size. The schematic diagram was first converted into a layout of components pin pads, then traces were routed to provide the required interconnections. Pre-printed non-reproducing mylar grids assisted in layout, and rub-on dry transfers of common arrangements of circuit elements (pads, contact fingers, integrated circuit profiles, and so on) helped standardize the layout. Traces between devices were made with self-adhesive tape. The finished layout "artwork" was then photographically reproduced on the resist layers of the blank coated copper-clad boards. Modern practice is less labor intensive since computers can automatically perform many of the layout steps.

The general progression for a commercial printed circuit board design :

1. Schematic capture through an electronic design automation tool.
2. Card dimensions and template are decided based on required circuitry and case of the PCB. Determine the fixed components and heat sinks if required.
3. Deciding stack layers of the PCB. 1 to 12 layers or more depending on design complexity. Ground plane and power plane are decided. Signal planes where signals are routed are in top layer as well as internal layers.
4. Line impedance determination using dielectric layer thickness, routing copper thickness and trace-width. Trace separation also taken into account in case of differential signals. Microstrip, stripline or dual stripline can be used to route signals.
5. Placement of the components. Thermal considerations and geometry are taken into account. Vias and lands are marked.
6. Routing the signal traces. For optimal EMI performance high frequency signals are routed in internal layers between powers or ground planes as power planes behave as ground for AC.
7. Gerber file generation for manufacturing.
8. In the design of the PCB artwork, a power plane is the counterpart to the ground plane and behaves as an AC signal ground, while providing DC voltage for powering circuits mounted on

the PCB. In electronic design automation (EDA) design tools, power planes (and ground planes) are usually drawn automatically as a negative layer, with clearances or connections to the plane created automatically.

Advantages of PCB :

1. The circuit board fabrication cost (pcb cost) is lower with mass quantity production
2. Electronic circuit characteristics will be maintained without introducing parasite capacitance with a proper circuit board design.
3. Component wiring and assembly can be mechanized in a circuit board manufacturing facility.
4. PCB's offer uniformity of electrical characteristics from assembly to assembly.
5. The location of electronic parts is fixed and so it simplifies components identification and maintenance of equipment.
6. Inspection time is reduced because printed circuitry eliminates the probability of error.
7. Chances of miswiring or short-circuited wiring are minimized.

References

[1] Rifkin, Jeremy (1995). The End of Work: The Decline of the Global Labor Force and the Dawn of the Post-Market Era. Putnam Publishing Group. pp. 66, 75.ISBN 0-87477-779-8.

[2] Bennett, S. (1993). A History of Control Engineering 1930-1955. London: Peter Peregrinus Ltd. On behalf of the Institution of Electrical Engineers. ISBN 0-86341-280-7

[3] Bennett, S. (1979). A History of Control Engineering 1800-1930. London: Peter Peregrinus Ltd. pp. 47, 266. ISBN 0-86341-047-2

[4] Field, Alexander J. (2011). A Great Leap Forward: 1930s Depression and U.S. Economic Growth. New Haven, London: Yale University Press. ISBN 978-0-300-15109-1<Factory productivity was greatly increased by electrification in the 1920s. Manufacturing productivity growth fell from 5.2%/yr 1919-29 to 2.76%/yr 1929-41. Field notes that spending on non-medical instruments increased significantly from 1929-33 and remained strong thereafter.)>

[5] Water Supply of Dhaka City - Unnayan Onneshan http://www.unnayan.org/documents/RightsParticipation/water_supply_dhaka.pdf , p. 2 , p. 14

[6] Azharul Haq, Khondaker (June 2006). "Water Management in Dhaka". Water Resources Development 22 (2): 291–311. doi:10.1080/07900620600677810., p. 296

[7] www.parallax.com/product/28015

[8] http://learn.parallax.com/propeller-c-simple-devices/sense-distance-ping

[9] Heath, Steve (2003). Embedded systems design. EDN series for design engineers (2 ed.). Newnes. pp. 11–12.ISBN 9780750655460.

[10] http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_l_datasheet.pdf

[11] https://solarbotics.com/product/re1/

[12] Mason, C. R. "Art & Science of Protective Relaying, Chapter 2, GE Consumer & Electrical". Retrieved October 9, 2011.

[13] http://spemco.com/Resources/Electronics101/Relays101/tabid/101/Default.aspx

[14] "Arc Suppression to Protect Relays From Destructive Arc Energy". Retrieved December 6, 2013.

[15] A.C. Keller. "Recent Developments in Bell System Relays -- Particularly Sealed Contact and Miniature Relays". The Bell System Technical Journal. 1964.

[16] http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet

[17] https://www.sparkfun.com/products/255

[18] http://arduino.cc/en/Tutorial/LiquidCrystal

[19] http://msdn.microsoft.com/en-us/library/ee658098.aspx

[20] Lawlis, Patricia K. (August 1997). "Guidelines for Choosing a Computer Language: Support for the Visionary Organization". Ada Information Clearinghouse. Retrieved 18 July 2006.

[21] http://www.arctan.ca/files/cvavrman.pdf

[22] http://www.hpinfotech.ro/html/cvavr_features.htm

[23] http://www.cytron.com.my/usr_attachment/AVR-USBasp%20User%20Manual.pdf

[24] http://extremeelectronics.co.in/downloads/usbavrprogrammer/docs/Tutorial.pdf

[25] www.circuitstoday.com

[26] www.galco.com

[27] www.wisegeek.org

[28] www.engineersgarage.com

[29] "ZigBee Specification FAQ". Zigbee Alliance. Zigbee Alliance. Retrieved 14 June 2013.

[30] http://newengineeringpractice.blogspot.com/p/programming-pic-microcontrollers.html

[31] www.sourcecodester.com