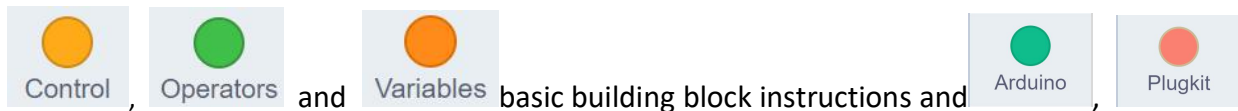
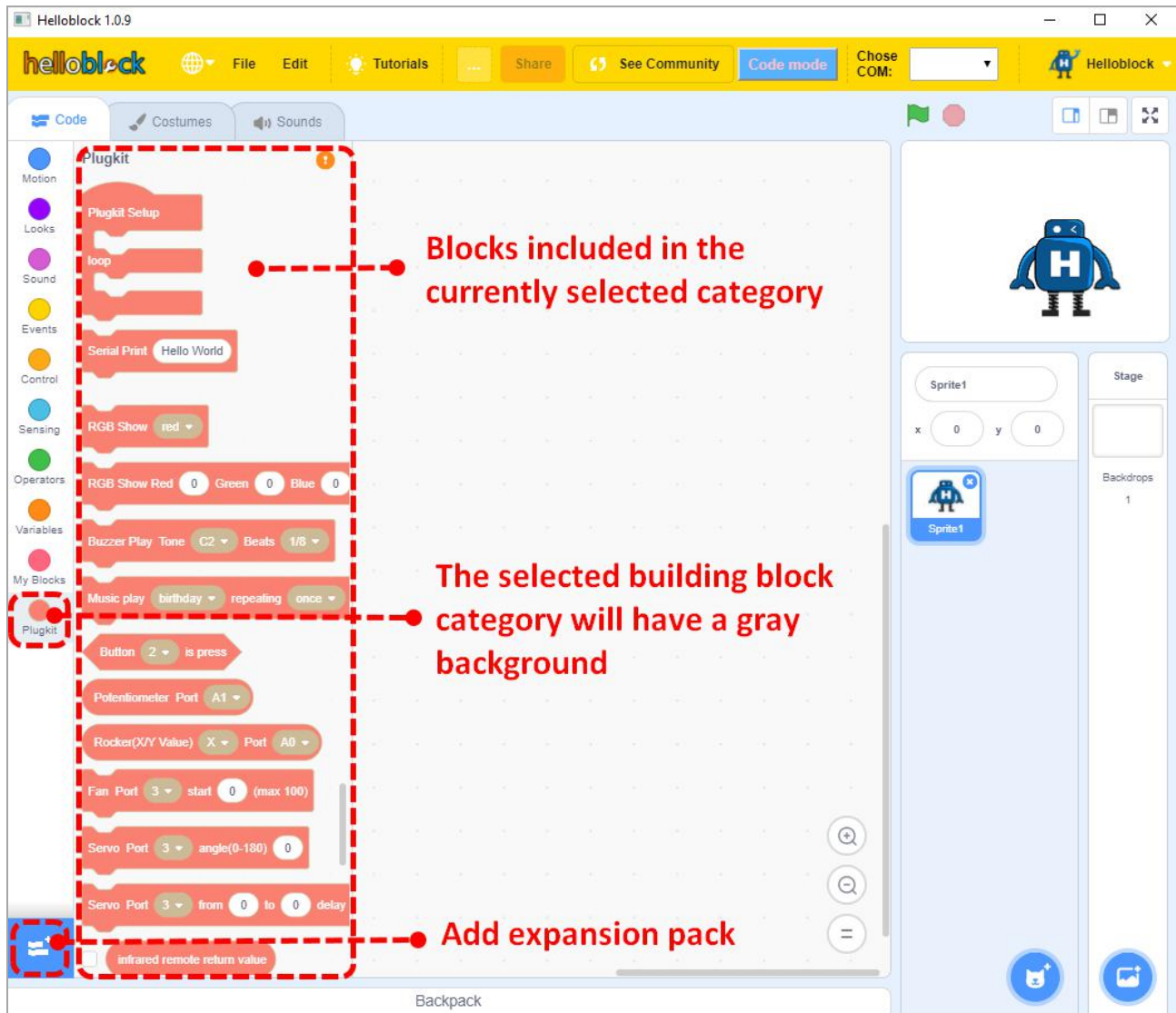


The basic building block instructions in Helloblock are divided into nine categories, and there are also eight types of Yahboom building blocks expansion packs. This course mainly uses the

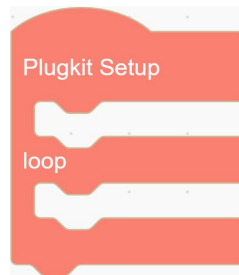


expansion packs. Different modules are marked with different colors, which easy to find quickly. In this course, they are expressed by brackets and category names. For example, the operation class is 【Operation】 , the control class is 【Control】 , the variable class is 【Variable】 , the Arduino class is 【Arduino】 , and the Plugkit class is 【Plugkit】 .



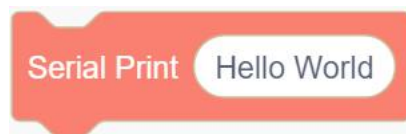
7.1 Plugkit category

Main function



Setup only executes once, and mainly places initialization instructions. Loop execution, mainly places instructions and statements that we need to execute repeatedly.

Serial Print



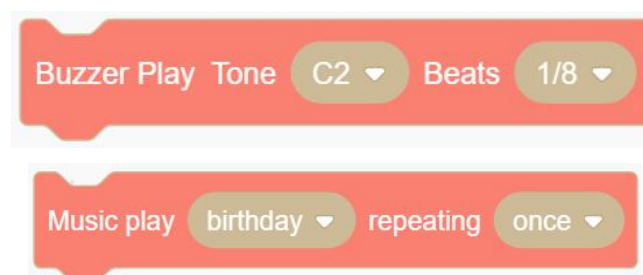
The serial port prints character strings or sensor data. The default baud rate is 115200.

RGB light



Set color or color value of RGB light

Buzzer



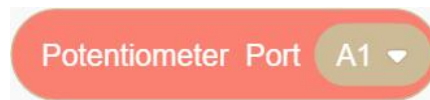
The buzzer module plays the corresponding beat tone or music. The tone CDEFGAB indicates different pitches, and 2345678 indicates same pitches, from low to high. For example, 4, 5, and 6 are low, medium, and high respectively, C4 < C5 < C6, C5 < D5 < E5 < F5 < G5 < A5 < B5.

Button



The key module outputs a Boolean value, which is one of “True” or “False”.

Potentiometer



Output potentiometer data value

Rocker



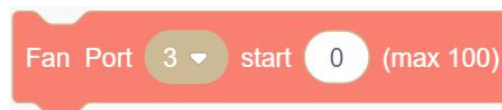
Output rocker data value

Fan Motor

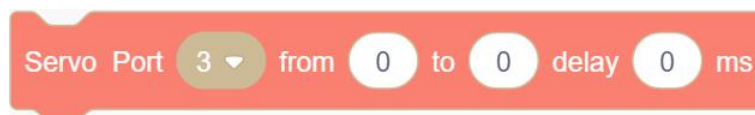


Set the fan module port and wind speed (wind speed range recommended 0 ~ 100)

Servo



Set port and angle of servo



Set the start angle and end angle of servo rotation, every 1 degree time interval

OLED

OLED Show string font(1~4) 1 X(0~128) 0 Y(0~32) 0 string Hello World!

OLED Show number font(1~4) 1 X(0~128) 0 Y(0~32) 0 number 123

OLED show static icon X(0~128) 0 Y(0~32) 0 circle ▾

OLED static display

OLED show move icon circle ▾

OLED show string Hello World! style static ▾ time 0

OLED show icon circle ▾ style static ▾ time 0

OLED dynamic display

OLED Show sensor font(1~4) 1 X(0~128) 0 Y(0~32) 0 string light: sensor 0

OLED display string and sensor data

OLED draw point X(0~128) 0 Y(0~32) 0

OLED draw line X1(0~128) 0 Y1(0~32) 0 X2(0~128) 0 Y2(0~32) 0

OLED Custom draw points and lines

OLED Show

OLED Clear

OLED display and clear

IR Control

A red rounded rectangle block with the text "infrared remote return value" in white.

IR control return value

A red rounded rectangle block with the text "infrared value" in white and a dropdown menu on the right showing "POWER" with a downward arrow.

Selection corresponding buttons for IR controller

Sound sensor

A red rounded rectangle block with the text "Sound sensor Port" in white and a dropdown menu on the right showing "A1" with a downward arrow.

Returns the value of the sound sensor

Photosensitive sensor

A red rounded rectangle block with the text "LDR Port" in white and a dropdown menu on the right showing "A1" with a downward arrow.

Returns the value of the photosensitive sound sensor

Hall sensor

A red rounded rectangle block with the text "Hall Sensor Port" in white and a dropdown menu on the right showing "A1" with a downward arrow.

Returns the value of the hall sensor

Vibration sensor

A red arrow-shaped block pointing right, containing the text "Shake sensor" in white, a dropdown menu showing "3" with a downward arrow, and the text "is pressed" in white.

Vibration sensor output boolean

Color recognition sensor



Color recognition sensor output boolean

color_sensor return red value(0~255)

color_sensor return green value(0~255)

color_sensor return blue value(0~255)

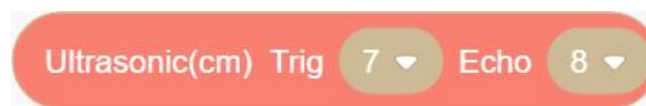
Color recognition sensor returns color values

Temperature and humidity sensor



Temperature and humidity sensor returns temperature and humidity value

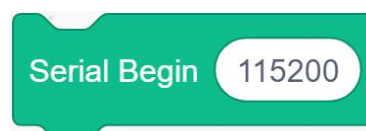
Ultrasonic sensor



Ultrasonic sensor returns ultrasonic value

7.2 Arduino category

Serial baud rate setting



Set the serial port baud rate. The default is 115200.

Determine whether the serial port has data

A green arrow-shaped block with the text "Serial Available" in white.

If data is detected on the serial port, return True otherwise return False.

Read serial port data

A green rounded rectangular block with the text "Serial Read" in white.

Read the data received by the serial port

Mapping function

A green rounded rectangular block with the text "Map" followed by a sequence of inputs: "100", "from", "0", "~", "255", "to", "0", "~", "1024".

Map a data value or variable from one range to another. The first range can be larger or smaller than the second range.

7.3 Control category

Wait 1s

An orange block with a notch on the left and a tab on the right. It contains the text "wait", a white circle with the number "1", and the text "seconds".

Wait (delay) function, default is 1s.

Repeated execution function (for loop)

An orange block with a notch on the left and a tab on the right. It contains the text "repeat", a white circle with the number "10", and a white curved arrow icon at the bottom right.

The function is executed repeatedly, which is repeated 10 times by default.

Repeat execution function (while loop)



Keep repeated execution

Conditional statement (if condition)



If the conditions are met, execute ...

Conditional statement (if...else...)



If the conditions are met, execute ... If conditions are not met, execute...

7.4 Operators category



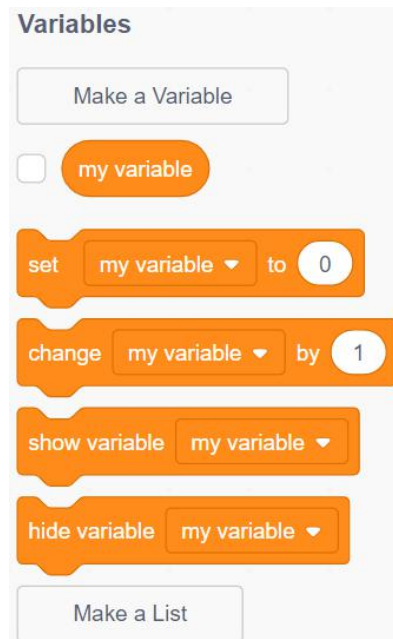
In the process of programming, we often need to judge some conditions. At this time, we need to use our operation instructions, including four operations, random number generation, size comparison, and, or functions. Operation instructions are often related to control instructions are used together.



7.5 Variables category

Variables are derived from mathematics and are abstract concepts in computer languages that can store calculation results or represent values. We can think of variables as a box, and the program can access the data in the box at any time. We use variables reasonably to improve programming efficiency and increase program readability.

In Helloblock, we can create variables, click to create a variable, and give the variable a name.



The created variables will be displayed below my variables, so we can use the variables we created, and my variables are created by default.

