

Faces Gamepad3 Arduino Tutorial

1. Preparation

- Environment Setup: Follow the [Arduino IDE Getting Started Tutorial](#) to install the IDE, then install the board manager package and driver libraries for the development board you are using.
- Required Driver Libraries:
 - [M5Unified](#)
 - [M5Faces](#)
- Required Hardware:
 - [CoreS3](#)
 - [Faces Gamepad3](#)
 - [Faces Bottom3](#)



2. Notes

Pin Compatibility

Faces Gamepad3 connects to the host device through the Faces Bottom3 M5-Bus. M5-Bus pin definitions vary between host devices, so check the pin compatibility table below and update the example for your actual pin connections before use.

3. Example Program

- This tutorial uses a CoreS3 as the host device. Insert Faces Gamepad3 into the panel connector on Faces Bottom3, then connect Faces Bottom3 to CoreS3 through M5-Bus. Faces Gamepad3 communicates with the host over I2C; after connection, the I2C IO pins are `G12 (SDA)` and `G11 (SCL)`, and the interrupt pin is `G10 (INT)`.

cpp

```

1  #include <Arduino.h>
2  #include <M5Faces.h>
3  #include <M5Unified.h>
4
5  namespace {
6
7  M5Faces_Gamepad3 gamepad;
8
9  void draw_round_button(int x, int y, int w, int h, const char* label, bool pressed)
10 {
11     const uint16_t fill = pressed ? TFT_GREEN : TFT_DARKGREY;
12     const uint16_t text = pressed ? TFT_BLACK : TFT_WHITE;
13     M5.Display.fillRoundRect(x, y, w, h, 8, fill);
14     M5.Display.drawRoundRect(x, y, w, h, 8, TFT_WHITE);
15     M5.Display.setTextColor(text, fill);
16     M5.Display.drawString(label, x + w / 2, y + h / 2);
17 }
18
19 void draw_circle_button(int x, int y, int radius, const char* label, bool pressed)
20 {
21     const uint16_t fill = pressed ? TFT_GREEN : TFT_DARKGREY;
22     const uint16_t text = pressed ? TFT_BLACK : TFT_WHITE;
23     M5.Display.fillCircle(x, y, radius, fill);
24     M5.Display.drawCircle(x, y, radius, TFT_WHITE);
25     M5.Display.setTextColor(text, fill);
26     M5.Display.drawString(label, x, y);
27 }
28
29 void draw_gamepad_page()
30 {
31     M5.Display.fillScreen(TFT_BLACK);
32     M5.Display.setTextDatum(middle_center);
33     M5.Display.setTextColor(TFT_GREEN, TFT_BLACK);
34     M5.Display.drawString("Faces Gamepad3", M5.Display.width() / 2, 16);
35
36     // Symmetric D-pad layout.
37     draw_round_button(60, 51, 40, 34, "^", gamepad.isButtonPressed(GAMEPAD3_BTN_UP));
38     draw_round_button(20, 91, 40, 34, "<", gamepad.isButtonPressed(GAMEPAD3_BTN_LEFT));
39     draw_round_button(100, 91, 40, 34, ">", gamepad.isButtonPressed(GAMEPAD3_BTN_RIGHT));
40     draw_round_button(60, 131, 40, 34, "v", gamepad.isButtonPressed(GAMEPAD3_BTN_DOWN));
41
42     // Action buttons on the right.
43     draw_circle_button(246, 70, 22, "A", gamepad.isButtonPressed(GAMEPAD3_BTN_A));
44     draw_circle_button(204, 108, 22, "B", gamepad.isButtonPressed(GAMEPAD3_BTN_B));
45
46     // SELECT and START are centered as a pair.
47     draw_round_button(60, 176, 90, 34, "SELECT", gamepad.isButtonPressed(GAMEPAD3_BTN_SELECT));
48     draw_round_button(170, 176, 90, 34, "START", gamepad.isButtonPressed(GAMEPAD3_BTN_START));
49 }
50
51 void draw_error()
52 {
53     M5.Display.fillScreen(TFT_BLACK);
54     M5.Display.setTextDatum(middle_center);
55     M5.Display.setTextColor(TFT_GREEN, TFT_BLACK);

```

```

56     M5.Display.drawString("Faces Gamepad3", M5.Display.width() / 2, 80);
57     M5.Display.setTextColor(TFT_RED, TFT_BLACK);
58     M5.Display.drawString("init failed", M5.Display.width() / 2, 120);
59 }
60
61 } // namespace
62
63 void setup()
64 {
65     Serial.begin(115200);
66
67     auto config          = M5.config();
68     config.fallback_board = m5::board_t::board_M5StackCoreS3;
69     M5.begin(config);
70
71     if (!M5.In_I2C.isEnabled()) {
72         const int sda = M5.getPin(m5::pin_name_t::in_i2c_sda);
73         const int scl = M5.getPin(m5::pin_name_t::in_i2c_scl);
74         M5.In_I2C.begin(I2C_NUM_1, sda, scl);
75     }
76
77     M5.Display.setRotation(1);
78     M5.Display.setTextColor(TFT_GREEN, TFT_BLACK);
79     M5.Display.setFont(&fonts::FreeMonoBold12pt7b);
80     M5.Display.setTextSize(1);
81     M5.Display.setTextWrap(false);
82     M5.Display.setTextScroll(false);
83
84     if (gamepad.begin(&M5.In_I2C) != M5FACES_OK) {
85         Serial.println("Gamepad3 init failed");
86         draw_error();
87         return;
88     }
89
90     Serial.println("Gamepad3 ready");
91     draw_gamepad_page();
92 }
93
94 void loop()
95 {
96     if (gamepad.update()) {
97         const uint8_t raw = gamepad.getState().raw;
98         char buttons[64] = {};
99         M5Faces_Gamepad3::gamepad3_code_parse(raw, buttons, sizeof(buttons));
100        Serial.printf("Gamepad3 raw=0x%02X buttons=%s\n", raw, buttons[0] ? buttons : "NONE");
101        draw_gamepad_page();
102    }
103    delay(5);
104 }

```

4. Compile and Upload

In Arduino IDE, select the board and port for CoreS3, then click "Verify" to compile the program. After compilation succeeds, click "Upload" to flash it to CoreS3. When the upload is complete, open the Serial Monitor to view the output.



5. Reference Links

- [Faces Gamepad3 Product Documentation](#)
- [M5Faces Arduino Library](#)