**Getting Started with Mixly 1.0**

# **Download**

There are two kind installation packages provided for you in the following links:

[Mixly1.0 for Win 7/8/10](http://116.62.49.166/Mixly_WIN.7z)：<https://fs.keyestudio.com/Mixly1-Windows>

[Mixly1.0 for Mac](https://116.62.49.166/MixlyMac.zip)：<https://fs.keyestudio.com/Mixly1-MACOS>

## **Windows version：**

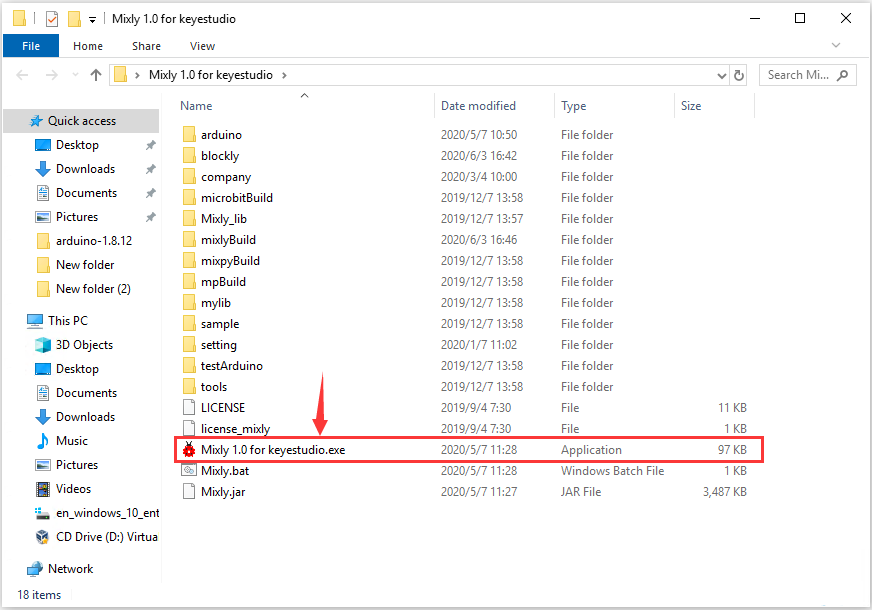
We will take Mixly1.0 (Windows version) as example.

## **Install Software：**

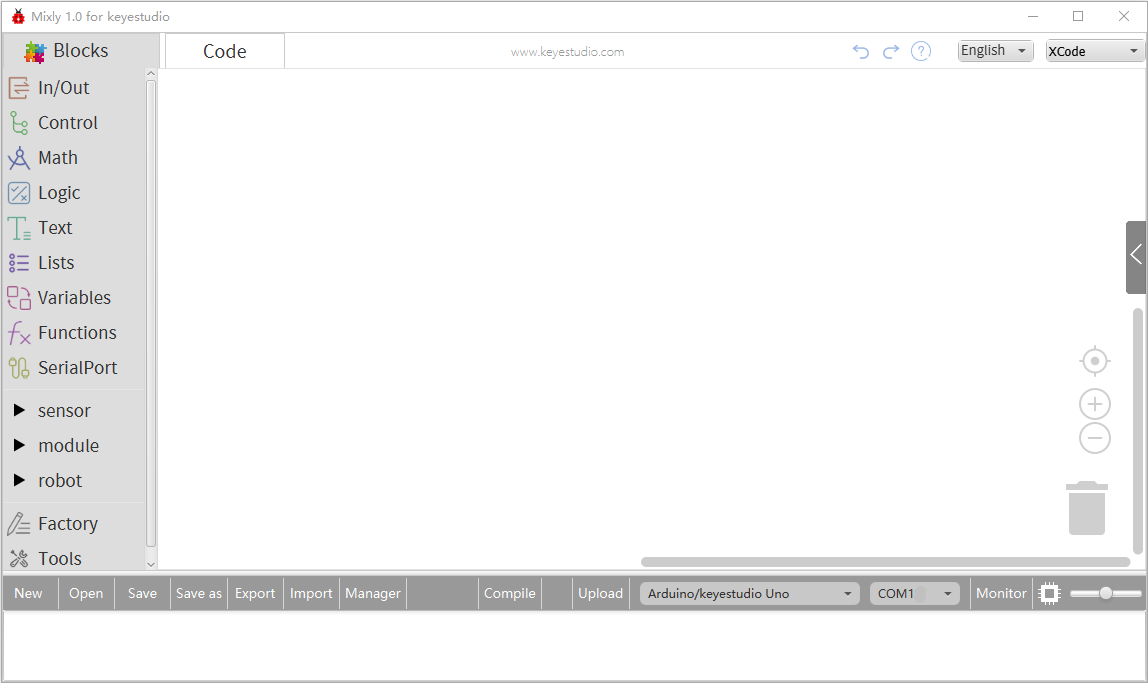
You will get installation package after downloading. As shown below:



Unzip the package, you will see “Mixly 1.0 for keyestudio.exe”

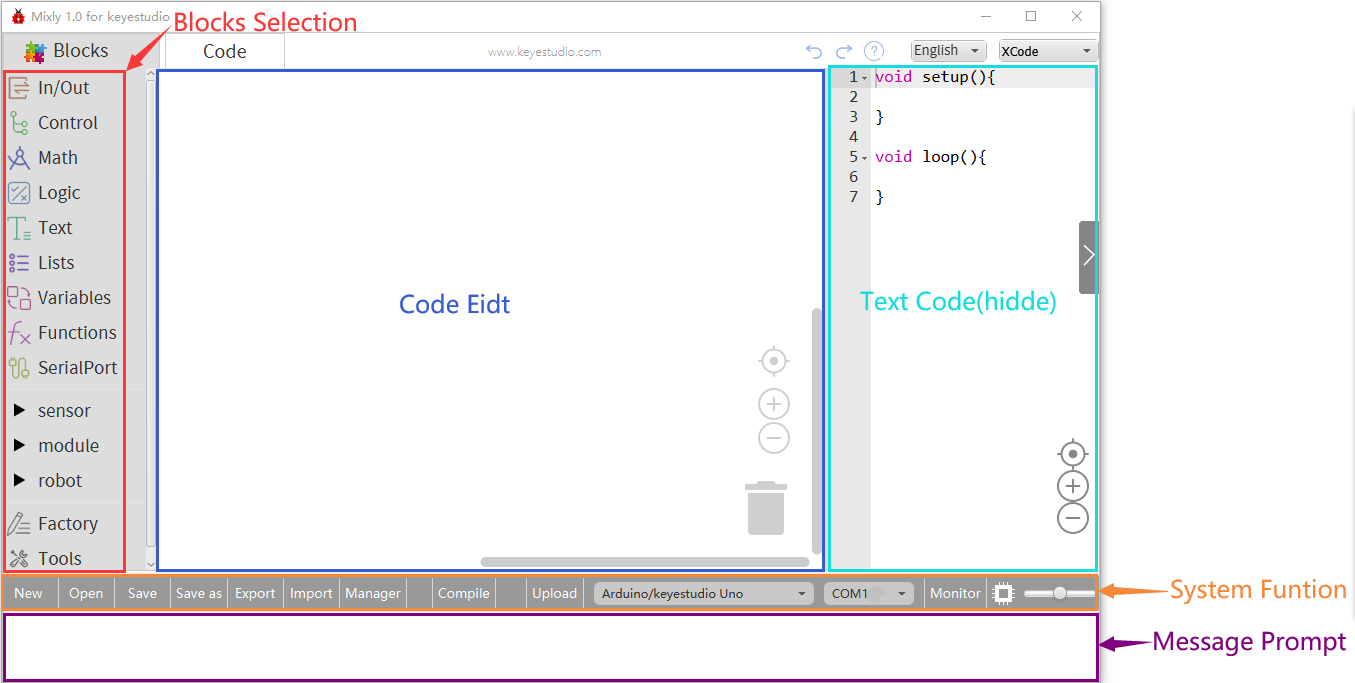


Double-click ”Mixly 1.0 for keyestudio.exe”, the following interface pops up.

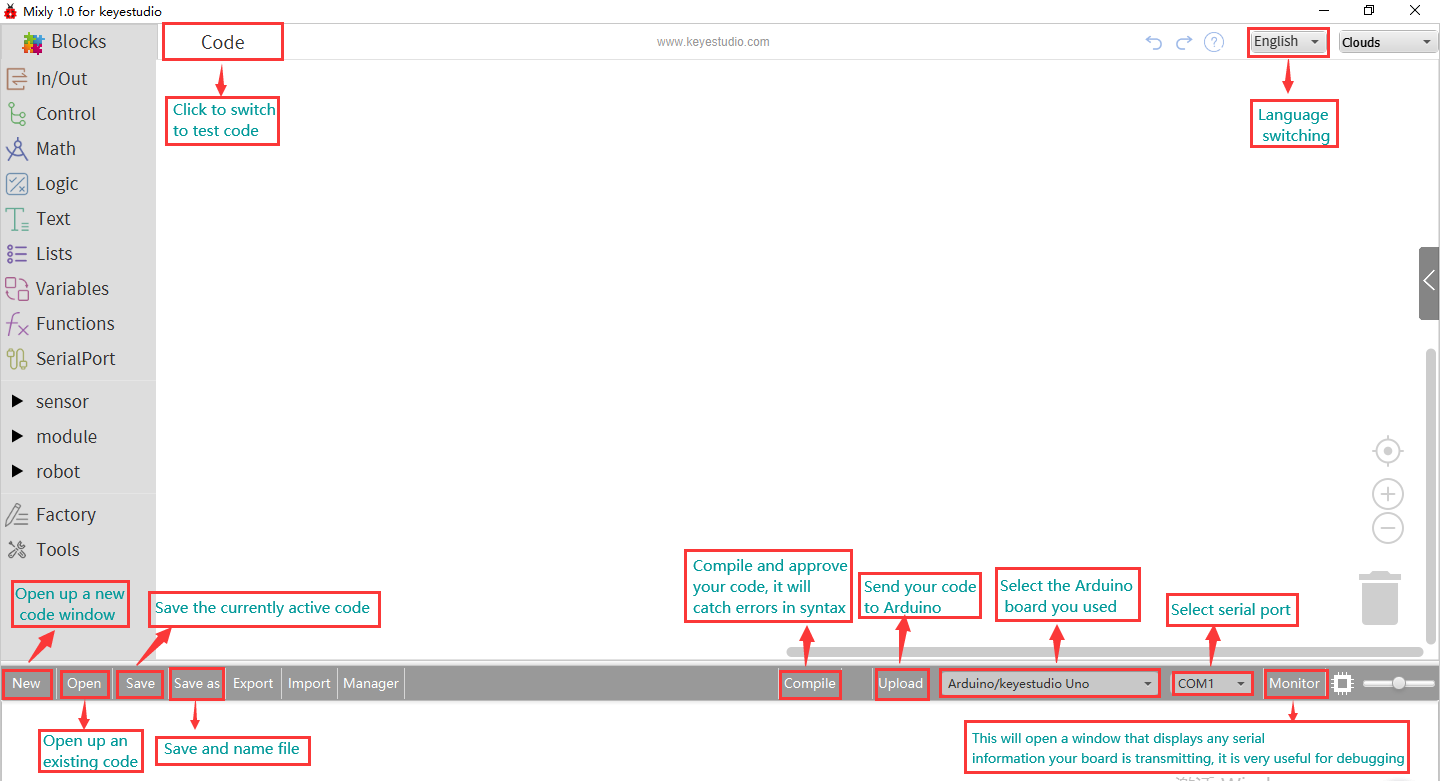


# Interface Instruction**：**

Look at the main interface of Mixly 1.0, it includes five parts, that is, blocks selection, code edit, text code (hidden), system function and message prompt area. As shown below.

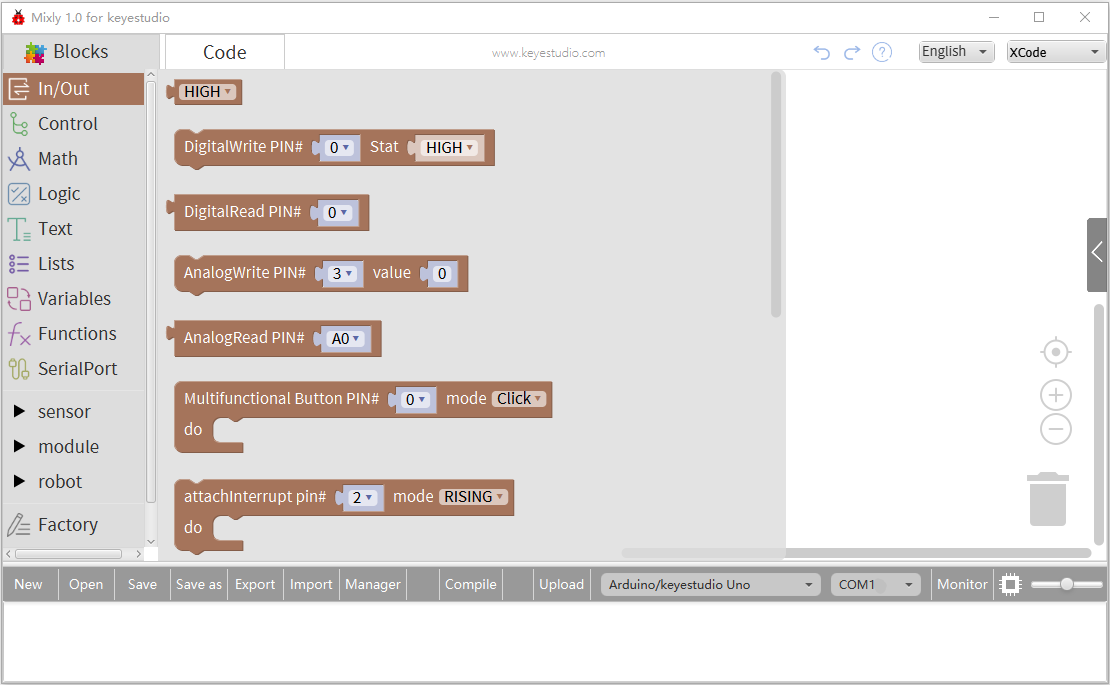






## Blocks selection

There are different blocks in the block selection area, you could finish programming via them.



## Code Edit Area



Design program by combine the blocks you need.

## Delete program：

1. Drag the useless code to the recycle bin at the lower right corner
2. Move the code to the block selection area
3. Right click the block to delete, or tap “Delete” and “Backspace” on keyboard.
4. **Zoom in and out program**  ()

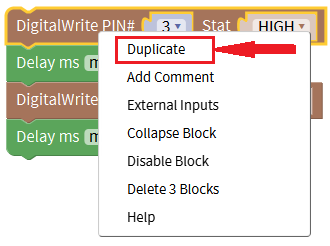
Of course, you could zoom in and out program by the scroll wheel of mouse

：Center the program

: Zoom in program of code edit area

: minify program of code edit area

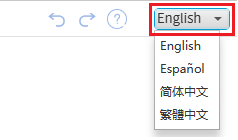
## Copy program：

Right click to select “Duplicate” and you can copy an instruction block. 

Select program via mouse, then press “Ctrl+C” and “Ctrl+V” to replicate block.

## Switch Language

Mixly 1.0 supports four languages: English、Español (Spanish)、中文简体(Chinese Simplified)、中文繁体(Chinese Traditional).



: undo, we could get the deleted block back by this icon (or press Ctrl + Z）when deleting blocks mistakenly.

: redo, click this icon or press Ctrl + Y key to go back to the previous step

1. **Assistant document：**

: click this icon to enter the page that contains the detailed information and introduction about Mixly software

## System function area:

As shown below:



General function: New, open, save and save as function.

The saved format of the code is .mix.



Enable Mixly 1.0 software and click “Open” to find \*.mix file to reach the saved document, or you could double-click \*.mix file or drag it to the code edit area.

**Compile & Upload:**



After programming, click “Compile” to check errors in the code

If display “Compile failed!”, you need to look over if code is wrong

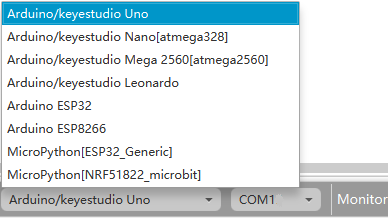
If popping up “Upload failed!”, unplug and connect USB cable to address this problem.

If “Upload success” appears, which means the code is uploaded successfully on the board.

**Control board & COM port：**



Click drop-down triangle to see the options for distinct control boards, select the board you use.



Next to choose the corresponding COM port which is the communication path between PC and control board.



Serial monitor：



Monitor only works with the blocks of serial port, it is used to display the output variables and value of sensors.

**Enlarge page**：



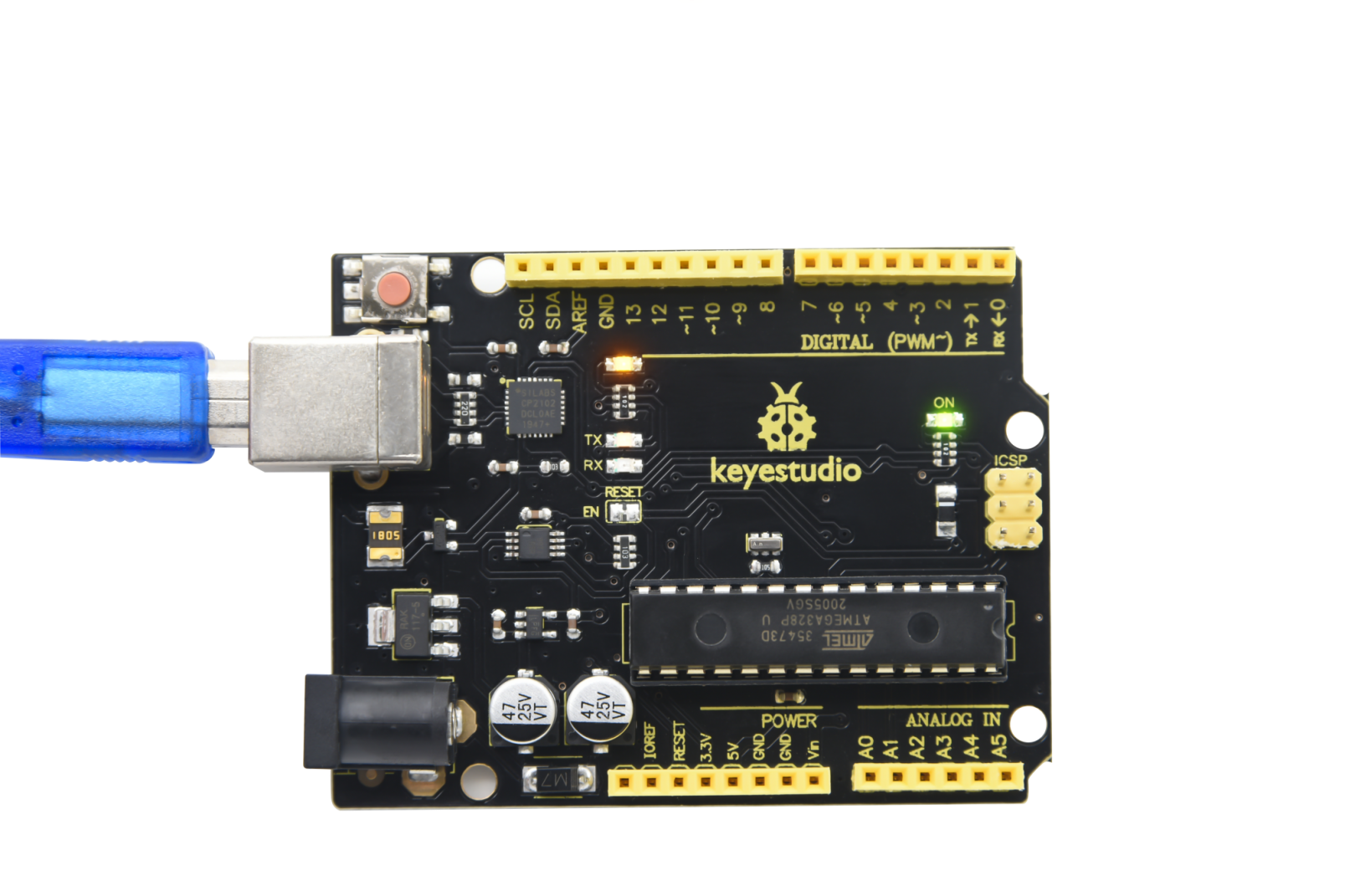
There is a scroll button beside the monitor, drag this button to magnify the whole page.

# 3. Start your first program---Saying Hello World！

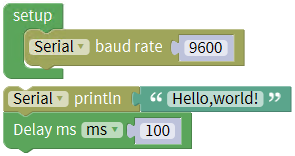
Next, we will take example as Arduino V4.0 control board：

1. **Wire-up Guide:**

Link control board with computer by USB cable

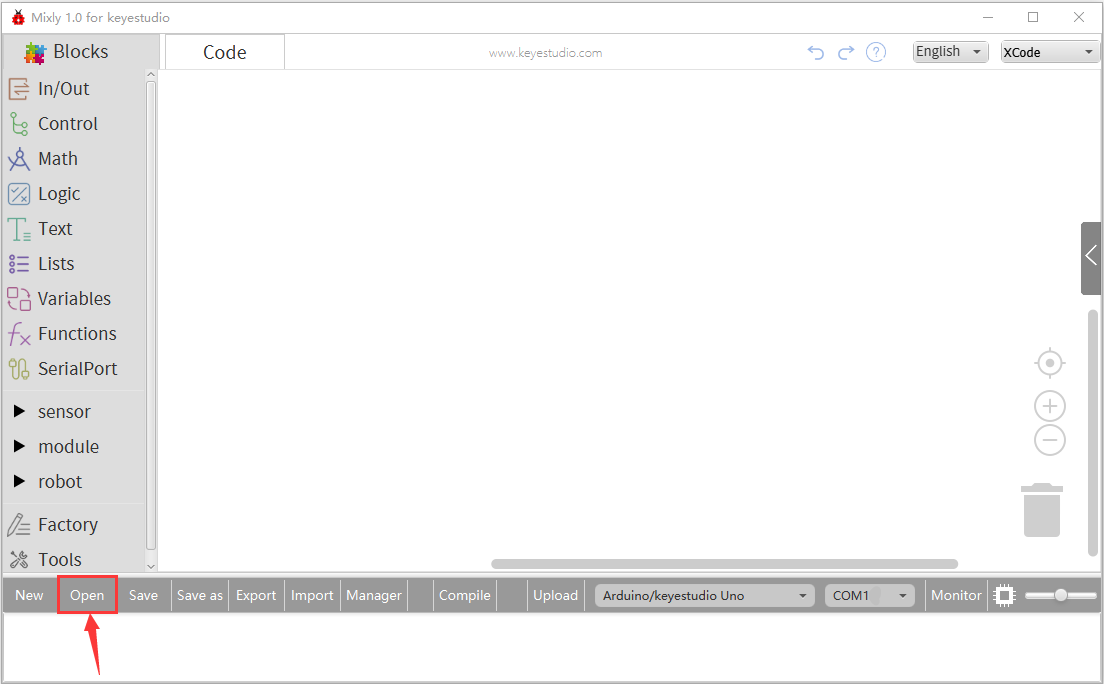


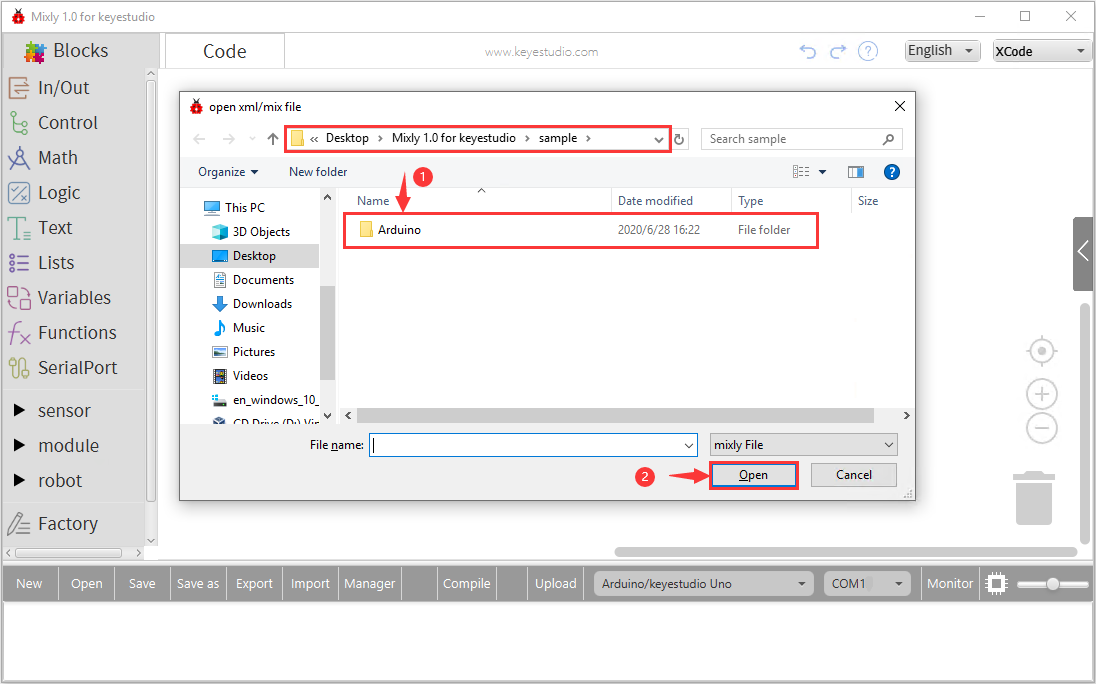
1. **Open Program Code**

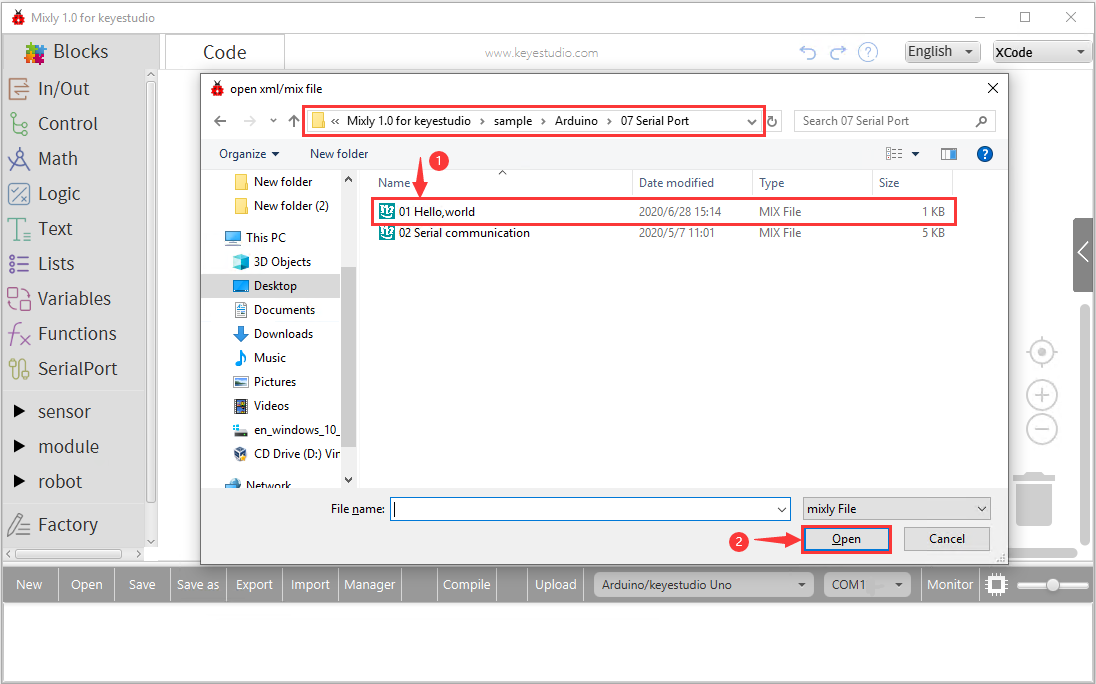
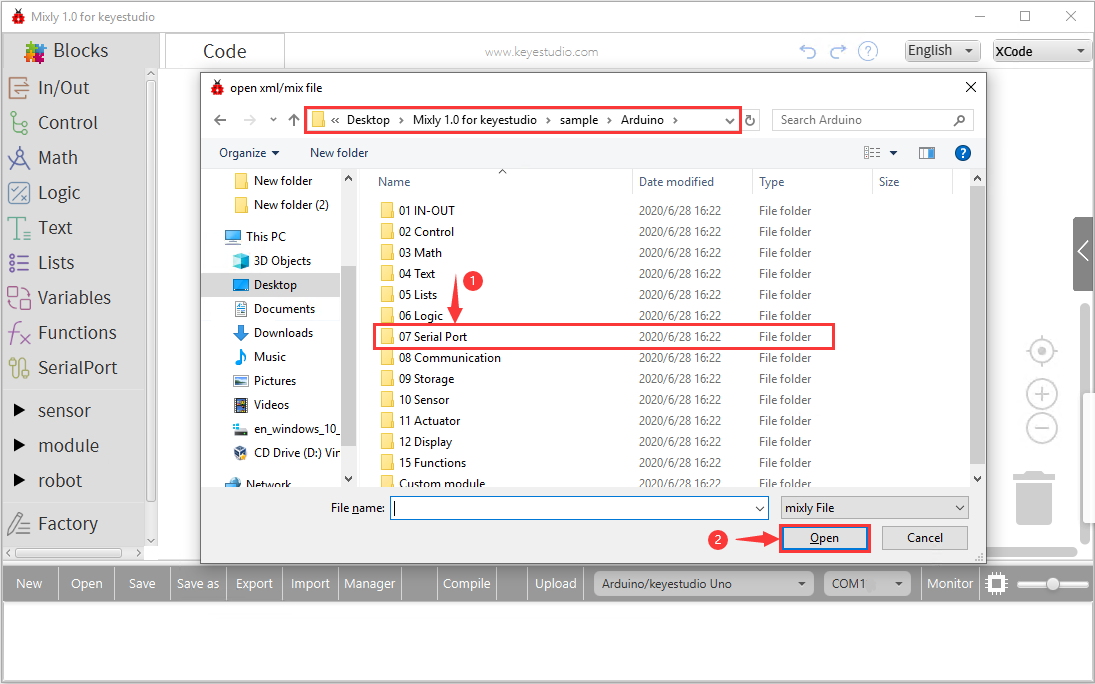


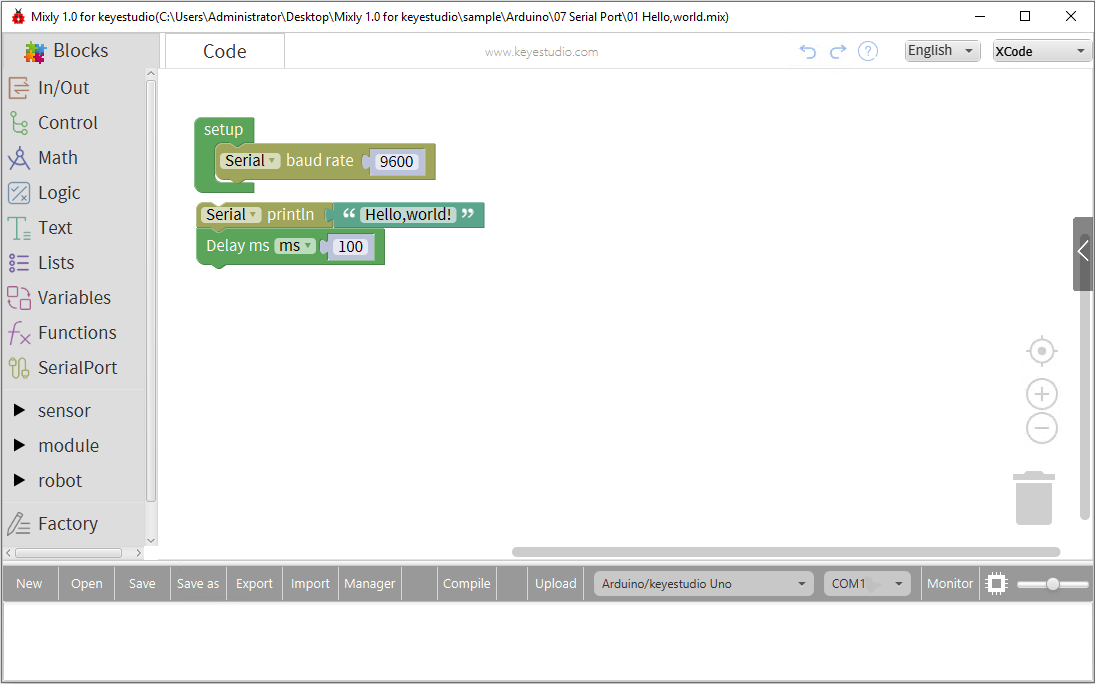
**The sample code** ”**Hello World” is provided for you, you could directly open it**

1. Click”Open” to find sample code“Hello World”, as shown below:

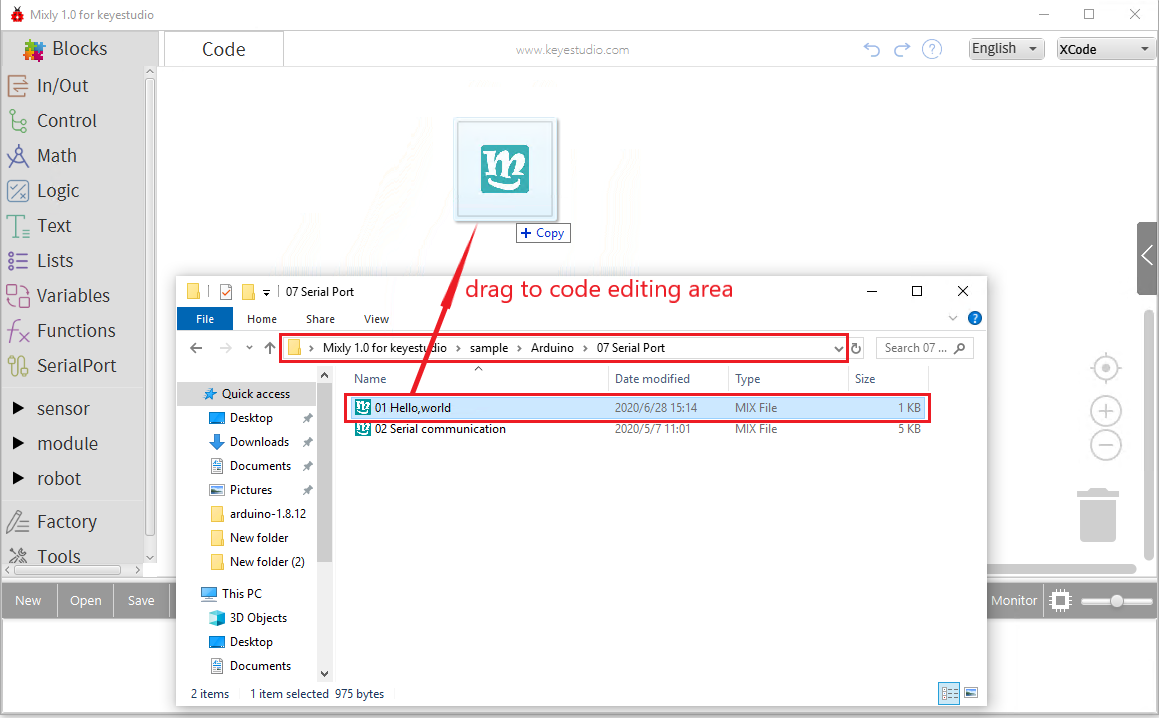


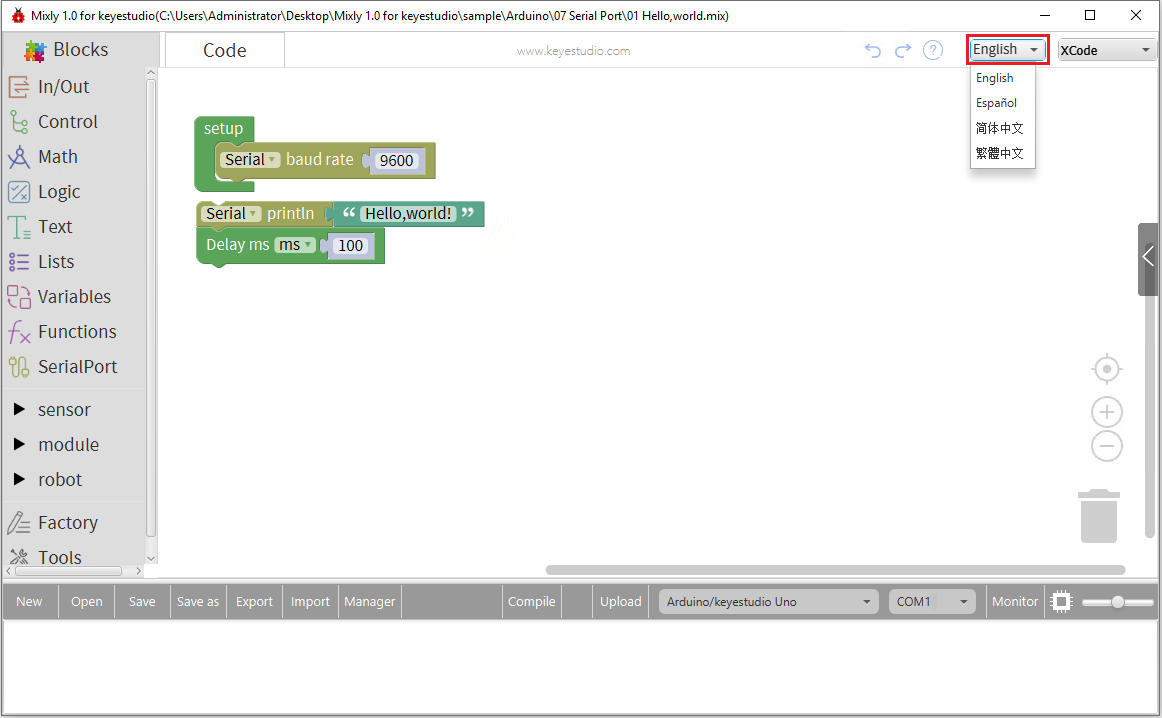




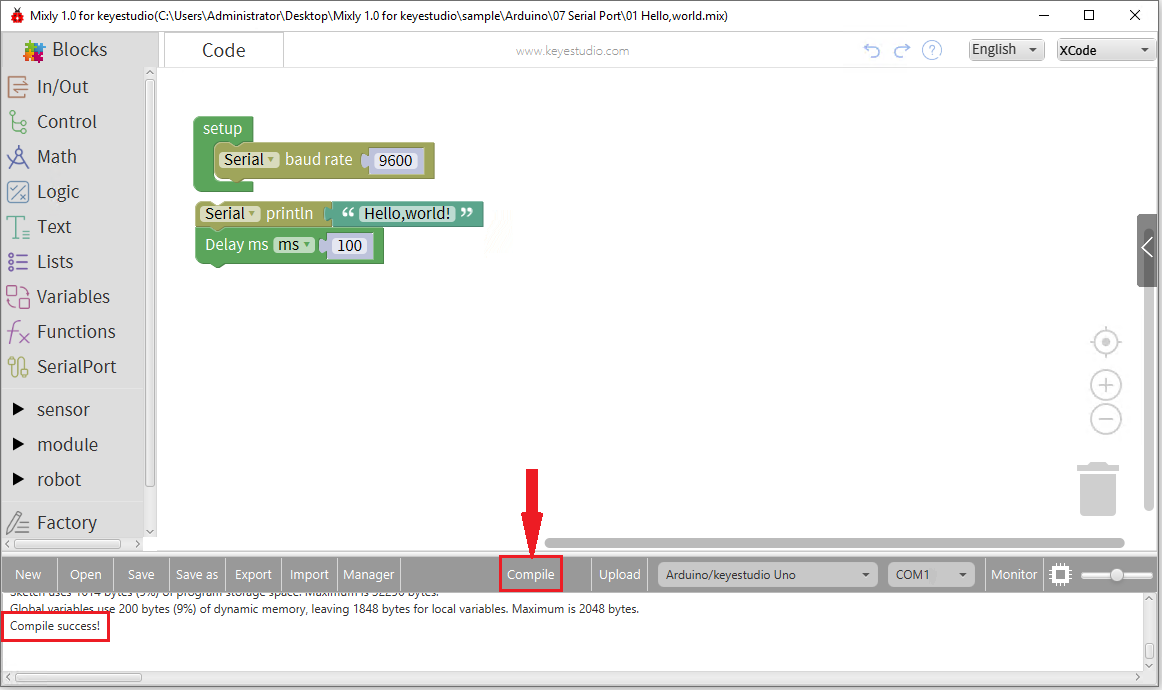


In fact, open it directly( click “Mixly 1.0 for keyestudio”package ->“sample”->“Arduino”->07 Serial Port ->“Hello World”) and drag to code edit area, then switch language.

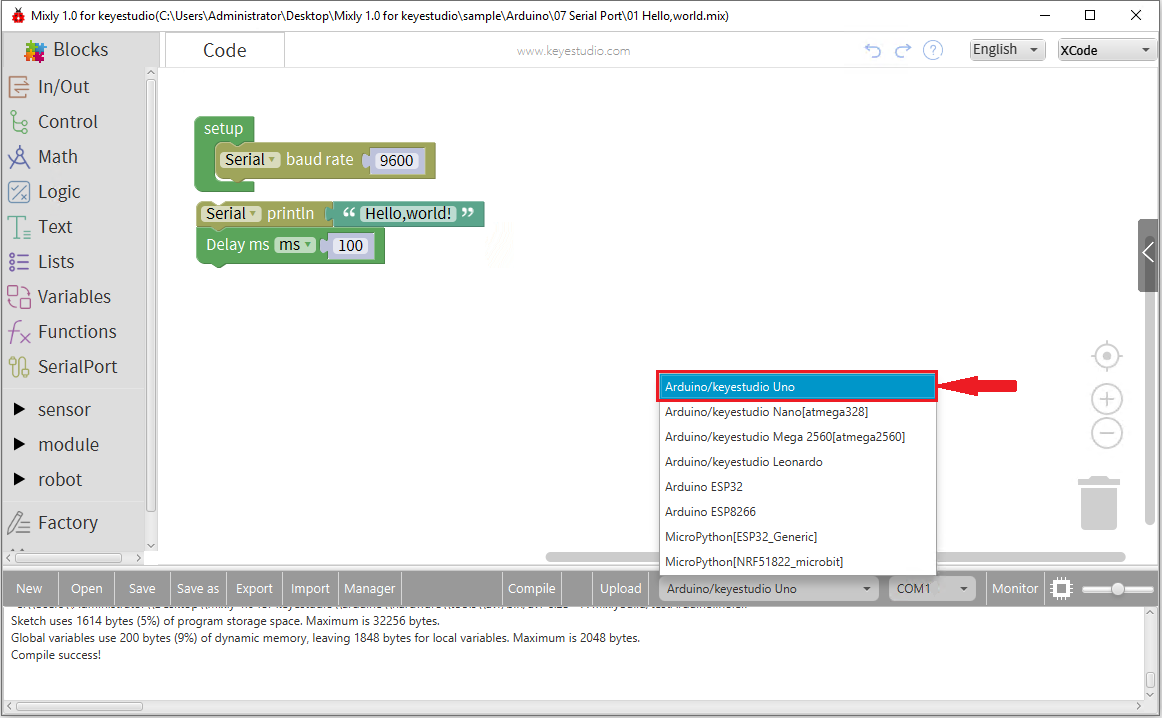


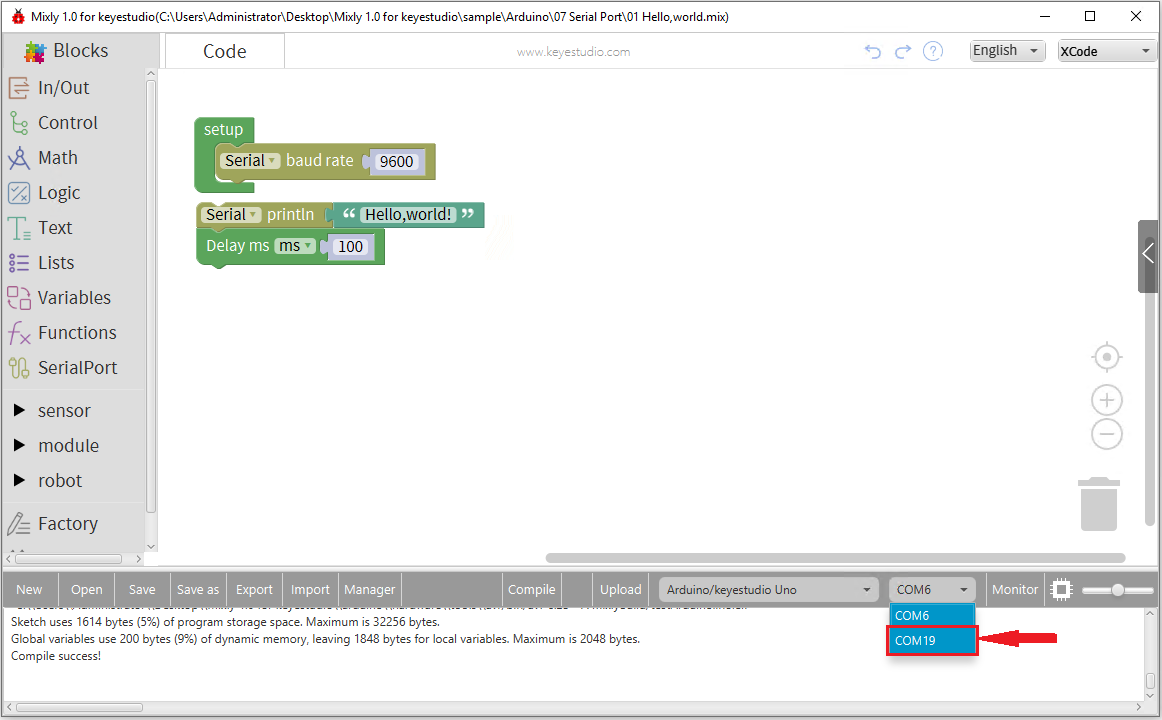


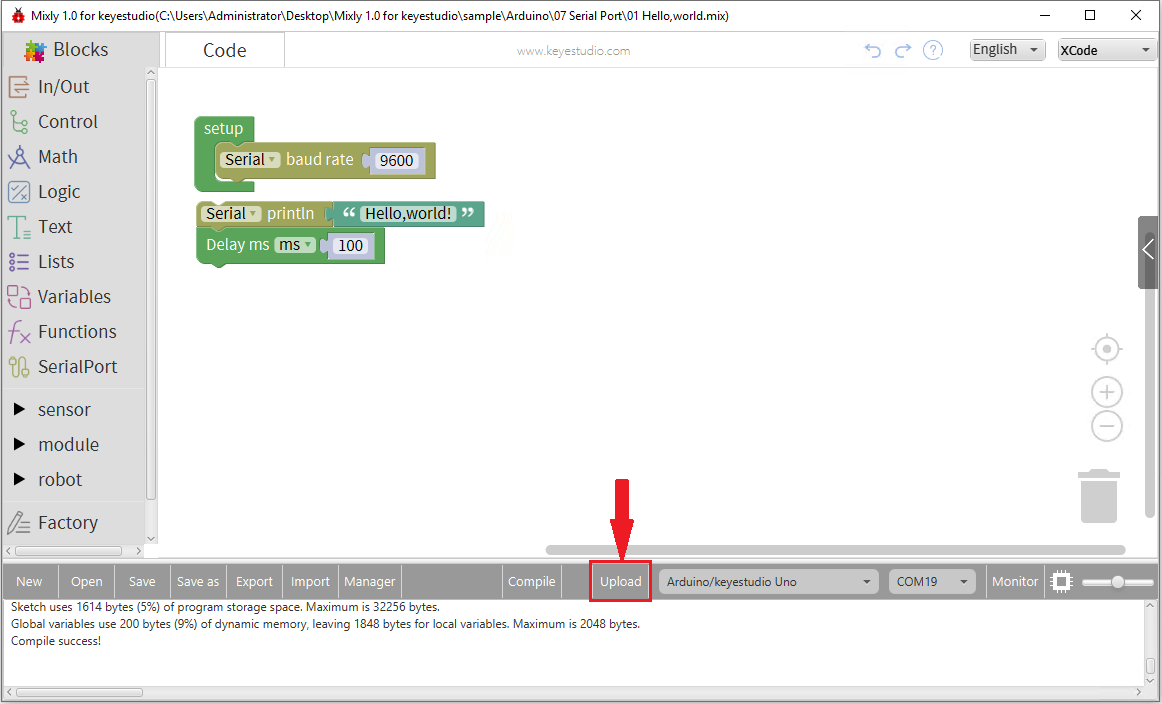
1. Then tap “Compile”, if compile successfully, the code is approved well, enter the next step.



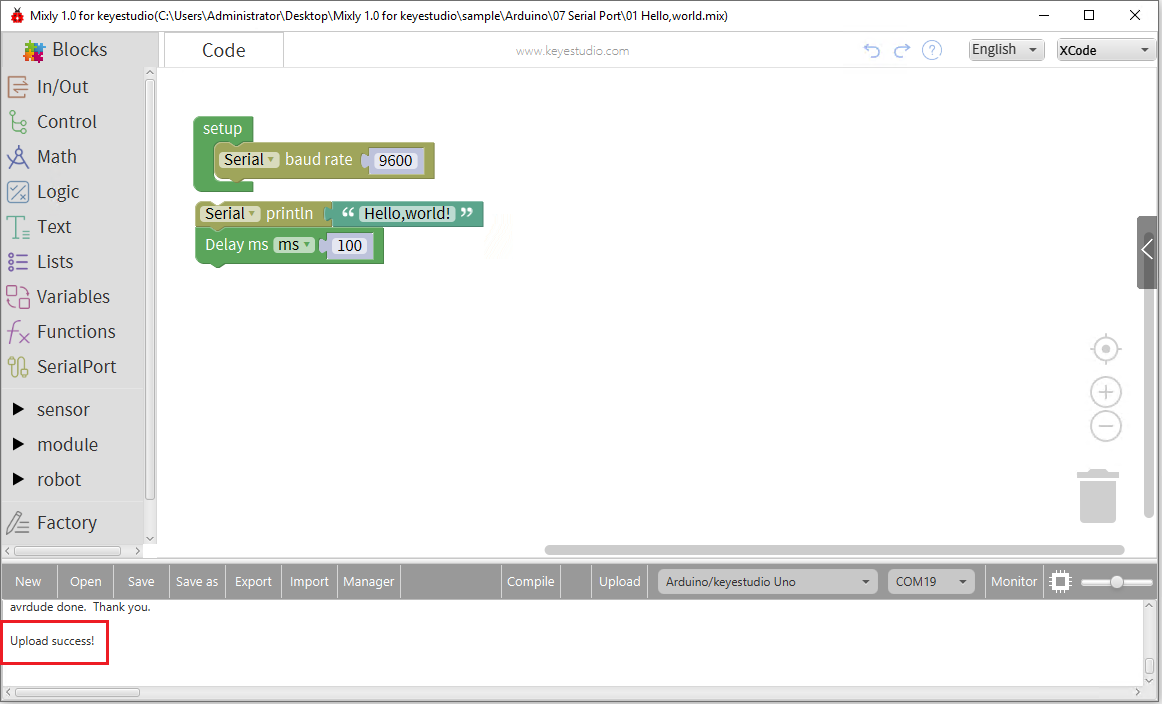
1. Connect control board to your PC via USB wire, choose the control board type and COM port(Right-click “Computer”-> “Properties”-> “Device Manager”, then you could see COM port at the “Other devices”), upload the program code to your control board. (illustrate by COM19)





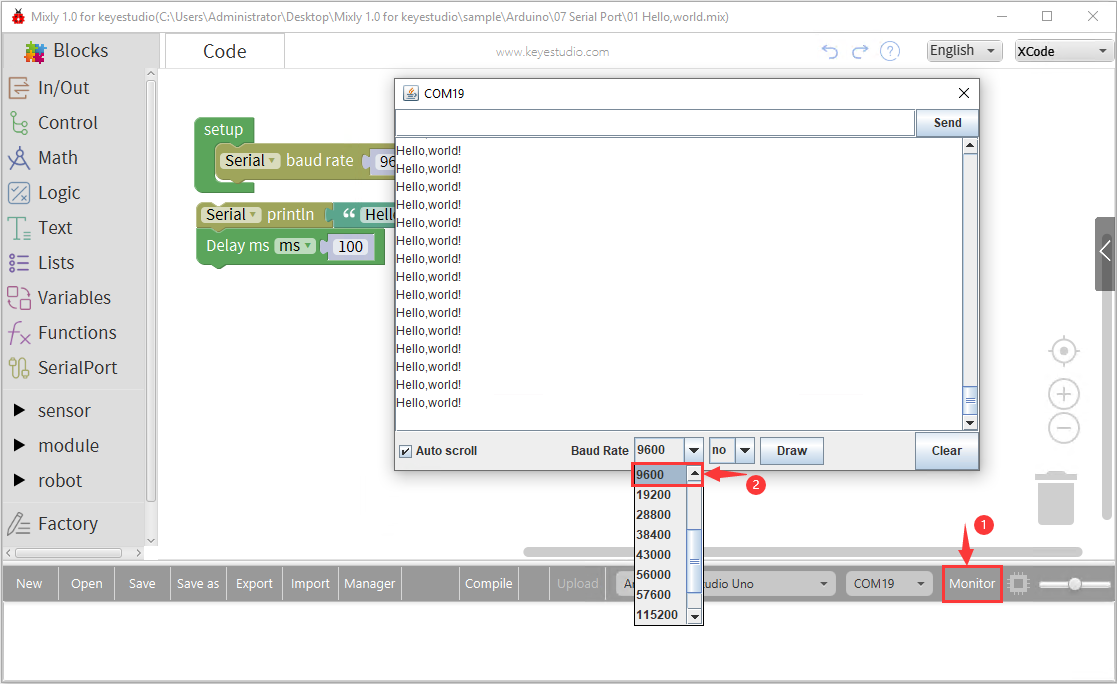


Upload program successfully.



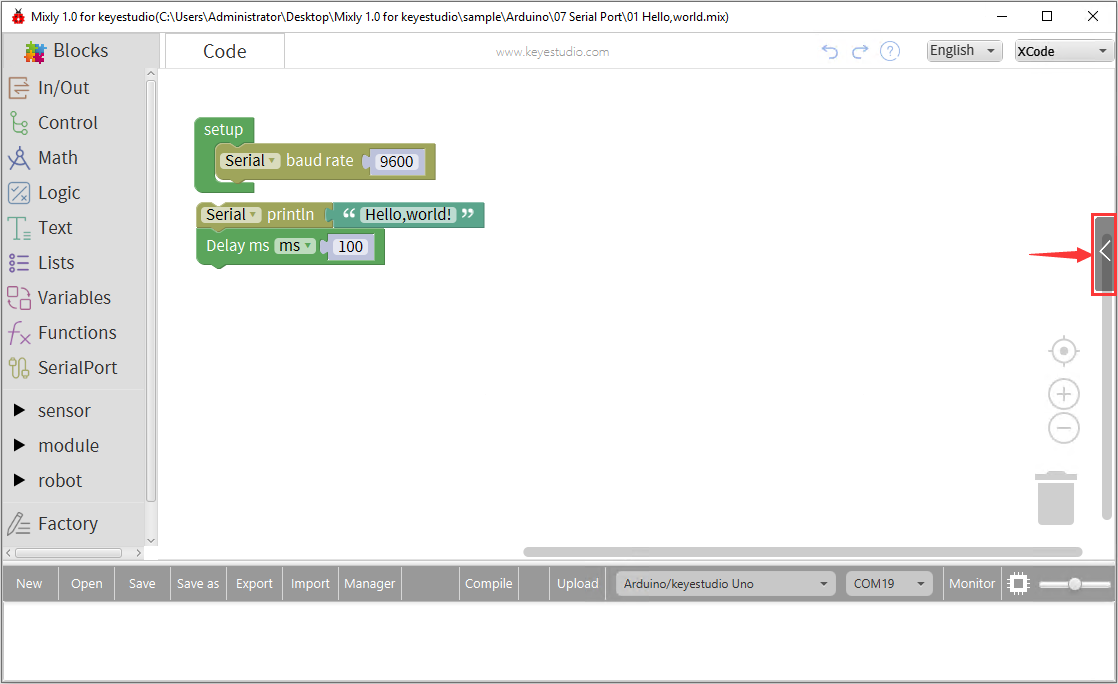
1. **Display Hello World!**

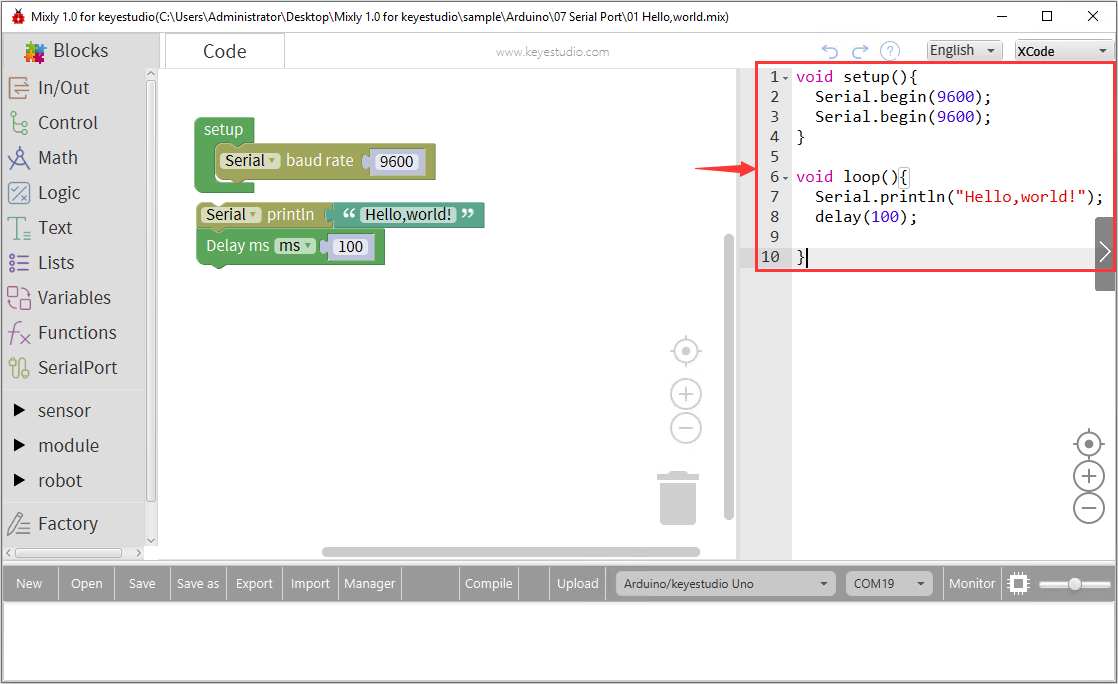
Then open serial monitor and set baud rate to 9600, “Hello World!” is shown on monitor.



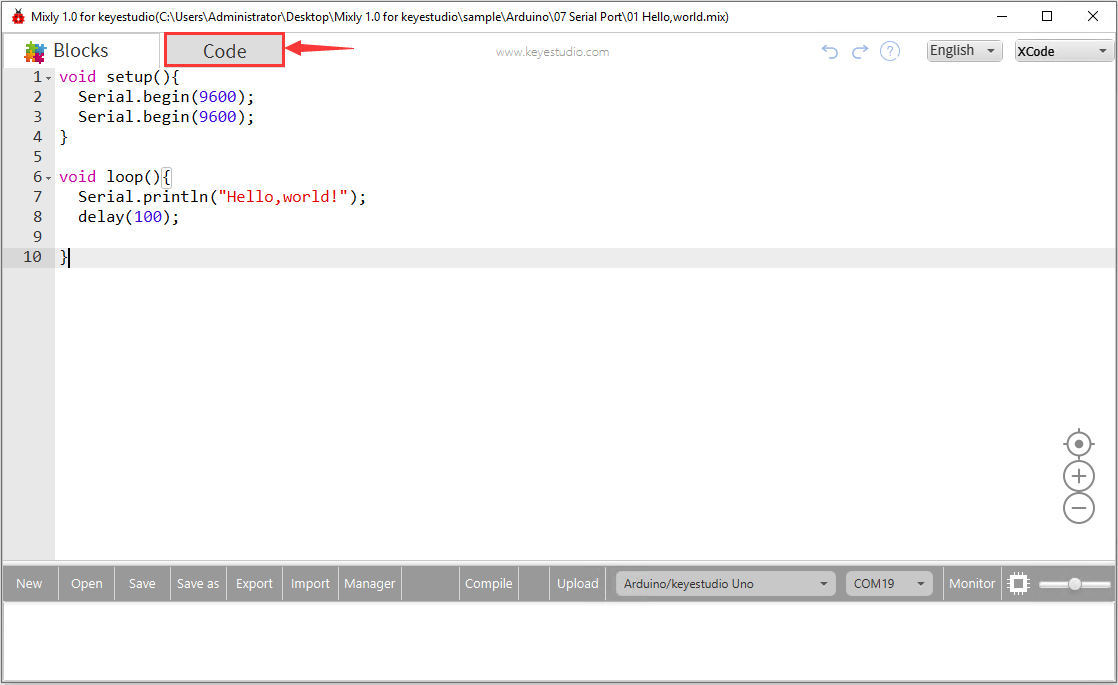
**Check Arduino Code：**

Click button to check Arduino code on the right side.



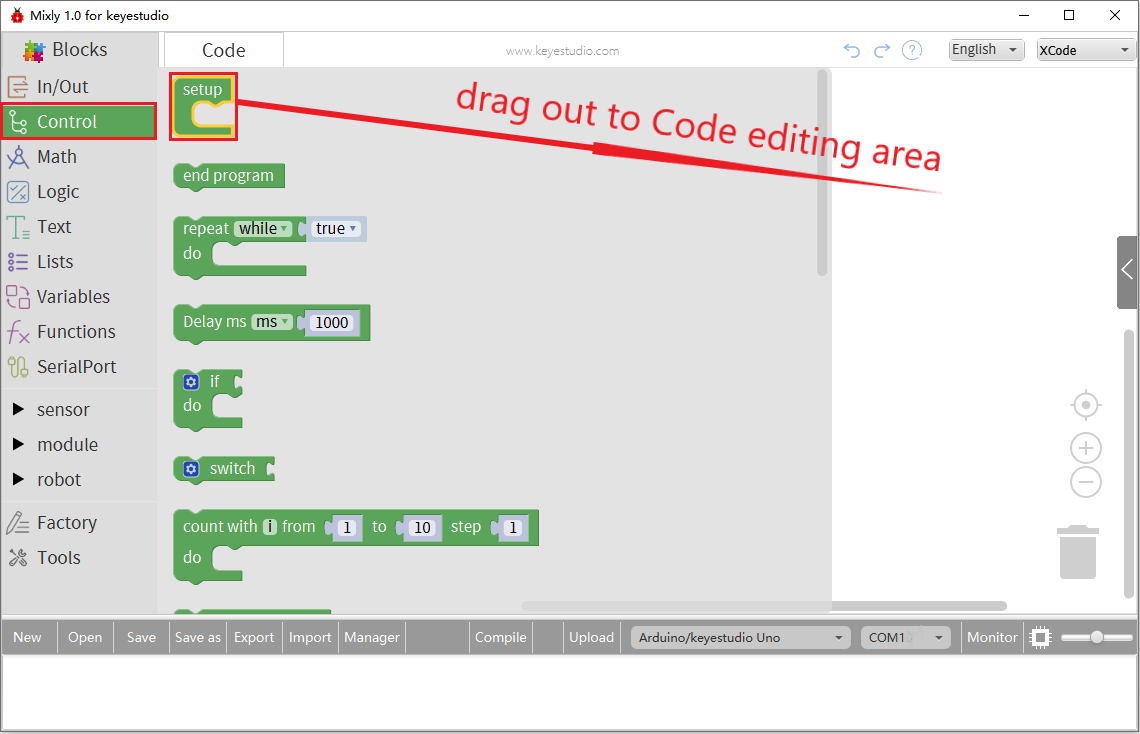


Or click “Code” to look through Arduino code.

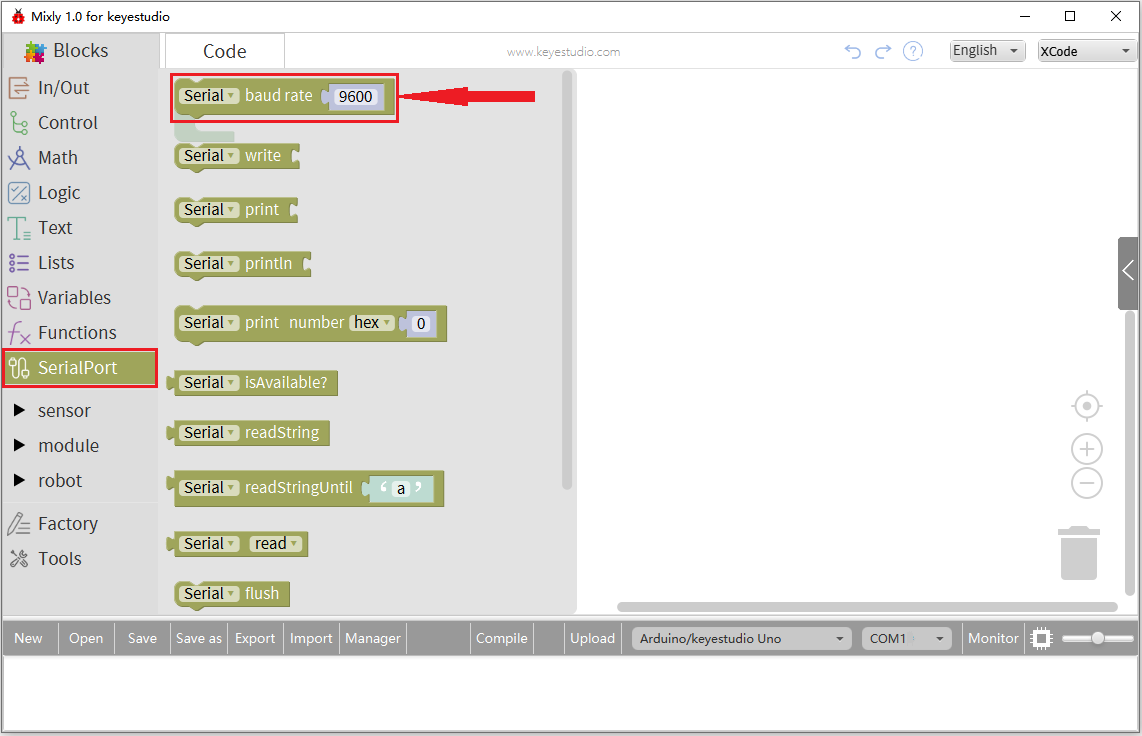


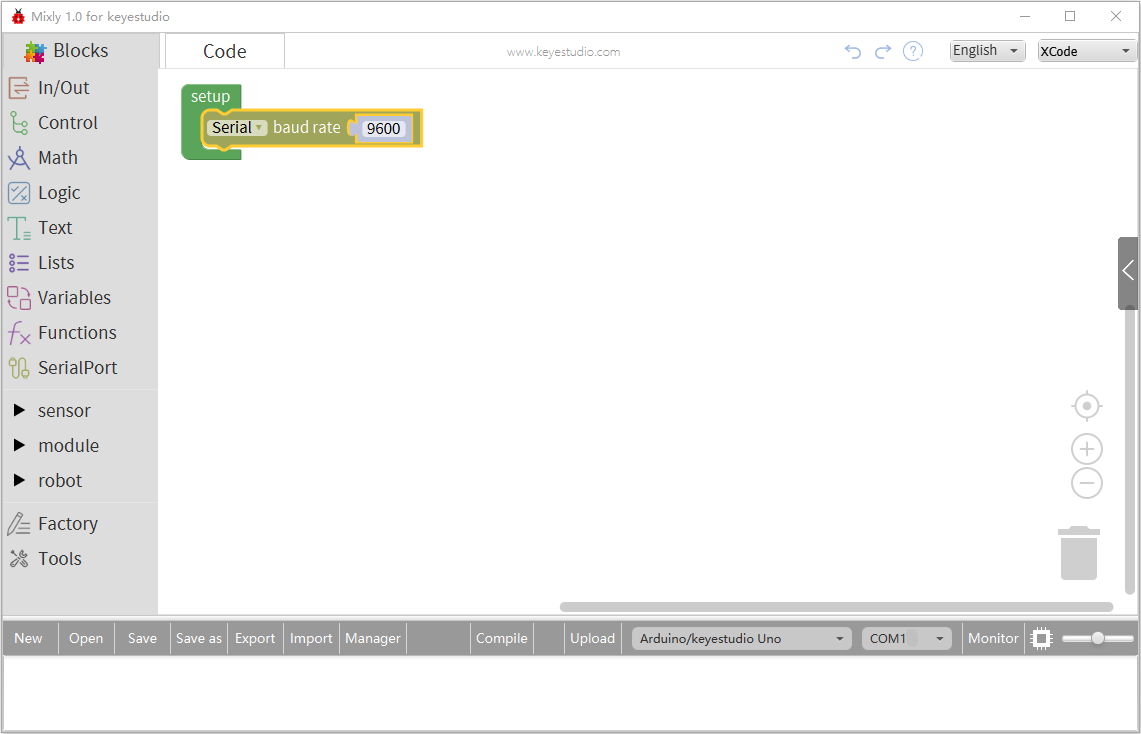
**You could design code by yourself.**

1. Click“Control” to find “set up” block and drag it to the edit area.

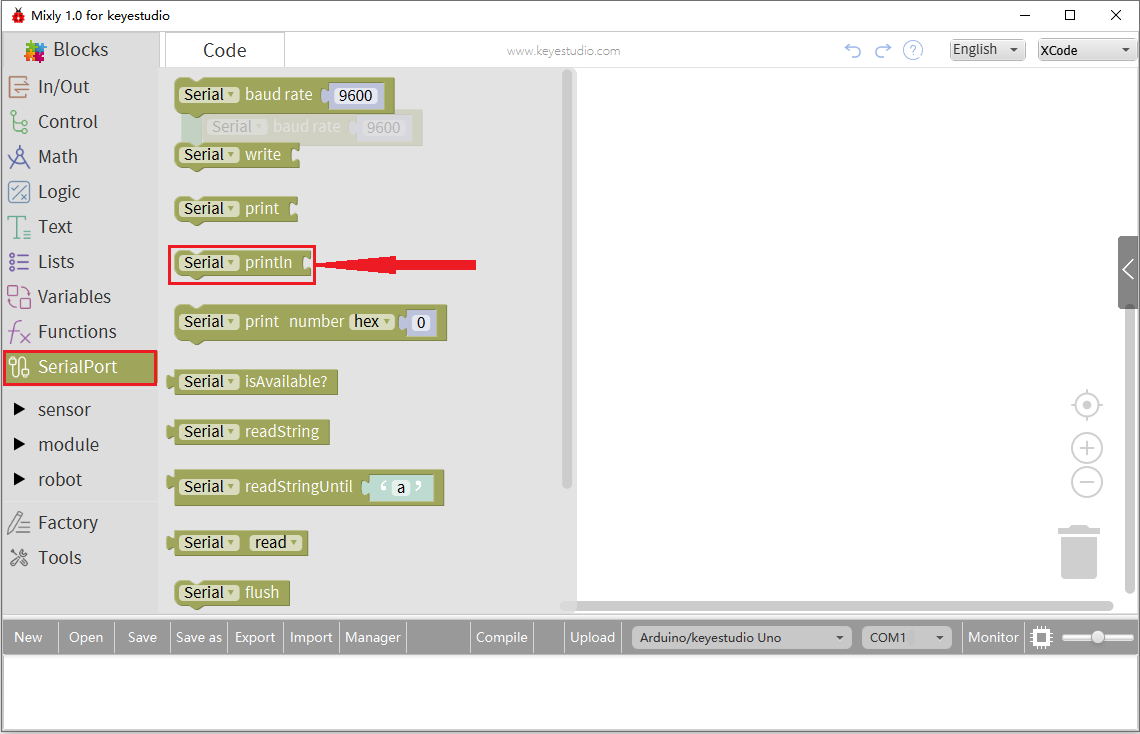


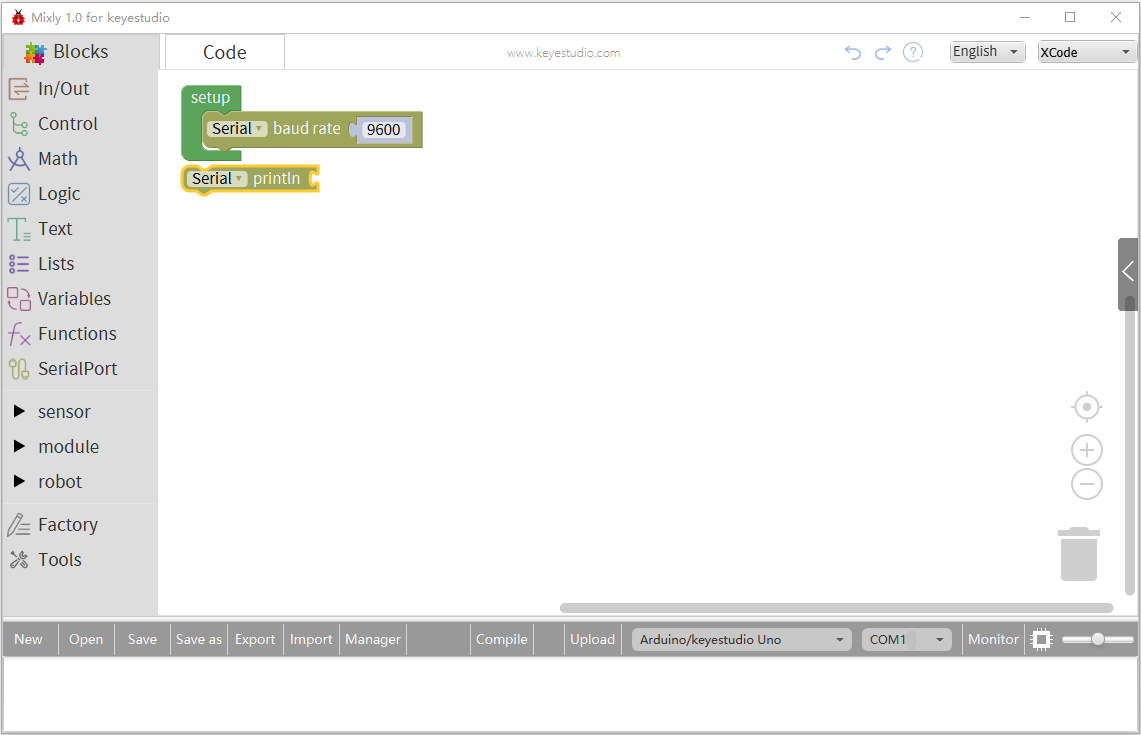
1. Click “Serial port”, move “Serial baud rate 9600”block into “set up” block.



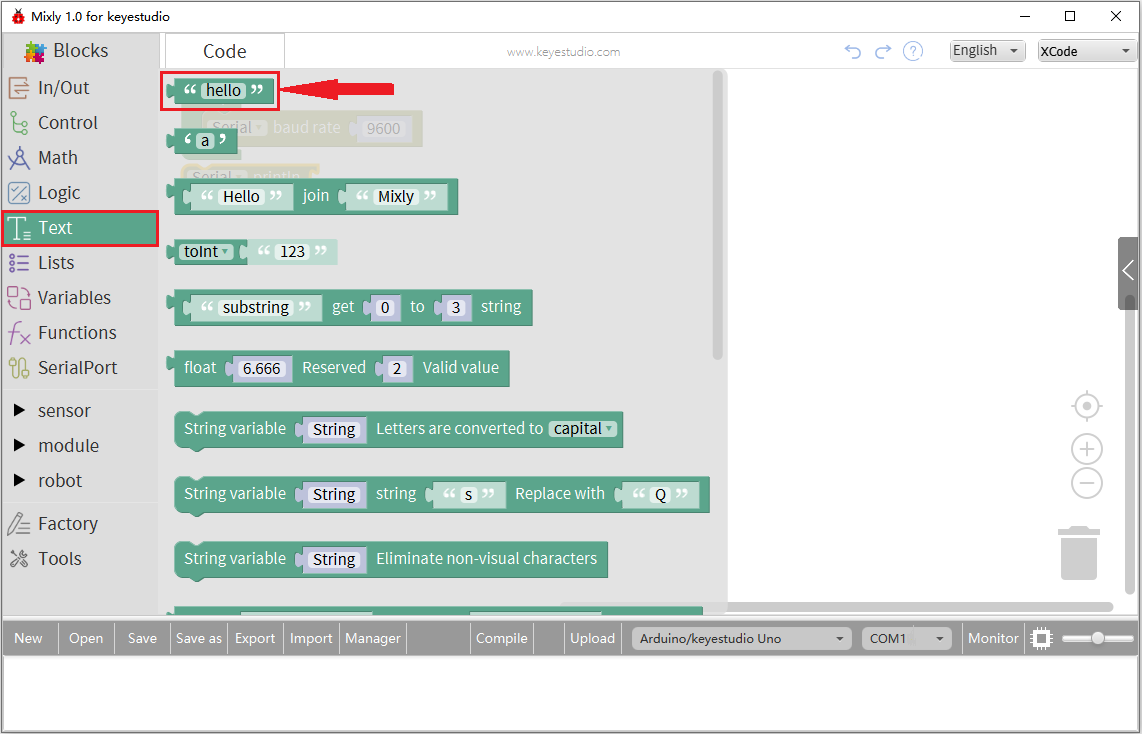


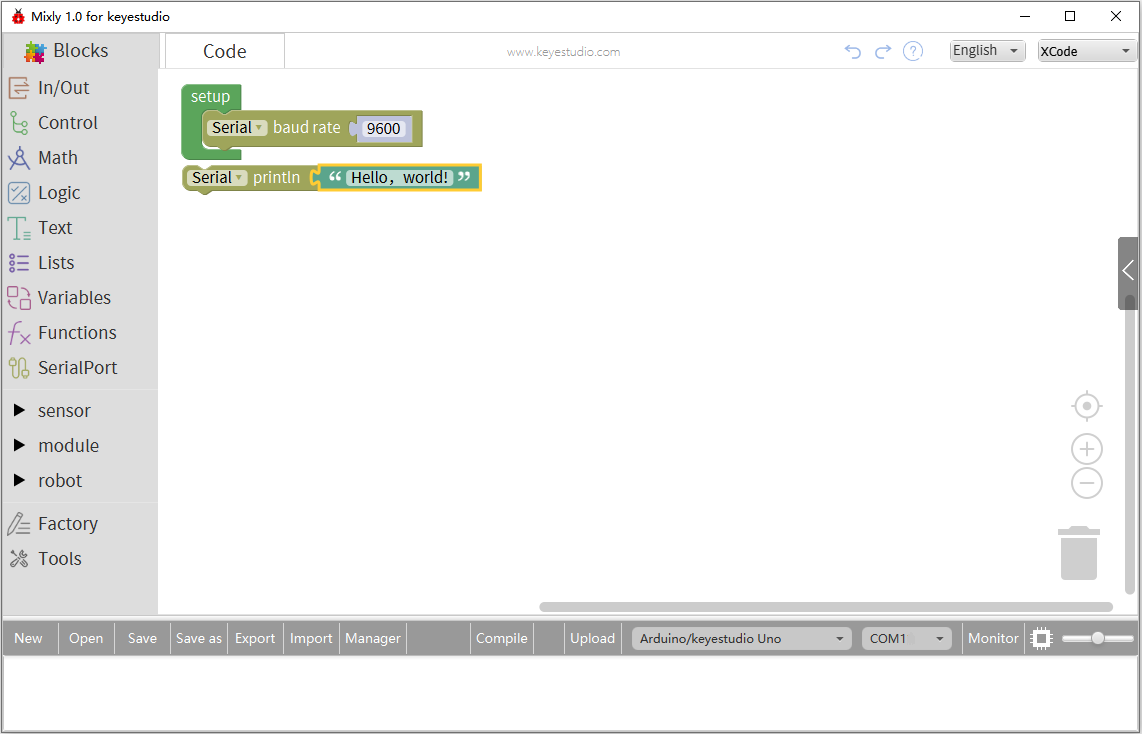
1. Select “Serial port” to drag “ Serial println” out.



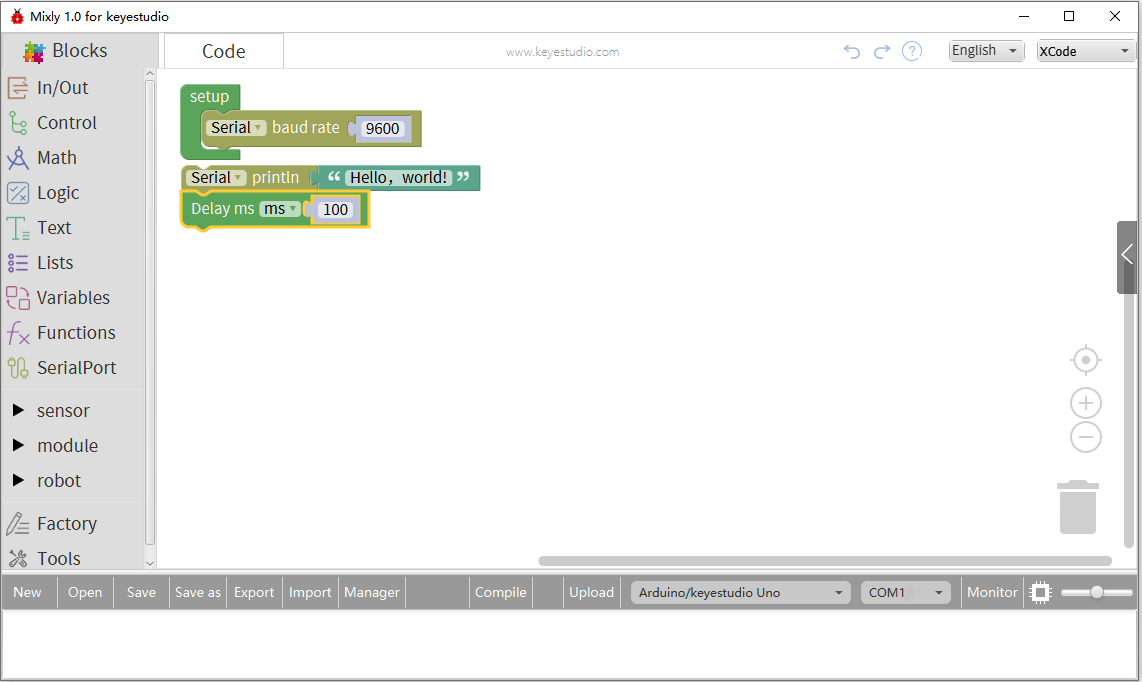


1. Add a text to be printed, click “Text” to find “hello” block and combine it with “serial println” block, then change “hello” into“Hello,world!”

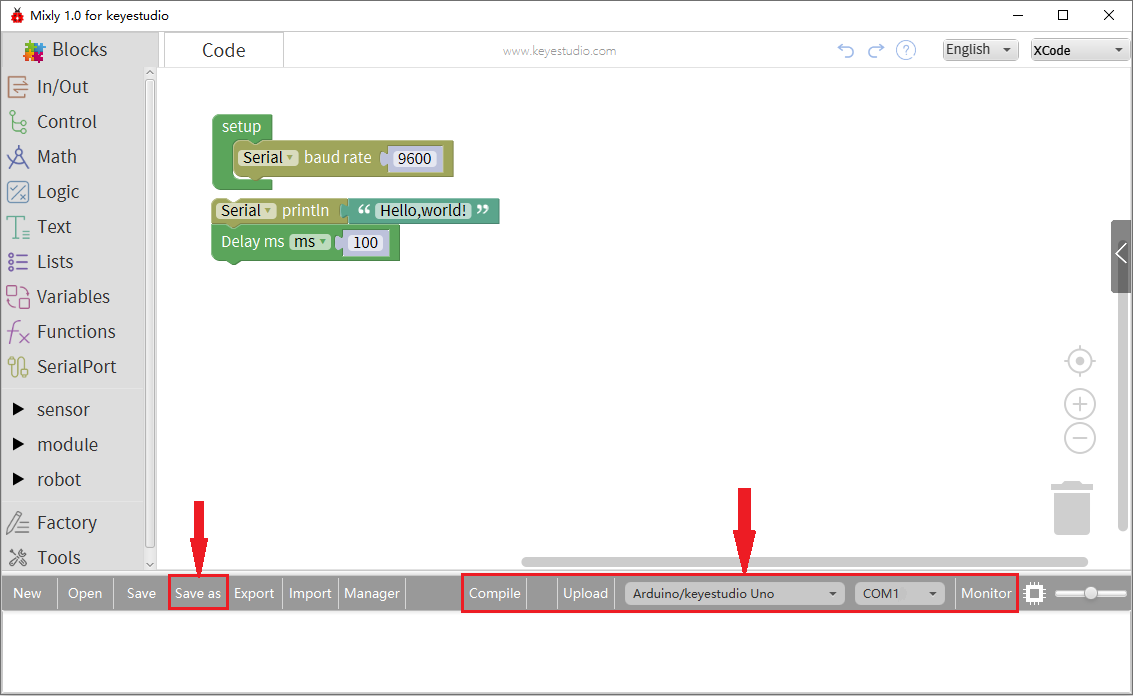




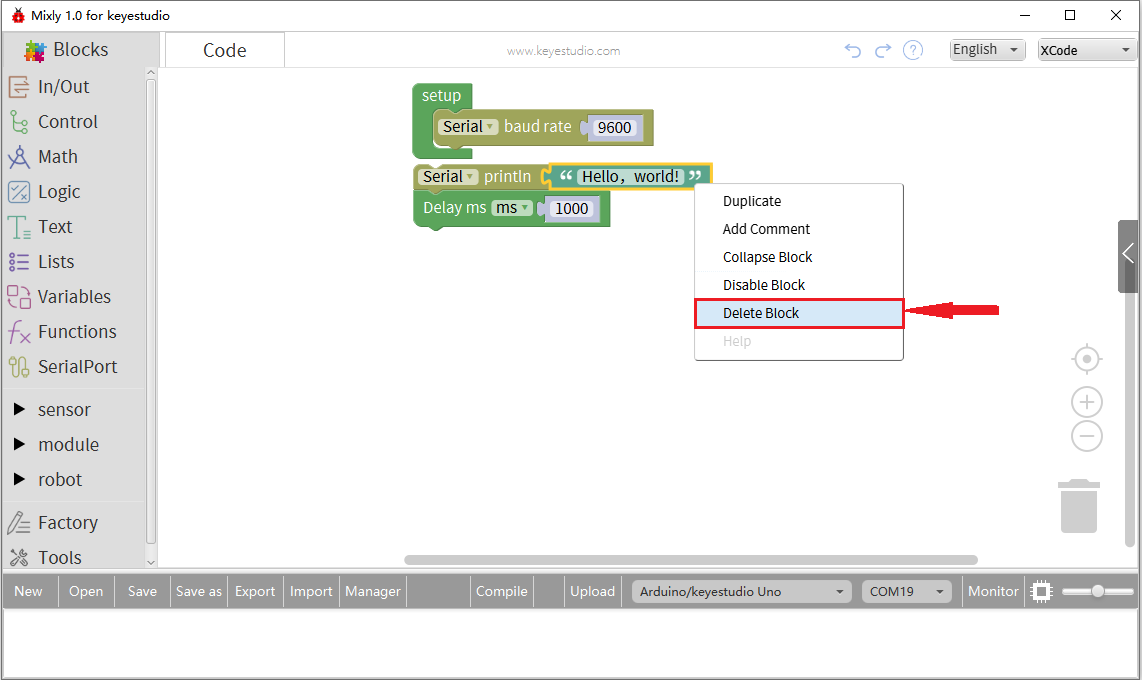
1. Enter “Control” to add delay block in editing area, delay time can be set freely, set to 100ms.

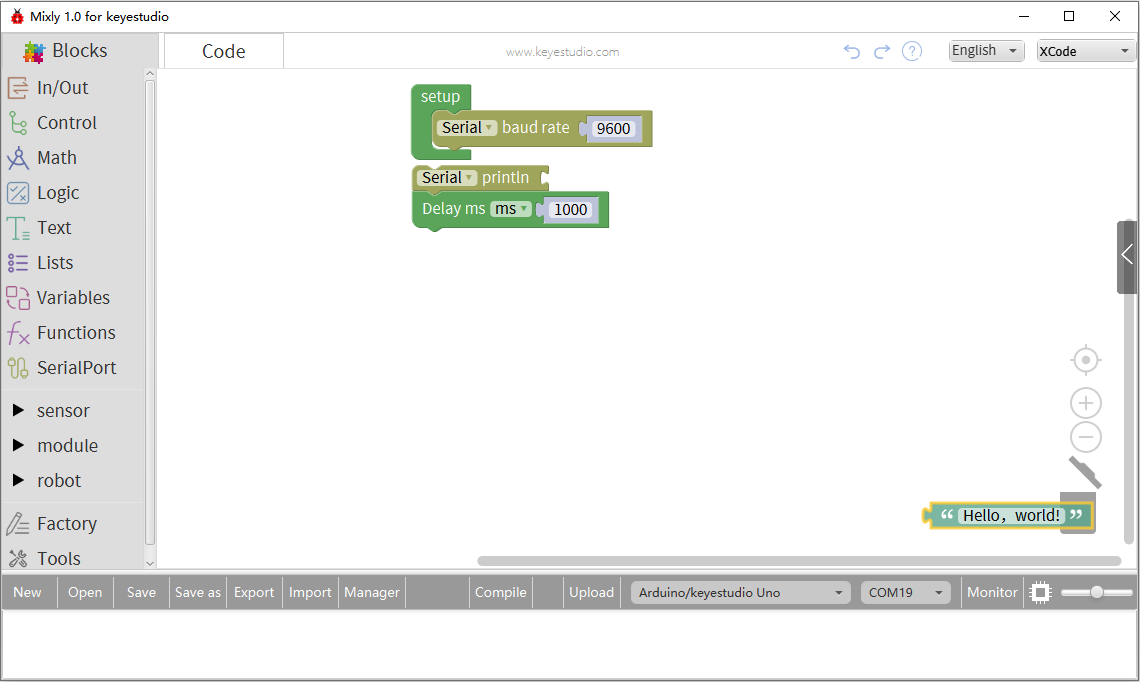


Upload code, click “Save as” to save and name this file. You could upload on the control board directly. Then see the result on serial monitor.



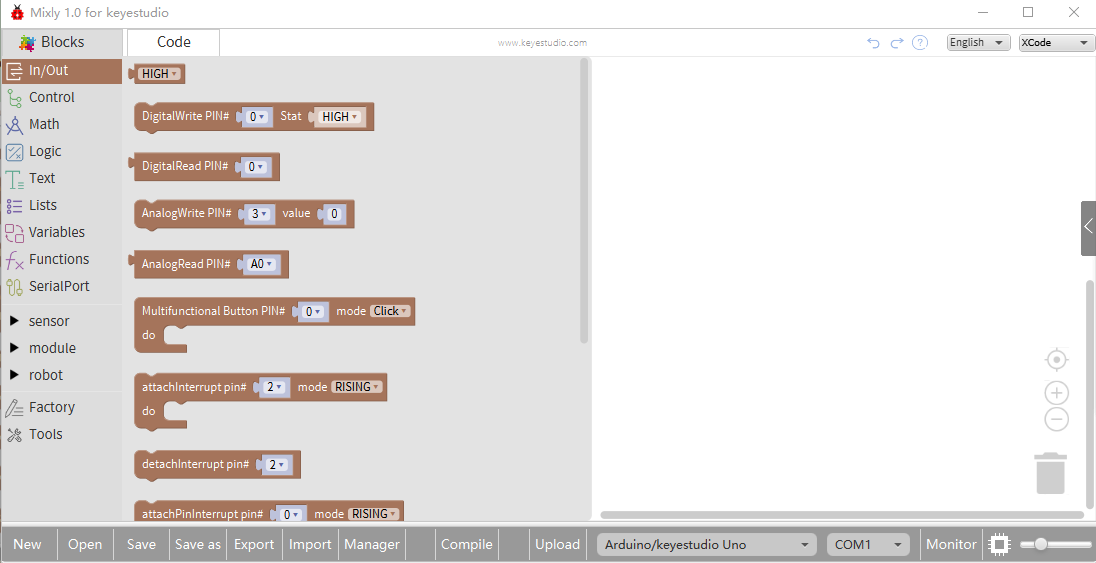
On the other hand, if you intend to delete some code in editing area, just right-click to select the block to be deleted, or directly put it into recycle bin.





**4. Some common functions:**

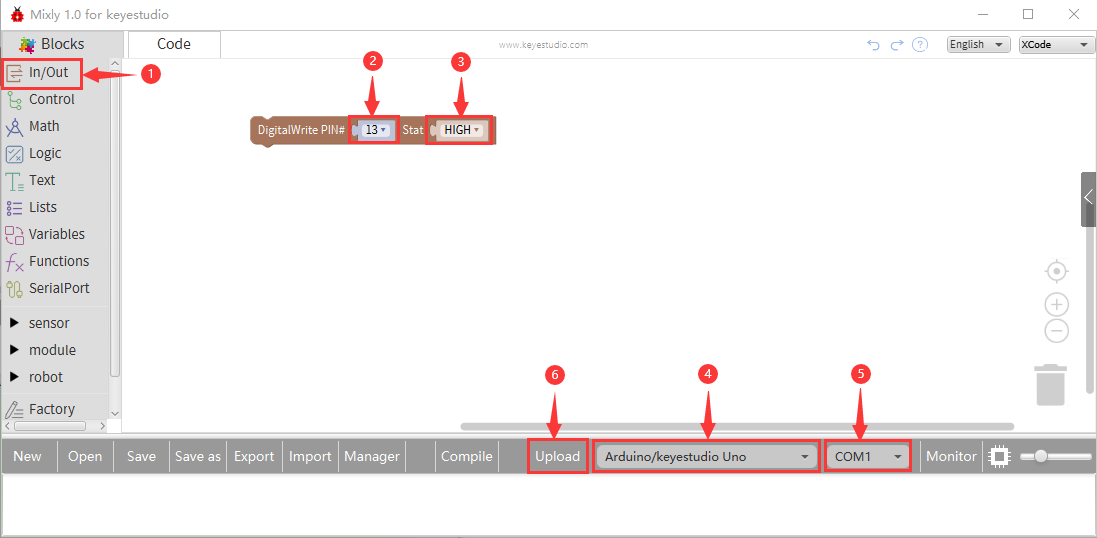
## 4.1 In/Out Block



|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | Returns HIGH or LOW voltage |
| **2** |  | Write digital value to a specific Port.  Digital Output: set the HIGH or LOW output for IO pins |
| **3** |  | Returns a digital value of a specific Port.  Digital IO Read Pin, generally used to read the HIGH or LOW level detected by Digital sensor |
| **4** |  | Write analog value between 2 and 255 to a specific Port.  Analog Output: set the Analog value output by Analog IO pins (0~255). |
| **5** |  | Returns value between 0 and 1023 of a specific Port.  Analog IO Read Pin, generally used to read the Analog value detected by Analog sensor. |
| **6** |  | Set buttons connected to specific pins to multifunctional buttons, and confirm the different reactions on different mode.  Mode：Click/Double-click/Long Press Start/During Long Press/Long Press End |
| **7** |  | Hardware external interrupt function only supports pin 2 and 3, when external interrupt happens, we call a function to replace the program which is being executed.  Mode: Rising, Falling and Change |
| **8** |  | Detach interrupt to a specific Port.  Turn off the given interrupt function. |
| **9** |  | Software interrupt function supports all pins, when the external interrupt happens, we call a function to replace the program which is being executed.  Mode: Rising, Falling and Change |
| **10** |  | Turn off the specific interrupt function |
| **11** |  | Set the IO pins as Output or Input state |
| **12** |  | Read the continuous time of HIGH or LOW pulse from IO pins  ( generally used for ultrasonic ranging) |
| **13** |  | Read a pulse (either HIGH or LOW) on a pin within a time set in timeout. |
| **14** |  | Set the ShiftOut data pin, clock pin. Output the data needed from the bitOrder MSBFIRST or LSBFIRST (Most Significant Bit First, or, Least Significant Bit First).  Generally used for controlling the 74HC595 CHIP. |

**For example:**

Connect your Arduino Uno board, then follow the steps below to light the Pin13 led on Arduino UNO.



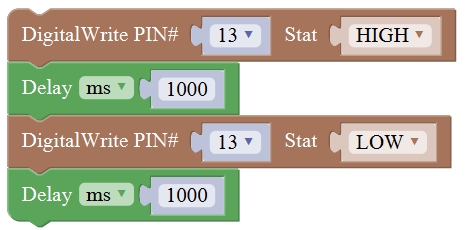
## 4.2 Control Block



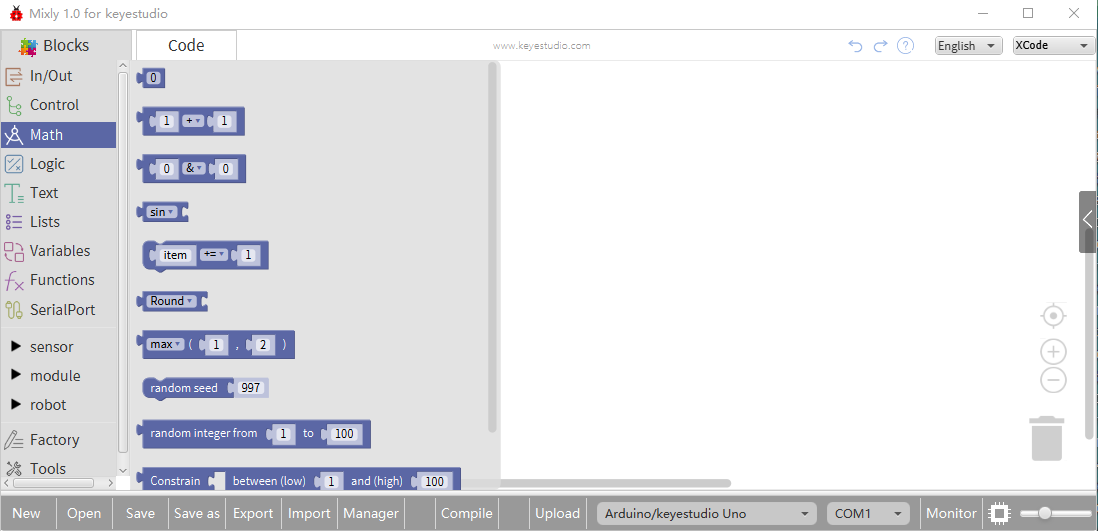
|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | Initialization (run only once) |
| **2** |  | End the program, means the program will stop running when use this block. |
| **3** |  | Delay function, click to select **ms** or **us**  (pause the program for the amount of time (in milliseconds) specified as parameter. There are 1000 milliseconds in a second.) |
| **4** |  | **if\_do** function (first evaluate a value be [true or false](https://www.arduino.cc/reference/en/language/variables/constants/constants/), if a value is true, then do some statement. You can click the blue gear icon to select the **else if** block or **else** block.) |
| **5** |  | **switch** function. You can click the blue gear icon to select the **case** block or **default** block. (used to evaluate several programs then execute the corresponding function matched with program.) |
| **6** |  | Equal to **[for](https://www.arduino.cc/reference/en/language/structure/control-structure/for/)** [statement](https://www.arduino.cc/reference/en/language/structure/control-structure/for/). |
| **7** |  | A **while** loop statement. |
| **8** |  | **break** function, used to exit from the containing loop. |
| **9** |  | **millis()** function, returns the system running time since the program started.  (The unit can be **ms** (milliseconds) or **μs**（microsecond)). |
| **10** |  | Timer interrupt function, that is, set a trigger interrupt for the amount of time (in milliseconds) specified as parameter. |
| **11** |  | Timer interrupt start block |
| **12** |  | Timer interrupt stop block |
|  |  | Set different simple timer, execute the corresponding program every interval time |
|  |  | Register a delayed function, and put the code to be executed in this function |
|  |  | Execute the delayed function, interval time is 1000ms，1 time |
|  |  | Allow to input and output the running of pins interrupt |
|  |  | Ban to input and output the running of pins interrupt |
|  |  | SCoop block is used to execute multithreading tasks，up to 8 tasks |
|  |  | Execute multithreading tasks |
|  |  | Only used for SCoop delay block |

**For example:**

Compile and upload the program below to your Arduino board, you should see Pin13 LED on Arduino UNO continue to flash.(with an interval of 1s, equal to 1000ms)

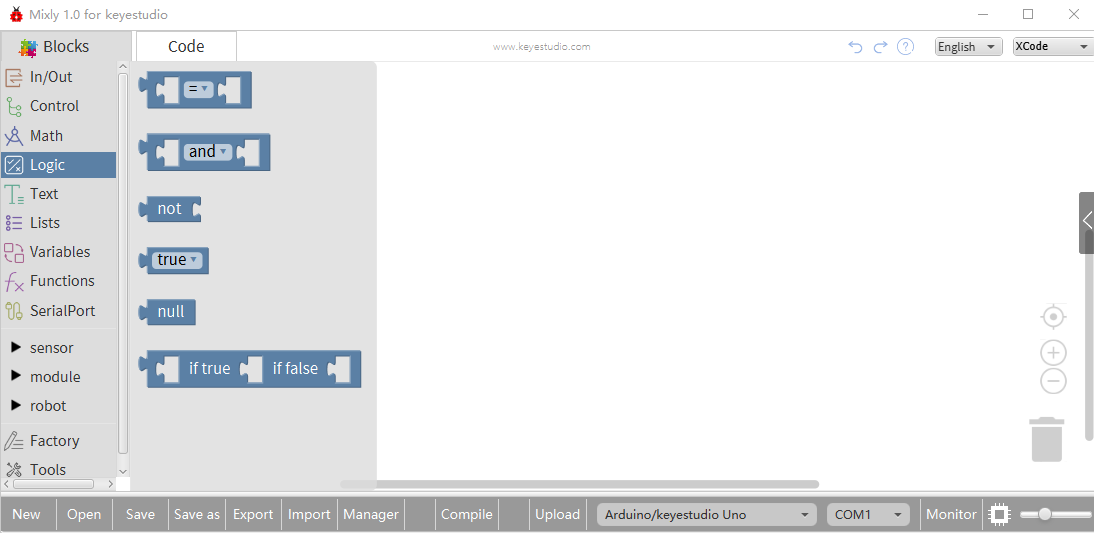


## Math Block



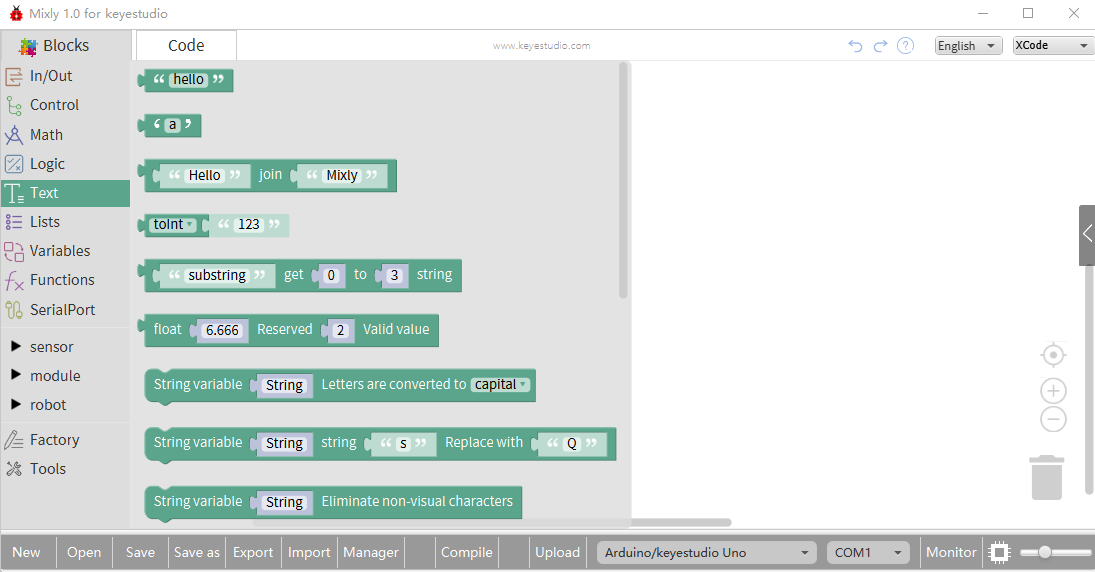
|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | A number |
| **2** |  | Click to select the Arithmetic Operators:  **[＋(addition)](https://www.arduino.cc/reference/en/language/structure/arithmetic-operators/addition/); [－(subtraction)](https://www.arduino.cc/reference/en/language/structure/arithmetic-operators/subtraction/);**  **[x (Multiplication)](https://www.arduino.cc/reference/en/language/structure/arithmetic-operators/multiplication/); [÷ (division)](https://www.arduino.cc/reference/en/language/structure/arithmetic-operators/division/);**  **[% (remainder)](https://www.arduino.cc/reference/en/language/structure/arithmetic-operators/remainder/); [^ (bitwise xor)](https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitwisexor/)** |
| **3** |  | Click to select the **[& (bitwise end)](https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitwiseand/); [l (bitwise or)](https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitwiseor/); [<< (bitshift left)](https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitshiftleft/);  [>> (bitshift right)](https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitshiftright/)** |
| **4** |  | Click to select the **[sin](https://www.arduino.cc/reference/en/language/functions/trigonometry/sin/); [cos](https://www.arduino.cc/reference/en/language/functions/trigonometry/cos/); [tan](https://www.arduino.cc/reference/en/language/functions/trigonometry/tan/); asin; acos; atan; ln; log10; e^; 10^; [++ (increment)](https://www.arduino.cc/reference/en/language/structure/compound-operators/increment/) ;**  **[-- (decrement)](https://www.arduino.cc/reference/en/language/structure/compound-operators/decrement/)** |
| **5** |  |  |
| **5** |  | Click to select the **Round; Ceil;** **Floor; [abs](https://www.arduino.cc/reference/en/language/functions/math/abs/); [sq](https://www.arduino.cc/reference/en/language/functions/math/sq/); [sqrt](https://www.arduino.cc/reference/en/language/functions/math/sqrt/)**  **Round:** Returns the integer part a number using around.  **Ceil:** Returns the integer part a number using ceil.  **Floor:** Returns the integer part a number using floor.  **abs:** Return the absolute value of a number.  **sq:** Return the square of a number.  **sqrt:** Return the square root of a number. |
| **6** |  | If select the **max**, returns the larger number;  if select the **min**, returns the smaller number. |
| **7** |  | Initialize the random seed |
| **8** |  | Return a random integer between the two specified limits, inclusive. |
| **9** |  | Constrain a number to be between the specified limits (inclusive).  (generally used to constrain an analog value read from sensor) |
| **10** |  | Map a number from the first interval to the second interval.  (For instance, potentiometer-controlled servo, map the range of potentiometer (0, 1023) to the angle of servo (0, 180)). |

## Logic Block



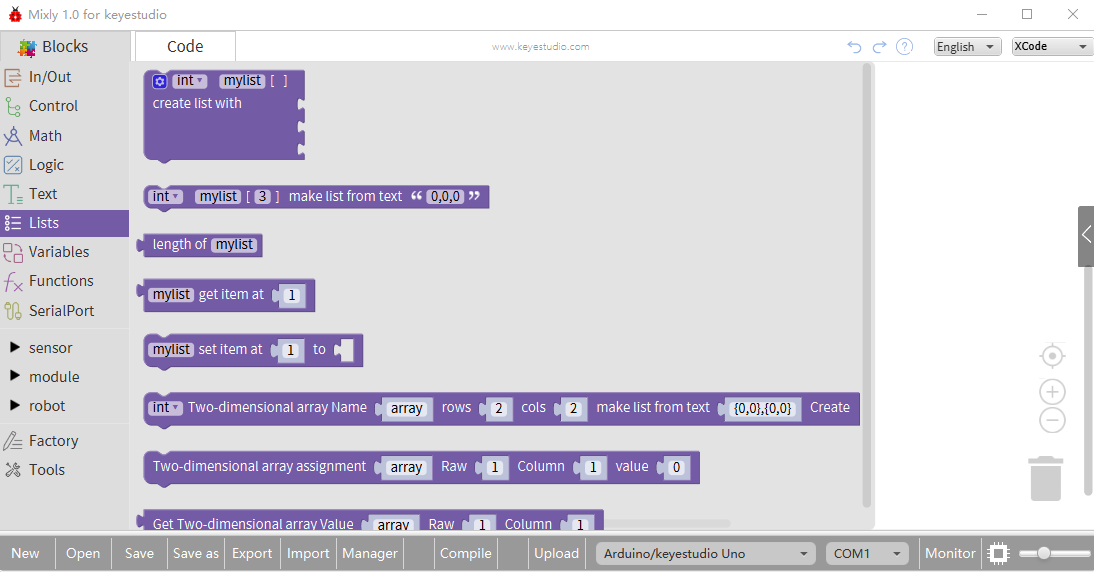
|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | **logic comparision**  **=**: Return true if both inputs equal each other.  **≠** : Return true if both inputs are not equal to each other.  **<**: Return true if the first input is smaller than the second input.  **≤** : Return true if the first input is smaller than or equal to the second input.  **>**: Return true if the first input is greater than the second input.  **≥ :** Return true if the first input is greater than or equal to the second input. |
| **2** |  | **and:** Return true if both inputs are true;  **or:** Return true if at least one of the inputs is true |
| **3** |  | Returns true if the input is false. Returns false if the input is true. |
| **4** |  | Returns either true or false. |
| **5** |  | Returns null |
| **6** |  | If the first number is true, the second number is returned, otherwise the third number. |

## Text Block



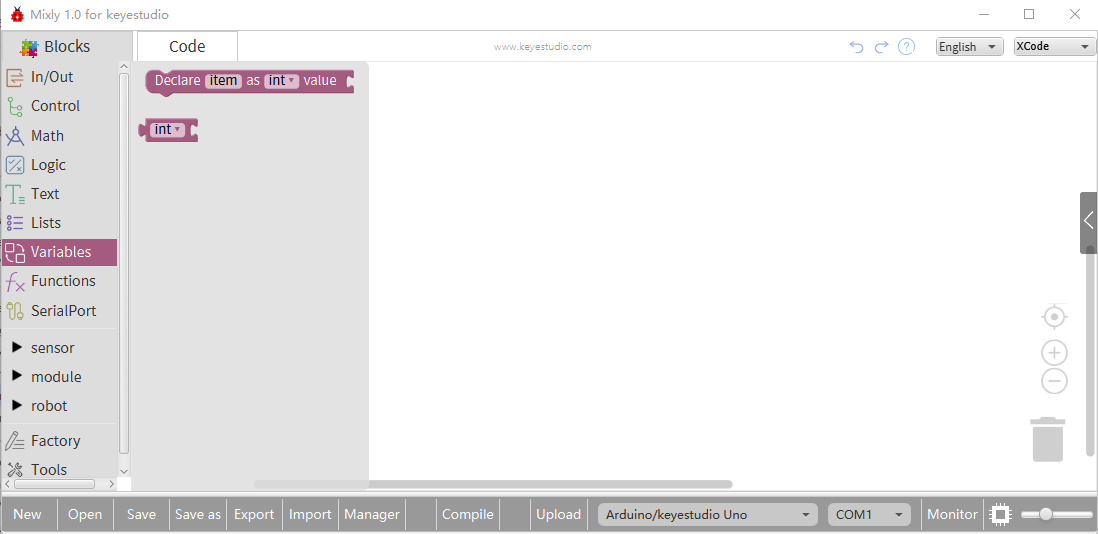
|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | character string: a letter, word, or line of text. |
| **2** |  | A character |
| **3** |  | Creates a piece of text by joining together two piece of text.  ( Here Hello join Mixly equals HelloMixly) |
| **4** |  | Converts a string into an integer or an float. |
|  |  | obtain the character string from x to y（eg：acquire the character string from 0 to 3) |
|  |  | Simplify the decimals in compliance with designated bits（eg：make decimal 6.66 simplify according to designated 2 bits |
|  |  | Transfer the all lower-case letters of designated character strings into capital letters or capital ones into lower-case letters |
|  |  | Replace the designated characters or character strings(the first character string is original, the second is to be replaced , the third one is character string which replaces. |
|  |  | Eliminate the non-visual characters in the  character strings |
|  |  | Whether the first character string starts or end with the second character string, if it is, return 1; otherwise, return 0.(the first character string needs to be determined, the second one is character string which determines |
|  |  | Transfer data type into character strings/ characters/bites/integers/long integers/decimals/words |
| **5** |  | Returns the char corresponding to an ASCII code  (Decimal number 97 corresponding to a) |
| **6** |  | Returns the ASCII code corresponding to a char. |
| **7** |  | Converts a number into a string. |
| **8** |  | Calculates the length of a string |
| **9** |  | Output the char of a string (the char at 0 of hello is h) |
| **10** |  | The first string equals or startsWith or endsWith the second string, returns 1, otherwise returns 0.  (if equals, both strings are abc, returns 1.) |
| **11** |  | Returns a decimal value of the first string subtracts the second string. |

## List Block



|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | Create a list with any number of items |
| **2** |  | Creats a list from a text. (int mylist [3]={0,0,0};) |
| **3** |  | Returns the length of a list |
| **4** |  | Returns the value of at the specified position in a list. |
| **5** |  | Sets the value of at the specified position in a list.  Set the first item in mylist to another item. |
|  |  | Initialize two-dimensional array, select the data type including integers, long integers, decimals, character strings, bites, char and characters |
|  |  | Define value on M row and N column |
|  |  | Obtain the content from array on M row and  N column |

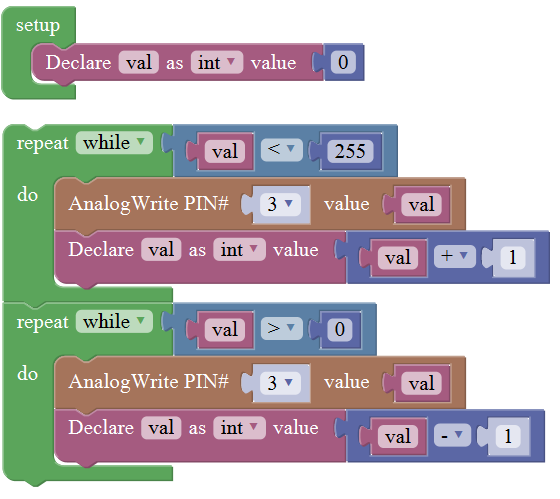
## Variable Block



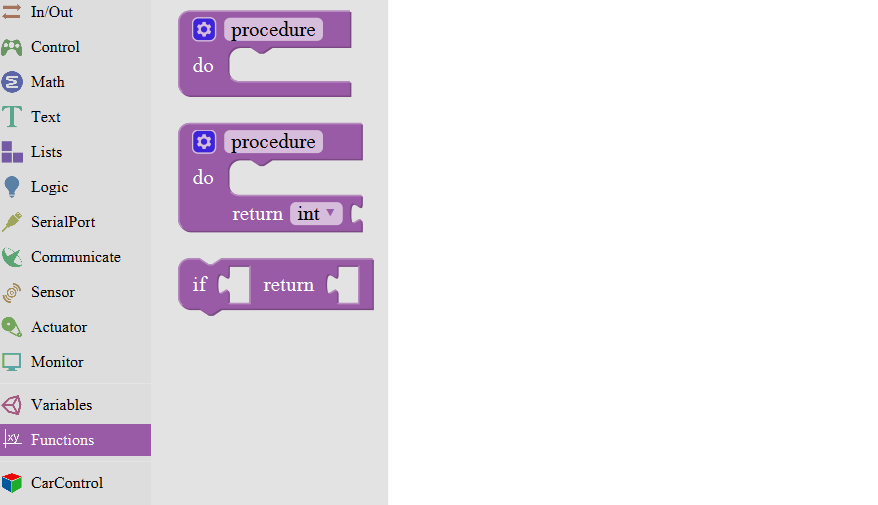
|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | Declare and initialize a variable.  Click to select**[int](https://www.arduino.cc/reference/en/language/variables/data-types/int/), [long](https://www.arduino.cc/reference/en/language/variables/data-types/long/), [float](https://www.arduino.cc/reference/en/language/variables/data-types/float/), [boolean](https://www.arduino.cc/reference/en/language/variables/data-types/boolean/), [byte](https://www.arduino.cc/reference/en/language/variables/data-types/byte/), [char](https://www.arduino.cc/reference/en/language/variables/data-types/char/), [string](https://www.arduino.cc/reference/en/language/variables/data-types/string/)** |
| **2** |  | Define the data types |

**For example: LED breath**

You need an Arduino Uno and one LED module. Connect the control pin of LED module to Pin 3 of Uno board (or other pins with “~”，that is, those pins can output PWM signal). LED will gradually light then gradually dim, repeatedly.

****

## Functions Block

****

|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | Creates a function with no output.  Click the blue icon to set the procedure parameter.  (no return value) |
| **2** |  | Creates a function with an output.  Click the blue icon to set the procedure parameter.  (with return value and can set the data types) |
| **3** |  | If a value is true, then return a second value. |
| **4** |  | Return |

**For example:**

Below is an example code for line tracking car. We use three tracking modules (left to D6, middle to D7, right to D8). of course you need a tracking car to test it. First edit the forward, backward, turn left, turn right and stop into functions block. Then compile and upload the code below.

****

****

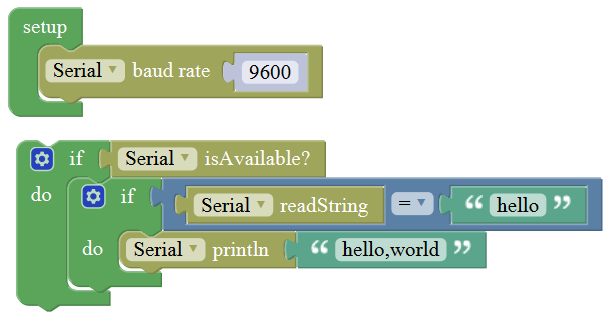
## SerialPort Block

****

|  |  |  |
| --- | --- | --- |
| **NO.** | **BLOCK ICON** | **DEFINITION** |
| **1** |  | Set the serial buad rate to 9600 |
| **2** |  | Write the specified number, text or other value. |
| **3** |  | Print the specified number, text or other value on monitor. |
| **4** |  | Print the specified number, text or other value on newline of monitor. |
| **5** |  | Print the specified number in hexademical format on newline of monitor. |
| **6** |  | If the serial port is available, it returns true, otherwise returns false.  (generally used in Bluetooth communication) |
| **7** |  | Returns a string in serial port |
| **8** |  | A string read from serial port to a string variable, pause until read the specified character. |
| **9** |  | Read the serial data by byte (generally used to read the value sent from Bluetooth) (delete the data has been read) |
| **10** |  | Wait for the output data completed |
| **11** |  | Set the software serial port  (call this function if need to use several serial ports) |
| **12** |  | Event function trigger by serial port data, that is, serial port is ready to call this function.  (equal to an interrupt function) |

**For example: serial communication**

Done uploading the code, open the Arduino monitor, then enter a “hello” on the top bar, and click Send, it will print out “hello,world”.

****

