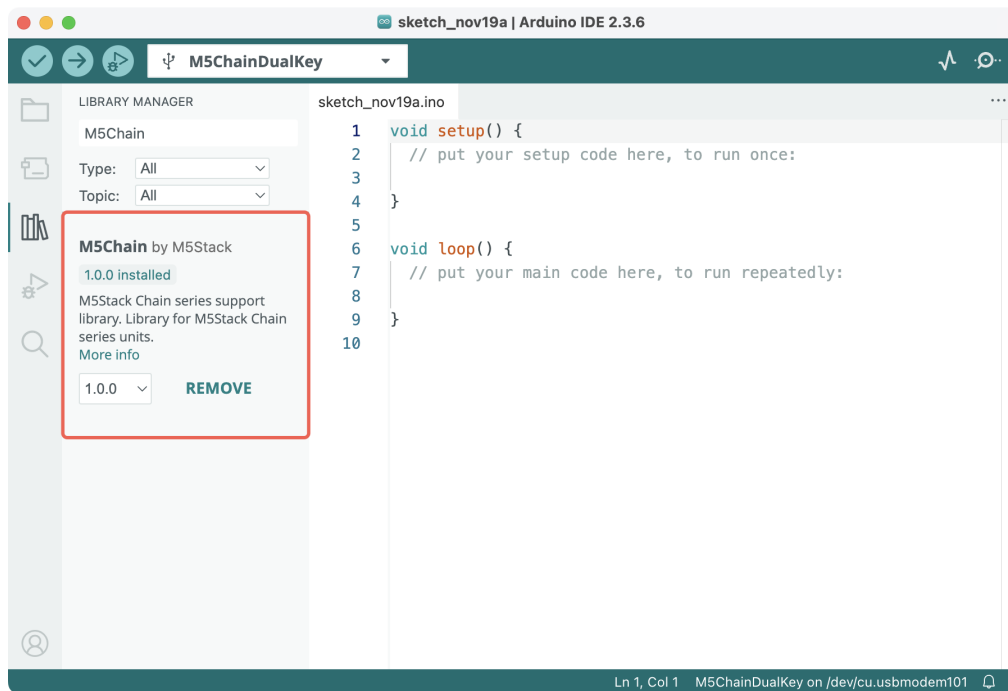


Chain Joystick 使用教程

1.准备工作

- 环境配置：参考 [Arduino IDE 上手教程](#) 完成 IDE 安装，并根据实际使用的开发板安装对应的板管理与需要的驱动库。
- 使用到的驱动库：
 - [M5Chain](#)



- 使用到的硬件产品：
 - [Chain DualKey](#)
 - Chain 系列连接器，如 [Chain Bridge](#) 或 [Chain Return](#)
 - [Chain Joystick](#)



2.示例程序

编译要求

M5Stack 板管理版本 $\geq 3.2.4$

M5Chain 库版本 $\geq 1.0.0$

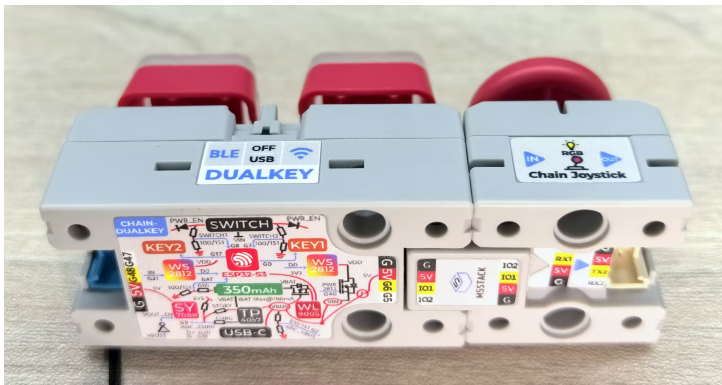
```
1  #include "M5Chain.h"
2
3  #define RXD_PIN GPIO_NUM_5 // 47 for the other side of Chain DualKey
4  #define TXD_PIN GPIO_NUM_6 // 48 for the other side of Chain DualKey
5
6  Chain M5Chain;
7
8  device_list_t *device_list = NULL;
9  uint16_t device_count = 0;
10
11 int16_t x_value, y_value;
12 uint8_t button_status, opr_status;
13 chain_button_press_type_t button_press_type;
14
15 void setup() {
16     Serial.begin(115200);
17     delay(1000);
18     Serial.println("=====");
19     Serial.println("M5Stack Chain Joystick Test");
20
21     M5Chain.begin(&Serial2, 115200, RXD_PIN, TXD_PIN);
22     while (!M5Chain.isDeviceConnected()) {
23         Serial.println("No device connected");
24         delay(1000);
25     }
26
27     M5Chain.getDeviceNum(&device_count);
28     device_list = (device_list_t *)malloc(sizeof(device_list_t));
29     device_list->count = device_count;
30     device_list->devices = (device_info_t *)malloc(sizeof(device_info_t) * device_count);
31     M5Chain.getDeviceList(device_list);
32
33     if (device_list->devices[0].device_type == CHAIN_JOYSTICK_TYPE_CODE) {
34         Serial.println("ID[1] is Chain Joystick\n");
35         delay(1000);
36     } else {
37         Serial.println("ID[1] is NOT Chain Joystick\n");
38         return;
39     }
40
41     // Device ID, double click interval (100MS/200MS/.../900MS/1000MS), long press interval (3S/4S/...
42     M5Chain.setJoystickButtonTriggerInterval(1, BUTTON_DOUBLE_CLICK_TIME_500MS, BUTTON_LONG_PRESS_TIME
43     Serial.println("Set double and long press intervals\n");
44     delay(1000);
45 }
46
47 void loop() {
48     M5Chain.getJoystickMappedInt16Value(1, &x_value, &y_value); // Device ID
49     Serial.print("x_value:");
50     Serial.println(x_value);
```

```

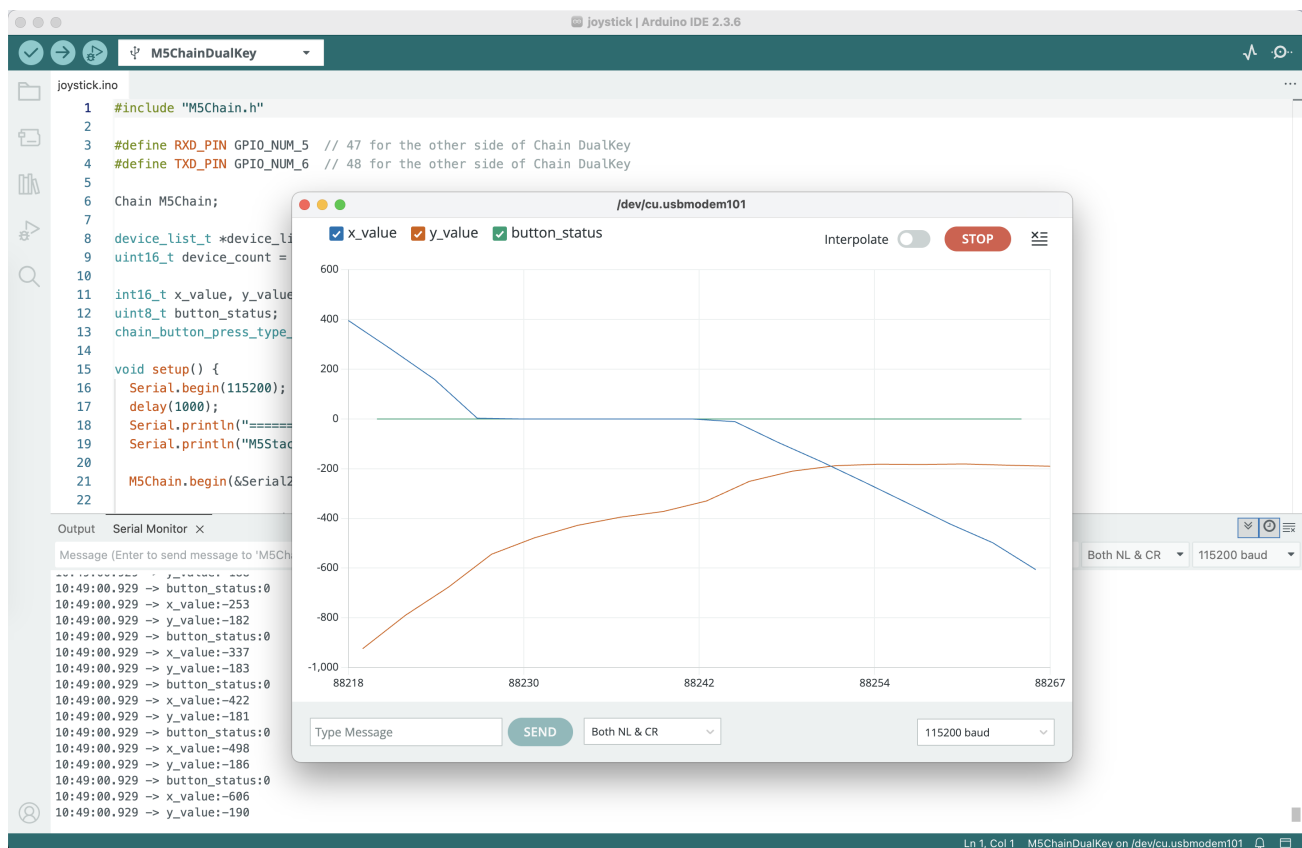
51 Serial.print("y_value:");
52 Serial.println(y_value);
53
54 M5Chain.getJoystickButtonStatus(1, &button_status); // Device ID
55 Serial.print("button_status:");
56 Serial.println(button_status);
57
58 while (M5Chain.getJoystickButtonPressStatus(1, &button_press_type)) { // Device ID
59     switch (button_press_type) {
60         case CHAIN_BUTTON_PRESS_SINGLE:
61             Serial.println("Single pressed");
62             break;
63         case CHAIN_BUTTON_PRESS_DOUBLE:
64             Serial.println("Double pressed");
65             break;
66         case CHAIN_BUTTON_PRESS_LONG:
67             Serial.println("Long pressed");
68             break;
69     }
70 }
71 }

```

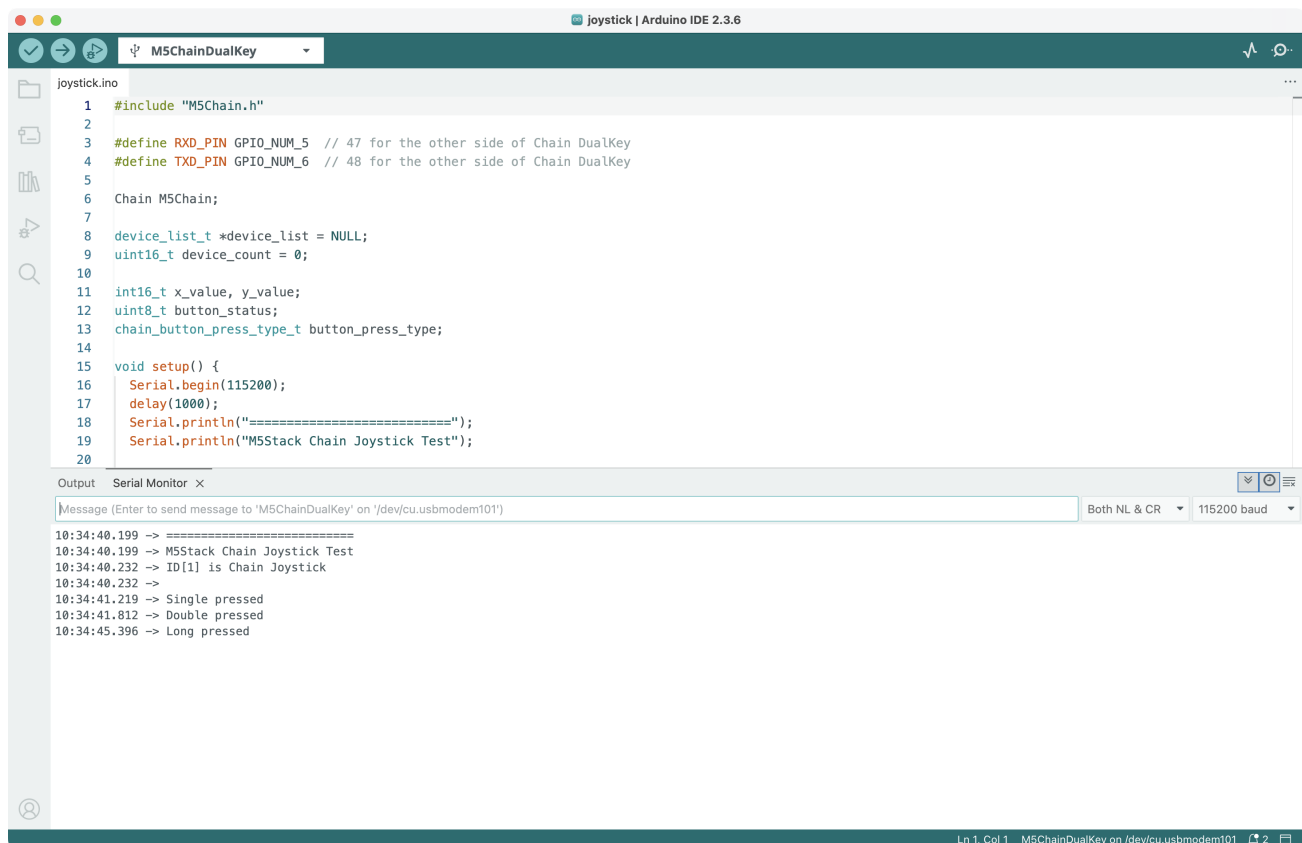
用 Chain Bridge 连接器连接主控 Chain DualKey 和 Chain Joystick。连接时需要注意方向，三角箭头从主控 Chain DualKey 指向外侧，如图：



将以上程序编译并上传至设备，点击 Arduino IDE 右上角的按钮打开串口绘图器（Serial Plotter）。移动 Chain Joystick 的摇杆，X、Y 值将实时显示在图表中：



注释掉以上程序 loop 部分的前两段（第 48-56 行）再次上传，打开串口监视器，程序会检测摇杆的按下事件（单击、双击、长按）并输出消息：



你还可以修改程序中的 `BUTTON_DOUBLE_CLICK_TIME_500MS`、`BUTTON_LONG_PRESS_TIME_5S` 为双击、长按事件设置不同的触发间隔时间。

3. 参考链接

- [M5Chain Lib - GitHub](#)
- [Chain 系列设备 Bus 通信](#)