# Experiment

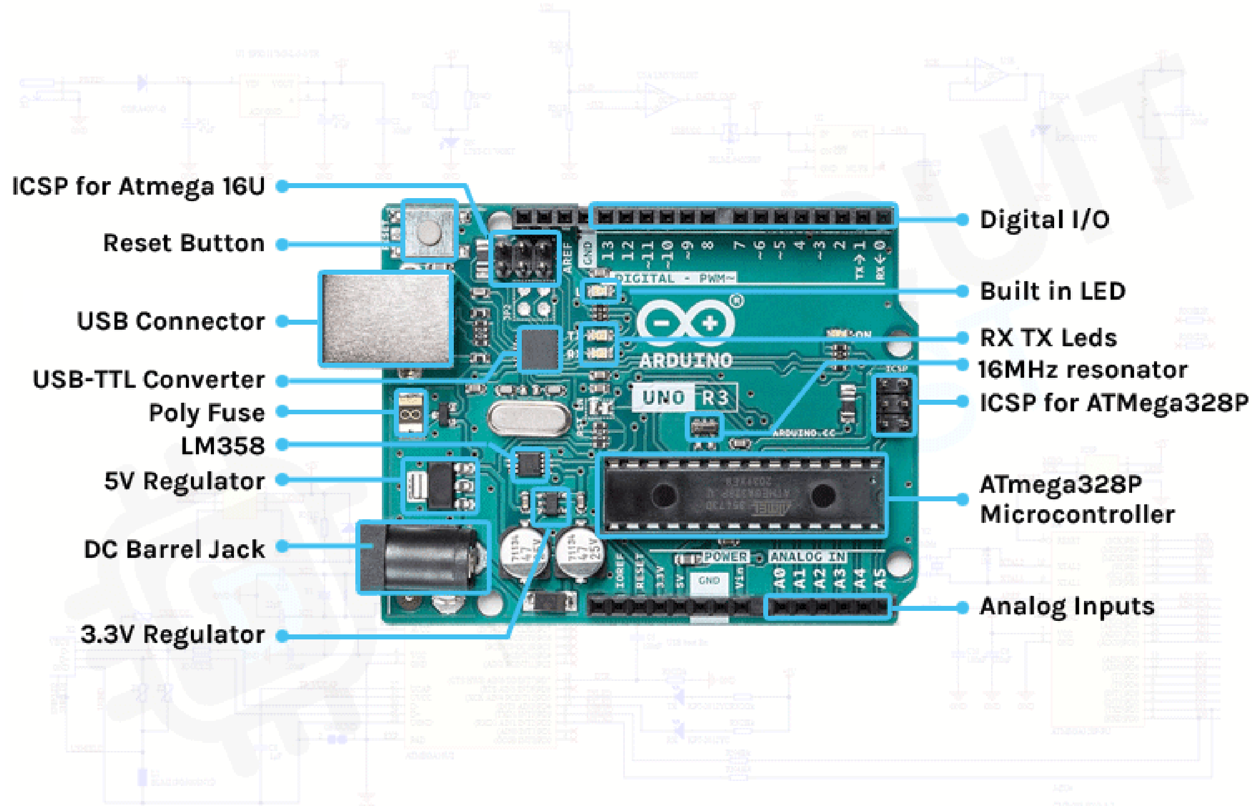
## AIM

**Design a Prototype for basic Arduino(UNO R3) Board**

## Apparatus Used

1. ATmega328 with Base
2. Zero PCB
3. 10k Ω Resistor x5
4. Female header
5. DC Barrel Jack 12 volt
6. 5 volt Regulator IC LM-7805
7. 16 Mhz Crystal Oscillator
8. 22 PF x2 Capacitor Ceramic
9. LED x 4
10. Reset Button
11. 220 Ω Resistors x 5
12. 10 μF Capacitor Electrolyte
13. ICSP Header

# Theory




In the UNO board, the main component is the ATMega328P. It is the heart of the Arduino UNO. Near the MCU you can see a 16MHz resonator which will give the ATMega328P the clock signal to work. Near that, you can also see a connector named ICSP. It is used to burn the Arduino bootloader into the chip. And you can also see the header pins for the I/O.

If you look at the other side of the board, you can spot another microcontroller in a QFN package. It is an ATMega16U and is used as a USB -TTL converter. Near that, it will have its crystal and ICSP port to burn the firmware. There will be a reset button near it, which will reset the ATMega328P.

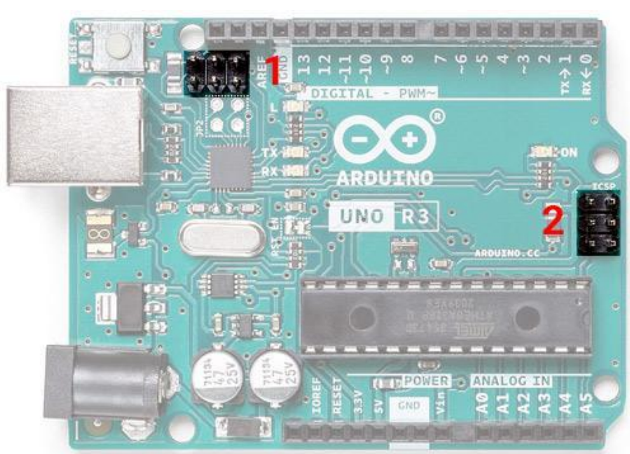
You can see the USB port and DC barrel jack on the left side. You can power the Arduino either through the USB port or the barrel jack. The barrel jack will accept a voltage range of 7-12V. And near the barrel jack, you can find two voltage regulators. One for 5V and one for 3.3V. Let’s check out each component.

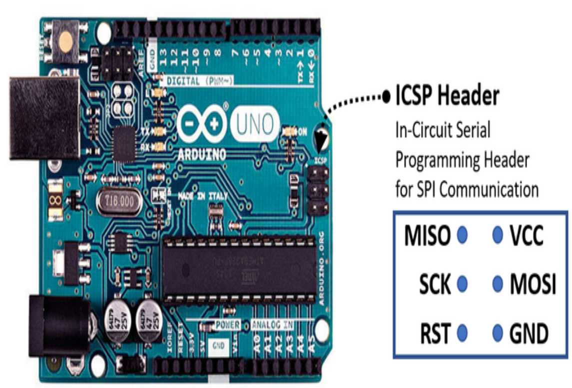
## USB - B Socket

The USB socket on the UNO has two functions. One is for communication, to connect with the computer through a USB port, and also to load the firmware into the Arduino with the help of the bootloader. The second is to power the Arduino. You can use the USB port to power the Uno directly from any USB port.

## ISCP Pins

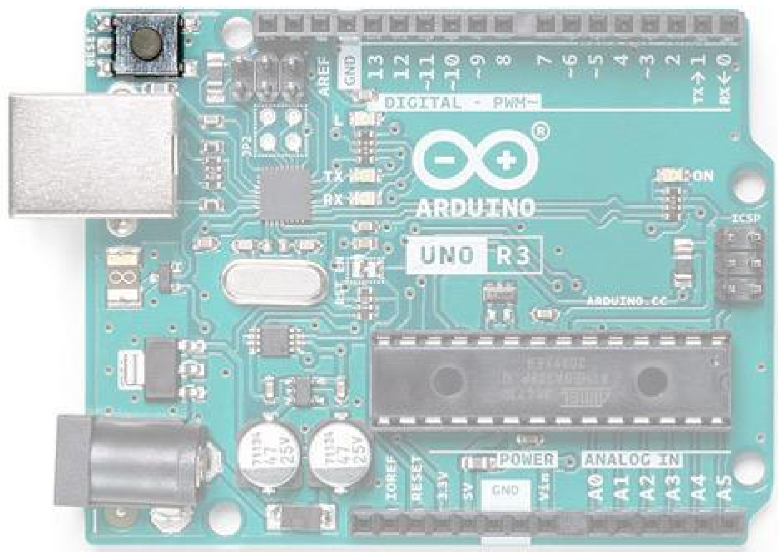
In the UNO you can find two 6 pin connectors. One is near the USB – TTL Chip and the other one is at the end of the board. These pins are used to program those two microcontrollers. The USB – TTL chip on this board is an ATMgega16U. The connector marked as 1 is used to program the USB-TTL firmware into this chip. And the connector marked as 2 is used to burn the bootloader into the ATMega328 microcontroller.

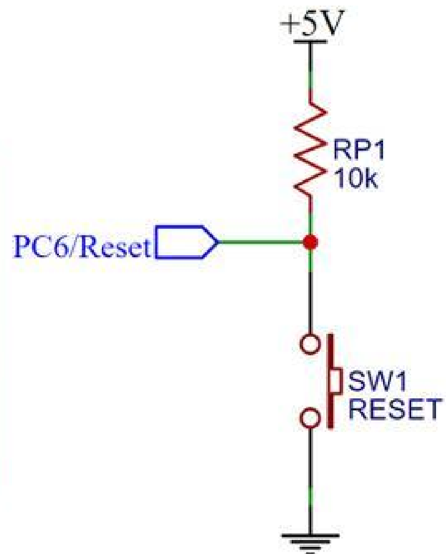


## Reset Button

**As the name indicates this tactile switch is used to reset the ATMega328 microcontroller. It's connected to the PC6/Reset pin, which is pulled up through a 10K. When the switch is pressed the pin is pulled to the ground and the chip will reset.**





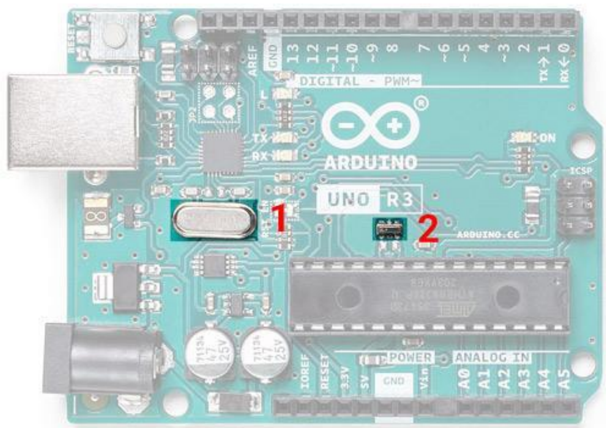

## USB-TTL Interface Chip

**To communicate with the computer, the Arduino relies on a USB-TTL interface. In UNO, ATMega16U with custom firmware act as a USB – TTL interface chip.**

## Crystal Oscillator/ Ceramic resonator

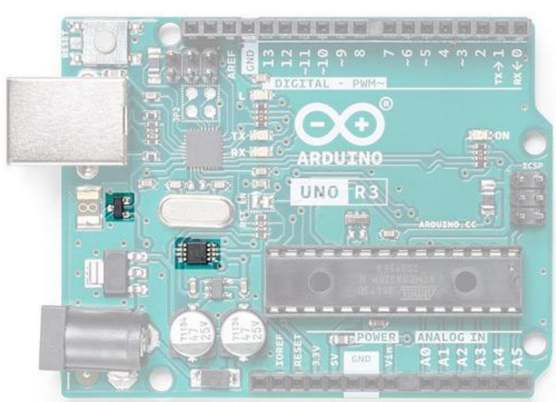
**For a microcontroller to work it needs a clock source. The clock circuit determines the speed with which the microcontroller operates. How many instructions per second it will execute is dependent on the clock frequency. The ATMega series microcontrollers can use two types of clock sources. One is an internal RC oscillator that is already built into the microcontroller. But the drawback of using the internal oscillator is that its maximum frequency is limited and it is not that accurate. That is where the second option comes into place, i.e.,**

using an external clock generator. In this case, we will be using a Quartz crystal oscillator or a ceramic resonator for this purpose. In the picture below, you can see two components are marked. The first one is a 16MHz crystal oscillator used for the ATMega16U2 chip and the second one is a 16MHz resonator used for the ATMega328P microcontroller.



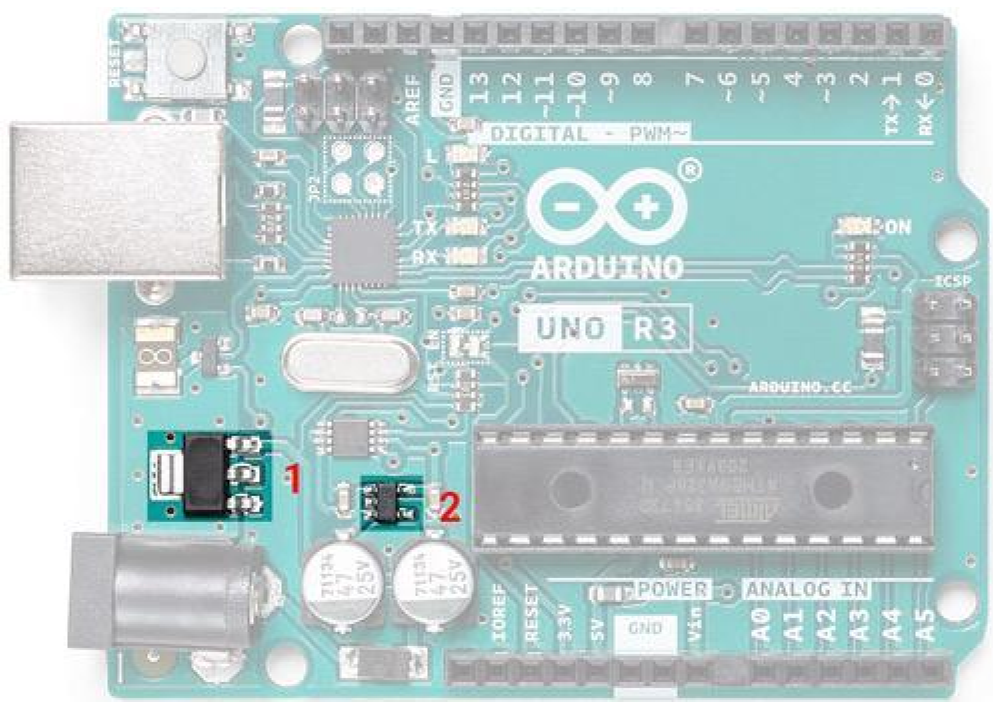

## Power Path control

If you inspect a UNO, you can find an LM358. You might think what's its role here. It's used as a comparator to control the input power path. When the input power is provided through the barrel jack or Vin pin the power path control circuit will cut off the USB power pin from the circuit which in fact will protect the USB port.



## Voltage Regulator

**The ATMega328 and ATmega16U2 have a maximum input voltage of around 5V and most modules or accessories work on either 5V or 3.3V. The Arduino can accept 7-12V through the Vin pin or the DC barrel jack. So, to step it down, there are two regulators onboard. One is a 5V regulator (marked as 1) for the microcontrollers and the other one is a 3.3V regulator which is used to provide 3.3V through 3.3V pin.**




## DC Barrel Jack

**The DC barrel jack is used to supply power to the UNO. We can supply 7-12V through it and hence we can use a 12V DC adapter or 9V DC adapter on this Jack to power the Arduino board.**

## Digital and Analog I/O

**The Arduino UNO has 14 digital I/O pins and 6 Analog inputs. The digital I/O pins are 5V logic level and you can also use**

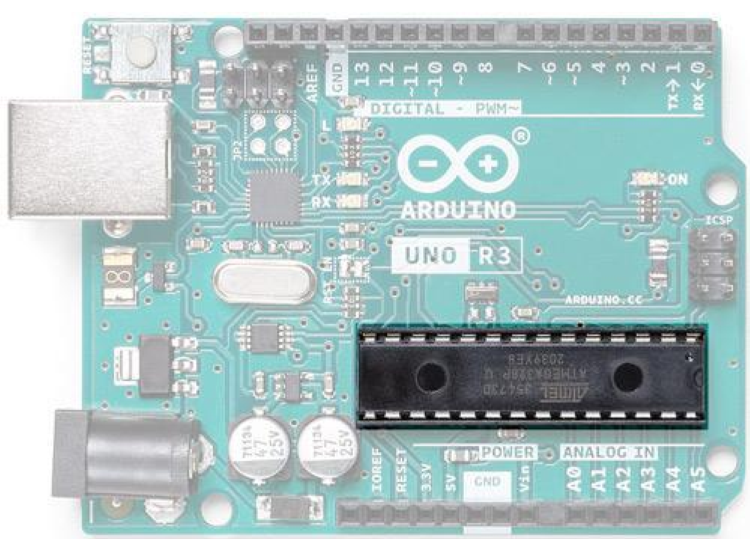
the Analog pins as digital I/O too. Arduino UNO supports 6 channel 10 bit ADC inputs through A0-A5, which can be sampled and analyzed using UNO.

## Status LEDs and Inbuilt LED

**Uno has 4 LEDs onboard. One is used as a power indicator and two are used to show the activity of the Rx and Tx pin. The other one is tied to the Digital pin 13, which can be used to test the Arduino board or simply as an indicator.**

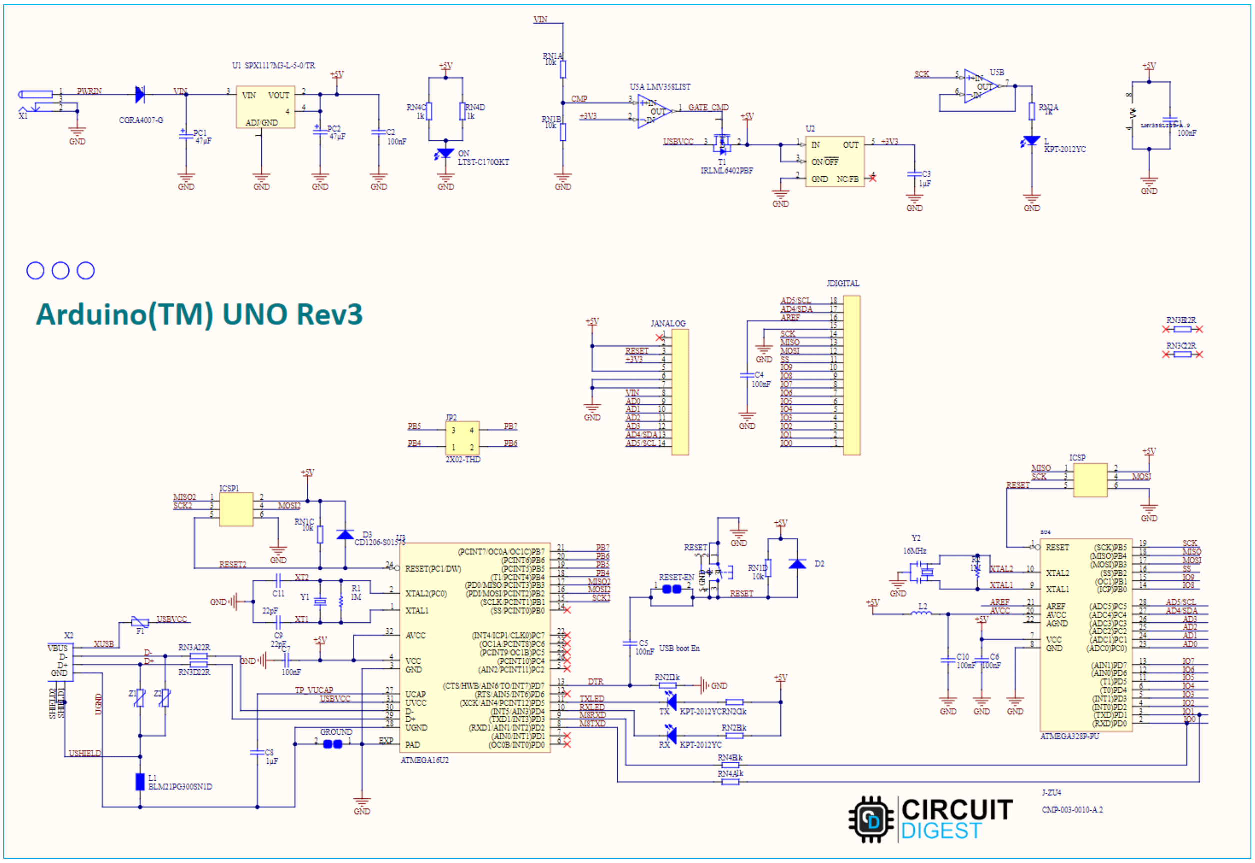
## ATMega328P – The Brain

**Last but not least is the main component on the Arduino board – the ATMega328P Microcontroller. UNO uses a 28Pin DIP version of ATMega328P. Atmega328P is pre-programmed with a bootloader that allows you to directly upload the program to Arduino through USB without the need for an external programmer.**



## Arduino UNO Circuit Diagram

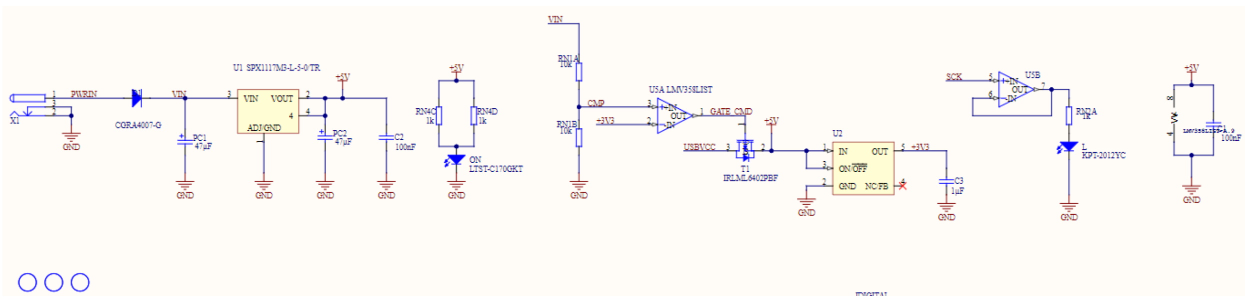
**Now let's have a look at the Arduino UNO Schematics. The below circuit diagram, clearly shows all how the various components are connected to design an Arduino UNO board.**



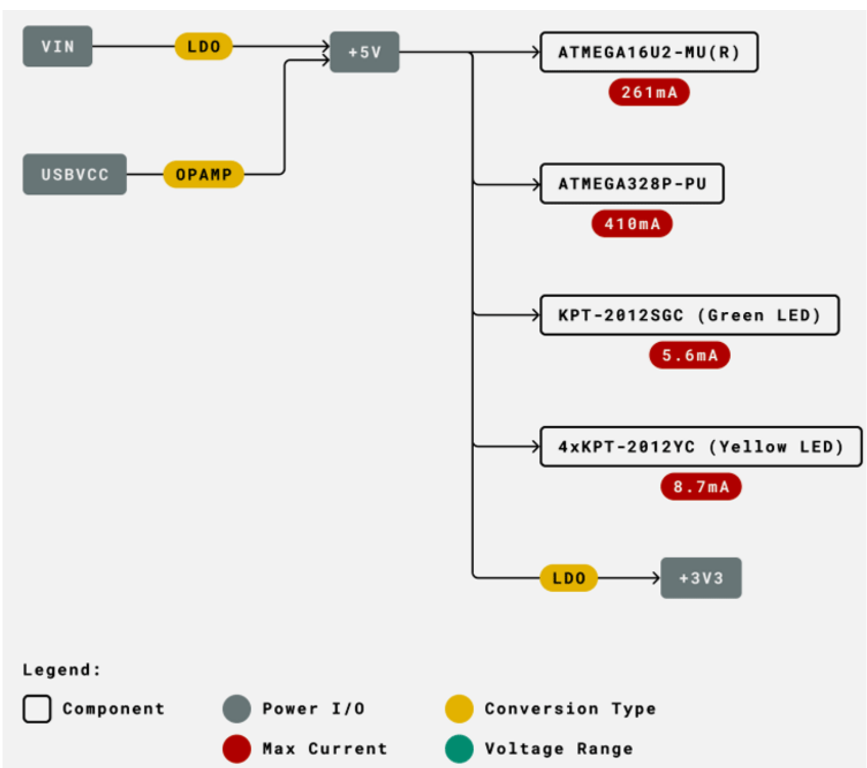

**Here is the power section of the circuit in which you can find the DC barrel jack for the power input, 5V, and 3.3V regulators, and the power path control circuit around the LM358 comparator circuit.**

**The 5V regulator is the NCP1117ST50T3G(We used LM7805) and the Vin of this regulator is connected via DC jack input through the M7 diode which acts as reverse polarity protection. The output of the 5V regulator is connected to the rest of the 5V net in the circuit and also to the input of the 3.3V regulator, LP2985-33DBVR.**

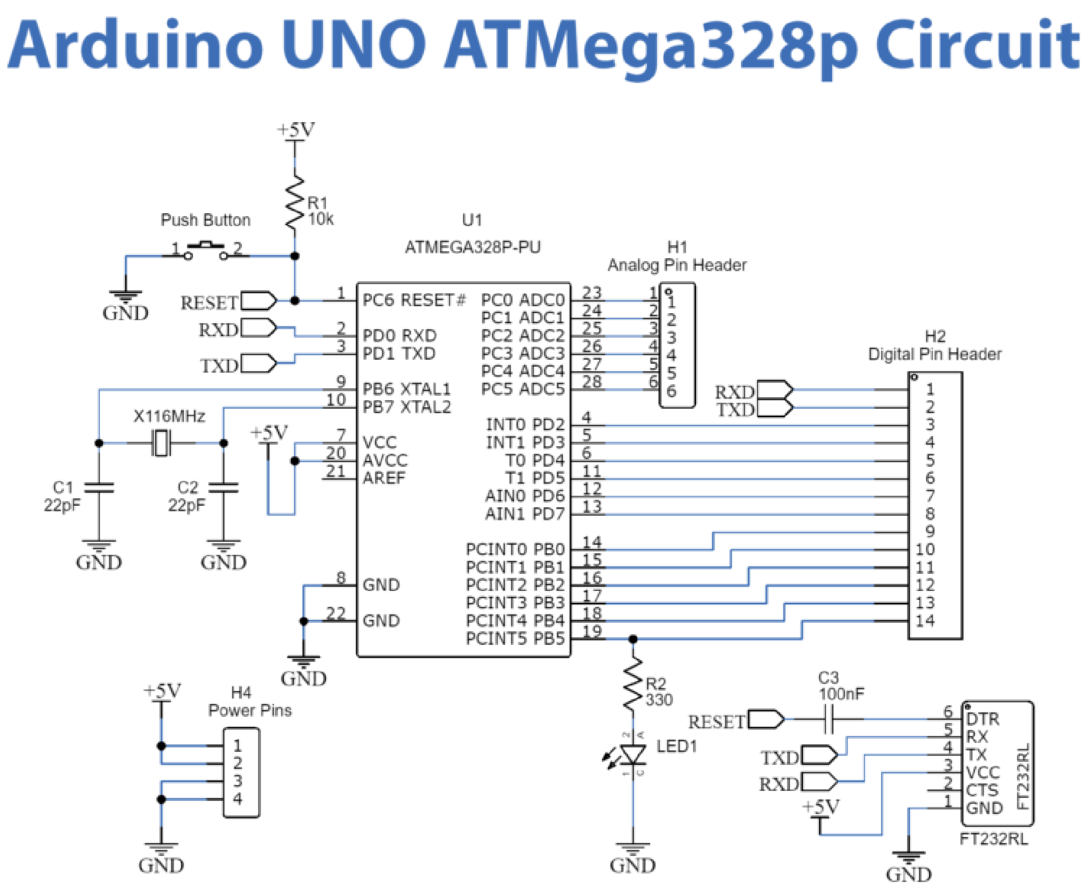
**Another source of 5V is the VCC pin of the USB which is connected to the drain of an FDN340P, a P-channel MOSFET, and the source is connected to the 5V net. The gate of the transistor is connected to the output of an LMV358 op-amp used as a comparator. The comparison is between 3V3 and Vin/2. When Vin/2 is larger, 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 USB VCC is connected to 5V.**



***Here is the power tree view of Arduino UNO***



**Now let's have look at the USB-TTL section. As we discussed earlier the main component in this is the ATMega16U2 microcontroller. From an electronic design perspective, this section is similar to the microcontroller section. This MCU has an ICSP header, an external crystal with load capacitors (CL), and a Vcc filter capacitor. Z1 and Z2 are varistors or VDRs and are used as ESD protection devices. And we can also find two resistors in series with the D+ and D- pin which is used for the proper termination impedance for the USB data lines.**




**Now the main part of the schematics, which accommodates the ATMega328P microcontroller and its supplementary component**

# Circuit

## Voltage Regulator Circuit

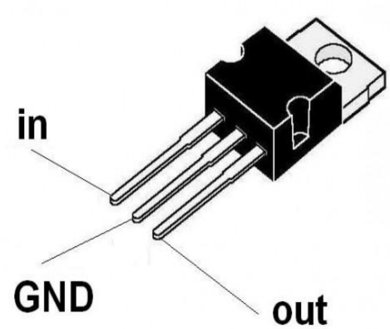
**DC Barrel Jack-**
**Series: DCJ200-10**
**PCB Mount type: Thru Hole**

**Orientation: Horizontal**

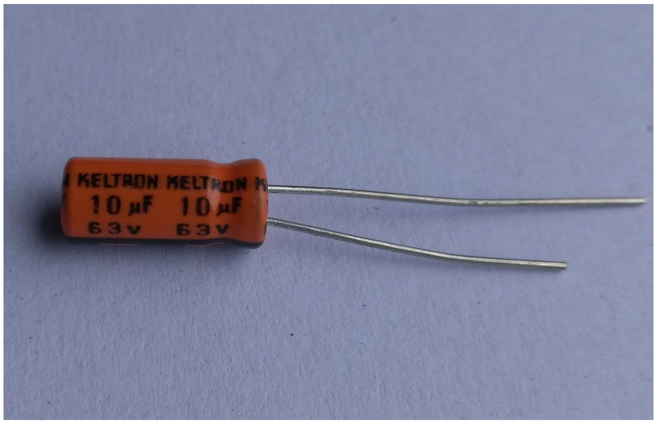
**Jack Pin Diameter: 2.00mm**
**Jack Cavity Diameter: 6.30mm**
**Switch Type: 2 conductors / 3 contacts**
**Current Rating: 5.0 Amps**
**Mate With Plug Inner: 2.05mm**
**Mate With Plug Outer: 5.50mm**
**Shielding: Unshielded**
**Locating Peg Options: Without locating peg**

### LM7805

| | |
|---|---|
| **Maximum input voltage** | **25V** |
| **Minimum input voltage** | **7V** |
| **Output voltage (regulated)** | **5V** |
| **Output current** | **1.5A** |
| **Junction temperature (in degree Celsius)** | **0-125** |
| **Operating current** | **5mA** |
| **Protection** | **Short circuit and thermal overload** |



## Capacitors(2 used)




**Capacitance = 10uf**
**Maximum Working Voltage = 63 volts**
**Capacitor Diameter = 5mm**
**Capacitor Length = 11mm**
**Load Life at 85°c = 3000 Hours**
**Useful life at 40°c = 60,000 to 90,000 Hours Approx**.7

The 10 micro farad (µF) capacitor is typically used in conjunction with the Arduino Uno's ATmega328P microcontroller to provide power supply decoupling. Decoupling capacitors ( low pass filter ) are used to filter out any noise or voltage fluctuations that may be present on the power supply lines, preventing them from affecting the performance and stability of the microcontroller.

## LED and Resistor

| | | | | | |
|---|---|---|---|---|---|
| **RC02 83-01** | **Super Bright Red** | **Intensity:** | 4,500mcd | **Voltage:** | 1.6v-1.9v |
| | | **Colour Freq:** | 620-628nm | **Typical:** | 1.8v |
| | | **Viewing Angle:** | 18º | **Current:** | 18mA |
| | | | Lens: Water Clear | | |

**Resistance =Voltage/Current**

**R=V/I ;V=5-1.8,V=3.2; I= 18mA**

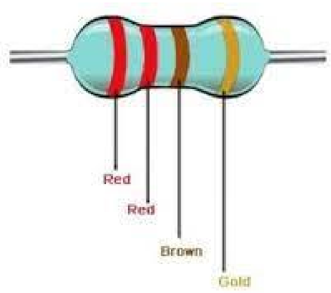
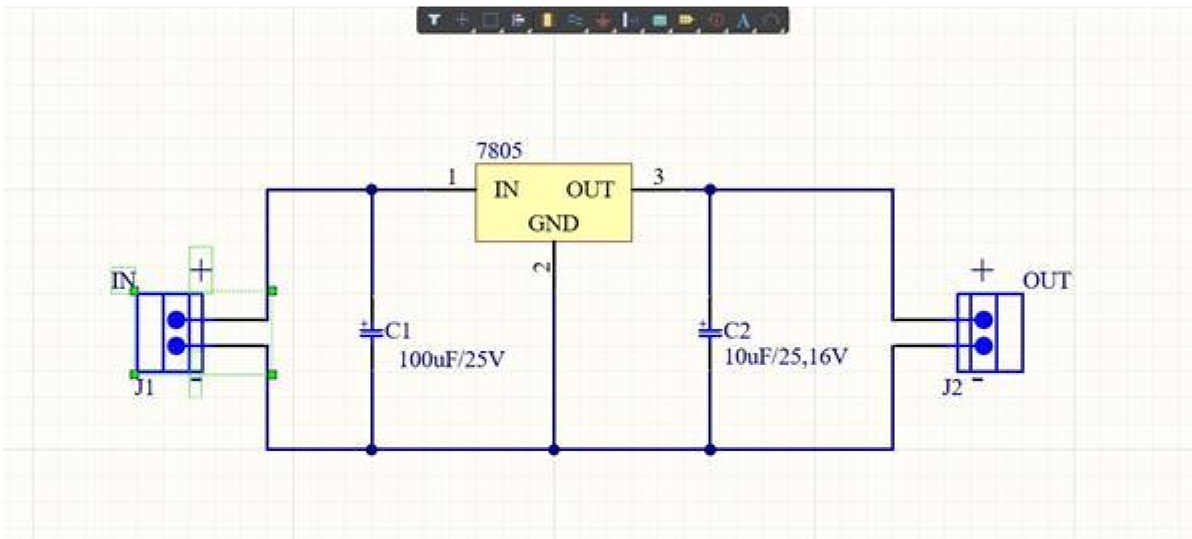


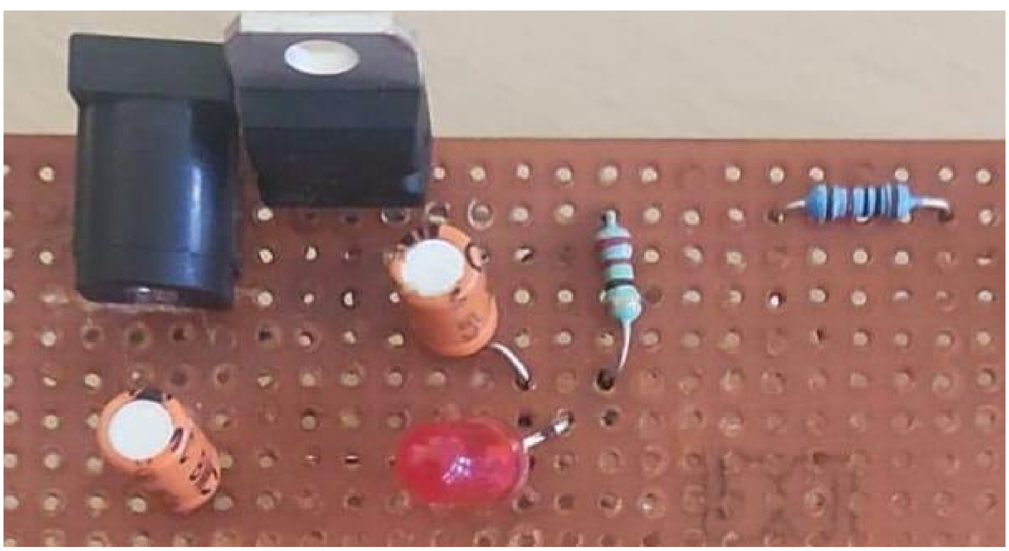

fig. showing the color code of 220 ohm resistor

**Hence Resistance is 177 ohm**

**We used a resistance of 220 ohms**

## Circuit-



## Reset Switch Circuit

## Pull-Up Resistors

Pull-up resistors are resistors used in logic circuits to ensure a well-defined logical level at a pin under all conditions. As a reminder, digital logic circuits have three logic states: high, low and floating (or high impedance). The high-impedance state occurs when the pin is not pulled to a high or low logic level, but is left “floating" instead. A good illustration of this is an unconnected input pin of a microcontroller. It is neither in a high or low logic state, and the microcontroller might unpredictably interpret the input value as either a logical high or logical low. Pull-up resistors are used to solve the dilemma for the microcontroller by pulling the value to a logical high state.

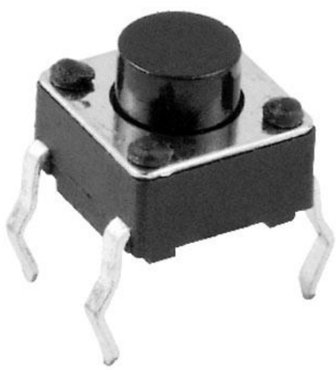
## Pull-Down Resistors

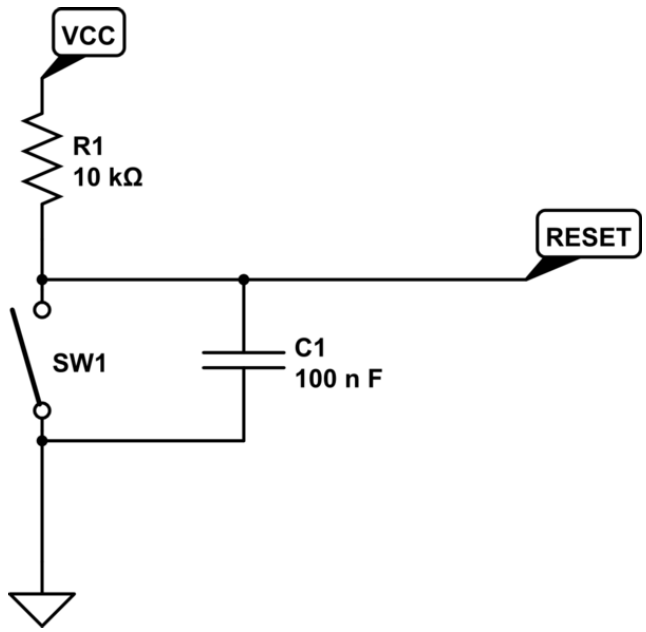
Pull-down resistors work in the same manner as pull-up resistors, except that they pull the pin to a logical low value. They are connected between ground and the appropriate pin on a device

## Value of Pull-Up Resistors

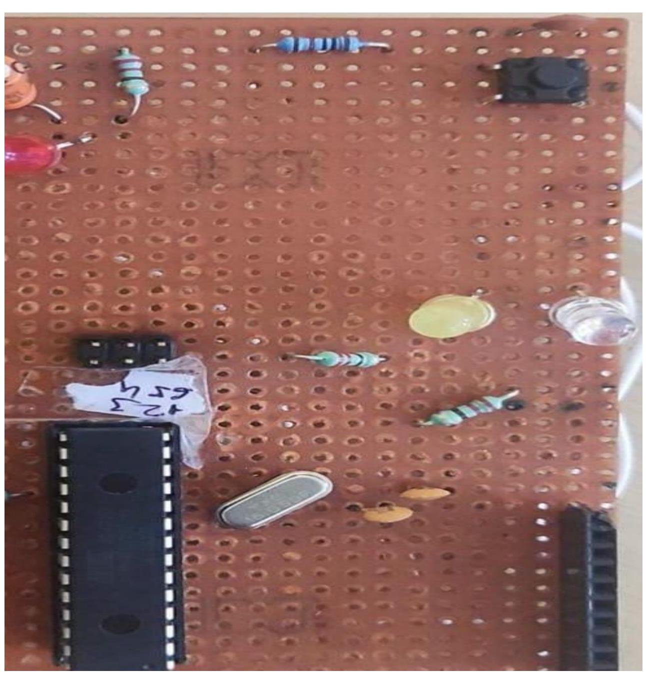
The appropriate value for the pull-up (or pull-down) resistor is limited by two factors. The first factor is power dissipation. If the resistance value is too low, a high current will flow through the pull-up resistor, heating the device and using up an unnecessary amount of power when the switch is closed. This condition is called a strong pull-up and is avoided when low power consumption is a requirement. The second factor is the pin voltage when the switch is open. If the pull-up resistance value is too high, combined with a large leakage current of the input pin, the input voltage can become insufficient when the switch is open. This condition is called having a weak pull-up. The actual value of the pull-up’s resistance depends on the impedance of the input pin, which is closely related to the pin’s leakage current.

A rule of thumb is to use a resistor that is at least 10 times smaller than the value of the input pin impedance. In bipolar logic families which operate at operating at 5 V, the typical pull-up resistor value is 1-5 kΩ. For switch and resistive sensor applications, the typical pull-up resistor value is 1-10 kΩ

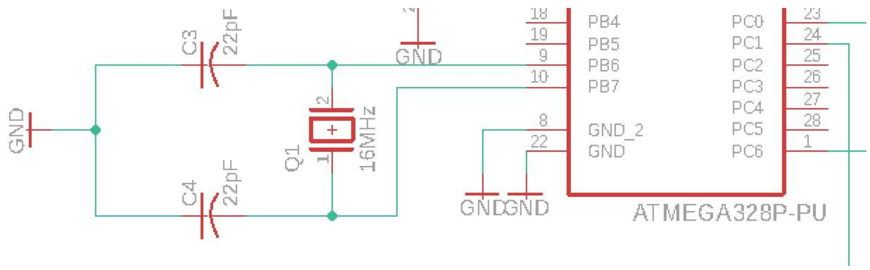





Added a resistor because a floating pin is subject to electrical noise which could cause random resets.
the value of R is so high (10K) that there is no risk of short circuit: I=V/R = 5/10000 = 0.5mA



# Crystal Oscillator circuit




## A 16 MHZ Oscillator is used

The two 22pF capacitors are used cause-
The low-value capacitors (often between 22 and 39pF) in crystal circuits are almost always used to “load” the crystal so that the circuit oscillates at the correct frequency. In the most common circuit the crystal acts like a large inductor, and as such it needs a small capacitance in parallel with it in order to resonate at the desired frequency. Sometimes one of the capacitors is a variable capacitor to allow the resonant frequency to be adjusted, but this is rarely required in microprocessor-based (or microcontroller-based) devices - particularly when the crystal is used to set the micro’s clock frequency.

$Cl = ((C1 * C2) / (C1 + C2)) + Cs$

Where,

CS = Stray capacitance = 7 μF

CL = load capacitance = 18 μF

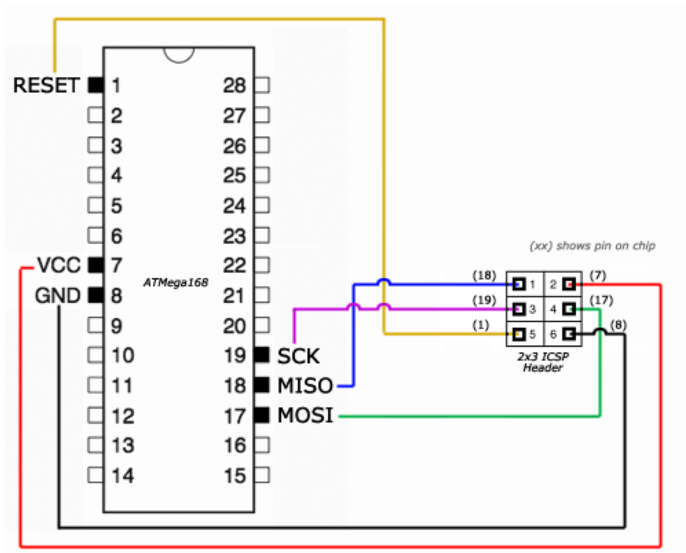
Let’s assume for ease, C1 = C2 = x

Calculation:-

$18 = (x^2)/(2x) + 7$ ; $18 = x/2 + 7$ ; $18 - 7 = x/2$

$11*2 = x$ ; x= 22 μF

## ICSP Header Connection




Pin 1 is MISO
Pin 2 is SCK
Pin 3 is MOSI
Pin 4 is GND
Pin 5 is VCC
Pin 6 is Reset

Where

- MISO (Master In Slave Out) - The Slave line for sending data to the master,
- MOSI (Master Out Slave In) - The Master line for sending data to the peripherals,
- SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master

## Programming Through ICSP

The first step is to set the `programmer` Arduino as `ISP`. To do this, go to the tools tab in Arduino IDE and set the `programmer` as Arduino as `ISP`.

The next step to use an example code present in the Arduino examples section as `ArduinoISP`. In this example code, you can see many comments which will guide you through setting out different parameters. This code also supports three LEDs which are used to check the operation of the process. For example, the middle LED indicates if there is an error in programming. The next step is to burn the `bootloader`. To do this, you can go to the `Tools` tab and select burn `bootloader`. You can set different parameters of the bootloader like how much memory is reserved for the `bootloader` and disabling reset. So make sure to read the documentation of your Arduino board before doing this. You can also use this method to program other AVR microcontrollers because, in some projects, you don't need a board; instead, you only need a microcontroller like a PIC microcontroller.

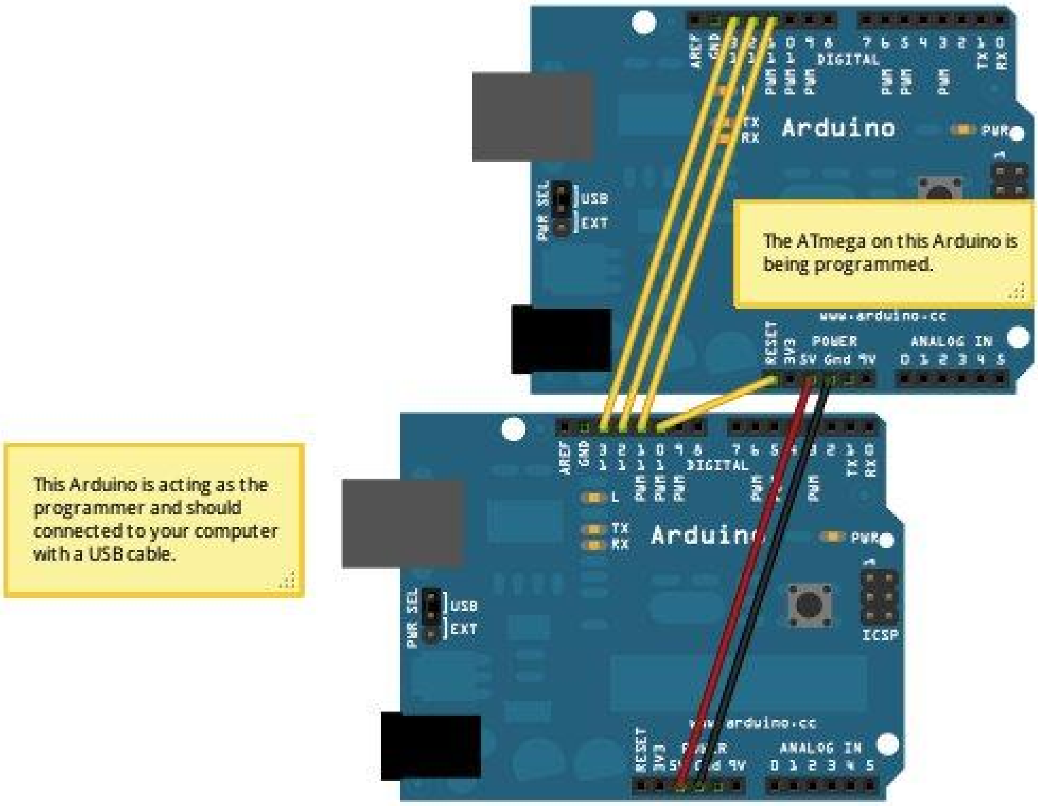
Following the schematic and Fritzing, connect the ICSP to the breadboard. The ICSP tries to program at a clock speed of 375kHz, too fast for a brand new ATmega328P. Our ICSP has a jumper connector for "slow-clock", which slows the programming speed to 8kHz.

Connect the "slow-clock" jumper, followed by connecting the USB of the ICSP to the computer.

Windows users will need to install a driver for the ICSP located on the manufacturer's website. Mac and Linux users do not need to install drivers.

On the Arduino IDE, locate Tools › Programmer › USBasp.

To upload the Arduino bootloader, select Tools › Burn Bootloader. Once the bootloader has been installed, we can upload the blink sketch to the microcontroller by selecting Sketch › Upload Using Programmer .


The ATmega on this Arduino is being programmed.
This Arduino is acting as the programmer and should connected to your computer with a USB cable.
Arduino
DIGITAL
POWER
ANALOG IN
www.arduino.cc
ICSP
PWR SEL
USB
EXT

# TX-RX and Test LED Connection

## Tx-Led

| | | | | |
|---|---|---|---|---|
| WC0283-01 | Super Bright White | Intensity: 6,000mcd | Voltage: | 1.6v-1.9v |
| | | Colour Freq: 6000-6500 | Typical: | 1.8v |
| | | Viewing Angle: 28º | Current: | 18mA |
| | | Lens: Water Clear | | |

## Rx-Led

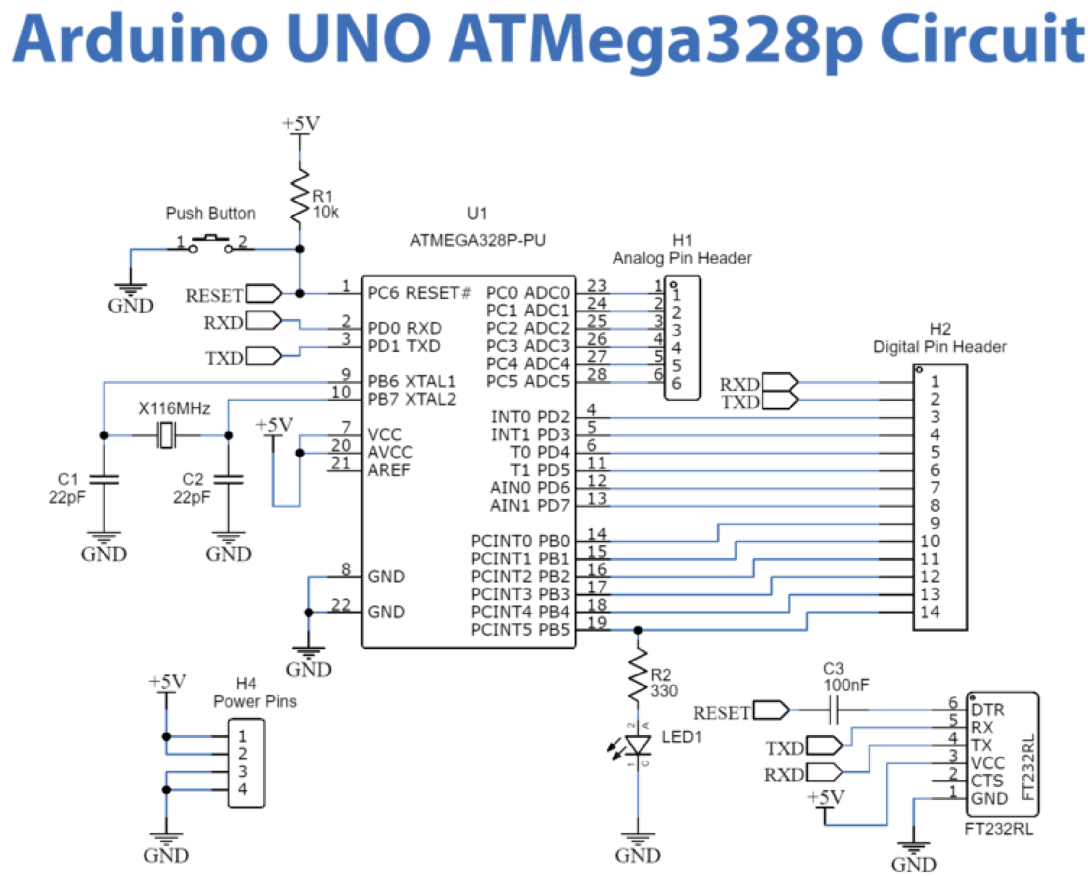
| | | | | |
|---|---|---|---|---|
| YC0283-01 | Super Bright Yellow | Intensity: 4,500mcd | Voltage: | 1.6v-1.9v |
| | | Colour Freq: 580-590nm | Typical: | 1.8v |
| | | Viewing Angle: 18º | Current: | 18mA |
| | | Lens: Water Clear | | |

## Test-Led

| | | | | |
|---|---|---|---|---|
| GC0283-01 | Super Bright Green | Intensity: 3,500mcd | Voltage: | 1.6v-1.9v |
| | | Colour Freq: 520-570nm | Typical: | 1.8v |
| | | Viewing Angle: 28º | Current: | 18mA |
| | | Lens: Water Clear | | |

A 220 ohm resistance is used with each LED
The Rx Led is connected with Pin 2 of Microcontroller
The Tx Led is connected with Pin 3 of Microcontroller
The Test Led is connected with Pin 19 of Microcontroller

## MICROCONTROLLER AS WHOLE




### Microcontroller Specifications

- IC type: AVR microcontroller
- Core size: 8-bit
- Speed: up to 20MHz
- Number of I/O: 23
- Program memory size: 32KB (16K x 16)
- Program memory type: Flash
- EEPROM size: 1KB

- **RAM size: 2KB**
- **Package: 28 PIN**
- **Supply voltage: 1.8 V - 5.5 V**
- **Manufacturer: Atmel**
- **Manufacturer part number: ATmega328P-PU**