



# **ARDUINO BASED SMART PLANT WATERING SYSTEM**

**Submitted by**

**Azim Ahmed**

**BME 1601008148**

**Abdullah Al Mamun**

**BME 1602009195**

**Md. Sobuj Ali**

**BME 1602009197**

**Submitted to**

**Department of Mechanical Engineering  
Sonargaon University (SU)  
Dhaka 1215 ,Bangladesh**

**Supervised by**

**Md. Ali Azam**

Lecturer

Department of Mechanical Engineering  
Sonargaon University (SU)  
Dhaka 1215, Bangladesh

February, 2020

# Declaration of Authorshop

We, Azim Ahmed ,Abdullah Al Mamun and Md. Sobuj Ali declare that this thesis titled,” ARDUINO BASED SMART PLANT WATERING SYSTEM” and the work presented in it are our own and has been generated by me as the result of my own original research .

We confirm that :

- 1.This work was done wholly or mainly while in candidature for a research degree at this University .
- 2.Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution , this has been clearly stated .
- 3.Where we have consulted the published work of others ,this is always clearly attributed .
4. Where we have quoted from the work of others , the source is always given .With the exception of such quotations ,this thesis is entirely our own work .
5. We have acknowledge all main sources of help .
6. None of the part of this work has been published before submission .

Signed :

Azim Ahmed

Abdullah Al Mamun

Md. Sobuj Ali

Supervised by

Md. Ali Azam

Lecturer

Dept. of Mechanical Engineering  
Sonargaon University

Date:03.02.2020

# Abstract

The team worked on the Arduino based automatic plant watering system with a built in pump and a water tank .This system is useful especially for user's who have a day travel a lot as this system can water plants only when necessary without any human intervention .What makes our project unique than the other innovations in the current market from watering plants is it's built in function which allows the user to be updated from time to time about the status and plant . Proper watering is exigent for the plants and trees which are grown in containers. Many people un surmounted to take good care of expensive plants or trees like Bonsai because of improper timing or lack of watering. An automated system which is capable of perceiving the proper situation of watering plants can be convenient in this regard. In this paper, a micro-controller board Arduino has been used, which has an ATMEGA 328p chip in it. A couple of sensors are integrated into it for detecting daylight, moisture of soil and water level in the plant's pot. A program has been set to the Arduino to take the norm from the sensor, comparing the retrieved value with our predefined standard threshold and turn on or off the water pump according to the plants need. A message alert system has been set to notify the owner if there is any problem in supplying water from the main source.

# Acknowledgement

First of all we are very grateful to almighty Allah ,who has blessed us with knowledge and ability to write this report successfully .

We would like to express our deepest gratitude to our supervisor **Md. Ali Azam**, for his guidance on this project showing us the path of conduction successful research and above all for always being there as our mentor. He shared his wisdom with us in analyzing subject matters and at the same time valued our thinking approach to synthesizing thus topics. His suggestions drove us towards better ways of thinking, has reviews enriched us in solving problems and his support gave us strength at the time of our disappointment.

We shall forever cherish the moments of working with him.

We acknowledge with appreciated the cooperation of **Md. Mostofa Hossain** (Associate Professor & Head of Mechanical Department, Sonargaon University) for him help at various stage of our project work.

I also want to thank Professor **M.A Razzaque**, Vice-chancellor, Sonargaon University for his assistance.

We deeply thank our friends and families for always believing in us even at the moment when we were losing our confidence

Thanks to all, May Allah bless all of you.

# TABLE OF CONTENTS

Abstract	iii
Acknowledgement	iv
TABLE OF CONTENTS	v
List of Figure	vii
<b>Chapter 1</b>	
1.1 Introduction	8
1.2 Objectives	9
<b>Chapter 2</b>	
2.1 Literature Review	10
<b>Chapter 3</b>	
3.1 Project Planning	13
3.2 Block Diagram	13
3.3 Schematic Diagram & Working Explanation	15
3.4 Description of the Circuit Diagram	16
<b>Chapter 4</b>	
4.1 Components Used	18
4.2 Project Setup	18
<b>Chapter 5</b>	
5.1 Arduino Nano	20
5.2 Programming	21
5.3 Warning	22
5.4 Difference with others boards	22
5.5 Power	22
5.6 The power pins are follows	23
5.7 Memory	23
5.8 Communication	24
5.9 Automatic (Software) Reset	24
5.10 Physical Characteristic	25
5.11 Schematic & Reference Design	25
5.12 Input and Output	25
5.13 USB Overcurrent Protection	26
5.14 Summary	26

<b>Chapter 6</b>	
6.1 Arduino Details	27
6.2 Power (USB/Barrel Jack)	27
6.3 Pins	28
6.4 Reset Button	29
6.5 Power LED Indicator	30
6.6 TX RX LEDs	31
6.7 Main IC	32
6.8 Voltage Regulator	33
<b>Chapter 7</b>	
7.1 Description of Tmega328 Microcontroller	35
7.2 Block Diagram of ATmega328p	37
<b>Chapter 8</b>	
8.1 LCD Panel	38
8.2 Water Level Sensor	39
8.3 Soil Moisture Sensor	40
8.4 Water Pump	41
	42
<b>Chapter 9</b>	
Conclusion	42
<b>Chapter 10</b>	
Recommendation	43
<b>Chapter 11</b>	
Reference	44
Appendix	46

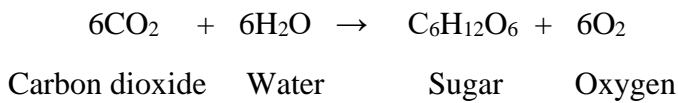
# LIST OF THE FIGUREURES

Figure 2.1 Overview of Automated Watering System	12
Figure 3.2(i) Automatic plant watering block diagram	14
Figure 3.2(ii) Block diagram of hardware configuration	15
Figure 3.3 Schematic diagram of Smart plant watering system	16
Figure 3.4 Schematic diagram of Smart plant watering system	17
Figure 5.1 Arduino Nano board with labeled parts	20
Figure 6.2 AC-to-DC adapter	28
Figure 6.3 Pins	29
Figure 6.4 Reset Button	30
Figure 6.5 Power Led Indicator	31
Figure 6.6 TX RX LEDs	32
Figure 6.7 Main IC	33
Figure 6.8 Voltage Regulator	33
Figure 7.1 Microcontroller ATmega328p	35
Figure 7.2 Block diagram of ATmega328p	37
Figure 8.1 LCD display	38
Figure 8.2 Water Level Sensor	39
Figure 8.3 Soil Moisture Sensor	40
Figure 8.4 Water Pump	41

# Chapter 1

## 1.1 Introduction:

The process of producing foods by plants or trees is called Photosynthesis. During photosynthesis, plants and their organisms use Carbon Dioxide from Air and Water from soil or other source in the presence of light and produce Sugar and Oxygen. The chemical reaction occurs in photosynthesis is:



The core elements of photosynthesis are light, water and carbon dioxide, which they need together during photosynthesis. Plants can get the carbon dioxide anytime as it can be found from the air and also daylight can be provided if it is kept in the right place. Most of the time for the lacking of water, the plants cannot raise well and become morbid. On the other hand, supply of excessive water at a time can result a venturesome effect on the plants especially the sensitive trees like bonsai. Such trees also need timely water supply for the perfect growth. That means, for the perfect growth, water is needed in proper time along with proper quantity. Currently there are some micro-controller based plant watering project available, but the problem with the projects are, they don't have a particular time for watering. But there are some sensitive and expensive plants which need water at the particular time and unusual watering could cause them to death. Moreover, the best time for watering is early morning, while the environment is cool. This allows water to go into deep and reach the roots of the plants without too much excess water lost to vaporization. Watering at night could cause damage in the base of the plant and could also be responsible for fungal problems such as, powdery mildew or sooty mold, which is very harmful for plants. Besides, the existing works do not have the feature to track any leakage in water supply or identify the shortage in the water source. As a result, any implausible situation can be created in absence of person at home.

To extrude all of these problems, we have propounded a system for automated watering plants to make sure that the plants are getting water when its moisture of the soil is below a threshold value and sunlight is available.

The system has also feature of preventing over watering by measuring water level height, maintaining continuous water supply so that the plants never gets de-moisturized and notify the owner by sending a particular message about the leakage in water supply or lack of water in the main water source.

For the automatic plant watering system ,a moisture sensor is attached to the soil of the plant and when the reading of the sensor is bellow the pre-set value , the pump will be activated . On the other hand , when the reading of the sensor is above the pre-set value , the pump will deactivated .The pump will pump water from a built in water tank .



## **1.2 Objectives :**

The objectives of this project is

- 1.To provide the necessary moisture for the crop development .
- 2.To reduce such wastages of water for better utilization of resources .
3. To design a small-scale automated irrigation system based on water level sensor content .
- 4.To ensure sufficient supply of water during droughts of short duration and unpredictable climate .
- 5.To improve the ambient condition for the vegetative growth .

# Chapter 2

## 2.1 literature review :

ATMEGA microcontroller based plant irrigation system is proposed in [1]. A moisture sensor used in the system which further connected with an op-amp and used to control the relay. But they used only a single sensor to handle the whole situation and the system cannot find the existence of any leaks in the system. A smart irrigation system is propounded in [2] for the savings of water in the large field. But they considered only soil and temperature sensor in which temperature sensor is almost of no use. Moisture sensor can handle the whole criteria. Besides, the disadvantage of the system is that, it cannot understand the water level, as a result, savings of water actually not optimized. In [3] States feature of their system .

>The system supports water management decision ,used for monitoring the whole system with GSM(RS232) module

>The system continuously monitors the water level (Water level Sensor) in the tank and provide accurate amount of water required to the plant or tree (crop).

>The system checks the temperature ,and humidity of soil to retain the nutrient composition of the soil managed for proper growth of plant .

>Low cost and effective with less power consumption using sensors for remote monitoring and controlling devices which are controlled via SMS using a GSM using android mobile .In Irrigation Control System Android and GSM for Efficient Use of Water and Power – Laxmi Shabadi ,Nandini Patil ,Nikita. M, Shruti. J, Smitha. P & Swati . C[4], an automatic irrigation system using ZigBee in WSN has been proposed. They used temperature, soil moisture and the air humidity sensor to control the irrigation system. But the system is not capable of predicting the best timing of watering and also unable to measure the water level. Moreover, there is no auto notification system if there is a shortage in water supply. An automatic irrigation system has been proposed in [5] using solar power. But the most downside of a solar power system is that it hardly works at the rainy season or winter. If the crops are in a shaded zone, crops won't get water and the system may fail. Besides, the system will only work in paddy fields. There is also another pullback of the

system is that the system only measures the water level above the ground. Here, the fact is, water doesn't always stay on the ground after watering. The soil absorbs it

and reaches them to the plant's root. Now, if the base of a plant is always submersed in water, it could get rotten and the plants would die. In such situation, the system may cause bad damage. In [6]Remote Sensing and Control of An Irrigation System Using a Distributed Wireless Sensor Network Yunseop (James) Kim, Member, IEEE ,Robert G. Evans ,And William M. Iversen [7] The setup of technical system describe in this paper is broad based and is relatively one of the efficient system that has

developed windows application to monitor the field .Field is equipped with wireless communication and covers wider field area . Detailed description on site field sensor and Internet technology is described briefly . The statistical data provide is measured to be efficient and used for research work .Microcontroller Based Automatic Plant Irrigation System .Venkata Naga Rohit Guntri [8], It planter has been introduced which is an indoor cultivation device that has LEDs and water pump controlled by PIC. This device is for researching plants by monitoring temperature, light and water level using thermos, photo and water level sensor in it. Some water measurement results have also been reported. In [9], the water status of economic plants has been monitored using continuous terahertz wave radiation. Since any radiation is harmful for human body or any animal, so do terahertz radiation. But using a sensor for monitoring the water status would be effective and healthy rather than terahertz radiation.[10] Irrigation by help of freshwater resources in agricultural areas has a crucial importance .Traditional instrumentation based on discrete and wired solutions ,presents many difficulties on measuring and control systems especially over the large geographical areas .If different kinds of sensors (that is ,temperature ,humidity ,and etc) are involved in such irrigation in future works ,it can be said that an internet based remote control of irrigation automation will be possible .The developed system can also transfer fertilizer and the other agricultural chemicals (calcium, sodium, ammonium, zinc) to the field with adding new sensors and valves .Solar Powered Smart Irrigation System ,Advance in Electronic and Electric Engineering[11]

>cost effective solar power can be the answer for all our energy needs .

>Conserves electricity by reducing the usage of grid power and conserves water by reducing water losses .

A typical drip water supply assembly is shown in Figure (2.1) below

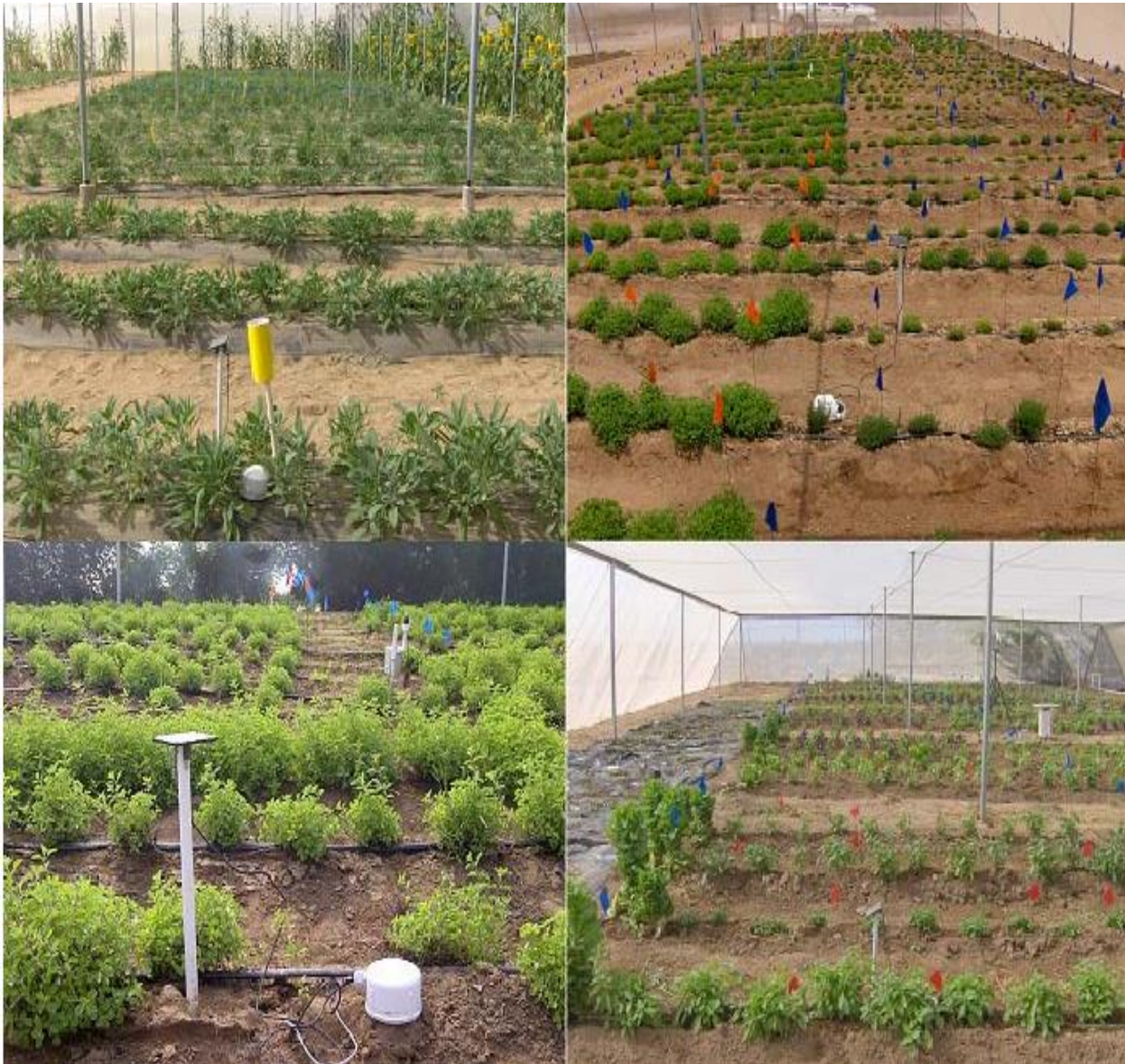


Figure :2.1 Overview of Automated Watering System

# Chapter 3

## 3.1 Project Planning :

Project planning is the process of defining the scope, objectives and ways in general to achieve them. This process goes before a plan as it is here where everything is discussed. All crucial planning points and outputs then make up a plan.

A project plan is a formal document that contains all planning decisions, approved project scope and costs. Apart from mentioned things, its main goals are to guide control, facilitate communication between stakeholders and schedule baselines.

Creating a project plan, a manager should have a good command of key knowledge and skills from the very beginning. Knowing them, he/she will raise the bar for successful project's implementation. It will also help to foresee and avoid unnecessary mistakes and wrong decisions as well as simply help to save time and reduce costs.

- a. Analysis of the existing situation and the exact nature of problem faced through discussions with the project guide .
- b. Study of process of different technologies used in the system .
- c. With the help of the guide the specification of the program were decided and then implemented in the project .
- d. Use of Accelerometer sensor to interface the computer and embedded system meant for process and control .
- e. Testing ,development and troubleshooting still under way to enhance user interface .

## 3.2 Block Diagram

The whole system is based on the microcontroller board Arduino which is integrated two sensors-soil moisture sensor, water level sensor . A Single Channel Relay module has been used to activate and deactivate the Water Pump and the 1602 LCD Display is attached to show the current sensor values as well as to report the current status of the soil, water level, daylight and water pump. A 5v DC power supply is needed to power up the circuit .The function of the moisture sensor is to

sense the level of moisture in the soil ,the water level sensor is to sense the level of water in the soil .The water pump supplies water to the plants .

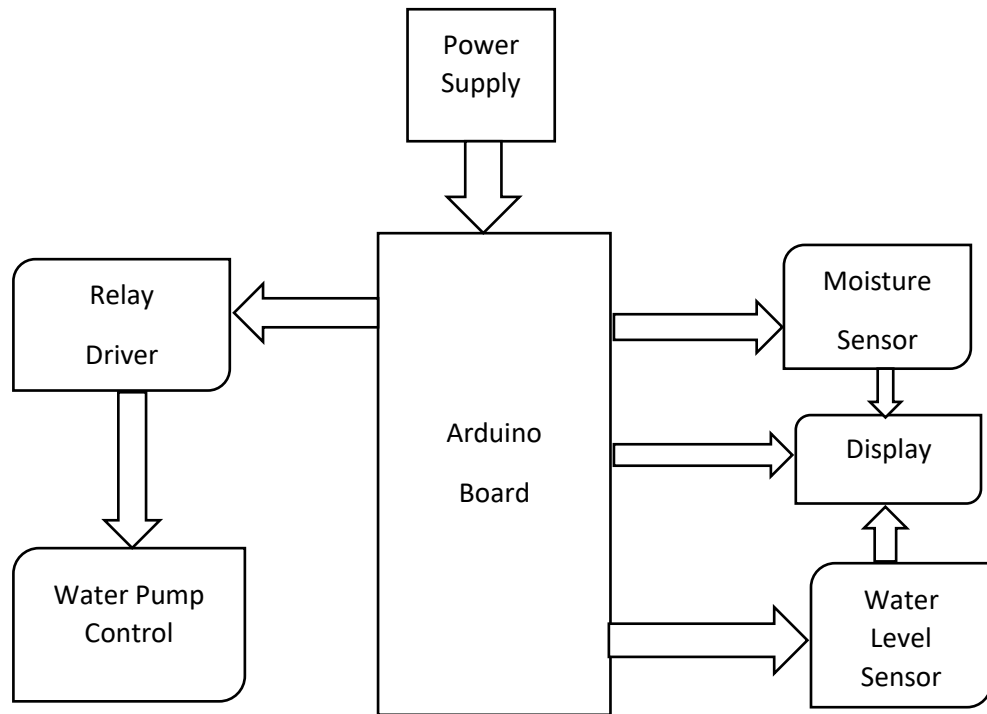


Figure 3.2 (i)Automatic Plant Watering Block Diagram

The Arduino Nano can be powered via an external power supply which can come either from an AC-to-DC adapter

or battery and use USB connection . The board can operate on an external supply of 6 to 20 volts .If supplied with less than

7V , however ,the 5V pin may supply less than five volts and the board may be unstable .If using more than 12V,

the voltage regulator may overheat and damage the board .The recommended range is 7 to 12 volts

Above is the manner in which we are going to implement the hardware circuit .The first part of the block diagram is different sensor and the second part is an LCD panel and motors for supplying water .

The major hardware modules which are needed ,Arduino processor ,motor ,different sensors and LCD panel .

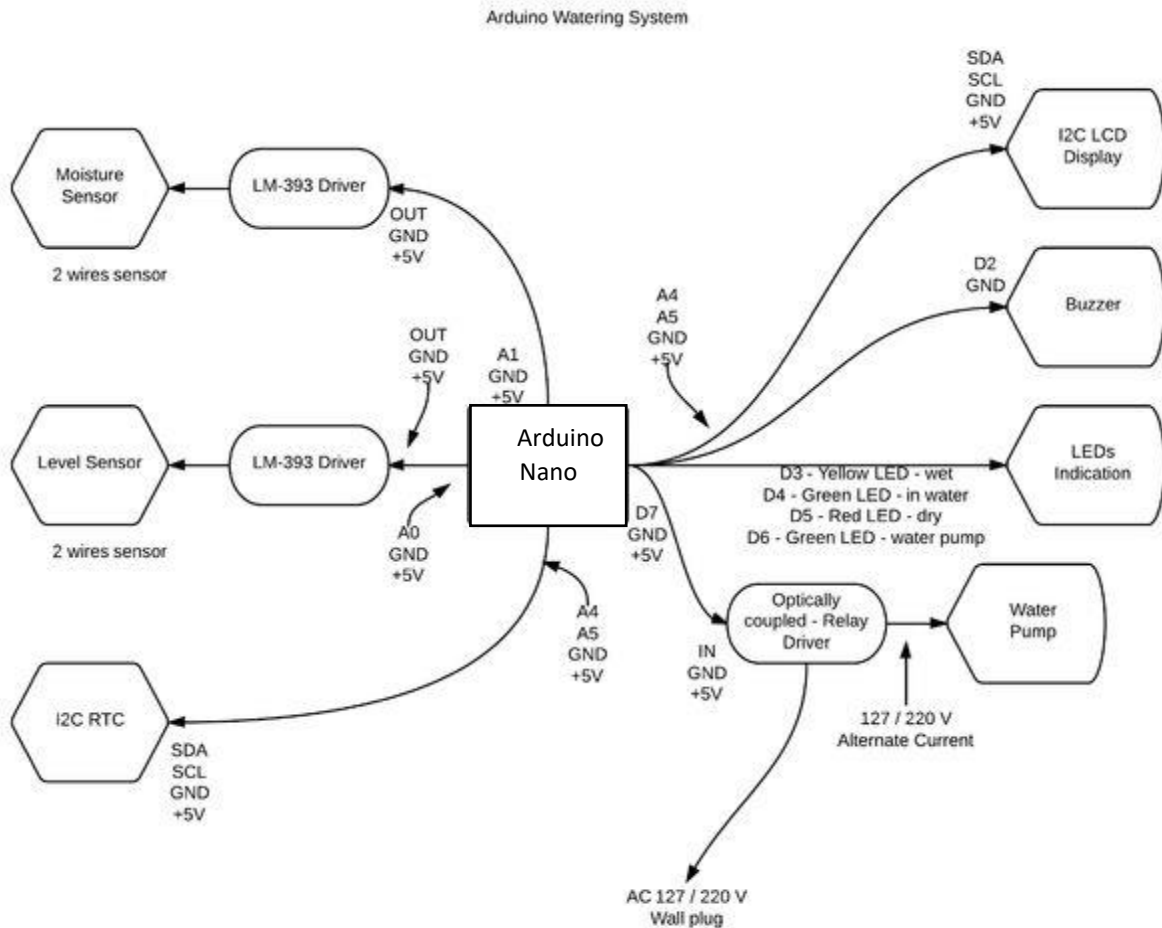


Figure.3.2(ii) Block diagram of Hardware Configuration

### 3.3 Schematic Diagram & Working Explanation :

Moisture sensor senses the volumetric water content in soil .After inserting a probe into soil for approximately 60 seconds ,it starts feeding the controller with some live moisture data .Code was written in such a way that Arduino decides the cut off level of dry soil and wet soil using the data provided by the sensor .

Once Arduino decides whether the soil was in wet or dry condition it will decide either to activate the water pump .If the sensor reading indicates the soil is dry it will activate a simple DC motor which will pump the water to soil .By this way the moisture in the soil . By this way the moisture in the soil will be maintained indefinitely .A little help, here is an intractable to build your own water pump using a simple DC motor .For those who are really worried about your plants .Here is a feature that might delight you .When the moisture sensor senses low moisture content ,the motor gets on and message displays on LCD screen .

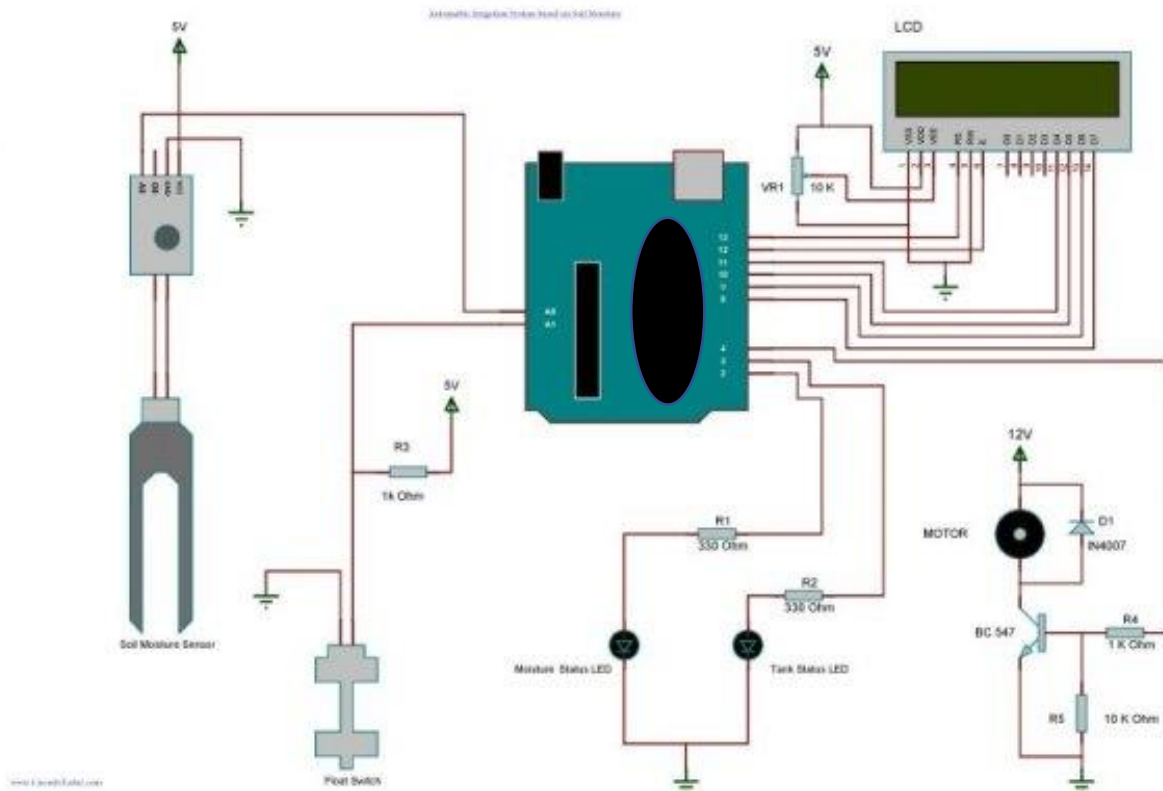


Figure :3.3Schematic Diagram of Smart plant watering system

### 3.4 Description of the Circuit Diagram :

The heart of the automatic irrigation system is the AT mega 328p microcontroller .The ATmega328 on the Arduino /Genuine Nano comes programmed with a boot loader that allows you to upload new code to it without the use of an external hardware program .Each of the 14 digital pins on the Nano can be used as an input ,Using pin Mode (),digital write (),and digital read ()functions, they operate at 5 volts ,Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default ) of 20-50 K Ohm ,A maximum of 40 mA is the that must not be exceeded on any i/o pin to avoid permanent damage to the microcontroller .



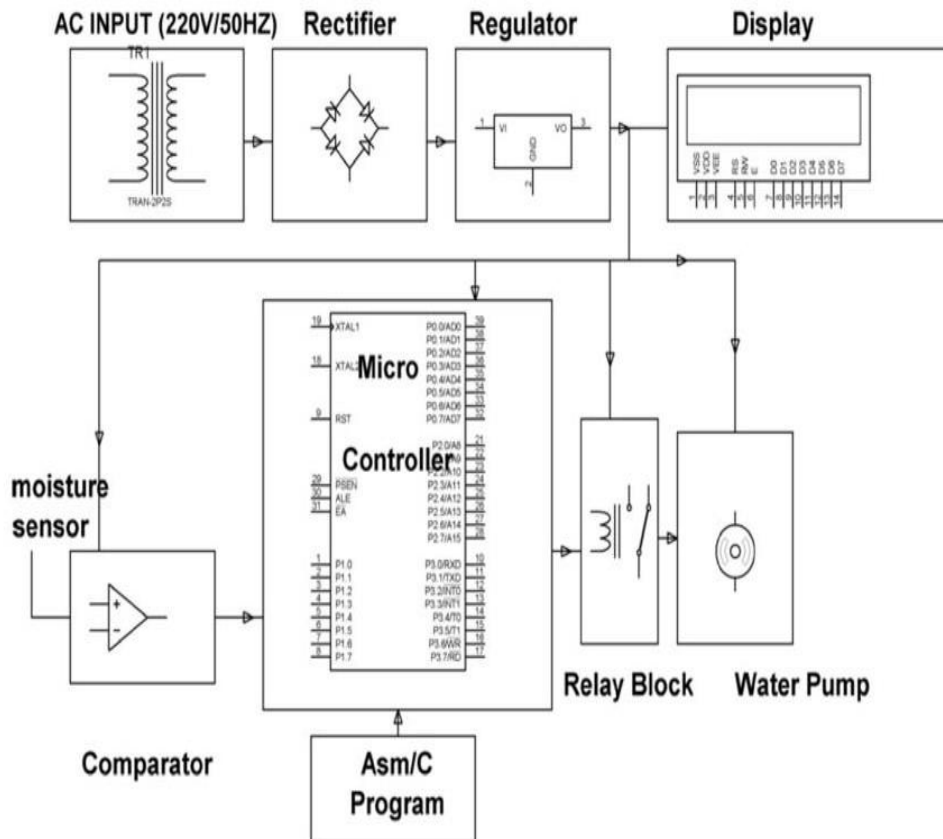


Figure :3.4 Circuit diagram of microcontroller based irrigation system

A moisture sensor is associated with each of the plurality of zones .Each such sensor is periodically interrogated by a pulse signal provided by the microcontroller via a driver or buffer circuit .This interrogation signal causes the moisture sensors to output an analog voltage which is proportional to the amount of moisture in the soil in which the sensors are embedded .The analog signal is fed to an analog /digital converter which operates to create a digital representation of the measured analog quality on the data bus leading to the microcontroller .An analog-to-digital converters suitable for use in the system is ADC 0809(4).It comprises a monolithic CMOS device with an 8-bit A/D converter ,an 8-channel multiplexer and microcontroller compatible control logic .Using successive approximation as the conversion technique ,this 8-bit A/D converter is readily interfaced to its associated microcontroller by the latched and decoded multiplexer address inputs and latches TTL tri-state outputs .

## Chapter 4

### 4.1 Components Used :

- >Arduino Nano
- >LCD Display (16X2)
- >Soil Moisture Sensor
- >Water Level Sensor
- >10 Ampere Relay board
- >12V/2.5 Ampere SMPS Adaptor
- >1:1 Male to male mistake proof wire ,Female to male mistake proof wire and female to male mistake proof wire
- >Water pumps
- >12 volt DC adapter
- >12 volt DC motor
- >C828 Transistor
- >Regulator IC 7805
- >Resistor as required

### 4.2 Project Setup:

- >Different soil retain moisture level differently so I test our soil nature and set the moisture value accordingly .
- >Upload the below code into my Arduino .
- >Fix the moisture sensor into the dry soil where your plants are planted .
- >Turn the system ON ;you will see the soil moisture in LCD .
- >In my project I have taken 50 as cut off value for dry soil but if I feel differently about my soil I can change this cut off value after testing it .

>This will be helpful in case if I feel that this automatic plant watering system is feeding plants more/less frequently .Changing the value in line 22 will customize this project as per my requirements .

>Plant has enough water level , so I check the water level sensor data correctly .

>Upload the below code into my Arduino .

>Fix the level sensor into the soil where your plants are planted .

>Turn the system ON ;you will see the water level in LCD .

>>In my project I have taken 40 as cut off value for water level in the soil but if I feel differently about my water level I can change this cut off value after testing it .

# Chapter 5

## 5.1 Arduino Nano:

The Arduino Nano is the most common and well known of many boards that use raw input and output connections to process electronic information .They are used to create electronic devices such as a thermostat controlled over the Internet or a robot that can feed your pets when you're not home .The possibilities are endless .An Arduino board is displayed below in Figure .

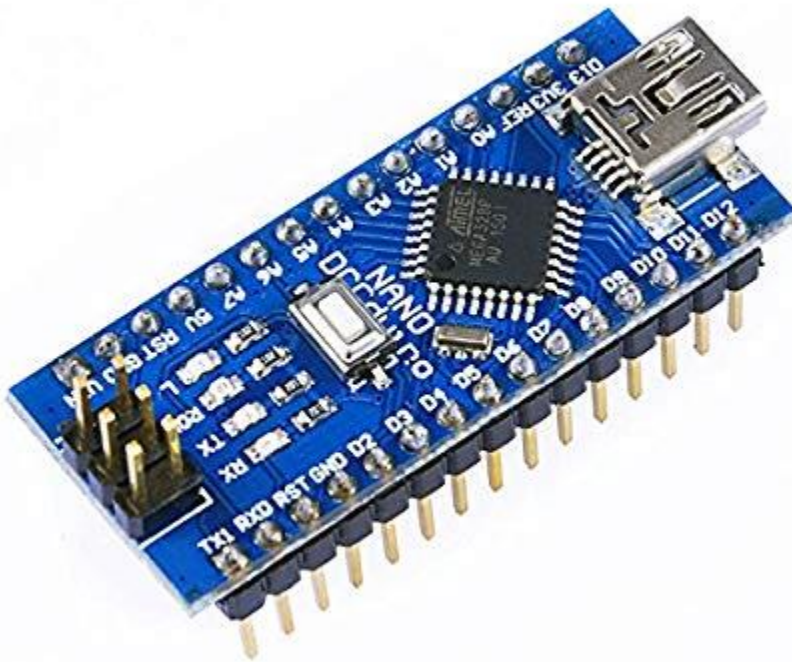


Figure :5.1 Arduino Nano board with labeled part

The Arduino gets power and communication from a computer through the USB cable .Power can also be provided through the power connector in Figure 1 once the program is loaded to the Arduino through the USB cable ,the Arduino has the program stored and running .Once programmed ,all it requires to control the thermostat is power .

### Power Requirements and Usage

The Arduino require 5 volts DC and approximately 1 amp to run properly .If not supplied with the proper power it can act strangely or not run at all .With too much power supplied it can overheat and will be destroyed .Most smart phones ,MP3 Players, and other USB devices require the same 5 volts and 1 amp that the Arduino requires .A 9-volt battery is optimal for supplying the Arduino with battery power .The Arduino will regulate the 9 volts down to 5 volts and work well .In the thermostat example in Figure 2,the power connector is used instead of a USB cable .This is because the program is already stored and running on the Arduino .Optimally ,the power connector for this example is providing 9 volts from a small transformer plugged into a nearby outlet so that the user doesn't have to replace batteries .

## 5.2 Programming :

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Nano from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials. Now open the Arduino IDE and choose the correct board by selecting Tools>Board>Arduino/Genuine Nano, and choose the correct Port by selecting Tools>Port . The ATmega328 on the Arduino Nano comes pre burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). To get it started with Arduino Nano board and blink the built-in LED ,load the example code by selecting Files>Example>Basics>Blink .

Once the example code (also shown below) is loaded into your IDE ,click on the “upload” button given on the top bar . You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

> On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

>On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

### **5.3 Warning :**

The Arduino/Genuine Nano has a resettable poly fuse that protects your computers USB ports from short and over current .Although most computers their own internal protection ,the fuse provides an extra layer of protection .If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until short or overhead is removed .

### **5.4 Difference with other boards :**

The Nano differs from all preceding boards in that do not use the FTDI USB to serial driver chip .Instated ,it futures the Atmega16U2(Atmega8U2 up to version R2) programmed as a USB to serial converter .

### **5.5 Power :**

The Arduino Nano can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

For a power source ,you have the option of using the USB or a DC jack . Now it's time to answer the following :''If I connect both a DC adapter and the USB , which will be the power source ?''

The 5V regulator is the NCP1117ST50T3G and the VIN of this regulator is connected via DC jack input through the M7 diode ,the SMD version of the famous .This diode provides reverse-polarity protection .

The output of the 5V regulator is connected to the rest of 5V directly from the power header 5V pin .

Another source of 5V is USBVCC which is connected to the drain of an FDN340P ,a P-channel MOSFET ,and the source is connected to the 5V net. The gate of transistor is connected to the output of an LMV358 op-amp used as a comparator .The comparison is between 3V3 and Vin/2 .When Vin/2 is large ,this will produce a high output from the comparator and the P-channel

MOSFET is off .If there is no VIN applied ,the V+ of the comparator is pulled down to GND and Vout is low ,such that the transistor is on and the USBVCC is connected to 5V.

The LP2985-33DBVR is the 3V3 regulator .Both the 3V3 and 5V regulators are LDO(Low Dropout),which means that they can regulate voltage even if the input voltage is close to the output voltage .This is an improvement over older linear regulators ,such as the 7805 .

The last thing I'll talk about is the power protection that is provided in Arduino NANO .

As mentioned above, VIN from a jack is protected from reverse polarity by using a serial M7diode in the input . Be aware that the VIN pin in the power header is not protected .This is because it is connected after the M7 diode .Personally ,I don't know why they decided to do that when they could connect it before the diode to provide the same protection .

When you use USB as a power source ,and to provide protection for your USB port, there is a PTC (positive temperature coefficient) fuse (MF-MSMF050-2) in series with the USBVCC .This provides protection from over current ,500mA .When an over current limit is reached, the PTC resistance increases a lot .Resistance decreases after the over current is removed .

## 5.6 The power pins are follows :

>VIN the input voltage Arduino/Genuine board when it's using an external power source (opposed to 5V from USB connection or other regulated power sources).We can supplied voltage through this pin or if suppling voltage via the power jack ,access if through this pin .

>5V this pin outputs a related 5V from the regulator on the board ,the can be supplied with power either from the DC power jack (7 -12 v), the USB connector (5V) or the VIN of the board (7-12v),supplying via the 5v or 3.3 v pins by passed the regulator ,and can damage your board ,we don't advice it .

>3v3. A3.3v supply generated by the on-board regulator .Maximum current draw is 50mA .

>GND. Ground pin ,

>IOREF, This pin on the Arduino/Genuine boards provides the voltage reference with the microcontrollers operates , A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the output to work with 5v or 3.3v

## 5.7 Memory :

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library). The ATmega328 microcontroller is the MCU used in Arduino NANO R3 as a main controller . ATmega328 is an

MCU from AVR family ;it is an 8-bit device ,which means that its data-bus architecture and internal registers are designed to handle 8 parallel data signals .

ATmega328 has three types of memory :

>Flash memory :32KB nonvolatile memory .This is used for storing application ,which explains why you don't need to upload your application every time you unplug Arduino from its power source .

>SRAM memory: 2KB volatile memory .This is used for storing variables used by the application while it's running .

>EEPROM memory :1KB nonvolatile memory .This can be used to store data that must be available even after the board is powered down and the powered up again .

## **5.8 Communication :**

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

## **5.9 Automatic (Software) Reset :**

Rather than requiring a physical press of the reset button before and upload ,the Arduino/Genuine Nano board is design in way that allows it to be reset by Software running on a connected computer .One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 Nano farad capacitor .When this line is asserted (taken law) ,the reset line drops long enough to reset the chip .The Arduino Software (IDE) uses this capability to allow you to upload code simply pressing the upload button in the interface toolbar ,this means that the boot loader can have shorter timeout as the lowering of DTR can be well-coordinated with the start of the upload .



This setup has other implications, when the Nano is connected to either a computer running Mac OS X or Linux ,it resets each time a connection is made to it from software (via USB).For the following half-second of so, the boot loader is running on the Nano, while is it is programmed to ignore malformed data (I, e anything beside and upload a new code ), it will intercept the first few bytes of data sent to the board after a connection is open , make sure that the software with which it communicates waits a second after opening the connection and before sending this data ,the Nano board contains trace that can be cut to disable the auto-reset ,this pads on either side of the trace can be soldered together to re-enable it, It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resister from 5 volt to the reset line ;see this forum thread for details .

## 5.10 Physical Characteristics :

The maximum length and width of the Nano PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

## 5.11 Schematic & Reference Design :

EAGLE files: arduino-Nano-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)  
Schematic: arduino-Nano-Rev3-schematic.pdf Note: The Arduino reference design can use an Atmega8, 168, or 328, Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

## 5.12 Input and Output :

Each of the 14 digital pins on the Nano can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 k Ohms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference() function. Additionally, some pins have specialized functionality:

□ TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

□ AREF. Reference voltage for the analog inputs. Used with analog Reference(). □ Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the Atmega8, 168, and 328 is identical.

## 5.13 USB Overcurrent Protection :

The Arduino Nano has a resettable poly fuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## 5.14 Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	

# Chapter 6

## 6.1 Arduino Details :

The Arduino Nano is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Nano differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. Revision 2 of the Nano board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode. Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

## 6.2 Power (USB/Barrel Jack):

The Arduino Nano can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.



Figure:6.2 AC-to-DC adapter

Every Arduino board needs a way to be connected to a power source .The Arduino NANO can be powered from USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack . In the picture above the USB connection is labeled

- (1) and the barrel jack is labeled
- (2) The USB connection is also how you will load code onto your Arduino board

### **6.3 Pins (5V, 3.3V, GND ,Analog ,Digital, PWM, AREF ):**

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire .They usually have black plastic ,headers ,that allow you to just plug a wire right into the board .The Arduino has several different kinds of pins ,each of which is labeled on the board and used for different functions .

GND (3) :Short for ,”Ground” there are several GND pins on the Arduino ,any of which can be used ground your circuit .

5V(4) & 3.3 V(5) :As you might guess ,the 5V pin supplied 5 volts of power ,and the 3.3V pin supplies 3.3 volts of power .Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts .

Analog (6): The area of pins under the, Analog in label (A0 through A5 on the NANO) are Analog in pins .These pins can read the signal from an analog sensor (like a temperature sensor ) and convert it into a digital value that we can read .

Digital (7): Across from the analog pins are digital pins (0 through 13 on the NANO). These pins can be used for both digital input (like telling if a button is pushed ) and digital output (like powering an LED).

PWM(8) :You may have noticed the tiled (-) next to some of digital pins (3,5,6,9,10 and 11on the NANO).These pins act as normal digital pins ,but can also be used for something called pulse-width modulation (PWM).We have a tutorial on PWM ,but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).

AREF (9):Stands for Analog reference .Most of the time you can leave this pin alone .It is sometimes used to set an external reference voltage (between 0 and 5 volts ) as the upper limit for the analog input pins .

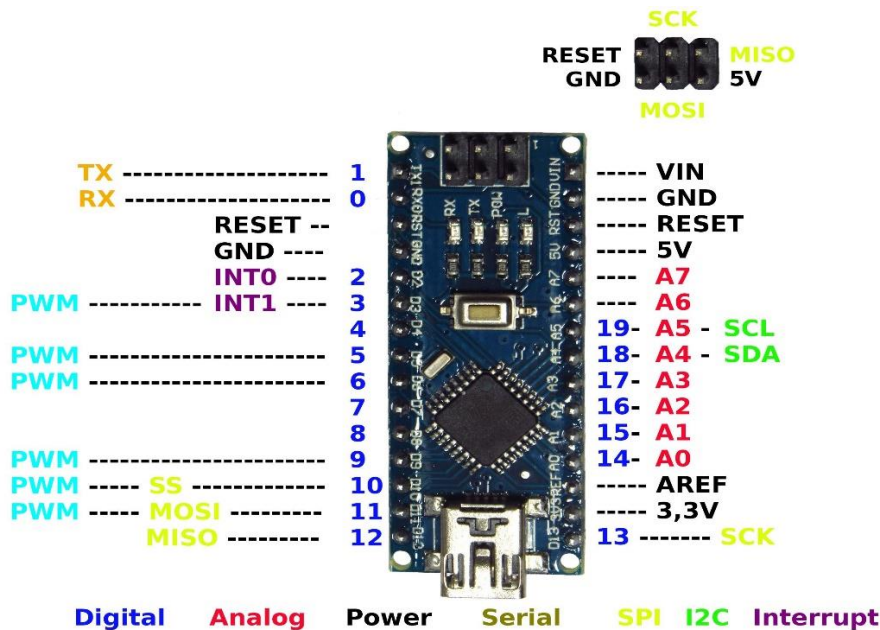


Figure: 6.3 Pins

## 6.4 Reset Button :

The reset button does pretty much the same as unplugging the board and plugging it back in .It restarts your program from the beginning .The same thing happens when your program the board- the USB interface presses the reset button for you .That then enters the boot loader for a second or two so it can try and program it .When you reset the board the LED on pin 13 should flash a couple

of times while it's in the boot loader before it runs whatever program you have programmed in .If that LED doesn't flash when you press the reset button then there is a serious fault with your board which will take further diagnostic .

>If you have anything eles plugged the Arduino then unplug it .

>Try propering the Arduino from different power supplies

>If you have another Arduino or an AVR programmer try re-flashing the bootloader .

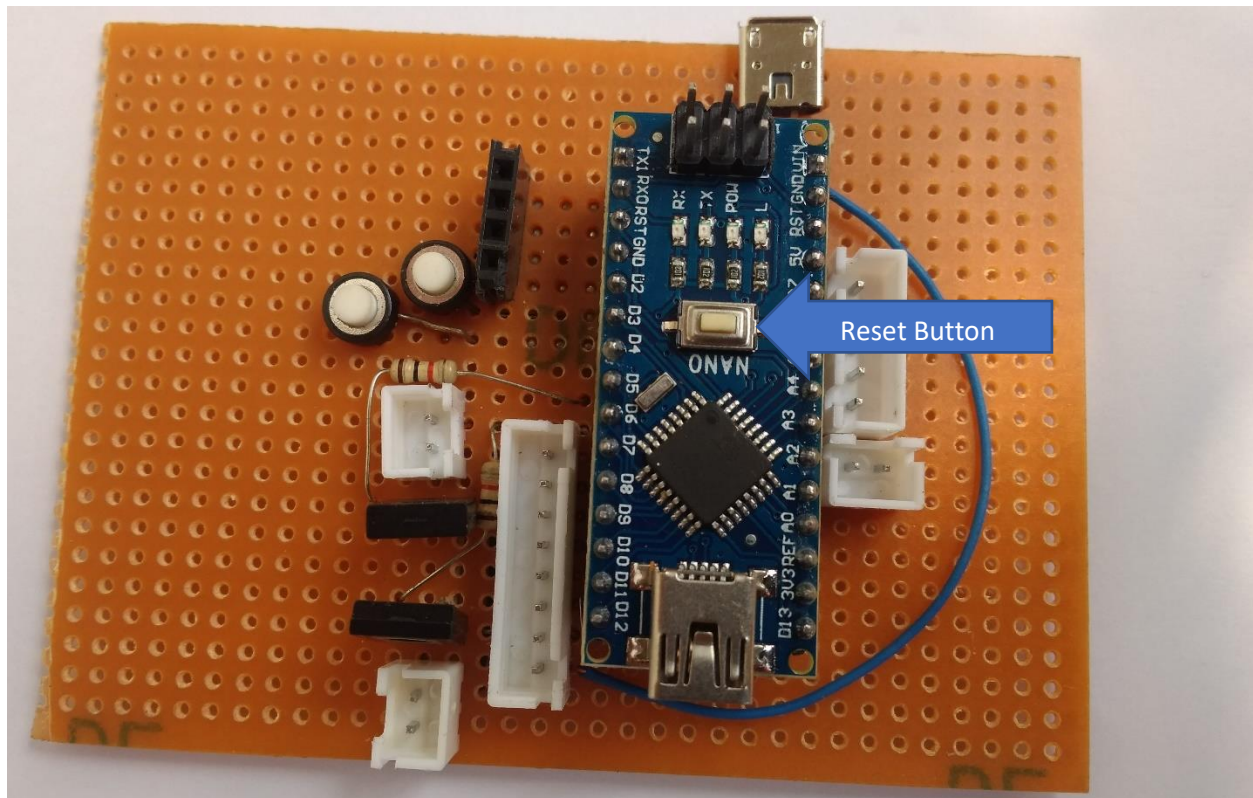


Figure:6.4 Reset Button

## 6.5 Power LED Indicator :

It was just to stop the light illuminating when the Arduino was being used in a dark environment but the best solution is the simply cover it up. Although perhaps it would be nice if it could be controlled programmatically so, for example, I could turn on the LED during the day and turn it off at night.

After thinking about it, if your Arduino board was going to be used in a final solution, mounting it in an enclosure would be a good idea: it would protect your board and hide the power LED where you could use an externally mounted LED as a power or status indicator.

Just beneath and to the right of the world “NANO” on your circuit board ,there’s a tiny LED next to the world “ON” (11).This LED should light up whenever you plug your Arduino into a power source .If this light doesn’t turn on, there’s a good change something is wrong .Time to re-check your circuit .

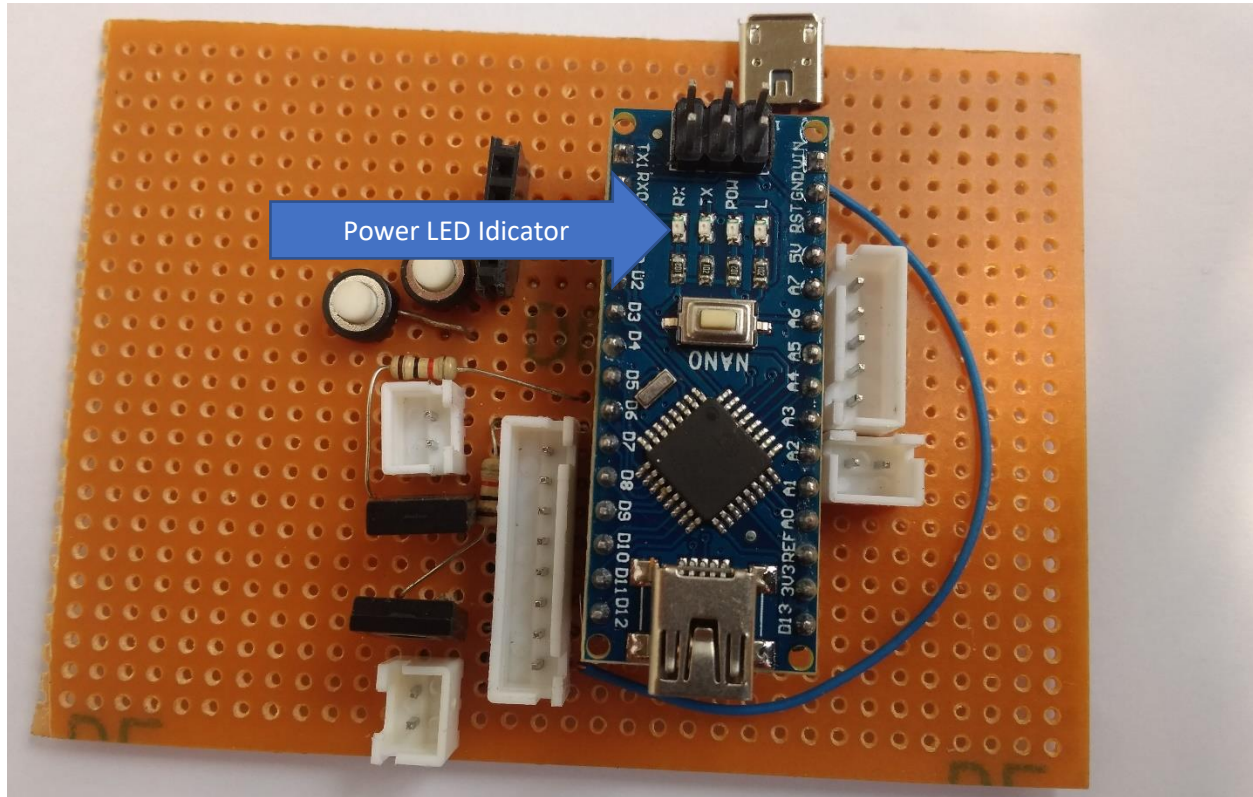


Figure:6.5 Power LED Indicator

## 6.6 TX RX LEDs:

This seems incomplete as it does not address how to implement the said interrupts .When I can see using ISR (UART\_RX\_vect) and ISR (UART\_RX\_vect),except they are only double buffered (small) .So they would only trip quickly and once , on the completion of the initial byte .Or is that it TX’s Um buffer is filled between ,ready for next and completely empty .After UART\_UDRE\_vect but before UART\_UDRE\_vect .This would not be the case for RX .Also the interrupts (); is needed to enable the ISR () vectors . The arrow points to the most obvious integrated circuit or chip in you pic, there is another square one to the left of the TX and RX markings. There are lots of different types of ICs -- most are black and come in different shapes and sizes, different number of legs, but most have some form of writing on the back that Google can use to locate the data sheet, or general help. Have a quick look on Wikipedia for a more authoritative description.



Figure: 6.6 TX RX LEDs

## 6.7 Main IC :

IC stands for integrated circuit, which is a generic term encompassing any class of devices that has multiple electronic components (transistors, and possibly capacitors, resistors, diodes, and more) encased and interconnected within. The opposite of an integrated circuit is a discrete component, which is a single electronic component in its own package that relies on wiring (or a circuit board) to connect to other components.

Taking a look at the Arduino Nano R3 below (click for larger image), the long rectangular black thing is the primary IC. You can see the ATMEL logo on it, followed by a short series of numbers (a date code if I'm not mistaken). Below is the chip model itself; in this case, "ATMEGA328P-PU". With respect to the Arduino IDE, it's sufficient to know that the chip is an ATmega328P, which is sometimes incorrectly called ATmega328 in the Arduino world.

The black thing with all the metal legs is an IC, or Intergrated Circuit (13) .Think of it as the brains of our Arduino .The main IC on the Arduino is slightly different from board type of board type ,but is usually from the ATmega line of IC's from the ATMEL company .This can be important ,as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software .This can usually be found in writing on the top side of the IC . If you want to know more about the difference between various IC's reading the data sheets is often a good idea .



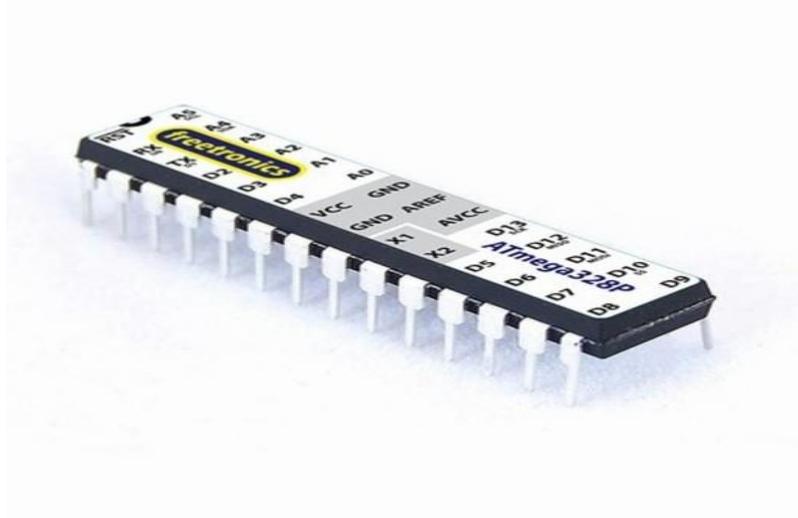


Figure:6.7 Main IC

## 6.8 Voltage Regulator :

All most all the projects I do are based on 5V Voltage Supply ,and to regular the voltage supply requires a voltage regulator .The most commonly used voltage regulator is a 7805 IC which operates at 1A .In this instructable I'm going to show you how to construct a 5V voltage regulator ,this can be used to power an atmega328 if you are replicating an Arduino or building a portable power supply .

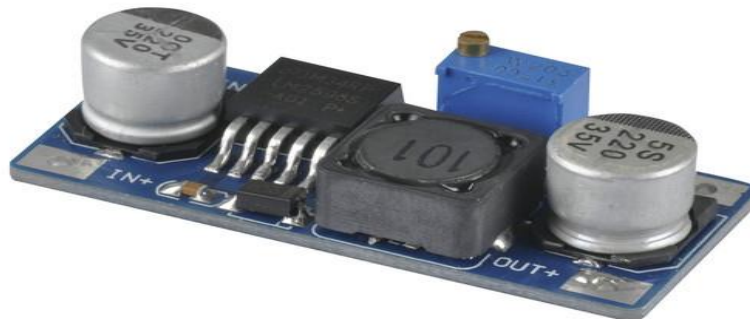


Figure: 6.8 Voltage Regulator

The voltage regulator (14) is not actually something you can (or should) interact with on the Arduino .But it is potentially useful to know that it is there and what it's for .The voltage regulator does exactly what it says it controls the amount of voltage that is let into the Arduino board .Think of it as a kind of gatekeeper ,it will turn away an extra voltage that is let into the Arduino board .Think of it as a kind of gatekeeper ,it will turn away an extra voltage that might harm the circuit .Of course ,it has its limits ,so don't hook up your Arduino to anything greater than 20 volts .As a Arduino uses the same 7805 as the voltage regulator to power up an atmega328 ,or if you happen to burn the Arduino regulator you can buy one off eBay and replace the on board one .This project is also the stater for a virtual USB Project .

# Chapter 7

## 7.1 Description of ATmega328 Microcontroller :

ATmega328 is basically an Advanced Virtual RISC (AVR) microcontroller .It supports the data up to eight (8) bits .ATmega328 has 32KB internal built-in memory .This microcontroller has a lot of other characteristic .You should also have a look at Introduction to PIC16F877a and the compare functions of these two Microcontrollers .

ATmega328 has 1KB Electrically Erasable Programmable Read Only Memory (EEPROM) .This property shows if the electric supplied to the micro-controller is removed even then it can store data and can provide results after providing it with the electric supply .Moreover ,ATmega328 has 2KB Static Random Access Memory (SRAM) .Other characteristic will be explained later .ATmega328 has several different features which make it the most popular device in today's market .These features consist of advanced RISC architecure ,good performance ,low power consumption ,real timer counter having separate oscillator ,6 PWM pins ,programmable Serial USART , programming lock for software security ,throughput up to 20 MIPS etc .ATmega328 is mostly used in Arduino .



Figure :7.1 Microcontroller ATmega328P

>ATmega328 is an 8-bit and 28 Pins AVR Microcontroller ,manufactured by Microchip ,follows RISC Architecture and has a flash type program memory of 32KB .

>It has an EEPROM memory of 1KB and its SRAM memory is of 2KB .

>It has 8 Pin for ADC operations ,which all combines to form PortA (PA0-PA7) .

>It also has 3 built-in Timers ,two of them are 8 Bit timers while the third one is 16 Bit Timer .

>We must have heard of Arduino Nano ,Nano is based on atmega328 Microcontroller .It's Nano's heart .

>It operates ranging from 3.3V to 5.5V but normally we use 5V as a standard .

>Its excellent features include the cost efficiency ,low power dissipation programming lock for security purposes ,real timer counter with separate oscillator .

>It's normally used in Embedded Systems applications .You should have a look at these Real Life Examples of Embedded System ,we can design all of them using the Microcontroller .

## 7.2 Block Diagram of ATmega328p :

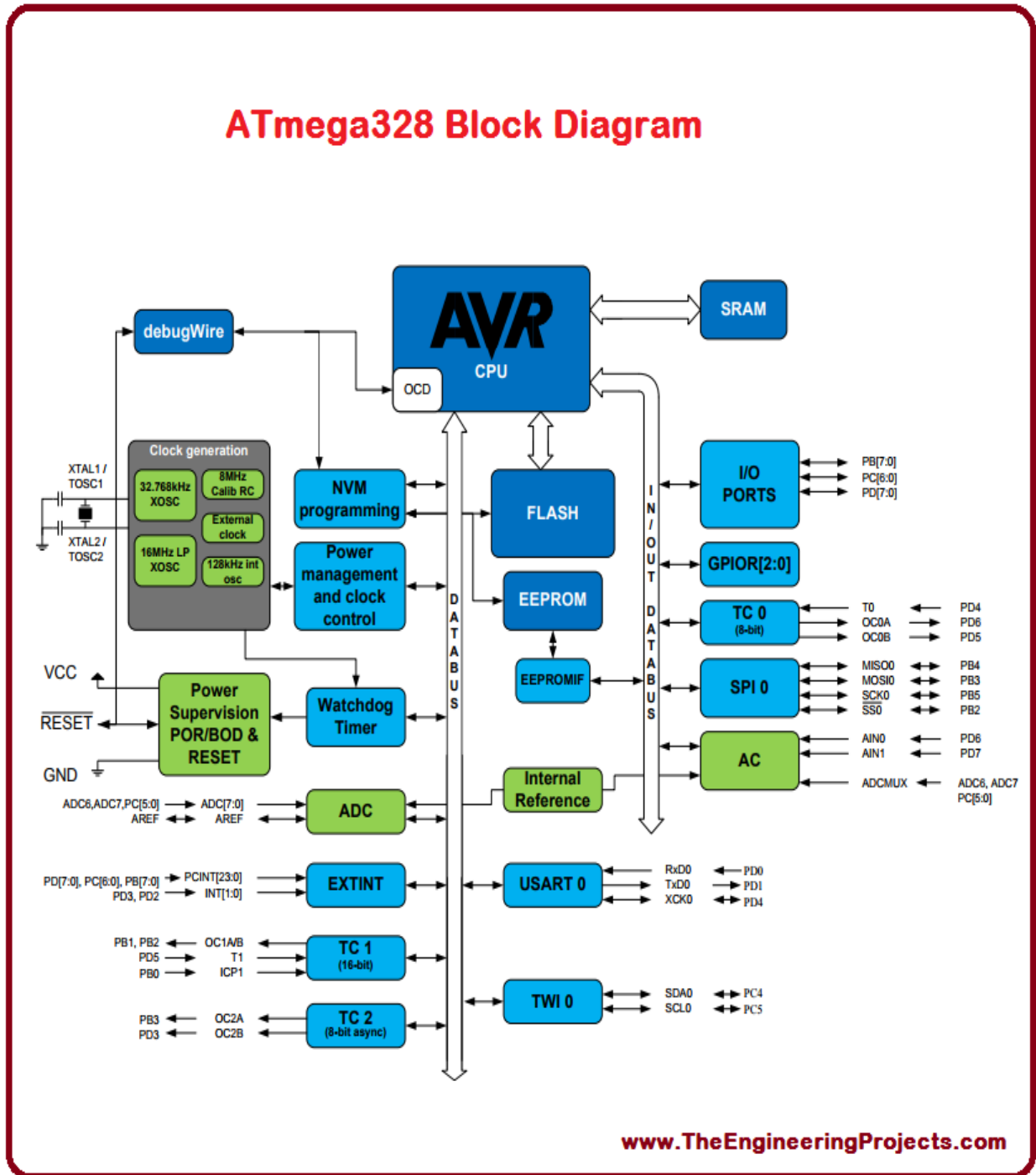


Figure :7.2 Block Diagram of ATmega328p

# Chapter 8

## 8.1 LCD Panel :

A Liquid crystal display (LCD) is a flat display that uses the light modulating properties of liquid display .They are common is consumer device , such as video platers , gaming device , clocks , telephones , computers ,calculator etc .A16X2 LCD module has been used as a display that means it can display 16 character per line and there are 2 lines .LCDs are preferred over LED and seven segment displays because they are economical and easily programmable .



Figure :8.1 LCD Display

## 8.2 Water Level Sensor :

This sensor is very important parts of our project .It takes its input from the soil . If there is less water level in the soil ,the sensor would give an Analog input to the ADC inside the MSP430 (ADC explained later in report ).Basically water level sensor indicate the water level in the soil . Basic level sensors can be used to identify the point at which a liquid falls below a minimum or rises above a maximum level. Some types use a magnetic float, which rise and fall with the liquid in the container. Once the liquid, and by extension, the magnet, reach a certain level, a reed magnetic switch is activated .The ADC (Analog to Digital Converter ) which is one of the most important inbuilt features of the MSP , samples this Analog input value given by the sensor registers (ADC10MEM) .

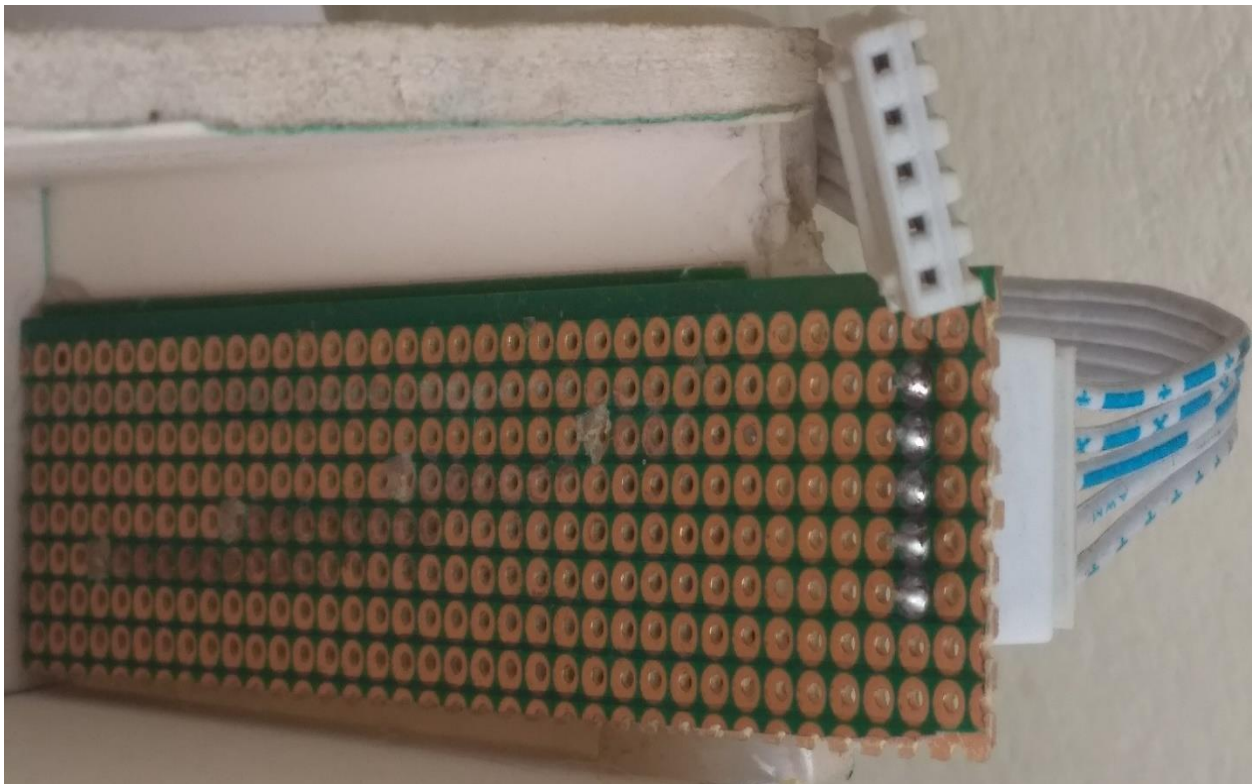


Figure :8.3 Water Level Sensors

### 8.3 Moisture Sensor:

The moisture sensor consists of two leads that are used to measure volume of water content in soil .These leads allow the current to pass through the soil and in return calculates the resistance value to measure the moisture level .If there is more water in soil then the soil will conduct more electricity ,means less resistance value along with high level of moisture .In the same manner if there is less water in soil then soil will conduct less electricity ,means high resistance value along with low level of moisture .This sensor is one of the main parts of our project .It takes its input from the soil . If there is less moisture in the soil ,the sensor would give an Analog input to the ADC inside the MSP430 (ADC explained later in report ).Soil moisture sensors measure the volumetric water content in soil .Since the direct gravimetric dielectric constant of a certain volume element around the sensor is obtained by measuring the operating frequency of an oscillation circuit .The ADC (Analog to Digital Converter ) which is one of the most important inbuilt features of the MSP , samples this Analog input value given by the sensor registers (ADC10MEM) .

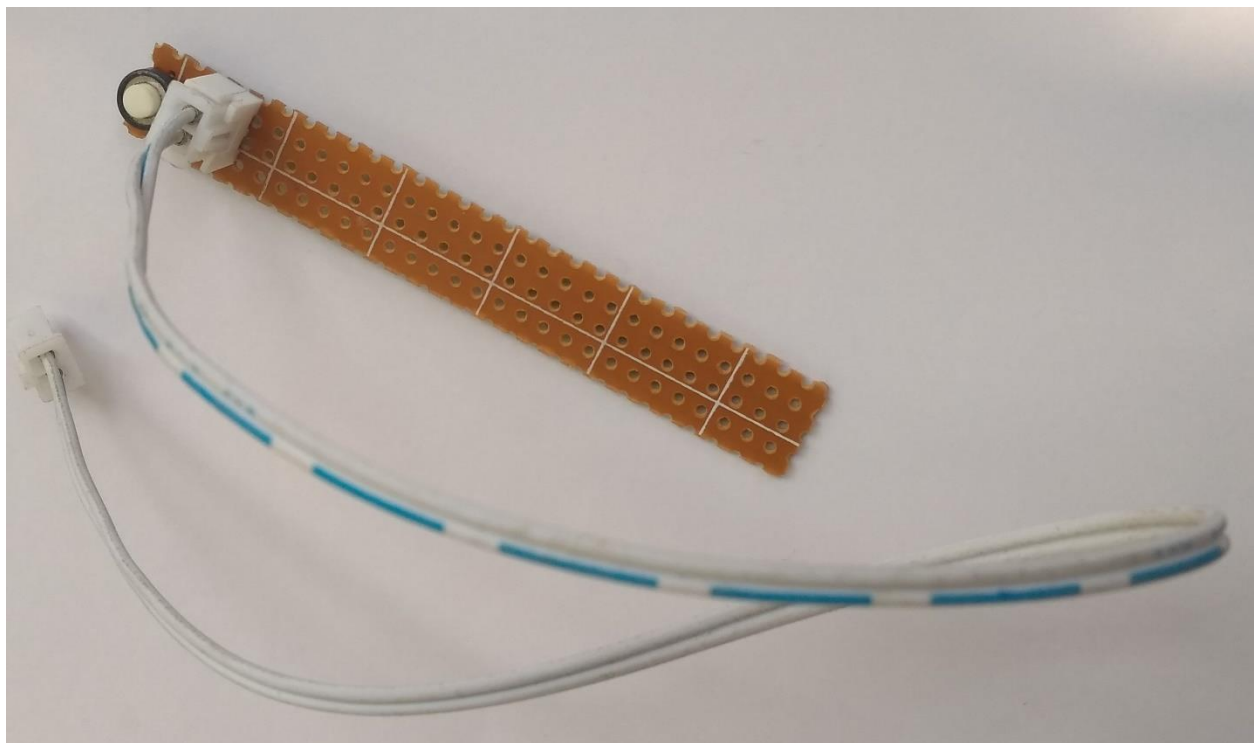


Figure :8.3 Moisture sensors



## 8.4 Water Pump :

Use DC motor to make water pump . DC motor has two leads one is positive and another one is negative .If we connect them directly to the Arduino board then it will damage the board .To overcome this problem NPN transistor is used to control the switching activity of the motor according to the code .Arduino pin 13 (named as WATERPUMP in code ) is used to turn on and off the transistor .According to the code to control the speed of the motor we need to enter the value between 0 and 255 in the Serial Monitor .I used 200 value for the speed of the motor . A small submergible pump a driver is to provide enough current for the pump ,my application needs spray distance about one meter ,so this pump is enough .But if you need to make a system that needs a large spray range ,you may need large pumps , or even a pressurized device to make the projectile even farther ,such as the watering system in a rice plant .



Figure :8.4Water Pump

## **Chapter 9**

### **Conclusion :**

The system is ideal for the expensive, rare and luxurious plants which are basically planted in pots. It can also be applied to gardening in the apartment's or any space. The whole circuit consumes a low power . The main advantage of the system is that it only turns the water pump on when its needed and instantly turns off when the water is sufficient. Besides, it can detect any leakage in the water supply and notify the user. In this way, it prevents wastage of water and ensures the appliance of water optimally. In the future, we have a plan to consider more criteria in determining perfect timing for watering and apply the whole system for large firm land. As the system is optimized for perfect timing of watering, it may help the plants to remain healthy and grow perfectly. Thus the system would not only keep the plants alive, but also facilitate to grow up in a healthy way.

## CHAPTER 10

### **Recommendation:**

1. Our project is based on automation technology and we tried our best to implement all the possible mechanism for that .
- 2.A helpful add on feature could be the addition of Bluetooth technology along with it for the better automatic functioning .
3. Useful Irrigation in Fields .
4. Useful Irrigation In Gardens , Parks .
- 5.This water supply system is very efficient for Paddy ,Rice Fields .
- 6.It is effective in Pisciculture also .
- 7.Use it as a home automation controller ,by adding a few more 240 volts relays .
- 8.Remotely perform jobs .
- 9.Use a float switch in a tank ,so that the system automatically shuts the pump down once the reservoir is full .
- 10.Use it in conjunction with a solar panel , so that the entire system is eco-friendly

## CHAPTER 11

### REFERENCE

- [1] Ganesh, Mr Sundar. "Efficient Automatic Plant Irrigation System using ATMEGA Microcontroller." International Journal of Emerging Trends in Electrical and Electronics (IJETEE–ISSN: 2320-9569) Vol 7 (2013).
- [2] Darshna, S., T. Sangavi, Sheena Mohan, A. Soundharya, and Sukanya Desikan. "Smart Irrigation System." IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), Volume 10, Issue 3, Ver. II, PP 32-36, May - Jun.2015.
- [3] Chikankar, Pravina B., Deepak Mehetre, and Soumitra Das. "An automatic irrigation system using ZigBee in wireless sensor network." In Pervasive Computing (ICPC), 2015 International Conference on, pp. 1-5. IEEE, 2015.
- [4] Laxmi Shabadi , Nandin Patil ,Nikita M, Shruit J, Smitha .P & Swait C ,“Irrigation Control System Using Andriod And GSM for Efficient Use of Water and Power”, International Journal of Advanced Research in Computer Science and Software Engineering ,Volume 4,Issue 7 ,July 2014
- [5] Abbas, Ahmed Hussein, et al. "Smart watering system for gardens using wireless sensor networks." Engineering and Technology (ICET), 2014 International Conference on. IEEE, 2014.
- [6] Yunseop (James) Kim , Member , IEEE , Rebot G. Evans ,And Williams M. Iversen “Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network ”, IEEE TRANSACTION ON INSTRUMENTATION AND MEASUREMENT ,Volume 57, Number 7, JULY 2008 .
- [7] Sakaguchi, Yoshiyuki, Torahiro Nakamura, Yuki Sugisaka, and Hiromi T. Tanaka. "Water-volume measurement of plants by using itplanter." In SICE Annual Conference (SICE), 2011 Proceedings of, pp. 2798-2800. IEEE, 2011.

[8] Venkata Naga Rohit Gunturi ,“Micro Controller Based Automatic Plant Irrigation System ” , International Journal of Advancements in Research & Technology , Volume 2, Issue 4, April-2013

[9] Gunturi, Venkata Naga Rohit. "Micro Controller Based Automatic Plant Irrigation System." International Journal of Advancements in Research & Technology 2, no. 4 (2013): 194-198.

[10] Bathan, Jazelle Paula M., Ma Carla Angelica R. Belen, Paula Jianelli M. Lao, Jasper C. Tiu, and Enrique M. Manzano. "Automated Irrigation System Using Thermoelectric Generator As Soil Moisture Detector." (2013).

[11] S. Harishankar , R. Sathish Kumar , Sudharsan K>P, U. Vignesh and T. Viveknath , “Solar Powered Smart Irrigation System ”, Advance in Electronic and Electric Engnieering ,Volume 4 ,Number 4 (2014) ,pp. 341-346

[12] Pavithra, D. S., and M. S. Srinath. "GSM based Automatic Irrigation Control System for Efficient Use of Resources and Crop Planning by Using an Android Mobile." IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE) e-ISSN (2014): 2278-1684 .

## Appendix

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,10,9,8,7);
int ricePump=5;
int potatoPump=6;
int riceBtn=2;
int potatoBtn=3;

int level;
int percent;

int rps=1;
int pps=1;
void setup() {
  Â pinMode(ricePump,OUTPUT);
  Â pinMode(potatoPump,OUTPUT);
  Â pinMode(riceBtn,INPUT_PULLUP);
  Â pinMode(potatoBtn,INPUT_PULLUP);
  Â lcd.begin(16, 2);
  Â Serial.begin(9600);
}
void loop() {
  Â if(analogRead(A7)>=100) level=99;
  Â else if(analogRead(A6)>=100) level=75;
  Â else if(analogRead(A5)>=100) level=50;
  Â else if(analogRead(A4)>=100) level=25;
  Â else level=0;
  Â
  Â percent=map(analogRead(A2),0,250,0,99);
  Â if(percent>=99) percent=99;
  Â else if(percent<=0) percent=0;

if(digitalRead(riceBtn)==1 && rps==1)
{
  Â if(level<=50) digitalWrite(ricePump,1);
  Â else if(level==99) digitalWrite(ricePump,0);
}
else if(digitalRead(riceBtn)==0 && rps==1)
```

```

{
  ^ if(digitalRead(ricePump)==1) digitalWrite(ricePump,0);
  ^ else digitalWrite(ricePump,1);
  ^ rps=0;
  ^ delay(1000);
}

if(digitalRead(potatoBtn)==1 && pps==1)
{
  ^ if(percent<=25) digitalWrite(potatoPump,1);
  ^ else if(percent>=60) digitalWrite(potatoPump,0);
}
else if(digitalRead(potatoBtn)==0 && pps==1)
{
  ^ if(digitalRead(potatoPump)==1) digitalWrite(potatoPump,0);
  ^ else digitalWrite(potatoPump,1);
  ^ pps=0;
  ^ delay(1000);
}
^ Serial.print("Water Level=");
^ Serial.print(level);
^ Serial.println("%");
^ Serial.print("Moisture=");
^ Serial.print(percent);
^ Serial.println("%");
^ Serial.print("Rice Pump:");
^ Serial.println(digitalRead(ricePump));
^ Serial.print("Potato Pump=");
^ Serial.println(digitalRead(potatoPump));
^ Serial.println(".....");
^
^ lcd.setCursor(0, 0);
^ lcd.print("Water Level: ");
^ lcd.setCursor(13, 0);
^ lcd.print(level);
^ lcd.setCursor(0, 1);
^ lcd.print("Moisture: ");
^ lcd.setCursor(10, 1);
^ lcd.print(percent);
^ delay(1000);
}

```