

ATmega Board Documentation

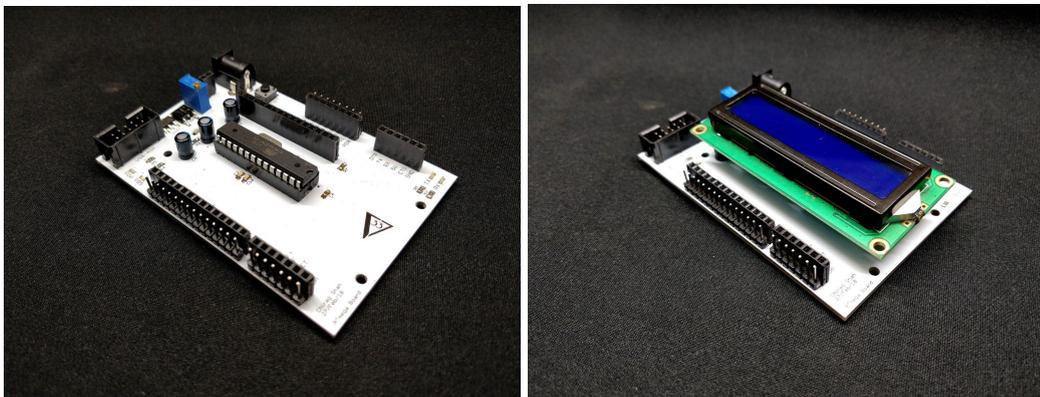
Objective	2
Features	2
Overview of using the board	3
How to upload the Blink code for the first time	3
Installing Atmel Studio	3
Installing the drivers for USBasp	4
Option 1: With Zadig	4
Option 2: Without Zadig	4
USBasp programmer	5
Programming using SinaProg	5
Programming with Arduino	6
Steps to program using Arduino IDE	6
Steps to burning the bootloader	6
Serial Communication	7
Installing FTDI driver	7
Fuse settings on a new ATmega 328p	8
ATmega to LCD mapping	8
ATmega board to Arduino mapping	8

Objective

The objective of this document is to give an overview of the ATmega board, what all software installations and pin configurations one should be aware off so that one can get started with using the board.

Features

1. ATmega 328p microcontroller
2. Serial (Arduino) as well as ISP programmable
3. Onboard 3.3v regulator
4. Onboard 5v regulator
5. Input voltage range: 7-20v using the DC Jack
6. Port for and onboard 16x2 LCD



You can get the PCB design on https://github.com/chirags98/ATmega_Board

You can get the example codes for Atmel Studio on <https://github.com/chirags98/ATmega328-Example-Codes>

Using the board

General instructions to use the board after the initial setup is complete are (detailed first-time instructions given below)

1. Write your code in Atmel studio (or use the example codes)
2. Build the solution to generate the hex file
3. Program the hex file on the board using USBasp programmer and SinaProg software (USBasp driver should be installed on the computer).

If you want to program the microcontroller using Arduino IDE, then

1. Write your code in the Arduino IDE
2. Select Arduino Nano and the COM port
3. Upload the code on the board using FTDI USB to Serial converter

Connect the FTDI directly to the header on the board ensuring that the pinout of the FTDI matches with the pinout on the board. Rx goes to Tx and Tx goes to Rx.

ATmega microcontroller should have the Arduino Nano bootloader on it. FTDI drivers should be installed.

Uploading the Blink code for the first time

1. Download the example codes from [here](#)
2. You will need to download and install Atmel studio for writing new codes and to update existing codes ([Procedure](#))
3. Disable the device driver signature enforcement and install the drivers for USB-ASP ([Procedure](#))
4. Download SinaProg to upload the hex file to the microcontroller ([Procedure](#))
5. Check the USB-ASP jumper ([Procedure](#))
6. Upload the hex file. The hex file will be located in Blink>Blink>Debug>Blink.hex ([Procedure](#))

Installing Atmel Studio

Atmel Studio is the recommended IDE for embedded C programming for AVR microcontrollers.

Download and install the latest version from

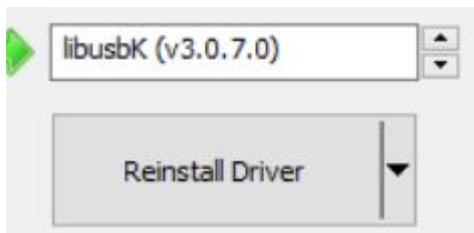
<https://www.microchip.com/avr-support/atmel-studio-7>

While installing only select support for 8-bit AVR microcontrollers.

Installing the drivers for USBasp

Option 1: With Zadig

1. Download Zadig software from http://chiragrshah.com/zadig_download
Or from <https://zadig.akeo.ie/>
2. Unzip and run zadig-2.4.exe
3. Click on Options>list all device
4. Select USBASP from the list
5. Select and install libusbK driver



6. If this does not work try other drivers than libusbK

Option 2: Without Zadig

The drivers for USPasp programmer are not digitally signed thus for security purposes windows has blocked the installation of unsigned drivers.

Get the USPasp drivers from

<https://drive.google.com/open?id=1goBblAIIR3vfRTkgHP2Q47yjqxCTrpaF>

To disable the Driver Signature Enforcement for installing and using the drivers follow the steps given below

1. Recovery
2. Advanced startup>Restart Now
3. Troubleshoot
4. Advanced Options
5. Startup Settings
6. Restart
7. Disable Driver Signature Enforcement (press 7)

Use the steps given below for installing the drivers

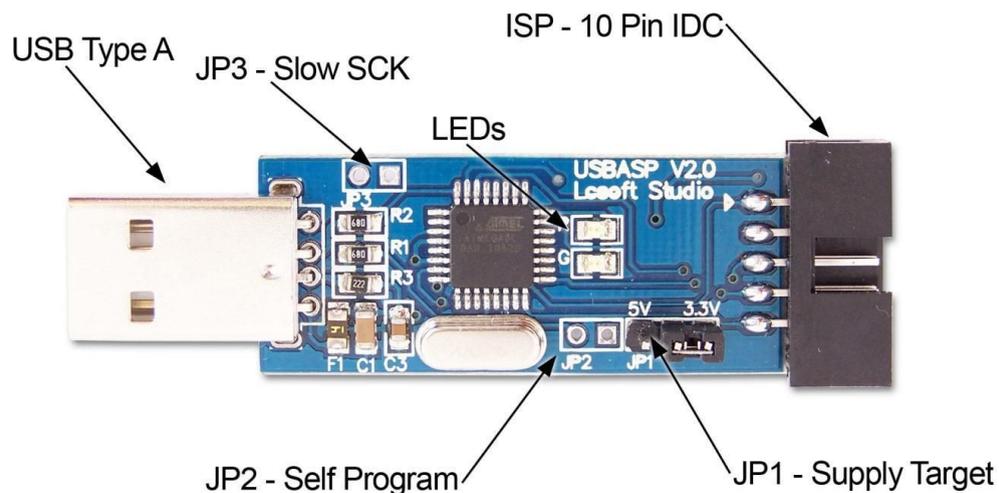
1. Device Manager
2. USBasp

3. Update Driver
4. Browse my computer for driver software
5. Navigate and select the libusb_0.1.12.1 folder. (use libusb_0.1.10.1 if this doesn't work)

If the programmer stops working occasionally then you might need to disable the drivers again.

USBasp programmer

1. Make sure the “Slow SCK” jumper is removed if available on your programmer
2. Make sure the switch/jumper is set to 5v and not 3.3v
3. Make sure the drivers are installed before trying to use the programmer



Programming using SinaProg

Download SinaProg from

http://chiragrshah.com/download_sinaprogram

1. Open the folder and run the application
2. Navigate to and select the .hex file of the project
3. Click on “Program flash”

Programming with Arduino

This board can act exactly like an Arduino uno/nano. To program it with Arduino IDE make sure the microcontroller has bootloader on it. You cannot program it using an FTDI converter if it does not have a bootloader on it. To burn the bootloader follow the steps given below.

Steps to program using Arduino IDE

1. Write your code in the Arduino IDE
2. Select Arduino Nano and the COM port
3. Connect the FTDI directly to the header on the board ensuring that the pinout of the FTDI matches with the pinout on the board. Rx goes to Tx and Tx goes to Rx.
4. Click on upload

Note: ATmega microcontroller should have the Arduino Nano bootloader on it. FTDI drivers should be installed.

Steps to burning the bootloader

If you have a new ATmega328, you'll need to burn the bootloader onto it. You can do this using an Arduino board as an in-system program (ISP).

If the microcontroller already has the bootloader on it, you can skip this section.

To burn the bootloader, follow these steps:

1. Upload the ArduinoISP sketch onto your Arduino board. (You'll need to select the board and serial port from the Tools menu that correspond to your board.)
2. Connect the pins on Arduino to the pins on atmega board as shown on the table
3. Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu.
4. Select "Arduino as ISP" from Tools > Programmer
5. Run Tools > Burn Bootloader

Pin on Arduino	Pin on target ATmega board
D13	PB5
D12	PB4
D11	PB3
D10	RESET
5V	5V
GND	GND

Option 2: Use another ATmega board instead of an Arduino

Pin on master ATmega board	Pin on target ATmega board
PB5	PB5
PB4	PB4
PB3	PB3
PB2	RESET
5V	5V
GND	GND

Further reference for this can be found at:

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

Serial Communication

If you want to interface your ATmega board to your computer via USB you will need a FTDI USB To TTL Serial Converter Adapter Module.

Your PC can communicate with the board via the serial port (also known as a UART or USART Serial). It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB.

Installing FTDI driver

FTDI USB to serial converter is required for USB to serial communication.

Download the FTDI driver from <http://www.ftdichip.com/Drivers/VCP.htm>

Fuse settings on a new ATmega 328p

If you are using a new ATmega IC you will need to program the fuse bits the first time.

The recommended settings for the fuse bits are

Lock Bits	x0F
High Fuse	xDA
Low fuse	xFF
Ext. Fuse	x05

Steps to program the fuse bits

1. Open SinaProg
2. Fuses > Advanced
3. Set the correct values and write (Try once again if you get a verification error)

References:

<http://www.engbedded.com/fusecalc>

<http://www.martyncurrey.com/Arduino-atmega-328p-fuse-settings/>

ATmega to LCD mapping

ATmega Board	LCD
PD2	D7
PD3	D6
PD4	D5
PD5	D4
PD6	E
PD7	RS

ATmega board to Arduino mapping

This section gives the pin mappings between the ATmega to Arduino board

ATmega Board	Arduino
PD0/RX	D0/RX

PD1/TX	D1/TX
PD2	D2
PD3	D3
PD4	D4
PD5	D5
PD6	D6
PD7	D7

ATmega Board	Arduino
PC0	A0
PC1	A1
PC2	A2
PC3	A3
PC4/SDA	A4
PC5/SCL	A5

ATmega Board	Arduino
PB0	D8
PB1	D9
PB2	D10
PB3/MOSI	D11
PB4/MISO	D12
PB5/SCK	D13