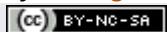


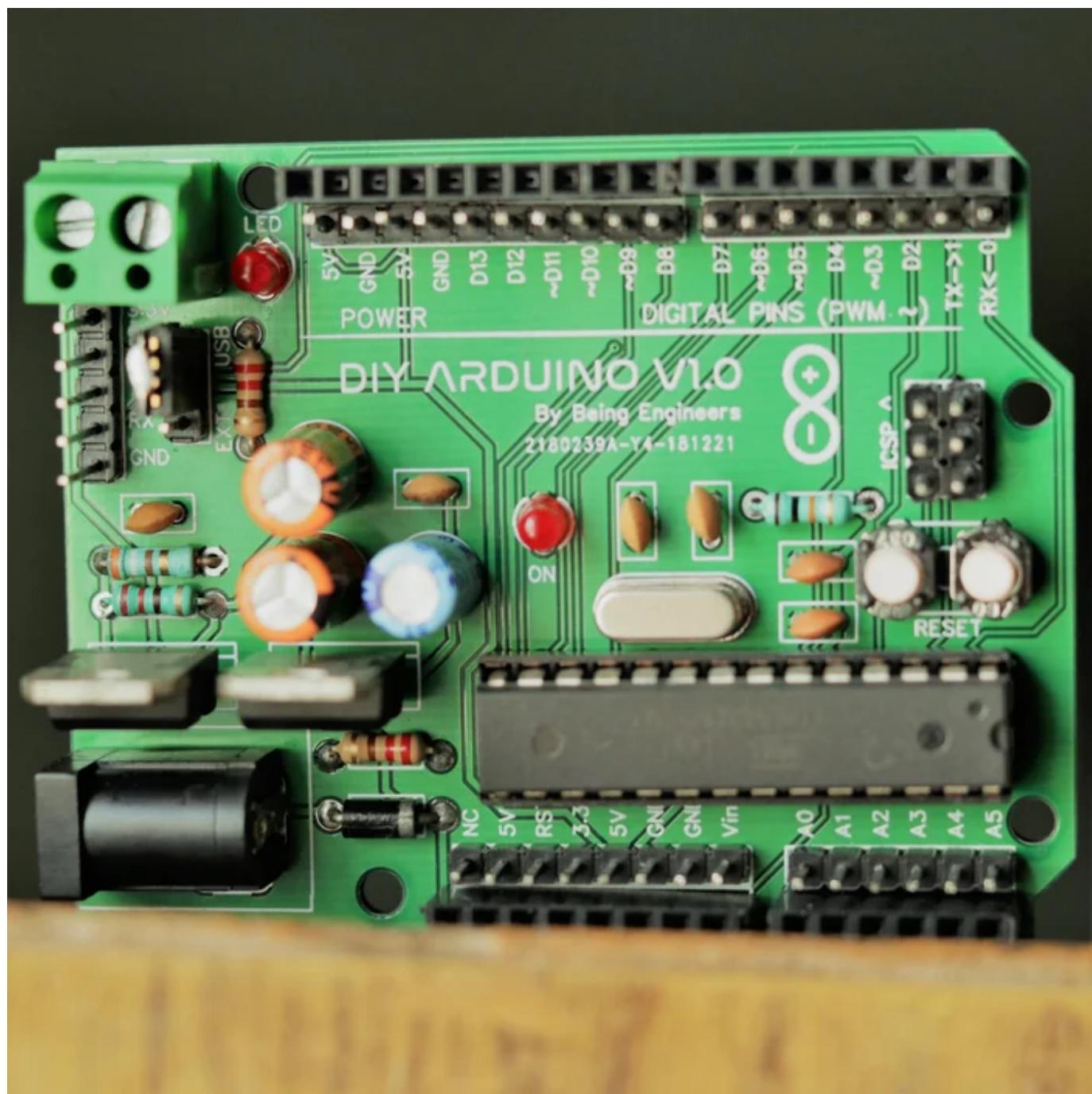
AUTODESK
Instructables

DIY Arduino UNO | How to Make Your Own Arduino Uno Board

By [Being Engineers](#) in [CircuitsArduino](#)



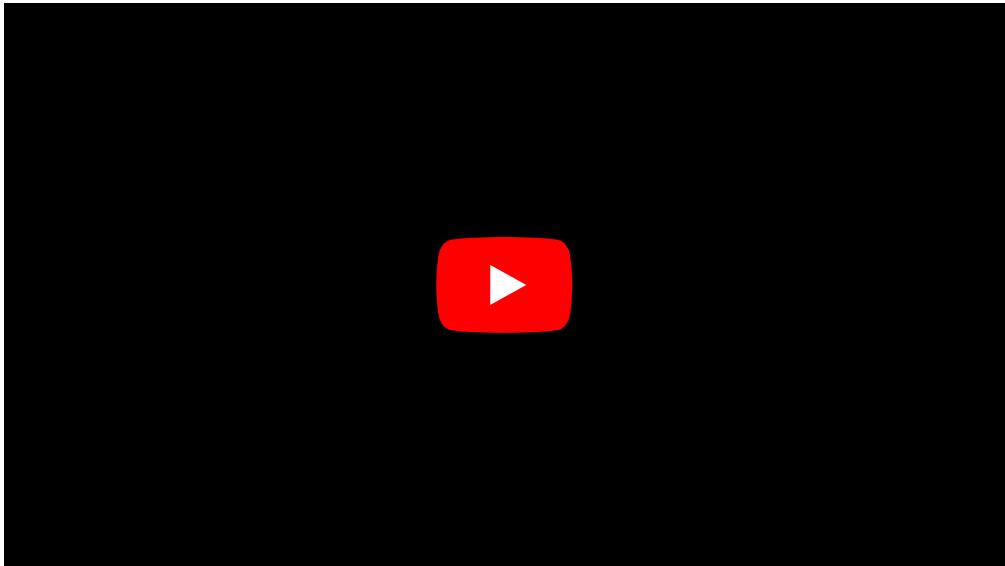
Introduction: DIY Arduino UNO | How to Make Your Own Arduino Uno Board



Hello guys, welcome to Being Engineers. Hope you all are doing great. In today's tutorial I will show you how I made my own arduino uno board using the atmega328p IC. A year ago I have

made a series of tutorials about homemade DIY arduino. This is kind of a updated version of that. Before starting this let me thank the whole arduino community for the immense support they has provided us throughout the time. Without arduino this channel might not even started at all. Now it being a open source platform it's quite easy to get to know about the insides and the bits and pieces of all the things that makes arduino what it is. So in this tutorial we will be looking into the schematic of Arduino Uno, modify it a little to our needs, make a PCB out of it and solder the required components to make the final product. But before starting let me tell you that I will not be using any SMD components to make my version of arduino because not everybody has a soldering station and sometimes, specially here where I live, finding SMD components is harder. Moreover through hole components are cheaper than SMD components in most of the cases.

I have also made a detailed video discussing the whole process in youtube, so if you don't feel like reading the whole doc here, watch that video. You will be sorted.



Also if you like this tutorial then follow us on this page "Being Engineers" and check out our youtube channel -

www.youtube.com/c/being_engineers1

Now, enough of this chit chat. Let's start the tutorial.

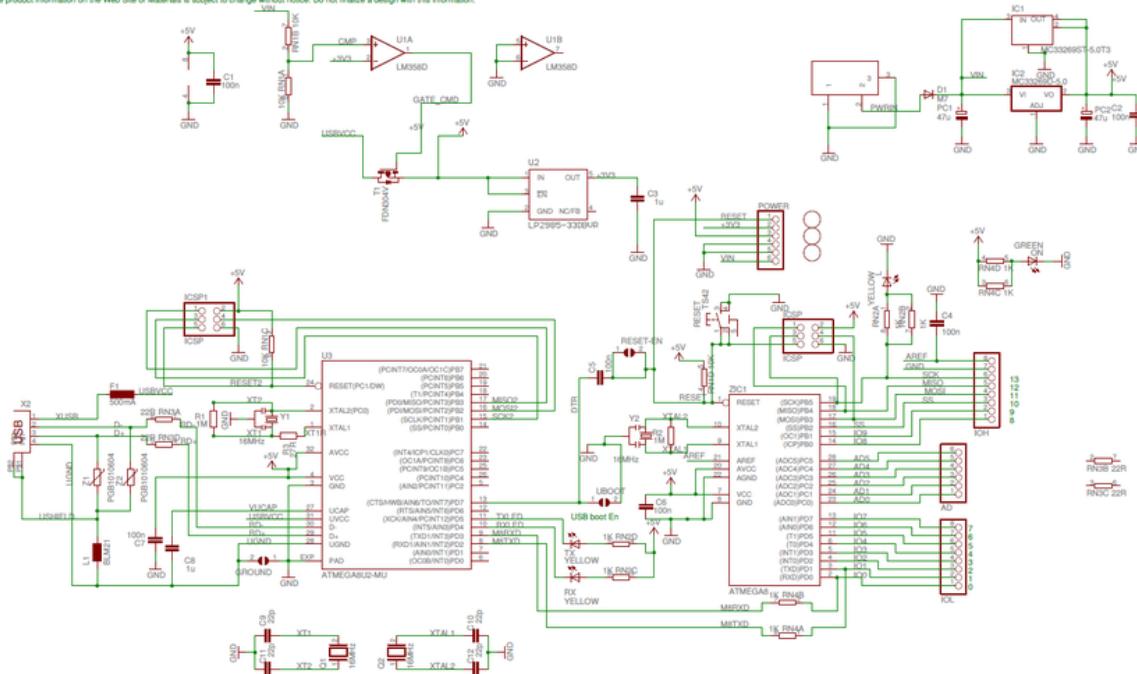
Step 1: Alter the Changes in the Original Schematic

Arduino™ UNO Reference Design

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino reserves the right to change a specification at any time. The Customer should make its own evaluation of the products before applying changes to them.

The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



First of all let's talk about the changes that I am gonna make in the original Arduino Uno schematic that you can download from the link below.

<https://bit.ly/2qrA8AC>

The changes will be the following -

- As I have told earlier, I will not be using any SMD components on board. All the items will be in through hole format.
- I didn't find any FTDI chip in through hole format, so the USB to TTI conversion will not be done on board. A separate FTDI breakout board will be used to program the new arduino.
- Original arduino uses a mosfet comparator to determine whether we are powering up the board with a usb power supply or a DC-in. But in our version we will be manually switching that using a jumper.
- The tradition one uses a LP2985 IC from Texas Instruments to get the 3.3v power supply on board. But due to unavailability of the board in TH format, we will be using a simple linear regulator. So LM1117 should be a obvious choice here, but to keep the making cost even lower, we will be using LM317 with R1 and R2 as 240E and 390E respectively.
- The last thing I want on my board is ample number of power lines and two headers for each IO ports. So I will be placing two rows of header, one male and one female around the board which will help me to connect more number of devices directly to the arduino.

So keeping in mind these changes, we can chalk down the final list of components.

Step 2: Gather the Components Required

These are the components that you need to make this project. Wherever quantity is not mentioned consider it as one.

- Atmel Atmega328p-pu Microcontroller
- 28 pin IC base
- 16MHz crystal oscillator
- 22pF caps X 2 nos
- 100nF caps X 4 nos
- 100uF electrolytic caps X 3 nos
- 3mm red led X 2 nos
- 330E 1/4W resistor X 2 nos
- 240E 1/4W resistor X 1 nos
- 390E 1/4W resistor X 1 nos
- 10K 1/4W resistor X 1 nos
- Push Button for reset switch
- 1N4007 General purpose diode
- 7805 Linear voltage regulator
- LM317 Linear variable voltage regulator
- DC female jack
- 2 pin Screw terminal block
- lot of male and female headers

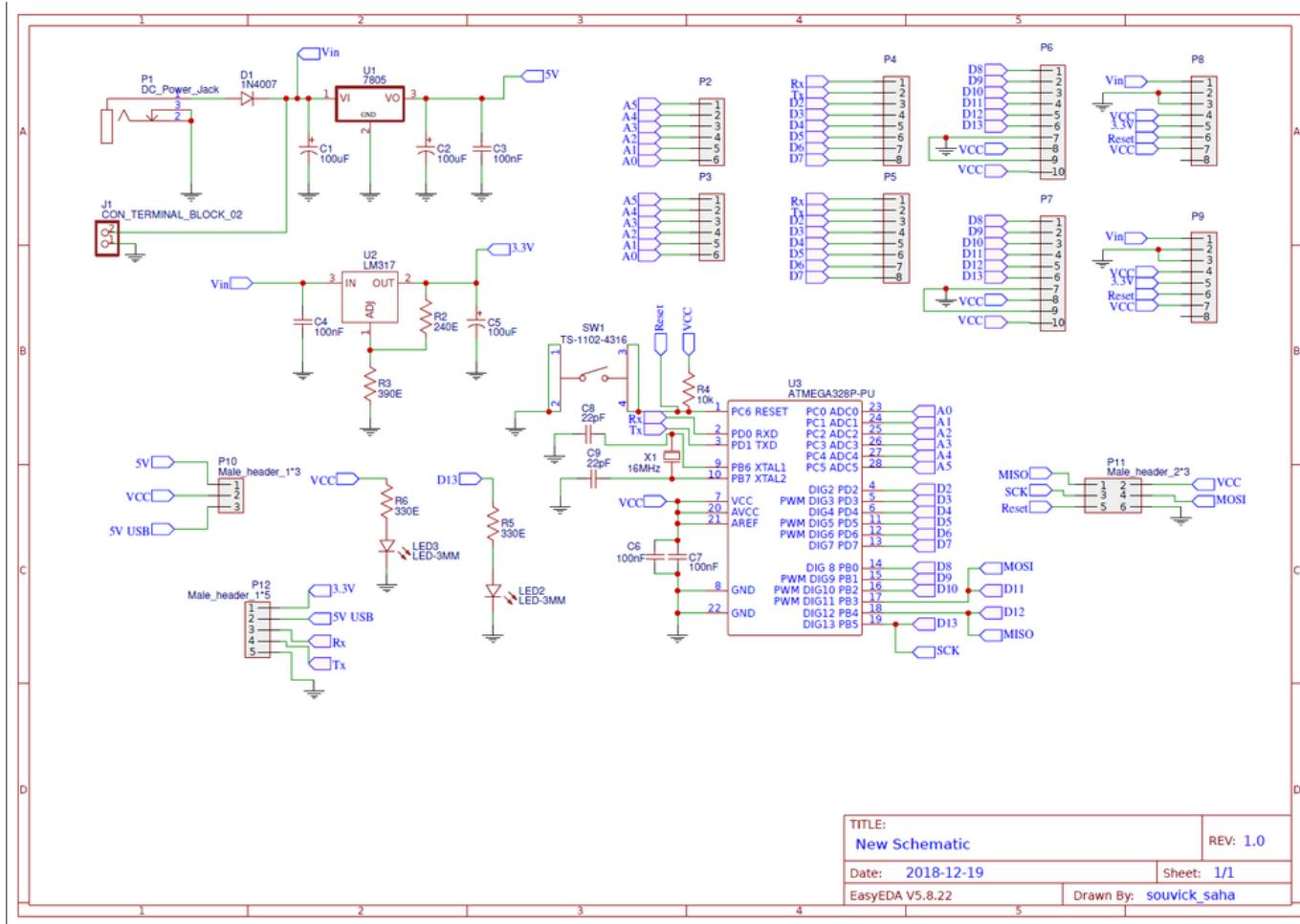
Except these you will also need soldering equipments and some hardware tools to make life easier.

You will also need a USBASP ICSP programmer or a USB to TTL converter a.k.a FTDI breakout board to program the arduino from your computer.

Here is the Project BOM from easyeda - <https://bit.ly/2R5YpvR>

Gather all these items and proceed to next step.

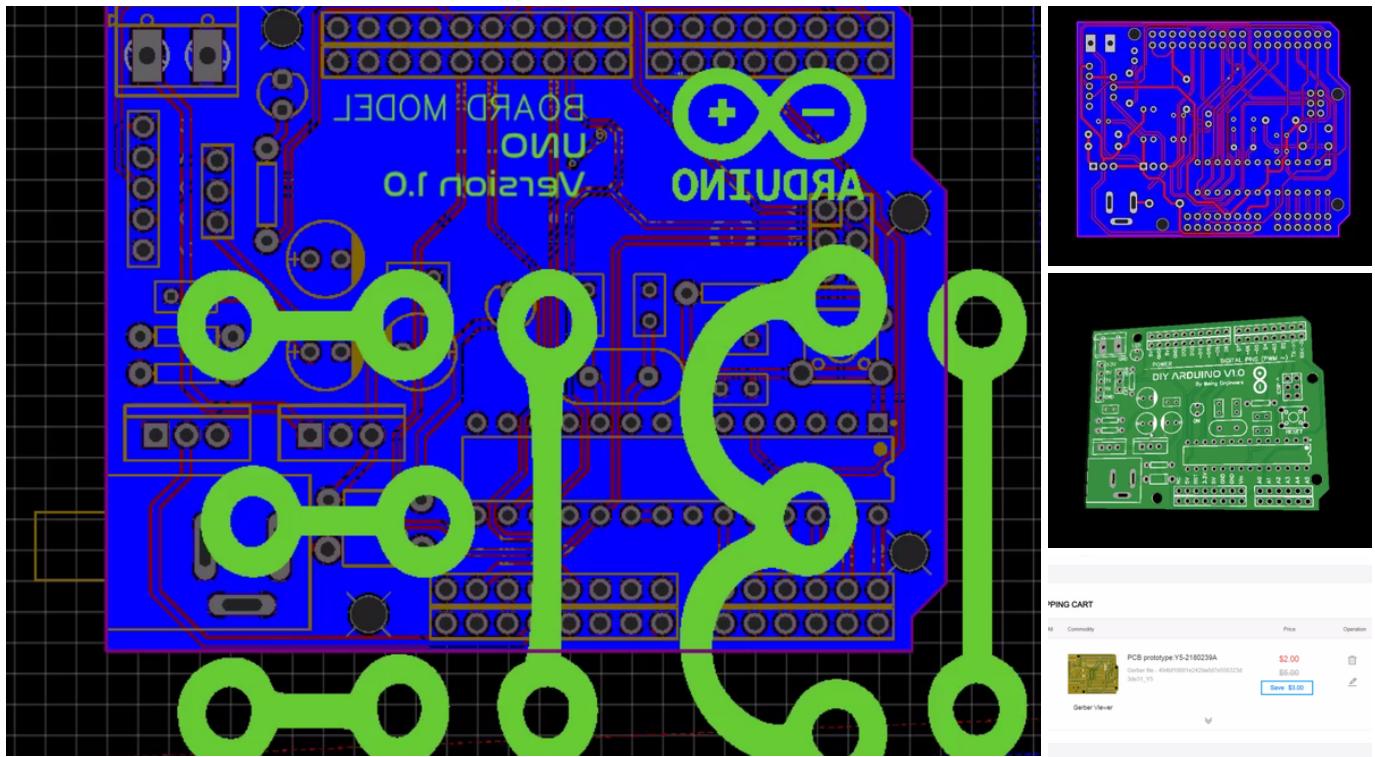
Step 3: Draw the Final Schematic



To draw the final schematic I used Easyeda, a web-based EDA tool suite. Drawing big circuits like this can be very easy in this portal. Also it's a online service. So by usability it gets nothing better. I will also recommend you to use it in your design. The circuit that I designed can be found in the link below. That is a PDF doc. Download and take a look if you want.

Circuit diagram - <https://bit.ly/2LNCDqT>

Step 4: Design the PCB and Order It



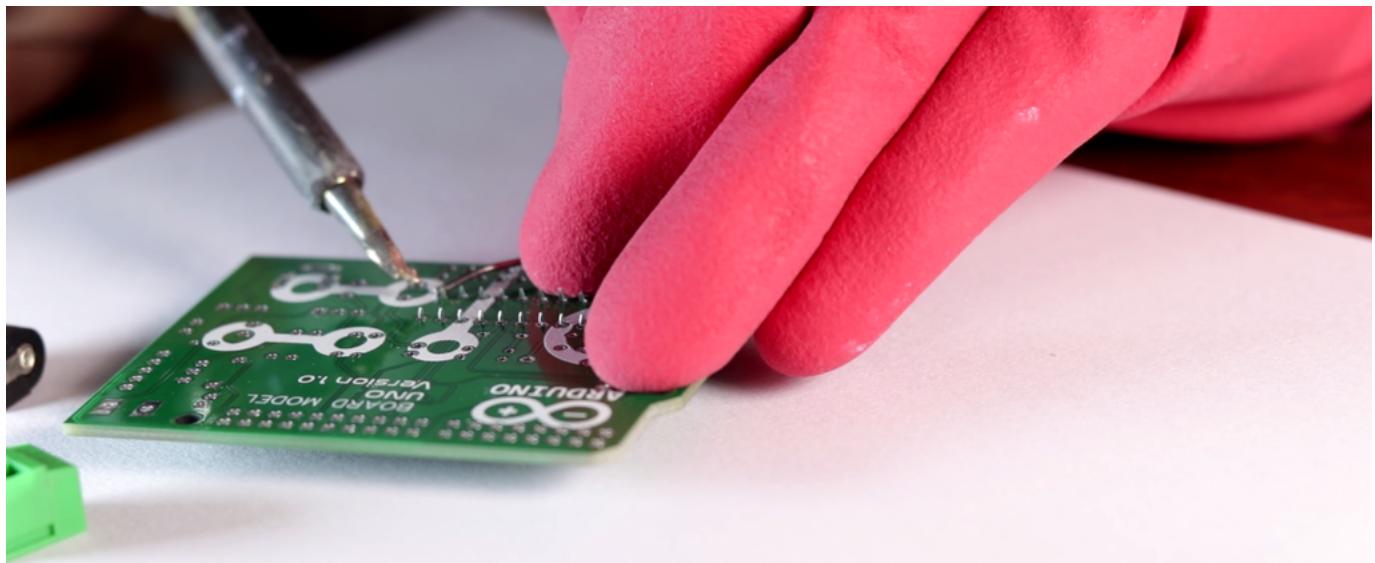
Once the circuit design is completed, it's time to make the PCB. I used the [JLCPCB](#) website to make my prototype board. They are one of the best in PCB making in recent days in my opinion.

After circuit design is complete, convert the circuit to PCB and design the PCB in [easyEDA](#) website. Be patient with it. A mistake here will ruin your PCBs. Check multiple times before generating gerber file. You can also check the 3d model of your PCB from here. Click on make gerber file and from there you can directly order this board via [JLCPCB](#). Upload the gerber files, select proper specification, don't change anything in this section. Keep it as it is. These are good enough settings to start with. Place the order. You should get it in a week.

PCB Gerber file - <https://bit.ly/2AsFzox>

PCB Layout in PDF with 1:1 Scale - <https://bit.ly/2VjJY5U>

Step 5: Solder the Components on the PCB

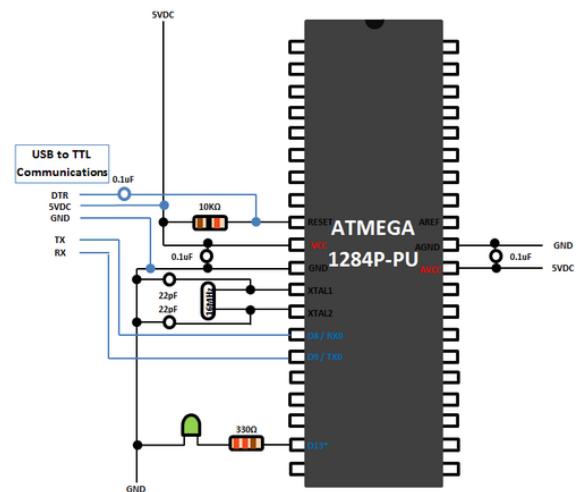


Once you have received the PCBs, it's time to solder the components on it to make the final product. There is nothing complicated in it. Just keep a printout of the schematic in front of you and start putting the components one by one in the PCB. Make sure there is no short in power and ground after the completion of this step.

One thing I just want to clear here is that the caps values are don't need to be perfect. Something close to those values will do the job. Same goes for the resistors. But keep the R1 and R2 values of LM317 as it is told.

One thing you may find odd that the arduino that I made has two reset buttons on it. Actually what happened is that when I designed the layout, I used a four pin push button as a reference. But at the time of soldering I realized that I don't have that. So I soldered 2 two pin reset switches on place. Nothing is special in there.

Step 6: Burn the Bootloader on the Microcontroller



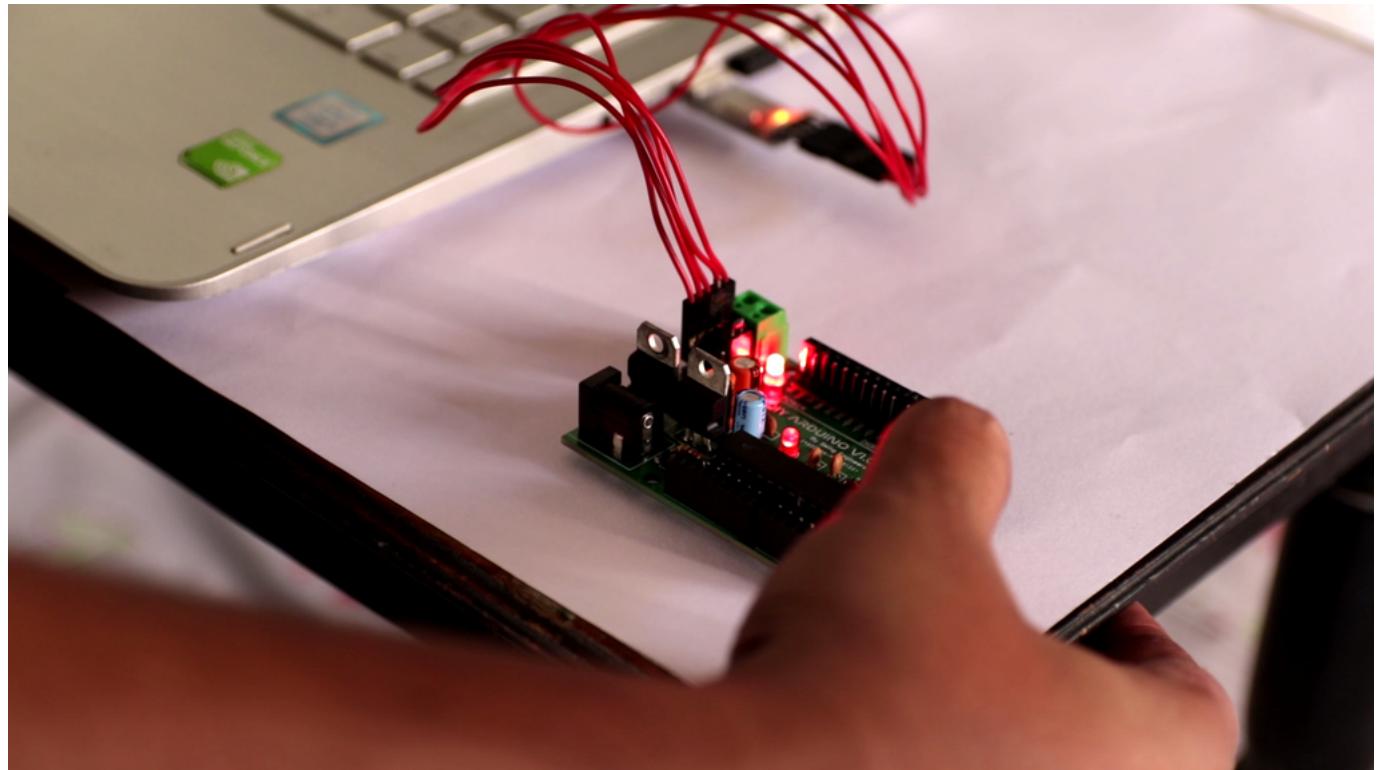
If you use a USB to TTL converter to program the microcontroller then the arduino bootloader has to be installed in the new atmega328p chip. To do that follow the steps as stated in the following page. After this the process of uploading the code is exactly same as it would be in a regular arduino. See the next step.

<https://bit.ly/2Qj1gwq>

If you use ICSP programmer i.e. USBASP programmer then this step is not needed. But the process of uploading the code is little different. Watch this video to know more.

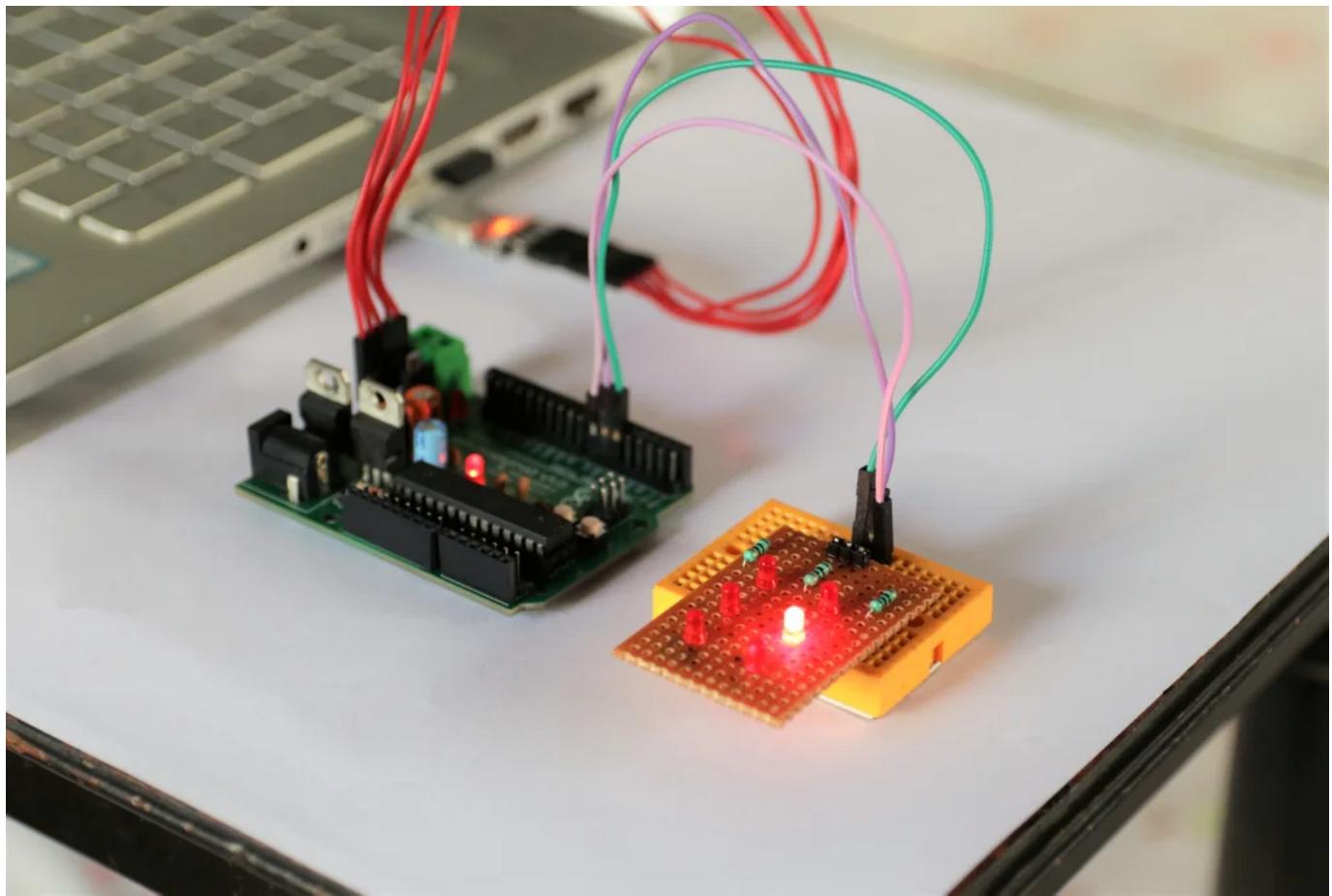
<https://bit.ly/2QybptU>

Step 7: Program the Arduino (using FTDI Breakout Board)



Connect the breakout board to the arduino and connect that to the computer. Open device manager and observe the com port of the usb to ttl converter. In arduino IDE select the com port and board correctly. Now here comes a tricky part. If your ftdi board das a DTR pin and it is connected to reset then just save the program and upload that to the arduino as usual. There will be no error. But if you don't have the DTR pin like me, then before clicking on upload, hold down the reset button on the board, and then press upload. Hold the button till the program is compiling, when the ide says 'uploading' , then release the reset switch. Then the code will be uploaded.

Step 8: It's Done!



Here you can see I have uploaded a 3 pin charlieplexing code in the new arduino and everything is working as it was supposed to do. Using only 3 pins we are controlling 6 leds with 200ms gap in between. I have tested with other codes as well and all of those work flawlessly.

That's it. The project is completed hereby. I hope you liked this project and learned something new. If you have then don't forget to follow us in this page and also subscribe to our youtube channel.

Channel link - www.youtube.com/c/being_engineers1

Stay creative and see you next time. Till then bye and take care :)