

STONE STVI056WT-01 + Arduino + Physical Therapy Instrument

Today is Friday, January 29, 2021, control of the COVID-19 in Beijing is also promoting the "New Year in place", family said the Spring Festival can not visit each other, or each person is fined 1500. It seems to go out as little as possible, if there is nothing to do you'd better to stay at home, do more projects, it is also considered to make a contribution to the country.

Just a few days ago the serial touch screen [STVI056WT-01](#) arrived, I am ready to use it to upgrade the physical therapy instrument! Now start drawing up a plan and steps.

1. This project is based on the upgrade of the old physical therapy instrument in order to show a better experience to customers.

- a. Add touch screen button function.
- b. Through communication with arduino serial port, realize the touch screen input to control the output of the product.

2. Upgrade the interface, add touch buttons to replace the silicone physical buttons and encoder on the right side of the product.

- a. Make English pictures with the location of the touch buttons.
- b. Import the picture into [Tool4.3](#), make and configure normal buttons, incremental buttons and return value buttons.

3. Connect STVI056WT-01 serial screen and arduino control board to realize programming control.

- a. Connect the serial screen to the MCU.
- b. Serial communication and control function programming.
- c. Online debugging.

Planning as above, the next step is to do the specific work and solve the problems that will be encountered in the process of doing it.

First, make English pictures with touch button locations

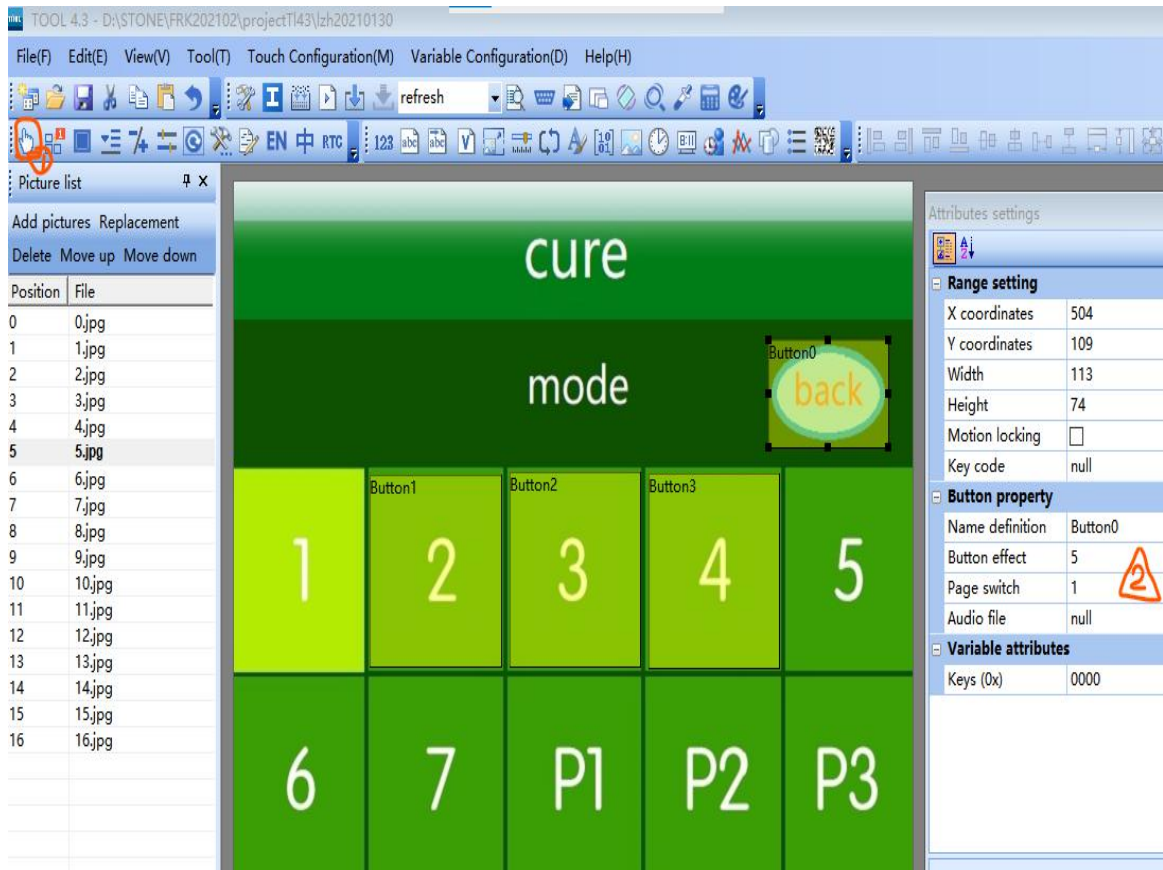
That is, add the necessary buttons in each interface, so that these buttons can achieve a reasonable jump between interfaces; there are also incremental buttons to adjust the output intensity instead of the encoder. Two examples are as follows.



Next, import the image into Tool4.3 and create three buttons

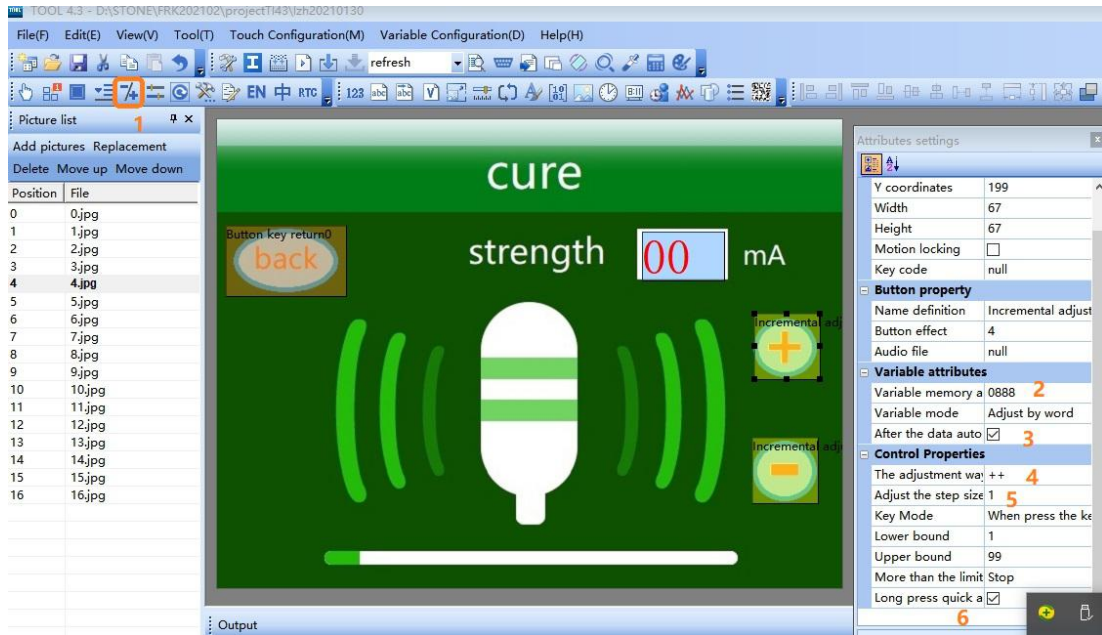
This will be done by continuing to watch the related videos and consulting the Tool software instruction manual that you downloaded last time. Of course, the point is to take good notes.

***General button.** In Tool4.3 Tools menu Touch Cnfiguration (M), select "Botton", or click on the button tool at "1" to create a button in the desired location, and fill in the "2" at the "botton effect" for the effective page location, "Page switch" for the jump page, you can complete the button jump function. The same button can be copied in different pages.



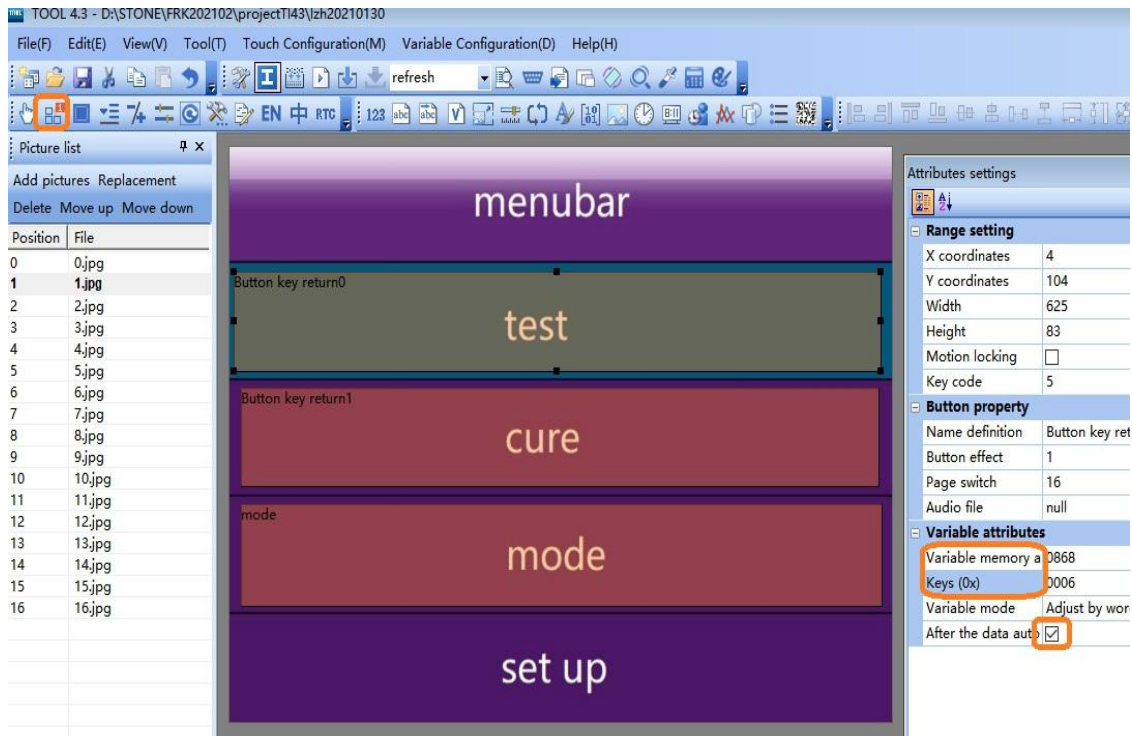
(Tool4.3 tool software to create a general button parameters chart)

****Incremental button.** Check "Incremental adjustment" in the Tool4.3 menu Touch Configuration (M) or click on the button tool at "1" in the figure to create a button at the desired location and fill in the "variable memory address" at "2" so that it is the same as the data variable address (0X0888); check the box "after the data automatically upload" at "3" to automatically upload changes to the serial port of the arduino development board; select "the adjustment way" as "+" or "--" at "4" ; in "5", fill in "adjust the step size" with a step value of 1; in "6", check the box "long press quick adjust" to achieve long press quick adjustment by touching the button.

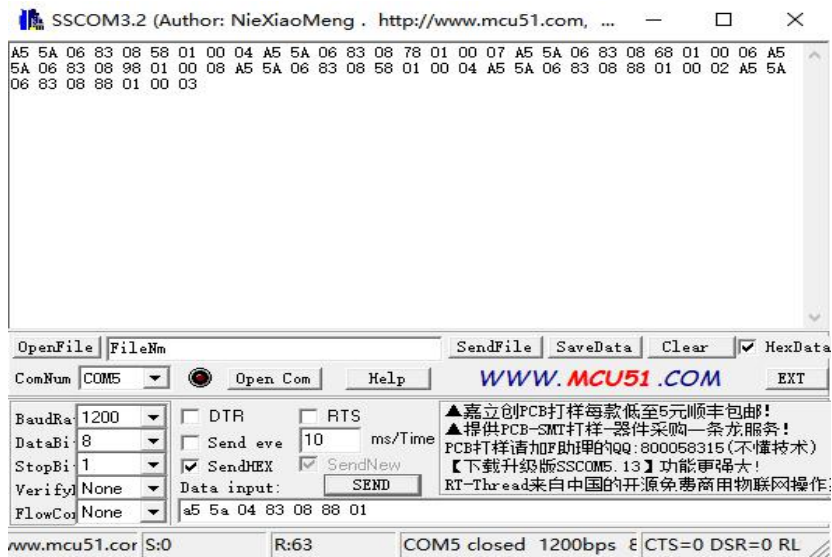


(Tool4.3 tool software to create incremental button parameters chart)

***** Return value button.** Check "Return pressed key value" in the menu Touch Cinfigration (M) of Tool4.3 tool software, or click on the button tool in the picture as above, create a button in the desired location, and fill in the variable address and key value in the marker as above. Check the box to automatically upload the return value when pressed. The "test" button variable address is 0X0868, and the key value is changed to "0055" (0x0055); the "cure" button variable address is 0X0858, the key value is changed to "0066" (0x0066); the key test serial assistant returns the following values.



(Tool4.3 tool software to create a return value button parameters chart)

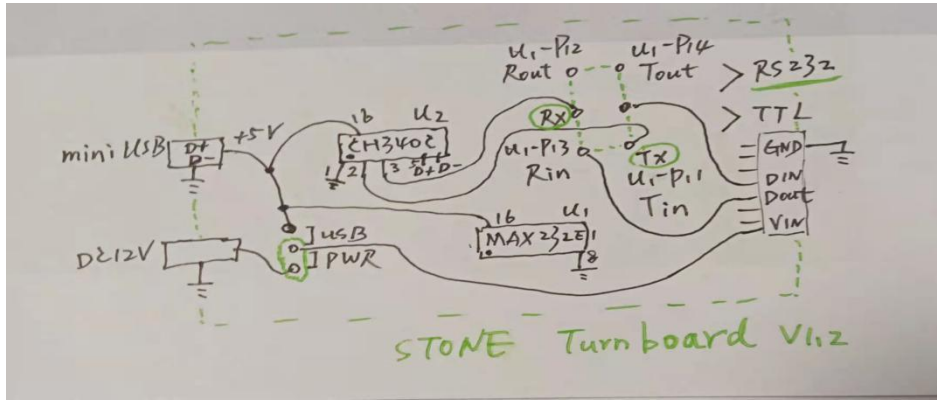


(Keystroke return value and incremental data 0X0888 return value screenshot)

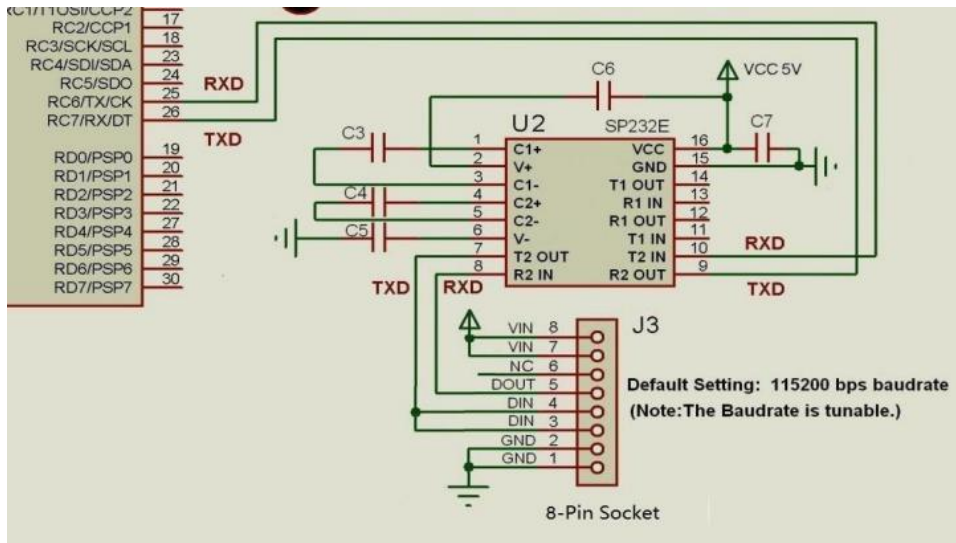
Next, connect the serial screen to the MCU.

By experience simple connection [arduino](https://www.arduino.cc/) development board TX, RX, GND to the serial screen DIN, DOUT, GND, properly "enjoy" a failure.

Quickly check the "Datasheet STVI056WT-01" pdf electronic file manual, in page 31 to find the serial screen circuit diagram, found that the screen interface DIN, DOUT signal are RS232. Also check the "LY-F2 development board circuit diagram", it is obvious that arduino development board TX, RX is directly from the microcontroller out of the TTL signal. The reason for failure is obvious, how to solve the problem? By the way, STONE comes with an adapter board, I was looking for a long time, but also has not seen the circuit diagram of the adapter board, so I drew a circuit diagram(manual drawing, because it is relatively simple, I did not start protel99SE, PADS2000, AltiumDesigner 10, Cadence or AUTOCAD2010, etc.software, my computer is slow, so that I can save some time). From the drawing I found a solution to the problem. The adapter board has a MAX232E1 chip, and the TX, RX signal lead out through the jumper , it plays a conversion role, so that arduino development board serial TTL and STVI056WT-01 serial screen RS232 signal can match each other. From the drawing also know that the STONE adapter board DC12V power supply is supplied to the serial screen, the internal MAX232E1, CH340C power supply +5V are provided by the USB, so when using RS232 to serial TTL level conversion, the USB needs to be powered. For the adapter board MAX232 chip, Tin, Rout are 0-5V, while Tout, Rin pin is + - 15V logic. Screenshots are as follows.



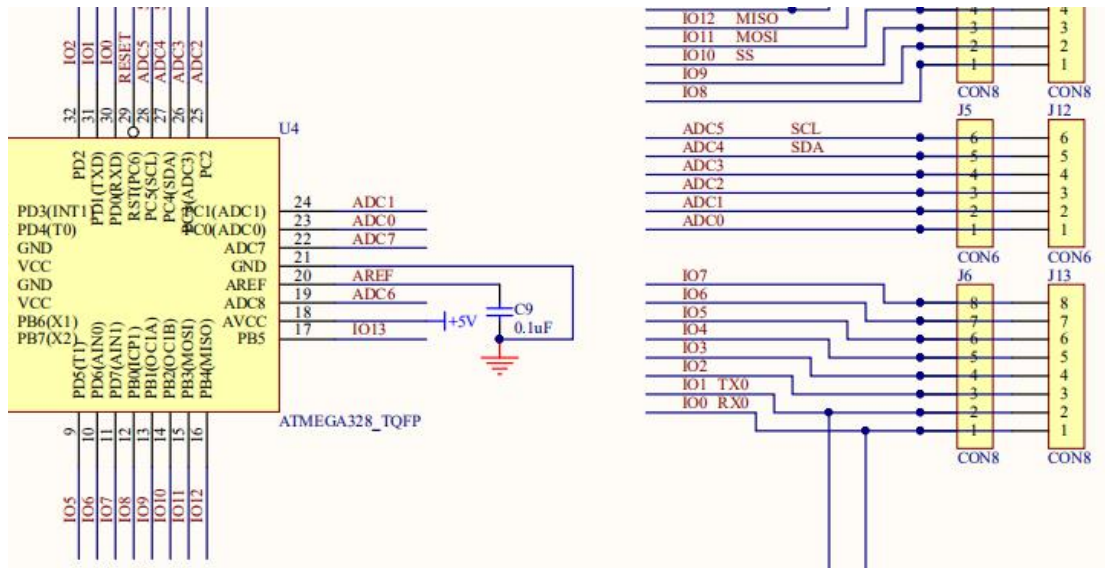
(My hand-drawn STONE adapter board wiring connection diagram)



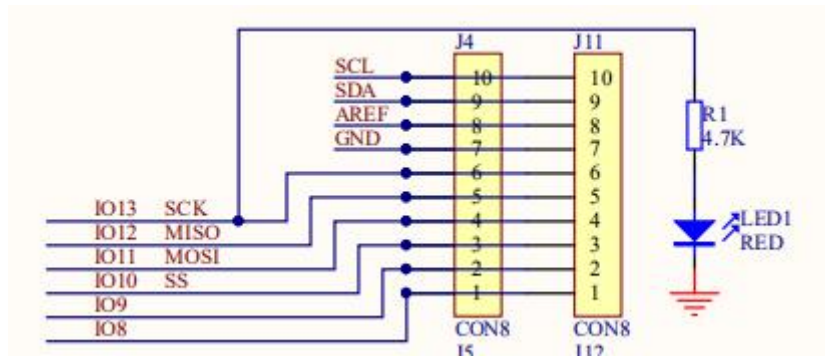
(Screenshot from Datasheet STVI056WT-01 pdf e-file manual, page 31)

As you can see in the picture, the serial screen comes out at RS232 level, and since RS232 is $\pm 15V$ logic, it is able to transmit farther!

Check the "LY-F2 development board circuit diagram", screenshot as follows.



(arduino development board serial line TX, RX pinout diagram)



(Green LED pinout for indicating treatment output - and test output IO9 servo control interface pinout)

Then secondly, the serial communication and control functions are programmed.

The arduino development board was selected to communicate with the STVI056WT-01 serial screen. I have already done serial communication with STVI056WT-01 screen before, and it happens that I used arduino development board some time ago to finish a project of servo-controlled ticket box automation, and the result was quite good, and it solved the

pain point of the demand side in time. The credit goes to arduino! This is not a lie, because arduino is not only good, universal, and ready, cheap, do not need layout, do not need to play the board, do not need to find parts welding, simple development environment, but also comes with a variety of routines, it is very good to use! (This is not advertise for arduino) The following directly show the real goods.

To the "arduino language instructions" focus on the application of part of the serial port transceiver function instructions excerpted as follows.

Serial port send/receive functions

- `Serial.begin(speed)` : Serial port definition baud rate function, speed means baud rate, such as 9600, 19200, etc.
- `int Serial.available()` : Determine the buffer status.
- `int Serial.read()` : Read the serial port and return the received parameters.
- `Serial.flush()` : flushes the buffer.
- `Serial.print(data)` : Output data from the serial port.
- `Serial.println(data)` : Output data with carriage return.

In the 232-page "Getting.Started.with.Arduino" pdf full English electronic document, there is a more detailed description of the instructions, screenshot saved below.

Serial.begin(speed)

Prepares Arduino to begin sending and receiving serial data. You'll generally use 9600 bits per second (bps) with the Arduino IDE serial monitor, but other speeds are available, usually no more than 115,200 bps.

Example:

```
Serial.begin(9600);
```

(it is good to use arduino serial port maximum baud rate setting)

As you can see, the arduino can support serial screen baud rate of 115200, which is also the highest value that can be chosen for this time.

Serial.print(data) **Serial.print(data, encoding)**

Sends some data to the serial port. The encoding is optional; if not supplied, the data is treated as much like plain text as possible.

Examples:

```
Serial.print(75);           // Prints "75"
Serial.print(75, DEC);      // The same as above.
Serial.print(75, HEX);     // "4B" (75 in hexadecimal)
Serial.print(75, OCT);     // "113" (75 in octal)
Serial.print(75, BIN);     // "1001011" (75 in binary)
Serial.print(75, BYTE);    // "K" (the raw byte happens to
                          // be 75 in the ASCII set)
```

(use arduino serial port send data command)

Serial.println(data) **Serial.println(data, encoding)**

Same as *Serial.print()*, except that it adds a carriage return and linefeed (*\r\n*) as if you had typed the data and then pressed Return or Enter.

Examples:

```
Serial.println(75);        // Prints "75\r\n"
Serial.println(75, DEC);  // The same as above.
Serial.println(75, HEX);  // "4B\r\n"
Serial.println(75, OCT);  // "113\r\n"
Serial.println(75, BIN);  // "1001011\r\n"
Serial.println(75, BYTE); // "K\r\n"
```

(use arduino serial port send data command with carriage return character)

After testing with the serial assistant, the two print functions above transmit in character form. For example.

```
Serial.print(75,HEX);
```

The string transmitted is "4B", which is the ASCII value of the characters "4" and "B" (0x34, 0x42); therefore, serial. print() function, whether string, or data, are sent ASCII, not suitable for command transmission!

Fortunately, in arduino 1.0.5, under the menu help, click on "Reference" item, it will pop up the function description of each function displayed by category, among which the latest serial function is serial.write(Val), screenshot as follows.

write()

Description

Writes binary data to the serial port. This data is sent as a byte or series of bytes; to send the characters representing the digits of a number use the `print()` function instead.

Syntax

Serial.write(val)
Serial.write(str)
Serial.write(buf, len)

Arduino Mega also supports: Serial1, Serial2, Serial3 (in place of Serial)

Parameters

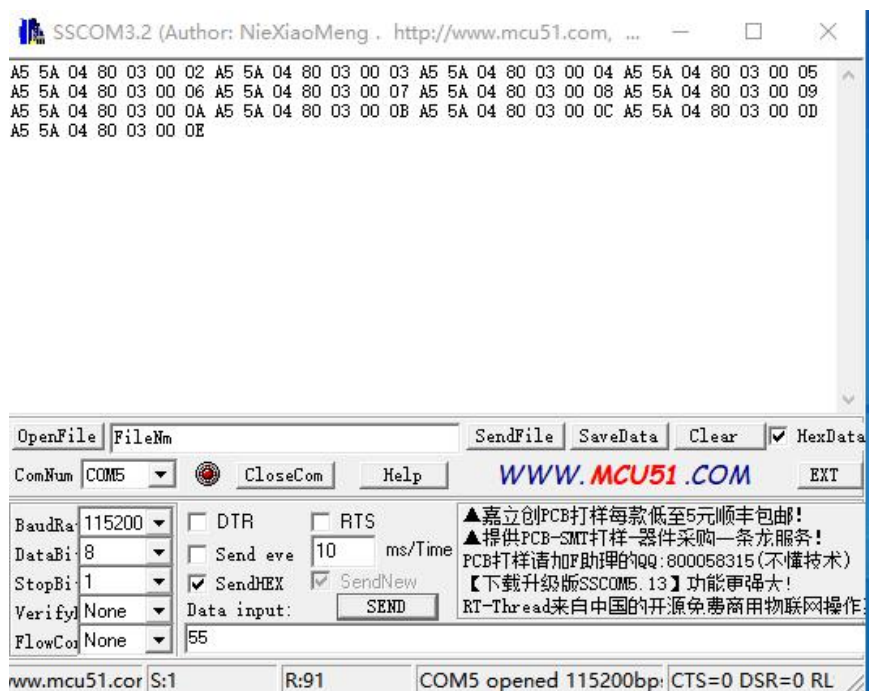
val: a value to send as a single byte

str: a string to send as a series of bytes

buf: an array to send as a series of bytes

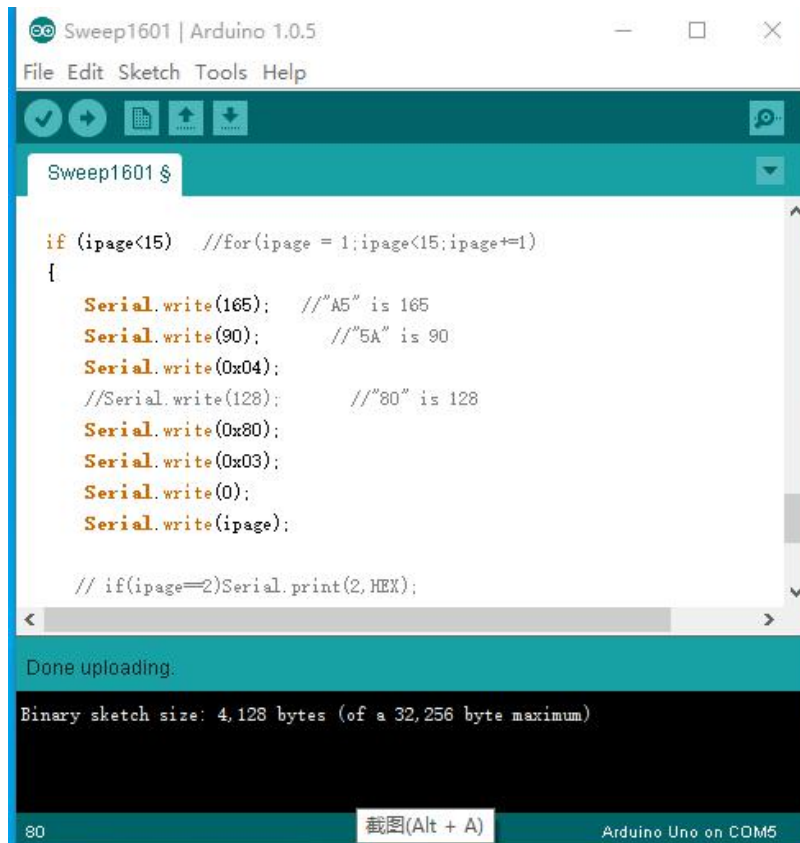
len: the length of the buffer

The function transmits the binary code, and the test transmission is correct and can jump pages correctly; as follows.



(The effect of the serial screen command transmitted using the serial.write(Val) function)

Moreover, the `serial.write(Val)` function supports writing hexadecimal directly, such as `serial.write(0x80)`, which compiles and uploads successfully, as shown below. Therefore, the `serialwrite()` function sends the value VAL in binary and the string in ASCII.



int Serial.available()

Returns how many unread bytes are available on the Serial port for reading via the `read()` function. After you have `read()` everything available, `Serial.available()` returns 0 until new data arrives on the serial port.

Example:

```
int count = Serial.available();
```

(use arduino serial port status query command)

int Serial.read()

Fetches one byte of incoming serial data.

Example:

```
int data = Serial.read();
```

(use arduino serial port read command)

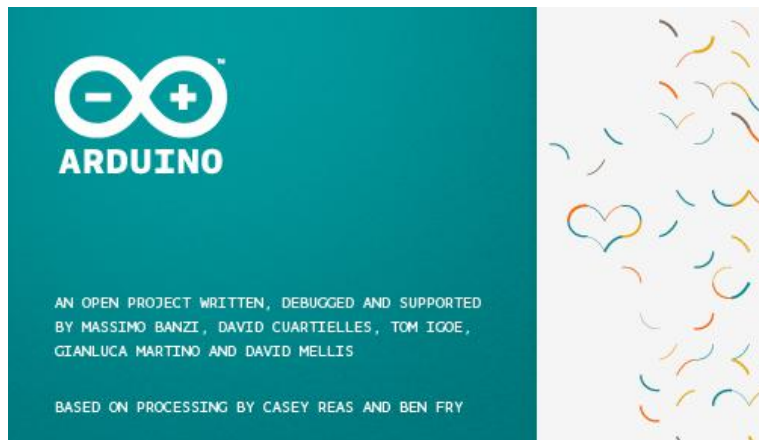
For the arduino development environment, just run the `arduino.exe`

directly under the arduino-1.0.5 package directory.

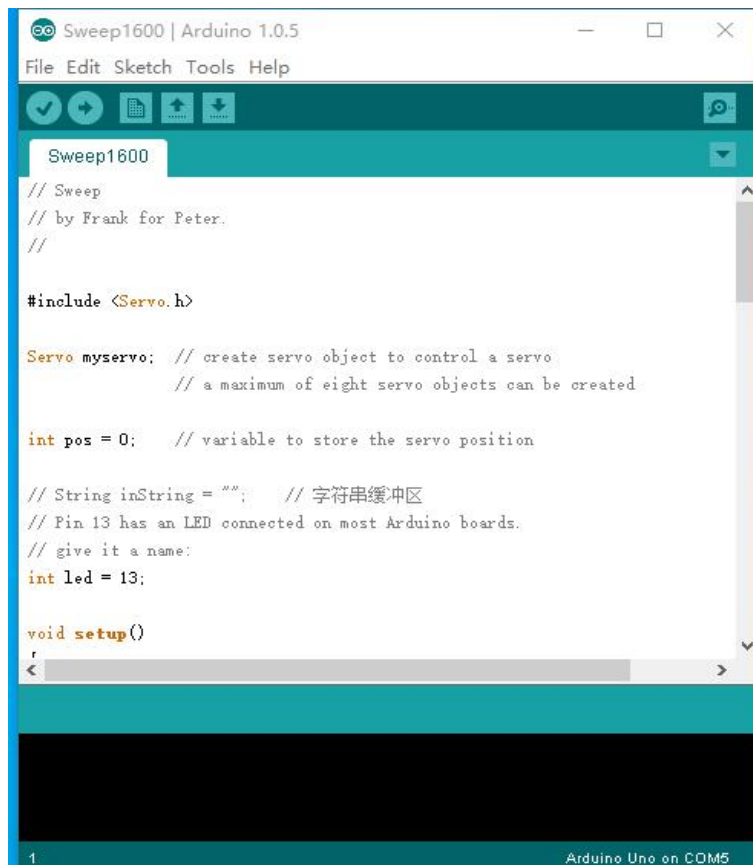
新加卷 (D:) > arduinoAVR > 德飞莱 增强版arduino资料 > arduino-1.0.5

名称	修改日期	类型
drivers	2020/4/16 星期四 1:12	文件夹
examples	2020/4/16 星期四 1:12	文件夹
hardware	2020/4/16 星期四 1:12	文件夹
java	2020/4/16 星期四 1:15	文件夹
lib	2020/4/16 星期四 1:16	文件夹
libraries	2020/4/16 星期四 1:16	文件夹
reference	2020/4/16 星期四 1:17	文件夹
tools	2020/4/16 星期四 1:17	文件夹
arduino.exe	2013/5/17 星期五 22:...	应用程序
cygconv-2.dll	2013/5/17 星期五 22:...	应用程序扩展
cygwin1.dll	2013/5/17 星期五 22:...	应用程序扩展
libusb0.dll	2013/5/17 星期五 22:...	应用程序扩展
revisions.txt	2013/5/17 星期五 22:...	文本文档
rxtxSerial.dll	2013/5/17 星期五 22:...	应用程序扩展

After double-clicking, wait for a while, the following startup screen will appear



(arduino development environment startup interface)



The image shows a screenshot of the Arduino IDE interface. The window title is "Sweep1600 | Arduino 1.0.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for checking, running, uploading, and downloading. The main text area displays the following code:

```
// Sweep
// by Frank for Peter.
//

#include <Servo.h>

Servo myservo; // create servo object to control a servo.
               // a maximum of eight servo objects can be created.

int pos = 0;   // variable to store the servo position

// String inString = ""; // 字符串缓冲区
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

void setup()
{
```

The status bar at the bottom indicates "1" and "Arduino Uno on COM5".

(arduino development environment to open Sweep1600.ino file interface)