ATMEGA328P

by Ricky Gai

Revision 1.2
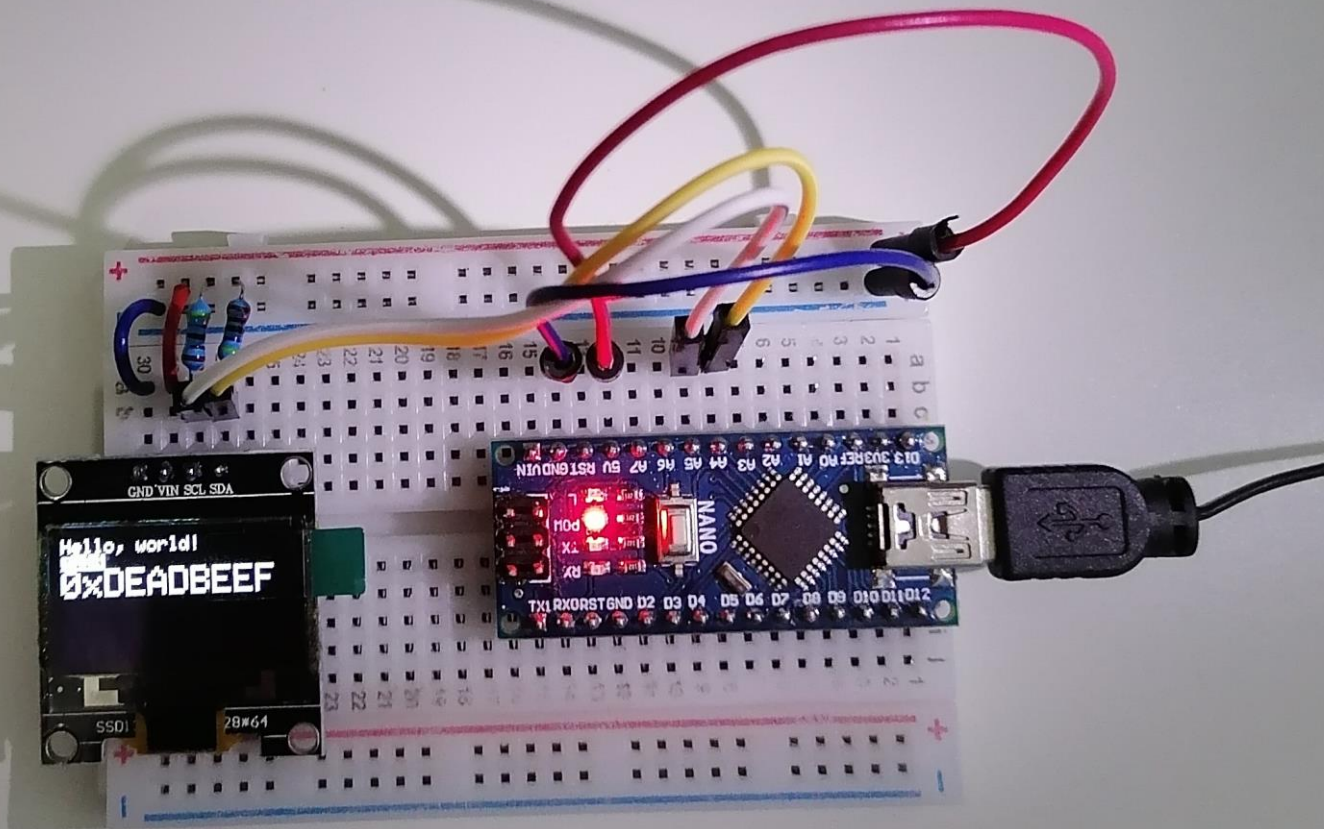
FEBRUARY 9, 2021

Introduction

About this Book

This is a hands-on information documented to illustrate the use of ATMEGA328P chips that has been tested and run successfully on selected Arduino sketch projects using Arduino C/C++ programming language.

In the learning of Arduino platform and compatible microcontrollers, there often lack of information, incomplete online comments and the circuitry implemented may not be the same as the original circuit diagram, all these problems eventually leads to the darkness.

Therefore, I decided to write this book to document issues encountered, resolved via physical trial and error approach. It is not perfect, but with necessary guidelines to get through and move on.

Arduino Nano and Pro Mini were adopted in this book as to facilitate the learning process, and we do want to build something optimum at the end, in case for commercial purposes.

You should be familiar with Arduino IDE, basic electronics fundamental, PCB design, C/C++ software programming and Microsoft Windows environment in order to understand this book.

About the Author

Ricky Gai
is the founder and technical director of Nexuz Innovation, a small R&D IT company established in Kuala Lumpur, Malaysia.

After receiving certification from Oxford Computer Engineering discipline in 1992, my carreer was mostly exposed to C/C++ system software development for decades about 30 years, since the MS-DOS time until today's Windows environments including real-time, networking, file system, 2D/3D games, software driver, application and mobile programming.

Nevertheless, much spare time devoted to further the electronics studies for two years before coming to Arduino platform, and my wife often staring at me. Arduino programming reminded me the MS-DOS season, it brought back memory of something like interrupt, vector and bootsector (eg. Bootloader in Arduino).

All the reference materials and source code are available via Github at:
https://github.com/rickygai/arduino

For any errors found, suggestions and questions, please do email to:
support@nexuzinnovation.com

Well, passion is everything and the key to success, I hope you find something useful here.

DISCLAIMER

Abbrevation	Descriptions
NEXUZ INNOVATION / AUTHOR	refers to the author, Ricky Gai.
READER / READER(S) / READER'S	refers to the person who read or knowledge transferred, accessed the circuitry setup based on the contents illustrated in this document.
COMPONENTS / EQUIPMENTS	refers to electronics components, tools, materials that used as part of the circuitry setup.
CONTENTS	Information described within the document, including software and hardware solutions or mathods described by the author.
IP / INTELLECTUAL PROPERTY / COPYRIGHT / PERMISSION	refers to the copyrighted materials (eg. Photo, Diagram, Source Code, Links) that owned by other creators.

THIS CONTENTS OF THIS DOCUMENT IS SOLELY BASED ON THE ELECTRONICS COMPONENTS OR MATERIALS ADOPTED AND TESTED BY AND AT THE AUTHOR'S PREMISES. DUE TO VOLTAGE SUPPLY VARIES FROM DIFFERENT COUNTRIES AND PHYSICAL SPECIFICATION OF COMPONENTS MAY CHANGE FROM TIME TO TIME, READER(S) TAKE OWN RESPONSIBILITY TO ACCESS THE EXPERIMENTAL SOLUTIONS BASED ON THE INFORMATION DESCRIBED IN THIS DOCUMENT.

THE COMPANY NEXUZ INNOVATION AND THE AUTHOR BARES NO RESPONSIBILITY UPON ANY DAMAGE OR HARM IN CASE HAPPENS TO THE READER(S) OR READER'S SIDE, THIS INCLUDE READER'S RELATED SUCH AS HUMAN HEALTH, HARDWARE EQUIPMENTS AND OTHER LOSSES.
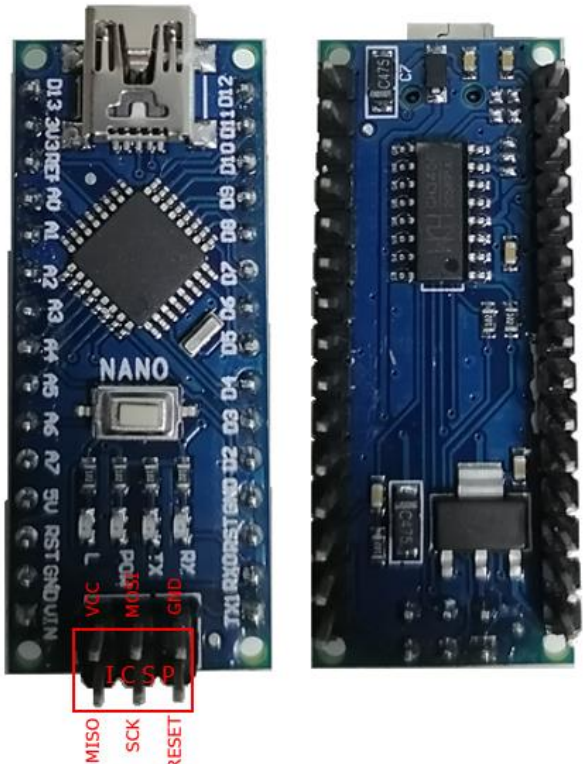
NEXUZ INNOVATION IS A SOLE PROPRIETORSHIP COMPANY WITH NO RELATION TO ANY HARDWARE MANUFACTURERS OR VENDORS MENTIONED IN THIS DOCUMENT, SUCH AS LOGIC GREEN, ARDUINO AND ETC. THE MENTIONED OF INTEGRATED CIRCUIT (IC) OR CHIPS PRODUCT MODELS ARE SOLELY FOR RESEARCH AND DEVELOPMENT PURPOSES ONLY.

THE AUTHOR RESPECTS THE INTELLECTUAL PROPERTY FROM OTHER CREATORS, THIS DOCUMENTATION MAY SHARE SOME PARTLY EXTRACTED PORTION OF PHOTO OR DIAGRAM AS PART OF THE ILLUSTRATION USAGE. IF THERE IS ANY COPYRIGHT INFRINGEMENT, PLEASE DO INFORM THE AUTHOR TO EXCLUDE FROM THIS DOCUMENT.

Types of ATMEGA32P chips

Arduino Nano

[Physical Layout](#)

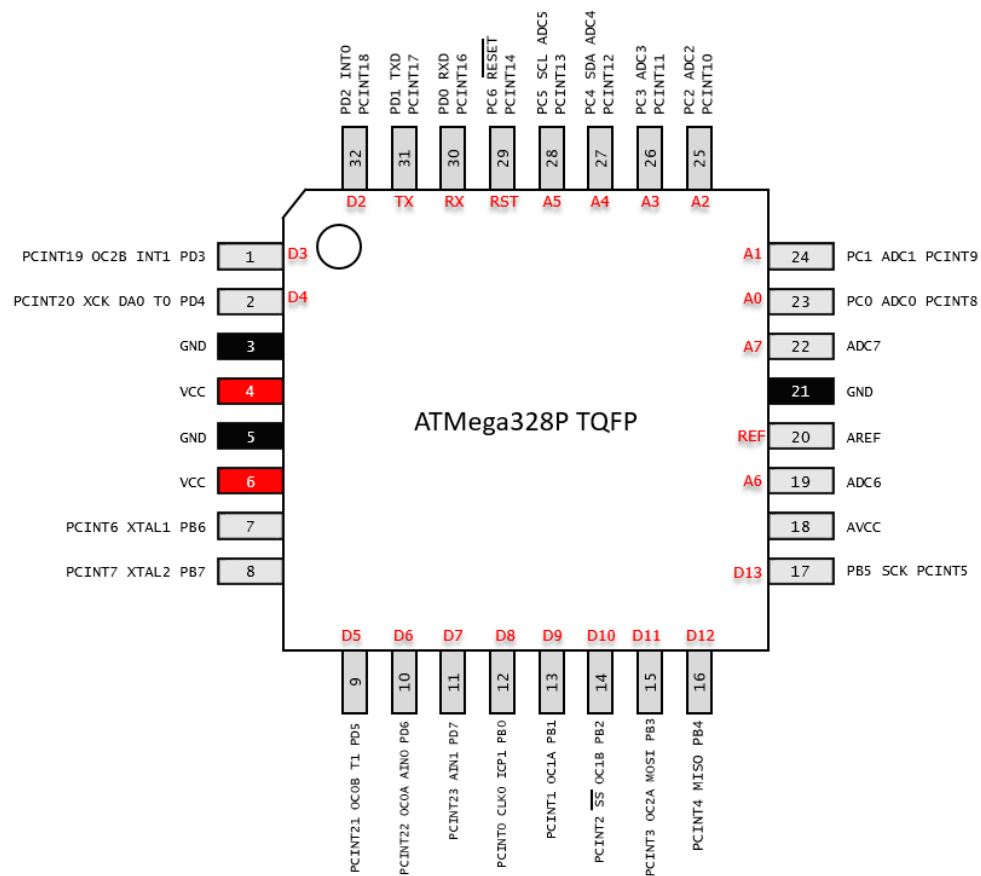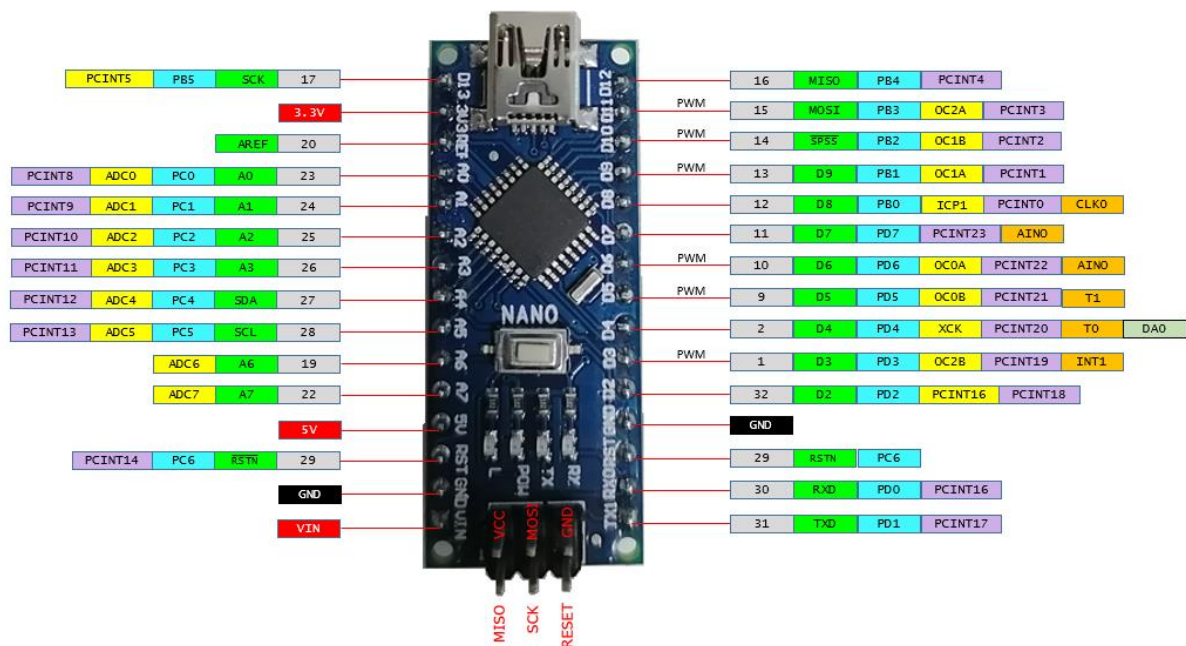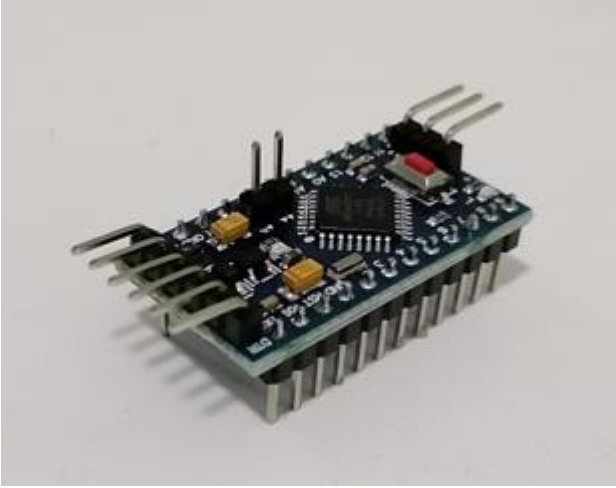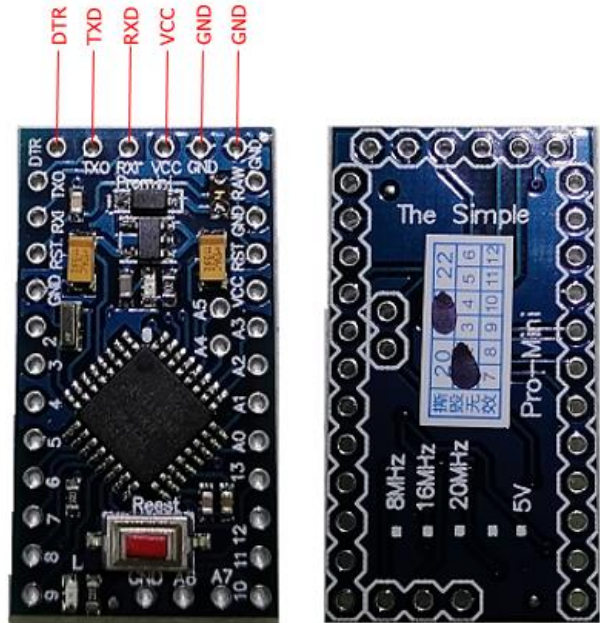| Pinout Diagram	References
Figure 1: Arduino Nano clone board.	ATMEGA328P is one of the Microcontroller created by created by [Atmel](#) where later acquired by [Microchip Technology](#). [ATMEGA328P datasheet](#)

Pinout

Figure 2: ATmega328P TQFP chip pinout.

Figure 3: Arduino NANO clone pinout.

Arduino Pro Mini

Physical Layout

Pinout Diagram	References
 Figure 4: Arduino Pro Mini clone board.	There two models of Arduino Pro Mini, 3.3V and 5V. **NOTE:** 5V Pro Mini model is used in this documentation.

Pinout
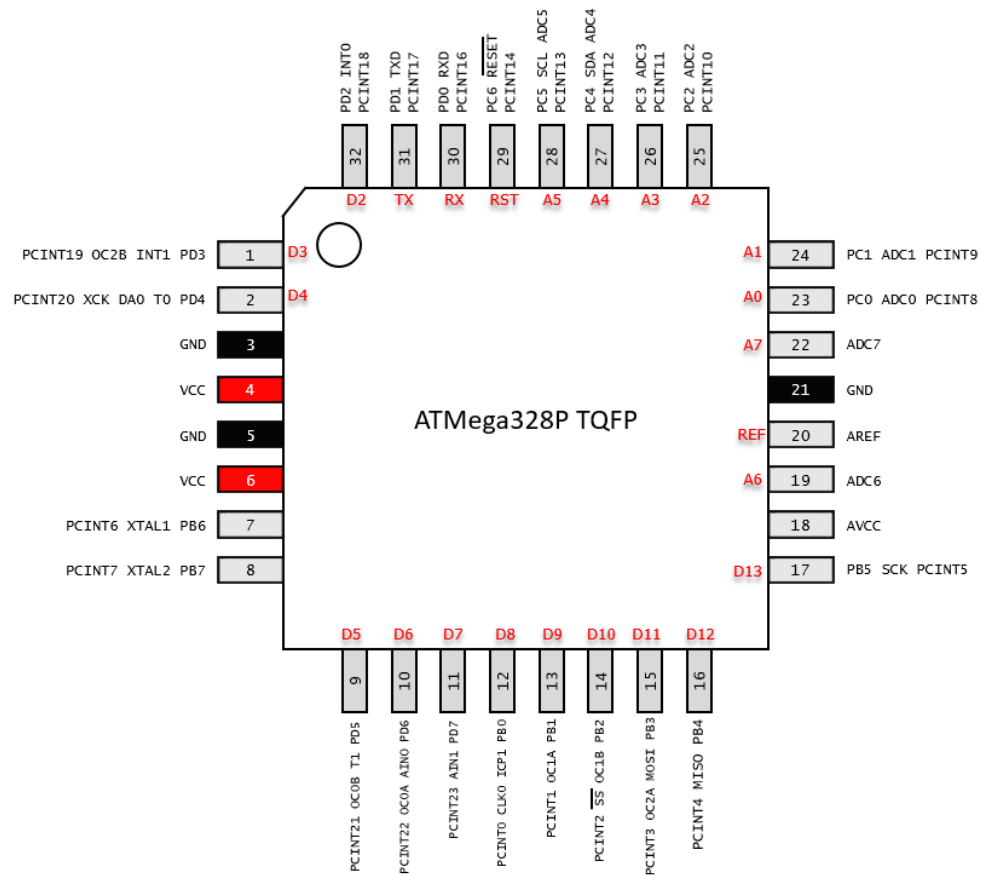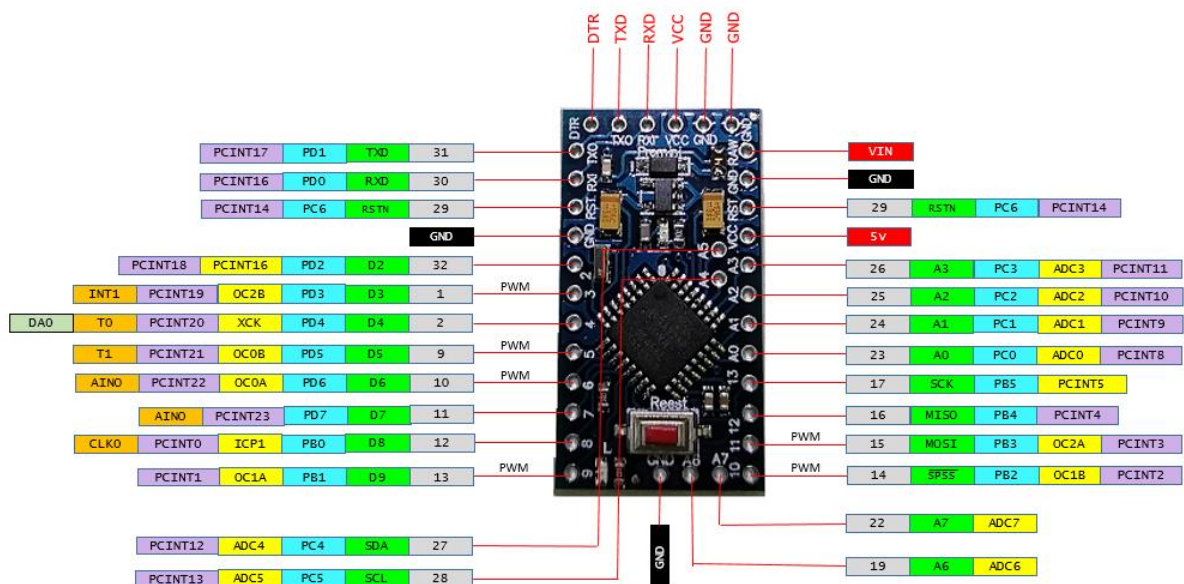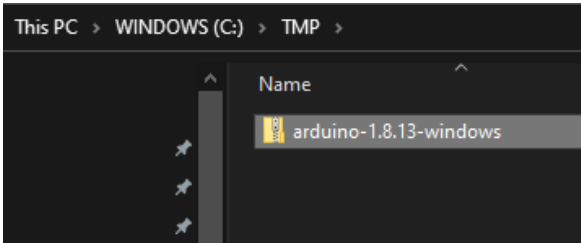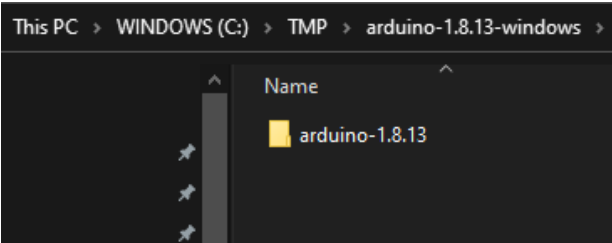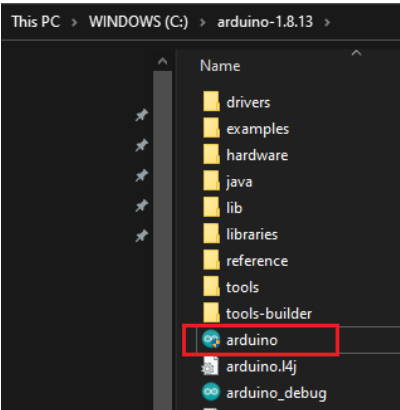

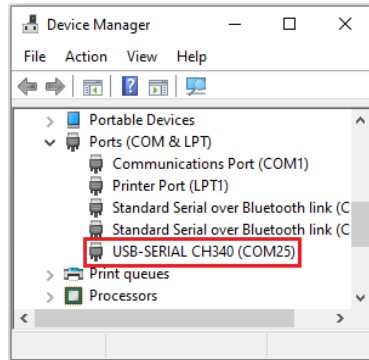Figure 5: ATmega328P TQFP chip pinout.

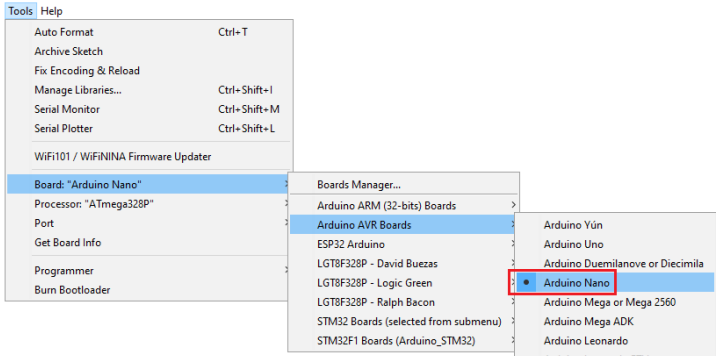Figure 6: Arduino Pro Mini clone pinout.

Prerequisition

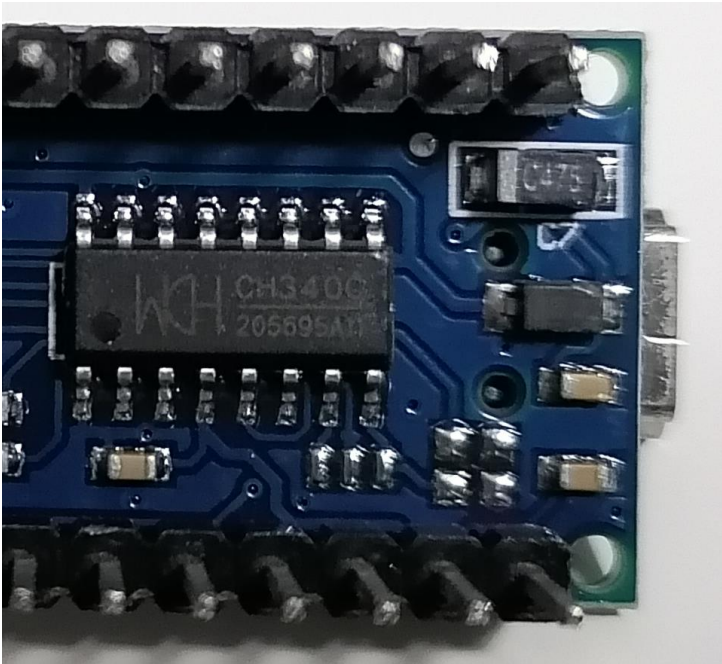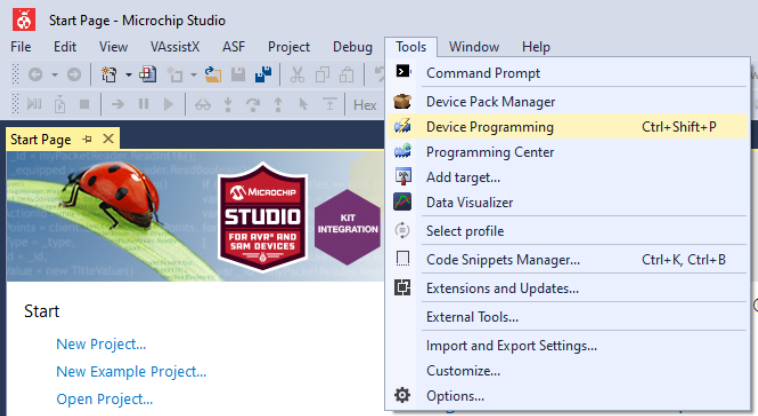## Arduino IDE - Setup

Download, install Arduino IDE software

Diagrams	Descriptions & References
 Figure 7: Arduino IDE download site. Figure 8: Arduino IDE downloaded to C:\TMP Figure 9: The content of the ZIP file, copy the folder to C:\ Figure 10: The arduino.exe	There are two softwares to download and install so the Arduino Nano or Pro Mini can works properly. **Arduino IDE** https://www.arduino.cc/en/software **NOTE:** Download the ZIP version so you are in control and aware of where to put the application and to find related configuration files later as shown in **Figure 7**. **CH340/CH341 USB to serial port** http://www.wch-ic.com/downloads/CH341SER_EXE.html Here, the file **arduino-1.8.13-windows.zip** shown on **Figure 8** is placed under C:\TMP Double click on the file you will see folder **"arduino-1.8.13"**, copy and paste to C:\ When you browse the folder **C:\arduino-1.8.13**, you will find a file "**arduino.exe**" as shown on **Figure 9**. Double click the "**arduino.exe**" to run and you will come to Arduino IDE. Next, make sure to install the **CH340/CH341 software**, this is to create a serial port that talks to the Arduino Nano. Once it is installed, Windows will detect and set a serial port for it like below: Figure 11: The CH340 Serial Port. Next page will show how to configure the Arduino Nano settings.

Configure the Arduino Nano settings

Diagrams	Descriptions & References
 Figure 12: Selecting the "Arduino Nano" board. Figure 13: Selecting the "Arduino Nano" board.	Choose the "**Arduino Nano**" board as shown on **Figure 12**. Alternative, if you are using "Pro Mini" board, you may select "**Arduino Pro or Pro Mini**" in this case. From **Figure 13**, there is "Atmega328p (Old Bootloader) option which is for clone Nano chip. However, I will show you how to burn a new bootloader to a Nano clone and use the new bootloader "**Atmega328P**" without the "(Old Bootloader)" option. The "Port" is to select the serial port or COM port reported when the Arduino Nano is connected via USB to PC.
 Figure 14: The CH340C USB to Serial converter.	**NOTE:** Arduino Nano clone board normally is using **CH340 USB to serial converter IC** as shown on Figure 14. Therefore, it will not work with FTDI FT232RL software driver.

Installing the Microchip Studio software

Diagrams	Descriptions & References
Figure 15: The Microchip Studio.	The chip ATMEGA328P is a product of Microchip Technology Inc., therefore Microchip Studio is used here to program its device with the use of AVRISP MKII Programmer which will be discussed in the section - **How to burn the BOOTLOADER ?** You can download and install the Microchip Studio at the link below: **Microchip Studio** https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices

Arduino Sketch Projects

How to burn the BOOTLOADER ?

Introduction

Diagrams	Descriptions & References
Figure 16: Arduino ICSP interface with AVRISP MKII Programmer. 　Figure 17: The USBASP STK500 interface to Arduino ISP port.	There are many ways to burn bootloader to Arduino Nano. Since the ATMEGA328P chip is the product of Microchip Technology Inc., I decided to choose its legacy AVRISP MKII Programmer to show you how to burn bootloader to Arduino Nano. The 3 reasons why AVRISP MKII Programmer is adopted: - it is supported under Microchip Studio, previously known as Atmel Studio. - in case unstable firmware is burned into, you can refresh and erase the entire ATMEGA328P chip. - although the AVRISP MKII Programmer is no longer produced by Microchip, it is still many similar clone version, functioning the same available in the market such as the link below: **AVRISP MKII Programmer** https://www.aliexpress.com/item/32827854235.html?spm=a2g0s.9042311.0.0.27424c4dKEYS9W To interface the Arduino Nano ICSP port, we need the USBASP STK500 like this: 　Figure 18: The USBASP STK500 10-to-6 pins converter. The USBASP STK500 normally comes together with the AVRISP MKII Programmer, but you can always purchase it separately. **USBASP STK500** https://shopee.com.my/product/291146738/3552110759 Both devices are not expensive, is highly recommended to purchase and use it for the project.

　　　　　　　　　　www.nexuzinnovation.com

The Parts list

Components	Quantity
Arduino Nano	1
MB102 Mini Breadboard 8.5CM x 5.5CM 400 Holes	1
AVRISP MKII Programmer	1
USBASP STK500	1
USB 2.0 Mini B cable connector	1

Setup the Arduino Nano with AVRISP MKII Programmer
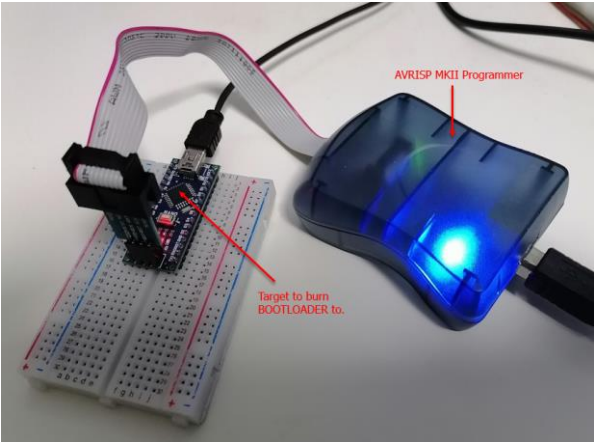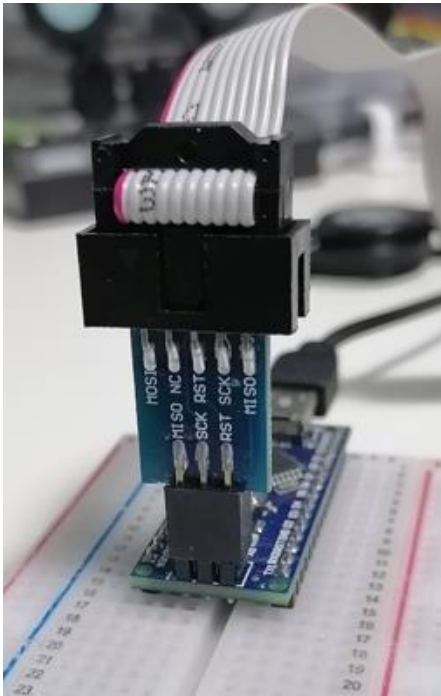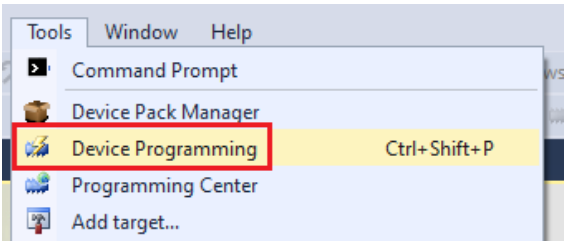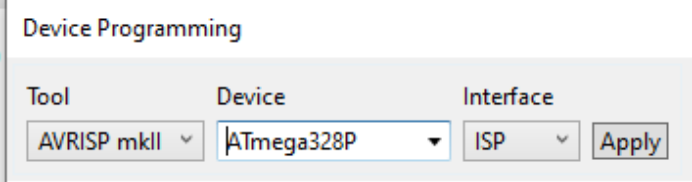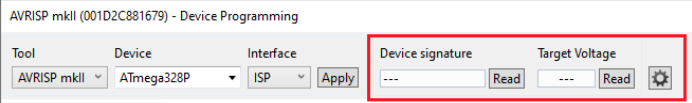
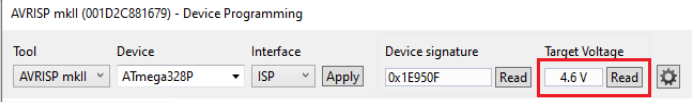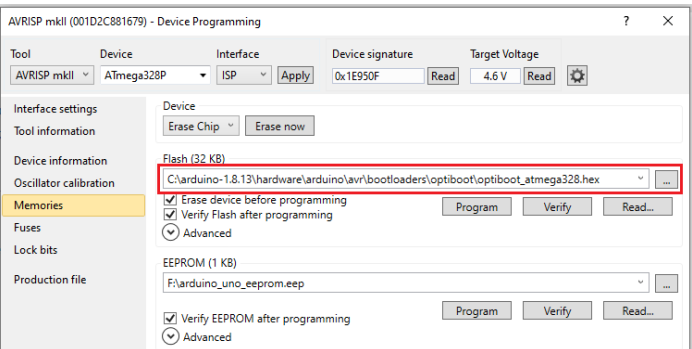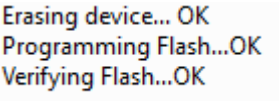| Illustrations	Descriptions
1. First, you have to complete the **Prerequisition** stage. 2. After you have completed step 1, turn OFF all devices and connect the Arduino Nano with AVRISP MKII Programmer as shown at **Figure 20**. 3. Make sure the interface connection is based on **Figure 21**. 4. Turn ON both devices, you have to power the Arduino Nano with USB Mini B connection also to ensure enough voltage level reading value that used in Microchip Studio later. 5. If the connection established successfully, the AVRISP MKII Programmer will show up the green light: Figure 19: The Green light indicated success connection.	 Figure 20: Arduino ICSP interface with AVRISP MKII Programmer. Figure 21: The USBASP STK500 interface to Arduino ISP port.

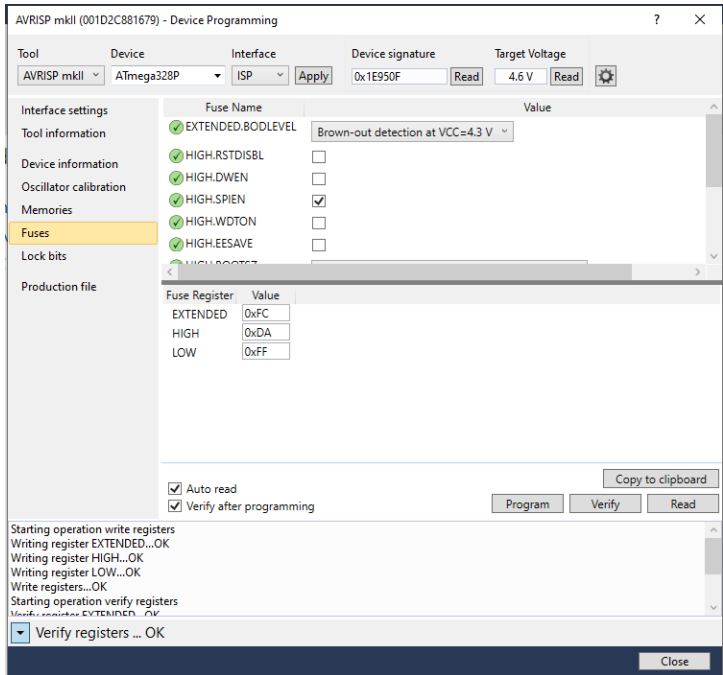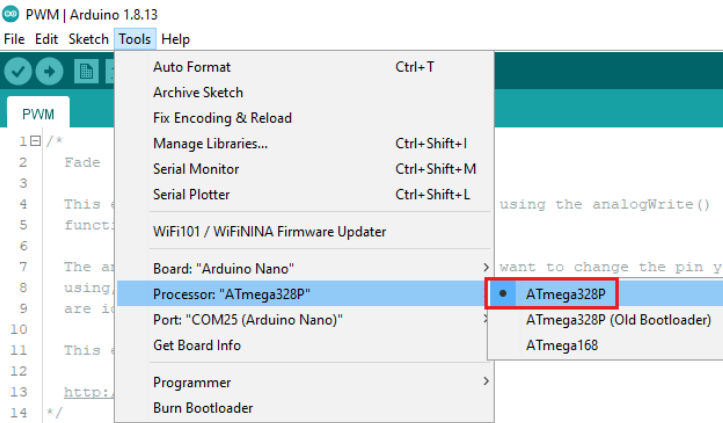www.nexuzinnovation.com

Opening the Microchip Studio

Illustrations	Descriptions
6. At the Microchip Studio IDE, goto Tools → Device Programming, click on it such as **Figure 22**.	Figure 22: The Device Programming option.
7. This will bring you to the "Device Programming" dialog, select the Tool "AVRISP mkII", Device "Atmega328P" and Interface "ISP" then click the "Apply" button as shown on **Figure 23**. 8. Once you have clicked the "Apply" button, the right options will be enabled such as **Figure 24**. 9. Click the "Read" button of Device signature will auto fill in the "---" textboxes like **Figure 25**. 10. The Target Voltage is 4.6V as shown on Figure xx, this is where the Arduino Nano is powered from its USB connection, otherwise it will be less eg. 3.xV which is not the valid level to program into the chip. 11. Goto "Memories", click the "Erase now" button to empty the flash area on the chip. 12. Browse for the file **"optiboot_atmega328.hex"** which is the bootloader for Arduino Nano, in my case it is located under: C:\arduino-1.8.13\hardware\arduino\avr\bootloaders\optiboot\optiboot_atmega328.hex 13. Click the "Program" button of "Flash (32 KB)" to burn the bootloader to the chip. 14. Ignore the EEPROM as not in use. 15. If the Erase and Program process completed with no issues, the messages at the bottom of the "Device Programming" dialog will be like **Figure 27**.	Figure 23: The initial settings of Device Programming. Figure 24: The initial settings of Device Programming. Figure 25: The Target Voltage with Arduino Nano USB powered is 4.6V Figure 26: The Target Voltage with Arduino Nano USB powered is 4.6V Figure 27: The success messages of Flash erased and new bootloader is burned into the chip.

Updating the Fuses to accomplish

Illustrations	Descriptions and References
16. Goto "Fuses", key in the following values for Arduino Nano: EXTENDED 0xFC HIGH 0xDA LOW 0xFF 17. Fuses values normally set by manufacturer, and varies from different Arduino boards. 18. Click the "Program" button. 19. If the operation is a success, the messages should be showing like the bottom part of the dialog on **Figure 28**. 20. If you wish to protect the flash contents or bootloader, "Lock bits" is the option. 21. To understand more about "Fuses", and "Lock bits", is recommended to read the book Arduino Software Internal. 22. Congratulations, you have successfully burned a new bootloader into the ATMEGA328P chip. 23. Now, you may upload sketch to it using Arduino IDE with or without the AVRISP MKII Programmer and using the new bootloader as shown on **Figure 29**. 24. DONE	 Figure 28: The Fuses settings for Arduino Nano. Figure 29: The use of new bootloader "ATmega328P" option.