

[Videos](#)[Tutorials](#)[Support](#)[Contact](#)[Blog](#)[About](#)

## Arduino on a Breadboard

[Home](#) / [Arduino Projects](#) / [Arduino on a Breadboard](#)[< Previous](#) [Next >](#)

## Arduino on a Breadboard

Dear friends welcome back. Today we are going to build an Arduino on a breadboard. It is very easy and it won't take us more than five minutes.

### Arduino Uno (ATMEGA328P) on a breadboard Tutorial DIY proje...



#### Free Project Articles

Get the latest Arduino, Raspberry Pi and electronics project articles first.

[Join Now](#)

100% Privacy. We don't spam.

## Intro to the Arduino on a Breadboard Tutorial

I can hear you ask: Why to bother when we can buy an Arduino Pro Mini for 2 euros? There are two reasons for that. The first one is that by building our own Arduino board we can understand better the hardware of all Arduino boards and how that works. After building your own Arduino you will have a deeper understanding well the Arduino software and hardware. The second reason to build a bare Arduino board is that the power consumption of that board is minimal. Have you imagined running your Arduino projects on batteries for months or even years? With this Arduino bare board that's possible. But let's start.

## WHERE TO BUY

1. Atmega 328 with Arduino Bootloader: <https://educ8s.tv/part/ATMEGA328P>
2. 2x 22pF capacitors: <https://educ8s.tv/part/CeramicCapacitors>
3. 1x 100nF capacitors: <https://educ8s.tv/part/CeramicCapacitors>
4. A USB to serial converter module: <https://educ8s.tv/part/FTDI>
5. Multimeter Mastech 8268: <https://educ8s.tv/part/Multimeter>
6. A resistor: <https://educ8s.tv/part/Resistors>
7. LED: <https://educ8s.tv/part/LEDs>
8. AA battery holder: <https://educ8s.tv/part/BatteryHolderAA>
9. Small Breadboard: <https://educ8s.tv/part/SmallBreadboard>
10. Wires: <https://educ8s.tv/part/Wires>



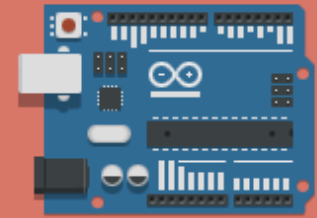
Full disclosure: All of the links above are affiliate links. I get a small percentage of each sale they generate. Thank you for your support!

[adsense]

The parts that are required are these: an ATMEGA328P microcontroller with the Arduino bootloader. It is that chip here. It costs around 2-3 euros. We also need two 22pF capacitors, these ones one 100nF capacitor, this one here a crystal at 16 MHz, some wires a USB to serial converter module this one is using an FTDI chip We're going to need a resistor and of course an LED if we want to use the standard Blink program.

So let's begin by building our own Arduino on a breadboard. Let's start. First of all, we insert the microcontroller chip in the breadboard. Be careful to insert it correctly. The marking, that little dot here, you can see it goes to the left. Now I'm going to place a sticker on it, in order to be easier to connect it with their other parts. OK, I have now put a sticker on the IC and we are ready to start connecting it. You can find a file in the description of the video that you can print and make your own stickers. That will make your life a bit easier. First we connect the crystal. The crystal goes to pins 9 and 10 that are marked as X1 and X2 at the sticker. So the crystal goes here. Next we connect a 22pF capacitor at pins 8 and 9 labeled as Ground and X1. So X1 and Ground. The capacitor goes here. Next we connect another 22pF capacitor between pins 8 and 10. Next, we have to connect Vcc Vcc is pin 7. Now we have to connect Ground. Ground goes from pin number 8 to the breadboard Ground. Now, let's connect the LED. In order to use the standard blink program, we have to use the digital pin 13. Digital pin 13 is here and the resistor goes here. Now, the positive leg of the LED goes this way and all we have to do is to connect the other pin to the breadboard Ground. okay, we're now ready to power this projects up. We're going to power it from this little power bank and as you can see if I plug in the Ground to the breadboard our circuit started working and the LED is flashing.

The IC the integrated circuit has no program loaded when you first buy it, except the Arduino bootloader. so if you connect it you won't see the LED flashing. You have to program it first. So let's see how to program this bare Arduino board. In order to program the integrated circuit, we're going to use a USB to Serial converter and one 100nF capacitor. I have connected the capacitor to pin 1 which is pin RESET of the integrated circuit. So, the first pin of the FTDI converter is named DTR. DTR pin goes to pin 1 of the integrated circuit but we have to connect it to the capacitor first. So we connect the first pin to the and the other pin of



**CHEAP  
ARDUINO**

**\$3.99**

**GET IT NOW!**

the capacitor to the RESET pin of the microcontroller. Next, the second pin is named Rx. Rx pin goes to Tx pin of the microcontroller which is pin 3. The third pin of the converter is named TX so TX pin goes to Rx pin of the microcontroller which is pin 2. Now, all we have to do is to connect two more pins.

Pin 4 is named Vcc so Vcc goes to breadboard Vcc. And the last pin of the converter, we don't have to use the fifth pin, the last pin is Ground, so Ground goes to breadboard ground. We are now ready to connect it to the computer. I will now plug in the USB in my computer. As you can see Arduino is up and running the LED is flashing, the FTDI converter provides power to the circuit. So let's now program it to blink the LED faster. I have made a simple program that puts LED on for two seconds and off for half a second. Let's upload it I'm now pressing upload. Some LEDs with flash on the FTDI converter and now we are going to see the program running. Two seconds on ... half a second off. That's how easy it is to program the bare-board Arduino. Let's now demonstrate the big advantage of the Arduino that it is built on a breadboard. I have modified the Blink LED program. So Arduino spends eight seconds sleeping and lights up the LED for only one second.

As you can see when Arduino is sleeping as it is right now, the power consumption, the current is 0.06mA! Can you see that? Now it's flashing ... 0.06 mA. And when Arduino puts LED on the current rises up to 36 or 37 mA. The power consumption when Arduino is sleeping is that low that we can power the bare-board Arduino from this power bank for over 4.5 years! yes, that's correct we can make an Arduino run on batteries for years! A lot of projects can benefit from that. Imagine a weather station that can run on batteries for months. Or even years!

I am really excited by this development as it increases the potential of the Arduino platform drastically. That's today's video I hope that you like it and that it is useful for you. Many future projects are going to be built around this bare Arduino because of its extremely low, power consumption. You can find the code of the project and the file for the stickers below.

---

## CODE & STICKERS

---

