DEVI AHILYA VISHWAVIDYALAYA, INDORE
( Formerly University of Indore ) , NAAC A+ Grade
State University of Madhya Pradesh, India

# A

# LAB REPORT

# On

# Designing our own Arduino Board on Zero PCB

**M.Tech IoT (2 - year)**

**Amit Singh**

**Ankit Goyal**

**Srija Loya**

**Vaishnavi Dehariya**

**Dipti Sharma**

**Submitted**

**To**

**School of Instrumentation**

**Devi Ahilya Vishwavidhyalaya**

# Contents

**Chapter – 1**

# Chapter - 1

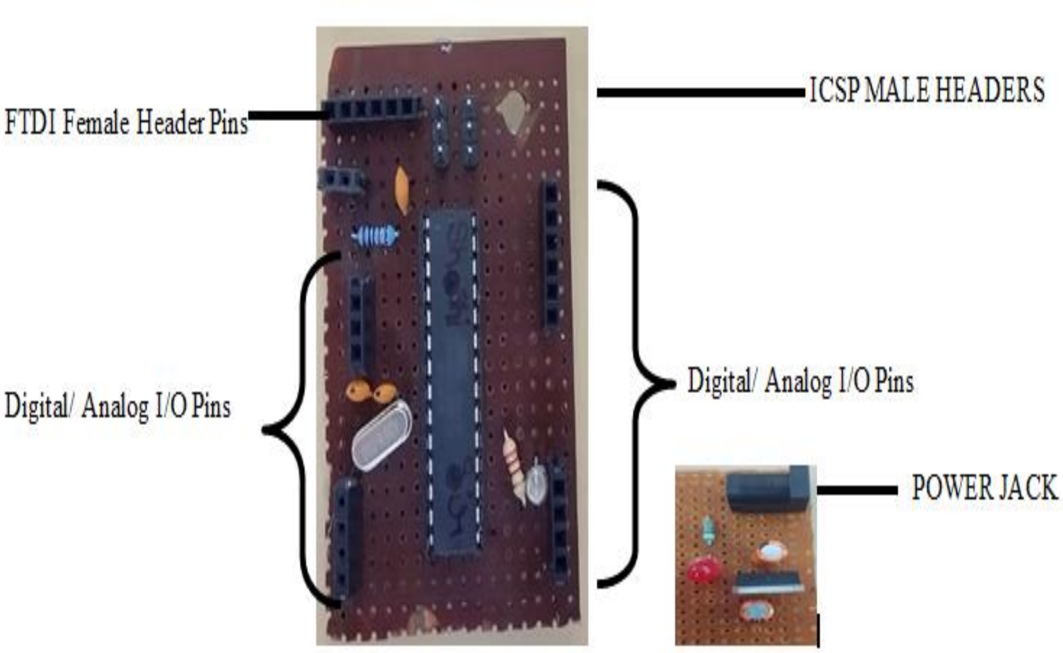
## Getting Started

## 1.1 Introduction

A microcontroller is a small and low-cost microcomputer, which is designed to perform the specific tasks of embedded systems like displaying microwave's information, receiving remote signals, etc. The general microcontroller consists of the processor, the memory (RAM, ROM, and EPROM), Serial ports, peripherals (timers, counters), etc. Microcontrollers are divided into various categories based on memory, architecture, bits and instruction sets. When it comes to bit wise categorization of microcontrollers, they are 8-bit, 16-bit and 32-bit microcontrollers. According to memory wise categorization, they are external memory, and embedded memory microcontroller. Based on instruction set, they are CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer). We are making an arduino, which is an 8-bit microcontroller. Most of the arduinos are considered as CISC. Arduino is neither a microcontroller nor an embedded system. Arduino uses a microcontroller, and it can be used to create an embedded system. The Arduino UNO uses Harvard Architecture. This architecture's main characteristic is that it uses two separate memory units, one for storing program instructions and one for storing program data. Both memory units in the Harvard architecture are accessed by the CPU using different communication buses. We made a microcontroller, based on Arduino UNO by studying about all the components in detail.

### 1.2 The Equipment

The arduino board uses FTDI to upload the program in the ATMega328P- PU microcontroller which is used for uploading the program. This FTDI is connected to the computer to upload the program from Arduino UNO board. The female header jack is used to supply the power to the Arduino board. The LM 7805 voltage regulator is used to regulate the voltage supply till 5V. So, if there is a supply of the voltage level more than 5V, then this LM 7805 voltage regulator regulates the voltage to 5V. FTDI chip is one of the most popular USB to TTL converter IC. They are used to interface devices like Arduino with the laptop through a USB cable. In simple words, an FTDI chip makes it super easy to convert a USB signal to the UART signal understood by the microcontrollers**.** FTDI majorly focuses on USB technology, and the USB to TTL interface is their most popular product. These are available as cable as well as a module (like a chip). These products are widely used as an interface in microcontroller development boards like Arduino, ESP-01s, etc. because they require a USB interface. The FTDI interface is the easiest solution to connect the devices with the TTL level interface to USB since it does not require any additional software installation in advance. Thus they act as simple plug-and-play devices and are very handy. The major application of the FTDI interface is in general TTL serial communication. The following figure shows the external connections of the arduino board.



### 1.2.1 FTDI Female Header Pins:

The FTDI Female header pins are used to connect the FTDI adapter module to the microcontroller. FTDI cable is a popular interface used to connect Universal Asynchronous Receiver-Transmitter (UART) devices to PC. These cables fall under the category of USB to TTL serial UART converter cables which are usually available with FT232R IC. They are easy to handle and establish fast connections with TTL and USB. The USB interfacing with FTDI cables are used for, the transfer of low bandwidth USB data through computer and external devices. FTDI adapter module is a complete package in which the FTDI chip is integrated with connectors, voltage regulators, Tx/Rx, and other breakout points. The module thus falls under the category of UART board and is mostly used for TTL serial communication. This FTDI adapter module is used in the devices where there is no in-built USB interface.

### 1.2.2 Digital/ Analog I/O pins:

These pins are internally connected to the ATMega328P-PU microcontroller. Where, the analog and digital output pins of ATmega 328P-PU are interfaced with the female header pins, which will work as I/O pins of external devices like sensors. The ATMega328P is a single-chip microcontroller commonly used in Arduino products. It has a high performance and consumes low power, thus executing approximately 131 instructions. In brief, the high performance is due to its advanced Reduced Instruction Set Computer architecture (RISC). Thus, the microcontroller is effective in systems or devices that require low power and cost microcontrollers. There are totally 28 pins in ATmega 328P- PU microcontroller.

The following description shows the functionality of the pins in ATmega328P- PU and also shows how they are functioned in arduino board.

Pin1 (PC6 - RESET): It is a 6th pin of port C. this pin is a RESET pin. Thus, it's also used as an I/O pin. In arduino this pin acts as RESET pin.

Pin 2 (PD0 - RxD): It is a 0th pin of port D. Its secondary functions include RxD (data input pin). In arduino, this pin works as both digital pin 0 and also Rx.

Pin 3 (PD1 - TxD): It is the 1st pin of port D. It is a data output pin. Its other secondary function is USART serial communication interface. In arduino, this pin acts as digital pin1 and also Tx.

Pin 4 (PD2 – INT0): It is the 2nd pin of the port D. It serves as an external interrupt for source 0. In arduino, this pin acts as a digital pin 2.

Pin 5 (PD3 – INT1/OC2B): It is the 3rd pin of the port D. It serves as an external interrupt for source 1. In arduino, this pin acts as the digital pin 3 with pulse width modulation (PWM).

Pin 6 (PD4 – XCK/T0): It is the 4th pin of port D. Its timer 0 for the external counter input. In arduino, this pin acts as the digital pin 4.

Pin 7 (Vcc): It is usually connected to positive voltage. In arduino, this pin acts as Vcc pin.

Pin 8 (GND): It is a pin connected to the ground. In arduino, this pin acts as a GND pin.

Pin 9 (PB6 – XTAL1): It is the 6th pin of the port B. It is an external clock input and a timer oscillator pin. In arduino, this pin acts as a crystal pin.

Pin 10 (PB7 – XTAL2/ TOSC2): It is the 7th pin of the port B. It is a chip clock oscillator and a timer oscillator for 2nd pin. In arduino, this pin also acts as a crystal pin.

Pin 11 (PD5 - T1/ OC0B): It is the 5th pin of port D. Its timer 1 external counter. In arduino, this pin acts as a digital pin 5.

Pin 12 (PD6 – AIN0/ OC0A): It is the 6th pin of port D. It functions as an analog comparator positive input and timer output. In arduino, this pin acts as digital pin 6.

Pin 13 (PD7 – AIN1): It is the 7th pin of port D. It functions as analog comparator negative. In arduino, this pin acts as digital pin 7.

Pin 14 (PB0 – ICP1/CLK0): It is the 0th pin of the port B. It functions as a timer input capture pin and divided system clock. In arduino, this pin acts as a digital pin 8.

Pin 15 (PB1 – OC1A): It is the pin 1 of the port B. It serves as a timer output compare watch. In arduino, this pin works as a digital pin 9 with pulse width modulation (PWM).

Pin 16 (PB2 – SS/OCIB): It is the 2nd pin of port B. It functions as a slave select input. In arduino, this pin acts as digital pin 10 with pulse width modulation (PWM).

Pin 17 (PB3 – MOSI/ OC2A): It is the 3rd pin of port B. It functions as master output and slave input. In arduino, this pin acts as a digital pin 11 with pulse width modulation.

Pin 18 (PB4 - MISO): It is the 4th pin of port B. It functions as master input and slave output. In arduino, this pin acts as a digital pin 12.

Pin 19 (PB5 - SCK): It is the 5th pin of port B. It is a SPI bus serial clock. It also functions as a serial peripheral interface for programming. In arduino, it also acts a digital pin 13.

Pin 20 (AVCC): It serves as power for an internal ADC converter. In arduino, this acts as Vcc.

Pin 21 (AREF): It serves as an analog reference pin for ADC. In arduino, this pin acts as a reference pin (AREF).

Pin 22 (GND): It serves as a ground pin. In arduino, this acts as GND.

Pin 23 (PC0 – ADC0): It is the 0th pin in port C. It serves as an ADC input channel 0. In arduino, this pin acts as an analog input pin 0.

Pin 24 (PC1 – ADC1): It is the 1st pin of port C. It serves as ADC input channel 1. In arduino, this pin act as an analog input pin 1.

Pin 25 (PC2 – ADC2): It is the 2nd pin of port C. It serves as ADC input channel 2. In arduino, this pin acts as an analog input pin 2.

Pin 26 (PC3 – ADC3): It is the 3rd pin of port C. It serves as ADC input for channel 3. In arduino, this pin acts as an analog input pin 3.

Pin 27 (PC4 – ADC4/SDA): It is the 4th pin of port C. It serves as ADC input for channel 4. In arduino, this pin acts as an analog input pin 4.

Pin 28 (PC5 – ADC5/ SCL): It is the 5th pin of port C. it serves as ADC channel for channel 5 and serves as an SCL (serial clock line). In arduino, this pin acts as an analog input pin 5.

ICSP Male headers:

ICSP (In-Circuit Serial Programming) These pins are used to code and boot an Arduino from an external source. These pins allow inner workings of two or more Arduino boards and also allow you to upload your firmware. The ICSP pins act as an AVR programmer which is used to code or boot the Arduino.

### 1.3 Components:

These are the components to make the arduino:

1. ATmega 328P microcontroller

2. 16 MHZ crystal oscillator
3. 100 Pf capacitor
4. 22 pf capacitor
5. IC base:
6. Zero PCB
7. Resistor 10 k
8. Resistor 220 ohm
9. Female header and male header
10. Female header power jack-1
11. LEDs

### 1.3.1 Component description

**I. ATmega 328P microcontroller:**

The Atmel ATmega328P Microcontroller is an 8-bit megaAVR device based on the AVR enhanced RISC architecture. Featured is pico Power technology that offers ultra-low power consumption and low-power sleep modes, ideal for battery powered applications.

**Specification:**

These are some of the features of ATmega328P-PU:

i. Pin count – 28

ii. Flash memory (bytes) – 32K

iii. SRAM memory (bytes) – 2K

iv. EEPROM memory (bytes) – 1K

v. Interrupt vector size - $\frac{1}{1/2}$

vi. General purpose I/O lines: 23

vii. SPI – 2

viii. TWI (I2C) – 1

ix. USART – 1

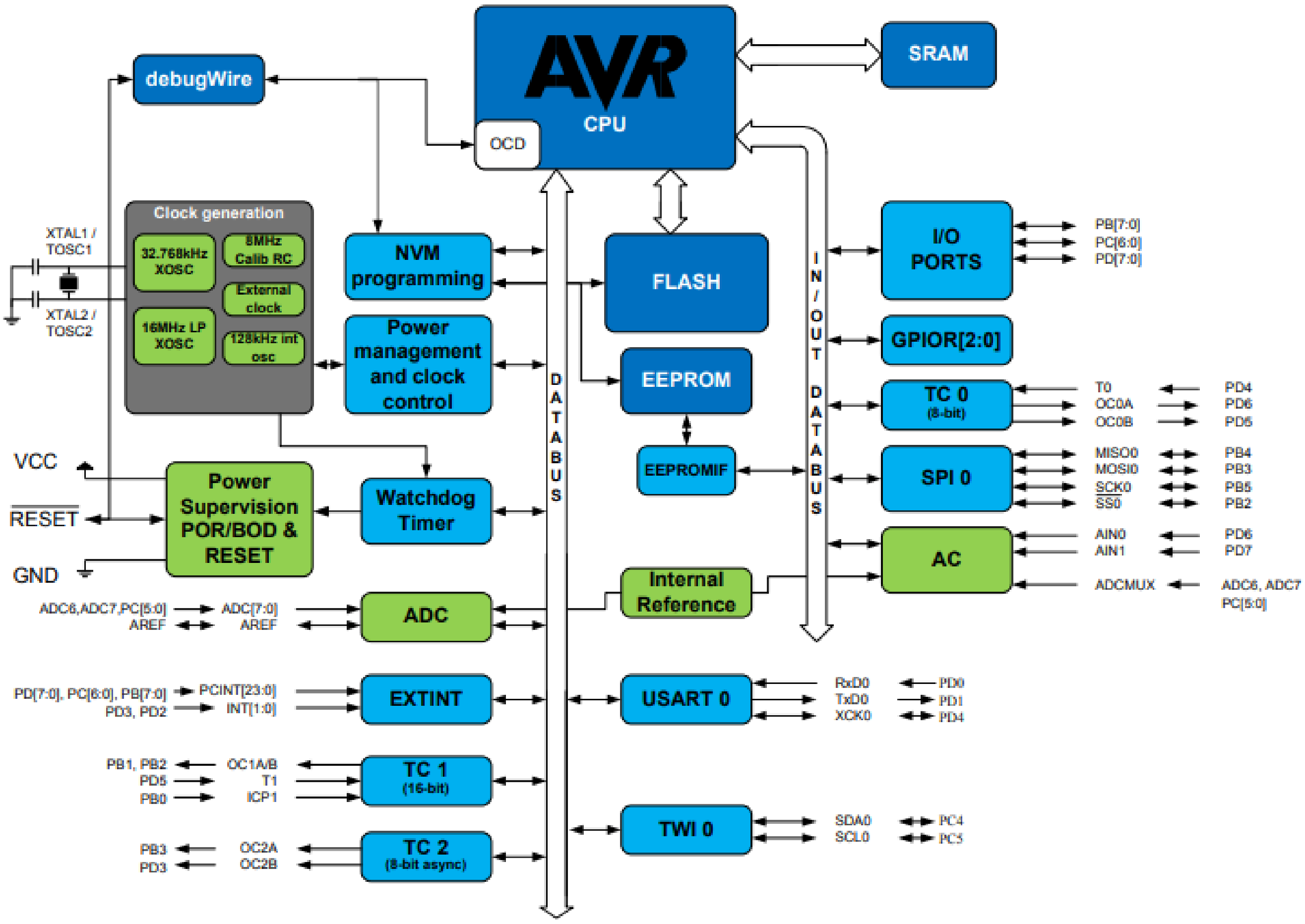
x. ADC – 10 bits 15kSPS

xi. ADC channels – 8

xii. 8-bit timer/counters – 2

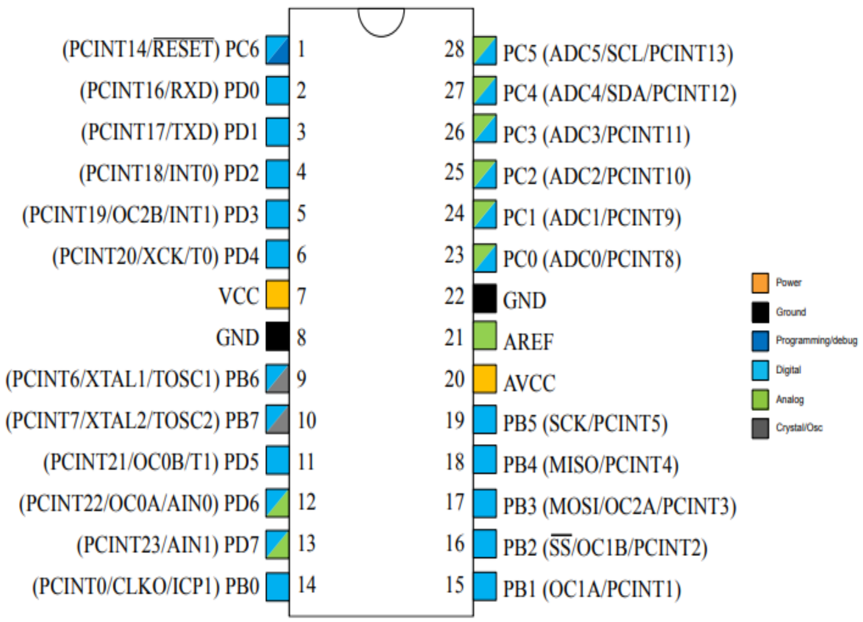
xiii. 16-bit timers/counters – 1

The following is the block diagram of ATmega 328P-PU microcontroller:



**PIN DIAGRAM:**

The following is the pin diagram of ATmega 328P-PU microcontroller:



**Pin Description:**

1. VCC: Digital voltage supply

2. GND: Ground

3. Port B (PB[7:0]) XTAL1/XTAL2/OSC1/OSC2: 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 fuse settings, 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, PB [7:6] is used as TOSC [2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

4. Port C (PC[5:0]): Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC [5:0] 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.

5. 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 un-programmed, 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. The various special features of Port C are elaborated in the Alternate Functions of Port C section.

6. Port D (PD[7:0]): 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.

7. AVcc: AVCC is the supply voltage pin for the A/D Converter, PC[3:0], and PE[3:2]. 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 PC[6:4] use digital supply voltage, VCC.

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

9. ADC [7:6] (TQFP and VFQFN Package only): In the TQFP and VFQFN package, ADC[7:6] serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

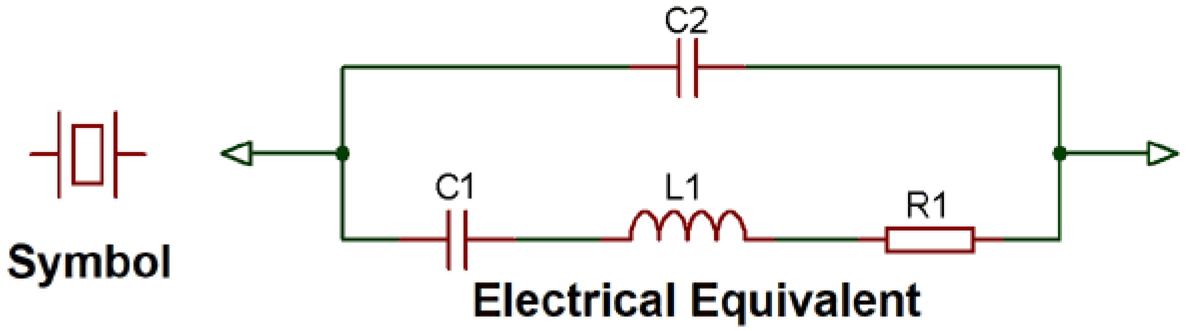
## II. 16MHz crystal oscillator

**Crystal Oscillator** is an Electronics Oscillator circuit which uses the mechanical resonance of a vibrating crystal of piezoelectric material to generate an electrical signal with an accurate frequency. It also has automatic amplitude control and frequency drift is also very low due to change in temperature. Crystal Oscillators are only suitable for high-frequency application. Every microcontroller needs a **crystal oscillator**

Crystal Oscillators have two leads, there is no polarity for crystals and hence can be connected in both directions.

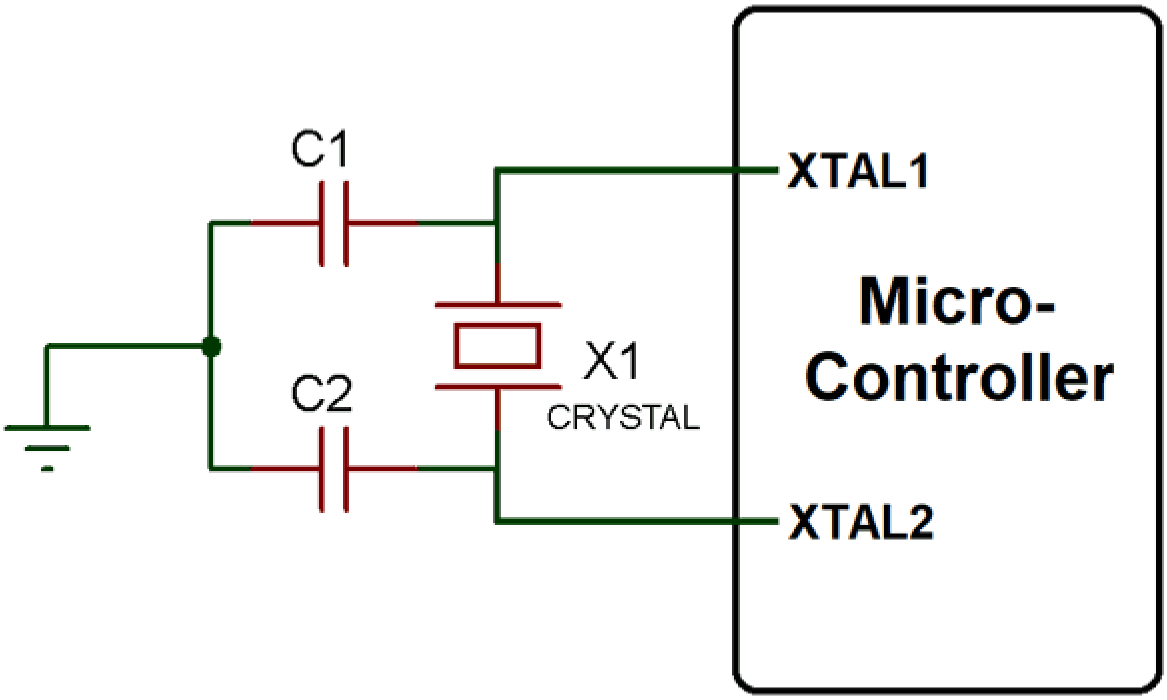
### Electrical Equivalents

A crystal oscillator is a piezoelectric device used to convert electrical energy to mechanical energy. The conversion occurs at resonant frequency. The simplified electrical equivalent of crystal oscillator is given below:



Crystal oscillator works on the principle of the Inverse Piezoelectric Effect, the applied electrical field will generate a mechanical distortion across some material. Therefore, it utilizes the vibrating crystal's mechanical resonance, which is made through a piezoelectric material for generating an electrical signal of a certain frequency.

## Crystal Oscillator with Micro-controller



The clock source used for microcontrollers are based on mechanical resonant devices such as crystal oscillator. All microcontrollers have 9th and 10th pins for connecting crystal oscillator. The pins generally named as XTAL1 and XTAL2, here the connection of crystal oscillator with micro-controller is given below:

The reason for using two capacitors in series with crystal oscillator is to resonate with the crystal inductance which cause the crystal to oscillate on its fundamental parallel resonant mode.

There are some factors which affect the frequency stability of an oscillator like variation in temperature, load and supply.

- What is the use of crystal oscillator in Arduino?

Arduino crystals are used because it helps when Arduino is dealing with time issues, for example if we build a project that turns a switch A OFF after 15 minutes ON and turn ON switch B, C & D one after another with a 20 minutes step. We can use Arduino software to program Arduino to do that, but how Arduino calculate the time?

The answer is by using this crystal, the number on the top of Arduino crystal is 16.000H9H this gives us information about the frequency which is 16,000,000 Hertz or 16 Mhz, this small component can make 16 millions cycles per second so using this concept Arduino can calculate time.

- Why only used 16MHZ crystal oscillator not used 32 MHZ and other ?

  In the data sheet of ATmega328P their it is clearly mentioned that maximum frequency should not exceed 20MHz,so 16Mhz is used.

**III 22 Pf capacitor :**

22pf capacitor is used for the crystal oscillator circuit . the two 22 pf capacitor is used serially with the 16 MHZ crystal oscillator , where the value of c1 and c2 are same . the crystal oscillator is designed for a 22 pf load capacitance .

The following formula may be used to calculate a parallel resonant crystal's external load capacitors:
CL = ((CX1 x CX2) / (CX1 + CX2)) + Cstray
where:
CL = the crystal load capacitance
Cstray = the stray capacitance in the oscillator circuit, which will normally be in the 2pF to 5pF range.

Assuming that CX1=CX2 then the equation becomes:
CL = ((CX1 x CX1) / (2 x CX1)) + Cstray
CL = (CX1 / 2) + Cstray
Rearranging the equation, we can find the external load capacitor value:
CX1 = 2(CL - Cstray)

For example, if the crystal load capacitance is 14pF, and assuming Cstray=3pF, then:

CX1 = CX2 = 2(14pF - 3pF) = 22pF

**IV 10 Pf capacitor**

On this Arduino Uno, we used a 10uF capacitor to prevent the Arduino from resetting when a serial connection is opened.

the reset circuit is a 100nF capacitor between DTR and the Reset pin (which is pulled high by a 10k resistor). So when DTR toggles low, you get a low pulse into the Reset pin until the 100nF capacitor has charged via the 10k resistor.

When you put a 10uF capacitor between the Reset pin and Ground, that 10uF capacitor is charged by the 10k resistor, and when the 100nF capacitor "goes low", it's instantly charged by the 10uF capacitor instead of pulling low while being charged by the 10k.

**V 10 k resistor :**

10K Ohm is a common pull-down or pull-up resistor value. This is likely to be a pull-down in this case since the Arduino's microcontroller has internal pull-up resisotrs you can turn on in your code. By connecting a resistor between the switch pin and ground it puts the pin in a known state when the button is open. Without this resistor the pin will be randomly switching from HIGH to LOW and so reading the button will not be accurate. If you use too high of a resistor value as a pull-down then it will not be strongly pulled to ground and electrical noise could still cause the pin to read as HIGH even when the button is not pressed. 10K is a conservative pull-down resistor value that should be quite reliable under normal circumstances. In a situation where you have the button on a long wire (which acts as an antenna) and lots of electrical noise, you might use an even lower resistor value. A low resistance pull-down resistor will cause more power consumption than a high resistance one so for a battery power application it's a good idea to make sure you're not using a lower value than needed

**220 ohm resistor :** A 210-ohm resistor is an electronic component that is used to resist the flow of electricity in a circuit. Resistors are used in a wide variety of electronic circuits to control the flow of current and protect other components from damage. 210-ohm resistors are a commonly used resistance value in electronic circuits

**POWER SUPPLY CIRCUIT:** Some Arduino boards have an **onboard barrel jack connector** that is used to connect external power supplies. Arduino board with onboard barrel jack are configured with positive polarity ; this means a negative sleeve and positive pin .board with an on barrel jack use a negative 5.5 mm sleeve and a 2.1 mm positive. The voltage line from the barrel jack connector is regulated in Arduino boards using their onboard voltage regulator; usually, it is first regulated to 5V and then regulated again to 3.3V. in most Arduino boards. The **recommended voltage and current ratings for external regulated DC power supplies** connected to the barrel jack connector are 7 – 12 volt, 1 ampere.

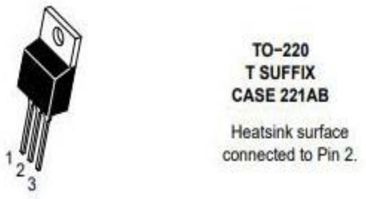
The component are used to make power supply are as follows;

1. Barrel jack connector - 1
2. LM 7805 voltage regulator - 1
3. 10 uf capacitor - 2
4. Led - 1

5. 220 ohm – 1

## Voltage regulator :

An LM7805 voltage regulator is a voltage regulator that outputs +5 volts that find its application in most electronic project . In electronic circuits ,voltage regulator are very common. for a variable input voltage ,they offer a constant output voltage . in our situation , the LM7805 IC is a well – known regulator IC that is used in a wide range of projects. The designation LM7805 has two meaning : “78” indicate that it is a positive voltage regulator ,and “05” indicates that it output 5V. As a result , our LM7805 will output a value of +5V. this IC has a maximum output current of 1.5 A. however , because the IC loses a lot of heat , a heat sink is recommended for project that use a lot of currents .If the input voltage is 12v and you are consuming 1A , then (12-5) * 1 = 7W is the result . these 7 watt will be converted to heat and dissipated



**Pin out:**

| Pin Number | Pin Name | Description |
|---|---|---|
| 1 | Input (V+) | Unregulated Input Voltage |
| 2 | Ground (Gnd) | Connected to Ground |
| 3 | Output (Vo) | Outputs Regulated +5V |

### 10 uf capacitor:

Electrolytes are great because they can pack *a lot* of capacitance into a relatively small volume. They're especially well-suited to high-voltage applications because of their relatively high maximum voltage ratings. Unfortunately, electrolytic caps are usually polarized. They have a positive pin -- the anode -- and a negative pin called the cathode. When voltage is applied to an electrolytic cap, the anode must be at a higher voltage than the cathode. The cathode of an electrolytic capacitor is usually identified with a '-' marking, and a colored strip on the case. The leg of the anode might also be slightly longer as another indication.

**Connection:**

Step 1: connect LM7805 voltage regulator at zero PCB

Step 2 : connect barrel jack in zero PCB and connect pin 1 and pin 2 together of barrel jack & then connected with the ground of voltage regulator LM7805

Step 3 : between barrel jack and voltage regulator connect 10 uf capacitor positive terminal with v-in negative terminal with vcc of barrel jack .

Step 3 at v-out of voltage regulator connect 10 k capacitor positive terminal and negative terminal of capacitor with one terminal of register and another terminal of register is connected to LED's positive terminal and negative terminal of led is connected to the ground .

Step 4: take a common vcc wire from the v out of the voltage regulator for futher connecting Arduino & similarly take a common ground from ground of voltage regulator.

## 1.4 How to burn a bootloader ?

The following instructions are for targets that use the AVR architecture (e.g., Uno, Nano, Leonardo, Mega).

You will need an ISP programmer. If you don't have a programmer, you can use a spare Arduino board as an "Arduino as ISP" programmer. Although the "Arduino as ISP" only works for programming targets of the AVR architecture, you can use boards of any architecture as an "Arduino as ISP" programmer.

Instructions:

1. Make the following connections between the Arduino board you will be using as the programmer and the target. Refer to the "Connections" table on the SPI library to determine the pins:

| Programmer | Target |
|---|---|
| MISO | MISO |
| VCC | 5V (or VCC on 3.3 V boards) |
| SCK | SCK |
| MOSI | MOSI |
| 10 | RESET |
| GND | GND |

2. Connect the programmer board to your computer with a USB cable.
3. **File > Examples > 11.ArduinoISP > Arduino ISP**
4. Select the port of your board from the **Tools > Port** menu.
5. **Sketch > Upload**
6. Wait for the upload to finish.

You are now ready to burn the bootloader using your "Arduino as ISP" programmer.

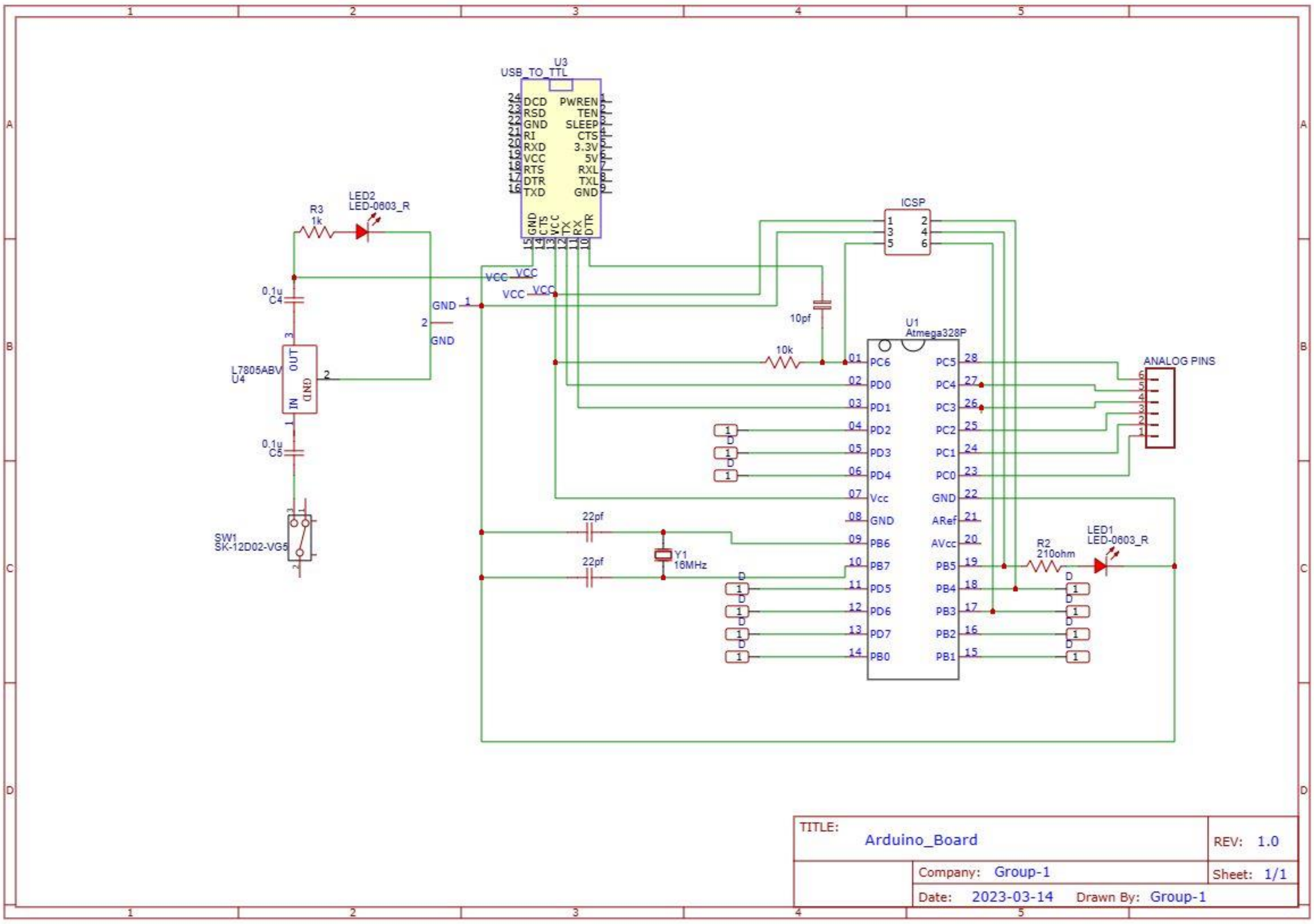
Instructions for burning the bootloader:

1. Connect an ISP programmer to the ICSP header on your Arduino board.
2. Select the target board from the **Tools > Board** menu and any other custom **Tools** menus (e.g. **Tools > Processor**).
3. Select the appropriate programmer from the **Tools > Programmer** menu.
4. **Tools > Burn Bootloader**
5. Wait for the process to finish successfully.

You can also use your ISP programmer to upload sketches to the target board via **Sketch > Upload Using Programmer**. Note that when you do this, the bootloader is erased. After using **Upload Using Programmer**, you would need to do another **Burn Bootloader** before you can go back to uploading to the target board normally via the USB cable.

## 1.5 Arduino board schematic :

+



### 1.6 Procedure:

Step1: soldering IC base , female header and male header at zero PCB .

Step 2: At pin 8 and pin 9 parallelly connected a crystal oscillator and two 22 pf capacitor are connected serially with the crystal oscillator and other end is connected to the ground . .

Step 3: connect 10 k resistor at reset pin and other end with vcc pin 4 of FDTI.

Step 4 : Now connect 100 nf between 10k resistor and DTR of FDTI .

Step 5 : at pin 19 connect 220 ohm resistor with positive terminal of LED and negative terminal to the ground .

Step 6 : connect pin 19 with ICSP 4 pin & pin 1 of ICSP connected with pin no 18 . and 6 pin of ICSP is connect with pin no 17.

Step 7: connect pin no 1 of ICSP with VCC , pin no 3 with ground & pin no 5 with reset pin of ATmega 328 p IC.

Step 8: pin no 22 is ground pin which is connected to the ground of FTDI.

Step 9 : there are 14 digital pin which is mapping according to the schematic one by one to the headers.

Step 10: there are 6 analog pin which pin no 28 to pin no 23 is mapping according to the schematic one by one to the headers(analog header).

### 1.8 Result:

The arduino is designed by our own on zero PCB.