

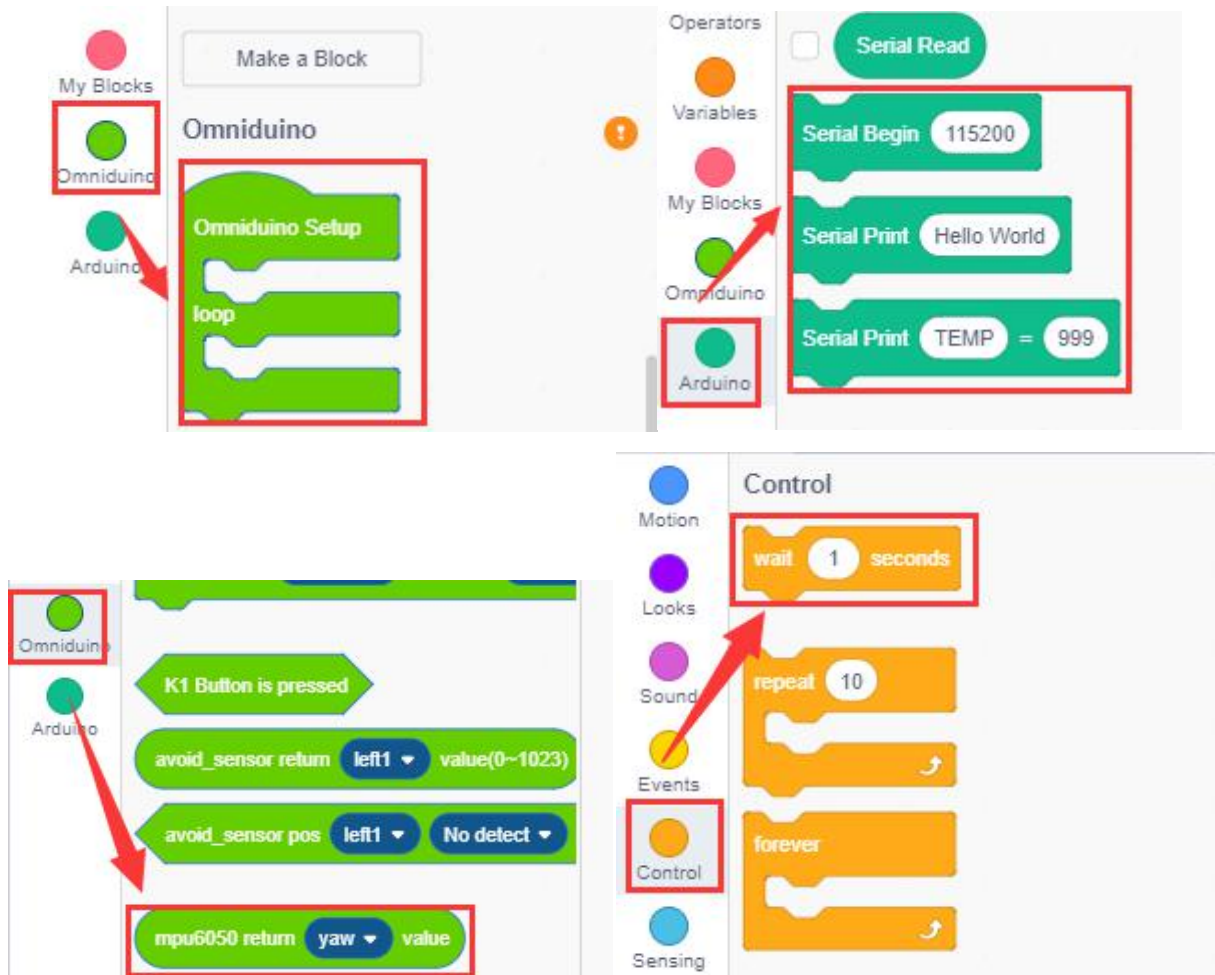
MPU6050 test

1. Learning goal

In this lesson, we will learn how to read MPU6050 data value(yaw,pitch,roll) and print them by serial port.

3. Looking for building blocks

The following is the location of the building blocks required for this programming.



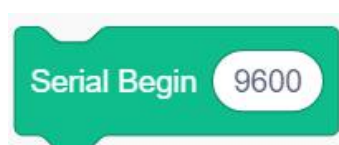
1) The content in the Omniduino setup block will only run once when the Omniduino is turned on or the reset button is pressed.

We can write into the initialization and other content in this block.

The content in the loop is the main loop function of the Omniduino car, most of the data processing and logic processing are completed in this function.

2) Select the serial port block in the Arduino category.

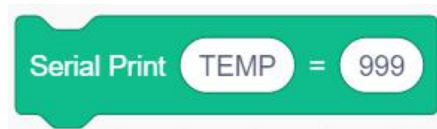
Serial Begin: select the baud rate



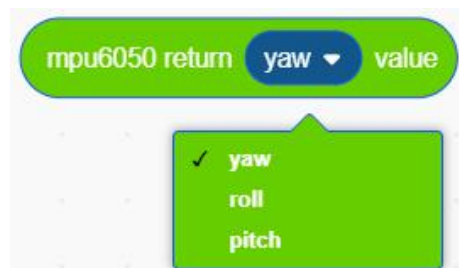
Serial Print: input what you want to print



Serial Print:



3) Get the value returned by MPU6050, you can choose different directions.



4) The function of waiting for the blocks is equivalent to the delay function in the program. We can enter different values according to our needs. (Unit: second)

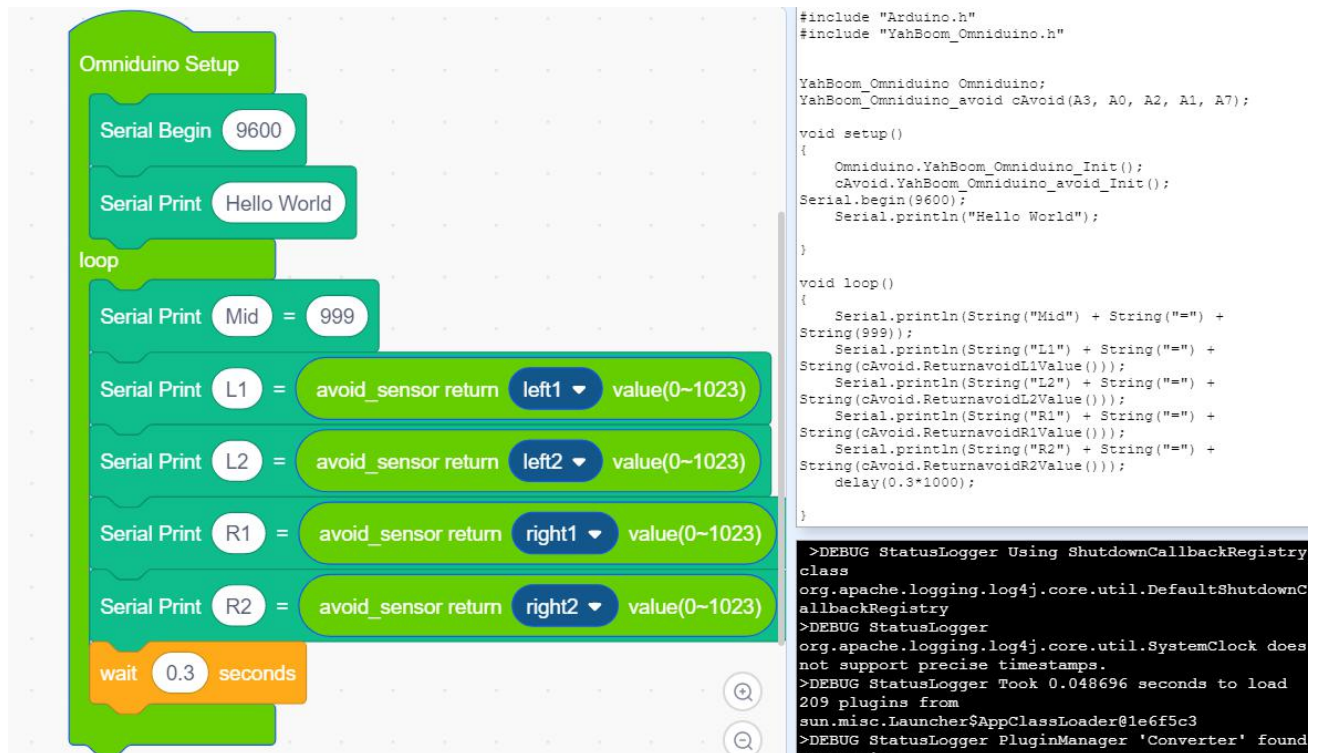


Combine blocks



3. Compiling and uploading the program

3.1 After building the blocks, click the **[code mode]** in the upper right corner of the Helloblock programming interface. We can see the corresponding Arduino code.



The left panel shows a block-based program for an Omniduino. It includes a 'Serial Begin' block set to 9600, a 'Serial Print' block with 'Hello World', a 'loop' block containing four 'Serial Print' blocks for sensor readings (Mid, L1, L2, R1, R2) and a 'wait' block for 0.3 seconds.

The right panel shows the corresponding Arduino code:

```
#include "Arduino.h"
#include "YahBoom_Omniduino.h"

YahBoom_Omniduino Omniduino;
YahBoom_Omniduino_avoid cAvoid(A3, A0, A2, A1, A7);

void setup()
{
  Omniduino.YahBoom_Omniduino_Init();
  cAvoid.YahBoom_Omniduino_avoid_Init();
  Serial.begin(9600);
  Serial.println("Hello World");
}

void loop()
{
  Serial.println(String("Mid") + String("=") +
String(999));
  Serial.println(String("L1") + String("=") +
String(cAvoid.ReturnavoidL1Value()));
  Serial.println(String("L2") + String("=") +
String(cAvoid.ReturnavoidL2Value()));
  Serial.println(String("R1") + String("=") +
String(cAvoid.ReturnavoidR1Value()));
  Serial.println(String("R2") + String("=") +
String(cAvoid.ReturnavoidR2Value()));
  delay(0.3*1000);
}
```

Below the code, a terminal window shows debug output from the StatusLogger, indicating that the program was successfully loaded and plugins were found.

3.2 Then, you need to connect Omniduino car to your computer. Select the CH340 port number identified in the previous step in the upper right corner. Then, click the up arrow to start compiling and uploading the program.



The screenshot shows the 'Stage mode' tab selected in the top right corner. The 'Chose COM' dropdown menu is open, showing 'COM13' as the selected port. A red arrow points to the 'upload' button (represented by an upward arrow icon) in the top right corner of the interface.

The left panel shows a block-based program for an Omniduino. It includes an 'Omniduino Setup' block, a 'loop' block, and an 'LED Set' block set to 'ON'.

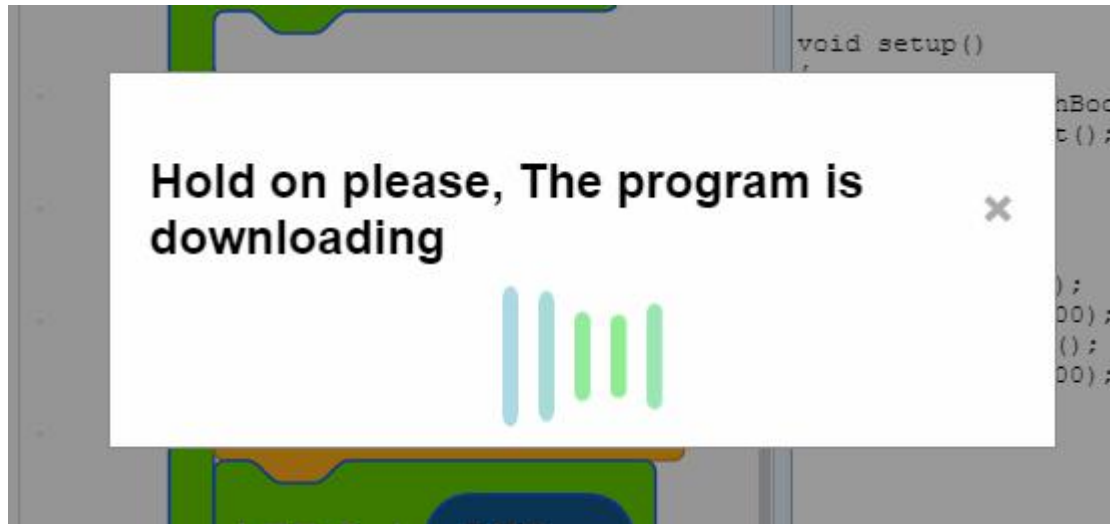
The right panel shows the corresponding Arduino code:

```
#include "Arduino.h"
#include "YahBoom_Omniduino.h"

YahBoom_Omniduino Omniduino;
YahBoom_Omniduino_LED cLED(5);

void setup()
{
  Omniduino.YahBoom_Omniduino_Init();
  cLED.LED_Init();
}

void loop()
{
  cLED.LED_ON();
}
```



3.3 When the words "**Done compiling Done uploading**" appear in the lower right corner of the programming interface, which means the program has been uploaded.

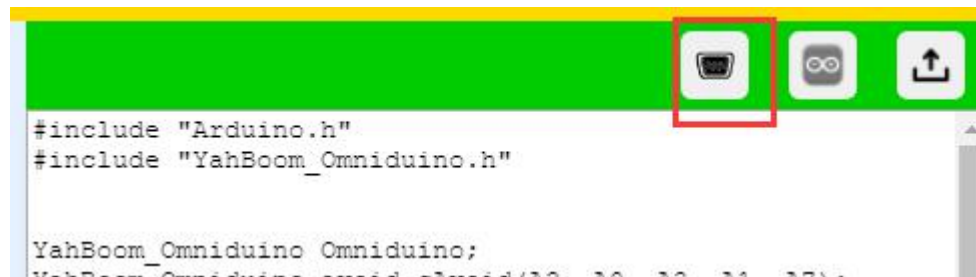
```
C:\Users\Administrator\AppData\Local\Arduino15\logs
/application.log
DEBUG StatusLogger All asynchronous threads have
terminated
DEBUG StatusLogger RollingFileManager shutdown
completed with status true
>DEBUG StatusLogger Shut down RollingFileManager
C:\Users\Administrator\AppData\Local\Arduino15\logs
/application.log, all resources released: true
>TRACE StatusLogger XmlConfiguration stopped 2
remaining Appenders.
TRACE StatusLogger XmlConfiguration cleaning
Appenders from 2 LoggerConfigs.
>DEBUG StatusLogger Stopped
XmlConfiguration[location=jar:file:/C:/Program%20Fi
les%20(x86)/HelloBlock/resources/Arduino/lib/pde.ja
r!/log4j2.xml] OK
>DEBUG StatusLogger Stopped
LoggerContext[name=1e6f5c3,
org.apache.logging.log4j.core.LoggerContext@16bc455
] with status true
>Done compiling. Done uploading!
```

4. Experimental phenomenon

After the program is downloaded. **It takes about 5~6 seconds to initialize the gyroscope.**

Wait for 5~6 seconds after powering on.

The gyroscope started to work normally, then we open the serial port to view the data value.



The serial port number selects the corresponding CH340 serial port, which is the same as the program download serial port.

Select 9600 for the baud rate, and then click "Open".

We can see that the value of the gyroscope will be printed out through the serial port.

```
【2020-08-14 11:23:49:442】mpu_x=-0.17
【2020-08-14 11:23:49:746】mpu_y=0.22
【2020-08-14 11:23:50:049】mpu_z=-0.01
【2020-08-14 11:23:50:358】mpu_x=0.02
【2020-08-14 11:23:50:662】mpu_y=0.16
【2020-08-14 11:23:50:968】mpu_z=-0.05
【2020-08-14 11:23:51:272】mpu_x=0.52
【2020-08-14 11:23:51:578】mpu_y=-0.07
【2020-08-14 11:23:51:885】mpu_z=-0.34
【2020-08-14 11:23:52:184】mpu_x=0.47
【2020-08-14 11:23:52:492】mpu_y=-0.07
【2020-08-14 11:23:52:793】mpu_z=-0.02
```