

Dial Rotary Encoder

APIs and example programs related to the Dial rotary encoder.

Example Program

Build Requirements

- M5Stack board manager version $\geq 3.2.2$
- Board option = M5Dial
- M5Dial library version $\geq 1.0.3$

cpp

```
1  #include "M5Dial.h"
2
3  long oldPosition = -999;
4
5  void setup() {
6      auto cfg = M5.config();
7      M5Dial.begin(cfg, true, false); // encoder, RFID
8
9      M5Dial.Display.setTextColor(GREEN);
10     M5Dial.Display.setTextDatum(middle_center);
11     M5Dial.Display.setTextFont(&fonts::Orbitron_Light_32);
12     M5Dial.Display.setTextSize(2);
13 }
14
15 void loop() {
16     M5Dial.update();
17
18     long newPosition = M5Dial.Encoder.read();
19     if (newPosition != oldPosition) {
20         M5Dial.Speaker.tone(8000, 20);
21         M5Dial.Display.clear();
22         M5Dial.Display.drawString(String(newPosition),
23                                   M5Dial.Display.width() / 2,
24                                   M5Dial.Display.height() / 2);
25         oldPosition = newPosition;
26     }
27
28     if (M5Dial.BtnA.wasPressed()) {
29         M5Dial.Encoder.readAndReset();
30     }
31     if (M5Dial.BtnA.pressedFor(5000)) {
32         M5Dial.Encoder.write(100);
33     }
34 }
```

After compiling and uploading, the screen will display a number that follows the changes of the rotary encoder. Pressing the button briefly will reset the number to 0, and pressing and holding the button for 5 seconds will reset the number to 100.



| begin

Function Prototype:

```
void begin();
```

Description:

- Initialize the encoder IO.

When calling `M5Dial.begin()` , you can set the parameter `enableEncoder` to `true` to initialize it together.

```
M5Dial.begin(m5::M5Unified::config_t cfg, bool enableEncoder, bool enableRFID);
```

Input Parameters:

- null

Return Value:

- null

| read

Function Prototype:

```
int32_t read();
```

Description:

- Read the current encoder value.

Input Parameters:

- null

Return Value:

- `int32_t`: Current encoder value.

| write

Function Prototype:

```
void write(int32_t p);
```

Description:

- Write and update the encoder value.

Input Parameters:

- int32_t p: Updated encoder value.

Return Value:

- null

| readAndReset

Function Prototype:

```
int32_t readAndReset();
```

Description:

- Reset the current encoder value.

Input Parameters:

- null

Return Value:

- int32_t: Encoder value before reset