

Atmega328P Standalone

Table of Contents

| | |
|---|----|
| Description | 2 |
| Components | 2 |
| Bootloader | 3 |
| Programming | 5 |
| Atmega Minimal Configuration | 7 |
| 5V Power Supply for your own Atmega Board | 8 |
| My Atmega Board..... | 9 |
| References..... | 10 |

Description

Atmega328 standalone means that you use the Atmega328 IC on a project without using an Arduino board.

Arduino Uno is basically a breakout board for Atmega328 μC for making the use of the μC beginner friendly.

The benefits of Atmega328 standalone are that you can make your project compact because the Atmega and the necessary components are in total smaller than the Arduino Uno board.

In advance Atmega standalone will help you reduce the power consumption of your project because you are not using extra components like leds. Also you can have a working μC with less components.

Components for Standalone Atmega328

1x Atmega328P IC

1x 16MHz Crystal

2x 22pF Capacitors

1x Button

1x Arduino Board with the Atmega chip on it to burn the bootloader

1x Arduino Board without the Atmega ic or a ftdi board to program the chip.

If you use a ftdi board you would need also a 0.1 μF capacitor

1x Breadboard or prototype board (for building your own "Arduino" board)

Not necessary parts (for making your own board with regulated 5v power supply)

Pin headers (female or male)

1 x 100 μF Capacitor

1 x 10 μF Capacitor

1 x 1N4001 Diode

1 x L7805 Voltage Regulator

Bootloader

First and foremost you need to burn the Arduino bootloader to your atmega so you can program it using Arduino IDE. If you have removed your atmega from an working Arduino board the atmega already have the Arduino bootloader. If your atmega is new or for some other reason the bootloader is not there you can burn the bootloader easily using an Arduino as in system program (ISP).

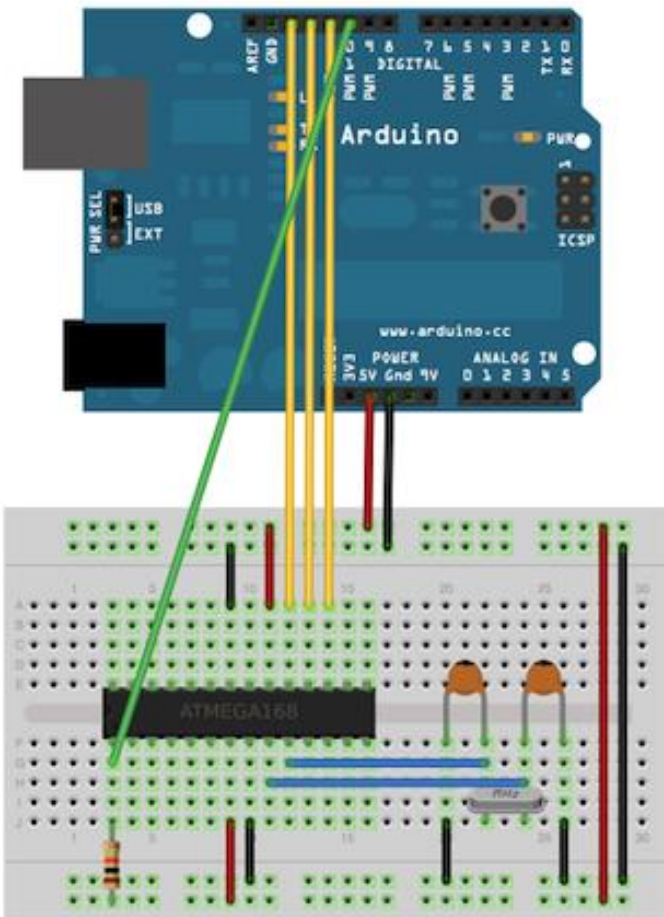
Without burning the bootloader you cannot upload code to your Atmega using Arduino IDE you will need a programmer to upload your code.

If you want to check if you have bootloader onto your chip you can try to upload code to it. If your chip does not have a bootloader you will get error while uploading.

Steps to burn Arduino Bootloader using an Arduino board

- 1 To start you need to open Arduino IDE and select the correct board and serial port from tools menu.
 - 2 Next you should upload ArduinoISP sketch on a working Arduino from Arduino IDE's examples
 - 3 Then you need to wire the μ C and the Arduino Board
- Arduino pins 13, 12, 13 to the equivalent pins on atmega and Arduino pin 10 to atmega's reset pin. Also you need to wire 5V and GND.
- 4 Select Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu.
 - 5 Select "Arduino as ISP" from Tools > Programmer
 - 6 Then burn bootloader from tools menu.

After that, you can remove the wires from pins 10, 11, 12, 13.



Connections for burning the bootloader to Atmega chip. Photo from official arduino website.

| Arduino function | | | | | Arduino function |
|---------------------|--------------------------|----|----|------------------------|----------------------|
| reset | (PCINT14/RESET) PC6 | 1 | 28 | PC5 (ADC5/SCL/PCINT13) | analog input 5 |
| digital pin 0 (RX) | (PCINT16/RXD) PD0 | 2 | 27 | PC4 (ADC4/SDA/PCINT12) | analog input 4 |
| digital pin 1 (TX) | (PCINT17/TXD) PD1 | 3 | 26 | PC3 (ADC3/PCINT11) | analog input 3 |
| digital pin 2 | (PCINT18/INT0) PD2 | 4 | 25 | PC2 (ADC2/PCINT10) | analog input 2 |
| digital pin 3 (PWM) | (PCINT19/OC2B/INT1) PD3 | 5 | 24 | PC1 (ADC1/PCINT9) | analog input 1 |
| digital pin 4 | (PCINT20/XCK/T0) PD4 | 6 | 23 | PC0 (ADC0/PCINT8) | analog input 0 |
| VCC | VCC | 7 | 22 | GND | GND |
| GND | GND | 8 | 21 | AREF | analog reference |
| crystal | (PCINT6/XTAL1/TOSC1) PB6 | 9 | 20 | AVCC | VCC |
| crystal | (PCINT7/XTAL2/TOSC2) PB7 | 10 | 19 | PB5 (SCK/PCINT5) | digital pin 13 |
| digital pin 5 (PWM) | (PCINT21/OC0B/T1) PD5 | 11 | 18 | PB4 (MISO/PCINT4) | digital pin 12 |
| digital pin 6 (PWM) | (PCINT22/OC0A/AIN0) PD6 | 12 | 17 | PB3 (MOSI/OC2A/PCINT3) | digital pin 11(PWM) |
| digital pin 7 | (PCINT23/AIN1) PD7 | 13 | 16 | PB2 (SS/OC1B/PCINT2) | digital pin 10 (PWM) |
| digital pin 8 | (PCINT0/CLKO/CP1) PB0 | 14 | 15 | PB1 (OC1A/PCINT1) | digital pin 9 (PWM) |

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Atmega's equivalent pins to Arduino's

Programming the Atmega

1st Way:

The simplest of all

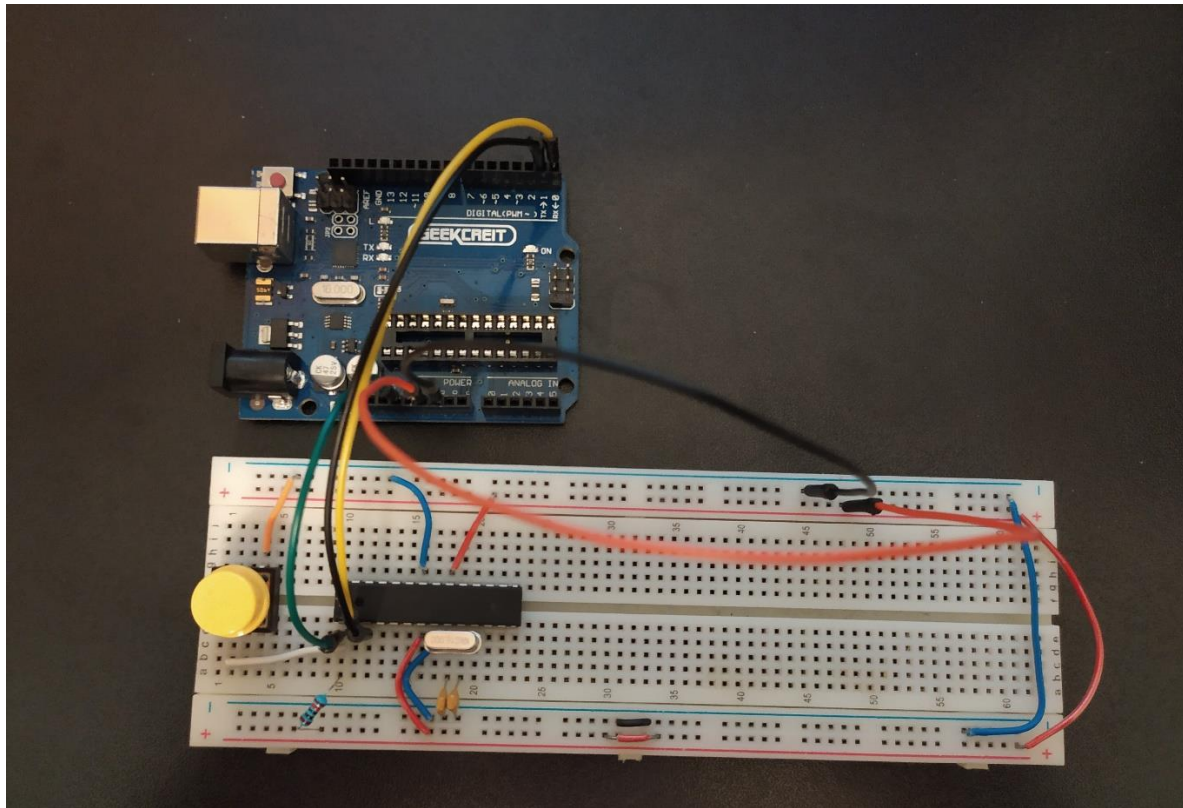
Put your Atmega chip back to an arduino board.

Upload the code and the remove it and put it back on your breadboard or your project.

2nd Way:

Using the Arduino's USB-to-serial convertor (FTDI chip)

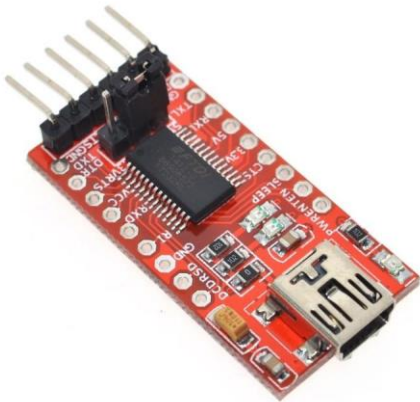
First you remove the μ C from the arduino board then you connect Arduino's board RX,TX and Reset pins to the corresponding pins on the Atmega and you upload your code as you do on an Arduino.



Programming my Atmega on breadboard using Arduino board.

3rd Way:

Using an external FTDI like the one below.



In order to program your Atmega with a FTDI Programmer it is mandatory that your Atmega has a bootloader on it. Go to page 3 of this guide to read about burning the bootloader.

First of all if your FTDI has a jumper to choose between 5V and 3.3V (like the board in the photo above) select 5V because Atmega328 runs on 5V.

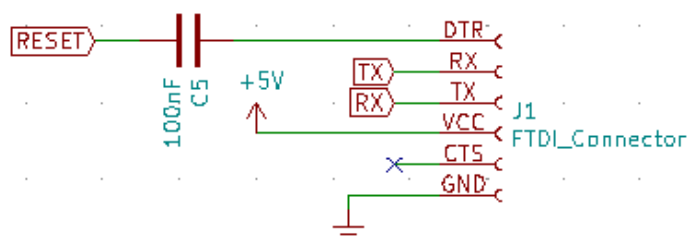
Then you have to make the connections between FTDI and Atmega.

You have to connect DTR pin with Atmega's Reset pin but it is very important to put a 100nF capacitor in series, between DTR pin and Atmega's Reset Pin. Then you have to connect FTDI's RX to Atmega's TX and FTDI's TX to Atmega's RX. Last but not least you have to connect VCC and GND to Atmega's VCC and GND pins.

You do not have to connect FTDI's CTS pin, let it floating.

Make the connections as shown in the schematic below.

FTDI Programmer (Using Arduino)



Your Atmega board will be powered from the FTDI so you do not need external power source.

Then you plug FTDI to your computer via USB cable and now you can upload your Arduino code as usual. Remember to select the right port and the correct board (Arduino Uno or Atmega328p).

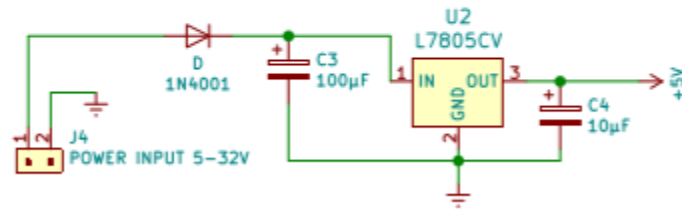
Atmega Minimal Configuration

If you do not have an 16MHz Crystal and the 22pF capacitors you can eliminate the external clock and use the internal 8MHz clock.

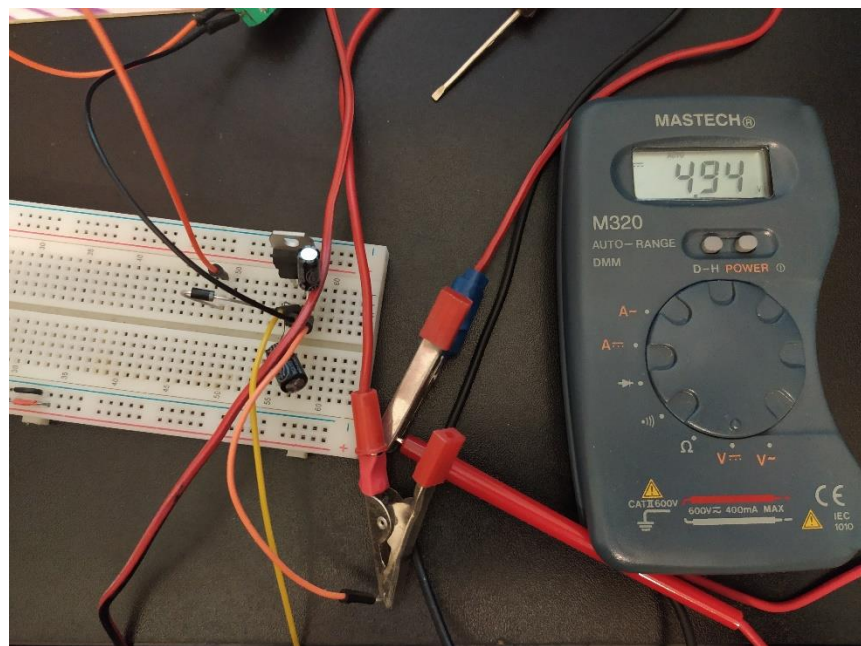
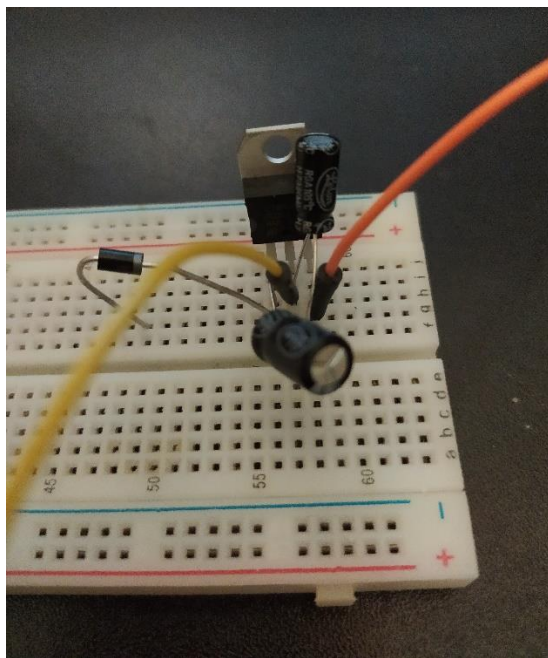
I will not cover this in the tutorial but you can read more at the official Arduino site <https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard>

5V Power Supply for your own Atmega Board

Power Supply



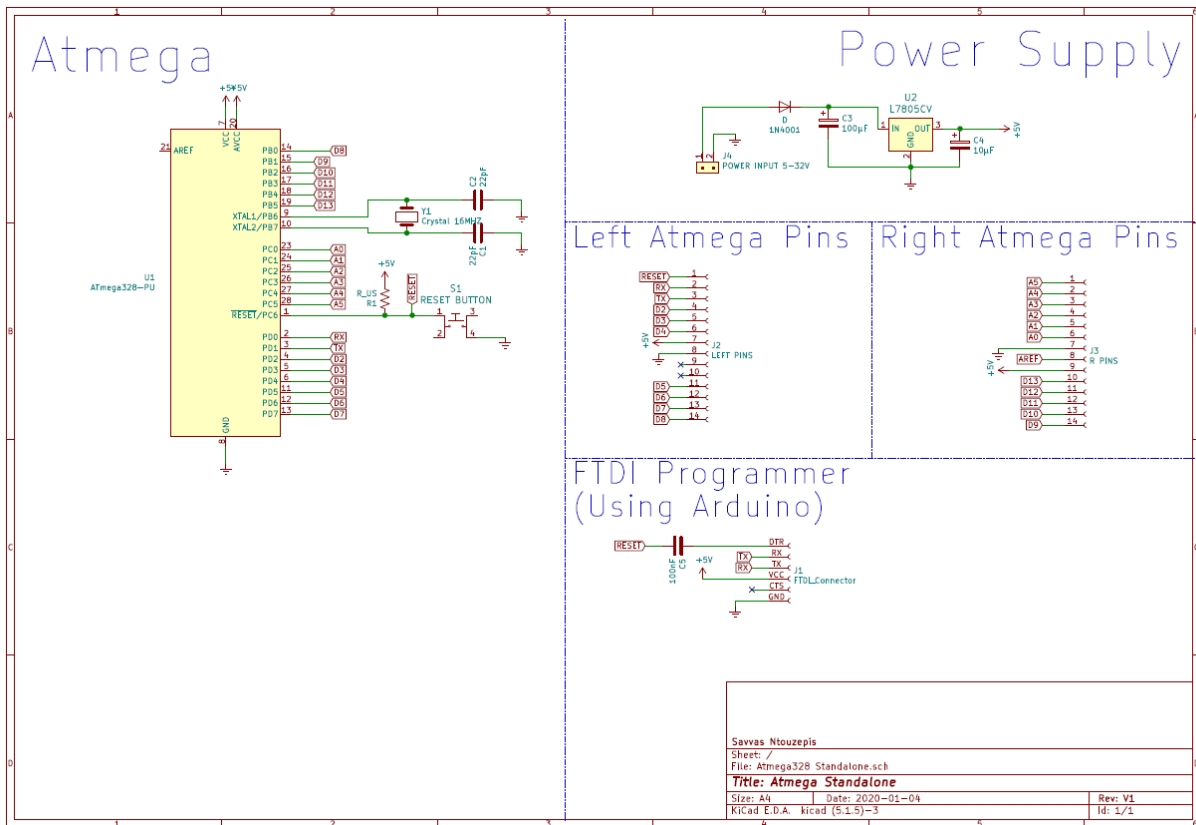
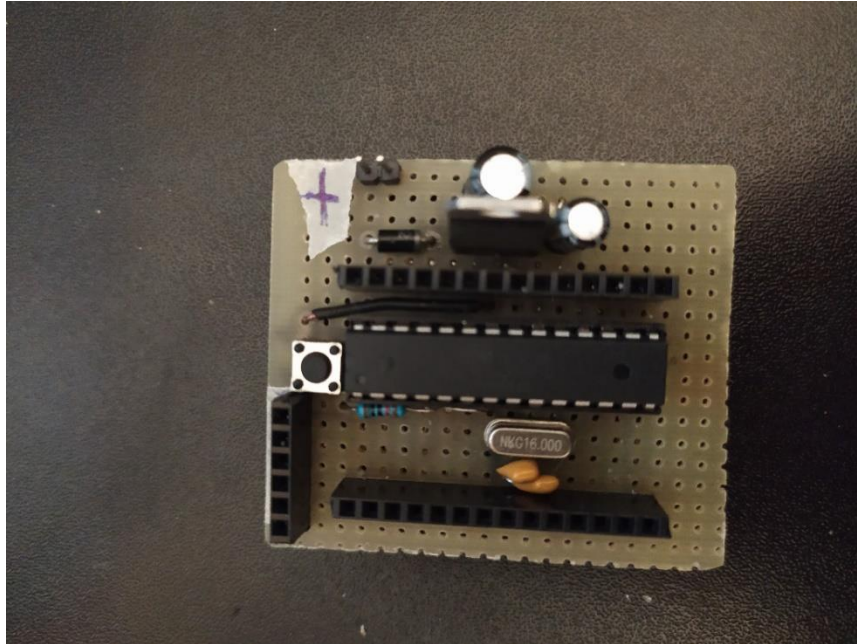
I am using a L7805 voltage regulator to drop the input voltage (5-32V) to 5V to power our board. I used two capacitors one at the input and one at the output to smooth the voltage at the output and make it stable (as the datasheet suggests). Also I used one 1N4001 diode for a basic reverse polarity protection.



The circuit on a breadboard and the output measurement on a multimeter.

My Atmega328P board

My Atmega328P board and the complete schematic. You can also find the schematic at my gitlab



References

Arduino Web Site

<https://www.arduino.cc/en/Tutorial/ArduinoToBreadboard>

L7805 Datasheet

<https://www.st.com/resource/en/datasheet/l78.pdf>

Made by Savvas Ntouzepis

E-mail: sdouze@gmail.com

GitHub: <https://github.com/sdouze>