

Unit Pbhub v1.1 Arduino Tutorial

1. Preparations

- Environment Setup: Refer to [Arduino IDE Getting Started Guide](#) to complete IDE installation, and install corresponding board manager and required driver libraries based on the development board used.
- Required Driver Libraries:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit-HUB](#)
- Required Hardware Products:
 - [CoreS3](#)
 - [Unit Pbhub v1.1](#)
 - [Unit RGB](#) (Only used when testing RGB functions)
 - [Servo Kit 180°](#) (Only used when testing servo functions)



2. Notes

Pin Compatibility

Since pin configurations vary across host devices, please refer to the [Pin Compatibility Table](#) in the product documentation before use, and modify the sample program according to your actual pin connections.

Pin Definition

This tutorial uses CoreS3 as the main controller paired with Unit Pbhub v1.1. This pin expansion module communicates via I2C. Modify pin definitions in the program according to actual circuit connections. The corresponding IOs after device connection are G2 (SDA) and G1 (SCL).

3. Analog Reading Experiment

- Read analog voltage values from all six channels



cpp

```
1  #include "M5Unified.h"
2  #include "M5UnitPbHub.h"
3
4  M5UnitPbHub pbhub;
5
6  void setup() {
7      M5.begin();
8      Serial.begin(115200);
9      M5.Display.fillScreen(WHITE);
10     M5.Display.setTextColor(BLACK);
11     M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
12     M5.Display.setCursor(0, 0);
13     if (!pbhub.begin(&Wire, UNIT_PBHUB_I2C_ADDR, 2, 1, 400000U)) {
14         Serial.println("Couldn't find Pbhub");
15         while (1) delay(1);
16     }
17     Serial.println("Pbhub v1.1 analog_read test");
18 }
19
20 void loop() {
21     // ch: 0-5
22     // index: 0-1
23     // status: 0/1
24     for (uint8_t ch = 0; ch < 6; ch++) {
25         // only one pin supports analog reading each channel
26         Serial.printf("ch:%d adc:%d\r\n", ch, pbhub.analogRead(ch));
27         M5.Display.printf("ch:%d adc:%d\r\n", ch, pbhub.analogRead(ch));
28         delay(500);
29     }
30     M5.Display.fillScreen(WHITE);
31     M5.Display.setCursor(0, 0);
32 }
```

4. Pin Read/Write Experiment

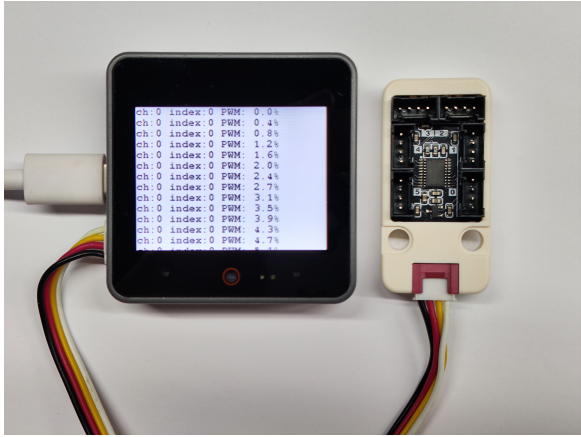
- Read the pin level status of different channels



```
1  #include "M5Unified.h"
2  #include "M5UnitPbHub.h"
3
4  M5UnitPbHub pbhub;
5
6  void setup() {
7      M5.begin();
8      Serial.begin(115200);
9      M5.Display.fillScreen(WHITE);
10     M5.Display.setTextColor(BLACK);
11     M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
12     M5.Display.setCursor(0, 0);
13     if (!pbhub.begin(&Wire, UNIT_PBHUB_I2C_ADDR, 2, 1, 400000U)) {
14         Serial.println("Couldn't find Pbhub");
15         while (1) delay(1);
16     }
17     Serial.println("Pbhub v1.1 digital write read test");
18 }
19
20 void loop() {
21     // ch: 0-5
22     // index: 0-1
23     // status: 0/1
24     for (uint8_t ch = 0; ch < 6; ch++) {
25         for (uint8_t index = 0; index < 2; index++) {
26             pbhub.digitalWrite(ch, index, 1);
27             delay(300);
28             Serial.printf("ch:%d index:%d\r\n", pbhub.digitalRead(ch, index),
29                           index);
30             M5.Display.printf("ch:%d index:%d\r\n", pbhub.digitalRead(ch, index),
31                               index);
32             delay(300);
33             pbhub.digitalWrite(ch, index, 0);
34             Serial.printf("ch:%d index:%d\r\n", pbhub.digitalRead(ch, index),
35                           index);
36             M5.Display.printf("ch:%d index:%d\r\n", pbhub.digitalRead(ch, index),
37                               index);
38             delay(300);
39         }
40     }
41     M5.Display.fillScreen(WHITE);
42     M5.Display.setCursor(0, 0);
43 }
```

5. Pin Duty Cycle Setting Experiment

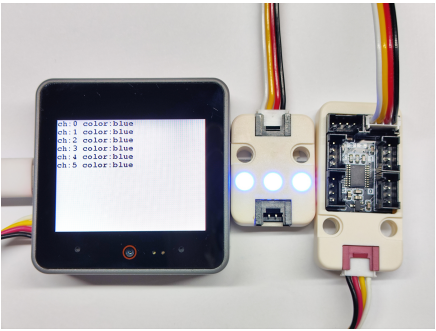
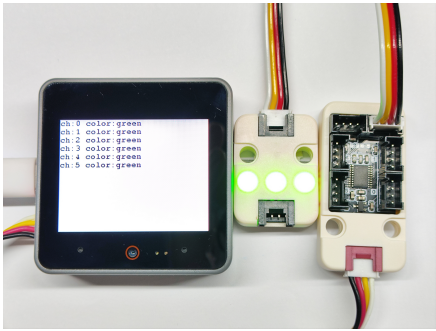
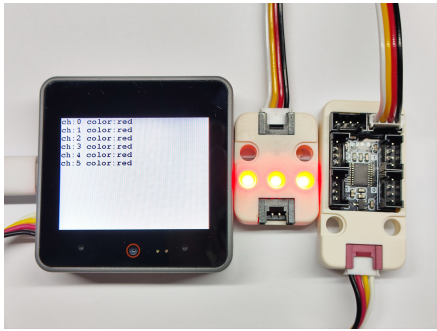
- Set and read duty cycles for different channel pins



```
1  #include "M5Unified.h"
2  #include "M5UnitPbHub.h"
3
4  M5UnitPbHub pbhub;
5
6  void setup() {
7      M5.begin();
8      M5.Display.fillScreen(WHITE);
9      M5.Display.setTextColor(BLACK);
10     M5.Display.setFont(&font::FreeMonoBold9pt7b);
11     M5.Display.setCursor(0, 0);
12     Serial.begin(115200);
13     if (!pbhub.begin(&Wire, UNIT_PBHUB_I2C_ADDR, 2, 1, 400000U)) {
14         Serial.println("Couldn't find Pbhub");
15         while (1) delay(1);
16     }
17     Serial.println("Pbhub v1.1 PWM test");
18 }
19
20 void loop() {
21     // ch: 0-5
22     // index: 0-1
23     // status: 0/1
24     for (uint8_t ch = 0; ch < 6; ch++) {
25         for (uint8_t index = 0; index < 2; index++) {
26             for (uint8_t h = 0; h < 255; h++) {
27                 pbhub.setPWM(ch, index, h);
28                 Serial.printf("ch:%d index:%d PWM: %.1f%%\r\n", ch, index,
29                             h / 255.0 * 100.0);
30                 M5.Display.printf("ch:%d index:%d PWM: %.1f%%\r\n", ch, index,
31                                 h / 255.0 * 100.0);
32                 delay(1000);
33             }
34         }
35         M5.Display.fillScreen(WHITE);
36         M5.Display.setCursor(0, 0);
37     }
38     delay(1000);
39 }
```

6. RGB Module Experiment

- Connect external Unit RGB module for color display



```
1  #include "M5Unified.h"
2  #include "M5UnitPbHub.h"
3
4  M5UnitPbHub pbhub;
5
6  void setup()
7  {
8      M5.begin();
9      Serial.begin(115200);
10     M5.Display.fillScreen(WHITE);
11     M5.Display.setTextColor(BLACK);
12     M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
13     M5.Display.setCursor(0, 0);
14     if (!pbhub.begin(&Wire, UNIT_PBHUB_I2C_ADDR, 2, 1, 100000U)) {
15         Serial.println("Couldn't find Pbhub");
16         while (1) delay(1);
17     }
18     for (uint8_t ch = 0; ch < 6; ch++) {
19         pbhub.setLEDNum(ch, 74);
20         pbhub.setLEDBrightness(ch, 40);
21     }
22 }
23
24 void loop()
25 {
26     // ch: 0-5
27     // index: 0-1
28     // status: 0/1
29     for (uint8_t ch = 0; ch < 6; ch++) {
30         pbhub.fillLEDColor(ch, 0, 73, 0xff0000);    // Red
31         Serial.printf("ch:%d color:red\r\n", ch);
32         M5.Display.printf("ch:%d color:red\r\n", ch);
33     }
34     delay(1000);
35     M5.Display.fillScreen(WHITE);
36     M5.Display.setCursor(0, 0);
37     for (uint8_t ch = 0; ch < 6; ch++) {
38         pbhub.fillLEDColor(ch, 0, 73, 0x00ff00);    // Green
39         Serial.printf("ch:%d color:green\r\n", ch);
40         M5.Display.printf("ch:%d color:green\r\n", ch);
41     }
42     delay(1000);
43     M5.Display.fillScreen(WHITE);
44     M5.Display.setCursor(0, 0);
45     for (uint8_t ch = 0; ch < 6; ch++) {
46         pbhub.fillLEDColor(ch, 0, 73, 0x0000ff);    // Blue
47         Serial.printf("ch:%d color:blue\r\n", ch);
48         M5.Display.printf("ch:%d color:blue\r\n", ch);
49     }
50     delay(1000);
```

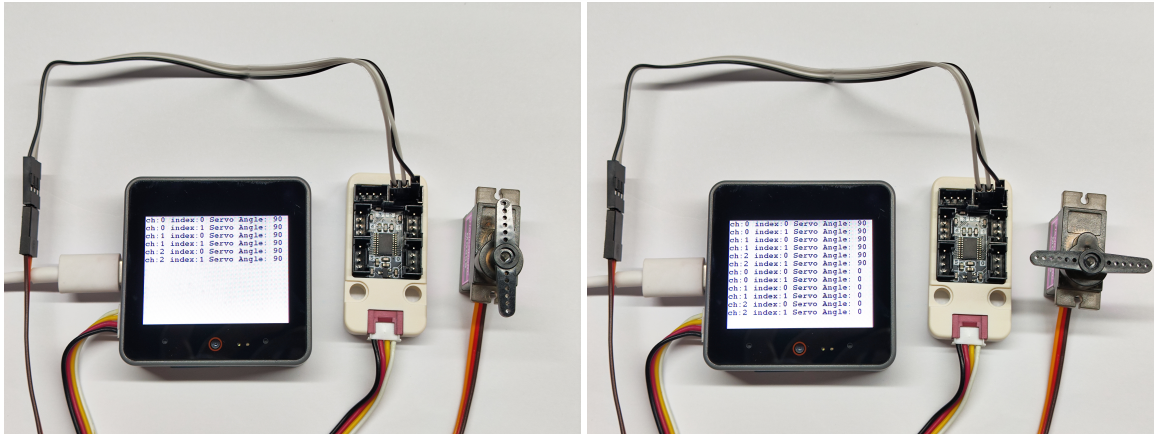
```
51 M5.Display.fillScreen(WHITE);  
52 M5.Display.setCursor(0, 0);  
53 }
```

7. Servo Control Experiment

- Control servo motor to rotate between $0^{\circ} \sim 90^{\circ}$

Circuit Connection

For servo connection: yellow wire is signal, red wire is 5V, brown wire is GND. Refer to the pinout diagram on the back of Unit Pbhub v1.1 when connecting.



```
1  #include "M5Unified.h"
2  #include "M5UnitPbHub.h"
3  M5UnitPbHub pbhub;
4
5  void setup() {
6      M5.begin();
7      M5.Display.fillScreen(WHITE);
8      M5.Display.setTextColor(BLACK);
9      M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
10     M5.Display.setCursor(0, 0);
11     Serial.begin(115200);
12     if (!pbhub.begin(&Wire, UNIT_PBHUB_I2C_ADDR, 2, 1, 400000U)) {
13         Serial.println("Couldn't find Pbhub");
14         while (1) delay(1);
15     }
16     Serial.println("Pbhub v1.1 Servo test");
17 }
18
19 void loop() {
20     // ch: 0-5
21     // index: 0-1
22     // status: 0/1
23     for (uint8_t ch = 0; ch < 3; ch++) {
24         for (uint8_t index = 0; index < 2; index++) {
25             pbhub.setServoAngle(ch, index, 90);
26             Serial.printf("ch:%d index:%d Servo Angle: 90\r\n", ch, index);
27             M5.Display.printf("ch:%d index:%d Servo Angle: 90\r\n", ch, index);
28         }
29     }
30     delay(2000);
31     for (uint8_t ch = 0; ch < 3; ch++) {
32         for (uint8_t index = 0; index < 2; index++) {
33             pbhub.setServoAngle(ch, index, 0);
34             Serial.printf("ch:%d index:%d Servo Angle: 0\r\n", ch, index);
35             M5.Display.printf("ch:%d index:%d Servo Angle: 0\r\n", ch, index);
36         }
37     }
38     delay(2000);
39     M5.Display.fillScreen(WHITE);
40     M5.Display.setCursor(0, 0);
41 }
```

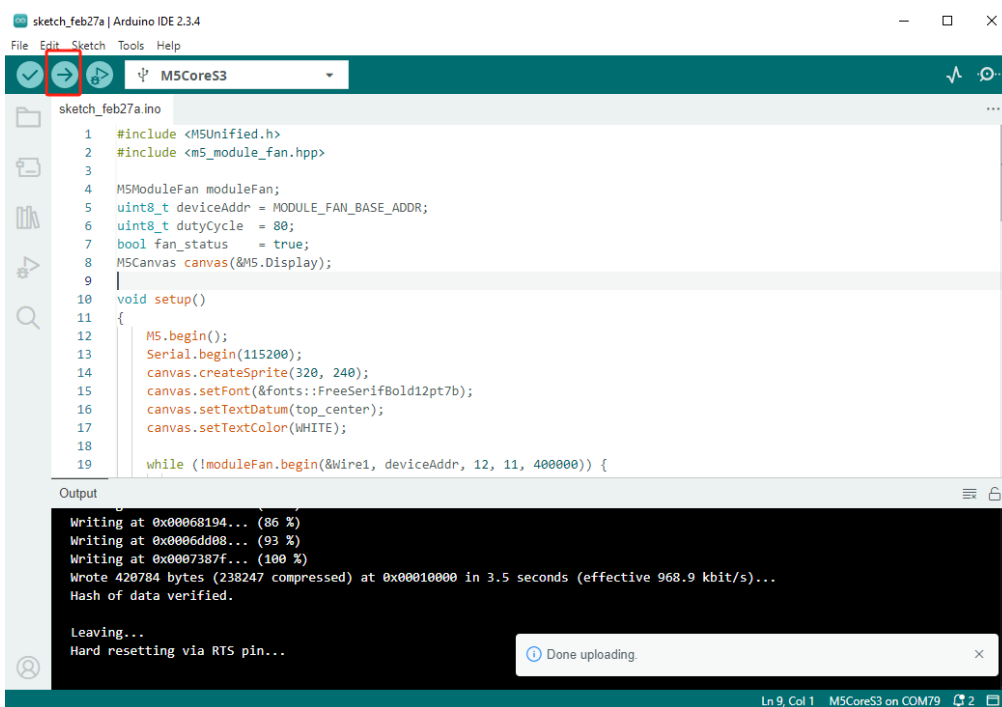
8. Compile and Upload

- Download Mode: Different devices need to enter download mode before programming. This step may vary depending on the main controller. For details, refer to the device-specific programming tutorial list at the bottom of the [Arduino IDE Getting Started Guide](#) page.

- For CoreS3: Press and hold the reset button (about 2 seconds) until the internal green LED lights up, then release. The device is now in download mode and ready for programming.



- Select the device port and click the upload button in the top-left corner of Arduino IDE. Wait for the program to compile and upload to the device.

A screenshot of the Arduino IDE 2.3.4 interface. The top toolbar shows the 'Upload' button (a right-pointing arrow) highlighted with a red box. The 'Tools' menu is open, showing 'M5CoreS3' selected. The main editor window displays the code for 'sketch_feb27a.ino'. The code includes headers for M5Unified and m5_module_fan, defines a module fan, and sets up a canvas for the display. The setup function initializes the module fan and the canvas. The main loop contains a while loop that calls moduleFan.begin() with specific parameters. The output window at the bottom shows the progress of the upload, including writing to memory and verifying the hash. A 'Done uploading.' dialog box is visible in the bottom right corner of the output window.

```
sketch_feb27a.ino
1  #include <M5Unified.h>
2  #include <m5_module_fan.hpp>
3
4  M5ModuleFan moduleFan;
5  uint8_t deviceAddr = MODULE_FAN_BASE_ADDR;
6  uint8_t dutyCycle = 80;
7  bool fan_status = true;
8  M5Canvas canvas(&M5.Display);
9
10 void setup()
11 {
12     M5.begin();
13     Serial.begin(115200);
14     canvas.createSprite(320, 240);
15     canvas.setFont(&fonts::FreeSerifBold12pt7b);
16     canvas.setTextDatum(top_center);
17     canvas.setTextColor(WHITE);
18
19     while (!moduleFan.begin(&Wire1, deviceAddr, 12, 11, 400000)) {
```

Output

```
Writing at 0x00068194... (86 %)
Writing at 0x0006dd08... (93 %)
Writing at 0x0007387f... (100 %)
Wrote 420784 bytes (238247 compressed) at 0x00010000 in 3.5 seconds (effective 968.9 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

Done uploading.
```

Ln 9, Col 1 M5CoreS3 on COM79