

Chain Encoder Tutorial

1.Preparation

- Environment Setup: Refer to the [Arduino IDE Getting Started Guide](#) to complete the IDE installation. Then install the corresponding board package and required driver libraries according to the development board you are using.
- Required Driver Library:
 - [M5Chain](#)



- Required Hardware Products:
 - [Chain DualKey](#)
 - Chain series connectors such as [Chain Bridge](#) or [Chain Return](#)
 - [Chain Encoder](#)



2.Example Program

Build Requirements

M5Stack Board Manager Version $\geq$ 3.2.4

M5Chain Library Version $\geq$ 1.0.0

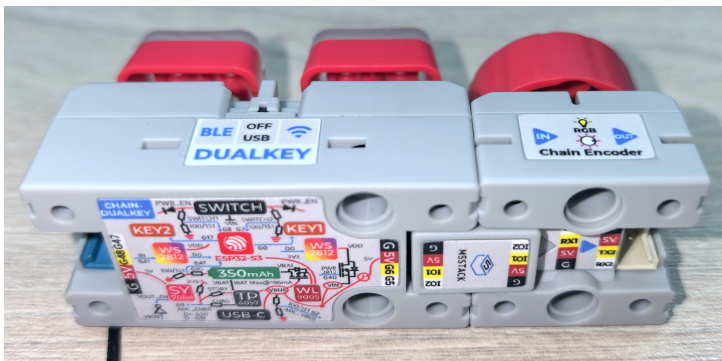
```
1  #include "M5Chain.h"
2
3  #define RXD_PIN GPIO_NUM_5  // 47 for the other side of Chain DualKey
4  #define TXD_PIN GPIO_NUM_6  // 48 for the other side of Chain DualKey
5
6  Chain M5Chain;
7
8  device_list_t *device_list = NULL;
9  uint16_t device_count = 0;
10 uint8_t opr_status = 0;
11
12 int16_t encoder_value;
13 int16_t encoder_incre;
14 uint8_t button_status;
15 chain_button_press_type_t button_press_type;
16
17 void setup() {
18     Serial.begin(115200);
19     delay(1000);
20     Serial.println("=====");
21     Serial.println("M5Stack Chain Encoder Test");
22
23     M5Chain.begin(&Serial2, 115200, RXD_PIN, TXD_PIN);
24     while (!M5Chain.isDeviceConnected()) {
25         Serial.println("No device connected");
26         delay(1000);
27     }
28
29     M5Chain.getDeviceNum(&device_count);
30     device_list = (device_list_t *)malloc(sizeof(device_list_t));
31     device_list->count = device_count;
32     device_list->devices = (device_info_t *)malloc(sizeof(device_info_t) * device_count);
33     M5Chain.getDeviceList(device_list);
34
35     if (device_list->devices[0].device_type == CHAIN_ENCODER_TYPE_CODE) {
36         Serial.println("ID[1] is Chain Encoder\n");
37         delay(1000);
38     } else {
39         Serial.println("ID[1] is NOT Chain Encoder\n");
40         return;
41     }
42
43     // Device ID, double click interval (100MS/200MS/.../900MS/1000MS), long press interval (3S/4S/...,
44     M5Chain.setEncoderButtonTriggerInterval(1, BUTTON_DOUBLE_CLICK_TIME_500MS, BUTTON_LONG_PRESS_TIME_!
45     Serial.println("Set double and long press intervals");
46
47     // Device ID, increment direction (AB = clockwise, BA = counterclockwise), operation status pointe
48     M5Chain.setEncoderABDirect(1, ENCODER_AB, &opr_status);
49     Serial.println("Set increment direction\n");
50 }
```

```

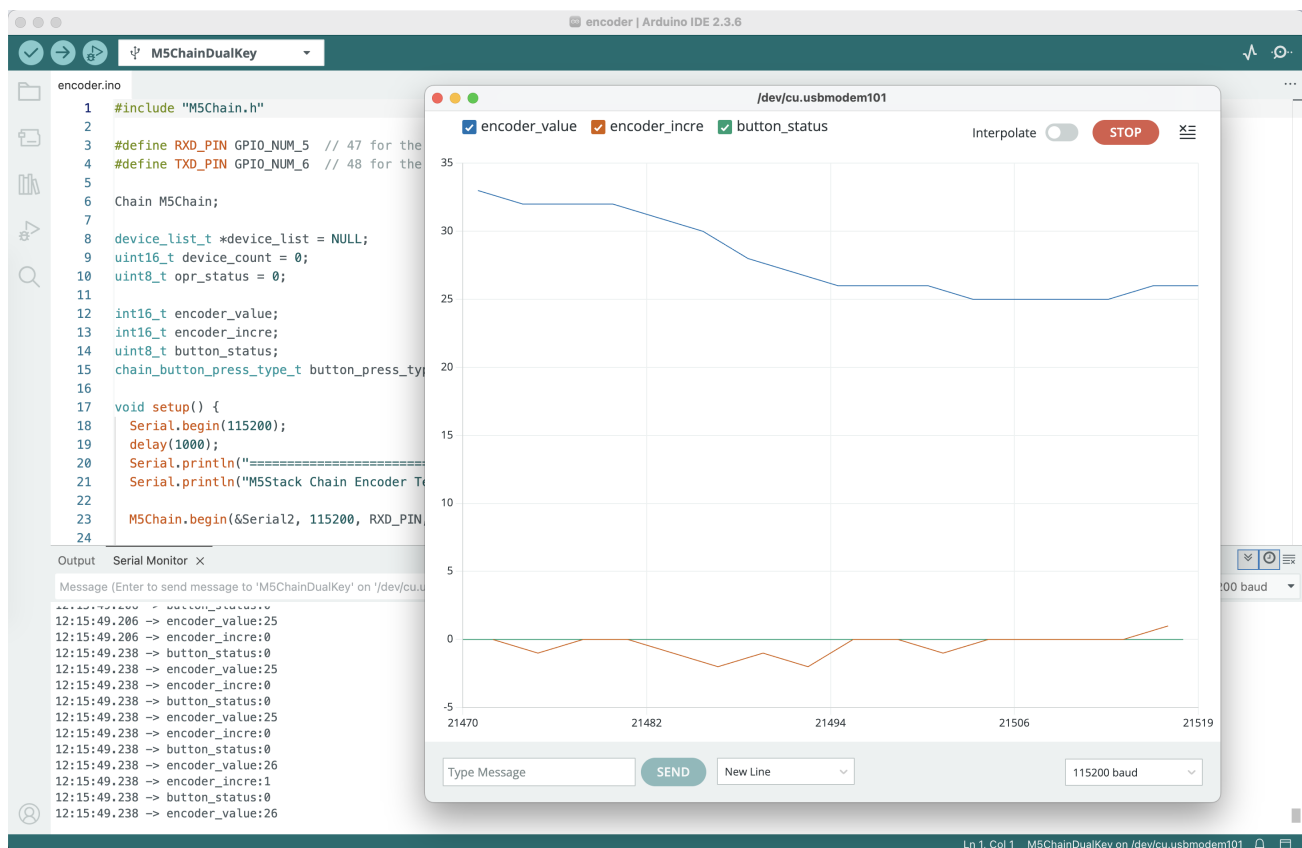
51   delay(1000);
52 }
53
54 void loop() {
55     M5Chain.getEncoderValue(1, &encoder_value); // Device ID
56     Serial.print("encoder_value:");
57     Serial.println(encoder_value);
58
59     M5Chain.getEncoderIncValue(1, &encoder_incre); // Device ID
60     Serial.print("encoder_incre:");
61     Serial.println(encoder_incre);
62
63     M5Chain.getEncoderButtonStatus(1, &button_status); // Device ID
64     Serial.print("button_status:");
65     Serial.println(button_status);
66
67     while (M5Chain.getEncoderButtonPressStatus(1, &button_press_type)) { // Device ID
68         switch (button_press_type) {
69             case CHAIN_BUTTON_PRESS_SINGLE:
70                 Serial.println("Single pressed");
71                 M5Chain.resetEncoderValue(1, &opr_status); // Device ID
72                 break;
73             case CHAIN_BUTTON_PRESS_DOUBLE:
74                 Serial.println("Double pressed");
75                 break;
76             case CHAIN_BUTTON_PRESS_LONG:
77                 Serial.println("Long pressed");
78                 break;
79         }
80     }
81 }

```

Use the Chain Bridge connector to link the main controller Chain DualKey with Chain Encoder. When connecting, make sure the direction is correct: the triangular arrow should point outward from the main controller Chain DualKey, as shown below:



Compile and upload the above program to the device. Click the button in the upper-right corner of Arduino IDE to open Serial Monitor and Serial Plotter. Rotate the knob on Chain Encoder and you will see the following information:



- **encoder_value**: current value of the knob
- **encoder_incre**: change amount of the knob in this step
- **button_status**: whether the knob is pressed
- Single-click, double-click, and long-press events of the knob
- In this example, a single click on the knob resets **encoder_value** to 0
- At the end of the **setup** function, you can configure the trigger time for double-click and long-press, as well as the direction in which the knob value changes (clockwise to increase / counterclockwise to increase)

3.Reference Links

- M5Chain Lib - GitHub
- Chain Series Device Bus Communication