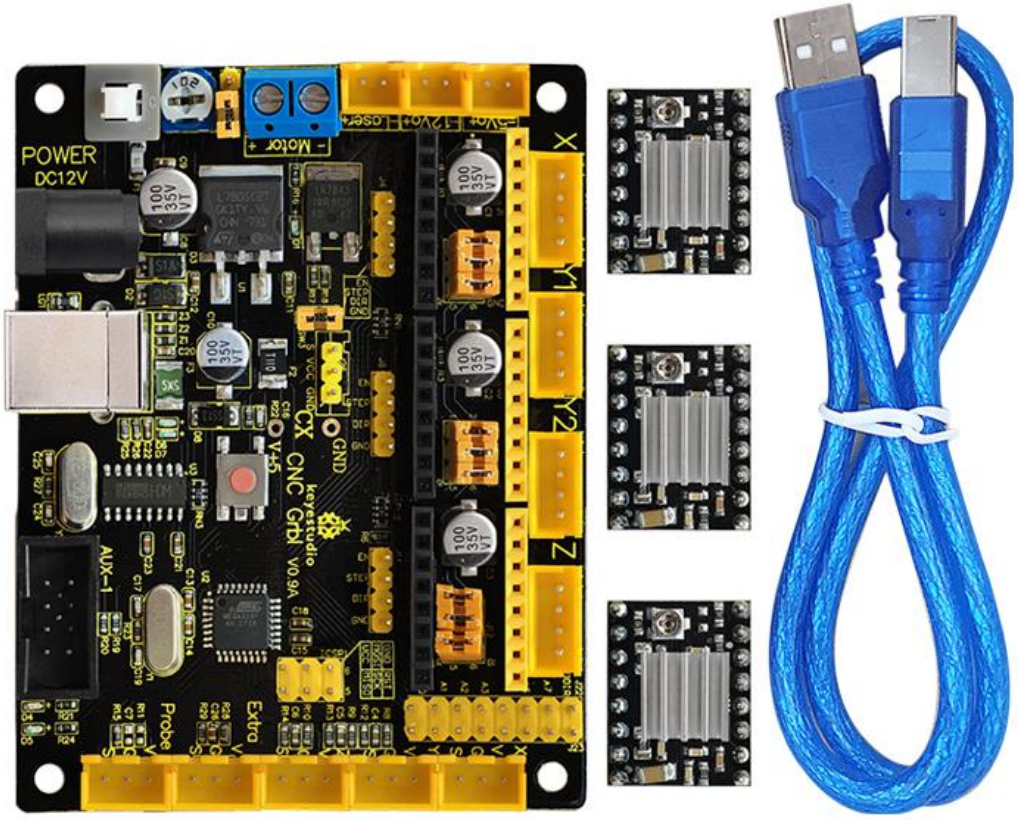
Keyestudio CNC V0.9A+4988 Driver With Heat Sink Kit



1. Introduction

This kit mainly includes a keyestudio CNC V0.9A，three A4988 driver modules with heat sink and a USB cable.

Keyestudio CNC GRBL V0.9 is a main board developed by keyestudio for CNC, laser engraving machine, writing robots and other robots. It has complete interfaces with cost-effective and can be driven via external connection, suitable for DIY and factory use.

The A4988 module is a DMOS micro-stepping driver with converter and over-current protection, which can operate a bipolar stepper motor in full, half, 1/4, 1/8 and 1/16 step modes, with output-driven performance up to 35 V and ± 2A.

The A4988 also includes a fixed off-time current regulator that operates in slow or mixed-decay modes.

2. Specifications of CNC V0.9A

- Microprocessor：MEGA328p
- Input Voltage：DC 12V
- Supporting File Format：Gcode
- Supporting Machine Structure：CNC, laser engraving machine, writing robots

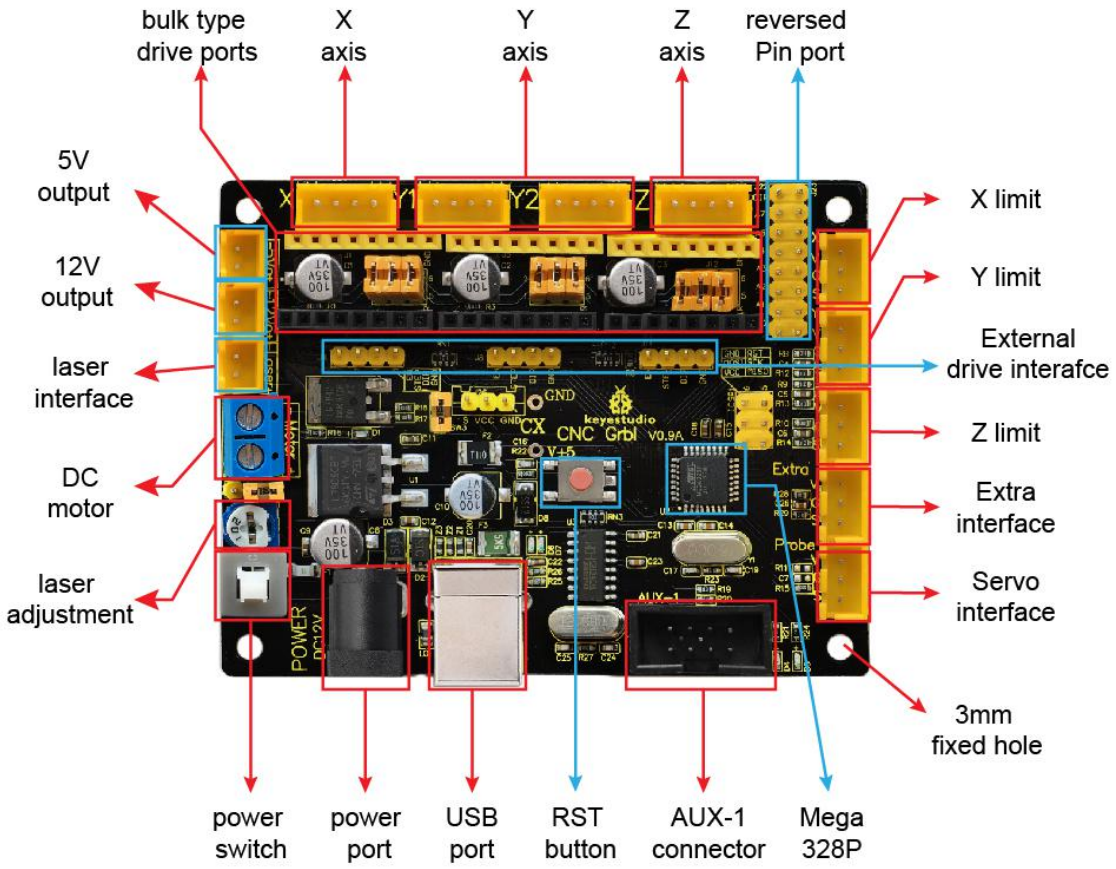
3. Features of A4988 Driver

- With simple stepper and direction control interface.
- Five different step modes: full, half, 1/4, 1/8 and 1/16 step modes.
- Adjustable potentiometer used to adjust the max current output to gain higher step rate.
- Automatic decay mode detection/selection.
- Overheat closed circuit, under-voltage lockout, cross-current

protection.

- Ground short-circuit protection and load short-circuit protection.

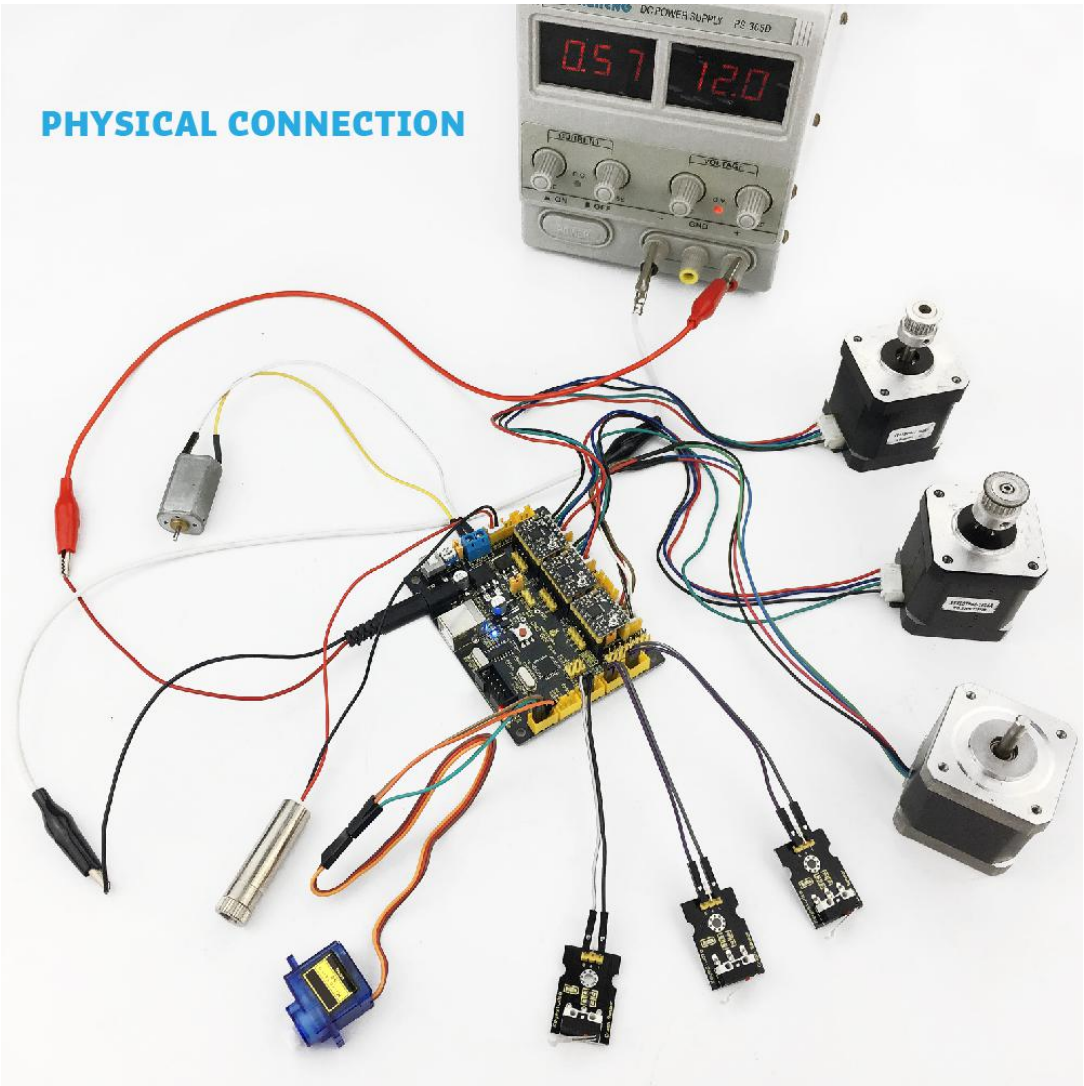
4. PINOUT Diagram



5. Setting Method of A4988 Working Mode

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

6. Wiring Diagram

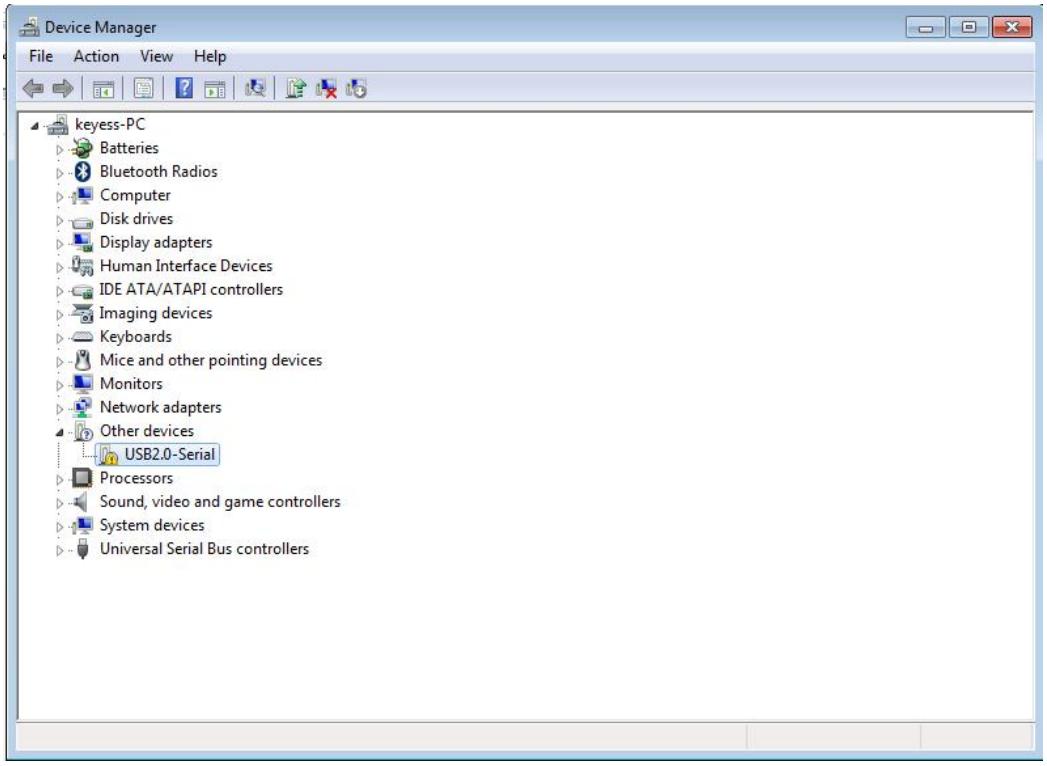


7. Install Driver Software and Development Environment Software IDE

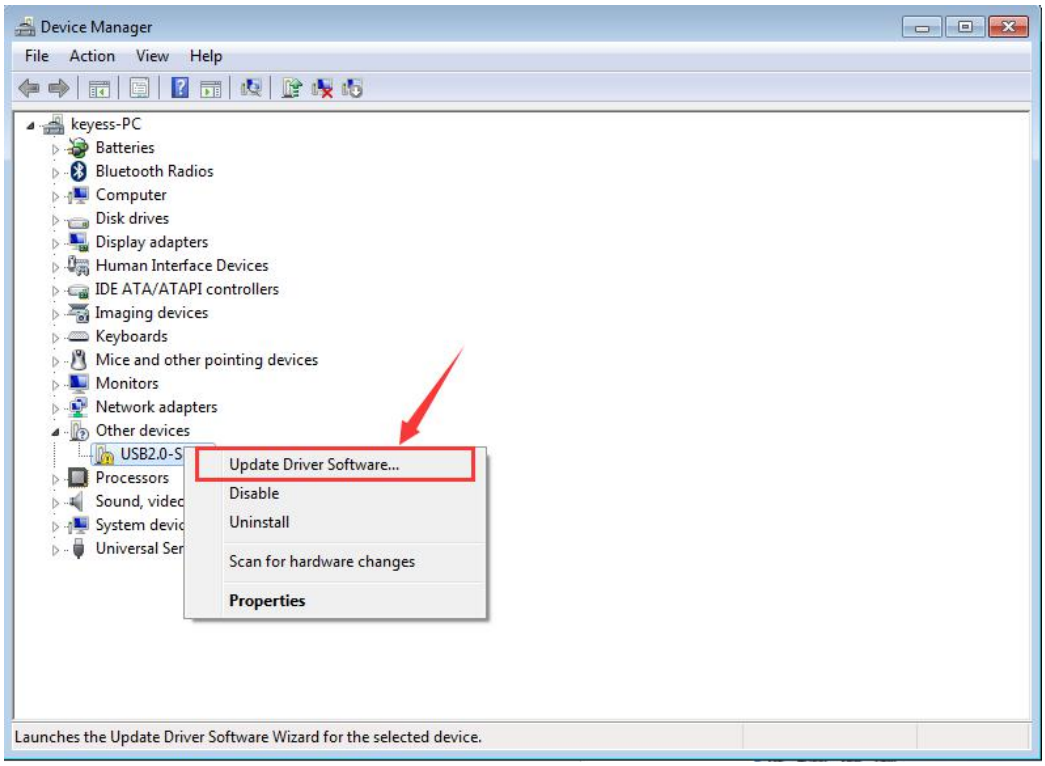
（1）Install Diver Software

For different operating system, there may be slight difference in installation method. Below is an example in WIN 7.

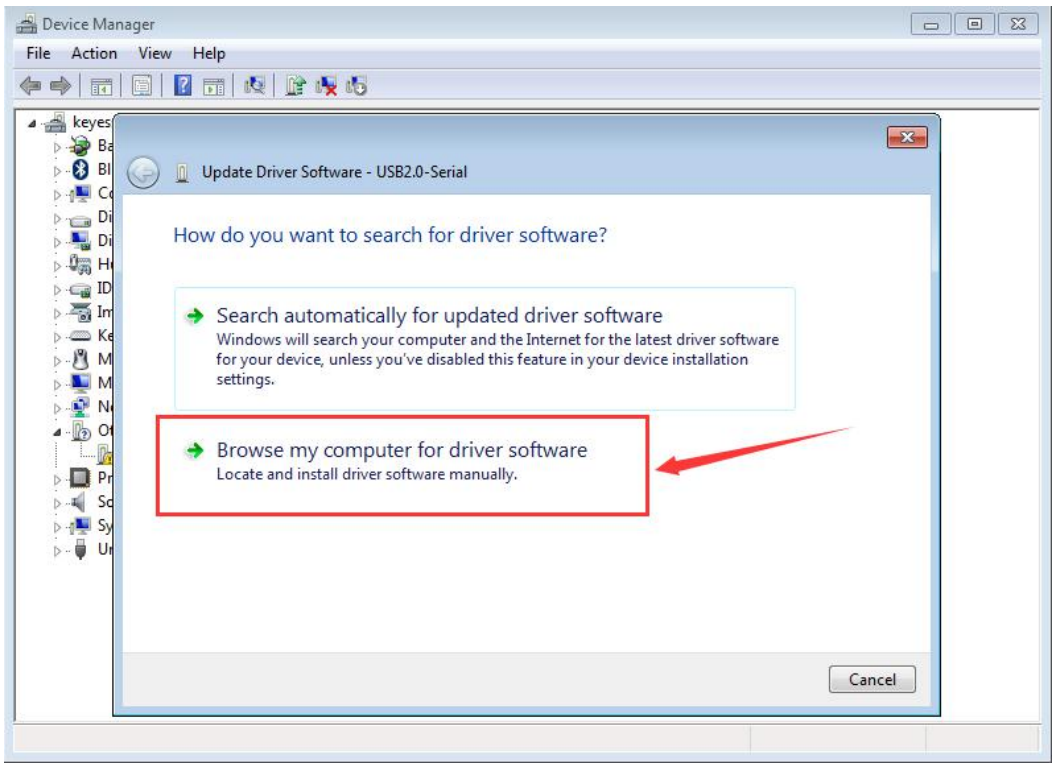
a. When you connect Keyestudio CNC GRBL V0.9 to your computer the first time, right click "Computer" —>"Properties"—> "Device manager", you can see "USB2.0-Serial".



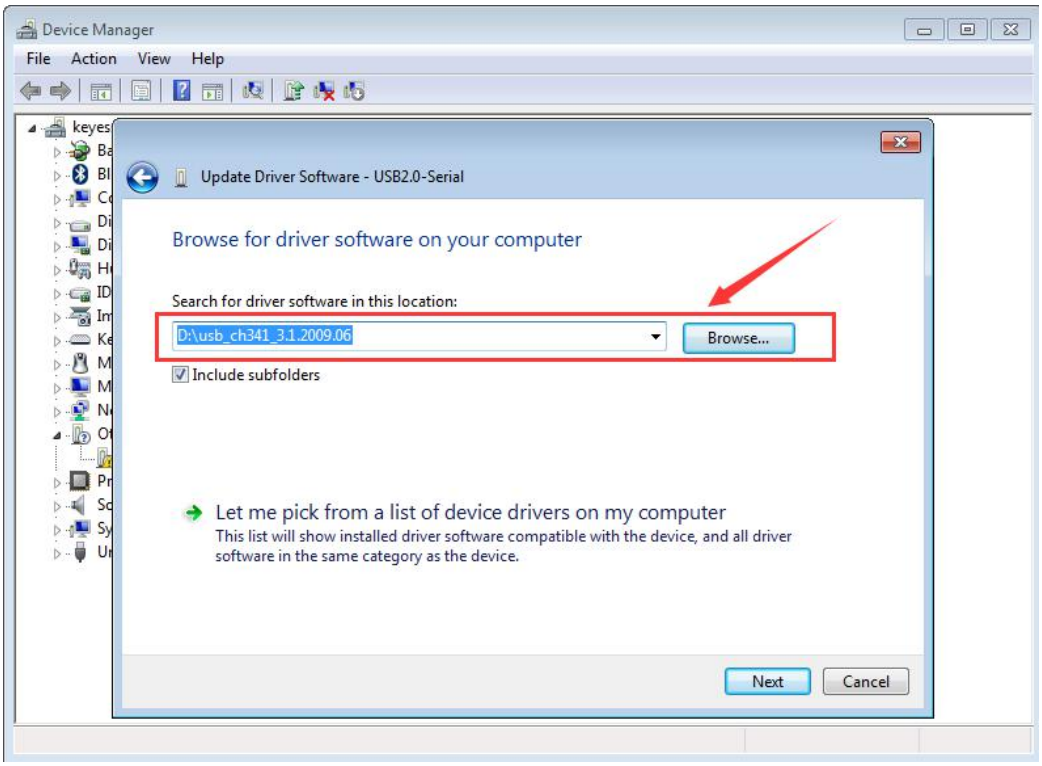
b. Click "USB2.0-Serial", select "Update Driver software".



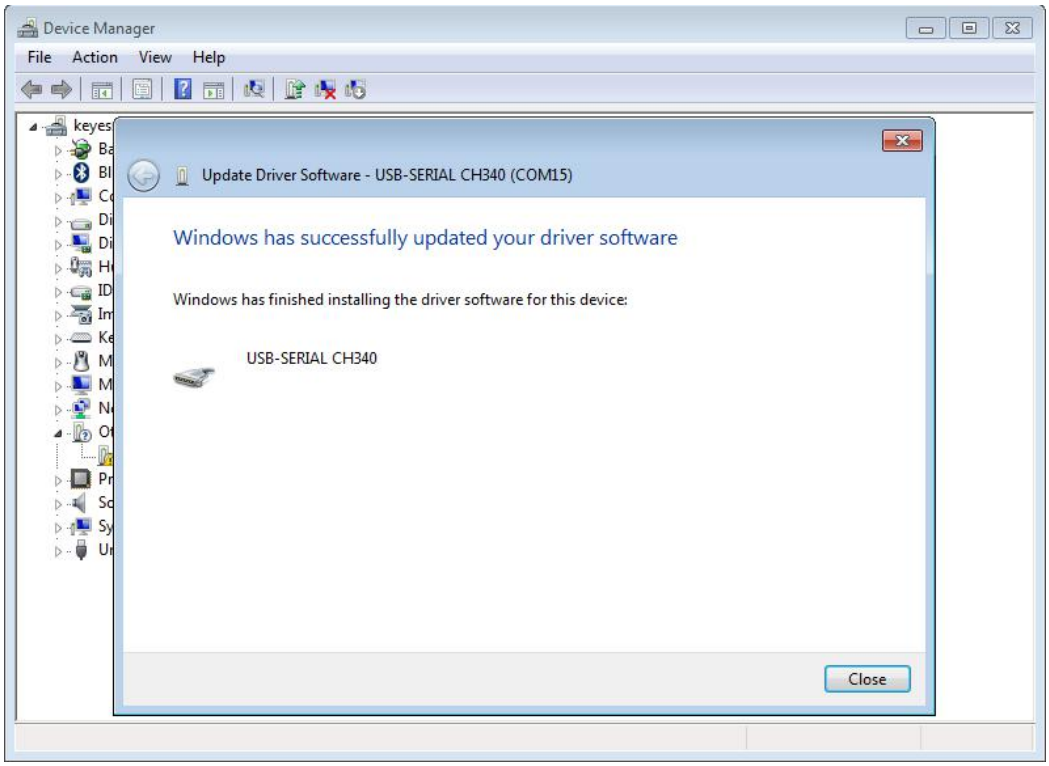
c. In this page, click "Browse my computer for driver software".



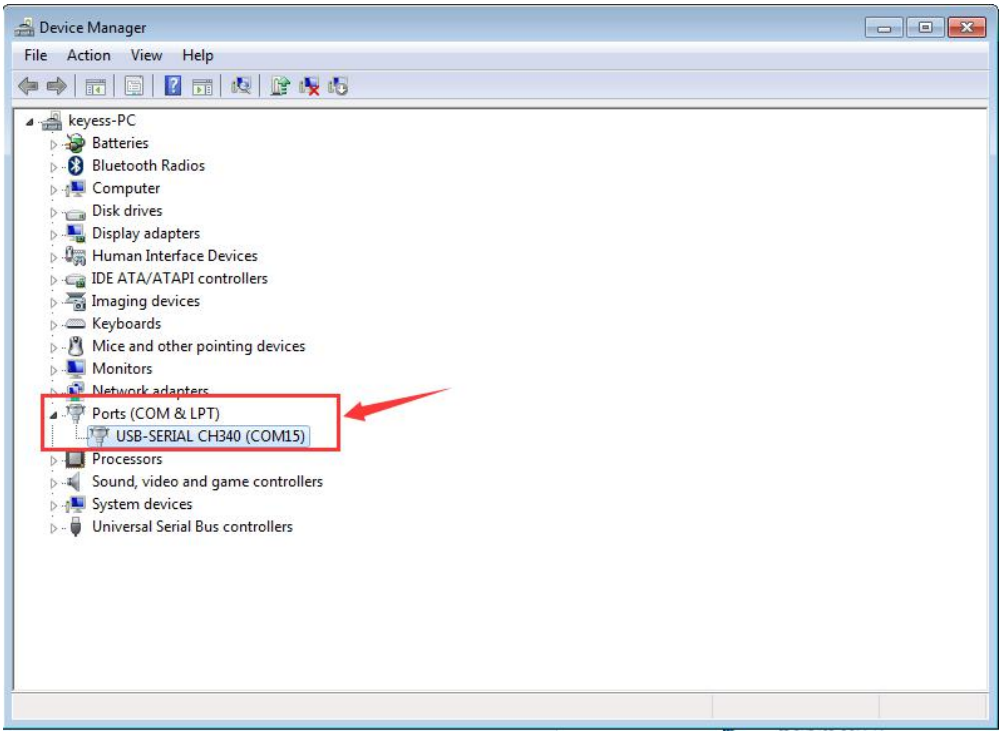
d. Find the "usb_ch341_3.1.2009.06" file.



e. Click "Next", installation completed; Then, click "Close".



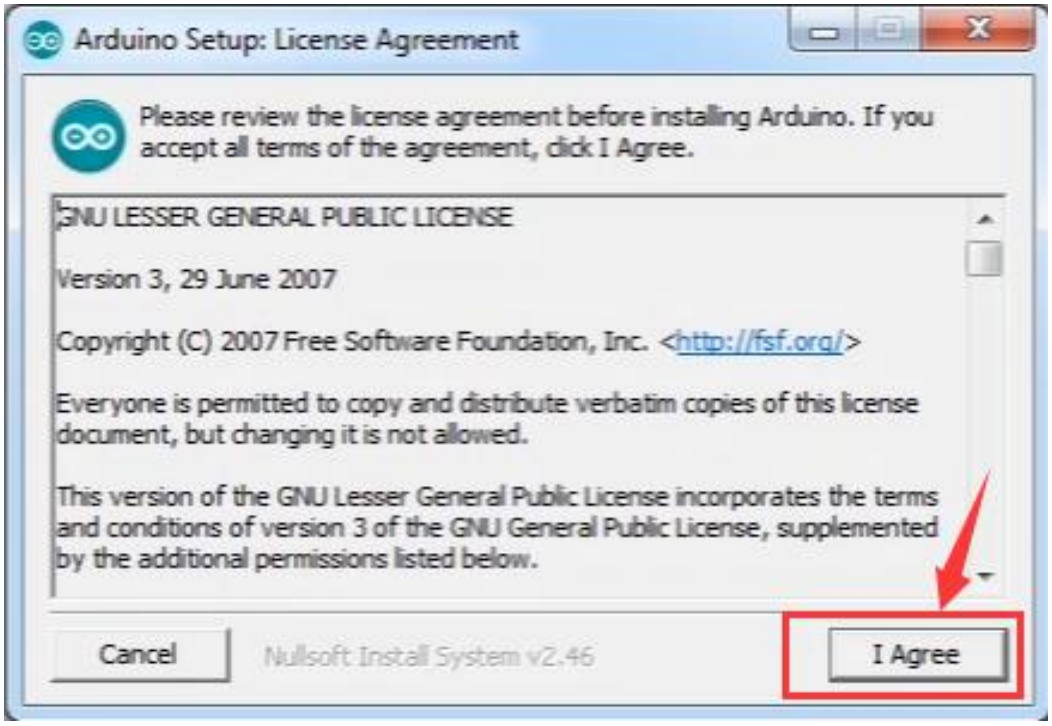
f. After driver is installed, go to "Device manager" again. right click "Computer" —> "Properties"—> "Device manager", you can see device as below figure shown, also the correct Com port.



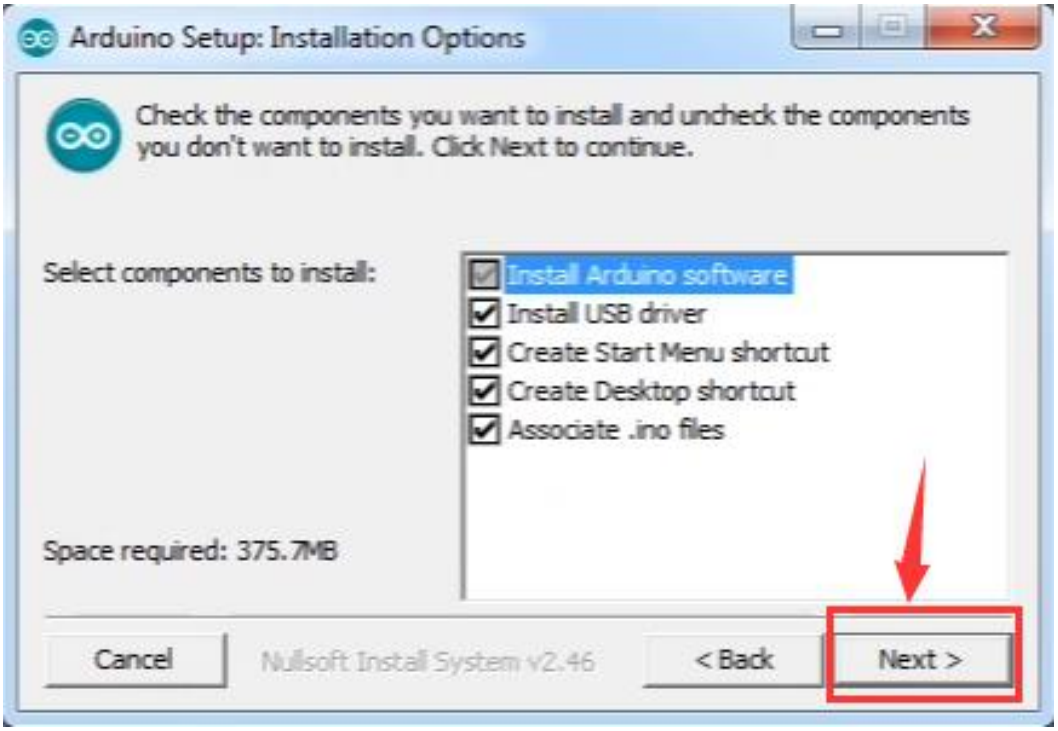
(2) Install Development Environment Software IDE

a. Double click arduino-1.5.6-r2-windows to start.

Select "I Agree"to accept license agreement.

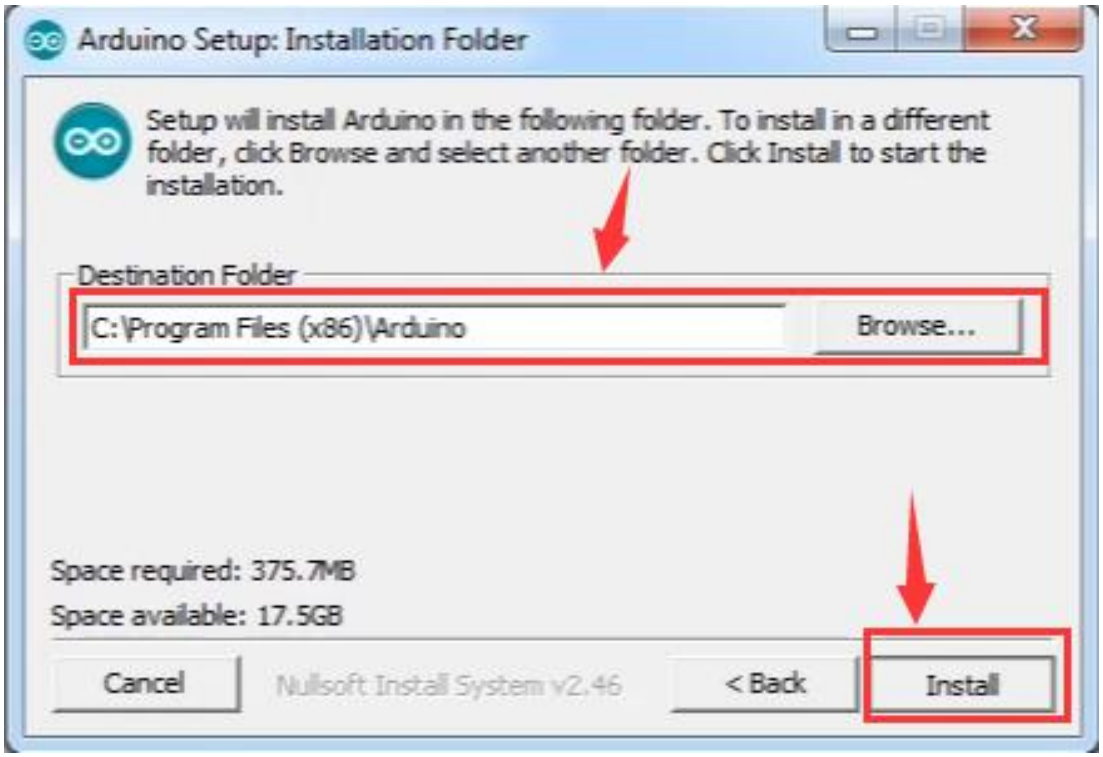


b. Select components to install and click "Next".

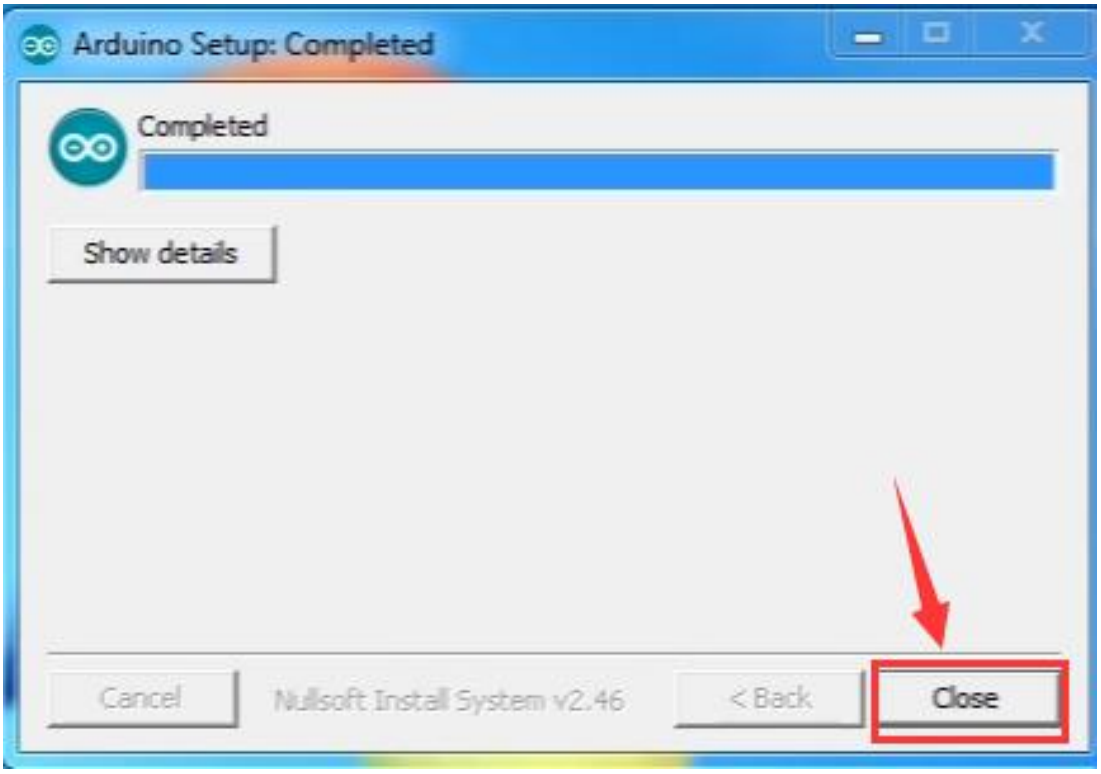


c. Click "Browse" and select another folder. Click "Install" to start

the installation.



d. Finally, wait for a few minutes to finish.



8. Testing Method

Wire it up well as wiring diagram shown, upload the below code to Keyestudio CNC GRBL V0.9 using IDE. Then you can check the function of each interface.

Sample Code:

```
////////////////////////////////////////////////////////////////////////
#define Light1   13
#define Light2   A1
#define Light3   A2
#define Light4   A3
#define Light5   A4
#define Light6   A5

#define EN1   8
#define X_DIR      5      //X axis   direction control of stepper motor
#define Y_DIR      6      //y axis   direction control of stepper motor
#define Z_DIR      7      //z axis   direction control of stepper motor
```

```
#define X_STP       2          //x axis step control
#define Y_STP       3          //y axis step control
#define Z_STP       4          //z axis step control
#define X_LIMIT     9          //X limit
#define Y_LIMIT     10         //Y limit
#define Laser       11         //motor or laser control pin
#define Z_LIMIT     12         //Z limit
#define E_LIMIT     A0         //E limit
const int Button_A7 = A7;
const int Button_A6 = A6;
int Button_value_A7 = 0;
int Button_value_A6 = 0;
int Button_valueX = 0;
int Button_valueY = 0;
int Button_valueZ = 0;
int Button_valueE = 0;

void setup() {
  pinMode(Light1, OUTPUT);      pinMode(Light2, OUTPUT);
  pinMode(Light3, OUTPUT);      pinMode(Light4, OUTPUT);
  pinMode(Light5, OUTPUT);      pinMode(Light6, OUTPUT);
```

```
  pinMode(EN1, OUTPUT);
  pinMode(X_DIR, OUTPUT);
  pinMode(Y_DIR, OUTPUT);
  pinMode(Z_DIR, OUTPUT);
  pinMode(X_STP, OUTPUT);
  pinMode(Y_STP, OUTPUT);
  pinMode(Z_STP, OUTPUT);
  pinMode(Button_A7, INPUT);
  pinMode(Button_A6, INPUT);
  pinMode(E_LIMIT, INPUT);
  pinMode(X_LIMIT, INPUT);
  pinMode(Y_LIMIT, INPUT);
  pinMode(Z_LIMIT, INPUT);
  Serial.begin(9600);
}

void EN()
{
  digitalWrite(EN1, LOW);
}
//stepper motor turn forward or reverse
void turn(boolean dir, int steps)
```

```
{
   EN();
   digitalWrite(X_DIR,dir);
   digitalWrite(Y_DIR,dir);
   digitalWrite(Z_DIR,dir);
   delay(100);
   for(int i=0;i<steps;i++)
   {
      digitalWrite(X_STP, HIGH);
      digitalWrite(Y_STP, HIGH);
      digitalWrite(Z_STP, HIGH);
      delayMicroseconds(100);
      digitalWrite(X_STP, LOW);
      digitalWrite(Y_STP, LOW);
      digitalWrite(Z_STP, LOW);
      delayMicroseconds(100);
    }
}
//laser is on
void Laser_ON()
{
   digitalWrite(Laser, HIGH);
```

```
    delay(500);
    //digitalWrite(Laser, LOW);
    //delay(500);
}
//laser is off
void Laser_OFF()
{
    digitalWrite(Laser, LOW);
    delay(500);
    //digitalWrite(Laser, LOW);
    //delay(500);
}

void loop()
{

    Button_valueX = digitalRead(X_LIMIT);
    if(Button_valueX == LOW)
    {
        digitalWrite(Light3, HIGH);
        turn(true, 4000);
    }
```

```
else
{
   digitalWrite(Light3, LOW);
}
Serial.print("Button_valueX = ");
Serial.println(Button_valueX);

Button_valueY = digitalRead(Y_LIMIT);
if(Button_valueY == LOW)
{
   digitalWrite(Light4, HIGH);
   turn(false, 4000);
}
else digitalWrite(Light4, LOW);
Serial.print("Button_valueY = ");
Serial.println(Button_valueY);

Button_valueZ = digitalRead(Z_LIMIT);
if(Button_valueZ == LOW)    digitalWrite(Light5, HIGH);
else digitalWrite(Light5, LOW);
Serial.print("Button_valueZ = ");
Serial.println(Button_valueZ);
```

```
Button_value_A7 = analogRead(Button_A7);
if(Button_value_A7 == 0)    digitalWrite(Light1, HIGH);
else digitalWrite(Light1, LOW);
Serial.print("Button_value_A7 = ");
Serial.println(Button_value_A7);

Button_value_A6 = analogRead(Button_A6);
if(Button_value_A6 == 0)    digitalWrite(Light2, HIGH);
else digitalWrite(Light2, LOW);
Serial.print("Button_value_A6 = ");
Serial.println(Button_value_A6);

Button_valueE = analogRead(E_LIMIT);
if(Button_valueE == 0)
{
   digitalWrite(Light6, HIGH);
   Laser_ON();
}
else
{
   digitalWrite(Light6, LOW);
```

```
    Laser_OFF();
  }
  Serial.print("Button_valueE = ");
  Serial.println(Button_valueE);
}
//////////////////////////////////////////////////////////////////////////
```

9. Install Firmware and Grbl Controller

a. Write test program to keyestudio UNO R3

copy the folder **GRBL_ Arduino_Library_keyes** in the data packet and paste it to the folder libraries in your Arduino IDE document installation.

Code:

```
**********************************************
#include <grblmain.h>
void setup(){
startGrbl();
}
void loop(){}
```

//Burn the code above to keyestudio UNO R3

****

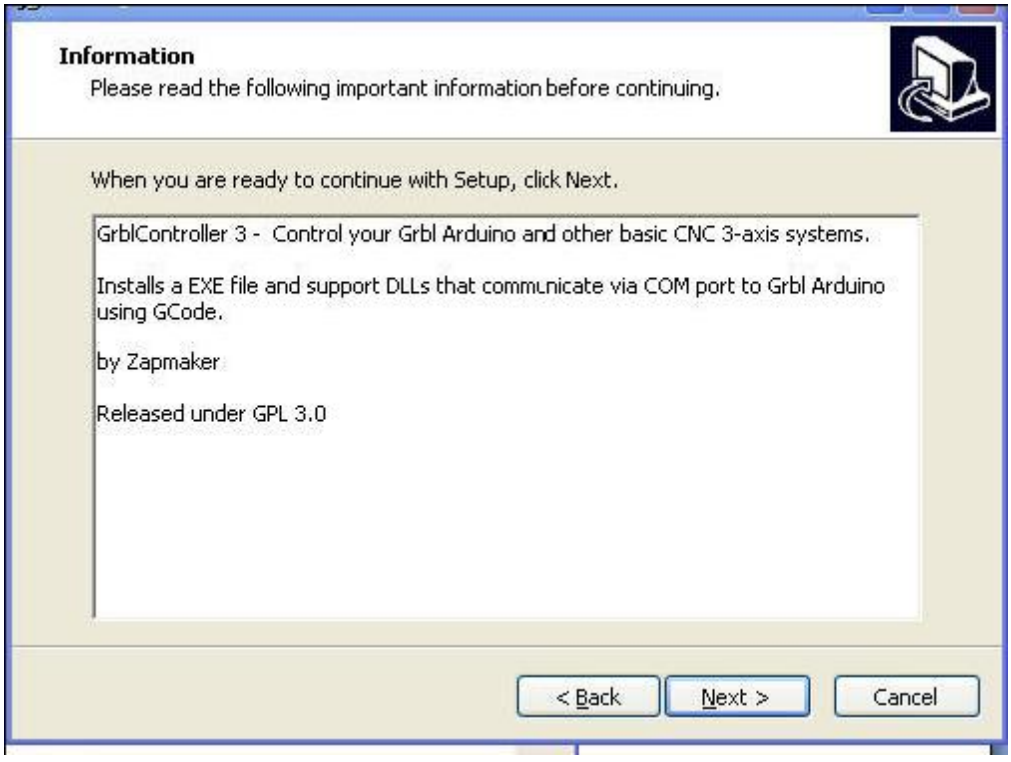
b. Install GrblController361 Software

Grbl Controller software is used to send GCode to CNC Machines. Run Grbl Controller361 Setup in your installation packet, the interface below will come out:

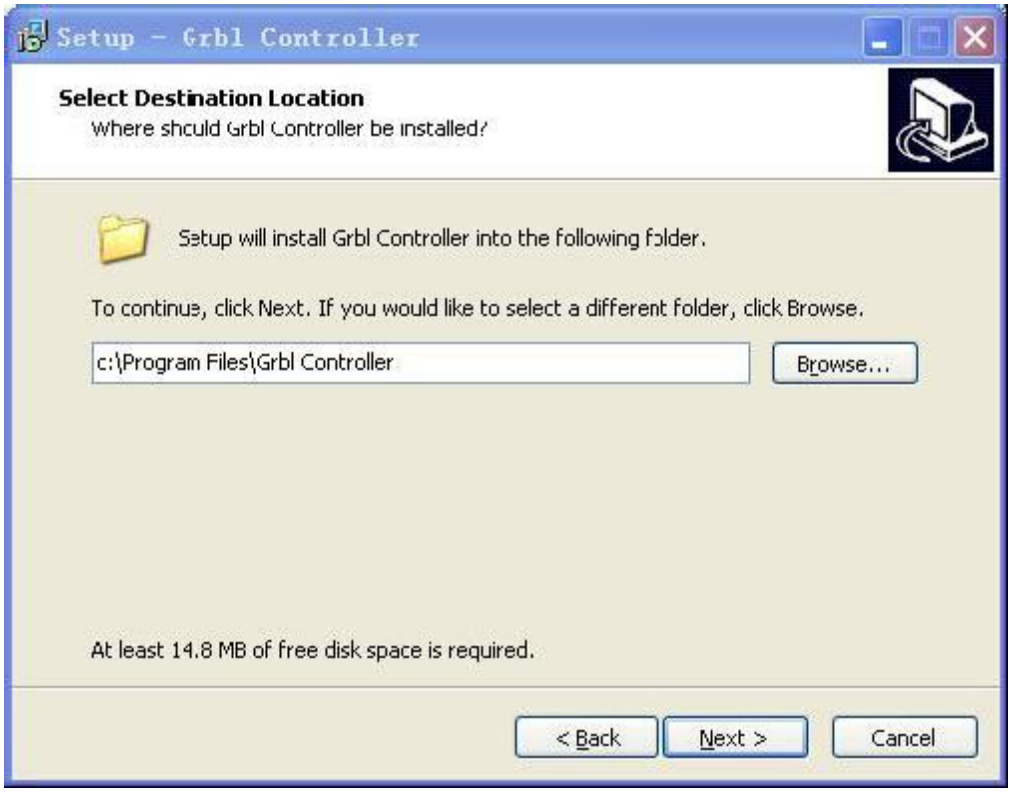
Click **Next** to continue.

For a license agreement, please check I accept the agreement and click **Next**.

When you are ready to continue with Setup, click **Next**.

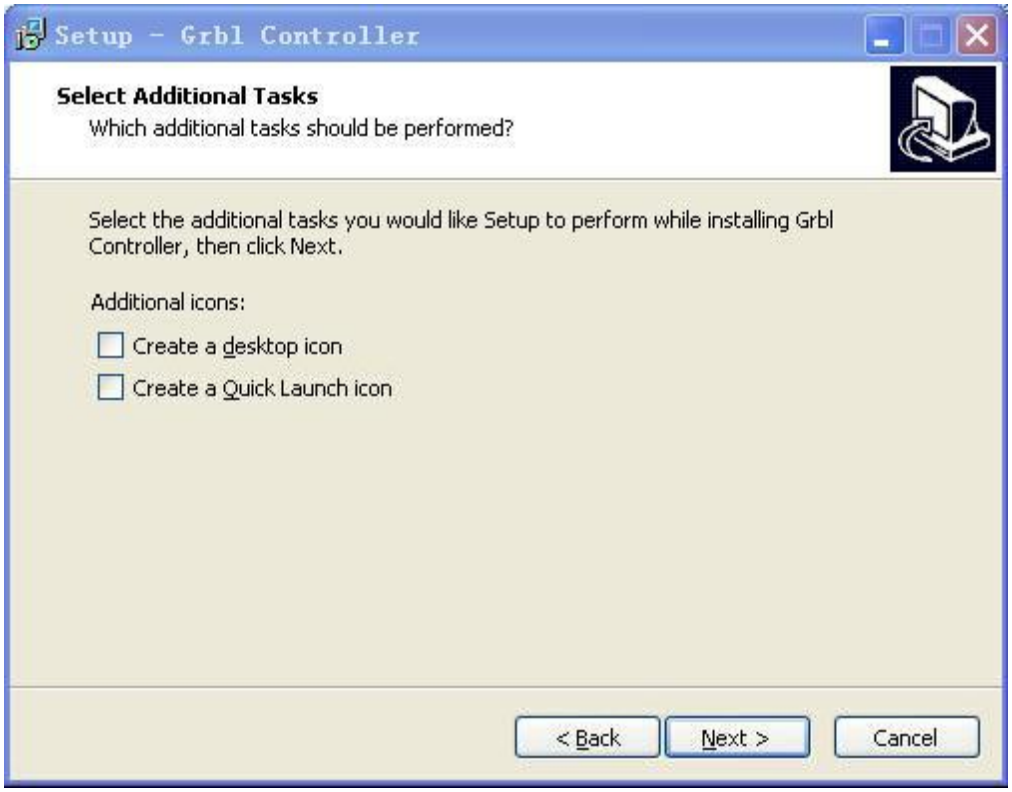


To continue, click **Next**. If you would like to select a different folder to install, click **Browse**.

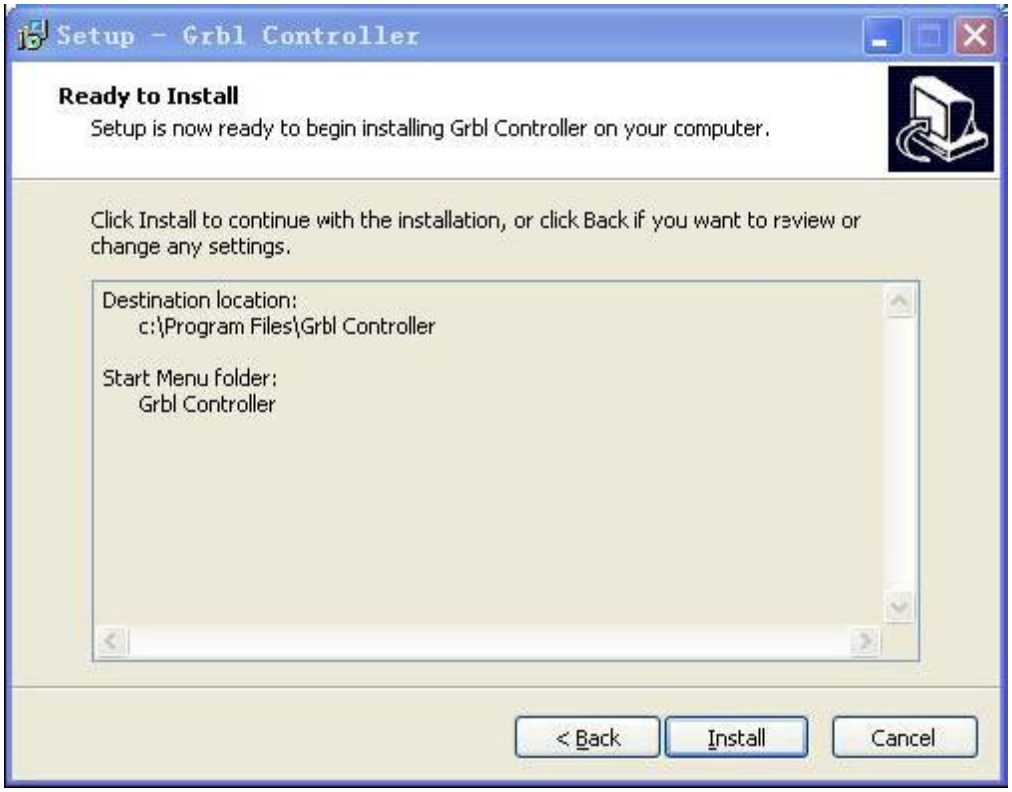


To continue, click Next. If you would like to select a different folder to place program's shortcuts, click **Browse**.

Select the additional tasks you would like Setup to perform while installing Grbl Controller, then click **Next**.



Click **Install** to continue with the installation.



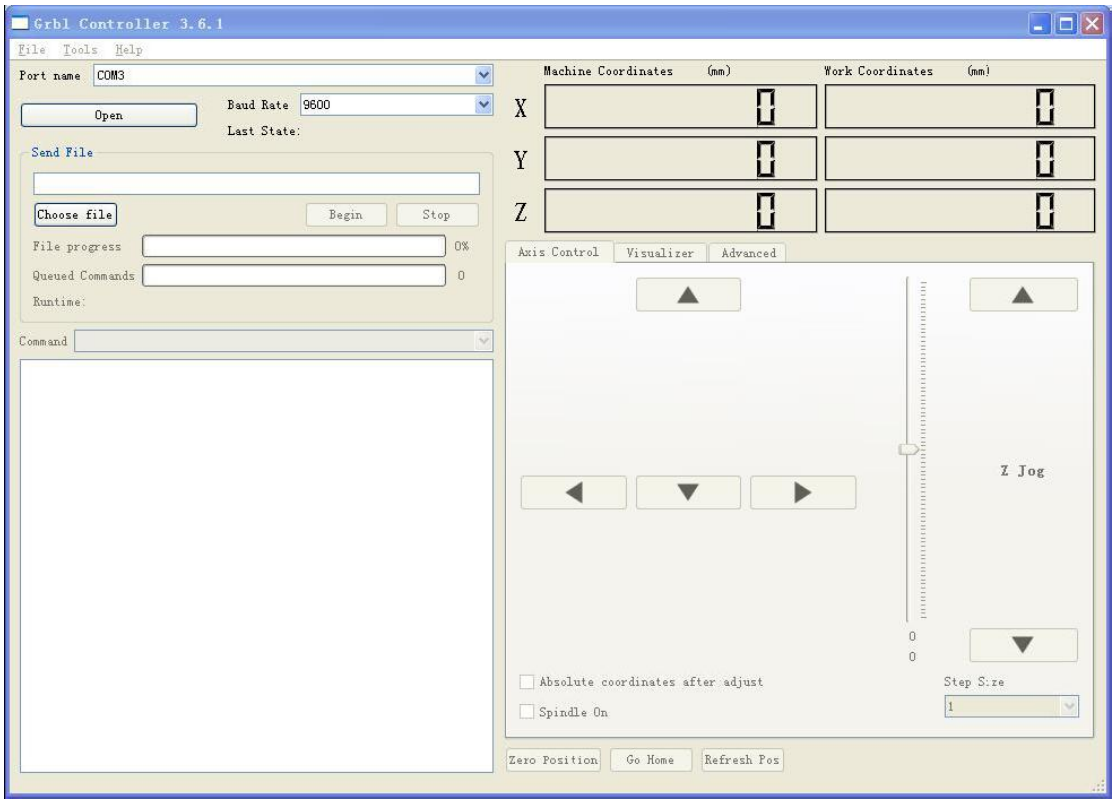
Click **Next**.

At last, click "**Finish**" to finish the installation.

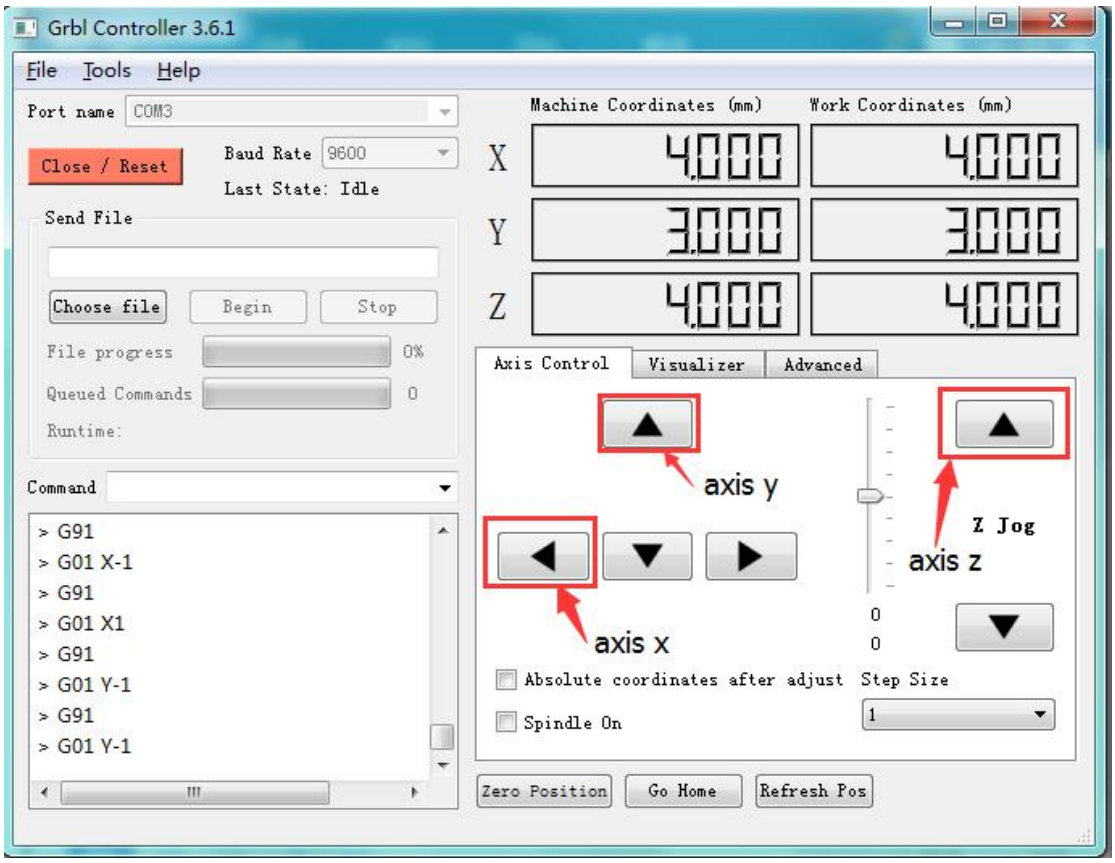
c. Test G-Code on Grbl Controller

Power the main board using a USB cable and connect correctly all your external devices, then run Grbl Controller.

Choose Port name the same as IDE COM port and click "Open" to open the serial port, connecting CNC Machines with computer.

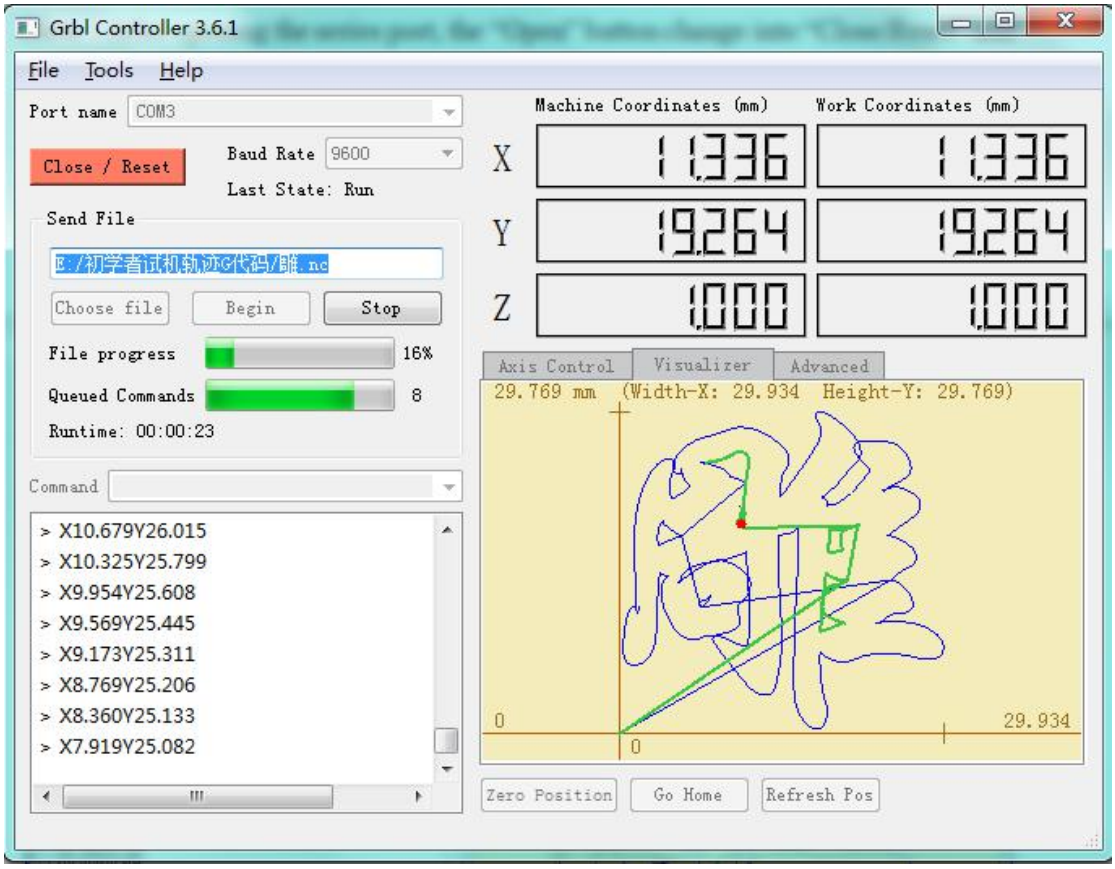


After opening the series port, the "Open" button change into "Close/Reset" and get red! At this time you can click the X axis、Y axis、 Z axis as shown in below diagram to adjust the motion direction of motors.



Note: after adjusting the axies, before beginning G-Code file, you must close and open again.

Now, it is time to have a try! Click ”Choose file” to choose one G-Code file named **cn.** to test in the data packet for a beginner, and the interface will come out:



Click "Begin" , and you can see how the motors move on coordinates.

10. Reference

- **All information download:**

https://drive.google.com/open?id=1t8YFL4sKQ6cQoSNN-R2LY02Rk-xGxQ8I

- **Arduino IDE Download:**

https://www.arduino.cc/en/Main/OldSoftwareReleases#1.5.x

- **GRBL_Arduino_Library_keyes (Firmware) Download:**

https://drive.google.com/open?id=1mmaspMTJAiPvFNpW5ihM-YqeTWKboTKt

- **Driver Download:**

https://drive.google.com/open?id=1IwohNN5tsgm3LzU0kdrBcekGQm--PoPf

- **G-Code for Testing Download:**

https://drive.google.com/open?id=1N5n5n5u4JIh2WLbFyW5VyBaabE2sZb84

- **Grbl Controller361 Setup Download:**

https://drive.google.com/open?id=1Ds6qJSpFfatAwthNaO-6sfFm_xbwCUDS