

Unit EXT.IO2 Arduino Tutorial

1. Preparations

- Environment Setup: Refer to the [Arduino IDE Getting Started Guide](#) to complete IDE installation, and install the corresponding board manager and required driver libraries based on the development board being used.
- Required Driver Libraries:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit-EXTIO2](#)

Note

You need to download the latest library version from GitHub: [M5Unit-EXTIO2 - M5Stack GitHub](#). Do not download from Arduino Library. (For any questions, please refer to [this tutorial](#))

- Required Hardware Products:
 - [CoreS3](#)
 - [Unit EXT.IO2](#)



2. Notes

Pin Compatibility

Since each host device has different pin configurations, please refer to the [Pin Compatibility Table](#) in the product documentation before use, and modify the example program according to the actual pin connections.

3. Example Program

- The main controller device used in this tutorial is CoreS3 with Unit EXT.IO2. This IO expansion unit communicates via I2C protocol. Modify the pin definitions in the program according to the actual circuit connections. After device connection, the corresponding communication IOs are G1 (SDA) and G2 (SCL).
- Analog ADC Input Pins

```
1  #include <M5Unified.h>
2  #include "M5_EXTIO2.h"
3
4  M5_EXTIO2 extio;
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(&font::FreeMonoBold9pt7b);
12     M5.Display.setCursor(0, 0);
13     while (!extio.begin(&Wire, 2, 1,
14                         0x45)) {
15         Serial.println("extio Connect Error");
16         M5.Display.println("extio Connect Error");
17         delay(100);
18     }
19     extio.setAllPinMode(ADC_INPUT_MODE); // Set all pins to ADC input mode.
20 }
21
22 char info[50];
23
24 void loop() {
25     M5.Display.fillScreen(WHITE);
26     M5.Display.setCursor(0, 0);
27     M5.Display.println("ADC INPUT MODE");
28     M5.Display.println("FW VERSION: " + String(extio.getVersion()));
29     for (uint8_t i = 0; i < 8; i++) {
30         uint16_t adc =
31             extio.getAnalogInput(i, _12bit); // Get ADC value.
32         Serial.printf("CH:%d ADC: %d", i, adc);
33         M5.Display.fillRect(0, i * 20 + 40, 200, 15, WHITE);
34         M5.Display.setCursor(0, i * 20 + 40);
35         M5.Display.printf("CH:%d ADC: %d", i, adc);
36     }
37     vTaskDelay(1000);
38 }
```

- o Digital Output Pins


```
1  #include <M5Unified.h>
2  #include "M5_EXTIO2.h"
3
4  M5_EXTIO2 extio;
5
6  extio_io_mode_t mode = DIGITAL_OUTPUT_MODE;
7
8  void btnTask(void *pvParameters) {
9      while (1) {
10         if (M5.BtnA.wasPressed()) {
11             if (mode == DIGITAL_INPUT_MODE) {
12                 mode = DIGITAL_OUTPUT_MODE;
13             } else {
14                 mode = DIGITAL_INPUT_MODE;
15             }
16         }
17         M5.update();
18         vTaskDelay(80);
19     }
20 }
21
22 void setup() {
23     M5.begin();
24     Serial.begin(115200);
25     M5.Display.fillScreen(WHITE);
26     M5.Display.setTextColor(BLACK);
27     M5.Display.setTextFont(&fonts::FreeMonoBold9pt7b);
28     M5.Display.setCursor(0, 0);
29     while (!extio.begin(&Wire, 2, 1, 0x45)) {
30         Serial.println("extio Connect Error");
31         M5.Display.println("extio Connect Error");
32         delay(100);
33     }
34     extio.setAllPinMode(
35         DIGITAL_OUTPUT_MODE);
36     xTaskCreatePinnedToCore(btnTask, "btnTask",
37                             4096,
38                             NULL,
39                             1,
40                             NULL, 0);
41 }
42
43 char info[50];
44
45 void loop() {
46     if (mode == DIGITAL_INPUT_MODE) {
47         M5.Display.fillScreen(WHITE);
48         M5.Display.setCursor(0, 0);
49         M5.Display.println("DIGITAL INPUT MODE");
50         M5.Display.println("FW VERSION: " + String(extio.getVersion()));
```

```

51     for (uint8_t i = 0; i < 8; i++) {
52         if (extio.getDigitalInput(i)) {
53             M5.Display.fillRect(i * 30 + 30, 145, 28, 30, WHITE);
54         } else {
55             M5.Display.fillRect(i * 30 + 30, 145, 28, 30, WHITE);
56         }
57     }
58 } else if (mode == DIGITAL_OUTPUT_MODE) {
59     M5.Display.fillScreen(WHITE);
60     M5.Display.setCursor(0, 0);
61     M5.Display.println("DIGITAL OUTPUT MODE");
62     M5.Display.printf("FW VERSION: %s",String(extio.getVersion()));
63     for (uint8_t i = 0; i < 8; i++) {
64         extio.setDigitalOutput(i, HIGH);
65         M5.Display.fillRect(i * 30 + 30, 145, 28, 30, BLACK);
66         M5.Display.setCursor(i * 30 + 30, 145);
67         vTaskDelay(100);
68     }
69     for (uint8_t i = 0; i < 8; i++) {
70         extio.setDigitalOutput(i, LOW);
71         M5.Display.fillRect(i * 30 + 30, 145, 28, 30, BLACK);
72         M5.Display.setCursor(i * 30 + 30, 145);
73         vTaskDelay(100);
74     }
75 }
76 extio.setAllPinMode(mode);
77 vTaskDelay(100);
78 }

```

◦ Digital Input Pins

```
1  #include <M5Unified.h>
2  #include "M5_EXTIO2.h"
3
4  M5_EXTIO2 extio;
5
6  extio_io_mode_t mode = DIGITAL_OUTPUT_MODE;
7
8  void btnTask(void *pvParameters) {
9      while (1) {
10         if (M5.BtnA.wasPressed()) {
11             if (mode == DIGITAL_INPUT_MODE) {
12                 mode = DIGITAL_OUTPUT_MODE;
13             } else {
14                 mode = DIGITAL_INPUT_MODE;
15             }
16         }
17         M5.update();
18         vTaskDelay(80);
19     }
20 }
21
22 void setup() {
23     M5.begin();
24     Serial.begin(115200);
25     M5.Display.fillScreen(WHITE);
26     M5.Display.setTextColor(BLACK);
27     M5.Display.setTextFont(&fonts::FreeMonoBold9pt7b);
28     M5.Display.setCursor(0, 0);
29     while (!extio.begin(&Wire, 2, 1, 0x45)) {
30         Serial.println("extio Connect Error");
31         M5.Display.println("extio Connect Error");
32         delay(100);
33     }
34     extio.setAllPinMode(
35         DIGITAL_OUTPUT_MODE);
36     xTaskCreatePinnedToCore(btnTask, "btnTask",
37                             4096,
38                             NULL,
39                             1 ,
40                             NULL, 0);
41 }
42
43 char info[50];
44
45 void loop() {
46     M5.Display.fillScreen(WHITE);
47     M5.Display.setCursor(0, 0);
48     M5.Display.println("DIGITAL INPUT MODE");
49     M5.Display.println("FW VERSION: " + String(extio.getVersion()));
50     extio.setDigitalOutput(0, HIGH);
```

```
51     M5.Display.println("Set Pin_0 level to high");
52     M5.Display.printf("Pin_1 level is %s", (String)extio.getDigitalInput(1));
53     extio.setAllPinMode(mode);
54     vTaskDelay(100);
55 }
```

- SERVO Control

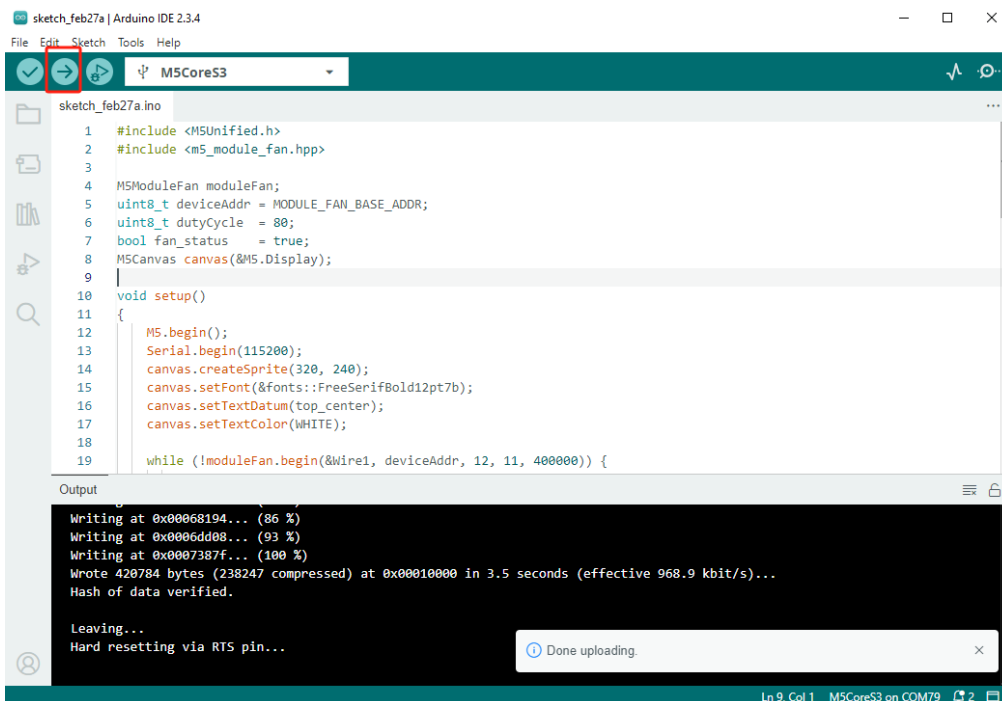
```
1  #include <M5Unified.h>
2  #include "M5_EXTIO2.h"
3
4  M5_EXTIO2 extio;
5
6  void setup() {
7      M5.begin();
8      Serial.begin(115200);
9      M5.Display.fillScreen(WHITE);
10     M5.Display.setTextColor(BLACK);
11     M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
12     M5.Display.setCursor(0, 0);
13     while (!extio.begin(&Wire, 2, 1, 0x45)) {
14         Serial.println("extio Connect Error");
15         M5.Display.println("extio Connect Error");
16         delay(100);
17     }
18     extio.setAllPinMode(SERVO_CTL_MODE);
19 }
20
21 char info[50];
22
23 void loop() {
24     M5.Display.fillScreen(WHITE);
25     M5.Display.setCursor(0, 0);
26     M5.Display.println("SERVO CTL MODE");
27     M5.Display.printf("FW VERSION: %s", String(extio.getVersion()));
28     for (uint8_t deg = 0; deg <= 180; deg += 45) {
29         for (uint8_t i = 0; i < 8; i++) {
30             extio.setServoAngle(i, deg);
31             Serial.printf("CH:%d DEG: %d", i, deg);
32             M5.Display.fillRect(0, i * 20, 200, i * 20 + 20, WHITE);
33             M5.Display.setCursor(0, i * 20);
34             M5.Display.printf("CH:%d DEG: %d", i, deg);
35         }
36         vTaskDelay(500);
37     }
38     for (int pulse = 500; pulse <= 2500; pulse += 100) {
39         for (uint8_t i = 0; i < 8; i++) {
40             extio.setServoPulse(i, pulse);
41             Serial.printf("CH:%d P: %d", i, pulse);
42             M5.Display.fillRect(0, i * 20, 200, i * 20 + 20, WHITE);
43             M5.Display.setCursor(0, i * 20);
44             M5.Display.printf("CH:%d P: %d", i, pulse);
45         }
46         vTaskDelay(500);
47     }
48     delay(100);
49 }
```

4. Compile and Upload

- Download Mode: Different devices need to enter download mode before programming. This process may vary depending on the main controller. For details, please refer to the device 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 will enter download mode and wait for programming.



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

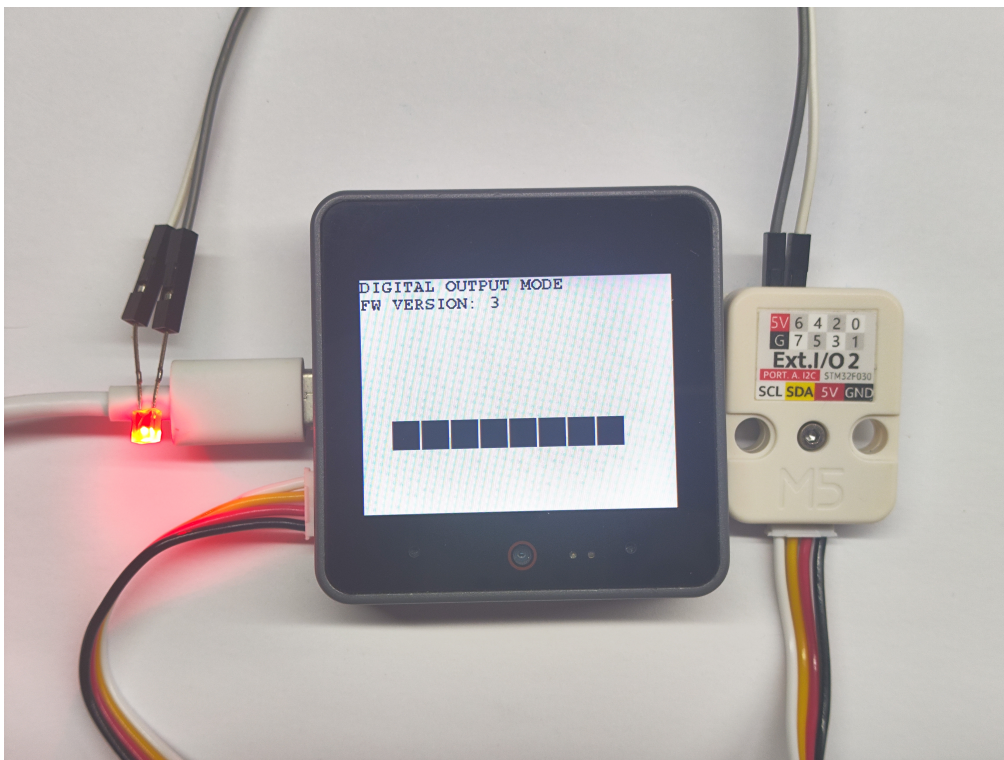


5. Extended Effects

- In ADC_INPUT mode, analog values from each channel can be read.



- In digital output mode, connect an LED to visualize the pin's level changes.



- In digital input mode, one pin outputs high level and another pin reads it to determine the level state.



- In SERVO mode, the pin can output a PWM signal to drive a servo.

