

Faces Keyboard3 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 Keyboard3](#)
 - [Faces Bottom3](#)



2. Notes

Pin Compatibility

Faces Keyboard3 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 Keyboard3 into the panel connector on Faces Bottom3, then connect Faces Bottom3 to CoreS3 through M5-Bus. Faces Keyboard3 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  M5Faces_Keyboard3 faces;
6  String screen_text;
7
8  void redraw_screen()
9  {
10     M5.Display.fillScreen(TFT_BLACK);
11     M5.Display.setCursor(0, 0);
12     M5.Display.print(">");
13     M5.Display.print(screen_text);
14 }
15
16 void setup()
17 {
18     Serial.begin(115200);
19     screen_text.reserve(256);
20
21     auto config          = M5.config();
22     config.fallback_board = m5::board_t::board_M5StackCoreS3;
23     M5.begin(config);
24
25     const int i2c_sda = M5.getPin(m5::pin_name_t::in_i2c_sda);
26     const int i2c_scl = M5.getPin(m5::pin_name_t::in_i2c_scl);
27     if (!M5.In_I2C.isEnabled()) {
28         M5.In_I2C.begin(I2C_NUM_1, i2c_sda, i2c_scl);
29     }
30
31     M5.Display.setRotation(1);
32     M5.Display.setTextColor(TFT_GREEN, TFT_BLACK);
33     M5.Display.setFont(&fonts::FreeMonoBold12pt7b);
34     M5.Display.setTextSize(1);
35     M5.Display.setTextWrap(true);
36     M5.Display.setTextScroll(true);
37     redraw_screen();
38
39     if (faces.begin(&M5.In_I2C) != M5FACES_OK) {
40         Serial.println("Keyboard3 init failed");
41         M5.Display.print(" init failed");
42         return;
43     }
44
45     Serial.println("Keyboard3 ready");
46 }
47
48 void loop()
49 {
50     if (faces.update()) {
51         const uint8_t raw = faces.getKey();
52         const char value  = faces.getChar();
53         const char* name  = M5Faces_Keyboard3::keyboard3_code_parse(raw);
54
55         if (faces.isPrintable()) {

```

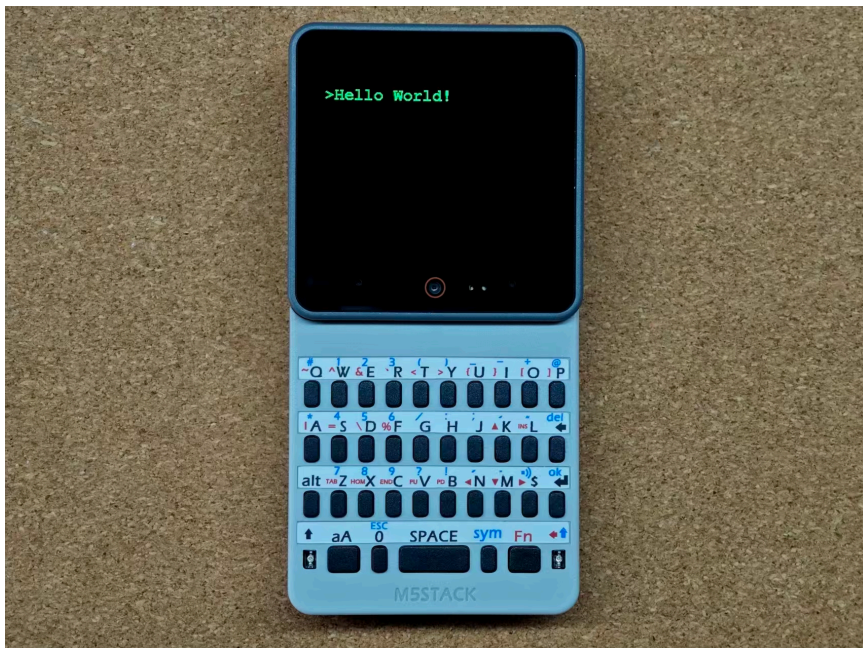
```

56     Serial.printf("raw=0x%02X key='%c'\n", raw, value);
57     screen_text += value;
58     redraw_screen();
59 } else {
60     Serial.printf("raw=0x%02X key=%s\n", raw, name ? name : "UNKNOWN");
61     if ((value == '\b' || raw == KEYBOARD3_KEY_DEL) && screen_text.length() > 0) {
62         screen_text.remove(screen_text.length() - 1);
63         redraw_screen();
64     } else if (value == '\n') {
65         screen_text += '\n';
66         redraw_screen();
67     } else if (value == '\t') {
68         screen_text += "    ";
69         redraw_screen();
70     }
71 }
72 }
73 delay(5);
74 }

```

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 Keyboard3 Product Documentation](#)
- [M5Faces Arduino Library](#)